

Workshop/Tutorial

ICRA14 Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots

Organizers: P. Martinet, K. Yoshida, M. Bergerman
<http://wmepc14.irccyn.ec-nantes.fr/>

Objectives

In robotics research, autonomy has presented a long standing challenge on many fronts, particularly for motion control. Mobile robot autonomy is dependent on a variety of technological solutions and their integration. Some key areas include enhanced traction methods, outdoor perception, localization, map building, obstacle avoidance, safety, maintenance etc. Great effort and resources have already been invested worldwide to develop such technologies to enable the autonomy of mobile robots. This workshop will focus on state-of-the-art developments in modeling, perception, estimation and control of all-terrain mobile robots. Mobility in outdoor unstructured environments remains a critical technology where many interesting research topics must be addressed. Precise modeling and estimation of the contact between tire and ground, localization and mapping in unstructured environment, robustness to the uncertainties of parameters and precise trajectory tracking in dynamic environment represent challenging issues for our scientific community. The proposed workshop will summarize existing results, exchange ongoing research and address future directions in these diverse areas.

Topics of interest

- Detection, tracking and classification
- Feature extraction and feature selection
- Cooperative techniques
- Collision prediction and avoidance
- Environment perception
- Robot localization
- Autonomous navigation
- Real-time perception and sensor fusion
- SLAM in dynamic environments
- Mapping and maps for navigation
- Real-time motion planning in dynamic Environments
- 3D Mapping, traversability
- Robust sensor-based 3D reconstruction
- Modeling and Control of mobile robot
- Machine Vision for Outdoor Environment
- 3D Sensing and Reconstruction of Outdoor Environment
- Robotics for Fruit/Vegetable production
- Guidance in Orchard Environment
- Sensing in partially cluttered partially structured environments
- Kinematic slip modeling and slip estimation
- Path tracking and control methods
- Off-road mobile robot control
- Mobile robot safety (integrity, traversability, stability)
- Low cost localization system
- Multi autonomous vehicles studies, models, techniques and simulations

- Trajectory/object tracking in hard conditions

This workshop is composed with 5 invited talks and 12 selected papers. Four sessions have been organized:

- Session I: Enhanced mobility & Modeling
- Session II: Perception in outdoor environment
- Session III: Navigation, Control, Planning
- Session IV: Cooperative Robots, Software architecture

Invited Speakers:

- Genya Ishigami, Keio University, Japan
- Paul Furgale, ETHZ, Switzerland
- Alberto Elfes, CSIRO, Australia
- Dimi Apostolopoulos, CMU, USA
- Roland Lenain, IRSTEA, France

This workshop is supported by IEEE RAS Technical Committees on :

- Autonomous Ground Vehicles and Intelligent Transportation Systems
- Space robotic
- Agricultural Robotics and Automation

Program

Time	Talk
08:30	Invited Talk: Rough Terrain Mobility: key issues and approaches for dynamics simulation of rough terrain mobile robot, Genya Ishigami (Dept. of Mechanical Engineering, Keio University, Japan)
09:10 – 9:28	Analyzing the Impact of Learning Inputs on Near-to-Far Terrain Traversability Estimation, K. Ho, T. Peynot and S. Sukkarieh
09:28 – 09:46	Terrain mapping with a pan and tilt stereo camera for locomotion on a quadruped robot, S. Bazeille, M. Camurri, J. Ortiz, I. Havoutis, D. G. Caldwell, and C. Semini
09:46 - 10:26	Invited Talk: There and back again: Dealing with highly-dynamic scenes and long-term change during topological/metric route following, Paul Furgale (ETH Zurich, Switzerland)
10:26 - 10:50	Coffee Break
10:50 – 11:08	Monocular Vision: A Real-Time Perception Toolkit for Mobile Robots in Outdoor Environments, A. Miranda Neto, A. C. Victorino and J. V. Ferreira
11:08 – 11:26	Multiplicative Extended Kalman Filter based on Visual Data for Attitude Estimation, A. Seba, A. El Hadri, L. Benziane, A. Benallegue
11:26 – 11:44	Gradient Based Multi-modal Sensor Calibration, Z. Taylor and J. Nieto
11:44 - 13:00	Lunch Break
13:00 - 13:40	Invited Talk: Multi-Modal Robot Perception and Augmented World Models, Alberto Elfes (Autonomous Systems Lab, CSIRO)
13:40 - 13:58	Localization of Planetary Exploration Rovers with Orbital Imaging: a survey of approaches, E. Boukas ¹ , A. Gasteratos ¹ and G. Visenti
13:58 - 14:16	A probabilistic distribution approach for the classification of urban roads in complex environments, G. B. Vitor, A. C. Victorino and J. V. Ferreira
14:16- 14:56	Invited Talk: Important Considerations in the Design of All-terrain Wheeled Robots, Dimi Apostolopoulos (Carnegie Mellon University, Robotics Institute, USA)
14:56 – 15:14	A Planner for All Terrain Vehicles on Unknown Rough Terrains based on the MPC Paradigm and D*- like Algorithm, A. Tahirovic, M. Brkic, G. Magnani and L. Bascetta
15:14 – 15:32	An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments, D. Alves de Lima and A. Correa Victorino
15:32 - 15:52	Coffee Break
15:52 - 16:10	Towards Lifelong Learning of Optimal Control for Kinematically Complex Robots, A. Dettmann ¹ , M. Langosz ² , Kai von Szadkowski ¹ , and S. Bartsch
16:10 - 16:50	Formation control of off-road fleet of UGVs: issues, advances and applications, Roland Lenain (IRSTEA, France)
16:50 – 17:08	Integrating ABSYNTH autonomous navigation system into ROS,

	A. Llamazares, E. Molinos, M. Ocana and F. Herranz
17:08 – 17:26	The Artemis Rover as an Example for Model Based Engineering in Space Robotics, J. Schwendner, T. M. Roehr, S. Haase, M. Wirkus, M. Manz, S. Arnold and J. Machowinski
17:30	End

Organizers

Professor Philippe Martinet, IRCCyN-CNRS Laboratory, Ecole Centrale de Nantes, 1 rue de la Noë, 44321 Nantes Cedex 03, France, Phone: +33 237406975, Fax: +33 237406934, Email: Philippe.Martinet@ircsyn.ec-nantes.fr, Home page: <http://www.ircsyn.ec-nantes.fr/~martinet>

Professor Kasuya Yoshida, Space Robotics Lab, Department of Aeronautics and Space Engineering, Tohoku University, Aoba 01, Sendai 980-8579, Japan Phone: +81 (22) 217-6992, Fax : + 81 (22) 795-6993, Email: yoshida@astro.mech.tohoku.ac.jp, Home page: http://db.tohoku.ac.jp/whois/e_detail/da1574fdf8452accf00e06c018e31523.html

Professor Marcel Bergerman, Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh PA 15213-3890 Phone: +1 (412) 513-6110 Fax : +1 412-206-0828, Email: marcel@cmu.edu, Home page: <http://www.cs.cmu.edu/~mbergerm/>

Contact

Professor Philippe Martinet, IRCCyN-CNRS Laboratory, Ecole Centrale de Nantes, 1 rue de la Noë, 44321 Nantes Cedex 03, France, Phone: +33 237406975, Fax: +33 237406934, Email: Philippe.Martinet@ircsyn.ec-nantes.fr, Home page: <http://www.ircsyn.ec-nantes.fr/~martinet>

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





ICRA14 Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots

Full Day Workshop

June 1st, 2014, Hong-Kong, China

<http://wmepc14.irccyn.ec-nantes.fr/>

Organizers

**Pr Philippe Martinet (IRCCyN, France),
Pr Kasuya Yoshida (Tohoku University, Japan)
Pr Marcel Bergerman (CMU, USA)**

Contact

Professor Philippe Martinet
Ecole Centrale de Nantes,
IRCCYN-CNRS Laboratory
1 rue de la Noë, BP 92101, 44321 Nantes Cedex - FRANCE
Phone: +33 240 376 975, Sec : +33 240 376 900, Fax : +33 240 376 6934
Email: Philippe.Martinet@irccyn.ec-nantes.fr
Home page: <http://www.irccyn.ec-nantes.fr/~martinet>

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Foreword

In robotics research, autonomy has presented a long standing challenge on many fronts, particularly for motion control. Mobile robot autonomy is dependent on a variety of technological solutions and their integration. Some key areas include enhanced traction methods, outdoor perception, localization, map building, obstacle avoidance, safety, maintenance etc. Great effort and resources have already been invested worldwide to develop such technologies to enable the autonomy of mobile robots. This workshop will focus on state-of-the-art developments in modeling, perception, estimation and control of all-terrain mobile robots. Mobility in outdoor unstructured environments remains a critical technology where many interesting research topics must be addressed. Precise modeling and estimation of the contact between tire and ground, localization and mapping in unstructured environment, robustness to the uncertainties of parameters and precise trajectory tracking in dynamic environment represent challenging issues for our scientific community. The proposed workshop will summarize existing results, exchange ongoing research and address future directions in these diverse areas.

Contributions will focus on the following topics:

- Detection, tracking and classification
- Feature extraction and feature selection
- Cooperative techniques
- Collision prediction and avoidance
- Environment perception
- Robot localization
- Autonomous navigation
- Real-time perception and sensor fusion
- SLAM in dynamic environments
- Mapping and maps for navigation
- Real-time motion planning in dynamic Environments
- 3D Mapping, traversability
- Robust sensor-based 3D reconstruction
- Modeling and Control of mobile robot
- Trajectory/object tracking in hard conditions
- Machine Vision for Outdoor Environment
- 3D Sensing and Reconstruction of Outdoor Environment
- Robotics for Fruit/Vegetable production
- Guidance in Orchard Environment
- Sensing in partially cluttered partially structured environments
- Kinematic slip modeling and slip estimation
- Path tracking and control methods
- Off-road mobile robot control
- Mobile robot safety (integrity, traversability, stability)
- Low cost localization system
- Multi autonomous vehicles studies, models, techniques and simulations

This workshop is composed with 5 invited talks and 12 selected papers. Four sessions have been organized:

- Session I: Enhanced mobility & Modeling
- Session II: Perception in outdoor environment
- Session III: Navigation, Control, Planning
- Session IV: Cooperative Robots, Software architecture

Marcel Bergerman, Philippe Martinet, Kasuya Yoshida

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





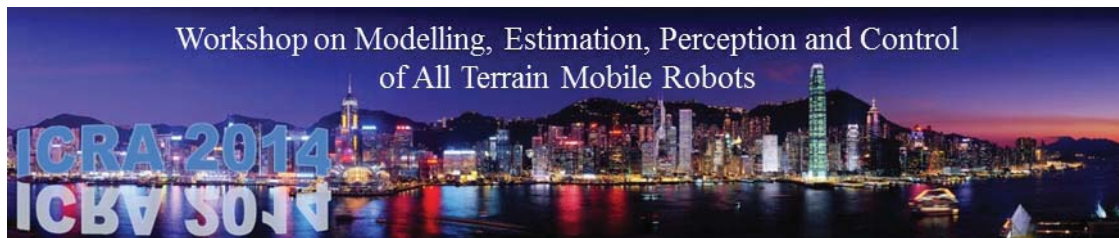
Session I

Enhanced mobility & Modeling

- **Invited Talk: Genya Ishigami (Keio University, Japan)**
Title: Rough Terrain Mobility: key issues and approaches for dynamics simulation of rough terrain mobile robot
- **Title: Analyzing the Impact of Learning Inputs on Near-to-Far Terrain Traversability Estimation**
Authors: K. Ho, T. Peynot and S. Sukkarieh
- **Title: Terrain mapping with a pan and tilt stereo camera for locomotion on a quadruped robot**
Author: S. Bazeille, M. Camurri, J. Ortiz, I. Havoutis, D. G. Caldwell, and C. Semini

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Session I

Invited Talk: **Genya Ishigami**
(Keio University, Japan)

Rough Terrain Mobility: key issues and approaches for dynamics simulation of rough terrain mobile robot

Abstract: Rough-terrain mobile robots are always subject to complicated dynamic interaction between their running gears (tire, wheel, or track) and ground. A well-defined mechanics for the robot-terrain interaction is of importance to the following technical aspects of the mobile robot: (1) mobility analysis such as slope traversability or obstacle crossing; (2) robot navigation, planning, and traction control; and (3) design of vehicle dimensions, suspension, and actuators. This presentation focuses on a topic related to a dynamics simulation of rough-terrain mobile robot using wheel contact mechanics. An overview of research and development of the dynamics simulation are described along with its application to mobility analysis, control, and design. In the presentation, typical issues and key approaches towards a next generation of dynamics simulation tools for rough-terrain mobile robot are also discussed.

Biography: Genya Ishigami received the PhD degrees from Tohoku University, Japan, in 2008. He was a postdoctoral associate at the Massachusetts Institute of Technology from 2008 to 2010, and a research associate at Japan Aerospace Exploration Agency from 2010 to 2013. He is currently an assistant professor at Keio University. His research interests are in the areas of mobility analysis, vehicle-terrain interaction mechanics, perception, navigation and control, for application to planetary exploration rovers, rescue robots, and field robots.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



ROUGH TERRAIN MOBILITY

Key Issues and Approaches for Dynamic Simulation of Rough Terrain Mobile Robot

Genya Ishigami
(Keio University, JAPAN)

Table of Contents

- Rough terrain mobile robot
 - Introduction to dynamic simulation
- Dynamic simulation
 - Wheel contact mechanics
 - Application to robotic analysis, control, & design
- Open issues
 - Limitation of classical model
 - Key approaches
- Conclusion
 - Dilemma: "*Accuracy or Cost* "
 - Towards next generation of dynamic simulation

Rough Terrain Mobile Robot: Introduction



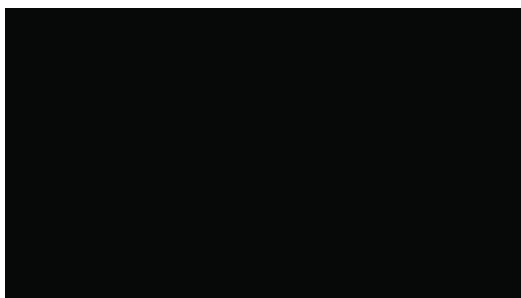
■ Rough terrain mobile robot applications:

- Transportation
- Disaster response
- Lunar/planetary exploration

■ Terrain properties

- Sand, mud, gravel, grass, wood, leaves, etc...
- Deformable, cluttered, loose, bumpy, irregular, steep terrain

Rough Terrain Mobile Robot: Introduction



■ Interaction between running gear and rough terrain

- Loss of traction, slippage, stuck, vehicle rollover

Dynamic simulation of rough terrain mobile robot

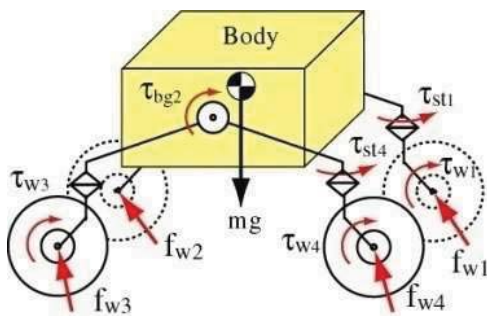


- Mobility prediction
- Trafficability analysis
- Traction control
- Vehicle design

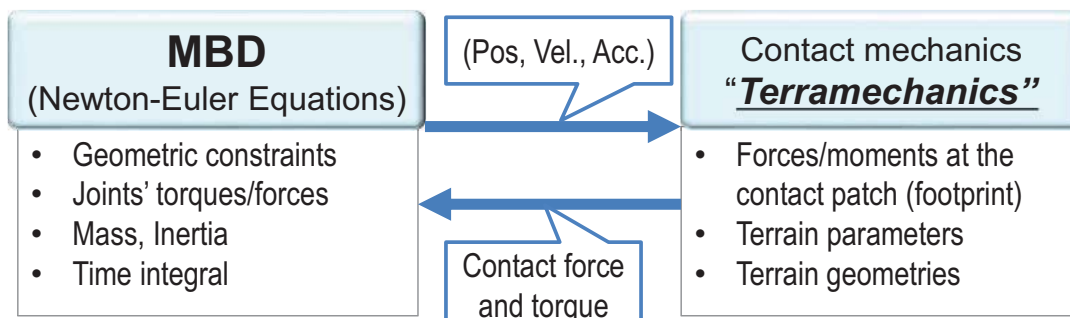
Table of Contents

- Rough terrain mobile robot
 - Introduction to dynamic simulation
- Dynamic simulation
 - Wheel contact mechanics
 - Application to robotic analysis, control, & design
- Open issues
 - Limitation of classical model
 - Key approaches
- Conclusion
 - Dilemma: “Accuracy or Cost”
 - Towards next generation of dynamic simulation

Dynamic simulation: General framework



- The interaction mechanics (or contact mechanics) is a “**subroutine**” in the multi-body dynamics calculation.



Contact Mechanics: “*Terramechanics*”

■ Terramechanics is:

- Study of the mechanical properties of the terrain and its response to vehicular loading.
- Interaction mechanics of off-road (wheeled or tracked) vehicles on various surfaces.

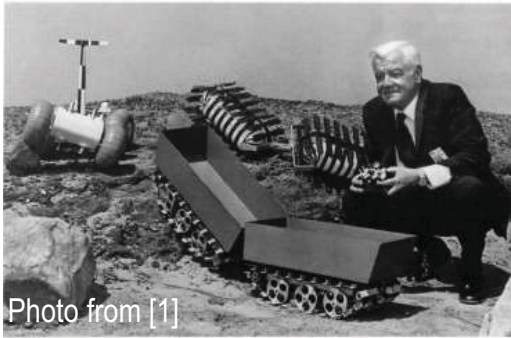
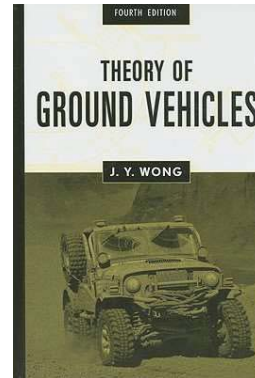


Photo from [1]

M.G. Bekker: Theory of Land Locomotion^[2]



J.Y. Wong: Theory of Ground Vehicles^[3]

Terramechanics: Basic Approach

Terrain properties
Vehicle parameters

- Friction, cohesion, stiffness
- Wheel size, load

Modelling of
stress distribution

- Semi-empirical model: Continuum model
 - Normal stress
 - ✓ Pressure-sinkage relationship
 - Shear stress
 - ✓ Mohr-Coulomb failure criterion

Contact forces and
torques

- Integral of the stress distribution around the contact patch

Terramechanics: Basic Approach

Terrain properties
Vehicle parameters

- Friction, cohesion, stiffness
- Wheel/track size, load

Soil type	Cohesion (kPa)	Friction angle (deg)	Sinkage exponent
Dry sand [3]	1.04	28.0	1.1
Sandy loam [3]	1.72	29.0	0.7
Clayey soil [3]	4.14	13.0	0.5
Regorith simulant [4]	0.8	37.2	1.0
Toyoura sand [4]	0.0	38.0	1.7

Terramechanics: Basic Approach

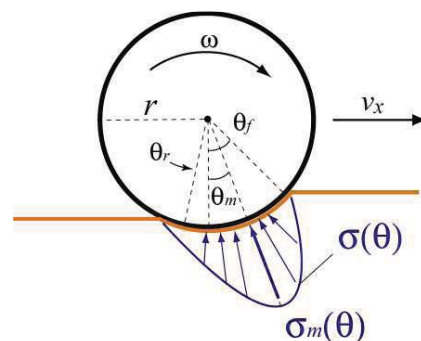
Terrain properties
Vehicle parameters

- Normal stress
 - Pressure-sinkage relationship (Wong and Reece, 1967)^[5]

Modelling of stress distribution

$$\sigma(\theta) = \begin{cases} r^n \left(\frac{k_c}{b} + k_\phi \right) [\cos \theta - \cos \theta_f]^n & (\theta_m \leq \theta < \theta_f) \\ r^n \left(\frac{k_c}{b} + k_\phi \right) \left[\cos \left\{ \theta_f - \frac{\theta - \theta_r}{\theta_m - \theta_r} (\theta_f - \theta_m) \right\} - \cos \theta_f \right]^n & (\theta_r < \theta \leq \theta_m) \end{cases}$$

Contact forces and torques



Terramechanics: Basic Approach

Terrain properties
Vehicle parameters

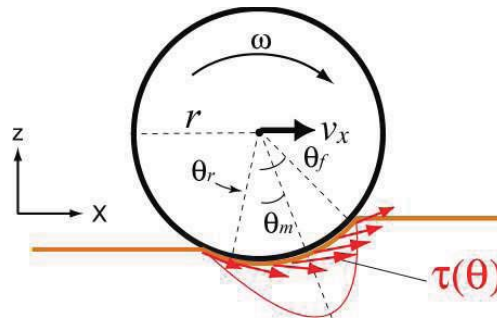
Modelling of
stress distribution

Contact forces and
torques

■ Shear stress

- Mohr-Coulomb failure criterion (Janosi and Hanamoto, 1961)^[6]

$$\tau(\theta) = (c + \sigma(\theta) \tan \phi) [1 - e^{-j(\theta)/k}]$$



Terramechanics: Basic Approach

Terrain properties
Vehicle parameters

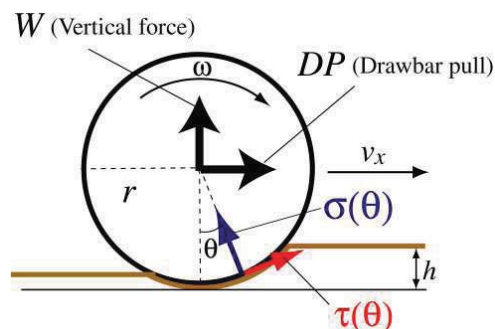
Modelling of
stress distribution

Contact forces and
torques

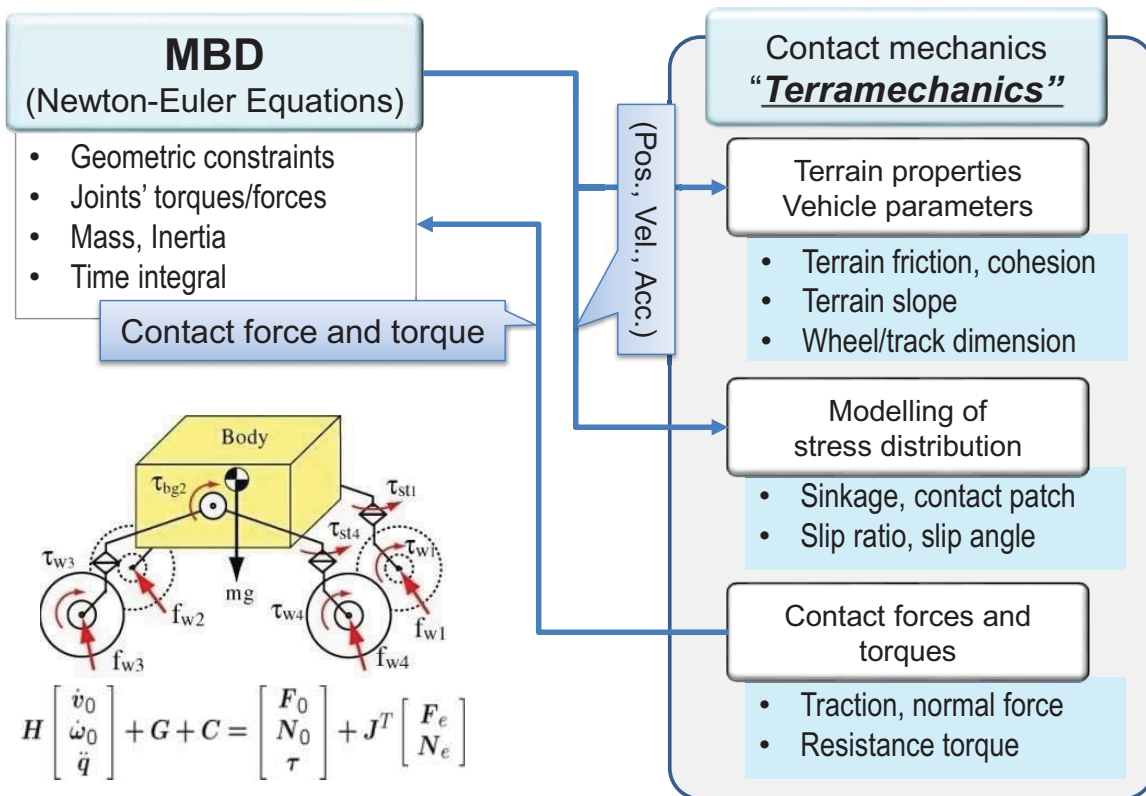
■ Integral of the stress distribution around the contact patch^[5]

$$DP = rb \int_{\theta_r}^{\theta_f} \{ \tau(\theta) \cos \theta - \sigma(\theta) \sin \theta \} d\theta$$

$$W = rb \int_{\theta_r}^{\theta_f} \{ \tau(\theta) \sin \theta + \sigma(\theta) \cos \theta \} d\theta$$



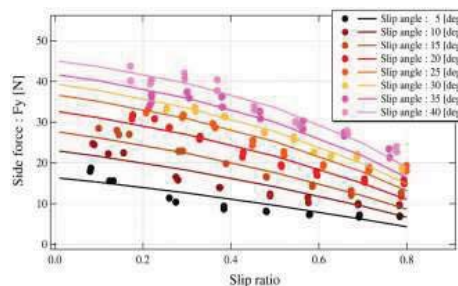
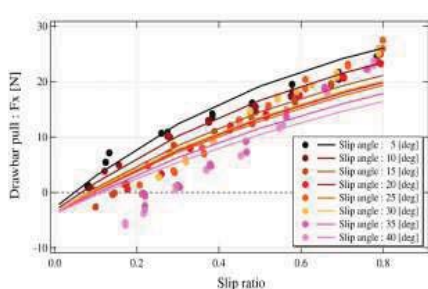
Dynamic simulation: General framework



Experimental Validation: Single wheel tests

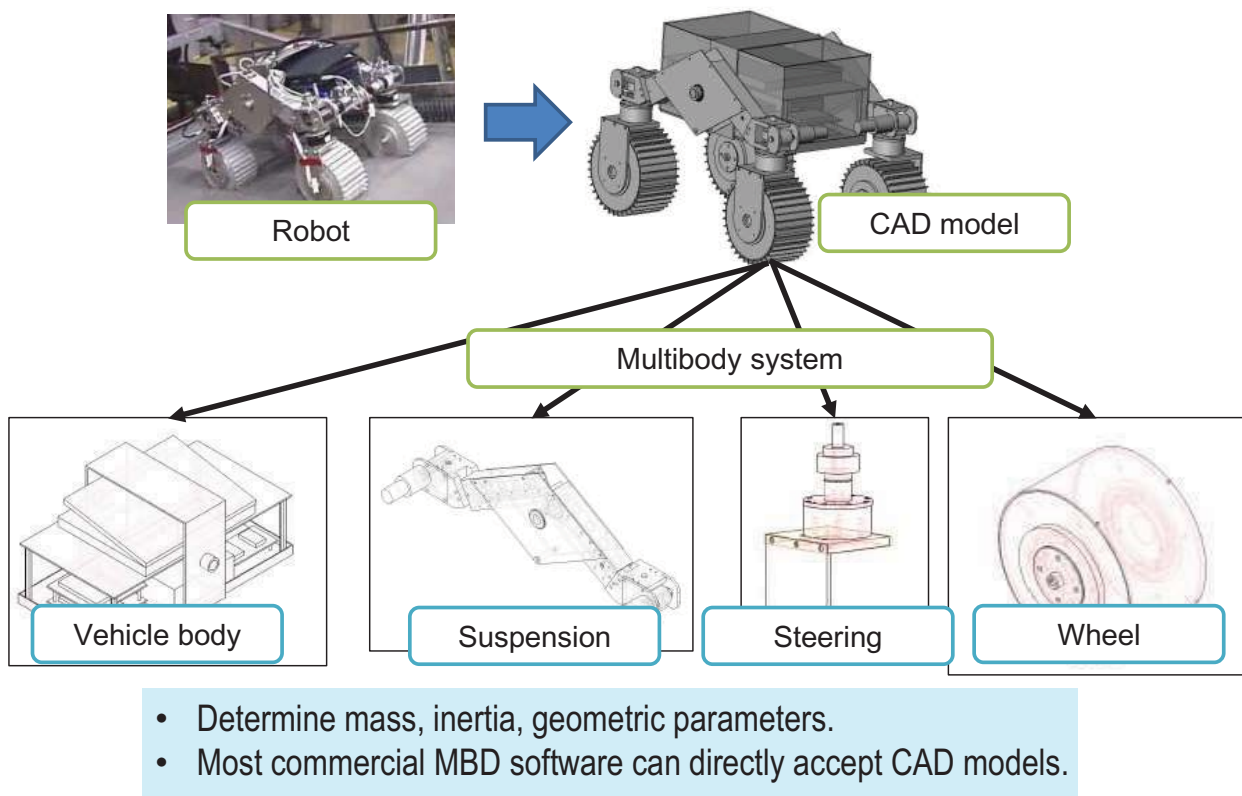


- Performs several 100s/1000s of experimental runs using single-wheel test bed.



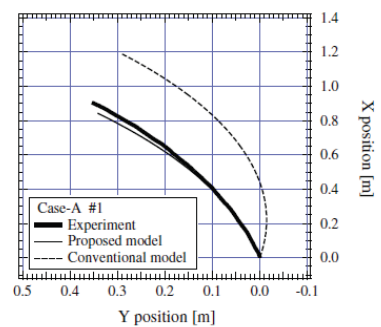
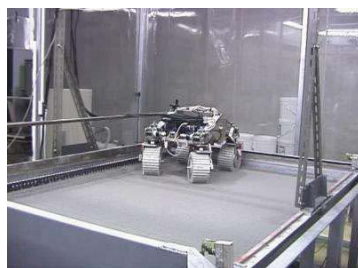
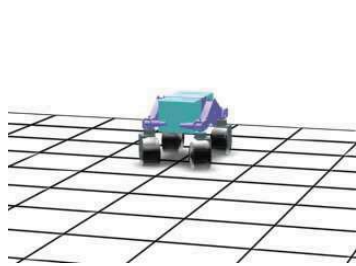
- Compare experimental results with numerical simulation.
- Find best-matched wheel-terrain parameters.

Experimental Validation: Vehicle dynamics



Experimental Validation: Vehicle dynamics

Steering^{[4][7]}



Side-slope^[4]

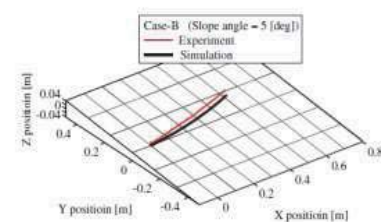
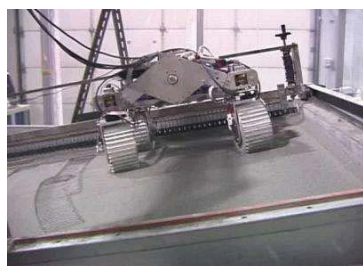
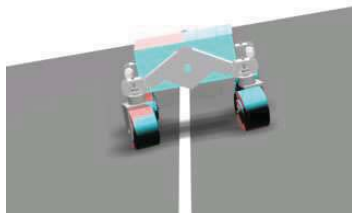


Table of Contents

- Rough terrain mobile robot
 - Introduction to dynamic simulation
- Dynamic simulation
 - Wheel contact mechanics
 - Application to robotic analysis, control, & design
- Open issues
 - Limitation of classical model
 - Key approaches
- Conclusion
 - Dilemma: “Accuracy or Cost”
 - Towards next generation of dynamic simulation

Dynamic simulation for rough terrain robot: Applications

Mobility Analysis

Slope traversability

Stochastic approach

GN&C

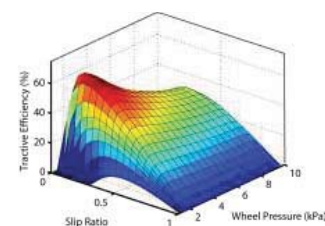
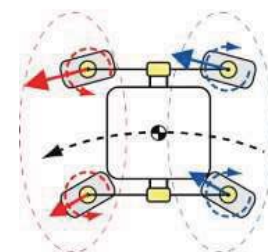
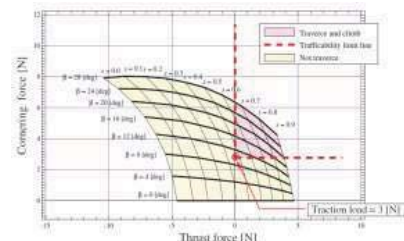
Traction/Slip Control

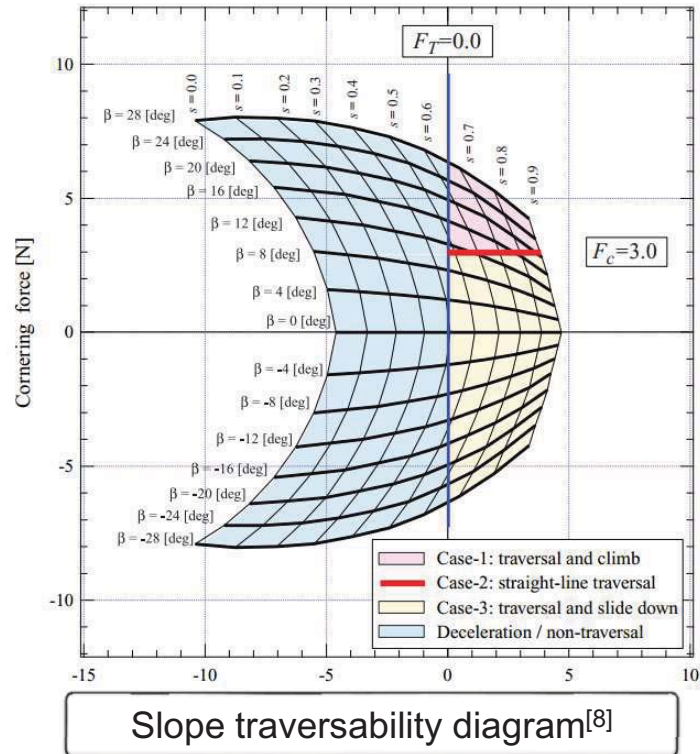
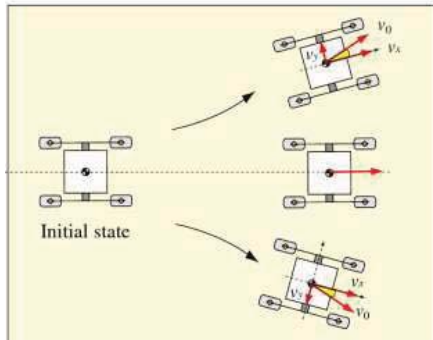
Path Planning

Design

Wheel/Vehicle dimension

Actuators / Suspensions





Applications: Mobility Analysis: Stochastic Approach

- Dynamic simulation needs **10 or more** input parameters that are (experimentally) predetermined: *Deterministic approach*
→ Monte Carlo: computationally expensive.

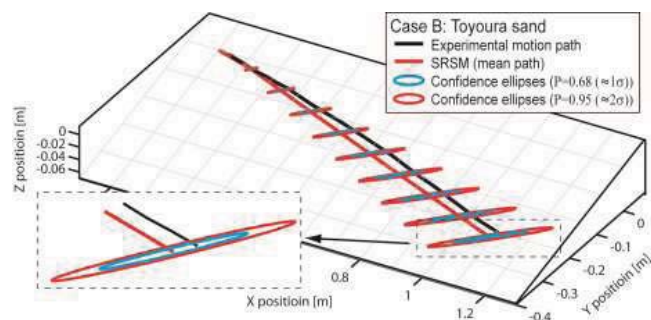
Stochastic Approach^[9]

MBD/Terramechanics



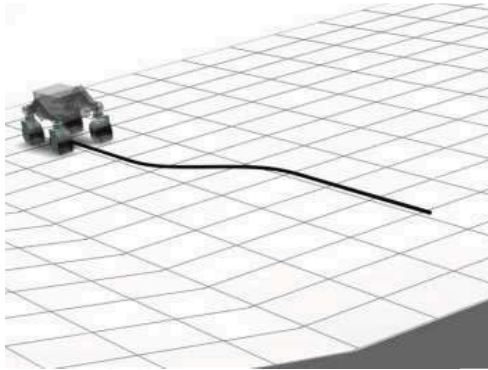
SRSM

- Modeling of uncertainty using a functional approximation
- An equivalent reduced model is used to derive the system output.

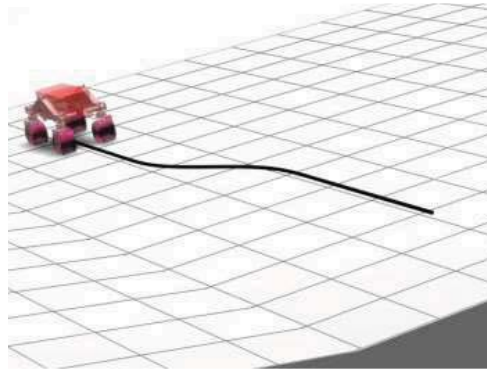


Method	Case A: Lunar simulant	Case B: Toyouira sand
SMC (500 runs)	17526.1 sec	79994.1 sec (one day)
LHSMC (100 runs)	3507.2 sec	16232.5 sec
Proposed approach (2nd order, 500 runs)	245.8 sec	1125.9 sec

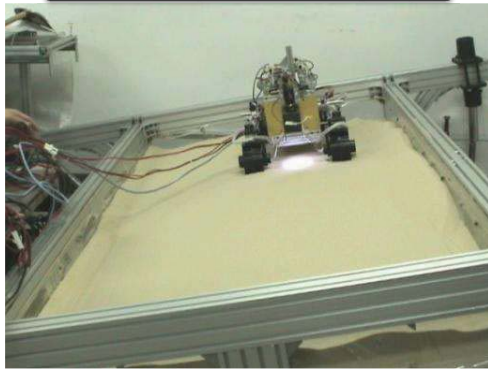
Applications: Slip compensation control^{[8][10]}



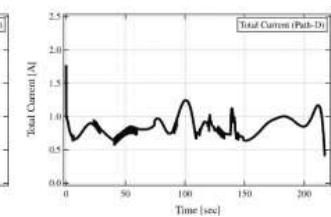
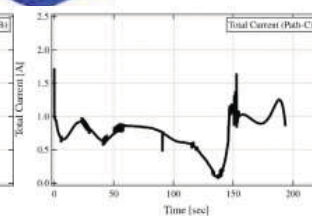
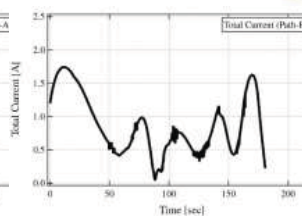
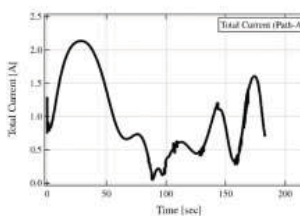
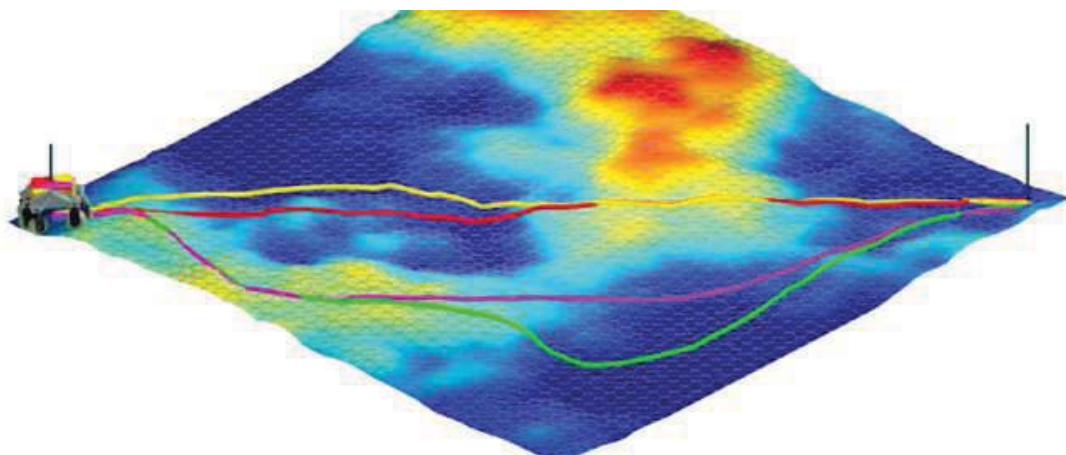
w/o compensation



w/ compensation



Applications: Path planning and evaluation^[11]

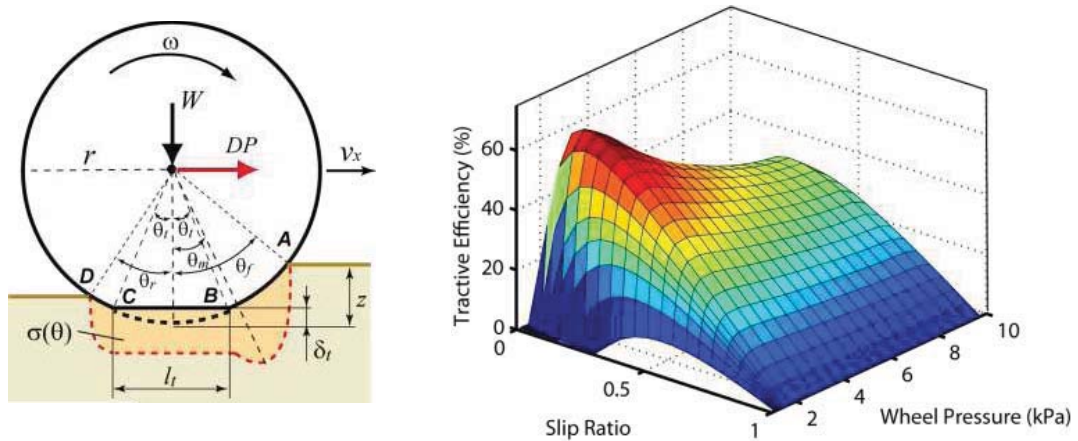


- The simulation provides the dynamic behavior of the robot, and also estimates the energy required for travel.

Applications: Wheel Design

■ Stiffness of flexible wheel^[12]

- Derives the most appropriate wheel **stiffness** maximizing the tractive efficiency.



Applications: Wheel Design

■ Rigid wheel design optimization^[13]

- Find the wheel dimension (width and radius) which maximizes the tractive efficiency.

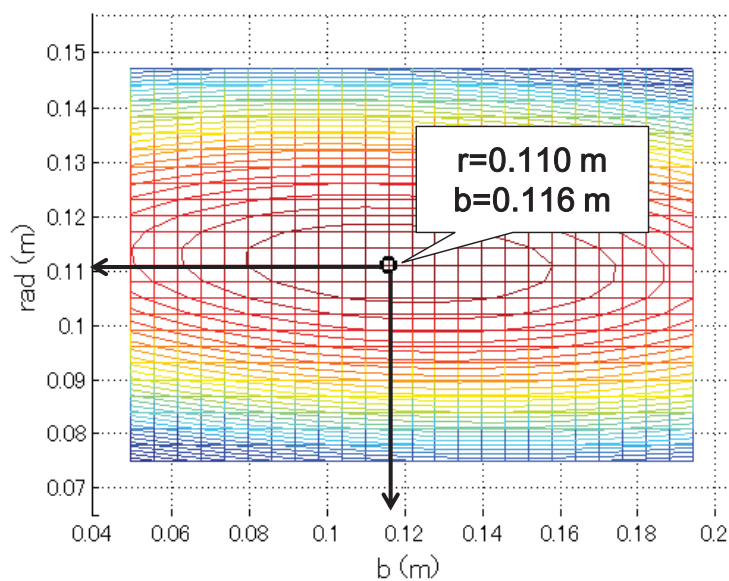


Table of Contents

- Rough terrain mobile robot
 - Introduction to dynamic simulation
- Dynamic simulation
 - Wheel contact mechanics
 - Application to robotic analysis, control, & design
- Open issues
 - Limitation of classical model
 - Key approaches
- Conclusion
 - Dilemma: “*Accuracy or Cost*”
 - Towards next generation of dynamic simulation

(Well-known) open issues

- Classical terramechanics approach includes the following open issues and limitations:

	Issues/Limitations	Negative impacts
Scale dependency	Devoted to the application to large-heavy vehicles	Small-light weight vehicle will not be accurately simulated.
Quasi-static	Not rate dependent No velocity-dependent terms/parameters	Slip-sinkage effect cannot be well determined.
Terrain geometry	No model for lug/grouser No update in the terrain surface deformation	Soil surcharge, multi-pass, or wheel lug effect are not explicitly addressed.

(Well-known) open issues

- Classical terramechanics approach includes the following open issues and limitations:

Issues/Limitations	Negative impacts
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Scale dependency</div> Devoted to the application to large-heavy vehicles	Small-light weight vehicle will not be accurately simulated.
<p><i>“Predictions for wheels smaller than 20 inches in diameter become less accurate as wheel diameter decreases, because the sharp curvature of the loading area was neither considered in its entirety nor is it reflected in bevameter tests”^[14]</i></p> <p style="text-align: right;">- M.G. Bekker, “Introduction to Terrain-Vehicle Systems,” 1969 -</p>	

Open issues and key approaches

Issues	Approach	Literatures**
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Scale dependency</div>	<ul style="list-style-type: none"> ● Scale-dependent parameters □ Revisit stress distribution/soil flow 	Meirion-Griffith et al. ^[15] , Senatore et al. ^[16]
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Quasi-static</div>	<ul style="list-style-type: none"> □ Transient state of stress-soil flow ● Velocity-dependent function 	Nagatani et al. ^[17] , Senatore et al. ^{[18][19]} , Moreland et al. ^[20] , Shirai et al. ^[21]
<div style="border: 1px solid black; padding: 5px; display: inline-block;">Terrain geometry</div>	<ul style="list-style-type: none"> ● Lug effect analysis ◆ Terrain surface grid, height-field, wheel polygons 	Ding et al. ^[22] Ding et al. ^[23] , Sutoh et al. ^[24] Holz et al. ^[25] Krenn et al. ^{[26]-[28]} Azimi et al. ^{[29]-[31]}

**Please note that there are more papers and researches devoted to the above issues.

Legends:

● : Modify the classical model

□ : Exploit experimental observations

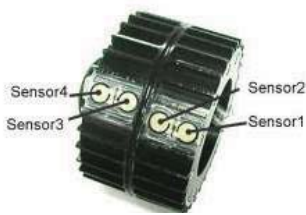
◆ : Employ numerical method

In-wheel sensor system
Particle Image Velocimetry (PIV)

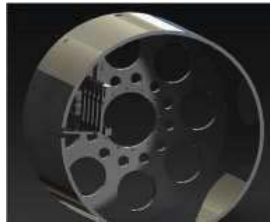
DEM/FEM approaches

Key approaches: Measurement of Stress-soil flow

“In-wheel sensor” : stress distribution



(Nagatani et al. 2009)
Built-in pressure sensor

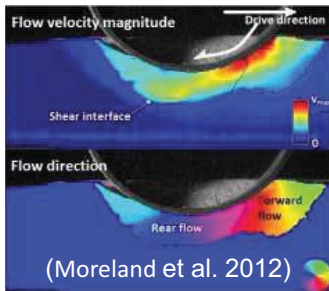


(Senatore et al. 2012)
Cantilever+strain gage

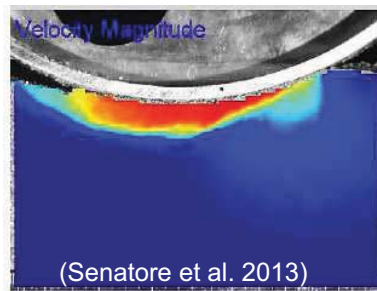


(Shirai et al. 2014)
Pressure + light sensors

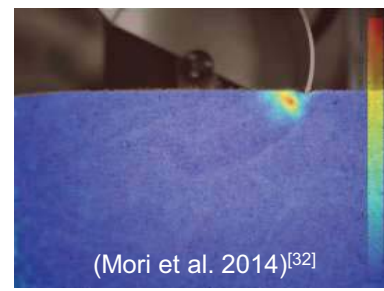
PIV : soil flow, failure zone, soil strain



(Moreland et al. 2012)



(Senatore et al. 2013)

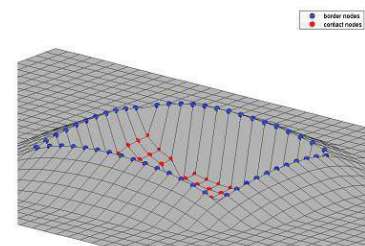
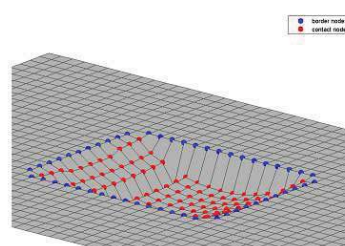
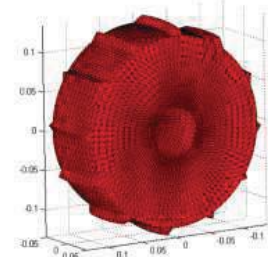
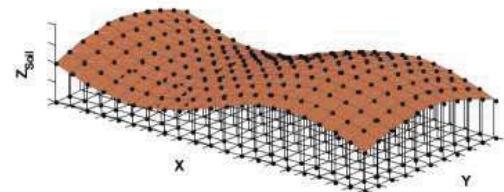


(Mori et al. 2014)^[32]

Key approaches: DEM/FEM approaches

SCM (Krenn et al.)

- Soil surface evaluation grid
- Wheel polygonal surface
- Footprint calculation
- Soil displacement/deposition/erosion



Key approaches: DEM/FEM approaches

Vortex (Azimi et al., Holz et al.)

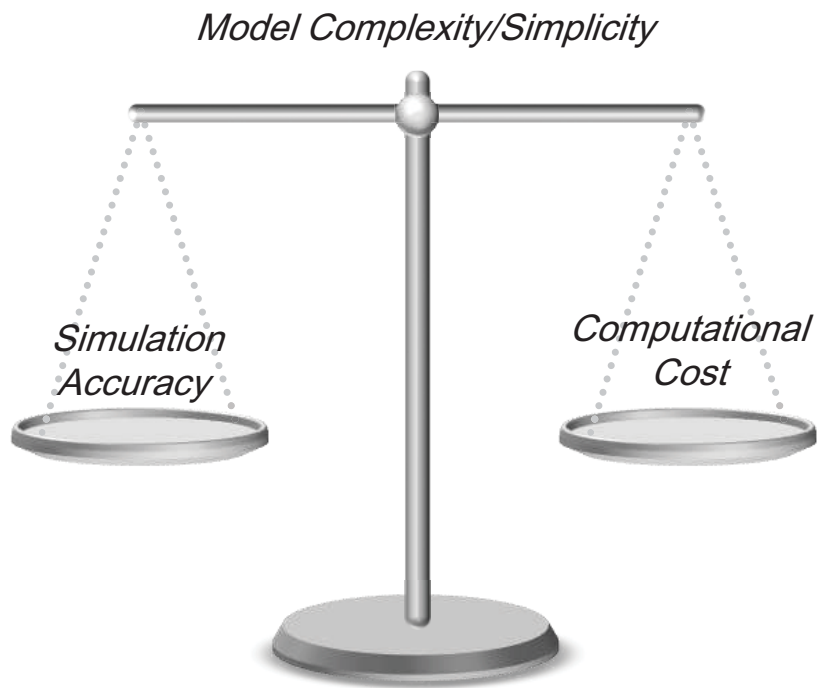
- **Multi-scale analytical approach** (Holz et al., 2009)
 - Hybrid model of soil deformation
 - ✓ particle-based: soil at dynamic state (micro)
 - ✓ mesh-based: soil at static state (macro)
- **Infinitesimal plasticity theory** (Azimi et al., 2011)
 - Velocity field in the vicinity of the contact patch
 - Stress field based on Drucker-Prager constitutive relation
 - ➔ **Time- and velocity-dependent interaction.**



Table of Contents

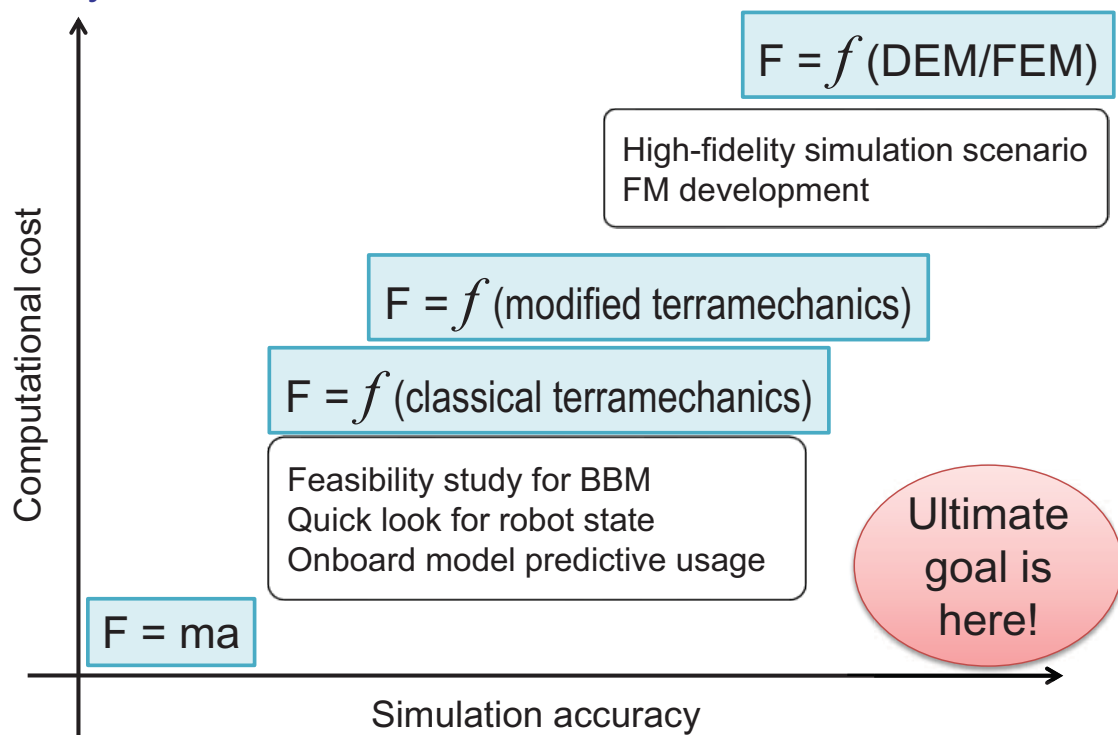
- **Rough terrain mobile robot**
 - Introduction to dynamic simulation
- **Dynamic simulation**
 - Wheel contact mechanics
 - Application to robotic analysis, control, & design
- **Open issues**
 - Limitation of classical model
 - Key approaches
- **Conclusion**
 - Dilemma: “*Accuracy or Cost*”
 - Towards next generation of dynamic simulation

Conclusion: Dilemma "Accuracy or Cost" ?



Conclusion: Dilemma "Accuracy or Simplicity" ?

- Dynamic simulation in the user-defined scenario



Conclusion: Towards next generation of dynamic simulation

Future possible direction of our research:

- 1. Try to decrease the number of model parameters
 - Rethink their physical meanings: “is it magic parameter?”
 - Exclude/combine low-sensitive ones
 - Simulation accuracy will be independent from their uncertainty.

- 2. Constitutive laws w/ well-defined experimental observations
 - Incorporate in-wheel sensor and PIV results
 - Partially utilized for inverse problems
 - Robot-terrain interaction mechanics applicable for small-light weight vehicles

- 3. Benchmark data for the model validation
 - Provide experimental data w/ robot motion, soil/terrain info, and others.

Acknowledgements

- Prof. K. Yoshida (Tohoku Univ., JAPAN)
- Prof. K. Nagatani (Tohoku Univ., JAPAN)
- Dr. K. Iagnemma (MIT, USA)
- Prof. T. Kubota (JAXA, JAPAN)
- Dr. M. Otsuki (JAXA, JAPAN)
- Dr. M. Sutoh (JAXA, JAPAN)
- Prof. Taizo Kobayashi (Fukui Univ., JAPAN)
- Prof. S. Ozaki (Yokohama National Univ., JAPAN)
- Mr. Daiki Mori (Keio. Univ, JAPAN)
- Mr. Takayuki Shirai (Keio. Univ, JAPAN)

References

- [1] Burkharter, B., and Sharpe, M., "Lunar Roving Vehicle: Historical Origins, Development, and Deployment," *J. of the British Interplanetary Society*, Vol. 48, 1995, pp. 199-212.
- [2] Bekker, M. G., *Theory of Land Locomotion: The mechanics of vehicle mobility*, Ann Arbor, MI, USA, The University of Michigan Press, 1956.
- [3] Wong, J.Y., *Theory of Ground Vehicles*, 4th edn., Wiley, New York, 2008
- [4] Ishigami, G., *Terramechanics-based Analysis and Control for Lunar/Planetary Exploration Robots*, Doctoral Thesis of Tohoku University, 2008.
- [5] Wong, J. Y., and Reece, A., "Prediction of Rigid Wheel Performance based on the Analysis of Soil-Wheel Stresses, Part I, Performance of Driven Rigid Wheels", *J. of Terramechanics*, vol.4, No. 1, 1967, pp.81-98.
- [6] Janosi, Z., and Hanamoto, B., "Analytical determination of drawbar pull as a function of slip for tracked vehicle in deformable soils," *Proc. of the 1st ISTVS Conference*, Torino, Turin, Italy, 1961, pp. 707-726.
- [7] Ishigami, G., Miwa, A., Nagatani, K., and Yoshida, K., "Terramechanics-Based Model for Steering Maneuver of Planetary Exploration Rovers on Loose Soil," *J. of Field Robotics*, Vol. 24, Issue 3, 2007, pp. 233-250.
- [8] Ishigami, G., Nagatani, K., and Yoshida, K., "Slope Traversal Controls for Planetary Exploration Rover on Sandy Terrain," *J. of Field Robotics*, Vol. 26, Issue 3, 2009, pp. 264-286.
- [9] Ishigami, G., Kewlani, G., and Iagnemma, K., "Predictable mobility - a statistical approach for planetary surface exploration rovers in uncertain terrain," *IEEE Robotics & Automation Magazine*, Vol. 16, No. 4, 2009, pp. 61-70.
- [10] Ishigami, G., Nagatani, K., and Yoshida, K., "Path Following Control with Slip Compensation on Loose Soil for Exploration Rover," *Proc. of the 2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 5552-5557.
- [11] Ishigami, G., Nagatani, K., and Yoshida, K., "Path Planning and Evaluation for Planetary Rovers Based on Dynamic Mobility Index," *Proc. of the IEEE Int. Conference on Robots and Systems*, San Francisco, CA, 2011, 2011, pp. 601-606.
- [12] Ishigami, G., Otsuki, M., and Kubota, T., "Terramechanics-based Model for Flexible/Rigid Wheels in Rough Terrain," *Proc. of Motion and Vibration Control*, 2011 (in Japanese).
- [13] Ishigami, G., "Wheel Parameter Evaluation using Uncertainty Analysis for High Traversability on Sandy Terrain," *Proc. of Motion and Vibration Control*, 2013 (in Japanese).
- [14] Bekker, M. G., "Introduction to Terrain-Vehicle Systems," Ann Arbor, MI, University of Michigan Press, 1969
- [15] Meirion-Griffith, G., and Spenko, M., "A Modified Pressure-sinkage Model for Small, Rigid Wheels on Deformable Terrains," *J. of Terramechanics*, Vol. 48, Issue 2, 2011, pp. 149-155.
- [16] Senatore, C., and Iagnemma, K., "Analysis of Stress Distributions Under Lightweight Wheeled Vehicles," *J. of Terramechanics*, Volume 51, 2014, pp. 1-17.
- [17] Nagatani, K., Ikeda, A., Sato, K., and Yoshida, K., "Accurate Estimation of Drawbar Pull of Wheeled Mobile Robots Traversing Sandy Terrain Using Built-in Force Sensor Array Wheel," *Proc. of the 2009 IEEE/RSJ Int. Conf. on Robots and Systems*, St. Louis, MO, 2009, pp. 2373-2378.

References

- [18] Senatore, C., Wulfmeier, M., Jayakumar, P., MacLennan, J., and Iagnemma, K., "Investigation of Stress and Failure in Granular Soils For Lightweight Robotic Vehicle Applications," *Proc. of the Ground Vehicle Systems Engineering and Technology Symposium*, Michigan, 2012.
- [19] Senatore, C., Wulfmeier, M., Vlahinic, I., Andrade, J., and Iagnemma, K., "Design and Implementation of a Particle Image Velocimetry Method for Analysis of Running Gear-Soil Interaction," *J. of Terramechanics*, Volume 50, Issues 5-6, 2013, pp. 311-326.
- [20] Moreland, S., Skonieczny, K., and Wettergreen, D., "Motion Analysis System for Robot Traction Device Evaluation and Design," *Proc. of the Int. Conf. on Field and Service Robotics*, Matsushima, JAPAN, 2012.
- [21] Shirai, T., and Ishigami, G., "Accurate Estimation of Wheel Pressure-Sinkage Traits on Sandy Terrain using In-wheel Sensor System," *Proc. of 12th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space*, Quebec CANADA, to be presented on June 2014.
- [22] Ding, L., Deng, Z., Gao, H., Tao, J., Iagnemma, K., and Liu, G., "Interaction Mechanics Model for Rigid Driving Wheels of Planetary Rovers Moving on Sandy Terrain Considering Multiple Effects," *J. of Field Robotics*, in press.
- [23] Ding, L., Gao, H., Deng, Z., Yoshida, K., Nagatani, K., "Slip Ratio for Lugged Wheel of Planetary Rover in Deformable Soil: Definition and Estimation," *Proc. of the 2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, St. Louis, MO, 2009, pp.3343-3348.
- [24] Sutoh, M., Yusa, J., Ito, T., Nagatani, K., Yoshida, K., "Traveling Performance Evaluation of Planetary Rovers on Loose Soil," *J. of Field Robotics* Volume 29, Issue 4, 2012, pp.648-662.
- [25] Holz, D., Beer, T., and Kuhlen, T., "Soil deformation models for real-time simulation: a hybrid approach", *Proc. of the 6th Workshop on Virtual Reality Interactions and Physical Simulation*, 2009, pp. 21-30.
- [26] Krenn, R., Gibbesch, A., and Hirzinger, G., "Contact Dynamics Simulation of Rover Locomotion," *Proc. of the 9th Int. Symposium on Artificial Intelligence, Robotics and Automation in Space*, 2007.
- [27] Krenn, R., and Hirzinger, G., "Simulation of Rover Locomotion on Sandy Terrain-modeling, Verification and Validation," *Proc. of the 10th Workshop on Advanced Space Technologies for Robotics and Automation*, ESTEC, 2008.
- [28] Krenn, R., and Gibbesch, A., "Soft Soil Contact Modeling Technique for Multi-Body System Simulation," *Trends in Computational Contact Mechanics*, Vol. 58, 2011, pp.135-155.
- [29] Azimi, A., Holz, D., Kovacs, J., Angeles, J., and Teichmann, M., "Efficient Dynamics Modeling for Rover Simulation on Soft Terrain," *Proc. of the 50th AIAA Aerospace Science Meeting*, 2012.
- [30] Azimi, A., Hirschhorn, M., Ghotbi, B., Kovacs, J., Angeles, J., Radziszewski, P., Teichmann, M., Courchesne, M., and Gonthier, Y., "Terrain modelling in Simulation-based Performance Evaluation of Rovers," *Canadian Aeronautics and Space J.*, Vol. 57, No. 1, 2011, pp.24-33.
- [31] Azimi, A., Kovacs, J., and Angeles, J., "Wheel-Soil Interaction model for Rover Simulation Based on Plasticity Theory," *Proc. of Int. Conf. on Intelligent Robots and Systems*, 2011, pp.280-285.
- [32] Mori, D., and Ishigami, G., "Soil Interaction Model of Hemispherical Sampling Device based on PIV Analysis," *Proc. of 12th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space*, Quebec CANADA, to be presented on June 2014.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Session I

Enhanced mobility & Modeling

- **Title: Analyzing the Impact of Learning Inputs on Near-to-Far Terrain Traversability Estimation**

Authors: K. Ho, T. Peynot and S. Sukkarieh

- **Title: Terrain mapping with a pan and tilt stereo camera for locomotion on a quadruped robot**

Author: S. Bazeille, M. Camurri, J. Ortiz, I. Havoutis, D. G. Caldwell, and C. Semini

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



Analyzing the Impact of Learning Inputs on Near-to-Far Terrain Traversability Estimation

Ken Ho, Thierry Peynot and Salah Sukkarieh

Abstract—With the increasing need to adapt to new environments, data-driven approaches have been developed to estimate terrain traversability by learning the rover’s response on the terrain based on experience. Multiple learning inputs are often used to adequately describe the various aspects of terrain traversability. In a complex learning framework, it can be difficult to identify the relevance of each learning input to the resulting estimate. This paper addresses the suitability of each learning input by systematically analyzing the impact of each input on the estimate. Sensitivity Analysis (SA) methods provide a means to measure the contribution of each learning input to the estimate variability. Using a variance-based SA method, we characterize how the prediction changes as one or more of the input changes, and also quantify the prediction uncertainty as attributed from each of the inputs in the framework of dependent inputs. We propose an approach built on Analysis of Variance (ANOVA) decomposition to examine the prediction made in a *near-to-far* learning framework based on multi-task GP regression. We demonstrate the approach by analyzing the impact of driving speed and terrain geometry on the prediction of the rover’s attitude and chassis configuration in a Mars-analogue terrain using our prototype rover *Mawson*.

I. INTRODUCTION

For autonomously planetary rovers to explore in challenging environments, estimating terrain traversability is necessary to anticipate situations that may compromise its safety and ability to conduct exploration missions, since many scientifically interesting sites on Mars are located in rough and heterogeneous terrain that poses significant risks to the rover [1]. As the Rover-Terrain Interaction (RTI) in such terrain can be very difficult to model correctly, data-driven approaches have been developed to estimate terrain traversability by learning the rover’s response on the terrain based on experience. [2] proposed a framework to estimate the mechanical properties of the terrain using proprioceptive data collected from experiments. A terramechanics model was then used to predict the rover’s wheel slip on the terrain.

Recent literature showed that the rover’s response on the upcoming terrain can be predicted by learning the correlation between exteroceptive and proprioceptive sensor information [3]. This concept is known as *near-to-far* learning. [4] extended the work in [2] by first using proprioceptive training data to learn terrain parameters, and then associated the parameters with terrain classes from a vision-based classifier

The work was partly supported by the Australian Centre for Field Robotics (ACFR) and the NSW government.

K. Ho, T. Peynot and S. Sukkarieh are with the Australian Centre for Field Robotics, University of Sydney, NSW 2006, Australia, (k.ho,tpeynot,salah)@acfr.usyd.edu.au. T. Peynot is also with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia, t.peynot@qut.edu.au.

to anticipate vehicle slip in operation. Similar frameworks to associate mechanical terrain properties with exteroceptive information include [5], which predicted soil softness using the learnt associations between angular acceleration along the pitch and roll axes and color descriptors. By learning the association between terrain appearance and RTI during experiments, these approaches are able to anticipate situations that are hazardous to the rover.

The aforementioned approaches commonly use multiple learning inputs to adequately describe the various aspects that are correlated with terrain traversability. These correlations are learnt using complex learning algorithms often considered as “black box” functions that provide little or no information about the impact of each learning input on the resulting estimate. Without a systematic procedure to determine the relevance of the learning inputs, it is difficult to understand the shortcomings of the system, or to evaluate the suitability of new learning inputs. The overall accuracy of the estimator and the validity of the error can be checked using cross validation [7]. However, in order to better understand the effects of the learning inputs, we need to analyze the contribution or impact of the inputs on the estimate.

Sensitivity analysis methods can be used to better understand the responses of estimation systems [8]. The Sobel index [9], based on variance decomposition, measures sensitivity by expanding the global variance into partial variances. To validate the response of Gaussian Process (GP) regression, frameworks based on sensitivity analysis methods were proposed in [10], [11]. These frameworks analyzed the effects of input variables on the estimate. However, both of the above methods rely on the assumption that the input variables are independent. If the input variables are dependent, the amount of response variance may be influenced by its dependence on other inputs, and thus lead to incorrect interpretations [12].

To account for the contribution from dependent inputs, [13] proposed to decompose the partial variance of an input into a correlated and an uncorrelated contribution components, assuming a linear effect from each component on the response. This approach was later extended by approximating the effect using a sum of functional components of low dimensions, and then computing the decomposition of response variance as a sum of partial variances [14]. [15] proposed sensitivity analysis methods to account for non-constant (heteroscedastic) variances in the estimate.

In this paper we analyze the impact of learning inputs on a near-to-far terrain traversability estimate using a sensitivity analysis method built on Analysis of Variance (ANOVA)

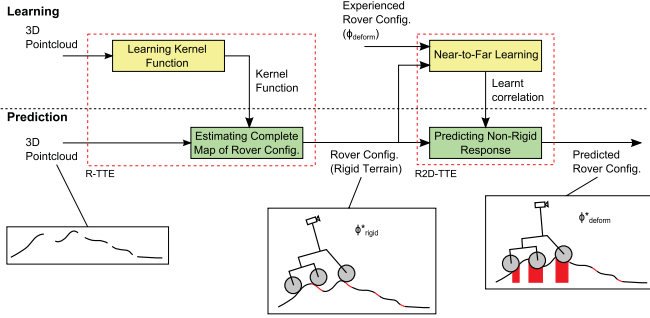


Fig. 1. System architecture of our approach for terrain traversability estimation. Given an incomplete point cloud, R-TTE makes a continuous initial estimate of the rover configuration over the entire map, assuming the terrain is rigid. R2D-TTE then refines this estimate to account for possible terrain deformation.

decomposition in a framework of dependent inputs. This quantifies the contributions from each learning input to the variability of the resulting estimate, including the extent at which the estimate uncertainty can be attributed to the learning inputs. The method first decomposes the estimate into a multi-dimensional representation of primary and interaction effects between the learning inputs. The analytical sensitivity measure is then calculated for combinations of learning inputs, and indicates the significance of each learning input. We validate the approach with experimental data collected using a prototype rover on a Mars-analogue terrain.

The paper is organized as follows. Sec. II outlines our near-to-far learning framework previously proposed in [16] to compute an estimate of the rover attitude and configuration that accounts for the effects of terrain deformation. Sec. III details the theory and implementation of sensitivity analysis method to decompose the resulting estimate. In Sec. IV and V we describe the experimental validation of the approach and discuss the results obtained, including the impact of driving speeds and terrain geometry on the resulting estimate. Finally, Sec. VI proposes a conclusion and possible future work.

II. ESTIMATING TRAVERSABILITY IN PARTIALLY OCCLUDED AND DEFORMABLE TERRAIN

We proposed a framework in [17] and [16] to address the problems of incomplete terrain data and terrain deformation sequentially in separate components. The proposed system architecture can be seen in Fig. 1. Given incomplete terrain data, the first component, named *Rigid-Terrain Traversability Estimation* (R-TTE), provides an initial estimate of the rover configuration Φ_{rigid}^* before any terrain deformation may occur. This is equivalent to assuming that the terrain is rigid. The second component, *Rigid-to-Deformable Traversability Estimate* (R2D-TTE), then refines this prediction by accounting for the effects of terrain deformation on rover configuration, learnt from experience. We name this final estimate Φ_{deform}^* . To account for uncertainties in the observations and knowledge base, both processes are stochastic.

A. R-TTE

The R-TTE module within the framework addresses the problem of incomplete terrain data. Using the method we proposed [17], we estimate a complete map of Φ_{rigid}^* by performing GP regression over an incomplete map of rover configuration. This approach exploits the explicit correlation in rover configuration during operation by learning a kernel function from experience. We set up the traversability estimation scenario as a GP regression problem to predict $\Phi_{rigid}^*(x, y, \psi)$ at each position (x, y) on a Digital Elevation Map (DEM) over different heading angles ψ . The GP posterior (estimate) \bar{f}^* and covariance $cov(f^*)$ can be given as:

$$\bar{f}^* = K(X^*, X)[K(X, X) + \sigma_n^2 I]^{-1}z \quad (1)$$

$$cov(f^*)$$

$$= K(X^*, X^*) - K(X^*, X)[K(X, X) + \sigma_n^2 I]^{-1}K(X, X^*) \quad (2)$$

where K represents the covariance matrix evaluated using the learnt kernel function at all the pairs of training points X and query points X^* , σ_n is the noise variance, and z is the training target.

B. R2D-TTE

The R2D-TTE module, previously proposed by the authors in [16], refines the estimate provided by R-TTE by accounting for the effects of terrain deformation. We extended the estimation process to exploit the local variations in Φ_{rigid} that correlate with the actual rover configuration resulting from terrain deformation, i.e. Φ_{deform} , and include driving speed as an additional learning input. This idea is implemented in a near-to-far learning approach by learning the correlation between the initial prediction, Φ_{rigid}^* , its local variations, and experience in Φ_{deform} collected during training (Fig. 1). Fig. 2 illustrates the outline of the R2D-TTE approach. During learning, the rover observes a patch of terrain and predicts Φ_{rigid}^* . When the rover traverses over the patch of terrain, it learns the correlation between Φ_{rigid}^* and the experienced rover configuration Φ_{deform} with terrain deformation. Once the training is complete, in operation, the rover uses the learnt correlations to predict Φ_{deform}^* from new exteroceptive data.

Learning is performed in a multi-task heteroscedastic GP framework that considers the interaction between multiple training inputs and targets. We use multiple-input GP regression by Automatic Relevance Determination (ARD) to learn the correlation between the training inputs X and each component in the target z . Using a separate lengthscale for each component in the training input, ARD determines the orders of interaction that are important in the GP regression [18]. We use convolution processes to account for the correlations between estimation outputs [19]. This approach uses a convolution between a smoothing kernel k_q and latent functions $u(z)$ to express each output f_q :

$$f_q(X) = \int_{-\infty}^{\infty} k_q(X - z)u(z)dz \quad (3)$$

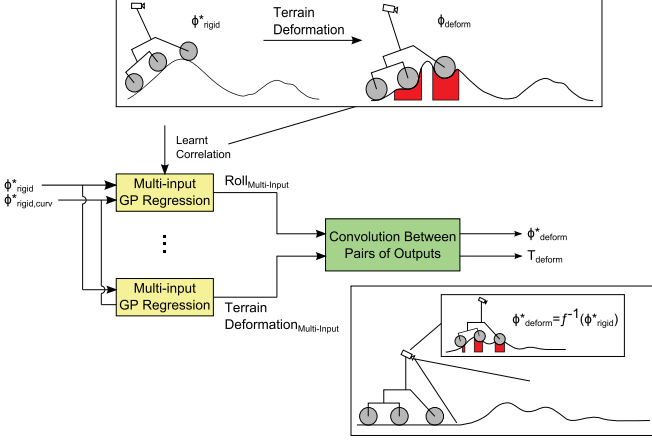


Fig. 2. Illustration R2D-TTE process to account for the effects of terrain deformation on the rover configuration, using correlations learnt in experiments.

We then use multiplication of Gaussian distributions to determine the correlation between pairs of outputs as well as between any given output and the latent function:

$$\begin{aligned} cov[f_q(X), f_s(X')] &= \sum_{r=1}^R \int_{-\infty}^{\infty} k_{qr}(X-z) \int_{-\infty}^{\infty} k_{sr}(X'-z') k_{u_r, u_r}(z, z') dz' dz, \\ cov[f_q(X), u_r(z)] &= \int_{-\infty}^{\infty} k_{qr}(X-z') k_{u_r, u_r}(z', z) dz'. \end{aligned} \quad (4)$$

Using the covariance matrices in Eqs. (4), we perform joint-prediction of the estimation outputs by iteratively calculating the matrices for each latent function and output.

III. SENSITIVITY ANALYSIS

We use Analysis of Variance (ANOVA) decomposition to analyze the effects of learning inputs on the resulting estimate. This method decomposes the total mean and variance of the GP estimator into contributions from dependent inputs. The percentage of total contribution attributed among the inputs then provides a measure of importance of the interaction effect between each learning input and the resulting estimate [15].

To decompose the resulting estimate, we need to first find the marginal effect $\bar{y}_e(x_e)$. This is the overall effect of all variables x_e on the estimate, and is defined by integrating out all other variables [10]:

$$\bar{y}_e(x_e) = \int_{\otimes_{j \neq e} \mathcal{X}_j} y(x_e, x_{-e}) \prod_{j \neq e} w_j(x_j) dx_j \quad (5)$$

for $x_e \in \otimes_{j \in e} \mathcal{X}_j$,

where $w_j(x_j)$ is a weight function that represents interest among x_j , and \mathcal{X}_j denotes the values of interest for variable x_j .

We then use Eq. (5) to decompose the resulting estimate $y(x)$ into corrected effects involving the contributions from any number of variables $x \in \mathcal{X}$:

$$y(x) = \mu_0 + \sum_{j=1}^d \mu_j(x_j) + \sum_{j=1}^{d-1} \sum_{j'=j+1}^d \mu_{jj'}(x_j, x_{j'}) + \dots + \mu_{1\dots d}(x_1, \dots, x_d), \quad (6)$$

where μ_0 , $\mu_j(x_j)$, $\mu_{jj'}(x_j, x_{j'})$ are the overall average, corrected primary effect, and corrected interaction effect respectively:

$$\begin{aligned} \mu_0 &= \int_{\mathcal{X}} y(x) w(x) dx \\ \mu_j(x_j) &= \bar{y}_j(x_j) - \mu_0 \quad \text{for } x_j \in \mathcal{X}_j \\ \mu_{jj'}(x_j, x_{j'}) &= \bar{y}_{jj'}(x_j, x_{j'}) - \mu_j(x_j) - \mu_{j'}(x_{j'}) - \mu_0 \\ &\quad \text{for } x_j, x_{j'} \in \mathcal{X}_j \otimes \mathcal{X}_{j'} \end{aligned} \quad (7)$$

where x_j is a complementary set of $x_{j'}$.

For example, to examine the contributions from learning inputs x_1 and x_2 on the resulting estimate, we can consider the overall joint effect:

$$\bar{y}_{12}(x_1, x_2) = \mu_0 + \mu_1(x_1) + \mu_2(x_2) + \mu_{12}(x_1, x_2) \quad (8)$$

for $x_1, x_2 \in \mathcal{X}_1 \otimes \mathcal{X}_2$

In practice, we first estimate the marginal effects using a best linear unbiased predictor (BLUP) [10], and then compute the corresponding estimated corrected effect by subtracting all estimated lower-order corrected effects. Using this decomposition, we can determine the impact of the learning inputs on the resulting estimate as a function of its interaction with other learning inputs.

The variance of the estimate can also be decomposed as [14]:

$$V(y(x)) = \sum_{u \in S} [V(f_j(x_j)) + Cov(f_j(x_j), f_{j'}(x_{j'}))] \quad (9)$$

where $f_{j'}(x_{j'}) = f(x) - f_j(x_j)$.

To quantify the contribution of the overall learning input x_j to the resulting estimate, we calculate the analytical sensitivity measure S_j that accounts for both the estimate mean and uncertainty, which can be computed as [20]:

$$S_j = \frac{V[\mu_j(x_j)] + Cov[\mu_j(x_j), \mu_{j'}(x_{j'})]}{V[y(x)]} \quad (10)$$

In our implementation, we compute S_j for each learning input to determine their impact on the resulting GP regression estimate.

IV. EXPERIMENTAL SETUP

A. Platform - Mawson Rover

The experiments were conducted using *Mawson*, a 6-wheeled prototype rover platform with a rocker-bogie chassis and individual steering motors on each wheel (see Fig. 3(a)). *Mawson* is approximately 80cm long, 63cm wide,

and 90cm tall. The radius of each wheel is 5cm . Onboard sensors include:

- two color cameras and a RGB-D camera (Microsoft KinectTM) mounted on a pan-tilt unit, tilted down $\approx 20^\circ$, used primarily for terrain modeling,
- two Hall-effect encoders measuring the two rear bogie angles (α_1, α_2 in Fig. 3(b)), and a potentiometer on the rocker differential,
- an IntersenseTM IS-1200 motion capture system that fuses data from a visual camera and an inertial measurement unit to provide the 6-DOF sensor pose.

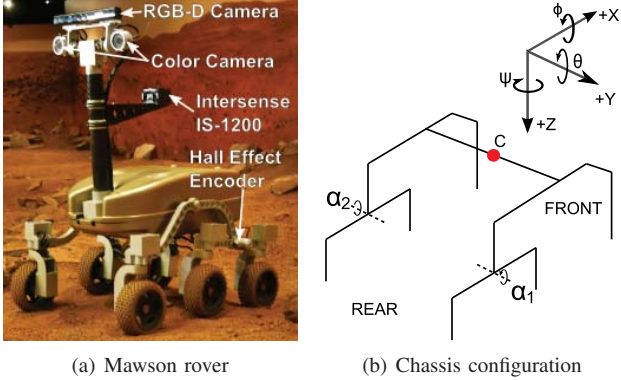


Fig. 3. Experimental rover platform.

In our experiments, the Intersense IS-1200 unit provided localization of the rover with respect to a constellation of fiducials in the environment that were geo-referenced using surveying equipment, with an average accuracy of 2cm in position and 1° in rotation. Since the experiments were performed in an indoor environment, we used the RGB-D camera to obtain 3D point clouds of the terrain. For outdoor operations, where the RGB-D camera may be unable to provide reliable data, dense stereovision can be used instead without affecting the conclusions of this study. In order to associate the acquired point clouds with the localization of the rover, we performed exteroceptive calibration between the two sensors off-line using the method in [21].

B. Test Environment

We conducted our experiments at the Marsyard, a Mars-analogue terrain in Sydney, Australia (see Fig. 4). The Marsyard is approximately $15\text{m} \times 8\text{m}$ and contains slopes, soil and rocks similar to Martian terrain. The typical obstacle size in the Marsyard is approximately 0.05m to 0.2m in radius. Combined with the mixed sizes in gravel granules, this presents a considerable challenge in traversability for Mawson since its wheel radius is 0.05m .

C. Experimental Data For Learning

We performed a range of traversals over different terrain to engage Mawson in a variety of situations that it is likely to encounter during operation. Before the rover traversed on the terrain, we recorded the point cloud of the terrain using an external depth sensor (Asus XtionTM). As the



Fig. 4. Marsyard in Sydney, Australia.

rover traversed the terrain, we collected the experienced rover configuration Φ_{deform} using the Intersense sensor, as well as terrain data using the onboard depth sensor. After terrain traversal, we acquired another pointcloud of the terrain using the external depth sensor. To quantify terrain deformation from terrain traversals, we compared the DEM generated from the point cloud of the terrain before and after rover traversal. In order to obtain terrain geometry data in the same navigation frame used by Mawson for its localization, we first used a theodolite to find the transformation between a reference point on Xtion sensor and the navigation frame of the rover [21]. Nearest-Neighbor Iterative Closest Point was then used to find the transformation between the reference point and the image frame.

D. GP Learning Inputs and Outputs

The training input X of our GP includes $\Phi_{rigid}^*(s)$, as defined in Fig. 3(b), and its local curvatures:

$$X = [\phi, \phi_{curv}, \theta, \theta_{curv}, \alpha_1, \alpha_{1_{curv}}, \alpha_2, \alpha_{2_{curv}}]. \quad (11)$$

The training target z includes the actual rover configuration $\Phi_{deform}(s)$ and terrain deformation \mathcal{T}_{deform} . We define \mathcal{T}_{deform} as the combined planform and platform curvature of the rover configuration on deformed terrain:

$$z = [\Phi_{deform}, \mathcal{T}_{deform}], \quad (12)$$

$$\Phi_{deform} = [\phi_{deform}, \theta_{deform}, \alpha_{1_{deform}}, \alpha_{2_{deform}}] \quad (13)$$

$$\mathcal{T}_{deform} = [\phi_{curv}, \theta_{curv}, \alpha_{1_{curv}}, \alpha_{2_{curv}}]_{deform}. \quad (14)$$

The GP training data was discretized over 8 equally spaced yaw angles to facilitate learning with fewer data points.

V. EXPERIMENTAL VALIDATION

A. Predicting Φ_{deform} using R2D-TTE

The experimental validation was performed using data collected from a 30m drive on the Marsyard. Figure 5 illustrates the rover configuration estimated between 12 and 22m along its traveled distance using R-TTE and R2D-TTE, with the rover operating at different speeds. Using correlations between exteroception, driving speed, and actual rover experience, the estimation made using R2D-TTE is able to anticipate the effects of terrain deformation.

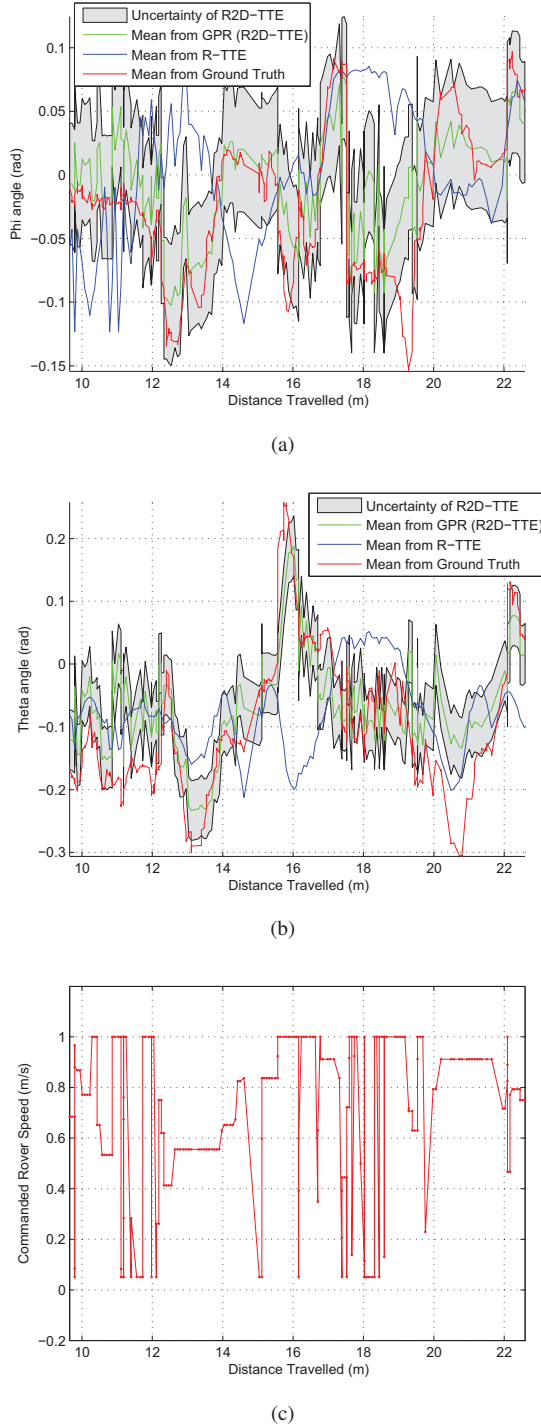


Fig. 5. GP regression results for predicting roll ϕ (a) and pitch θ (b). The commanded speed of the rover is shown in (c). The improvement in accuracy using R2D-TTE (green) over R-TTE (blue) can be seen particularly between 14 and 20m along its traveled distance where there is significant terrain deformation.

B. Sensitivity Analysis of Φ_{deform}^*

1) *Analytical Sensitivity Measure:* Table I shows the analytical sensitivity measure of first-order effects from each learning input to Φ_{deform} . It can be seen that Φ_{rigid} in the learning inputs contributes to the highest values in first order

effects in Φ_{deform} (highlighted in Table I). This is because Φ_{rigid} is expected to be very similar to Φ_{deform} in areas with minimal terrain deformation.

TABLE I
ANALYTICAL SENSITIVITY MEASURE OF FIRST ORDER EFFECTS FROM EACH LEARNING INPUT TO Φ_{deform} .

	ϕ_{deform}	θ_{deform}	$q1_{deform}$	$q2_{deform}$
ϕ_{rigid}	0.12	0.08	0.04	0.04
θ_{rigid}	0.10	0.13	0.06	0.07
$q1_{rigid}$	0.04	0.01	0.07	0.04
$q2_{rigid}$	0.03	0.05	0.03	0.08
$\phi_{rigid,curv}$	0.04	0.06	0.01	0.02
$\theta_{rigid,curv}$	0.02	0.01	0.02	0.01
$q1_{rigid,curv}$	0.01	0.03	0.05	0.02
$q2_{rigid,curv}$	0.03	0.01	0.04	0.04
<i>speed</i>	0.05	0.04	0.02	0.03
$\sum_{primary}$	0.44	0.41	0.34	0.35

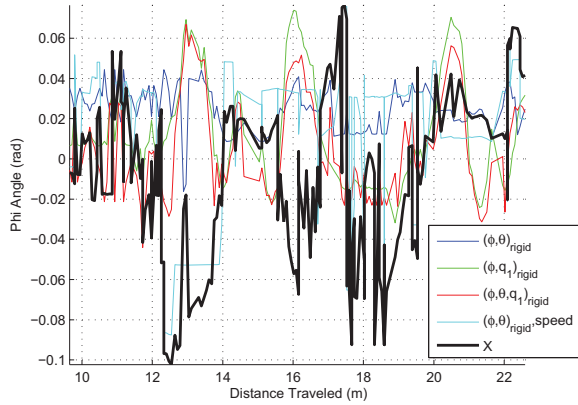
Table II shows the analytical sensitivity measure of selected interaction effects from combinations of learning inputs to Φ_{deform} . The interaction effects with the highest impact on the estimated rover roll and pitch are highlighted, and the sum of the selected interaction effects is shown at the bottom of the table. It can be seen that the interaction effects from combinations of Φ_{rigid} and driving speed are the highest compared to other combinations of learning inputs. This validates the choice of adding driving speed as a learning input. The interaction effects of $\Phi_{rigid,curv}$ with other learning inputs are also significant, having an analytical sensitivity measure between 50 and 65% of the highest values in the estimate of ϕ_{deform} and θ_{deform} .

It should be noted that other combinations of interaction effects also contribute to the resulting estimate Φ_{deform} , but are minor and thus not shown in Table II for clarity.

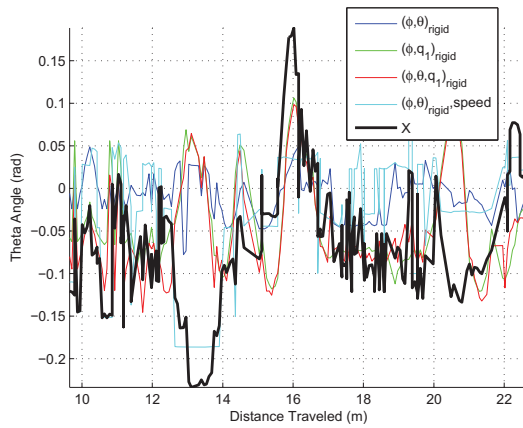
TABLE II
ANALYTICAL SENSITIVITY MEASURE OF SELECTED INTERACTION EFFECTS FROM EACH LEARNING INPUT TO ϕ_{deform} AND θ_{deform} .

	ϕ_{deform}	θ_{deform}
$(\phi, \theta)_{rigid}$	0.05	0.04
$(\phi, q1)_{rigid}$	0.03	0.04
$(\phi, q2)_{rigid}$	0.03	0.02
$(\theta, q1)_{rigid}$	0.02	0.01
$(\theta, q2)_{rigid}$	0.03	0.02
$(\phi, \theta, q1)_{rigid}$	0.06	0.03
$(\phi, \theta, q2)_{rigid}$	0.05	0.03
$(\phi, \theta, \phi_{curv})_{rigid}$	0.04	0.02
$(\theta, q1, \theta_{curv})_{rigid}$	0.03	0.02
$(\phi, \theta)_{rigid, speed}$	0.03	0.04
$\sum_{interaction, x \subset \mathcal{X}}$	0.37	0.27

2) *Decomposing GP Regression Estimate:* Fig. 6 shows a decomposition of the interaction effects with the highest impact on the resulting estimate, previously shown in Table II. It can be seen that the dominant interaction effect changes among the combination of learning inputs along the rover's trajectory. For example, $(\phi, q1)_{rigid}$ and $(\phi, \theta, q1)_{rigid}$ are the dominant interaction effects from 10 to 12m, whereas $((\phi, \theta)_{rigid}, speed)$ is the dominant interaction effect between 12 and 14.5m.



(a)



(b)

Fig. 6. Decomposition of interaction effects on rover roll ϕ (a), and pitch θ (b). The overall effect from all learning inputs X on the estimate of ϕ and θ is shown as the black line. The blue and green lines show the second order interaction effects from the learning inputs $(\phi, \theta)_{rigid}$ and $(\phi, q_1)_{rigid}$ respectively on the estimate. The impact of the third order interaction effects from the learning inputs $(\phi, \theta)_{rigid}^{speed}$ (teal) on the estimate can be seen particularly between 12 and 14m along its traveled distance where there is significant terrain deformation.

VI. CONCLUSION

In this paper, we analyzed the impact of learning inputs on a near-to-far terrain traversability estimation process, proposed by the authors in prior work, using a sensitivity analysis method built on Analysis of Variance (ANOVA) decomposition. The approach first decomposes the resultant estimate into a multi-dimensional representation of primary and interaction effects between the learning inputs, and then calculates the analytical sensitivity measure that indicates the significance of each learning input. We demonstrated the approach to assess the impact of terrain geometry and driving speed on the estimate of the rover's attitude and chassis configuration that accounts for the effects of terrain deformation. We showed that terrain geometry expressed as the rover's attitude and chassis configuration is the most informative among the learning inputs, having the highest analytical sensitivity measure as a primary effect. It was also

significant as an interaction effect on the resulting estimate, when combined with driving speed. In future work, we will analyze the impact of additional modes of exteroceptive sensor data, such as terrain color and texture, using this analytical framework.

REFERENCES

- [1] J. Grant, A. Steele, M. Richardson, S. Bougher, B. Banerdt, L. Borg, J. Gruener, and J. Heldmann, "Mars science goals, objectives, investigations, and priorities: 2006," MEPAG, Tech. Rep., 2006.
- [2] K. Iagnemma, S. Kang, H. Shibly, and S. Dubowsky, "Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 921–927, 2004.
- [3] A. Howard, M. Turmon, L. Matthies, B. Tang, A. Angelova, and E. Mjolsness, "Towards learned traversability for robot navigation: From underfoot to the far field," *Journal of Field Robotics*, vol. 23, no. 11–12, pp. 1005–1017, 2006.
- [4] C. A. Brooks and K. Iagnemma, "Self-supervised terrain classification for planetary surface exploration rovers," *Journal of Field Robotics, Special Issue on Space Robotics, Part I*, vol. 29, no. 3, pp. 445–468, 2012.
- [5] A. Krebs, C. Pradalier, and R. Siegwart, "Adaptive rover behaviour based on online empirical evaluation: Rover-terrain interaction and near-to-far learning," *Journal for Field Robotics*, vol. 27, no. 2, pp. 158–180, 2010.
- [6] T. P. Setterfield and A. Ellery, "Terrain response estimation using an instrumented rocker-bogie mobility system," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 172–188, 2013.
- [7] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [8] A. Saltelli, K. Chan, E. M. Scott, *et al.*, *Sensitivity analysis*. Wiley New York, 2000, vol. 134.
- [9] I. M. Sobol, "On sensitivity estimation for nonlinear mathematical models," *Matematicheskoe Modelirovanie*, vol. 2, no. 1, pp. 112–118, 1990.
- [10] M. Schonlau and W. J. Welch, "Screening the input variables to a computer model via analysis of variance and visualization," in *Screening*, A. Dean and S. Lewis, Eds. Springer New York, 2006, pp. 308–327.
- [11] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu, "A framework for validation of computer models," *Technometrics*, vol. 49, no. 2, 2007.
- [12] T. A. Mara and S. Tarantola, "Variance-based sensitivity indices for models with dependent inputs," *Reliability Engineering & System Safety*, vol. 107, pp. 115–121, 2012.
- [13] C. Xu and G. Z. Gertner, "Uncertainty and sensitivity analysis for models with correlated parameters," *Reliability Engineering & System Safety*, vol. 93, no. 10, pp. 1563–1573, 2008.
- [14] G. Li, H. Rabit, P. E. Yelvington, O. O. Oluwole, F. Bacon, C. E. Kolb, and J. Schoendorf, "Global sensitivity analysis for systems with independent and/or correlated inputs," *The Journal of Physical Chemistry A*, vol. 114, no. 19, pp. 6022–6032, 2010.
- [15] G. M. Dancik and K. S. Dorman, "Statistical analysis for computer models of biological systems using r," 2008, pp. 1966–1967.
- [16] K. Ho, T. Peynot, and S. Sukkariet, "A near-to-far non-parametric learning approach for estimating traversability in deformable terrain," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Intelligent Systems*, 2013.
- [17] —, "Traversability estimation for a planetary rover via experimental kernel learning in a gaussian process framework," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.
- [18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press, 2006.
- [19] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [20] G. Chastaing and L. L. Gratiet, "Anova decomposition of conditional gaussian processes for sensitivity analysis with dependent inputs," *arXiv:1310.3578*, 2013.
- [21] J. P. Underwood, A. Hill, T. Peynot, and S. J. Scheduling, "Error modeling and calibration of exteroceptive sensors for accurate mapping applications," *Journal of Field Robotics*, vol. 27, no. 1, pp. 2–20, 2010.

Terrain mapping with a pan and tilt stereo camera for locomotion on a quadruped robot

Stephane Bazeille¹, Marco Camurri¹, Jesus Ortiz¹, Ioannis Havoutis¹, Darwin G. Caldwell¹ and Claudio Semini¹

Abstract—Legged robots are expected to have superior mobility on rough terrain than wheeled robots. The main reason is that legged locomotion is more adaptable to a wide range of terrain types as the robot can decompose its path into a sequence of footholds and can use different locomotion strategies. In order to accomplish most of the locomotion tasks the robot requires high level control (*i.e.*, to adjust the locomotion parameters and to choose optimal footholds) which depends on real-time localization and accurate terrain mapping. In this paper, we propose a SLAM solution using a pan and tilt stereo camera mounted on an hydraulically actuated quadruped robot that builds a map and keeps track of the robot’s position. Since the computation needs to be carried out on board and the robot is subject to considerable motion during its locomotion (regular vibrations, impacts or slippages), we developed a dedicated implementation based on fast stereo depth computation, GPU based map building and mechanical motion compensation. Combined with a foothold planning framework presented in our previous work [1], this localization and mapping ability allows to perform locomotion in a fully planned manner. Successful results of foothold planning with our quadruped robot show the effectiveness of our method.

Keywords: Stereo Vision, SLAM, Legged Locomotion.

I. INTRODUCTION

Legged robots have the potential to navigate in more challenging terrains than other robots do. Unfortunately their control is more demanding because they have to deal with the traditional mapping and path planning as well as some more particular issues like balancing and foothold planning, that are specific to legged locomotion. The perception system is crucial to enable the robot to navigate, coping with terrain irregularities and avoiding obstacles.

At the *Istituto Italiano di Tecnologia*, we designed the fully torque-controlled Hydraulic Quadruped robot (HyQ) to perform highly dynamic tasks in difficult terrains; we showed crawling, walking, trotting and jumping capabilities. More recently, we demonstrated some visually assisted trotting, Inertial Measurement Unit (IMU) based balancing, step reflex, and foothold planning on known terrain [1], [2].

In this paper, we present the integration of a dedicated stereo vision-based Simultaneous Localization And Mapping (SLAM) system and terrain modeling on HyQ that allows foothold planning. To extend the robustness of these two

modules during navigation, a mechanical motion compensation based on IMU information has also been added. By decoupling camera and body motion, we improved the accuracy of mapping and reduced SLAM failures. We preferred a stereo camera instead of the RGBD camera because it provides a wider field of view, a larger range, and more flexibility for indoor/outdoor applications.

II. RELATED WORK

Significant progress has been achieved during the last few years in the field of robot perception abilities. In the context of quadruped robots, perception is required for different subsequent tasks and methods such as state estimation, robot SLAM, terrain modeling and classification or object recognition. Only few teams demonstrated the implementation of SLAM on a real system. In most cases they only handle parts of the problem, like doing on board terrain mapping but with the support of external localization or using accurate pre-existing maps while localizing the robot on board. Vision is rarely used on quadrupedal machines. Indeed, such platforms are commonly used to develop low-level controllers, rather than high-level cognitive processes. Furthermore legged locomotion expects precise and failsafe perception capabilities, regardless of difficulties like fast motion, impact shocks, complex visibility that make its use more difficult. However, up to now, few people have worked on the integration of vision sensors on quadrupedal platforms.

Kolter *et al.* [3] presented the most autonomous approach by removing the dependence on given maps and external state input. In their control framework they use a stereo camera with a simple ICP-based technique for point cloud registration to incrementally build a map of the environment. Then, they use a texture synthesis algorithm to fill occluded areas in order to obtain a complete map for the subsequent motion planning of their quadruped, *LittleDog* [4], [5]. While the camera was on the robot, the vision processing and path planning was performed on an external computer.

Chilian and Hirschmuller [6] implemented a multi sensor fusion algorithm by merging inertial measurements, legged odometry, and visual odometry for the DLR Crawler. A semi global matching algorithm for stereo vision was implemented in order to compute an elevation map where the traversability of their hexapod robot could be estimated.

In a similar way, [7] fused the information from stereo vision, legged odometry, and IMU in order to obtain accurate state estimation of their quadruped robot BigDog [8]. On

*This work was supported by Istituto Italiano di Tecnologia (IIT)

¹ Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163 Genova, Italy. `firstname.lastname at iit.it`

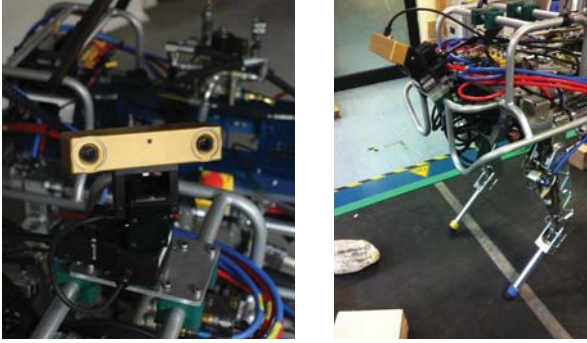


Fig. 1. Pictures of our experimental setup. (a) The HyQ vision setup (b) The stereo camera in the mapping position while the robot is trotting.

the above mentioned system they also developed obstacle detection algorithms using stereo vision and LIDAR, a registration pipeline and 2D cost map computation which was used eventually for A* based path planning.

Bajracharya *et al.* [9] recently showed terrain mapping for vision-in-the-loop walking on the LS3 robot from Boston Dynamics. The vision system was used to map the environment in the vicinity of the robot and inform the gait generation process about possible changes in the surface where the robot is locomoting. Their main contribution focused on the robustness of the mapping in difficult terrain (vegetation, slopes) and difficult light condition (day or night).

Filitchkin and Byl [10] used a monocular camera to perform terrain classification. The classification was then used to influence the locomotion behavior of their *LittleDog* quadruped. Finally Shao *et al.* [11] presented an obstacle avoidance approach for their quadruped robot that uses a stereo vision-based terrain modeling algorithm.

III. THE HYQ ROBOT, AND ITS STEREO SET UP

A. The HyQ robot

HyQ [12] is a versatile hydraulically actuated machine that weighs 80 kg, is 1 m long and 1 m tall (Fig. 1.a) and has upper and lower leg segments of 0.35 m in length. The robot's legs have three degrees of freedom each, two joints in the sagittal plane (hip and knee flexion/extension) and one joint for hip abduction/adduction. It is equipped with a PC104 for actuation control at 1 kHz.

B. Choice of the vision setup

On HyQ, vision is mainly needed to build a 3D model of the surroundings that is used to compute suitable footholds that allow the robot to overcome obstacles. The main features we have taken into account are:

- the camera height from the ground 1 m
- the resolution greater than the robot foot size (3 cm)
- the minimum size of the desired 3D model 2 m²

To obtain a map of the surroundings large enough at this close distance, we had to choose wide angle lenses. Also, we decided to mount the camera on a Pan and Tilt Unit (PTU) to enable some active motion and fixed the camera

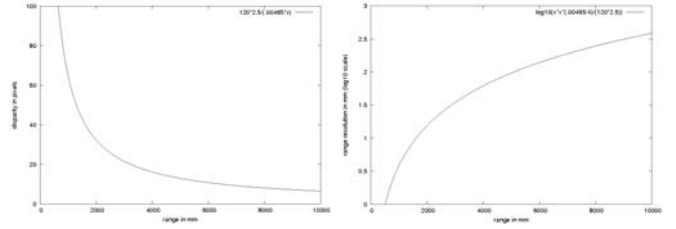


Fig. 2. Illustration of the depth capabilities and the depth resolution of our bumblebee camera. (a) Relationship between depth and disparity (cf Eq. 1), (b) Relationship between depth resolution and depth.

looking downwards with an approximate tilt angle of 55°. We also selected the stereo camera characteristics (baseline, image sensor, resolution, frame rate), trying to find the best compromise between accuracy and computation load.

The HyQ vision set up is a *Bumblebee2* firewire colour camera (*Point Grey*) mounted on a Pan and Tilt Unit (*Flir PTU-D46-17*). It is shown in Fig. 1.b.

- The camera has a focal length of 2.5 mm, a wide field of view of 97°, 2 CCD 1024 px × 768 px at 20 FPS, a sensor size of 4.80 mm × 3.60 mm, and a 12 cm baseline. It is accurately pre-calibrated with an accuracy of 0.11 pixel.
- The PTU has a pan range of ±159°, a tilt range of −47°/+31°, a maximum speed of 145 °/s, and a maximum control rate of 60 Hz.

The whole SLAM algorithm runs on a dedicated computer equipped with a quad-core Intel processor at 2.50 GHz and an NVIDIA GPU GeForce GTX 640.

C. Validation of the camera characteristics

In this section we will present some stereo vision basics to clarify our study of the camera characteristics. For more details on getting depth from stereo vision, the reader can refer [13]. Two images with slightly different viewpoints show the same object in different positions; the distance between the two is called disparity. Given a disparity value D , the 3D coordinates of the matched pixels in the two stereo images in m are computed using the projective camera equations: Eq. 1

$$Z = b \frac{f}{D}, \quad X = u \frac{Z}{f}, \quad Y = v \frac{Z}{f} \quad (1)$$

where b is the baseline, f is the focal length, and D is the disparity (in m), X, Y, Z are the coordinate of the object, and u and v are the location in the depth image (see Fig. 3, camera frame, image frame). All u, v, f and D are in pixels in Eq. 1.

Given a change in the disparity, the smallest change in depth that is detectable by the stereo geometry is the depth resolution r :

$$r = D \frac{Z^2}{b f} \quad (2)$$

Fig. 2.a shows that the minimum depth for this stereo camera is 0.65 m: at this point, the disparity is over 100 px.

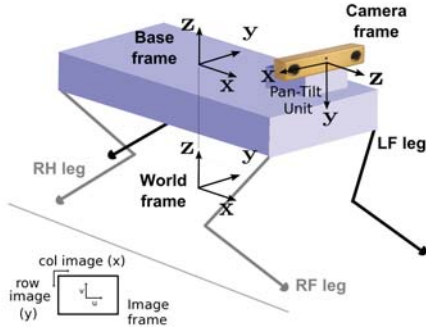


Fig. 3. Definition of HyQ reference frames: image frame, camera frame, base frame and world frame.

The maximum depth is above 40 m, but most of the change in disparity takes place in the first few meters. In our case, because we focus on building an accurate map in front of the robot's forward feet and we also want to reduce the computational load as much as possible, we set the stereo matching search window of disparities from 10 px to 74 px, which corresponds to a depth restricted to the interval 0.8 m to 4 m.

Fig 2.b shows the depth resolution r plotted on a log10 scale. At 1 m, the depth resolution is about 4 mm (assuming a system without video noise or matching errors). At 2 m, it has grown to 15 mm and at 3 m is 35 mm. This resolution justifies our decision of building a height map with minimum 5 mm^2 cells. It has to be noted that depth resolution is not the same as depth accuracy, which measures the difference between the depth computed and the actual depth. Depth accuracy is sensitive to errors in camera calibration, which is certified to be good on this fixed pre-calibrated stereo camera. Finally, the small focal length gives us a large field of view (about 97° and allows to cover 2 m^2 surface at 1 m of distance that ensures a good view on obstacles in front of the robot for foothold planning.

IV. ON BOARD SLAM WITH A STEREO CAMERA

SLAM is the process of incrementally building up a map within an unknown environment (without *a priori* knowledge), while at the same time keeping track of the current location of the robot. To implement this process we used a stereo camera and a code based on the Kinect Fusion algorithm for large scales [14]. The developed SLAM system, the terrain modeling and the path planning modules are described in Fig. 6.

A. Depth map computation

1) *Disparity computation*: The rectified images are extracted with the Point Grey proprietary library *Triclops* and the computation of the disparity for each pixel is performed using the Sum of Absolute Differences (SAD) method [15] on 4 threads. We used the SAD correlation based method because of its small computational load, that is suitable for real-time implementations. The disparity is computed on edges images, to allow matching on the changes in brightness

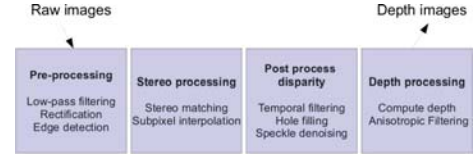


Fig. 4. Depth map computation flowchart.

rather than on the absolute pixel intensities. This approach is more robust in environment with variables light conditions, like outdoor settings, where our robot is intended to operate. In order to obtain a more accurate disparity we use a sub-pixel interpolation filter [16] on the disparity map. The filter attenuates the quantization effect and makes surfaces smoother (the disparities are interpolated up to 1/4 sub-pixel of accuracy).

2) *Filtering*: The SAD correlation method computes the similarity between pixels by comparing windows around pixels of interests. This method is fast but yields to several artefacts: in some regions disparities are uncertain and are left as gaps in the disparity map; this is mainly due to insufficient texturization of the surfaces or noise. To overcome this, we used different fast filtering methods to remove outliers and fill in holes. It has to be noted that only these filters are applied, no uniqueness, surface or texture check are applied. The filter we apply is the following:

```

for every pixel i in disp_t[]
  if disp_t[i] = 0xFF00 (outlier)
    then
      if disp_t-1[i] != 0xFF00
        then disp_t[i] = disp_t-1[i]
        else disp_t[i] = median of the
          3x3 neighborhood of pixel i
        end
      end
    end
  end
end

```

It uses the spatial and temporal information to fill in the disparity. Every pixel without disparity values is checked and small holes are filled. Then, outliers are removed by applying a speckle filter (see [17]). It removes spikes characteristic of mismatches in correlation.

As a reference, we compute on our dedicated computer the $640 \text{ px} \times 480 \text{ px}$ disparity at 15 Hz and then we extract the depth. The depth map is the pixel image that holds depth values, *i.e.*, distances from the camera to the 3D scene points.

B. Simultaneous Localization and Mapping

For the SLAM, we use a modified version of the Kinect Fusion (KinFu) Large Scale algorithm from PCL [14]. KinFu Large Scale is an extension of KinFu for building up larger maps. It performs high-quality reconstruction of geometrically accurate 3D models in real-time. KinFu was natively built for Kinect, it directly works with the depth map and performs surface reconstruction, which approximates more accurately the geometry of real world than point-based representations. It is also optimized for GPUs, which allows

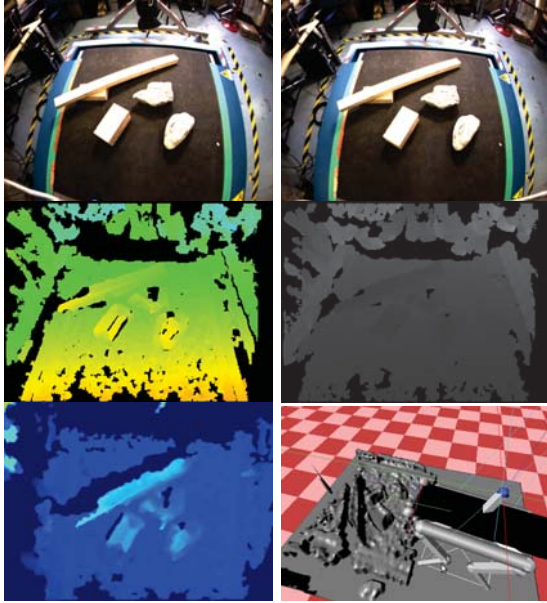


Fig. 5. Different steps of the height map building framework. (a) right image, (b) left image, (c) post processed disparity, (d) depth image, (e) height map built from one single point cloud, (f) terrain model and the robot in the simulation environment.

all the computation to be done in real time with a sufficient accuracy.

As the stereo vision module provides a depth map, we used it to feed the KinFu algorithm. The following sections describe the algorithm in more details.

1) *Depth Map conversion*: A depth map is converted into a 3D point cloud by using the calibration matrix (see Eq. 1). The result of this step is a point cloud with vertex and normal data for each point at three different levels of detail.

2) *Pose Estimation*: A modified ICP algorithm performs the alignment of two surfaces making the assumption that they are sufficiently close to each other. The ICP iterations are performed with three different resolutions and generate a six degrees of freedom transformation matrix that aligns the current point cloud with the previous ones.

3) *Surface reconstruction and meshing*: Ray casting is used for this surface reconstruction step. A prediction of the current global surface is obtained, with vertex data and estimated normal data. The refined ray-casted model is the same used in the next ICP step for alignment. By doing this, instead of using just the last frame point cloud as the source for alignment, a less noisy model is obtained.

C. Pan and tilt motion compensation

As this robot is subject to considerable motion during locomotion and since the ICP assumes that point clouds are close together, we added a motorized motion compensation system with a Pan and Tilt Unit based on the IMU information provided on the robot. This approach prevents that this assumption is broken, improves the SLAM results and reduces failures. To control our PTU motors we choose to implement a PD controller, that has been used to control

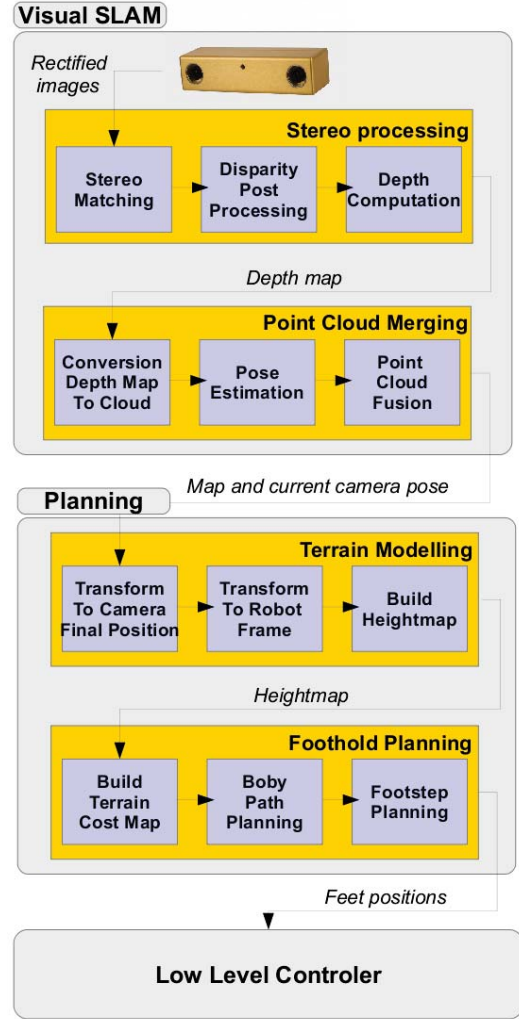


Fig. 6. Flowchart of the whole software developed.

most active head systems. For each axes, pan and tilt, we close the feedback loop in position. More details on the motion compensation controller and on the improvement of the SLAM can be found in [18].

V. FROM THE MAP TO THE ROBOT PATH PLANNING

Since we transformed all the point clouds into the same reference frame (*i.e.*, camera frame in Fig. 3). The goal is to build a full map of the terrain for foothold planning. Consequently the merged point cloud needs to be transformed in the robot frame, and the pose estimation needs to be used to position the robot in the map.

A. Calibration

To transform the map into the robot frame we need the full transformation between the camera frame and the robot frame. As the robot setup often changes — due to repairs, installation of new parts and sensors, or simply because of a change of application — we developed a module to compute automatically this transformation with high accuracy.

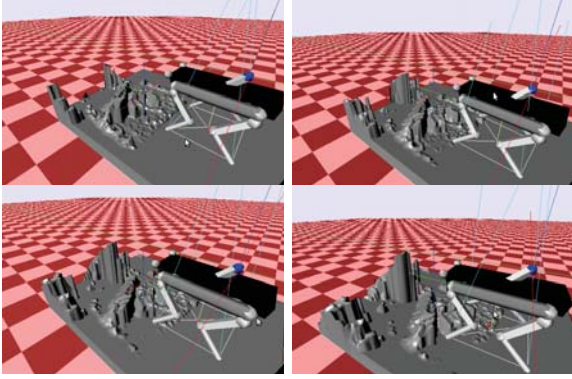


Fig. 7. The robots at different time stamp localized in its map while trotting. The robot has roughly moved 10 cm forward between each image.

The method consists in tracking a marker on the right front foot (see Fig. 1). As we know the position of the marker in the robot frame (thanks to the leg encoders and forward kinematics) and their positions in the camera frame (thanks to the visual tracking), we can estimate the transformation between the two.

1) *Tracking of markers*: For the tracking we use the color information with the Mean Shift algorithm. The method is easy-to-use, since there is no need of learning stage or parametrization. We implemented a modified version of the CAMShift algorithm [19]. This method sends at 20 Hz the position of the tracked marker inside the two stereo images.

For each tracker position computed in the camera frame, we associate the position of the tracker in the robot frame. A sample used for the calibration is defined as: $u_r v_r u_l v_l xyz$ where u_r and u_l are the row position (in pixels) of the barycenter of the marker in the right and left image respectively, and v_r and v_l are the column position (in pixels). Then x, y, z is the 3D position (in meters) of the marker in the robot frame (Fig. 3).

Note that the calibrated transformation of the camera is also expressed in robot coordinates.

2) *Calculation of calibration matrices*: For the camera calibration, we developed a flexible method that allows us to calibrate any kind of camera. We calibrate the position and rotation of the camera, as well as the intrinsic parameters. We chose a pinhole camera with Brown's distortion model [20]. The parameters that define our camera are:

- position and rotation: x, y, z , roll, pitch, yaw
- focal length and skew factor: f_x, f_y, s
- principal point and lens center: x_0, y_0, l_x, l_y
- 10 distortion parameters

Given a 3D point in space and using the above mentioned camera model, we can calculate the position of this point in the camera frame. Since we are tracking the point we know what is its actual position of this point in the image. The calibration error is defined as the sum of the squared distance of the predicted and measured marker position in the camera frame. By using a multi-variate Newton Raphson method we can iteratively minimize the error and find the camera position, rotation and internal parameters. We carried

out several calibrations and we got an average error below one pixel in less than 1000 iterations.

B. Transformation to world frame and height map

The merged point cloud is obtained in the initial camera frame. It is necessary to transform it to camera final position and next to the robot frame in order to be able to perform foothold planning. Eq. 3 represent this transformation.

$$X_f = P^{-1} C X_i \quad (3)$$

where P is the pose that is to say the final camera position in the initial camera frame and C the matrix representing the camera initial position in the robot frame.

Since we have of the map in the robot frame, the point cloud is projected in 2D where all the pixels of the image corresponds to a ($5 \text{ mm}^2/\text{px}$) surface, and finally some filtering is performed to remove the outliers, fill holes and improve the height map quality. Again we use the filter presented in Section IV-A.2 followed by an edge preserving smoothing (anisotropic filtering). Fig. 7 shows the robot in its maps in our simulation environment.

We chose to use a height map as a world model shared between the SLAM and planning modules because of it is efficient in terms of computation, analysis and storage space, and accuracy to perform foothold planning.

C. Terrain cost map for foothold planning

Height maps provide morphology information as 2D images whose pixel represents the vertical distance of a finite squared cell from the ground level. A straightforward approach to use such type of information is to create a map that associates a cost to each cell extracted from the height map and then compute the most suitable sequence of footholds that leads to the goal. The cost map penalize:

- High frequencies (*i.e.*, discontinuities on the morphology, such as the edges of a rock);
- Small uniform areas (*e.g.*, holes or small flat rocks), since the robot has a non negligible foot area and it could miss such foothold;
- The distance to the robot nominal leg position (56 cm). The maximum extension and the the minimum retraction are respectively 10 cm and 20 cm.

To achieve these goals, we extract information from derivatives of the image and then we assign a cost that depends both on the direction and the intensity of those derivatives. The cost map computation involves three steps:

- Edge detection: the edges of a height map reflects the actual discontinuities of the ground;
- Morphology operator: since the foot has a finite area, we want to avoid also areas around edges;
- Gaussian filter: the cost decreases proportionally with the distance from the computed edges.

The final cost map is then linearly combined to the original height map to take into account the height of the cells (Fig. 8.b). In this figure, the values of the colormap vary between dark cold colors for lower costs and dark warm

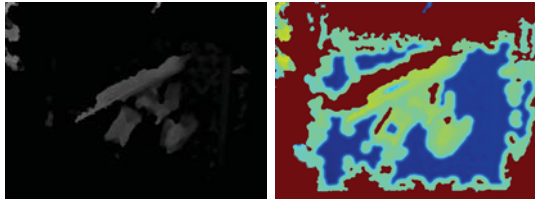


Fig. 8. Conversion of the height map (a) to a cost map (b).

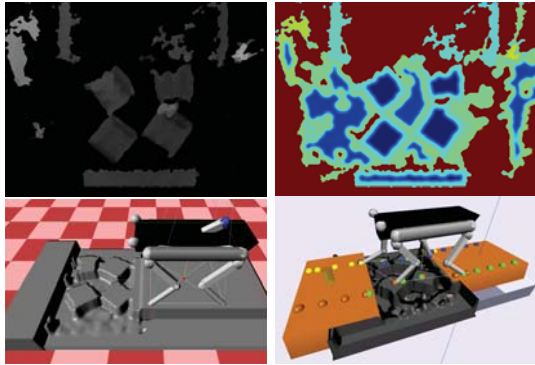


Fig. 9. Foothold planning results using an height map. (a) height map (b) cost map (c)(d) simulation environment without and with the footholds.

colors for higher costs. In particular, we compute the absolute difference between the robot ground level and the height map (for example the stepping stones in Fig. 9.b are darker than the ground because they are at the same level of the robot). Unknown heights were treated as special case by assigning them with the maximum cost.

VI. EXPERIMENTS ON FOOHOLD PLANNING

To experimentally validate our map building pipeline we use our foothold planner presented in [1]. This path planning framework compute from a cost map the appropriate footholds to overcome challenging terrain like stairs or stepping stones. It is based on *virtual model* based controller that guarantees the overall compliant behavior of the system. It has been demonstrated using maps a priori built with a Kinect and the robot execute blindly its planned trajectory. We refer the reader to [1] for more details about the foothold computation and the controller.

In this paper, we used the same algorithm but a map built online, directly. It has to noted that dynamic re-planning is not performed. The sequence of foothold is computed by the robot just before starting moving (Fig 9.c). Next when the foothold sequence has been accomplished, we can update our position and compute new footholds. The results show the effectiveness of our map building pipeline in order to compute footholds.

VII. CONCLUSIONS

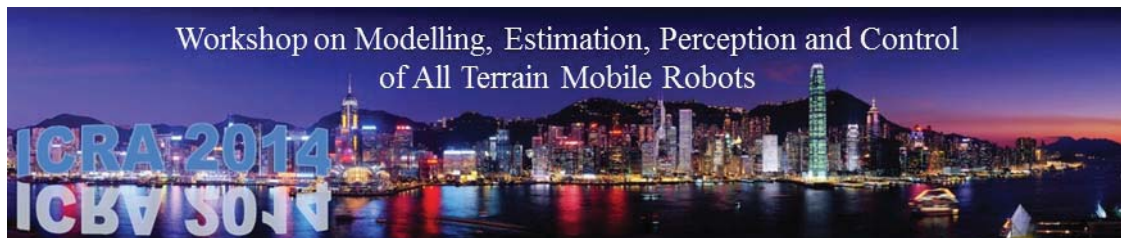
In this paper we propose a real-time SLAM solution using a pan and tilt stereo camera mounted on an hydraulically actuated quadruped robot that builds a map and keeps track of the robot's position. We validated the choice of the stereo vision set up, we detailed our stereo vision framework and

presented some results of path planning using the previously built map. Experiments of locomotion on our hydraulically actuated quadruped robot were successful with all the computation on board thanks to dedicated implementation based on fast stereo depth computation, GPU based map building and mechanical motion compensation. This localization and mapping ability allows to perform navigation on rough terrain in a fully planned manner.

Our future work will be to integrate the online dynamic re-planning to compensate the drift in case of slippage, improve the accuracy of the map and reduce SLAM failures by integrating the IMU measurement in the SLAM pipeline. Finally we would like to improve the disparity computation rate by adding FPGA computation, to relieve the vision computer, that today is fully used for stereo computation.

REFERENCES

- [1] A. Winkler, I. Havoutis, S. Bazeille, J. Ortiz, M. Focchi, R. Dillmann, D. G. Caldwell, and C. Semini, "Path planning with force-based foothold adaptation and virtual model control for torque controlled quadruped robots," in *IEEE ICRA*, 2014.
- [2] S. Bazeille, V. Barasuol, M. Focchi, I. Havoutis, M. Frigerio, J. Buchli, C. Semini, and D. G. Caldwell, "Vision enhanced reactive locomotion control for trotting on rough terrain," in *IEEE TePRA*, 2013.
- [3] J. Z. Kolter, K. Youngjun, and A. Y. Ng, "Stereo vision and terrain modeling for quadruped robots," in *IEEE ICRA*, 2009.
- [4] J. Pippine, D. Hackett, and A. Watson, "An overview of the Defense Advanced Research Projects Agency's Learning Locomotion program," *Int. J. of Robotics Research*, 2011.
- [5] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *Int. J. Robotics Research*, 2011.
- [6] A. Chiliani and H. Hirschmuller, "Stereo camera based navigation of mobile robots on rough terrain," in *IEEE/RSJ IROS*, 2009.
- [7] J. Ma, S. Susca, M. Bajracharya, L. Matthies, M. Malchano, and D. Wooden, "Robust multi-sensor, day/night 6-dof pose estimation for a dynamic legged vehicle in gps-denied environments," in *IEEE ICRA*, May 2012, pp. 619–626.
- [8] A. Howard, "Real-time stereo visual odometry for autonomous ground vehicles," in *IEEE/RSJ IROS*, 2008, pp. 3946–3952.
- [9] M. Bajracharya, J. Ma, M. Malchano, A. Perkins, A. Rizzi, and L. Matthies, "High fidelity day/night stereo mapping with vegetation and negative obstacle detection for vision-in-the-loop walking," in *IEEE/RSJ IROS*, 2013.
- [10] P. Filitchkin and K. Byl, "Feature-based terrain classification for littledog," in *IEEE/RSJ IROS*, 2012.
- [11] X. Shao, Y. Yang, and W. Wang, "Obstacle crossing with stereo vision for a quadruped robot," in *ICMA*, 2012.
- [12] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, "Design of HyQ - a hydraulically and electrically actuated quadruped robot," *J. of Systems and Control Engineering*, 2011.
- [13] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000, vol. 2.
- [14] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE ICRA*, 2011.
- [15] K. Mühlmann, D. Maier, J. Hesser, and R. Männer, "Calculating dense disparity maps from color stereo images, an efficient implementation," *Int. J. of Computer Vision*, 2002.
- [16] R. Szeliski and D. Scharstein, "Symmetric sub-pixel stereo matching," in *ECCV 2002*. Springer, 2002.
- [17] Z. Shi and K. B. Fung, "A comparison of digital speckle filters," in *Geoscience and Remote Sensing Symp. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., Int.*, vol. 4, 1994, pp. 2129–2133 vol.4.
- [18] F. Rovida, "Development of an active head for the hyq robot," 2014.
- [19] G. Bradski, "Computer video face tracking for use in a perceptual user interface," *Intel Technology J.*, 1998.
- [20] D. C. Brown, "Decentering distortion of lenses," *Photometric Engineering*, 1966.



Session II (1/2)

Perception in outdoor environment

- **Invited Talk: Paul Furgale (ETH Zurich, Switzerland)**
Title: There and back again: Dealing with highly-dynamic scenes and long-term change during topological/metric route following
Co-authors: P. Krüsi, F. Pomerleau, U. Schwesinger, F. Colas, and R. Siegwart
- **Title: Monocular Vision: A Real-Time Perception Toolkit for Mobile Robots in Outdoor Environments**
Authors: A. Miranda Neto, A. C. Victorino and J. V. Ferreira
- **Title: Multiplicative Extended Kalman Filter based on Visual Data for Attitude Estimation**
Authors: A. Seba, A. El Hadri, L. Benziane, A. Benallegue
- **Title: Gradient Based Multi-modal Sensor Calibration**
Author: Z. Taylor and J. Nieto

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Session II (1/2)

Invited Talk: **Paul Furgale**
(ETH Zurich, Switzerland)

There and back again: Dealing with highly-dynamic scenes and long-term change during topological/metric route following

Co-authors: P. Krüsi, F. Pomerleau, U. Schwesinger, F. Colas, and R. Siegwart

Abstract: Topological/metric route following, also called teach and repeat (T&R), enables long-range autonomous navigation even without globally consistent localization. This renders T&R ideal for applications where a global positioning system may not be available, such as navigation through street canyons or forests in search and rescue, reconnaissance in underground structures, surveillance, or planetary exploration. This talk will present our efforts to develop a T&R system suitable for long-term robot autonomy in highly dynamic, unstructured environments. We use the fast iterative closest point (ICP) algorithms from `libpointmatcher` (<https://github.com/ethz-asl/libpointmatcher>) to build a T&R system based on a spinning laser range finder. The system deals with dynamic elements in two ways. First, we employ a system-compliant local motion planner to react to dynamic elements in the scene during route following. Second, the system infers the static or dynamic state of each 3D point in the environment based on repeated observations. The velocity of each dynamic point is estimated without requiring object models or explicit clustering of the points. At any time, the system is able to produce a most-likely representation of underlying static scene geometry. By storing the time history of velocities, we can infer the dominant motion patterns within the map. The result is an online mapping and localization system specifically designed to enable long-term autonomy within highly dynamic environments. We validate the approach using data collected around the campus of ETH Zurich over seven months and at an outdoor 3D test site in Thun, Switzerland.

Biography: Dr. Paul Furgale received a BAsC (2006) from the University of Manitoba (Computer Science) and a PhD (2011) from the University of Toronto Institute for Aerospace Studies (UTIAS). He is now the Deputy Director of the Autonomous Systems Lab at ETH Zurich. His current research is focused on long-term autonomy for mobile robotic systems, including perception, mapping, localization, and planning over long timescales and in highly dynamic environments. He is the scientific coordinator for V-Charge, a European project and industry/academic collaboration that seeks to develop automated valet parking and charging of electric vehicles in mixed traffic. During his doctoral work, he developed algorithms to support over-the-horizon sample return for planetary exploration rovers as part of the Autonomous Space Robotics Lab. His PhD work was field tested in the Canadian High Arctic and subsequently integrated into several Canadian Space Agency rover prototypes.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



There and Back Again

Dealing with highly-dynamic scenes and long-term change during topological/metric route following

Paul Furgale¹, Philipp Krüsi¹, François Pomerleau², Ulrich Schwesinger¹, Francis Colas¹, and Roland Siegwart¹

Abstract—Topological/metric route following, also called teach and repeat (T&R), enables long-range autonomous navigation even without globally consistent localization. This renders T&R ideal for applications where a global positioning system may not be available, such as navigation through street canyons or forests in search and rescue, reconnaissance in underground structures, surveillance, or planetary exploration. This talk will present our efforts to develop a T&R system suitable for long-term robot autonomy in highly dynamic, unstructured environments. We use the fast iterative closest point (ICP) algorithms from `libpointmatcher`¹ to build a T&R system based on a spinning laser range finder. The system deals with dynamic elements in two ways. First, we employ a system-compliant local motion planner to react to dynamic elements in the scene during route following. Second, the system infers the static or dynamic state of each 3D point in the environment based on repeated observations. The velocity of each dynamic point is estimated without requiring object models or explicit clustering of the points. At any time, the system is able to produce a most-likely representation of underlying static scene geometry. By storing the time history of velocities, we can infer the dominant motion patterns within the map. The result is an online mapping and localization system specifically designed to enable long-term autonomy within highly dynamic environments. We validate the approach using data collected around the campus of ETH Zurich over seven months and at an outdoor 3D test site in Thun, Switzerland.

Index Terms—Long-term mapping, dynamic obstacles, ICP, kd-tree, registration, scan matching, robot, SLAM.

I. INTRODUCTION AND RELATED WORK

This short paper is an overview of our work at the ETH Zurich Autonomous Systems Lab to build a system capable of long-term topological/metric route following (often called teach and repeat (T&R)) that is suitable for navigation in rough terrain and dynamic environments. Our goal is to be able to deal with dynamic environments over all time scales, both slow scene changes over time, and the presence of moving obstacles such as pedestrians or cars. Our work follows the basic structure of the stereo-camera-based T&R algorithm described in [1]. However, as camera-based localization algorithms are inherently sensitive to changes in lighting, we have decided to build our system around a Velodyne HDL-32E spinning laser rangefinder (Fig. 1). While there have been other rough-terrain-capable T&R algorithms based on lidar in the past [2], [3], these systems used



Fig. 1. ARTOR, a search and rescue robot specialized for outdoor applications navigating in highly dynamic urban environments. Typical mobile elements include pedestrians, bikers, cars, trucks and trams.

an appearance-based approach that requires high-resolution intensity/range images. Instead, our work uses the iterative closest point (ICP) algorithm which is more suitable for our chosen lidar and produces highly accurate motion estimates.

This paper is not meant to present novel research. Rather, it summarizes two of our recent papers (often verbatim) and interested readers are referred to these papers for more details. Krüsi et. al [4] presents our ICP-based T&R system and describes how we included the capability for local motion planning to be able to repeat routes in highly dynamic environments, or when short sections of the route are blocked. Pomerleau et al. [5] describes our efforts to update maps to encode dynamic and static parts of the scene. These systems operate on the same input data but are currently separate. Our future work will aim to bring them together to become a T&R system that is capable of long-term operation in dynamic environments.

II. SYSTEM OVERVIEW

A. Fast ICP: `libpointmatcher`

Offline 3D mapping can be realized with many open source software packages currently available, such as Point Cloud Library (PCL), MeshLab and CloudCompare. When it comes to online registration applied to a mobile system, which needs to take real-time decisions based on its surroundings, specific requirements have to be handled carefully. For that reason, we based most of our registration solutions on `libpointmatcher` [6]. Its modularity allows the generation of solutions tailored to different kinds of platforms and locomotion, while maintaining real-time

¹The authors are with the Autonomous Systems Lab, ETH Zurich, and Laval University. Correspondence: paul.furgale@mavt.ethz.ch

¹ <https://github.com/ethz-asl/libpointmatcher>

capability even with large bandwidth sensors. Working applications have been presented for ground, water, and aerial platforms using multiple sensors ranging from the Kinect to the Velodyne HDL-32E, including the more traditional Hokuyo and Sick lidars [7].

B. ICP-based T&R

Robotic navigation in the context of T&R involves two distinct phases: the *teach pass* and the *repeat pass*. In the former, the robot is manually steered along the desired route, and the system builds up a map of the environment. In the latter, the robot autonomously repeats the route, using its sensor readings to localize within the map recorded in the teach pass. We use point clouds from a spinning 3D laser scanner (Velodyne HDL-32E) and ICP for incremental localization, map building, and map-based localization. Our system includes an obstacle avoidance scheme employed in the repeat pass (cf. Section II-C). The algorithm is able to detect obstacles obstructing the path and to plan avoidance maneuvers that temporarily leave the reference route whenever necessary, yielding an *adaptive* route following system.

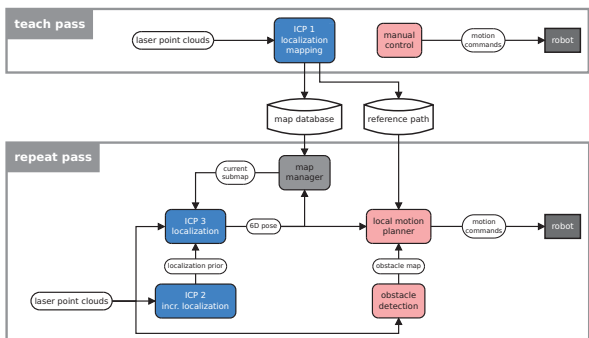


Fig. 2. System overview. Three main components are the basis of our adaptive route following system: a T&R framework, ICP-based registration modules, and an obstacle avoidance scheme (obstacle detection and local motion planning). The map built during the teach pass is saved in a database and used by an ICP-based localization module in the repeat pass, to obtain an estimate of the robot’s pose. Based on this and on a map of obstacles in the vicinity, the local motion planner computes safe commands for the robot, making it follow the desired reference path while avoiding collisions with obstacles.

Figure 2 gives a on overview of the complete system. In the teach pass, an ICP-based localization and mapping module is used to build up a database of connected local maps (a *pose graph*), and the driven route is stored as reference path. In the repeat pass, there are two different ICP processes: one for incremental localization, and one for localization within the previously created map. A map manager module accesses the map database and in each iteration selects the appropriate submap, based on the current pose estimate. The latter is input to the local motion planner, along with the reference path and a local obstacle map produced by the obstacle detection module, shown in Figure 3. Based on this data, the planner computes a feasible local trajectory and the corresponding motion commands for the robot.

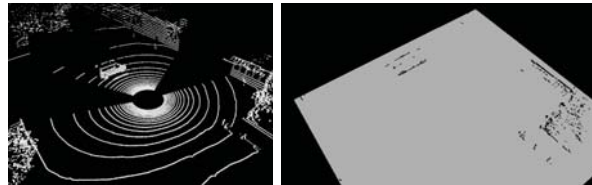


Fig. 3. An example of the input and the output of our obstacle detection module: a point cloud from the Velodyne 3D lidar (top), and the corresponding planar obstacle map (bottom). The scene shows a van on a large flat area, surrounded by buildings and some trees. The Velodyne consists of a series of individual lasers, spinning around the vertical axis. In a planar area, this results in an array of concentric circles. For obstacle detection, we exploit the fact that objects sticking out from the plane locally compress these rings.

C. Responding to Dynamic Elements: Local Motion Planning

Fully autonomous repeat runs necessitate functionality for addressing unforeseen changes and unmapped objects in the environment online. To this end, the navigation system needs to be able to autonomously bypass newly appearing objects obstructing any parts of the teach run’s reference path. The collision avoidance scheme on ARTOR is based on the sampling-based online planning framework presented in [8]. The framework enforces path alignment by shaping a tree of system-compliant motions along a reference path. It is based on a user-supplied system model and control law. The system model is (numerically) forward simulated towards samples drawn from a state-manifold aligned with the reference path. Within the framework, the state-manifold is defined as the heading and curvature aligned subspace of the robot’s state-space. An internal control law then regulates the simulated system towards a sampled version of this manifold. Within the forward simulation of the system model, any kind of system constraints can be enforced, making the approach appealing to constrained (such as non-holonomic) systems.

System Model. The highly configurable planning framework is able to incorporate arbitrary system models (suitable for forward simulation). The ARTOR system is modeled as a kinodynamic differential drive robot with constraints in both longitudinal and rotational velocity as well as longitudinal and rotational acceleration.

Sampling Scheme. The sampling scheme of the state-manifold used within the framework is also partially configurable. To generate candidate evasive trajectories, the framework samples lateral offsets from the reference path. To allow for longitudinal speed adaptations while driving, several target speeds of the robot are created by sampling speed offsets from the reference speed. The result of the state-manifold sampling is a tree of candidate trajectories, which is searched for the optimal candidate according to the cost function specified below. For the system described in this paper, we use a tree depth of one, as shown in Figure 4.

Collision Detection. Collision detection is performed on the two-dimensional binary occupancy grid shown in figure Figure 3. Objects are assumed to be static due to the lack of a classification module for dynamic objects in the

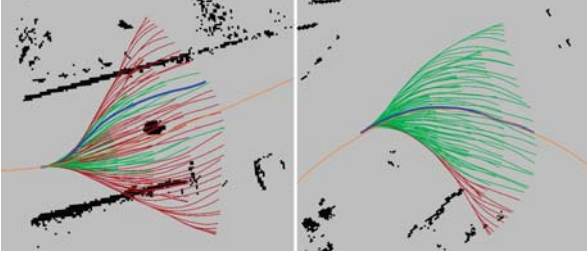


Fig. 4. Set of candidate trajectories shaped around the reference path, generated by the local planner for a fixed target speed and a single tree level. The figures show snapshots of the planner output during the T&R experiment described in Section III-A, containing the reference path (orange), the candidate trajectories (feasible: green, unfeasible: red), the selected trajectory (blue), and the underlying obstacle grid map. In the left figure, an evasive maneuver is planned due to an oncoming pedestrian obstructing the reference path. The trajectory sets contain 495 trajectories generated from 11 speed offset samples and 45 lateral offset samples (with a maximum lateral offset of eight meters).

current implementation of the obstacle detection, but are sufficiently accounted for through fast replanning at approximately 10 Hz.

Optimization Criterion. Naturally, the robot should follow the reference path as closely as possible while trying to keep a user-defined reference speed. The cost function is therefore divided into a lateral and longitudinal cost term. In our implementation, the lateral part is composed of the integrated distance between reference path and candidate trajectory over time plus a terminal cost that penalizes distance from the reference path at the end of a sample. Similarly, the longitudinal cost term integrates speed offsets to the reference speed and adds a terminal speed error.

Trajectories in collision with any object in the scene are assigned infinite costs. We include (soft) penalization of trajectories passing in close proximity to any collidable object in order to keep a desirable safety distance whenever possible.

D. Responding to Dynamic Elements: Map Maintenance

Combining local motion planning with T&R allows us to react to dynamic scenes but it doesn't help us to update our maps as the scene changes or to infer common motion patterns attached to particular places. Fig. 5 presents the general overview of our map-maintenance system that was designed specifically for these tasks. Our eventual goal is for this module to take the place of ICP2 and ICP3 in the full T&R system shown in Fig. 2. The *Registration* module takes point clouds from the sensor and computes its coordinates in a global reference frame while correcting odometry. The *Global Map Maintenance* module estimates if points in the global map are dynamic or not and integrates the new point cloud into the existing map. Finally, the *Velocity Estimation* module estimates the velocity of the dynamic points by comparing the dynamic elements at current and previous times.

a) *Dynamic Element Identification:* More specifically, the estimation of whether a given point of the map is dynamic or not is based on visibility assumptions: if a new point is

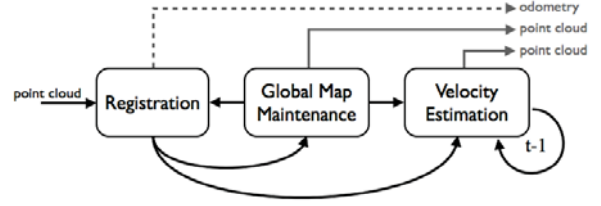


Fig. 5. Block diagram of the processing pipeline. Boxes represent separate processes running at different frequencies. Solid arrows represent point clouds being communicated to each module and the dashed line is odometry. The gray arrows show the output of each module.

observed behind a point in the map, that point was dynamic. In order to avoid the complex process of ray-tracing, we use spherical coordinates originating from the sensor. We can then use nearest neighbor search in this space to obtain points in the same conical aperture. It is implemented using the `libnabo` efficient kd-tree library [9].

We use a Bayesian approach to update the probability of a point to be dynamic:

$$P(Dyn|\theta, \phi, \delta) \propto \sum_{U, ODyn} \left| \frac{P(ODyn)P(U|\theta, \phi)}{\times P(Dyn|ODyn)P(\delta|Dyn)} \right| \quad (1)$$

where Dyn is the probability for a point in the map of being dynamic at given time and $ODyn$ at previous time; θ is the angle between the point in the map and the new point; ϕ is the incidence angle of the point in the map, δ the distance between the points, and U a binary variable indicating if we need to update the point.

b) *Velocity Estimation:* Building on top of the dynamic object classification, one can estimate the velocity of moving objects. Most approaches rely on the clustering of the points into objects for which the velocity is then estimated by looking at, for example, the change in position of the center of mass [10], [11]. From a newly acquired point cloud P_t at time t , we associate all of its points to the global map. A subset of mobile points M_t is generated from P_t , fulfilling the requirement of being a dynamic obstacle. Those dynamic obstacles M_t can then be compared to the last subset M_{t-1} to extract velocity vectors. We based our approach on point-cloud registration using ICP, where M_t is the *reading* point cloud and M_{t-1} is the *reference*. Having different transformation parameters for each point is known as non-rigid ICP [12]. We reuse the underlying principles but extracted only translation components instead of the full 6 degrees of freedom (DOF) transformation. In essence, we propose to do dual non-rigid ICP—both from reading to reference and from reference to reading—and, given that we have a timestamp per point, divide the alignment error with the difference of acquisition time to estimate the velocity vectors. We use a weak spatial smoothness prior to harmonize the velocities across nearby points.

III. EXPERIMENTAL RESULTS

A. ICP-based adaptive T&R

We tested our adaptive route following system in a dynamic urban environment around the ETH campus in Zurich.

The experiments were conducted with the robot ARTOR, a six-wheeled, skid-steered electric vehicle, equipped with—among other sensors—a 3D Velodyne laser scanner. We first evaluated the localization and mapping system, assessing its accuracy and its robustness to deviations from the path driven in the teach pass. The latter is crucial in the repeat pass when deviating from the original path to avoid dynamic obstacles. We further compared our approach to a state-of-the-art stereo-vision-based T&R system [1]. Ground truth with millimeter precision was provided by a Leica Total Station.

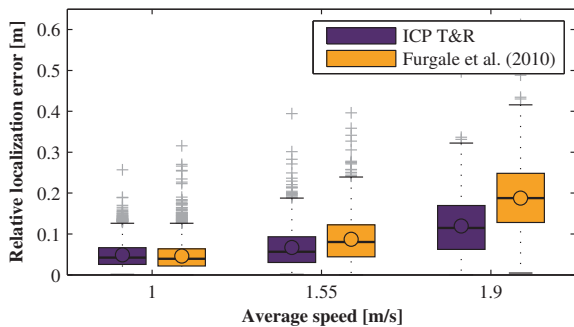


Fig. 6. Relative localization errors of the ICP-based T&R system in a dynamic urban environment. The bottom and the top of the boxes represent the first and the third quartile, respectively. That is, the box contains 50% of all measurements. The line inside the box is the median, the circle stands for the mean value, and the gray plus signs are outliers. The plot shows the results of six different runs (2 at each speed setting) along the same 130 m path. Both the laser-based and the stereo-based systems are very accurate at the typical speed of our robot. When increasing the speed, our system outperforms the stereo-based approach.

The results of the accuracy analysis are shown in Figure 6. Motivated by the relative nature of our system, we measured the *relative* localization error, that is, the error in the position estimate relative to the current map in the pose graph. For this experiment, we manually drove the robot seven times along a path of around 130 m, recording the laser point clouds as well as the stereo images. The first three runs were driven at the robot’s nominal speed of 1 m/s, the third and the fourth at around 1.5 m/s, and the last two at up to 2 m/s. All computations were run offline, but in real time on the same computer that is used on our robot. The first run was used to build the map (teach), and the remaining six passes to localize within this map (repeat).

To assess the system’s ability to detect and avoid obstacles, as well as the robustness of the localization to path deviations, we conducted an experiment where differently sized obstacles were put in the robot’s way. Using the full adaptive T&R system (ICP and obstacle avoidance), we first taught the system a straight path of around 23 m. We then let it autonomously repeat this route several times, while blocking the path with increasingly large obstacles. Figure 7 shows the four different obstacles used in the experiment, as well as the paths driven by our robot to avoid them. It can be seen that our obstacle avoidance scheme enables the robot to safely and smoothly drive around large obstacles. In this experiment, we deviated from the taught path by up

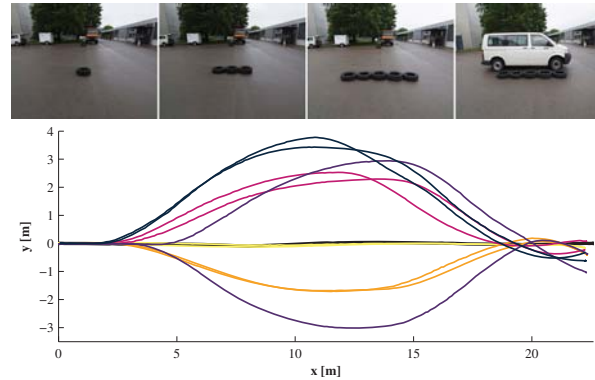


Fig. 7. Obstacle avoidance experiment. The experiment consisted of teaching a straight path, followed by autonomous repeat passes with an increasingly large obstacle in the middle of the path. The photographs on the top show the four different obstacle settings. The starting point of the robot was left of the orange truck in the background, the goal was behind the photographer. The graph on the bottom shows the result of our local planner: the paths autonomously driven by the robot to avoid the different obstacles, where brightness of the lines decreases with increasing obstacle size.

to 4 m, and the robot reliably found its way back onto the path after the obstacle had been passed. Figure 8 shows the relative localization errors of our system and the stereo-based framework as a function of the maximum deviation from the path. When the robot deviated from its original path, the error of the stereo-based system increased dramatically, while the performance of our laser-based approach remained nearly constant.

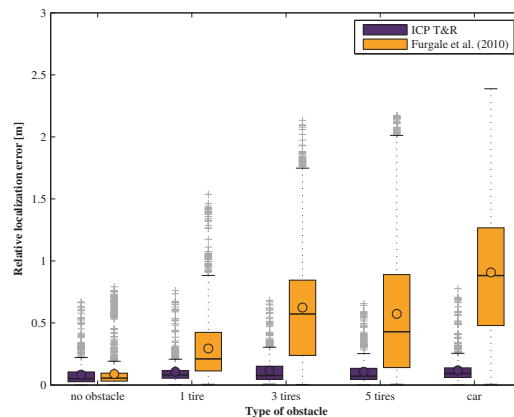


Fig. 8. Localization accuracy depending on the type of obstacle on the path. The size of the obstacle and the resulting deviation from the taught path increase from the left to the right. Each box plot contains the data of six experiments: the robot autonomously drove twice in each obstacle setting (using our ICP-based T&R scheme and the obstacle avoidance method), and we evaluated the data of each run three times with both T&R systems. The results clearly show the advantages of an omnidirectional, high-range sensor such as the Velodyne lidar over a directed sensor such as the stereo camera: it enables precise localization even when deviating from the taught path by several meters.

Having verified our system’s conformance to the requirements in terms of localization (accuracy, robustness to path deviations) and obstacle avoidance capability, we finally tested the performance of the complete system in an

extensive T&R experiment in a dynamic urban environment. We taught the system a 1.3 km path (Figure 9), and let it repeat the route eight times in autonomous mode. The first half of the path consisted of a quiet side street with parked cars on either side, the other half was a rather busy road with a lot of traffic including cars, trucks, bicycles and trams, as well as pedestrians. In total, our robot autonomously drove a distance of more than 10 km.

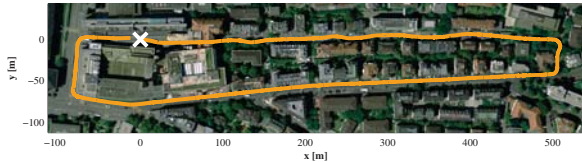


Fig. 9. The route in our long-range field tests in the urban environment around ETH Zurich. The path was a loop of around 1.3 km, which we drove in clockwise direction at a speed of approximately 1 m/s. Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000

The teach pass of the experiment was done during the night on March 12, 2013. The repeat runs were conducted in May, 2013 using the map recorded more than two months earlier. In the meantime, however, the environment had changed considerably: trees and bushes had leaves again, construction sites on the route had changed, and (most probably) none of the cars parked along the streets were still in the same place. Neither these changes, nor the dynamic elements (pedestrians, cars, trams), nor the different weather and lighting conditions (sunny/rain/snowfall, day/night) had any notable influence on the performance of our ICP-based localization system. Moreover, all the relevant static and dynamic obstacles (including strangers deliberately stepping into the way) have been detected and avoided reliably. Except from one single situation², the robot drove the entire distance in all eight repeat passes completely autonomously, without any intervention of the operator.

B. Responding to Dynamic Elements: Map Maintenance

The map-maintenance module was developed and tested independently from the T&R system. To test the identification of dynamic elements, we used the ARTOR robot to survey a visitor parking lot (see Fig. 1). This survey consisted of driving the robot around the area at different times during three consecutive days.

The resulting segmentation is shown in Fig. 11. The top panel shows the static map in which we can see the trees and the ground. The bottom panel shows the dynamic map in which we can see the cars and many points in the middle corresponding to bikes and pedestrians. More precisely, in (1) the static map shows a path while the dynamic map shows the trails of pedestrians. In (2) we can distinguish, in the dynamic map, the flow of bikes and pedestrians around parked cars.

As a second application of this framework, we surveyed a 1.3 km long route in the city of Zurich at three different times

²Probably due to reflections of the laser on the wet pavement, the obstacle detection module reported an obstacle that did not actually exist.

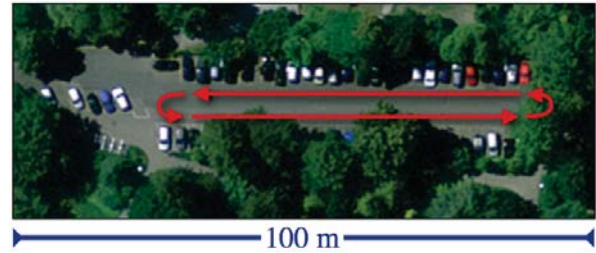


Fig. 10. Aerial view of the parking lot used for the segmentation experiment. In red, the survey path realized by the robot. Source: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000.

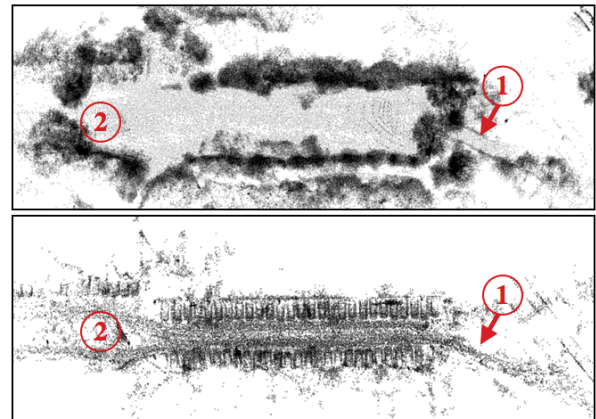


Fig. 11. Result of the segmentation after 9 surveys over the course of 3 days. Top: Reconstruction with $P(Dyn = true) < 0.5$. Bottom: Reconstruction with $P(Dyn = true) > 0.5$. The flow of pedestrians and bikes can be seen using a path (1) and splitting to avoid another row of parked cars (2).

in seven months (March, May and September 2013). The result is shown in Fig. 12. The middle plot shows the static map reconstructed with our algorithm. In the bottom plot, we show the amount of dynamic elements that can be used to indicate areas that can be potentially difficult or hazardous to navigate in.

The other experiment was conducted in front of the main building of ETH Zurich. It took place during the *information day*, which meant that many young students gathered in the streets, with sometimes as many as 15 persons in the vicinity of the robot. The main street consists of two large sidewalks, two lines for cars, and two lines for trams. The robot surveyed the area twice within 20 minutes, each time driving on the sidewalks on both sides of the street. Fig. 13 presents the results of the experiment. The two lower graphs show the extracted dynamic objects over the course of the survey, with their estimated speed and direction of motion. In the speed graph (left), blue corresponds to the range of typical walking speeds of pedestrians. The sidewalks and the pedestrian crossings (the latter marked with red arrows) can be clearly identified by looking at the blue objects. Furthermore, there are two lines of faster objects (yellow to red), which designate the car lanes. In the orientation graph (right), the two main directions of the cars are clearly visible. On the sidewalks the situation is naturally more chaotic, as

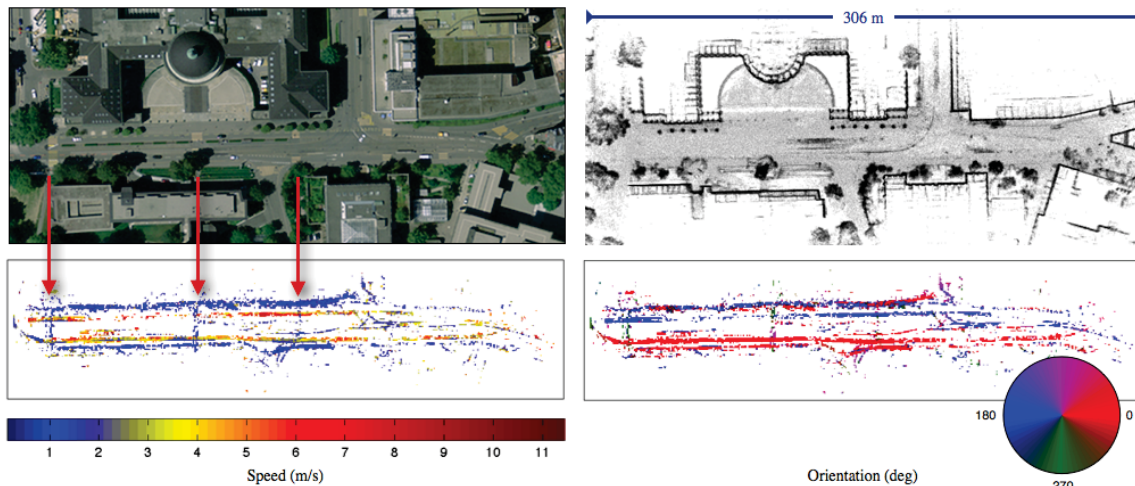


Fig. 13. Extraction of velocity information at a global scale. *Top left*: Aerial view of the street in front of ETH Zurich. *Source*: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000. *Top right*: 3D reconstruction after dynamic element removal. *Bottom left*: Average speed of the moving objects. *Bottom right*: Average orientation of the moving objects. The red arrows highlight the pedestrian crossings.

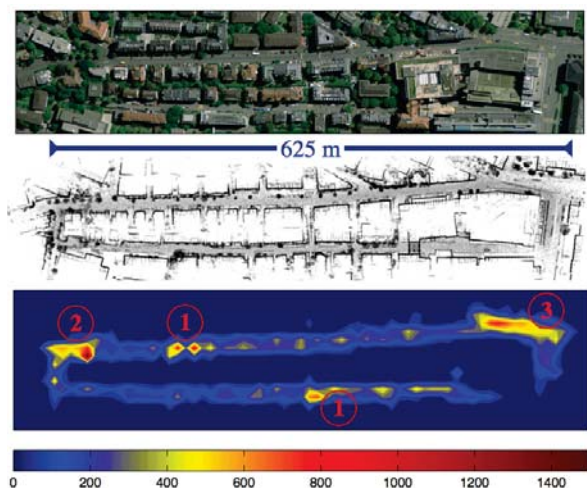


Fig. 12. Long range survey over a 1.3 km long path. The environment was monitored over a period of seven months. *Top*: Aerial view of surveyed area. *Source*: Bundesamt für Landestopografie swisstopo (Art. 30 GeoIV): 5704 000 000. *Middle*: 3D reconstruction after dynamic elements removal. *Bottom*: Occurrence of dynamic elements. The graph highlights the position of (1) construction sites, (2) a large tree and (3) a busy street intersection. Color represent to number of dynamic points over a cell size of 10 m.

pedestrians do not walk on distinct lanes.

IV. CONCLUSION

This paper provided a summary of our efforts to build a T&R system capable of adaptation when faced with dynamic scenes. For full details and more extensive experimental results, please refer to references [4] and [5].

V. ACKNOWLEDGMENTS

This work was supported by the EU FP7 IP projects Natural Human-Robot Cooperation in Dynamic Environments (ICT-247870), TRADR (ICT-609763), the armasuisse S+T UGV research program, and the EU FP7 2007-2013 Program, Challenge 2, Cognitive Systems, Interaction, Robotics,

under grant agreement No 269916, V-Charge. The authors are grateful to M. Ruffi and M.-È. Garneau for useful comments.

REFERENCES

- [1] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.
- [2] C. McManus, P. Furgale, and T. D. Barfoot, "Towards lighting-invariant visual navigation: An appearance-based approach using scanning laser-range-finders," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 836–852, 2013.
- [3] C. McManus, P. T. Furgale, B. E. Stenning, and T. D. Barfoot, "Lighting-invariant visual teach and repeat using appearance-based lidar," *Journal of Field Robotics*, vol. 30, no. 2, pp. 254–287, 2013.
- [4] P. Krüsi, B. Bücheler, F. Pomerleau, U. Schwesinger, R. Siegwart, and P. Furgale, "Lighting-invariant adaptive route following using icp," *Journal of Field Robotics*, 2014, conditionally accepted on March 17, 2014. (video).
- [5] F. Pomerleau, P. Krüsi, F. Colas, P. Furgale, and R. Siegwart, "Long-term 3d map maintenance in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 3 – June 7 2014. [Online]. Available: bib/pomerleau_icra14.pdf
- [6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, Feb. 2013.
- [7] F. Pomerleau, "Applied Registration for Robotics - Methodology and Tools for ICP-like Algorithms," Ph.D. dissertation, ETH Zurich, 2013.
- [8] U. Schwesinger, M. Ruffi, P. Furgale, and R. Siegwart, "A sampling-based partial motion planning framework for system-compliant navigation along a reference path," *Proceedings of the IEEE Intelligent Vehicles Symposium (IVS)*, 2013.
- [9] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter, "Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration," *Journal of Software Engineering for Robotics*, vol. 3, no. 1, pp. 2–12, Mar. 2012.
- [10] F. Moosmann and T. Fraichard, "Motion estimation from range images in dynamic outdoor scenes," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 142–147.
- [11] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, Sept. 2007.
- [12] J. Feldmar and N. Ayache, "Locally affine registration of free-form surfaces," in *Computer Vision and Pattern Recognition, 1994. Proceedings of the IEEE Computer Society Conference on*, 1994, pp. 496–501.



Session II (1/2)

Perception in outdoor environment

• **Title: Monocular Vision: A Real-Time Perception Toolkit for Mobile Robots in Outdoor Environments**

Authors: A. Miranda Neto, A. C. Victorino and J. V. Ferreira

• **Title: Multiplicative Extended Kalman Filter based on Visual Data for Attitude Estimation**

Authors: A. Seba, A. El Hadri, L. Benziane, A. Benallegue

• **Title: Gradient Based Multi-modal Sensor Calibration**

Author: Z. Taylor and J. Nieto

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICRA 2014



Monocular Vision: A Real-Time Perception Toolkit for Mobile Robots in Outdoor Environments

A. Miranda Neto, A. C. Victorino and J. V. Ferreira

Abstract—Many applications for control of autonomous platform are being developed and some important aspects are: (a) the estimation of drivable image area and (b) the excess of information, frequently redundant, that imposes a great computational cost in data processing. In this way, we have proposed (i) a robust algorithm for detecting the horizon line to generate (ii) the navigable area. It permits to investigate dynamically only a small portion of the image (road) ahead of the vehicle. Moreover, taking into account the temporal coherence between consecutive frames, we also have proposed a set of tools based on Pearson’s Correlation Coefficient: (iii) a discarding criteria methodology applied as (iv) a dynamic power management solution; (v) an environment observer method which selects automatically only the regions-of-interest; and taking place in the obstacle avoidance context, (vi) a method for collision risk estimation for vehicles in dynamic and unknown environments. Applying the PCC to these tasks has not been done yet, making the concepts unique. All these solutions have been evaluated from real data obtained by experimental vehicles.

I. INTRODUCTION

The perception of the environment is a major issue in autonomous and (semi)-autonomous systems. In the last three decades, visual navigation for mobile robots has become a source of countless research contributions.

The primary interest in this work, the machine vision is a powerful means for sensing the environment and is widely employed to deal with a large number of tasks in the automotive field [1]. For example, determining the area of free road ahead (and correlated information) is a key component of several driving assistance modules [2]. This modules use monocular camera systems instead of stereo camera systems because monocular systems have advantages in terms of reduced costs and the ease with which they can be fitted to vehicles [3]. The objective in terms of cost to fill ADAS functions has to be very lower than the current Adaptive Cruise Control (500 Euros) [4]. Additionally, the monocular vision contribution to the DARPA Grand Challenge [5] shows that the reach of lasers was approximately 22 meters, whereas the monocular vision module often looks 70 meters ahead.

To better understand our proposal, after acquiring the image by a monocular camera, the system fulfills its role through the tasks in the following layers: Section II (a) Environment observer method; Section III (b) Navigable area detection; Section IV (c) Autonomous navigation: a reactive

perception system. The results are presented in each respective section and the conclusions in Section V.

II. ENVIRONMENT OBSERVER METHOD

A. Pearson’s Correlation Coefficient

The Pearson’s Correlation Coefficient (PCC) [6] in (1) is widely used in statistical analysis, pattern recognition and image processing:

$$r_1 = \frac{\sum_i (x_i - x_m)(y_i - y_m)}{\sqrt{\sum_i (x_i - x_m)^2} \sqrt{\sum_i (y_i - y_m)^2}} \quad (1)$$

where x_i is the intensity of the i^{th} pixel in image 1, y_i is the intensity of the i^{th} pixel in image 2, x_m is the mean intensity of image 1, and y_m is the mean intensity of image 2.

B. Discarding Criteria methodology

The discarding criteria was presented as a simple solution to improve the performance of a real-time navigation system by choosing, in an automatic way, which images should be discarded and which ones should be treated at the visual perception system [7]. Basically, if the PCC indicates that there is a high correlation between a reference frame and another new frame acquired, the new frame is discarded without being processed (for example, the system can repeat a last valid command). Otherwise, the frame is processed and it is set as the new reference frame for the subsequent frame. In our case, the inclusion of an automatic image discarding method leads in a reduction of the processing time. Although the system spends some milliseconds computing the PCC, it gains much more time, in some cases, discarding more than 90% of the images.

C. Real-Time Dynamic Optimization

A robot can have many periodic tasks, such as motor and sensor control, sensing data reading, motion planning, and data processing. It may also have some aperiodic tasks, such as obstacle avoidance and communication. Moreover, for mobile robots, the tasks’ deadlines are different at different traveling speeds. At a higher speed, the periodic tasks have shorter periods [8].

Based on the PCC variation and by exploiting the temporal coherence between consecutive frames, it is proposed a new environment observer method [9]. This monocular-vision

system observes if there are no significant changes in the environment, permitting that some logical components may be shut down to save processor energy consumption, and/or to make the CPU available for running concurrent processes.

The Fig. 1 (a) shows an autonomous displacement through the Mojave Desert [10]. In Fig. 1 (b), due to PCC nature, taking the first frame of the Fig. 1 (a) as reference frame, a lower value of correlation is achieved when it is closer to the vehicle. That is, when the derivative approaches its maximum point (black line), there is the obstacle detection.

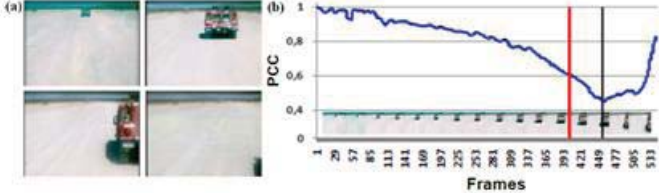


Figure 1. (a): The frames of the desert video [10]; (b) From a reference frame, its correlation with all others; Blue line: the Pearson's correlation in (1); the vertical black line: maximum point before collision; the vertical red line: Empirical Risk.

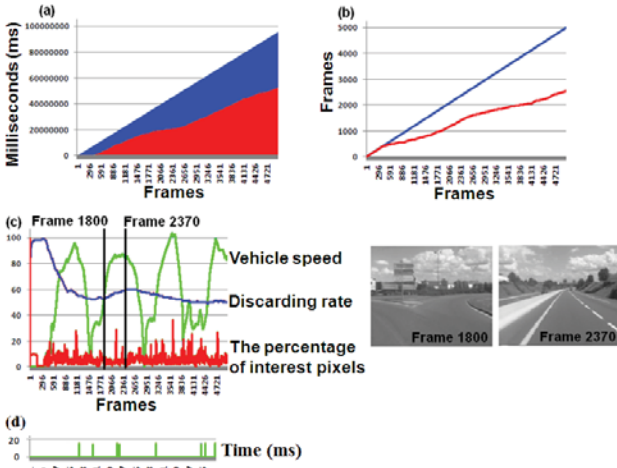


Figure 2. Real environment in France: (a) in blue: the cumulative impact computations (ms); in red: the cumulative computations (ms) by using the discarding criteria. (b) In blue: the number of frames; in red: the number of discarded frames by using the discarding criteria. (c) In blue: discarding rate; In red: the percentage of interest pixels; In green: The vehicle speed; In the analysis window, represented by two black vertical lines, the performance evaluation of the discarding criteria in acceleration from 37 Km/h to 86 Km/h; (d) Green line: computational time.

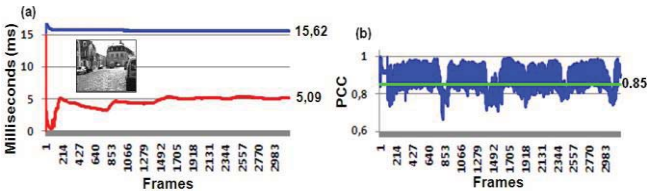


Figure 3. Real environment in France: The computational mean time of a horizon finding algorithm [11] in unknown and urban environment; (a) The red line: the computational mean time was 5.09 ms with the discarding criteria; (a) The blue line: the computational mean time was 15.62 ms without the discarding criteria; (b) The green line: an empirical PCC threshold equal to 0.85; (b) In blue: the performance based on discarding criteria: above the green line, it presents all discarded images.

The Figures 2 and 3 show the performance of this method in real, dynamic and unknown environments. For all these cases, the discarding rate remains over 65%. Fig. 3 (a) presents the computational mean time of a horizon finding algorithm [11] in unknown and urban environment. In this way, from an empirical PCC threshold equal to 0.85, the red line shows that the computational mean time was 5.09 ms, against 15.62 ms without the discarding criteria. In Fig. 3 (b), above the green line, it presents the discarded images.

D. Automatic Regions-of-Interest Selection based on PCC

One important aspect is the excess of information, frequently redundant, that imposes a great computational cost in data processing. We have considered a robot equipped with a vision perception system, and we proposed [12] an automatic regions-of-interest selection based on PCC, which processes it in real time.

According to the Pearson's correlation, in a certain analysis window (pair of frames), if the obstacle/object occupies a big portion of the scene, the PCC threshold tends to be low. Conversely, if obstacle/object occupies a small portion of the frame, it means that it is away from the vehicle and the system will have time enough to react. However, in real-time obstacle avoidance, for example, where are these interest points/pixels? Or, in a sequence analyzed, which pixels of the pair of images contributed most to the Pearson's coefficient computed? Which of them really need to be reprocessed?

Right after the Pearson's correlation in (1), for each pair of pixels analyzed in (2), the only possible result is: [-1 or +1]. That is, all pixels with intensities below these means will be candidates for interest points (ROI).

$$r_2 = \frac{(x_i - r_{1Xm})(y_i - r_{1Ym})}{\sqrt{(x_i - r_{1Xm})^2} \sqrt{(y_i - r_{1Ym})^2}} = \begin{cases} -1 \\ or \\ +1 \end{cases} \quad (2)$$

where x_i is the intensity of the i^{th} pixel in image 1, y_i is the intensity of the i^{th} pixel in image 2, r_{1Xm} and r_{1Ym} were obtained in (1): i.e.: x_m and y_m from (1).

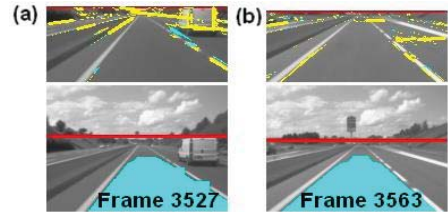


Figure 4. Real environment in France: After the horizon finding algorithm performance [11], red line: (a) Speed 97.01 Km/h, the interest pixels (yellow) represent 5% of the image; (b): Speed: 100.2 Km/h, the interest pixels (yellow) represent 2% of the image.

From these interest points, the Interactive Thresholding Algorithm (ITA) [12] reclassifies the background and foreground pixels based on Otsu Thresholding Method [13]. Fig. 4 *(b) present the results of this process, where the red pixels (interest points) represent $r_2 = -1$. Taking as base an image resolution equal to image 96x72 taking from a desert

video, by processing the images only when $r_2 = -1$, only about 205 thousand points were processed, instead of 3.7 million points. In off-road context the same result is observed, only about 10 million points were processed, instead of 48 million points [12], [14].

Results for different types of image texture (road surfaces) were selected and its results are presented in [14]. For obstacle avoidance task, the Fig. 4 presents results at high speed on real-time conditions.

E. Collision Risk Estimation based on PCC

The collision warning algorithms typically issue a warning message when the current range to an object is less than the critical warning distance, where the safety can be measured in terms of the minimum time-to-collision (TTC) [15]. To calculate the TTC, several techniques are presented in the literature [16], [17]. Measuring distances is a non-native task for a monocular camera system [16]. However, TTC estimation is an approach to visual collision detection from an image sequence. In this way, we have presented a novel approach to obtain Collision Risk Estimation (CRE) based on PCC from a monocular camera [18].

To better understand, let's go back to Fig. 1 where the robot Stanley has moved at an average speed of 30.7 km/h [19]. In Fig. 1 (b), a lower value of correlation is achieved when it is closer to the vehicle (black line). For our proposal an Empirical Risk-of-Collision needs to be defined. This is represented in Fig. 1 (b) by the red line. Taking into account this R_c , the CRE is estimate in (3):

$$CRE_s = \frac{R_c}{(1-r_1)} \quad (3)$$

where 1 (one) represents the reference frame and r_1 was obtained in (1) and $R_c = 1 - 0.6$.

Fig. 5 and Table I present the performance of the CRE in dynamic and unknown environment. These results were obtained in real conditions using our experimental vehicles. Since in real conditions this monocular-vision system has been designed to investigate only a small portion of the road ahead of the vehicle, where the absence of other vehicles has been assumed [1], the Fig. 4.a>(*a) presents the fix analysis region (yellow line). As shown in [18], the computational mean time of the CRE process was equal to 7.8 ms.

III. NAVIGABLE AREA DETECTION

A. Horizon Finding Algorithm (sky removal)

For land vehicle navigation, the monocular vision systems have been applied to investigate the road information, and in order to decrease the volume of data for processing, some systems have been designed to investigate only a small portion of the road ahead of the vehicle where it is unlikely the existence of other vehicles [1]. For example, some systems seek to identify the sky region because this is not a region of interest, and therefore the horizon line threshold is applied to subtract a road image [20]. Stanford Racing Team [5]

implemented the horizon finding algorithm originally proposed by [21] to eliminate all pixels above that horizon.

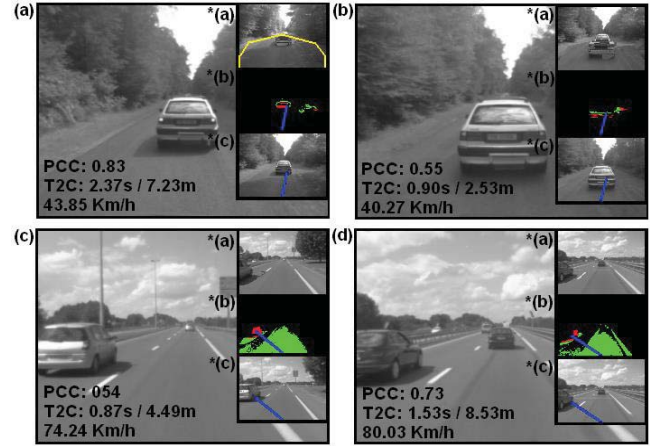


Figure 5. Real environment in France: *(a): the reference frame after the region-merging algorithm presented in [18]; *(b) ITA results [12]; *(c) Obstacle direction based on the center of area of the red points in *(b).

Frames	$(1-R_1)$	Variation in the Range	Risk of Collision	CRE in Sec(s)	Distance in Meters
(a) 1001	(1-0.8315)	0.1685	$(R_c / 0.1685)$	2.37s	7.23m
(b) 1024	(1-0.5584)	0.4416	$(R_c / 0.4416)$	0.90s	2.53m
(c) 1139	(1-0.5411)	0.4589	$(R_c / 0.4589)$	0.87s	4.49m
(d) 4654	(1-0.7394)	0.2606	$(R_c / 0.2606)$	1.53s	8.53m

TABLE I. Relationship between frames of the Fig. 7 and Collision Risk Estimation (CRE).

In this way, from a dynamic threshold search method, a robust horizon finding algorithm that finds the horizon line was proposed by us [11] and applied to generate the navigable area. It permits to investigate dynamically only a small portion of the image (road) ahead of the vehicle. This algorithm is robust to illumination changes and does not need any contrast adjustments. Although the Otsu method is an excellent method to choose an ideal threshold, it considers for all cases the information of the image as a whole (global information). After sky removal, it permits to investigate only a small portion of the image ahead of the vehicle, and the new result can be seen in Fig. 6.

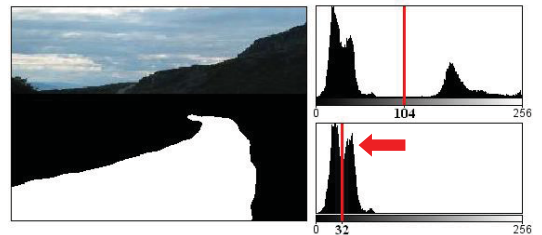


Figure 6. Left: Otsu segmentation after sky removal; Right: Global Otsu threshold (104); new Local Otsu threshold (32) after sky removal [11].

Figures 7 and 8 present the performance of this horizon line detection process in unknown environments: desert road videos available by DARPA [10], [19]; urban environment data obtained by our intelligent vehicles. For each image a real horizon line (black) was registered manually. In (a) the green line represents the Otsu horizon line detection; in (b) the magenta line represents the weighted average of the Hough

transformation lines; in (c) the blue line represents the weighted average between the Otsu horizon line detection and the Hough transformation [22] result; in (d), finally, the red line represents the robust horizon finding algorithm based on Otsu segmentation, Hough transformation and Kalman filtering. Fig. 9 presents the computational mean time of this algorithm in unknown environments. In the Fig. 9 (a) the mean time was as 6.27 ms without the discarding criteria [7]. In the Fig. 9 (b), from an empirical PCC threshold, the mean time was 1.20 ms with the discarding criteria [7].

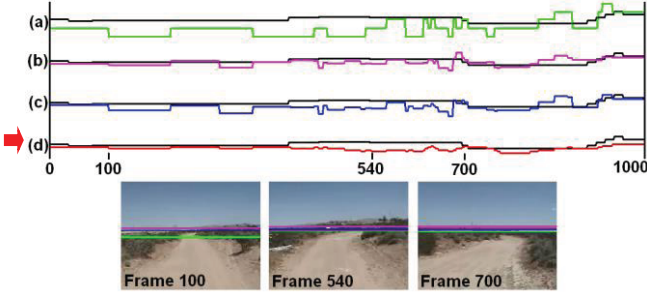


Figure 7. Desert road videos available by DARPA [10], [19].

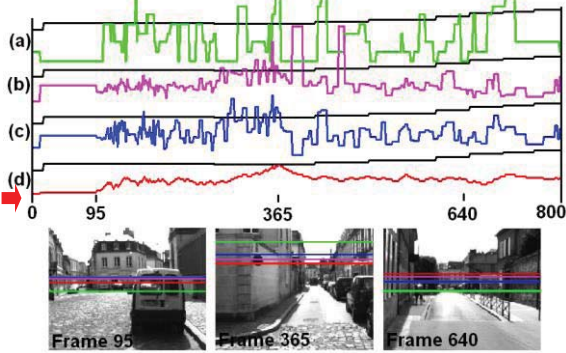


Figure 8. Real environment in France: data obtained by our intelligent vehicle.

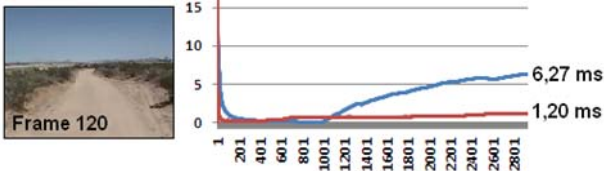


Figure 9. The computational mean time in desert roads.

B. Estimation of Drivable Image Area (Road Detection)

The vision-based sensors are defined as passive sensors and it can be used for some specific applications such as: road marking localization, traffic signs recognition and obstacle identification. To investigate the information's road, different techniques on automatic and semi-automatic road extraction methods are proposed in the literature [23], [24], [25], [26].

Aware that in the majority of the navigation systems, the machine vision system is working together with others low cost sensors, we have presented a monocular vision-based system that includes a robust road detection algorithm [27]. For this task, we apply the sky removal which we have proposed in [11], accelerating the process of identifying objects ahead of the vehicle because once using a global segmentation method, not always the analysis of a great portion of the image can contribute for a better result in the most critical region (region closer to the vehicle) where

obstacles should be detected and avoided as fast as possible. On the contrary, when discarding the superior portion of the original image, we are capable to get a more efficient segmentation and to distinguish with higher precision the obstacles from the navigable area.

For finding the drivable surfaces, [28] projects drivable area from the laser scan analysis into the camera image. This quadrilateral area is between 10 and 20 meters ahead of the robot. In this real-time approach, the basic idea is to consider a given region in the actual image as drivable. It assumes that the bottom center of the image contains road pixels for a large majority of the time [29], as we present in Fig. 10 (a): yellow region. This technique was first presented by [24].



Figure 10. (a) Original image after sky removal and its negated result in (b); (c) negated image and its Otsu's result in (d); (e) Canny edge detection result; (f) Hough transform result; (g) a single mass from the image (f).

In our proposed method, if the image is colored, in order to utilize the most important information of the color image, the candidate color channel that was dominant in the bottom center of the image is selected to generate the histogram image. A multimodal road image in (4) is then triggered based on the weighted average of the images intensities. In order to obtain a multimodal 2D drivability free-area, the algorithm performs the following tasks:

Task 1: Due to the different image textures on different roads, the original image, Fig. 10 (a), and its negated, Fig. 10 (c), are submitted to the Otsu's method. The algorithm then selects an image with the highest percentage of navigable area (white points) in the bottom center of the image, as shown in Fig. 10 (a): yellow area. The original image and its Otsu's result, Os^+ , can be seen in the Fig. 10 (a) and (b). The negated image and its Otsu's result, Os^- , can be seen in the Fig. 10 (c) and (d).

Task 2: Whereas there are homogeneous regions in the image, and in order to identify the limits of the road (which includes the obstacles), the Canny edge detector was employed as input of Hough transform [22] due to its robust performance and accurate edge localization. Respectively, the results can be seen in the Fig. 10 (e) and (f). Then, from the bottom center of the image, the algorithm concludes by finding a single image mass, Hc , Fig. 10 (g). It may also help to identify the textureless regions classified as road region, specular surfaces, traffic markings, etc.

$$FA_{w(x,y)} = \frac{FA_{w(x,y)} + Os_{(x,y)} + Hc_{(x,y)}}{3} \quad (4)$$

where $FA_{w(x,y)}$ (left) is the intensity of the i^{th} pixel after update in the new image FA_w , $FA_{w(x,y)}$ (right) is the intensity of the i^{th} pixel in the old image $FA_{w(x,y)}$, $Os_{(x,y)}$ is the intensity

of the i^{th} pixel in image obtained in Task 1, $Hc_{(x,y)}$ is the intensity of the i^{th} pixel in image obtained in Task 2.

Due the small variance of shades between objects, false path-markings, false route-markings, shadows, etc, the next step is applied to re-project this 2D drivability free-navigable area by considering a drag process which we presented in [27]. In fluid dynamics, drag forces act in a direction opposite to the oncoming flow velocity, i.e. forces (or resistance) that oppose the relative motion of an object through a fluid [30]. Right after applying our drag process, an example is shown in Fig. 11.



Figure 11. (a) Original image after sky removal and its Otsu's result in (c); (b) Canny edge detection result; (d) A multimodal 2D drivability free-area by considering the drag process.

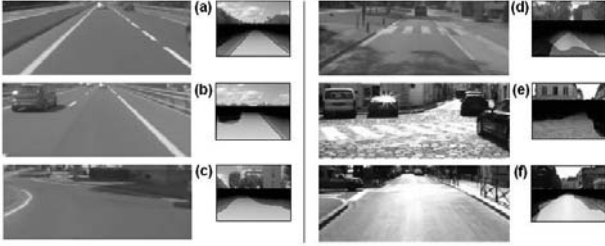


Figure 12. Real environment in France: urban and real experimental test-bank.

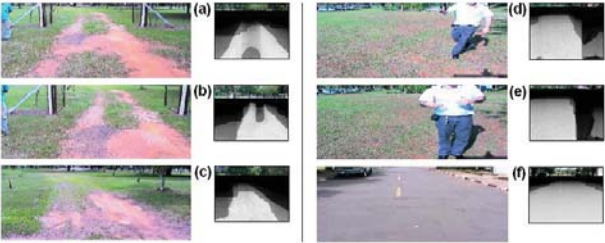


Figure 13. Real environment in Brazil: real-time autonomous displacement.

At first stage of testing, in order to evaluate the proposed algorithm performance, we used an urban and real experimental test-bank obtained using the vehicle shown in Fig. 15 (a). A typical urban environment was selected and its results are presented in Fig. 11. All false-navigable areas (red) in Fig. 11 (c) are eliminated by applying the drag process resulting in Fig. 11 (d). Additionally, different types of image texture (road surfaces) were selected and its results are presented in Fig. 12 (a) to (f). A result for a shadow context is presented in Fig. 12 (d). An occlusion case (vehicle) is shown in Fig. 12 (b).

At second stage of testing, in order to evaluate the proposed algorithm performance to autonomous displacement, the experiments on real-time conditions were performed using the vehicle VERO shown in Fig. 15 (b). Fig. 13 (a), (b) and (c)

show the successful task execution to go through a gate in off-road context. Fig. 13 (d) and (e) present the obstacle detection and our open-loop reactive navigation. In all experiments there was no collision. Additionally, a different type of image texture (road surfaces) was also selected and its result is presented in Fig. 13 (f).

IV. AUTONOMOUS NAVIGATION: REACTIVE PERCEPTION

Right after getting a 2D drivable road image in (4), we have two types of classes: c_1 and c_2 . The center of area is:

$$m_c = \sum_{x,y} h_c(x,y) \quad (5)$$

where $h_c(x,y) = \begin{cases} 1.se.S_{x,y} = c_1 \\ 0.se.S_{x,y} \neq c_1 \end{cases}$. The class c_1 (drivable area), denoted as class o_{c1} , is given by xo_{c1} and yo_{c1} :

$$xo_{c1} = \frac{\sum_{x,y} x.h_c(x,y)}{m_c} \quad \text{and} \quad yo_{c1} = \frac{\sum_{x,y} y.h_c(x,y)}{m_c} \quad (6)$$

Taking the origin point as:

$$x = \frac{image.width}{2}, y = image.height \quad (7)$$

and getting a destination point in (6), the steering angle correction (new direction) can be calculated in (8).

$$a = ((arc.cos(\frac{|(yo_{c1} - y)|}{\sqrt{(xo_{c1} - x)^2 + (yo_{c1} - y)^2}})) * 180) / PI \quad (8)$$

The final step is applied the sensor-based control [31] in the sensor space, where the robot's controller regulates to zero the error function of the reference trajectory. An example is presented in Fig. 14. The main objective is to drive the robot to the center of the drivable area.

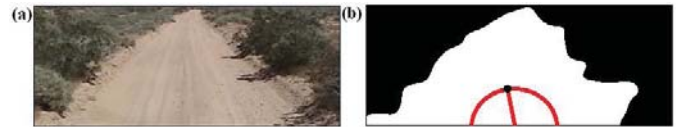


Figure 14. The center of area of the drivable area: the new robot direction.

V. CONCLUSION

A real-time perception problem is applied to autonomous robots. From an image captured by a single camera, the purpose was to present a real-time machine vision algorithm capable of estimating mainly the drivable area and the risk of collision.

The experiments showed that the inclusion of an automatic image discarding method based on PCC did result in a reduction of the processing time. This technique is also presented as an environment observer method and futures work will provide a real experimental test-bank to evaluate

the real energy consumption economy in terms of electrical current used by the visual machine.

A remarkable characteristic of all methodologies presented here is its independence of the image acquiring system and of the robot itself. For the long range navigation of intelligent electric vehicles, this could represent a considerable gain in the vehicle autonomy and will be studied for future implementations and in order to validate the proposed tools, future work would be also focused to provide ground truth measurements.

Besides the experimental DARPA test-banks, the results here were obtained using experimental vehicles on real, dynamic and unknown environments. Videos showing the application of these methods are available in [32] which include: the use of Karlsruhe Institute of Technology database; ROI Selection [33]; Collision Risk Estimation [34]; Sky Removal [35]; and Road Detection [36].

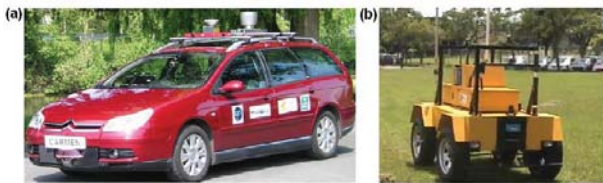


Figure 15. The experimental vehicles: (a) Carmen vehicle at Heudiasyc Laboratory in Compiègne, France; (b) Autonomous vehicle (VERO) at Renato Archer IT Center (CTI) in Campinas, Brazil.

ACKNOWLEDGMENT

The authors wish to thank Luiz Mirisola, Samuel Bueno, Josué Ramos, Hélio Azevedo and Gerald Dherbomez for their kind attention to this work and support in data acquisition. This work has been cofounded by ANR, FAPESP, CNPq and CAPES.

REFERENCES

- [1] M. Bertozzi, A. Broggi and A. Fascioli, "Vision-based intelligent vehicles: state of the art and perspectives". *Robotics and Autonomous systems* 32, 1–16, 2000.
- [2] F. Diego, J. M. Álvarez, J. Serrat and A. M. López, "Vision based Road Detection via On line Video Registration", *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2010.
- [3] K. Yamaguchi, A. Watanabe, T. Naito and Y. Ninomiya, "Road region estimation using a sequence of monocular images", *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, 2008.
- [4] Radio Spectrum Committee, European Commission, Public Document, Brussels, 5 July 2010, RSCOM10-35 http://ec.europa.eu/information_society/policy/ecom/radio_spectrum/_document_storage/rsc/rsc32_public_docs/rsc10_35.pdf [retrieved 02 Dec. 2010].
- [5] H. Dahlkamp et al., "Self-Supervised Monocular Road Detection in Desert Terrain", *Proc. of the Robotics Science and Systems Conference*, 2006.
- [6] K. Pearson, (1895), *Royal Society Proceedings*, 58, 241.
- [7] A. Miranda Neto, L. Rittner, N. Leite, D. E. Zampieri and A. C. Victorino, "Nondeterministic Criteria to Discard Redundant Information in Real Time Autonomous Navigation Systems based on Monocular Vision", *ISIC Invited Paper, Proc. of the IEEE Multi-conference on Systems and Control (MSC)*, 2008.
- [8] H. Y. L. C. Yongguo Mei, Yung-Hsiang Lu, "A case study of mobile robot's energy consumption and conservation techniques", *Proc. of the IEEE Int. Conf. on Advanced Robotics (ICAR)*, 2005.
- [9] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Real-Time Dynamic Power Management based on Pearson's Correlation Coefficient", *Proc. of the IEEE Int. Conference on Advanced Robotics (ICAR)*, 2011.
- [10] DARPA 2005. "DARPA Grand Challenge", <http://www.darpa.mil/grandchallenge05/> [June 10th, 2006]
- [11] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Robust Horizon Finding Algorithm for Real Time Autonomous Navigation based on Monocular Vision", *Proc. of the IEEE Int. Conference on Intelligent Transportation Systems (ITSC)*, 2011.
- [12] A. Miranda Neto, A. C. Victorino, I. Fantoni and D. E. Zampieri, "Automatic Regions-of-Interest Selection based on Pearson's Correlation Coefficient", *Proc. of the IEEE Int. Conference on Intelligent Robots and Systems IROS/ViCoMoR*, 2011.
- [13] N. Otsu, (1978), "A threshold selection method from gray-level histogram". *IEEE Transactions on Systems, Man, and Cybernetics*.
- [14] A. Miranda Neto, "Embedded Visual Perception System applied to Safe Navigation of Vehicles", PhD Thesis, UNICAMP-Brazil/UTC-France, 2011.
- [15] O. J. Gietlink, J. Ploeg, B. Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-the-loop simulations". *Mechatronics*, 19:1091–1104, 2009.
- [16] D. Müller, J. Pauli, C. Nunn, S. Görner, S. Müller-Schneiders, "Time To Contact Estimation Using Interest Points", *Proc. of the IEEE Int. Conference on Intelligent Transportation Systems (ITSC)*, 2009.
- [17] A. Negre, C. Brailon, J. Crowley and C. Laugier, "Real-time Time-To-Collision from variation of Intrinsic Scale", *Proc. of the Int. Symp. on Experimental Robotics*, 2006.
- [18] A. Miranda Neto, A. C. Victorino, I. Fantoni and J. V. Ferreira, "Real-time Collision Risk Estimation based on Pearson's Correlation Coefficient", *Proc. of the IEEE Workshop on Robot Vision (WORV)*, 2013.
- [19] Stanford Racing Team's Entry In The 2005 DARPA Grand Challenge, <http://www.stanfordracing.org/> [June 10, 2006]
- [20] King Hann Lim et al., "Vision-based Lane-Vehicle Detection and Tracking", chapter 13 of *IAENG Transactions on Engineering Technologies Vol. 3*, pp. 157-171, 2009.
- [21] S. Ettinger et al., "Vision-Guided Flight Stability and Control for Micro Air Vehicles", *Advanced Robotics*, Vol. 17: 617-640, 2009.
- [22] D. Ballard, "Generalized Hough transform to detect arbitrary shapes", *IEEE Trans. Pattern Anal. Machine Intell.* 13 (2), 111–122, 1981.
- [23] G. Aviña-Cervantes, M. Devy and A. Marín, "Lane Extraction and Tracking for Robot Navigation in Agricultural Applications", *Proc. of the IEEE Int. Conference on Advanced Robotics (ICAR)*, 2003.
- [24] I. Ulrich and I. Nourbakhsh, "Appearance-Based Obstacle Detection with Monocular Color Vision", *Proc. of the AAAI National Conference on Artificial Intelligence*, 866-871, 2000.
- [25] F. Diego, J. M. Álvarez, J. Serrat and A. M. López, "Vision based Road Detection via On line Video Registration", *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2010.
- [26] J. Chetan, K. Madhava and C. V. Jawahar, "An Adaptive Outdoor Terrain Classification Methodology using Monocular Camera", *Proc. of the IEEE Int. Conference on Intelligent Robots and Systems (IROS)*, 2010.
- [27] A. Miranda Neto, A. C. Victorino, I. Fantoni and J. V. Ferreira, "Real-Time Estimation of Drivable Image Area based on Monocular Vision", *Proc. of the IEEE Intelligent Vehicles Symposium (IV)*, 2013.
- [28] S. Thrun, et al. "Stanley, the robot that won the DARPA Grand Challenge", *Journal of Robotic Systems*, Volume 23, Issue 9, 661-692, 2006.
- [29] F. W. Rauskolb, et al., "Caroline: An autonomously driving vehicle for urban environments", *Journal of Field Robotics*, 25(9):674–724, 2008.
- [30] F. M. White, "Fluid Mechanics", 2nd Ed. McGraw Hill, 1986.
- [31] A. C. Victorino, P. Rives, and J. -J., "Safe navigation for Indoor Mobile Robots Part I: A Sensor-based Navigation Framework", Part II: Exploration, Self Localization and Map Building", *International Journal of Robotics Research*, v22, n12, 2003.
- [32] <http://www.youtube.com/user/kingdombr/videos> [Jan., 21th 2014]
- [33] <http://www.youtube.com/watch?v=VcUQVC1F8Xw> [Jan., 21th 2014]
- [34] <http://youtu.be/J8YuZIJFExk> [Jan., 21th 2014]
- [35] <http://www.youtube.com/watch?v=8KbZ1J0txUE> [Jan., 21th 2014]
- [36] <http://youtu.be/ZpEbRo32pY8> [Jan., 21th 2014]

Multiplicative Extended Kalman Filter based on Visual Data for Attitude Estimation

A. Seba, A. El Hadri, L. Benziane, A. Benallegue,

Abstract—In this paper a method for rigid body attitude estimation based on a new feature-tracking algorithm and a Multiplicative Extended Kalman Filter (MEKF) is proposed and developed. The new feature-tracking algorithm allows us to detect and to track the image features (points and lines). This algorithm is able to cancel the drift due to a prediction error accumulation when a long sequence of images is used. Indeed, in this algorithm we have added a correction of the prediction given by optical flow features, using criterion based on Euclidean distance only in the predicted search area. Thus, the extracted points and lines are used as visual data measurements, which enable us to estimate attitude when they are fused with gyros measurements using MEKF. Simulation results has been presented to show the effectiveness of the applied method.

I. INTRODUCTION

Attitude estimation of a rigid body means the determination of the orientation of the body frame with respect to a reference frame. Many applications have been carried out in different field, such as navigation systems, robotics, aerospace etc. For robotic applications, attitude estimation and control of a moving rigid body is often based on measurements from inertial sensors (IMU: Inertial Measurement Unit, INS: Inertial Navigation System). These lasts are subject to disturbances, measurement drift and noise and their use depends on the requirements of the application. An alternative to these methods is the use of data-based vision systems, based on the image features detection like points, lines applied in [1],[17],[19],[22],[24], and optical flow technique to analyse the motion in [13],[10],[9]. In [22], an iterative algorithm based on fusion of visual data using dynamic cooperation between stereo matching and temporal matching processes is proposed to reconstruct the motion of a mobile robot. In [21], ego-motion estimation algorithm is based on IMU measurements and visual information. In [19], a method of attitude estimation for indoor navigation is proposed based on the use of vanishing points and lines. A review of techniques to motion recovering of rigid bodies by knowing the location of their image corresponding features at different times is presented in [15]. In [24], the authors propose an observer that fuses data from inertial sensors and vision system. They use line correspondence based algorithm assuming that the direction of lines is known. In [17] the authors proposed an identifier based observer to estimate the range by tracking images features assuming the known of camera motion. For most of these methods, the camera is used as a projective

system and the extracted information is in 2D however the points and lines present in the environment are represented in 3D. This requires correspondence between the visual features extracted from the initial image and those obtained from the current ones. Extraction and tracking of image features has been widely studied in literature [12], [14], [20]. To track point features, author in [5] proposes an algorithm based on pyramidal decomposition with an iterative implementation of the Lucas-Kanade optical flow computation. This algorithm ensures sufficient local tracking accuracy but when tracking a point over a long sequence of images, the point may drift. In [11], the authors combine the method of sum-of-squared-difference (SSD) criterion and Tomasi-Kanade's algorithm in order to improve features tracking over a sequence of images. In [18], the authors combine block-matching and differential algorithm to refine optical flow by fusing data from IMU based on Extended Kalman Filter. For the line tracking, a few studies have been proposed in [27], [7]. In [27], the authors propose the token tracker algorithm that combine prediction based on Kalman filtering approach and a matching process. This algorithm has been used and implemented to track line segments in [7] and to recover 3D motion and structure in [8], [22].

In this work, we considered an application using visual data to estimate the rotational movement of a rigid body. Visual data used here are the points and lines extracted from images captured by a camera. To analyse a long sequence of images, we propose a new algorithm that can accurately tracking image features without drift. Then to track points and lines, the matching is performed using optical flow approach with updating predicted displacement based on prediction error measurement. The goal of the algorithm proposed is to improve accuracy of feature tracking over a sequence of images which is similar to the method proposed in [11], [18]. The main difference is based on the measurement of the prediction error using criterion of the Euclidean distance without necessarily combine two approaches of matching and without the need of other sensors. In addition, the proposed procedure of lines tracking doesn't need to detect them in each frame as in [7]. Thus, this algorithm is used to detect and track the lines in the 3D scene and corresponding measurements are fused with gyros measurements using MEKF. To demonstrate effectiveness and performance of our method, some tests are performed using a sequence of synthetic images and results are very promising.

The paper is outlined as follow: section 2 present some preliminaries and introduce the problem posed by tracking images features and measurement to be used for motion estimation.

Ali Seba (ali.seba@lisv.uvsq.fr), A. El Hadri (elhadri@lisv.uvsq.fr), L. Benziane (lotfi.benziane@lisv.uvsq.fr), A. Benallegue (benalleg@lisv.uvsq.fr), are all with Versailles Engineering Systems Laboratory (LISV), 10/12 Avenue de l'Europe 78140 Velizy, France.

Section 3, describe the proposed feature tracking algorithm and section 4 gives the structure of the MEKF proposed for attitude estimation. The simulation and experimental results are presented in section 5 and 6, which conclude this work.

II. PRELIMINARIES AND PROBLEM STATEMENT

We consider a moving rigid body equipped with a camera in 3D space we denote by I the inertial frame attached to the 3D space and by B the body-attached frame. We use a standard pinhole camera where the perspective projection describes the relationship between a 3D point and its corresponding 2D projection onto the image plane. We suppose that the center of the perspective projection (focal point) coincides with the B -origin and the optical axis is aligned with the z-axis of the B -frame.

A. kinematic model

Attitude of the rigid body represents the orientation of the body-frame B with respect to inertial frame I . Representing the attitude by rotation matrix, R , provides an unique and global parametrization of the orientation in [25]. The rotation matrix is an element of the special orthogonal group $SO(3) = \{R \in \mathbb{R}^{3 \times 3} \mid R^T R = R R^T = I, \det(R) = 1\}$ where I is the 3-by-3 identity matrix. The unit quaternion Q is another global and nonsingular representation of the attitude and it is an element of $S^3 = \{Q = (q_0, q), q_0 \in \mathbb{R}, q \in \mathbb{R}^3, q_0^2 + q^T q = 1\}$. Both $Q \in S^3$ and $R \in SO(3)$ are related to each other through the mapping $\mathcal{R} : S^3 \rightarrow SO(3)$ by the Rodriguez formula as follows:

$$\mathcal{R}(Q) = I + 2S(q)^2 + 2q_0 S(q) = R, \quad (1)$$

where $S(q) \in \mathbb{R}^{3 \times 3}$ is the skew-symmetric matrix such that for $x, y \in \mathbb{R}^3$, $S(x)y = x \times y$, where \times denotes the vector cross product.

Now, let a point P denoted by \bar{x} its coordinates in inertial frame and by x its coordinates in B -frame. These two coordinate vectors are related via rotation matrix R as:

$$x = R^T(\bar{x} - \bar{r}) \quad (2)$$

where \bar{r} is the vector from the origin of the inertial frame to that of the body frame. Generally, any vector \bar{u} in the inertial frame is related to its corresponding projection vector u in the B -frame by $u = R^T \bar{u}$. Using the rotation matrix representation, the camera motion results from the kinematics of rigid body given by:

$$\dot{R} = R S(\omega) \quad (3)$$

where ω being the angular velocity of the rigid body given in B -frame. Also in term of unit quaternion, we can have $\dot{Q} = \frac{1}{2} Q * (0, \omega)$ or equivalently, using the properties of quaternions:

$$\dot{Q} = \begin{bmatrix} \dot{q}_0 \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} q^T \omega \\ \frac{1}{2} (q_0 I + S(q)) \omega \end{bmatrix} \quad (4)$$

Now, let, l , be a 3D line segment fixed in inertial frame with a vector direction \bar{u} (see figure 1). This line can be represented in B -frame by the following equation:

$$x = x_0 + \lambda u \quad (5)$$

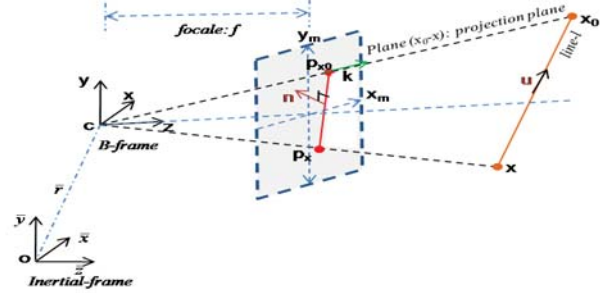


Figure 1. Line representation and perspective projection

where u is the vector direction of the line l expressed in B -frame such that $u = R^T \bar{u}$ and x_0 is any point in the line l . $\lambda \in \mathbb{R}$ is a scaling factor. From the equation (5), we can write:

$$(x - x_0) \times u = 0 \quad (6)$$

we can then represent the line l with a couple of Plückerian coordinates (u, n) where the vector n is the normal to the plane (x, x_0) . Then, we can write:

$$n = x_0 \times u \quad (7)$$

Since the vector direction u of the line l is perpendicular to the normal n , we can have the following relation:

$$u^T n = 0 \quad (8)$$

Then, the direction vector \bar{u} and observation vector n are related using rotation matrix R (see equation 8) by:

$$\bar{u}^T R n = 0 \quad (9)$$

B. Image features and camera motion

By using a camera as a vision sensor, we intend to use issued information from a sequence of image in order to describe the motion of the rigid body. Indeed, tracking image features such as points and line segments through a sequence of image can be applied in this case. Using the pinhole camera model, the relationship between a 3D point M of coordinates $x = (X_M, Y_M, Z_M)^T$ and its corresponding 2D perspective projection at position (x_m, y_m) onto the image plane (see figure 1), we can write:

$$x_m = f \frac{X_M}{Z_M}, \quad y_m = f \frac{Y_M}{Z_M} \quad (10)$$

where f is the focal length of the camera.

The projection of the 3D line l , defined previously, corresponds to line d on the image plane given by the intersection of the image plane and the plane defined by the camera focal point and the line l . A line 2D in the image plane can be described by the following equation:

$$\alpha x_m + \beta y_m + \gamma = 0 \quad (11)$$

From this parametrization, we can get a vector $N = (\alpha, \beta, \gamma)^T$ which is perpendicular to line d . The normalize value $n = \frac{N}{\|N\|}$ is the normal to the plane of projection (see figure 1). Thus, from measurements of the projected line l , we can derive the

normal vector n satisfying equation (8). This makes it possible to determine the motion of a rigid body based on the extraction and tracking image features. For this, we propose a method to track some lines in the image plane that correspond to the projection of non-collinear fixed lines in the inertial frame in order to estimate the rotational motion of the camera. The method is based on detection and tracking of points and line segments from a given sequence of images. The line segments can be obtained by edge detector and Hough Transform algorithm [16]. The detection of lines can be performed on the first frame. Then, to track these lines along the sequence, we propose first to extract a set of neighboring points of these lines. For this we use the Harris and Stephen corner-detector algorithm [12]. Indeed, for each selected line, the quality of a point with coordinates $p = [x_p, y_p]^T$ picked as a feature to be tracked is based on the following criterion:

$$C(p) = \det(G) + \alpha \times \text{trace}^2(G) \quad (12)$$

This criterion is computed on the window $W(p)$ centered at p with gradient matrix $G(p)$ given by:

$$G(p) = \begin{bmatrix} \frac{\sum_W I_x^2}{W} & \frac{\sum_W I_x I_y}{W} \\ \frac{\sum_W I_x I_y}{W} & \frac{\sum_W I_y^2}{W} \end{bmatrix} \quad (13)$$

where I_x and I_y are the gradients obtained by convolving the image intensity I with the derivatives of a pair of Gaussian filters and α is a constant parameter (the used default value $\alpha = 0.04$).

A point of the considered line is selected if the criterion $C(p)$ exceeds a certain defined threshold. Thereafter, the selected points will be tracking along the image sequences. The goal is to find the location of each point in the next frame and so forth along the sequence. This can be done by establishing correspondence between patches of the current and previous image using some approaches like differential and gradient methods ([14], [2], [3], [4] or correlation method [23]. In our work we used the pyramidal implementation of the Lucas Kanade feature tracker, see [5]. This algorithm is known to be accurate and robust with respect to changes of lighting, size of image and it is capable to handle large motions. However, this algorithm provides sufficient local tracking accuracy but when tracking a point over a long sequence of images, the point may drift. To remove this drift, we propose a new algorithm to improve features tracking over a sequence of images.

To summarize, by using the points and lines extraction and tracking, we can get the normal vector n corresponding to line l from its 2D line projection d . For a point x of the line l , its corresponding 2D perspective projection at position $p = [x_p, y_p]^T$ onto the image plane can be defined by unit vector k of coordinates $(x_p, y_p, 1)$ in the camera frame. Then according to equation (7), we can write:

$$n = k \times R^T \bar{u} \quad (14)$$

Thus the measurements of vectors n and k allows us to design the observer to estimate attitude of the rigid body. We assume that errors on vector measurement are assumed isotropic and measurements are given with additive gaussian white noise,

ν_c and we can write:

$$n_m = n + \nu_c \quad (15)$$

$$k_m = k + \nu_c \quad (16)$$

The angular velocity measurements ω_m given in the body-frame can be modeled by considering additive gaussian white noise ν_ω and slowly time-varying gyro bias b_ω as follows:

$$\omega_m = \omega + b_\omega + \nu_\omega \quad (17)$$

where ω is the real angular velocity and with the gyro bias is driven by gaussian white noise ν_b as:

$$\dot{b}_\omega = \nu_b \quad (18)$$

III. FEATURE TRACKING ALGORITHM

In this section, we describe our proposed method to improve algorithm of features tracking. Initially, we detect lines in the first frame, by using the Hough Transform algorithm [16]. To track selected lines over a sequence, we extract a set of points for each line (step (1) in figure 2). In fact, a point with coordinates $p_i = [x_{ip}, y_{ip}]^T$ of a line l_i is chosen to be tracked if the criterion $C(p_i)$ exceeds a threshold τ (see equation 12). Thus for a line l_i , the set E_i of points to be tracked is defined as:

$$E_i = \{p_i \in l_i : C(p_i) \geq \tau\} \quad (19)$$

The set E_i must contain at least two points. To ensure the robustness and accuracy of feature tracking, we use the procedure of pyramidal decomposition with an iterative implementation of the Lucas-Kanade optical flow computation as described in [5]. This one allows us to compute the displacement d_i of the point in the next frame by using matching process based on similarity function between two successive images. This displacement is a prediction of the location of the point p_i in the next image and does not necessarily represents the real one (step (2) in figure 2). Then, to determine the actual location of the point, we will try to evaluate the prediction error. For this, we start by selecting the search area that is centered at predicted point location $(p_i + d_i)$ and contains the real pixel location, (step (3) in figure 2). Afterward, we measure the quality of the candidate points in the search area by using criterion (12), (step (4) in figure 2). The set of candidate points is defined as:

$$S_i = \{\tilde{p}_j \in W : C(\tilde{p}_j) \leq \tau\} \quad (20)$$

where the window W corresponds to the search area and centered at predicted position $(p_i + d_i)$ of point in the next frame. $\tilde{p}_j = [x_{i\tilde{p}}, y_{i\tilde{p}}]^T$ denotes point coordinates in the second frame. To determine the prediction error, we use a criterion based on the Euclidean distance[23]. This involves to define a radius of search area that depends on desired accuracy of the tracking algorithm. If no point is found in the search area (i.e.. the set S_i is empty), then the feature point is lost. If some candidate points are found in the search area, then the selected one is chosen with a minimum error obtained by the criterion of the Euclidean distance defined as follows (step (5) in figure. 2):

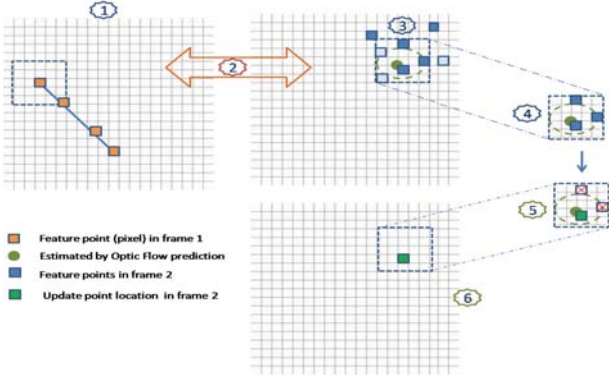


Figure 2. Tracking procedure

$$\varepsilon = \min \left(\sqrt{(x_{j\tilde{p}} - (x_{ip} + d_{ix}))^2 + (y_{j\tilde{p}} - (y_{ip} + d_{iy}))^2} \right) \quad (21)$$

where ε is the prediction error.

Thus, the actual position \tilde{p} in the second frame of the feature point p is given by:

$$\tilde{p} = p + d + \varepsilon \quad (22)$$

Then, the position \tilde{p} is taken as the measured position corresponding to displacement $(d + \varepsilon)$ of the point p between two successive images. The updated position is thereafter used for the next frame in the tracking algorithm (step (6) in figure 2).

IV. THE MEKF BASED ON VISUAL DATA FOR ATTITUDE ESTIMATION

In this section we present the structure of the continuous-time MEKF based on Visual Data fused with gyros measurements in order to estimate both attitude and gyro-bias. For this, we consider the kinematics equation of the rigid-body (3) with measurements given by (15), (16), and (17). MEKF design is based on the attitude estimation error according to $\tilde{R} = R\hat{R}^T$ where \hat{R} denotes the attitude estimate driven by:

$$\dot{\hat{R}} = \hat{R}S(\omega_m - \sigma) \quad (23)$$

where σ will be defined by the requirement that $\tilde{R} = I$ (identity matrix). Equivalently, the quaternion error is defined by $\tilde{Q} = (\tilde{q}_0, \tilde{q})$ such that $\tilde{R} = \mathcal{R}(\tilde{Q}) = \mathcal{R}(Q)\mathcal{R}(\hat{Q}^{-1})$ with $\tilde{R} = I \Leftrightarrow \tilde{Q} = (\pm 1, 0)$. Using quaternion error, the attitude estimation error dynamics is given using (23), (4) and (17) as:

$$\dot{\tilde{Q}} = \frac{1}{2} \begin{bmatrix} -\tilde{q}^T \\ \tilde{q}_0 I + S(\tilde{q}) \end{bmatrix} \tilde{\omega} \quad (24)$$

where $\tilde{\omega} = \hat{R}(\sigma - b_\omega - \nu_\omega)$. We can see that error \tilde{Q} depends only on gyros noise ν_ω , gyros bias b_ω which can be assumed to be small and the correction term σ . The error quaternion can be assumed to correspond to a small rotation and the following approximation is used $\tilde{Q} \approx (1, \tilde{q})$. Thus we can write:

$$\begin{aligned} \dot{\tilde{q}} &= \frac{1}{2} (I + S(\tilde{q})) \hat{R}(\sigma - b_\omega - \nu_\omega) \\ &= \frac{1}{2} \hat{R}(\sigma - b_\omega - \nu_\omega) - \frac{1}{2} \hat{R}S(\sigma - b_\omega - \nu_\omega) \hat{R}^T \tilde{q} \end{aligned}$$

then as the term $S(\nu_\omega) \hat{R}^T \tilde{q}$ is very small and can be ignored, we obtain:

$$\dot{\tilde{q}} = \frac{1}{2} \hat{R} \left((\sigma - b_\omega - \nu_\omega) - S(\sigma - b_\omega) \hat{R}^T \tilde{q} \right) \quad (25)$$

Let's define $x = \hat{R}^T \tilde{q}$, then it's time derivative is $\dot{x} = \dot{\hat{R}}^T \tilde{q} + \hat{R}^T \dot{\tilde{q}}$ and by using equations (23) and (25) we obtain:

$$\begin{aligned} \dot{x} &= -S(\omega_m - \sigma)x + \frac{1}{2}(\sigma - b_\omega - \nu_\omega) + \frac{1}{2}S(\sigma - b_\omega)x \\ &= -\frac{1}{2}S(2\omega_m - b_\omega - \sigma)x + \frac{1}{2}(\sigma - b_\omega - \nu_\omega) \end{aligned} \quad (26)$$

Now, using measurements n_{m_i} and k_{m_i} corresponding to the observation of N lines (d_i) in the 3D scene and indexed by i , we define the correction term σ as:

$$\sigma = \hat{b}_\omega - K_n(n_m - \hat{n}) \quad (27)$$

where $K_n \in \mathbb{R}^{3 \times 3N}$ is the gain matrix, $n_m^T = [n_{m_1}^T, \dots, n_{m_N}^T]$ is obtained from image measurements and the estimate $\hat{n}^T = [\hat{n}_1^T, \dots, \hat{n}_N^T]$ with $\hat{n}_i^T = k_{m_i} \times \hat{R}^T d_i$. The gyro bias estimation, \hat{b}_ω , is obtained from following equation:

$$\dot{\hat{b}_\omega} = K_\omega(n_m - \hat{n}) \quad (28)$$

where $K_\omega \in \mathbb{R}^{3 \times 3N}$ is the gain matrix to be determined.

Moreover, using equations (14), (15) and (16), we can develop measurement error $(n_m - \hat{n})$ as:

$$\begin{aligned} n_{m_i} - \hat{n}_i &= (k_i \times R^T d_i + \nu_{c_i}) - (k_{m_i} \times \hat{R}^T d_i) \\ &= ((k_{m_i} - \nu_{c_i}) \times R^T d_i + \nu_{c_i}) - (k_{m_i} \times \hat{R}^T d_i) \end{aligned}$$

by substituting cross product by skew-symmetric matrix $S(\cdot)$ and k_i by $k_{m_i} - \nu_{c_i}$ using (16), we obtain by introducing the attitude estimation error \tilde{R} :

$$n_{m_i} - \hat{n}_i = S(k_{m_i} - \nu_{c_i}) \hat{R}^T \tilde{R}^T d_i + \nu_{c_i} - S(k_{m_i}) \hat{R}^T d_i \quad (29)$$

then by using Rodriguez formula (1) with the approximation of \tilde{Q} , we have $\tilde{R} \approx I + 2S(\tilde{q})$ and thus equation (29) becomes:

$$n_{m_i} - \hat{n}_i \approx 2S(k_{m_i})S(\hat{R}^T d_i)x + (I + S(\hat{R}^T d_i))\nu_{c_i} \quad (30)$$

where we have substituted $\hat{R}^T \tilde{q}$ by x and neglected the term $S(\nu_{c_i})S(\hat{R}^T d_i)x$. Thus we can write measurement error $(n_m - \hat{n})$ as:

$$n_m - \hat{n} \approx H_n x + L_n \nu_c \quad (31)$$

with noise vector $\nu_c^T = [\nu_{c_1}^T, \dots, \nu_{c_N}^T]$ and matrices $H_n \in \mathbb{R}^{3N \times 3}$ and $L_n \in \mathbb{R}^{3N \times 3N}$ are:

$$H_n = \begin{bmatrix} 2S(k_{m_i})S(\hat{R}^T d_1) \\ \vdots \\ 2S(k_{m_i})S(\hat{R}^T d_N) \end{bmatrix}$$

$$L = \begin{bmatrix} (S(\hat{R}^T d_1) + I) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & (S(\hat{R}^T d_N) + I) \end{bmatrix}$$

Thus using equations (31) and (27), the error dynamics (26) becomes:

$$\dot{x} = \left(-S(\omega_m - \hat{b}_\omega) - \frac{1}{2}K_n H_n \right) x - \frac{1}{2}\tilde{b}_\omega - \frac{1}{2}K_n L_n \nu_c - \frac{1}{2}\nu_\omega \quad (32)$$

where terms $K_n(Hx + L\nu_c)x$ and $S(\tilde{b}_\omega)x$ have been neglected and $\tilde{b}_\omega = b_\omega - \hat{b}_\omega$ is the gyro bias error with dynamics obtained from equations (18) and (28) as:

$$\dot{\tilde{b}}_\omega = -K_\omega H_n x - K_\omega L_n \nu_c + \nu_b \quad (33)$$

Let's define the augmented vector $\chi^T = [x^T, \tilde{b}_\omega^T]$, then from the error dynamics (32) and (33) we can get the Kalman filter structure:

$$\dot{\chi} = (A - KH)\chi - KL_n \nu_c + M\nu_{\omega,b} \quad (34)$$

where matrices $A \in \mathbb{R}^{6 \times 6}$, $K \in \mathbb{R}^{6 \times 3N}$, $H \in \mathbb{R}^{3N \times 6}$ and $M \in \mathbb{R}^{6 \times 6}$ are:

$$A = \begin{bmatrix} -S(\omega_m - \hat{b}_\omega) & -\frac{1}{2}I \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}; \quad H = \begin{bmatrix} H_n & 0_{3N \times 3} \end{bmatrix}$$

$$M = \begin{bmatrix} -\frac{1}{2}I & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad K = \begin{bmatrix} \frac{1}{2}K_n \\ K_\omega \end{bmatrix}$$

The vector $\nu_{\omega,b}^T = [\nu_\omega^T, \nu_b^T]$ combine gyro noise assumed zero-mean Gaussian process with auto-covariance:

$$E \{ \nu_{\omega,b}(t) \nu_{\omega,b}^T(\tau) \} = Q_E \delta(t - \tau) \quad (35)$$

where $Q_E \in \mathbb{R}^{6 \times 6}$, is symmetric and positive definite, and $E \{ \cdot \}$ is the expectation operator. Also image measurement noise is assumed zero-mean Gaussian process with the corresponding auto-covariance:

$$E \{ \nu_c \nu_c^T \} = R_E \quad (36)$$

where $R_E \in \mathbb{R}^{3N \times 3N}$ is symmetric and positive-definite.

The error covariance of the error dynamics given in (34) is defined as :

$$P = E \{ \chi \chi^T \} \quad (37)$$

Then, we obtain the covariance propagation:

$$\dot{P} = [A - KH]P + P[A - KH]^T + MQ_E M^T + KL_n R_E L_n^T K^T \quad (38)$$

With assumption that $\tilde{\chi}$, $\nu_{\omega,b}$, ν_c are uncorrelated, the gain K is therefore selected to minimize the trace of \dot{P} , which yields[6] :

$$K = PH^T (L_n R_E L_n^T)^{-1} \quad (39)$$

Finally the state equations (23) and (28) and the covariance propagation of equation (38) with gain (39) are needed for MEKF to estimate both attitude and gyro bias.

V. RESULTS

In this section, we present the results of simulations using real images to demonstrate the effectiveness of the algorithm for detecting and tracking lines. Thereafter, we use sequences of synthetic images to test our method attitude estimation of a rigid body in a combined motion of rotation and translation. Figure 3-(a) presents a real scene and shows the detected lines, a set of points of these lines are extracted as described in the algorithm (2).

Figure 3 (b) shows the selected two lines and in (c) we can see the extracted point with a very low threshold $\tau = 0,01$. This set of points is used to match the tracked lines after motion. Thus, we can see that the matched lines in (d)

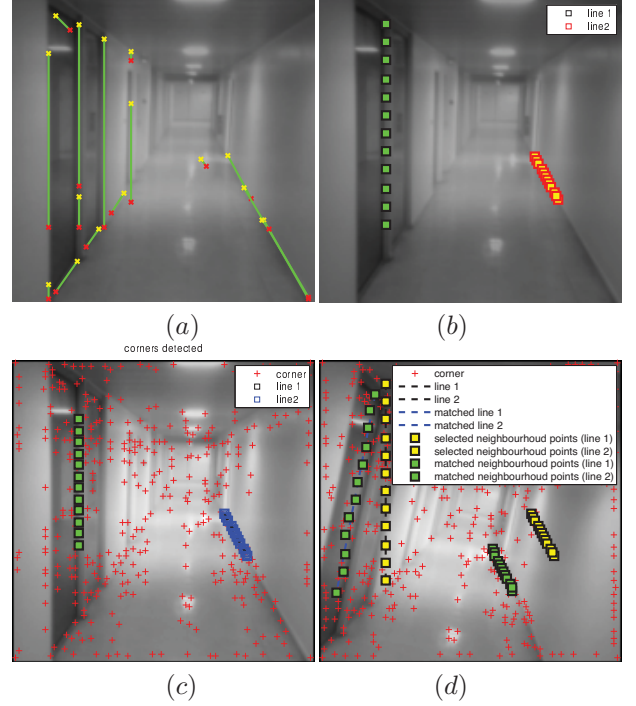


Figure 3. Detection and tracking of lines inside a building: (a) the detection of lines in a corridor of the laboratory. (b) Selection of two lines and a few points of each line. (c) Detecting points according to the selected gradient. (d) Detection of lines and points after rotation about the axis normal to image plane.

correspond to the real lines detected in the first image. This result shows the feasibility of the approach of tracking lines based on optical flow matching in building. In the case of an application outside of buildings in urban areas the problem does not occur, the lines exist and the process of detection of the lines is quite efficient. In contrast, in the case of deserts and forest and agricultural areas, which are highly textured, the solution is related to the use of the detection of specific points with high gradient and recovery lines from points or directly exploiting short segments and edgelet as exposed in the method of the unified point-edgelet tracking presented in [26].

In the second part of our test, the synthetic images generated from simulation of translational and rotational motion of the camera in 3D-space are used to detect environmental lines whose directions are known. For attitude estimation of the camera motion, we use the procedure of algorithm described in section (III) and we get the unit measurement vector n_m (15) and k_m (16), to be used in MEKF. The gyros measurement are modeled by angular velocity of the camera with a bias $b = [0.2, .01, -0.2]^T rad/s$ and Gaussian white noise $w = 2.5 \times 10^{-4} rad/s$. Then using the covariance propagation of equation (38) and the Kalman gain (39) with state equations (23) and (28). Through the measurement from the image and gyros, we have obtained the estimation of the attitude which is shown by roll, pitch and yaw angles in figure (4). For this test, we initially set an error of attitude of 10° on each axis (roll, pitch, and yaw). The obtained results show a good estimate of the attitude as illustrated by the attitude errors given in figure (5). We have also obtained a good estimation of gyros bias

as shown in figure 6.

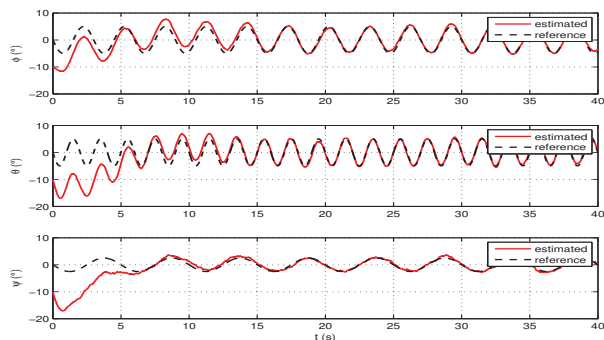


Figure 4. Attitude estimation

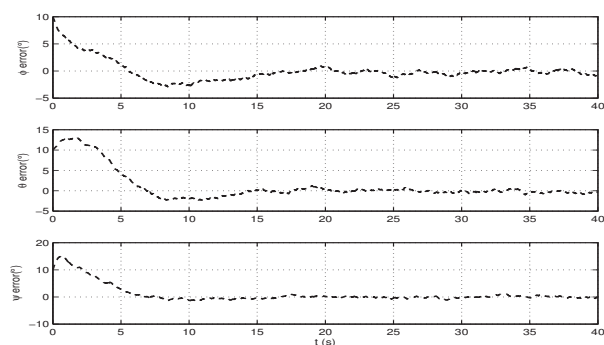


Figure 5. Attitude errors

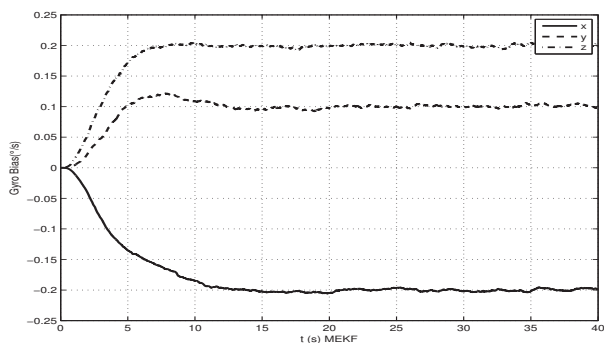


Figure 6. Gyro Bias estimation

VI. CONCLUSION

In this paper, a feature-tracking algorithm has been presented with application on estimation of the attitude of a rigid body equipped with a camera. The proposed algorithm is able to track image features over a long sequence by cancelling the drift that occurs when dealing with long sequences of images with classical algorithm. The main difference with other methods ([11], [18]) is that we update the predicted displacement between two successive images by measuring the prediction error using the criterion of the Euclidean distance. This algorithm is used to detect and to track lines in the 3D scene in order to estimate rotational motion of a rigid body by fusing obtained measurements vector with gyros measurement based on MEKF. The performed tests and the obtained results show the effectiveness of the proposed algorithm to improve feature-tracking and allow good estimation of rotational motion.

REFERENCES

- [1] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. In *International Conference on Robotics and Automation*, volume 3, pages 2070–2075, 2000.
- [2] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, pages 236–242, jun 1992.
- [3] S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Survey*, 17:433–467, 1995.
- [4] M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 231–236, 1993.
- [5] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5, 2001.
- [6] John L Crassidis and John L Junkins. *Optimal estimation of dynamic systems*. CRC press, 2011.
- [7] Rachid Deriche and Olivier Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.
- [8] Olivier Faugeras. *On the motion of 3D curves and its relationship to optical flow*. Springer, 1990.
- [9] Dario Floreano, Jean-Christophe Zufferey, Mandyam V. Srinivasan, and Charlie Elington. Flying insects and robots. In *Fumiya Iida Artificial Life*, volume 18, 2011.
- [10] N. Franceschini, F. Ruffier, and J. Serres. A bio-inspired flying robot sheds light on insect piloting abilities. In *Curr Biol*, volume 7, pages 329–335, 2007.
- [11] Bing Han, William Roberts, Dapeng Wu, and Jian Li. Robust feature-based object tracking. In *Proc. of SPIE Vol.*, volume 6568, pages 65680U–1, 2007.
- [12] C.G. Harris and M. Stephens. A combined corner and edge detector. *4th Alvey Vision Conference*, pages 147–151, 1988.
- [13] Bruno. Herisse, Sophie. Oustrieres, Tarek. Hamel, Robert. Mahonny, and Francois-Xavier. Russoto. A general optical flow based terrain-following strategy for vtol uav using multiple views. In *ICRA 2010*, 2010.
- [14] B.K.P. Horn and B.G. Shunck. Determining optical flow, artificial intelligence. *Artificial intelligence*, 17:185–203, 1981.
- [15] T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: a review. *Proceedings of the IEEE*, 82(2):252–268, 1994.
- [16] J. Illingworth and j Kittler. A survey of the hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.
- [17] M. Jankovic and Bijoy K. Ghosh. Visually guided ranging from observations of points, lines and curves via an identifier based nonlinear observer. In *System and Control Letters*, volume 25, pages 63–73, 1995.
- [18] Farid Kendoul, Isabelle Fantoni, and Kenzo Nonami. Optic flow-based vision system for autonomous 3d localization and control of small aerial vehicles. *Robotics and Autonomous Systems*, 57(6):591–602, 2009.
- [19] C. Kessler, C. Ascher, N. Frietsch, M. Weinmann, and G.F. Trommer. Vision-based attitude estimation for indoor navigation using vanishing points and lines. In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pages 310–318, 2010.
- [20] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *In International Joint Conference of Artificial Intelligence.*, 1981.
- [21] A. Makadia and K. Daniilidis. Correspondenceless ego-motion estimation using an imu. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3534–3539, 2005.
- [22] N. Navab and Z. Zhang. Fusion of visual data through dynamic stereo-motion cooperation. In *Proceedings of XVII Congress of the International Society for Photogrammetry and Remote Sensing ISPRS*, August 1992.
- [23] Jean-Louis Palomares. *Une nouvelle méthode d'appariement de points d'intérêt pour la mise en correspondance d'images*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2012.
- [24] Henrik. Rehbinder and Bijoy K. Ghosh. Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48:186–199, 2003.
- [25] M. D. SHUSTER and S. D. OH. Three-axis attitude determination from vector observations. *Journal of Guidance and Control*, 4:70–77, 1981.
- [26] Kalaivani SUNDARARAJAN. *Unified point-edgelet feature tracking*. PhD thesis, Clemson University, 2011.
- [27] Giorgio Toscani, Rachid Deriche, and Olivier D Faugeras. 3d motion estimation using a token tracker. In *MVA*, pages 257–261, 1988.

Gradient Based Multi-modal Sensor Calibration

Zachary Taylor and Juan Nieto
University of Sydney, Australia
{z.taylor, j.nieto}@acfr.usyd.edu.au

Abstract—This paper presents an evaluation of a new metric for registering two sensors of different modality. The metric operates by aligning gradients present in the two sensors’ outputs. This metric is used to find the parameters between the sensors that minimizes the misalignment of the gradients. The metric can be applied to a wide range of problems and has been successfully demonstrated on the extrinsic calibration of two different lidar-camera systems as well as the alignment of IR and RGB images. Unlike many previous techniques, this method requires no markers to be placed in the scene and can operate on a single scan from each sensor.

I. INTRODUCTION

Most mobile robotic platforms rely on a large range of different sensors to navigate and understand their environment. However before multiple sensors can work together to give information on the same target the sensor outputs must be registered. This registration is far from trivial due to the very different modalities via which different sensors may operate. This registration has traditionally been performed by either hand labelling points or placing markers such as corner reflectors or chequerboards in the scene. The location of these markers are detected by all of the sensors and their positions are used for calibration.

The calibration produced by hand-labelling or marker-based methods, while initially accurate, is quickly degraded due to the robot’s motion. For mobile robots working on topologically variable environments, such as agricultural or mining robots, the motion can result in significantly degraded calibration after as little as a few hours of operation. Under these conditions marker based calibration quickly becomes tedious and impractical. To maintain an accurate calibration, an automated system that can recalibrate the sensors using observations made during the robot’s normal operations is required. We envision a system that would periodically retrieve a set of scans from the sensors and then, while the robot continues its tasks, process it to validate the current calibration and update the parameters when needed.

Towards that aim, we have developed a new metric, the *gradient orientation measure* (GOM) that can effectively align the outputs of two sensors of different modalities. The metric can calibrate multi-sensor platforms by optimising through a set of observations, and, unlike most current calibration approaches, the metric is also able to calibrate from a single scan pair. This last

property makes our approach suitable for a broad range of applications since it is not restricted to calibration based on multiple observations from sensors attached to a rigid mount. To demonstrate the metric’s potential and versatility we present results on three different datasets: (i) the alignment of two hyper-spectral camera images, (ii) the calibration of a rotating panoramic camera with a single high resolution scan and (iii) the calibration of a panospheric camera with a series of Velodyne scans. In each of these tests the proposed approach is compared with state of the art methods. An example of the results obtained with our system is shown in Figure 1.

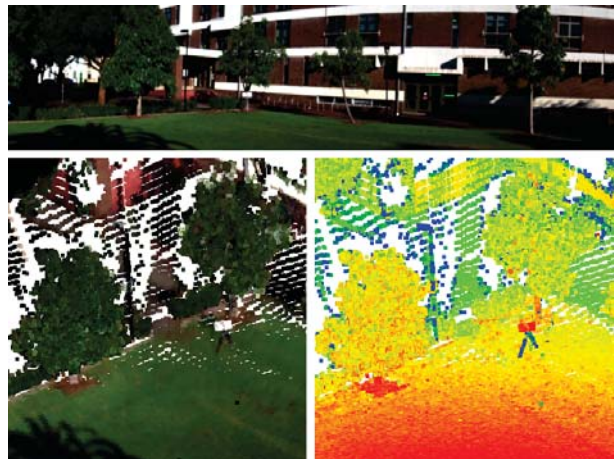


Fig. 1. Camera and lidar scan being combined. Raw lidar data is shown in the bottom right, with the camera image shown on top. The textured map obtained with our approach is shown at the bottom left.

II. RELATED WORK

The most common techniques in multimodal registration are *mutual information* (MI) and *normalized mutual information* (NMI). Both measures use Shannon entropy to give an indication of how much one sensor output depends on the other. They have both been widely used in medical image registration, a survey of MI-based techniques has been presented in [1].

A. Mastin et al. achieved registration of an aerial lidar scan by creating an image from it using a camera model [2]. The intensity of the pixels in the image generated from the lidar scan was either the intensity of the laser return or the height from the ground. The images were compared using the joint entropy of the images and

optimisation was done via downhill simplex. The method was only tested in an urban environment where buildings provided a strong relationship between height and image colour.

One of the first approaches used to successfully register Velodyne scans with camera images that did not rely on markers was presented in [3]. Their method operates on the principle that depth discontinuities detected by the lidar will tend to lie on edges in the image. Depth discontinuities are isolated by measuring the difference between successive lidar points and removing points with a depth change of less than 30 cm. An edge image is produced from the camera that is then blurred to increase the capture region of the optimiser. The average of all of the edge images is then subtracted from each individual edge image to remove any bias to a region. The two outputs are combined by projecting the isolated lidar points onto the edge image and multiplying the magnitude of each depth discontinuity by the intensity of the edge image at that point. The sum of the result is taken and a grid search used to find the parameters that maximise the resulting metric.

Two very similar methods that also operate on Velodyne-camera systems have been independently developed by Pandey et al. [4] and Wang et al. [5]. These methods use the known intrinsic values of the camera and estimated extrinsic parameters to project the lidar’s scan onto the camera’s image. The MI value is then taken between the lidar’s intensity of return and the intensity of the corresponding points in the camera’s image. When the MI value is maximised, the system is assumed to be perfectly calibrated. The only major difference between these two approaches is in the method of optimisation used; Pandey et al. makes use of the Barzilai-Borwein (BB) steepest gradient ascent algorithm, while R. Wang et al. makes use of the Nelder-Mead downhill simplex method. In both implementations, aggregation of a large set of scans is required for the optimisers used to converge to the global maximum.

III. METHOD

Our method can be divided into two main stages: feature computation and optimisation.

The feature computation stage converts the sensor data into a form that facilitates comparisons of different alignments during the optimisation stage. The initial step is to perform histogram equalisation on the input intensities to ensure high contrast in the data. Next, an edge detector is applied to the data to estimate the intensity and orientation of edges at each point; the edge detector used depends on the dimensionality of the data. The strength of the edges is histogram equalised to ensure that a significant number of strong edges are present. This edge information is finally passed into the optimisation, completing the feature computation step.

The sensors’ outputs are aligned during the optimisation. This is done by defining one sensor’s output as

fixed (called the base sensor output) and transforming the other sensor’s output (referred to as the relative sensor output). In our framework, the base output is always 2D. For two 2D images, an affine transform is used, and for 2D-3D alignment, a camera transform is used to project the 3D points of the relative output onto the 2D base output. Once this has been done, the base output is interpolated at the locations that the relative output was projected onto to give the edge magnitudes and directions at these points.

Finally, GOM is used to compare the edge features between the two outputs and to provide a measure of the quality of the alignment. This process is repeated for different transformations until the optimal set of parameters is found.¹

A. Transformation

The transformation applied to align the sensors’ outputs depends on the dimensionality of the two sensors. If one sensor outputs 3D data, for example a lidar, and the other sensor is a camera, then a camera model is used to transform the 3D output. If both sensors provide a dense 2D image, then an affine transform is used to align them. A more detailed look at calculating the transforms is covered in [6].

B. Gradient calculation

The magnitude and orientation of the gradient of a camera’s image intensity is calculated using the Sobel operator. Calculation of the gradient from 3D data sources is slightly more challenging and performed using the method outlined in [6].

C. The Gradient orientation measure

The formation of a measure of alignment between two multi-modal sources is a challenging problem. Strong features in one source can be weak or missing in the other. A reasonable assumption when comparing two multi-modal images is that, if there is a significant change in intensity between two points in one image, then there is a high probability there will be a large change in intensity in the other modality. This correlation exists as these large changes in intensity usually occur due to a difference in the material or objects being detected.

GOM exploits these differences to give a measure of the alignment. GOM operates by calculating how well the orientation of the gradients are aligned between two images. For each pixel, it gives a measure of how aligned the points are by taking the absolute value of the dot product of the gradient vectors:

$$alignment_j = |g_{(1,j)} \cdot g_{(2,j)}| \quad (1)$$

where $g_{(i,j)}$ is the gradient in image i at point j . The absolute value is taken, as a change going from low to

¹All the code used for our method as well as additional results and documentation is publicly available online at <http://www.zacharyjeremytaylor.com>

high intensity in one modality may be detected as going from high to low intensity in the other modality.

Summing the value of these points results in a measure that is dependent on the alignment of the gradients. An issue, however, is that this measure will favour maximising the strength of the gradients present in the overlapping regions of the sensor fields. To correct for this bias, the measure is normalised after the sum of the alignments has been made, by dividing by the sum of all of the gradient magnitudes. This gives the final measure as shown in Equation 2.

$$GOM = \frac{\sum_{j=1}^n |g_{(1,j)} \cdot g_{(2,j)}|}{\sum_{j=1}^n \|g_{(1,j)}\| \|g_{(2,j)}\|} \quad (2)$$

The measure has a range from 0 to 1, where, if 0, every gradient in one image is perpendicular to that in the other, and 1 if every gradient is perfectly aligned. Some typical GOM values for a range of images is shown in Figure 2. The NMI values are also shown for comparison.

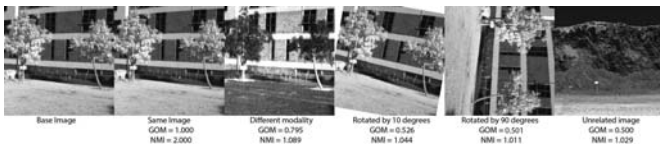


Fig. 2. GOM and NMI values when the base image shown on the left is compared with a range of other images.

D. Optimisation

The registration of one-off scans, and the calibration of a multi-sensor system tend to have significantly different constraints on their optimisation. Because of this, our approach for optimising each problem differs.

For cases where multiple scans can be aggregated, the optimisation is performed using the Nelder-Mead simplex method [7] in combination with a Gaussian pyramid. In our experiments, four layers were used in the pyramid, with Gaussians with σ of 4, 2, 1 and 0 applied.

When optimization from a single scan is required and/or there is significant error in the initial guess for the calibration, the search space becomes highly non-convex and a local optimization method such as Nelder-Mead cannot reliably find the global minimum. In these situations the metric is optimized using Particle swarm. Particle swarm optimisation works by randomly placing an initial population of particles in the search space. On each iteration a particle moves to a new location chosen by summing three factors: i) it moves towards the best location found by any particle, ii) it moves towards the best location it has ever found itself and iii) it moves in a random direction. The optimiser stops once all particles have converged. The implementation of particle swarm used was developed by S Chen [8]. In our experiments we used a particle swarm optimiser with 500 particles.

IV. EXPERIMENTAL RESULTS

A. Metrics Evaluated

In this section, a series of metrics are evaluated on three different datasets. The metrics evaluated are as follows:

- MI - mutual information, the metric used by Pandey et al. [4] in their experiments on the Ford dataset [9].
- NMI - normalised mutual information, a metric we had used in our previous work on multi-modal calibration [10].
- The Levinson method [3].
- GOM - the gradient orientation measure developed in this paper.
- SIFT - scale invariant feature transform, a mono-modal registration technique included to highlight some of the challenges of multi-modal registration and calibration.

B. Parameter Optimisation

To initialise the optimisation we use either the ground truth (when available) or a manually calibrated solution. We then added a random offset to it. The random offset is uniformly distributed, with the maximum value used given in the details of each experiment. This random offset is introduced to ensure that the results obtained from multiple runs of the optimisation are a fair representation of the method's ability to converge to a solution reliably. When particle swarm optimisation is used, the search space of the optimiser is set to be twice the size of the maximum offset.

On datasets where no ground truth was available the search space was always constructed so that the space was much greater than twice the estimated error of the manual calibration to ensure that it would always be possible for a run to converge to the correct solution. All experiments were run 10 times with the mean and standard deviation from these runs reported for each dataset.

C. Dataset I

A Specim hyper-spectral camera and Riegl VZ1000 lidar scanner were mounted on top of a Toyota Hilux and used to take a series of four scans of our building, the Australian Centre for Field Robotics from the grass courtyard next to it. The focal length of the hyper-spectral camera was adjusted between each scan. This was done due to the different lighting conditions and to simulate the actual data collection process in the field.

This dataset required the estimation of an intrinsic parameter of the camera, its focal length in addition to its extrinsic calibration. To test the robustness and convergence of the methods, each scan was first roughly manually aligned. The search space was then constructed assuming the roll, pitch and yaw of the camera were each within 5 degrees of the lasers. The camera's principal distance was within 40 pixels of correct (for this camera

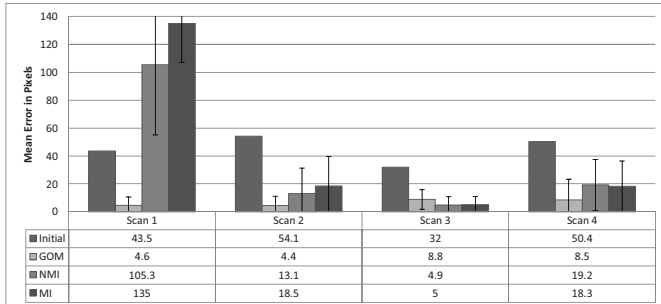


TABLE I
ACCURACY COMPARISON OF DIFFERENT METHODS ON ACFR
DATASET. ALL DISTANCES IN PIXELS

principal distance ≈ 780) and the X, Y and Z coordinates were within 1 metre of correct.

1) *Results:* No accurate ground truth is available for this dataset. To overcome this issue and allow an evaluation of the accuracy of the method, 20 points in each scan-image pair were matched by hand. An evaluation of the accuracy of the method was made by measuring the distance in pixels between these points on the generated images. The results are shown in Table I.

For this dataset GOM significantly improved upon the initial guess for all four of the tested scans. Scans 1 and 2 were however more accurately registered than scans 3 and 4. These last two scans were taken near sunset, and the long shadows and poorer light may have played a part in the reduced accuracy of the registration. NMI gave mixed results on this dataset, misregistering scan 1 by a large margin and giving results far worse than GOM's for scans 2 and 4. It did however outperform all other methods on Scan 3. MI gave a slightly worse, but similar, performance. Levinson's method could not be evaluated on this dataset as it requires multiple images to operate.

D. Dataset II

To test each method's ability to register different modality camera images such as IR-RGB camera alignment, two scenes were scanned with a hyper-spectral camera. Bands near the upper and lower limits of the camera's spectral-sensitivity were selected so that the modality of the images compared would be as different as possible, providing a challenging dataset on which to perform the alignment. The bands selected were at 420 nm (violet light) and 950 nm (near IR). The camera was used to take a series of three images of the ACFR building and three images of cliffs at a mine site. An example of the images taken is shown in Figure 3.

The search space for the particle swarm optimiser was setup assuming the X and Y translation were within 20 pixels of the actual image, the rotation was within 10 degrees of the actual image, the X and Y scale were within 10 % of the actual image and the x and y shear were within 10 % of the actual image.



Fig. 3. Images captured by hyper-spectral camera. The top image was taken at 420nm and the bottom at 950nm

1) *Results:* In addition to the GOM, MI and NMI methods that have been applied to all of the datasets, SIFT features were also used. SIFT was used in combination with RANSAC to give the final transform. To measure how accurate the registration was, the average difference in position between each pixel's transformed position and its correct location was obtained. The results of this registration are shown in Table II. The images taken at the ACFR were 320 by 2010 pixels in size. The width of the images taken at the mine varied slightly, but were generally around 320 by 2500 pixels in size.

SIFT performed rather poorly on the ACFR dataset and reasonably on the mining dataset. The reason for this difference was most likely due to the very different appearance vegetation has at each of the frequencies tested. This difference in appearance breaks the assumption SIFT makes of only linear intensity changes between images, and therefore the grass and trees at the ACFR generate large numbers of incorrect SIFT matches. In the mine sites that are devoid of vegetation, most of the scene appears very similar, allowing the SIFT method to operate and give more accurate results.

Looking at the mean values for each run MI, NMI and GOM gave similar performance on these datasets, all achieving sub-pixel accuracy in all cases. There was little variation in the results obtained using the multi-modal metrics, with all three methods always giving errors between 0.2 and 0.8 pixels.

E. Dataset III

The Ford campus vision and lidar dataset has been published by G. Pandey et al. [9]. The test rig was a Ford F-250 pick-up truck which had a Ladybug panospheric camera and Velodyne lidar mounted on top. The dataset contains scans obtained by driving around downtown Dearborn, Michigan USA. An example of the data is shown in Figure 4. The methods were tested on a subset of 20 scans. These scans were chosen as they were the same scans used in the results presented by Pandey et al. Similarly, the initial parameters used were those provided with the dataset. As all of the scan-image pairs on this dataset shared the same calibration parameters, aggregation of the scans could be used to improve the accuracy of the metrics. Because of this, each experiment was performed three times, aggregating 2, 5 and 10 scans.

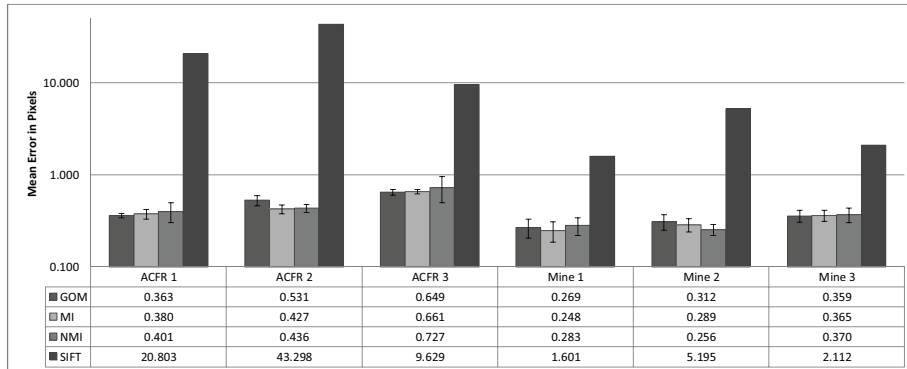


TABLE II

ERROR AND STANDARD DEVIATION OF DIFFERENT REGISTRATION METHODS PERFORMED ON HYPERSPECTRAL IMAGES. ERROR IS GIVEN AS THE MEAN PER-PIXEL-ERROR IN POSITION. NOTE THAT THE CHART'S AXIS USES A LOG SCALE.



Fig. 4. Overlapping region of camera image (top) and lidar scan (bottom) for a typical scan-image pair in the Ford Dataset. The lidar image is coloured by intensity of laser return

1) *Results:* The Ford dataset does not have a ground truth. However, a measure of the calibration accuracy can still be obtained through the use of the Ladybug camera. The Ladybug consists of five different cameras all pointing in different directions (excluding the camera pointed directly upwards). The extrinsic location and orientation of each of these cameras is known very accurately with respect to one another. This means that if the calibration is performed for each camera independently, the error in their relative location and orientation will give a strong indication as to the method's accuracy.



Fig. 5. Camera and velodyne scan being registered. Left, the velodyne scan. Centre, the Ladybug's centre camera image. Right the two sensor outputs overlaid.

All five cameras of the Ladybug were calibrated independently. An example of the process of registering one of the camera's outputs is shown in Figure 5. This calibration was performed 10 times for each camera using randomly selected scans each time. The error in each camera's relative position to each other camera in all trials was found and the average error shown in Table III.

In these tests GOM, NMI and MI gave similar re-

sults. GOM tended to give the most accurate rotation estimates while MI gave the most accurate position estimates. For all three of these metrics, scan aggregation slightly improved the accuracy of angles and position. Levinson's presented the largest improvement in accuracy when more scans were aggregated, resulting in the largest error with 2 and 5 scans and giving similar results to the other methods with 10 scans.

In this experiment, any strong conclusion about which metric performed the best is difficult to draw as the difference between any two metrics for 10 aggregated scans is significantly less than the variance in their values. In almost all of the tests, the estimate of the cameras Z position was significantly worse than the X and Y estimates. This was expected as the metric can only be evaluated in the overlapping regions of the sensors fields of view. The Velodyne used has an extremely limited vertical resolution (64 points, one for each laser). Thus making the parallax error that indicates an error in the Z position difficult to observe. The narrow beam width of the Velodyne is also why the yaw shows the lowest error, as there are more overlapping points that can be used to evaluate this movement.

The actual error of a Ladybug-Velodyne system calibrated using all five cameras simultaneously would give a far more accurate solution than the results obtained here. There are several reasons for this. Individually the single camera systems have a narrow field of view. Therefore, a forward or backward translation of the camera is only shown through subtle parallax error in the position of objects in the scene. This issue is significantly reduced in the full system due to the cameras that give a perpendicular view that clearly shows this movement. In the single camera problem, movement parallel to the scene is difficult to distinguish from a rotation. This is also solved by the full system due to the very different effect a rotation and translation have on cameras facing in significantly different directions. Finally the full system also benefits from the increase in the amount of overlap

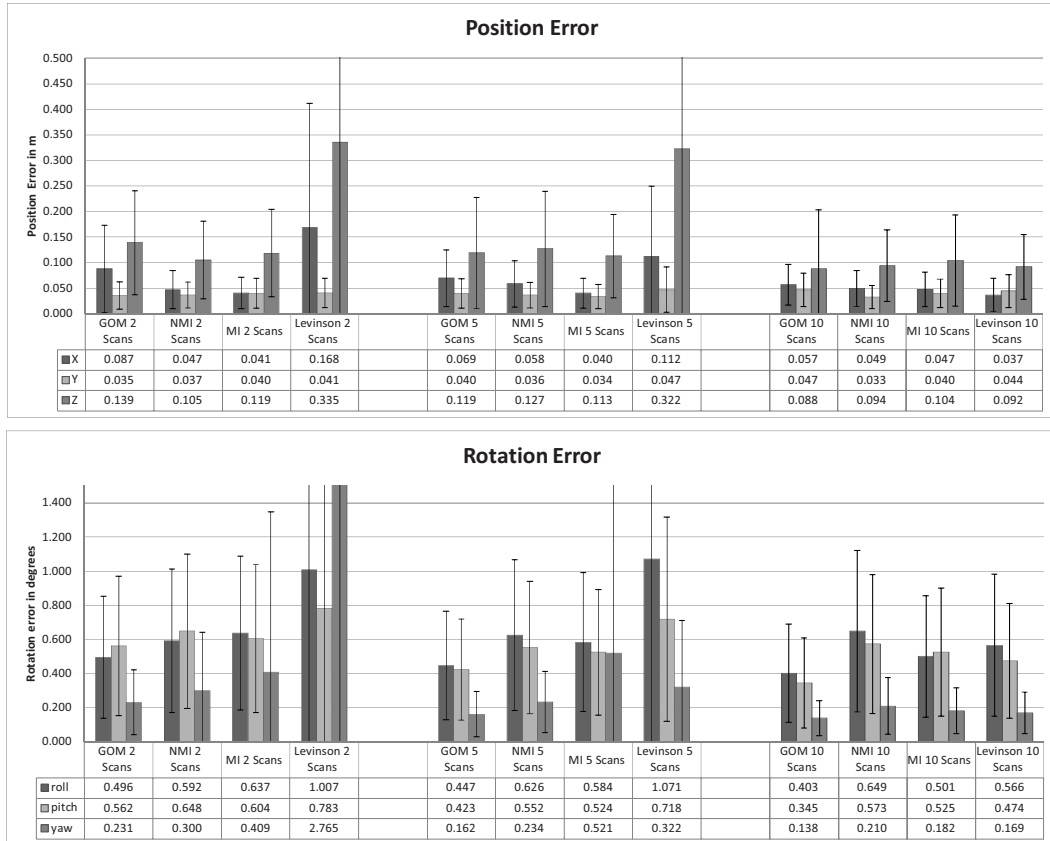


TABLE III

AVERAGE ERROR BETWEEN TWO ALIGNED LADYBUG CAMERAS. ALL DISTANCES ARE IN METRES AND ANGLES ARE IN DEGREES.

between the sensors' fields of view.

V. CONCLUSION

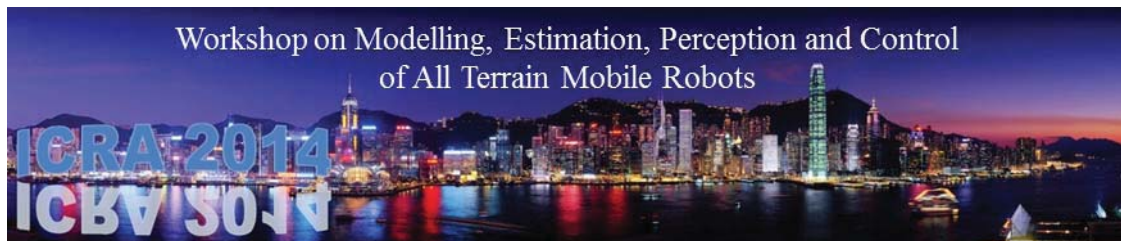
We have presented a detailed evaluation of our *gradient orientation measure* (GOM). The measure can be used to align the output of two multi-modal sensors, and has been demonstrated on a variety of datasets and sensors. Three other existing methods were also implemented and their accuracy tested on the same datasets. On the datasets tested GOM successfully registered all datasets to a high degree of accuracy, showing the robustness of the method, for a large range of environments and sensor configurations. We also examined the level of accuracy required for an initial guess for a system's calibration to be optimised to the correct solution.

ACKNOWLEDGMENT

This work has been supported by the Rio Tinto Centre for Mine Automation and the Australian Centre for Field Robotics, University of Sydney.

REFERENCES

- [1] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual-information-based registration of medical images: a survey," *Medical Imaging, IEEE*, vol. 22, no. 8, pp. 986–1004, 2003.
- [2] A. Mastin, J. Kepner, and J. Fisher III, "Automatic registration of LIDAR and optical images of urban scenes," *Computer Vision and Pattern Recognition*, pp. 2639–2646, 2009.
- [3] J. Levinson and S. Thrun, "Automatic Calibration of Cameras and Lasers in Arbitrary Scenes," in *International Symposium on Experimental Robotics*, 2012.
- [4] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic Targetless Extrinsic Calibration of a 3D Lidar and Camera by Maximizing Mutual Information," *Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 2053–2059, 2012.
- [5] R. Wang, F. P. Ferrie, and J. Macfarlane, "Automatic registration of mobile LiDAR and spherical panoramas," *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 33–40, Jun. 2012.
- [6] Z. Taylor, J. Nieto, and D. Johnson, "Automatic calibration of multi-modal sensor systems using a gradient orientation measure," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1293–1300, 2013.
- [7] J. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, 1965.
- [8] S. Chen, "Another Particle Swarm Toolbox," 2009.
- [9] G. Pandey, J. McBride, and R. Eustice, "Ford campus vision and lidar data set," in *The International Journal of Robotics Research*, 2011, pp. 1543–1552.
- [10] Z. Taylor and J. Nieto, "A Mutual Information Approach to Automatic Calibration of Camera and Lidar in Natural Environments," in *the Australian Conference on Robotics and Automation (ACRA)*, 2012, pp. 3–5.



Session II (2/2)

Perception in outdoor environment

- **Invited Talk: Alberto Elfes (CSIRO, Australia)**

Title: Multi-Modal Robot Perception and Augmented World Models

- **Title: Localization of Planetary Exploration Rovers with Orbital Imaging:
a survey of approaches**

Author: E. Boukas¹, A. Gasteratos¹ and G. Visenti

- **Title: A probabilistic distribution approach for the classification of urban
roads in complex environments**

Author: G. B. Vitor, A. C. Victorino and J. V. Ferreira

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Session II (1/2)

Invited Talk: **Alberto Elfes**
(Autonomous Systems Lab, CSIRO)

Multi-Modal Robot Perception and Augmented World Models

Abstract: There are tight interrelations between the mobility capabilities of a robotic platform, the exteroceptive sensors it uses, the world models it builds from these sensors, and the tasks that the platform is designed to execute. To date, the primary sources used by mobile robots to learn about their environment have been range sensors. These are used to build static 2D, 2.5D or 3D maps or dynamic 4D world models, which in turn are used for navigation or manipulation. However, as robots are deployed in increasingly more demanding missions in complex natural or artificial environments, they require richer world models with information beyond the spatial structure of their surroundings. This talk will discuss our work in using multiple sensor modalities, including lidar, RGB, thermal, hyperspectral and haptics, to build multi-property augmented world models (AWMs) for a variety of robot platforms and missions. These AWMs range from simple superposition of spatiotemporally registered sensor streams to multi-modal sensor fusion and to active models where inferences in one sensor stream are used to analyze and make inferences in other sensor streams. Results will be shown from domains that include exploration and documentation of natural and cultural sites, as well as in situ sensing for agricultural, environmental and industrial applications.

Biography: Alberto Elfes has a B.Eng. degree in Electronics Engineering and a M.Sc. in Computer Science from the Aeronautics Institute of Technology (ITA), Brazil, and a Ph.D. degree in Electrical and Computer Engineering from Carnegie-Mellon University, USA. He has held research positions at the Robotics Institute and the Engineering Design Research Center, CMU, and the Department of Computer Sciences, IBM T. J. Watson Research Center. He was director of the Automation Institute, Brazil, and visiting professor at the University of Ulm and the FAW research institute, Germany. From 2001 to 2011, Dr. Elfes was a Principal Researcher at NASA's Jet Propulsion Laboratory (JPL), and since August/2011 he has been a Senior Principal Research Scientist and Robotics Science Leader at the Autonomous Systems Laboratory (ASL) of CSIRO, Australia's national research organization. He is the author of the Occupancy Grid framework, a random lattice-based approach for autonomous robot perception, mapping, navigation and control that has been widely adopted by the robotics community. He has led a number of pioneering projects, including autonomous airships for planetary exploration; distributed, supervisory and cooperative architectures for perception, planning and control of multi-robot teams; and autonomous robot ocean vehicles for Earth science missions. His current research interests include multi-modal perception, exploration robots, and robot co-workers. He is recipient of the Mercator Professorship Award of the German Research Foundation (DFG), and of several NASA awards. Dr. Elfes has over 150 publications in international journals, conferences and books, and has lectured extensively in North America, Europe, Brazil, Japan and Australia.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



Multi-Modal Perception

Alberto Elfes

Autonomous Systems Laboratory | CSIRO

ICRA14 Workshop on Modelling, Estimation, Perception
and Control of All Terrain Mobile Robots
Hong-Kong, 01 June 2014

AUTONOMOUS SYSTEMS LABORATORY | CCI
www.csiro.au



About CSIRO

People 6500

Divisions 11

Locations 58

Flagships 11

Budget \$1B+



Top 1% of global research institutions in 14 of 22 research fields

Top 0.1% in 4 research fields

62% of our people hold university degrees

2000 doctorates

500 masters

With our university partners, we develop 650 postgraduate research students

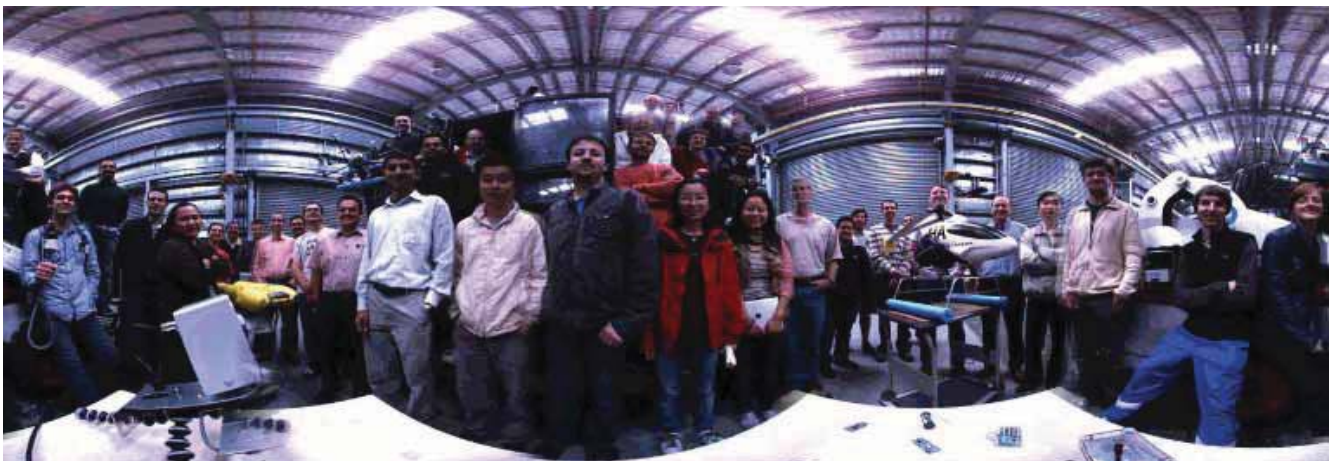


CSIRO Autonomous Systems Program

Vision: A world in which humans and autonomous systems are able to seamlessly, reliably and safely collaborate.



CSIRO Autonomous Systems Program



110+ Research Staff

50+ Students and Visitors

Major Test Facilities and Infrastructure

5 Sites in 3 Cities: Brisbane, Hobart, Sydney

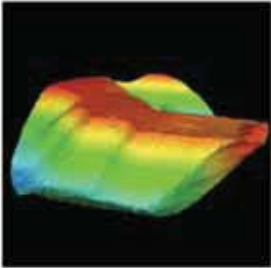
Autonomous Systems Program



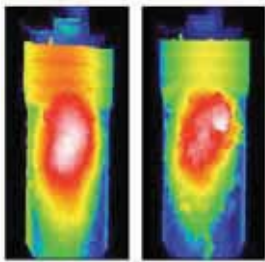
- Robotics
- Sensor Networks and Systems
- Cognitive Computing
- Computational Imaging

LIDAR: Primary Sensor for Robotics

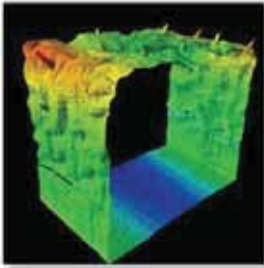
Dragline Terrain Mapping



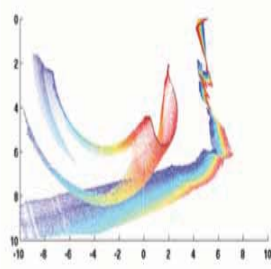
Haul Truck Scanning



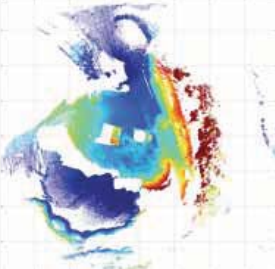
Explosive Loading



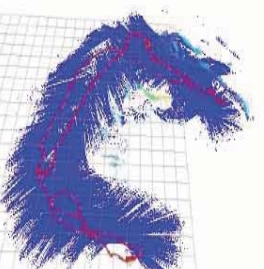
Excavator Bucket Tracking



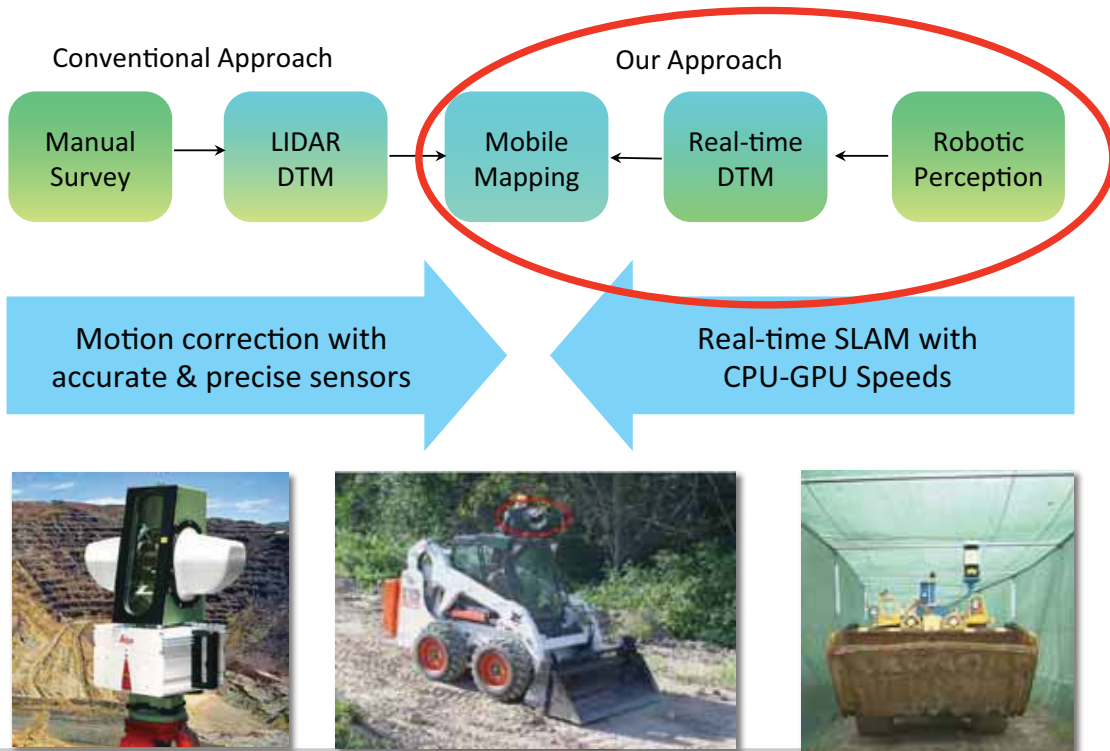
Truck and Shovel Mapping



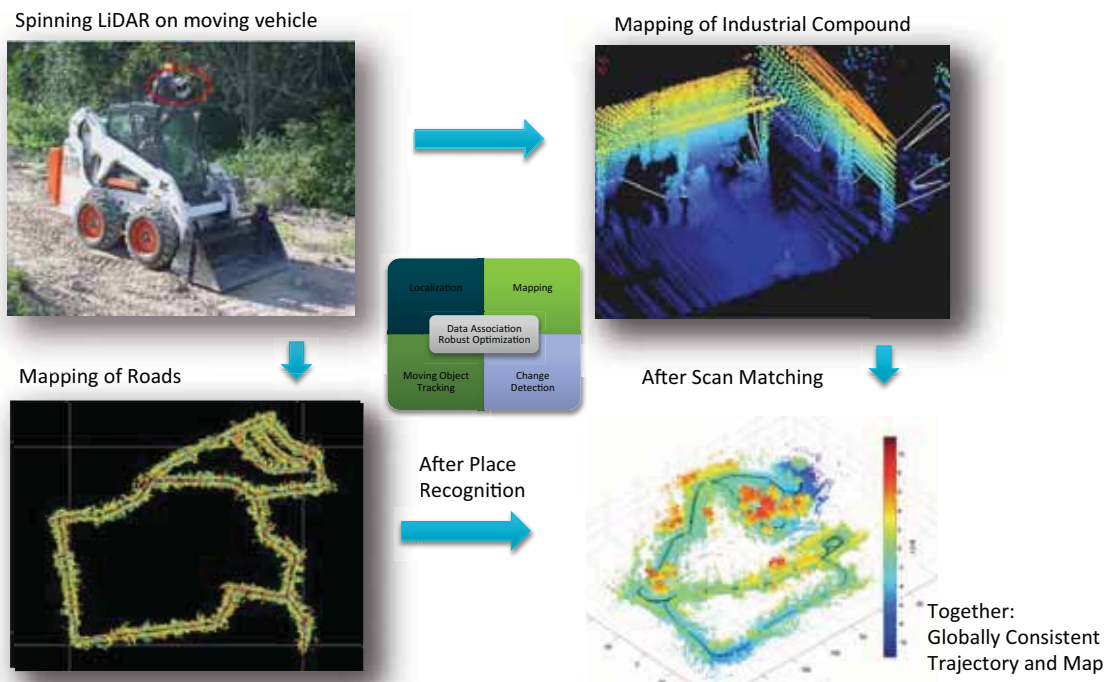
Airborne Mapping



Evolution of Mobile Mapping



3D SLAM



Zebedee

Mobile and handheld 3D mapping

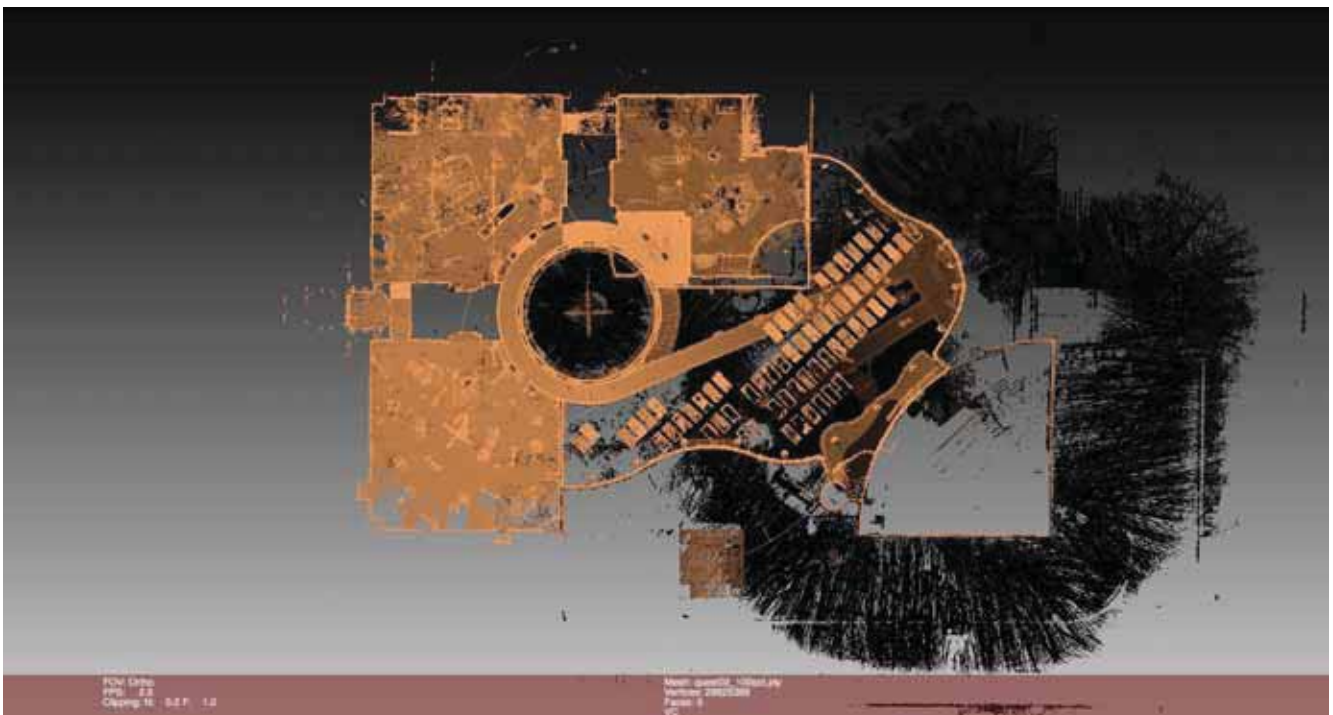


Fort Lytton
(Brisbane)

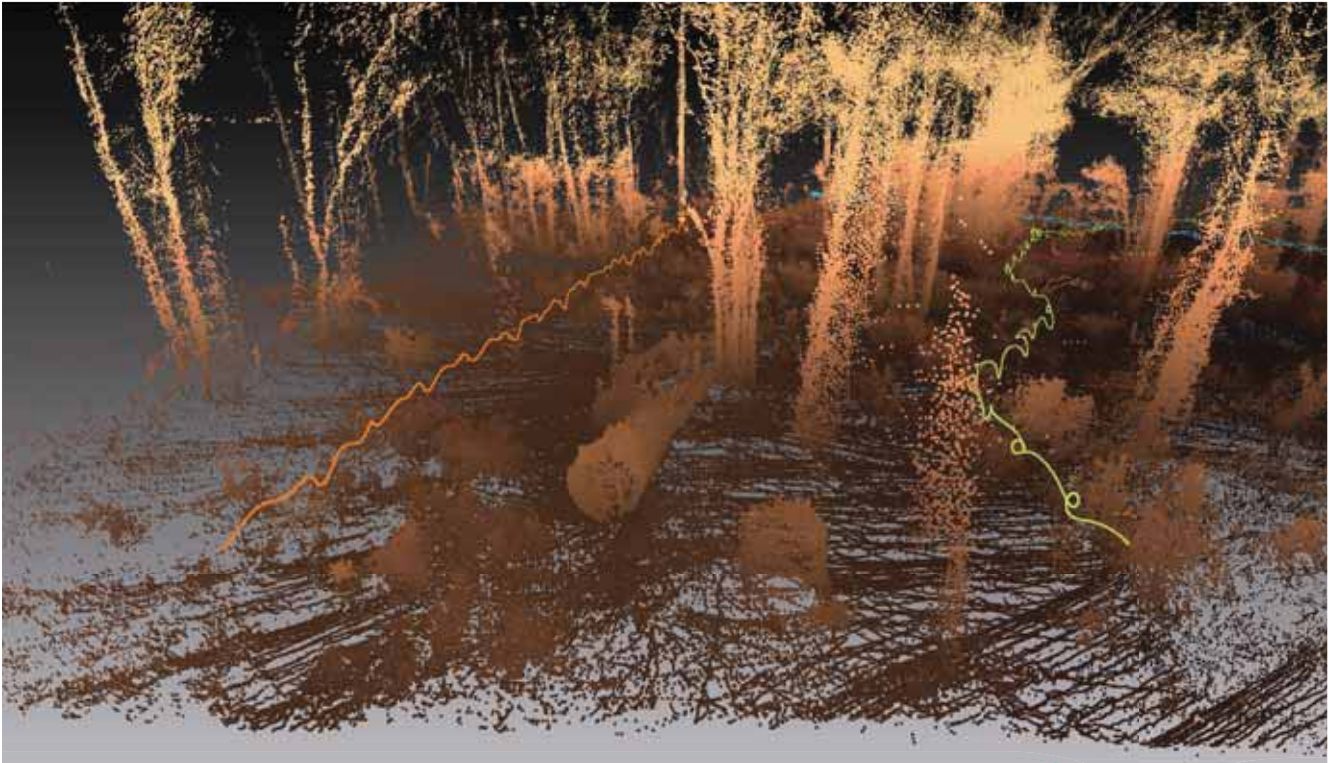


Revolutionizes the way 3D mapping can be achieved by cutting acquisition time from hours and days to minutes

Indoor Mapping: Questacon - Canberra



Forestry



Pisa - Piazza del Duomo

Bottom to top and around the bell tower in 20 mins

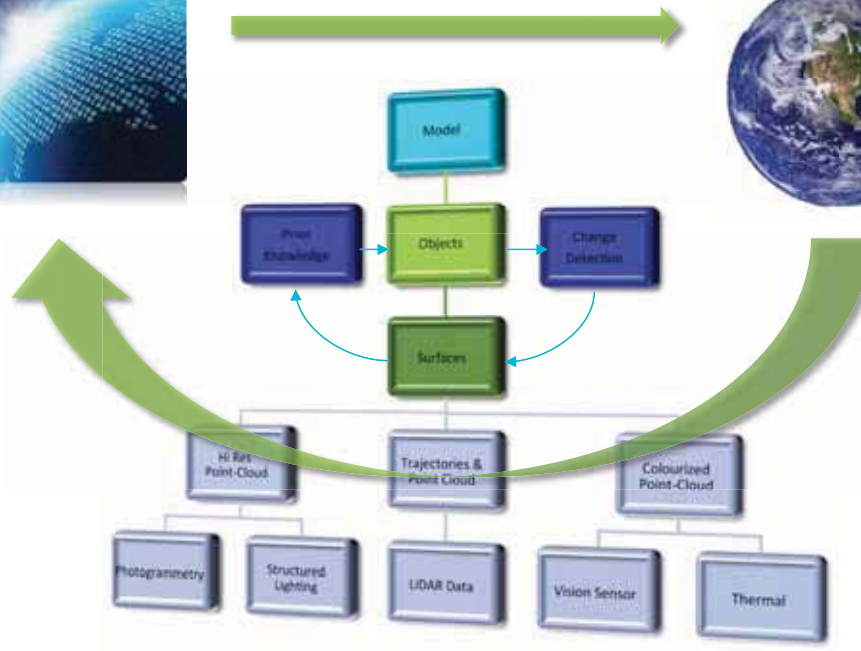


Augmented World Models

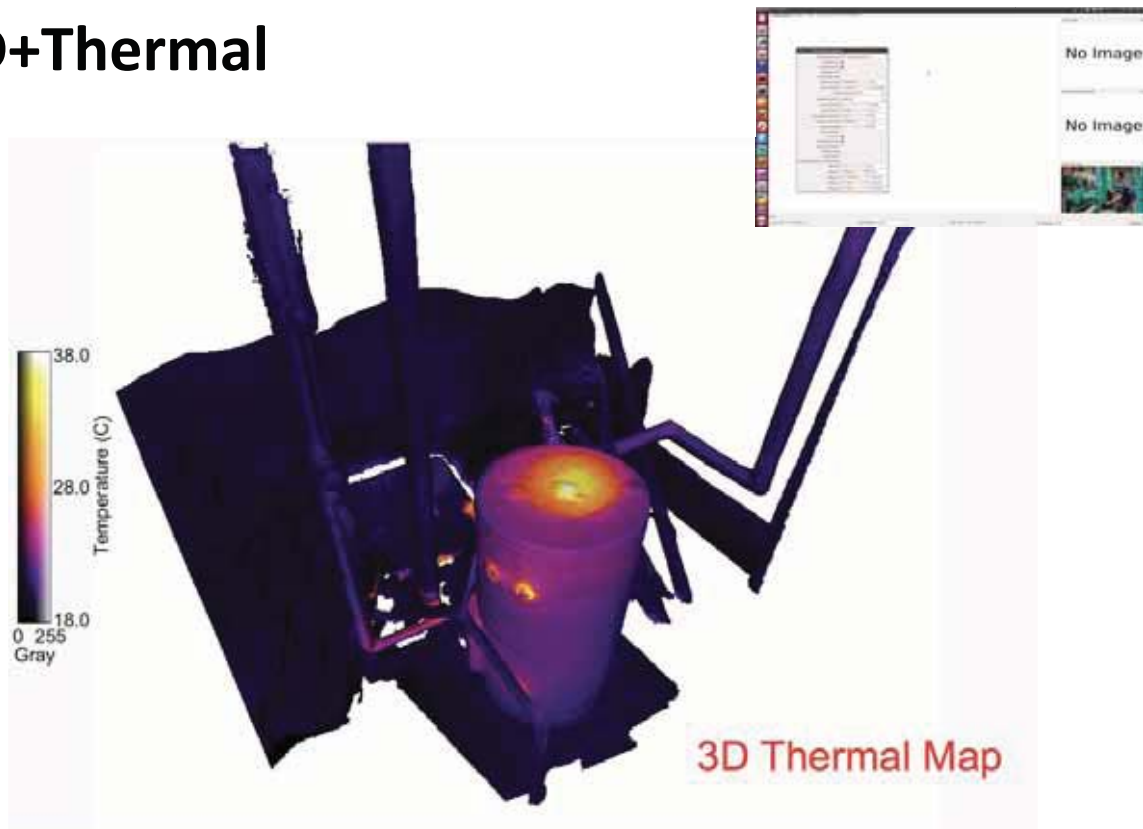
Digital World



Real World



3D+Thermal



Peel Island



Three operators (90 mins each)



Peel Island

Structures and trees mapped



Peel Island 3D + RGB



Seeker Rover with Multi-Modal Sensor Payload



Alberto Elfes, PhD
Senior Principal Research Scientist & Robotics Research Group Leader

Autonomous Systems Lab

t +61 7 3327 4355

e Alberto.Elfes@csiro.au

w <http://research.ict.csiro.au/research/labs/autonomous-systems>

AUTONOMOUS SYSTEMS LAB | CCI
www.csiro.au





Session II (2/2)

Perception in outdoor environment

- **Title: Localization of Planetary Exploration Rovers with Orbital Imaging: a survey of approaches**

Author: E. Boukas¹, A. Gasteratos¹ and G. Visenti

- **Title: A probabilistic distribution approach for the classification of urban roads in complex environments**

Author: G. B. Vitor, A. C. Victorino and J. V. Ferreira

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICRA 2014



Localization of Planetary Exploration Rovers with Orbital Imaging: a survey of approaches

Evangelos Boukas¹, Antonios Gasteratos¹ and Gianfranco Visentin²

Abstract—Owing to the rovers exploring the surface of Mars being assigned with ever more complex tasks, it is the autonomy of such operations that enables their effective planetary activities. An indicative case of the escalation in requirements is the upcoming Mars Sample Return mission (MSR), which is a joint effort of the National Aeronautics and Space Administration (NASA) and the European Space Agency (ESA). The mission involves a Sample Fetching Rover (SFR) to gather a previously deposited cache of soil that, eventually, will be directed towards Earth. With the aim to retrieve the cache and rendezvous with the Mars Ascent Vehicle (MAV), the rover should be apt to globally localize itself on the Martian surface. Due to the absence of Global Navigation Satellite System (GNSS) on Mars, the most suited approach is the blending of information stemming from ground rovers and orbital imagery. The scope of the paper in hand is to summarize the work delivered so far on the localization of space exploratory rovers based on such information.

I. INTRODUCTION

Upon a successful probe landing along the robotic Mars exploration missions, one of the most important tasks is the accurate global localization of the rover on inertial and fixed coordinate systems, such as the Mars Mean Equator and IAU vector of J2000 frame and Mars body-fixed rotating frame [1], [2]. The localization of a space exploratory rover on a georeferenced orbital image is equivalent to global localization and, hence, is considered sufficient. During the last decade, some approaches have been proposed employing urban structures earthly observed by the robot as patterns for orbital recognition. For example, the authors in [3], [4] employ the *skyline* perceived by a fisheye camera in a city block, in order to match it with the 3D models of the city buildings. Other approaches utilize stereoscopic techniques to create top views of either the fully perceived field of view [5], [6] or of prominent obstacles [7] which are then matched to aerial equivalents. However, such dense, prominent regions do not exist in the Martian surface and, therefore, these techniques are of limited interest for Mars exploration missions.

Bearing in mind that an one-way signal to Mars requires approximately 20 *min*, it is apparent that the non autonomous navigation of rovers poses a noticeable overhead to the exploration of the red planet. Therefore, recent and future space exploratory rovers are designed for long traverses [8],

[9]. This is the reason why the rovers are equipped with several cameras. An example of camera setup design for such a specific application is reported in [10], where a camera system for localization and mapping of space exploratory rovers is proposed. The algorithms required for the long range autonomous navigation are 3D reconstruction, mapping [11], [12], localization and path planning. The algorithms that are currently implemented in the space exploratory rovers can be found in [13], whilst a detailed review of visual odometry methods is available in [14].

Moreover, notwithstanding the advanced and noteworthy mechanical and electrical design of space exploratory rovers, their capabilities in terms of computational and power needs are modest, especially compared to contemporary robots. The algorithms for the localization, as well as for any other task, should be as lightweight as possible. Toward this end, the authors in [15] have implemented a Visual Odometry (VO) algorithm that is capable of achieving state of the art results, while being of low computational cost. In terms of power consumption, the authors in [16] have implemented the localization and mapping algorithms on FPGA devices for ESA's future space exploratory rovers. The gain of this implementation is twofold: Firstly the cost, in power, of the execution of these algorithms is diminished and, moreover, the parallel implementation on the FPGA speeds up the algorithms to frequencies higher than 1Hz.

The manuscript is organized as follows: Section II categorizes and surveys all the approaches that employ information stemming from aerial and orbital imagery in order to localize rovers operating on the Martian surface. A proposal for the assessment of orbital based localization techniques is described in Section III and, last, conclusions are drawn in Section IV.

II. CATEGORIZATION OF METHODS

An attempt to categorize the techniques employed in the localization of planetary exploration rovers based on orbital imaging results into three major classes of approaches. These are separated according to specific characteristics summarized as follows:

- employment of integrated descent imagery to capture the Martian surface (Subsection: II-A)
- registration of the skyline acquired from rover on the Orbital terrain model (Subsection: II-B)
- seeking for common interest areas on both rover and orbital imagery to accomplish localization; this category is further distinguished into methods utilizing points of

¹Evangelos Boukas and Antonios Gasteratos are with the School of Engineering, Democritus University of Thrace, Vas. Sophias 12, GR-67100 Xanthi, Greece. evanbouk@pme.duth.gr, agaster@pme.duth.gr

²Gianfranco Visentin is the Head of Automation and Robotics Section (TEC-MMA), European Space Agency, The Netherlands. Gianfranco.Visentin@esa.int

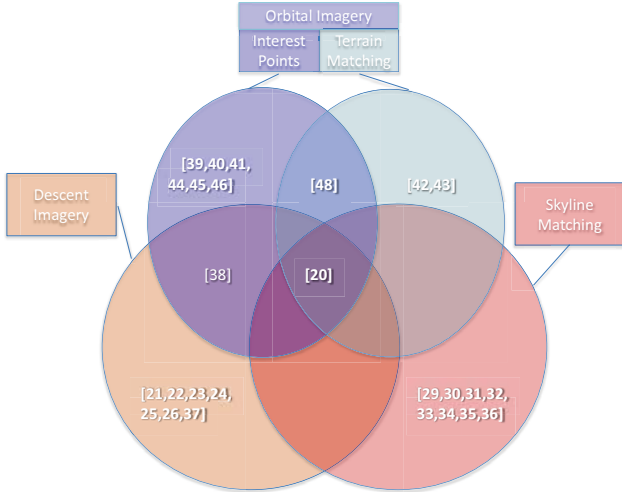


Fig. 1. The Venn diagram for the categorization of the approaches considered in this review.

interest and ones using terrain matching (Subsection: II-C)

The techniques being reviewed in the context of the paper in hand are graphically assembled in Fig. 1.

A. LOCALIZATION WITH DESCENT IMAGERY

Following the design and implementation of NASA's first Mars rover, namely the Pathfinder's Sojourner [17], the utilization of descent imaging for the localization of the robot near its landing position has been examined. The authors in [18], [19] presented a detailed explanation of state of the art descent and landing approaches followed by NASA and ESA. Matthies et al. [20] proposed the employment of *Digital Elevation Maps* (DEM), computed from consequent descent images, as prior information to the rover's localization system. The authors theoretically proved the possibility to create DEMs by applying cross-correlation to match common surface points appearing on descent images at different scales. They also identified three approaches for the localization of rovers:

- terrain matching among ground and orbital deriving 3D models
- skyline identification on rover stemming images
- pairing of structures of interest, such as ridges, ravines and rocks on descent and rover imagery

1) *DEM Construction from Descent Images*: The automatic generation of a hierarchical DEM from consequent descent images is explained in detail in [21]. At the descent image "timeline" every image is acquired at a half distance from the previous one (starting from 5000 m). Three steps comprise the overall methodology:

- initial one-meter spacing DEM
- refinement of the one-meter spacing DEM
- generation of the hierarchical DEM

Firstly, Ground Control Points (GCPs) are used to acquire the exterior orientation. Then, a six-point based affine transfor-

mation is performed to register each image to the previous one. A cross-correlation metric is used to find the correspondences between consecutive descent frames. The descent images present a vertical baseline and the 3D position of each matched point is inserted in the ground coordinate system by triangulation, forming the initial $1m$ resolution DEM grid. Since the lower images have greater resolution, the initial DEM grid is refined by a bottom-down and bottom-up methodology that performs bundle adjustment to ensure that each point has acquired the most accurate elevation. Lastly, the DEM is formed into a hierarchical layered grid with different resolutions ranging from $0.1-1m$.

2) *Incremental Localization*: In [22] the localization of the rovers position is performed at each time step, by an incremental bundle adjustment. The difference of the implementation presented in [23] is that for each observation only the current and previous state set of variables are preserved and not the entire set of system variables. In order to build a bundle adjustment that would take into consideration both descent and rover image data, features appearing in both kinds of images were manually selected by an operator to serve as tie points [24], [25]. For a detailed analysis of the incremental bundle adjustment that includes both descent and rover features, the reader should refer to [23]. The performance possibilities of the image network of descent and rover imagery is proved in [26]. Li et al. [22] presented an astonishing overall localization error of less than 0.1% on a 500 meters course.

It is noteworthy to mention that the aforementioned approaches suffer from some crucial drawbacks. The first and foremost is that the working area is limited due to the coverage and the resolution of the descent images and the produced DEMs, as explained in [20]. Moreover the localization of the rovers (without the existence of GCPs, which are unfeasible on Mars) is performed by employing a local coordinate frame. Therefore, the global location - a *condicio sine qua non* in advanced space applications - remains unknown. Lastly, all of the approaches require a human operator to select strong tie points, due to the fact that the high difference in scale and orientation between the rover and descent images does not permit robust extraction and matching of common features. In some situations, even a manual selection of such points is difficult, resulting in a deterioration of localization accuracy up to 300% [23].

B. SKYLINE MATCHING

The skyline is a curve that passes through the edge of the horizon. Contrary to other approaches, the horizon matching does not aim to accurately localize a rover, but to calculate a preliminary uncertainty location area. The approach can be considered as an extension to the sun detection methods [27], where the localization produces an area of uncertainty in the order of 20 km. Some approaches relying on the horizon matching for rover localization exist, which assume accurate rover orientation based on sun sensor as the one presented in [28]. The most influencing and informative approaches are the one introduced by Stein and Medioni [29], [30]

and the one presented by Cozman and Krotkov [31], [32], which is known as the VISual Position Estimation for Rovers (VIPER). The main parts of both methodologies are summarized as follows:

- skyline detection from rover images
- skyline detection on DEMs
- feature extraction on skylines (optional)
- search and locate rover skyline on DEM images
- pose estimation based on the skyline location

The main approach is relatively easy to understand. The rover is considered to be at an unknown position, also known as the “lost in space” or “drop off” problem, laying inside an area that has been mapped, by means of orbital imaging. Also, the robot is considered to hold precise information about its orientation, as explained previously. The rover is equipped with panoramic cameras, or normal cameras on pan units, capable of capturing full circle images of the horizon. Then, the skyline is detected employing image segmentation techniques. Assuming a DEM covering the area of uncertainty, within which the robot lays and given the orientation of the rover, a “simulated” skyline is rendered at each and every point of the DEM. A search follows to match the rover’s skyline to the rendered ones.

1) *Skyline Detection and Feature Extraction on Skyline:*

There are two approaches of including the skyline into the system: (i) feature based and (ii) signal based ones. Talluri and Aggarwal [33] as well as Stein and Medioni [30] and Cozman in [32] utilized a signal-based approach, as the skyline is the elevation of the highest observed obstacle from the horizontal plane. On the contrary, the authors in [31], [33], [34] utilize a feature vector of the aforementioned signal. The vectors of rover and orbital stemming skylines are computed in the same manner.

2) *Skyline detection on DEMs:* Such approaches require the computation of skylines for each and every point in the DEM, through a procedure called, “skyline rendering”. Assuming an interval ϕ , which represents an azimuth orientation and ranges into $[0 : N : 359]$, where N is the required resolution, a skyline is predicted at each point of the DEM. The calculation of the skyline employs the separate line scanning and sampling on the DEM at each ϕ and due to the discreteness of the DEM requires an interpolation. An analytic explanation of the actual VIPER implementation can be found in [32]. According to the authors in [35], the computation of all those possible skylines is computational expensive, as it might “take some days for each DEM” but it is only executed once.

3) *Rover Position Estimation:* Given a full set of possible locations in a DEM and the corresponding skylines, the next step comprises the matching with the rover’s actual skyline. This is performed by a Bayesian posterior estimator that provides the probability of a rover to be at each point on the map.

Extensive testing has been performed by Furgale et al. [35], proving that although the VIPER algorithm is able to provide adequate results in some circumstances, a lot of conditions exist that result in software unresponsiveness or

inaccurate estimation. The skyline approaches are prone to specific area formations, such as planar, repetitive scenery or even to a close obstacle that can cause occlusion [30]. An additional issue with such methods is the extend of the mapped area taken into consideration. The extend of the DEM should be large enough, to include all the areas that appear in the rovers horizon, but in the same time to be sufficiently small so as to allow adequate resolution and, hence, accuracy.

C. LOCALIZATION WITH ORBITAL-ROVER IMAGERY INTEGRATION

The global localization of Mars rovers at their landing site has been employed by Direct to Earth (DTE) radio signals, via two-way Doppler tracking [36] or via descent image analysis, as reviewed in Section II-A. The Mapping and GIS laboratory of the Ohio State University (OSU) has extensively analyzed the localization of the MER rovers, employing DEMs either from descent images or from the orbiting imagery [37]. Nevertheless, the majority of approaches comprise steps that require the manual selection of common points by a human operator. Some of the approaches that appear in the literature point towards the automation of the global localization, yet without any of them reaching the state of completeness that would allow its selection for realization.

1) *MER Rovers Localisation Employing Orbital Imagery:*

The authors in [38] describe the initial approaches for the localization of MER rovers. The OSU Mapping and GIS Laboratory performed incremental bundle adjustment on the rover image network and on the orbital images, separately, in order to assist the operations of MER [38], [39], [37]. Although the inter-stereo tie points were computed automatically by 90%, the cross-site ties were computed manually. Nevertheless, the orbital imaging proved that the mapping procedures of the rover level were performing adequately. Li et al. [40] noted that with the aim to produce high quality localization estimates, orbital and rover image networks should be connected through tie points. Moreover, the authors marked the difficulty of extracting such features due to great differences in the scaling and viewing angle. In order to practically prove the necessity of the integration of rover and orbital imagery, the authors in [41] compared the bundle adjustment VO with a traverse that was corrected utilizing orbital images. They overcome the barrier of ground points extraction by manually selecting the common points. The comparison results in a deviation of 1.5% among the two approaches, which can be corrected with the automation of the tie-points selection.

2) *Terrain Matching:* Van Pham et al. [42] employed a Bayesian recursive algorithm, namely a modified particle filter able to retrieve the location of the rover on a global DEM. The rover was equipped with a stereo camera used to create a local DEM. The state of the particle was chosen to be: $X = [x, y, \theta]$, where x, y are the spatial coordinates and θ is the orientation. This simplified approach was selected because of the global DEM formation. The particles were resampled into a discrete grid with the same spatial

resolution of the global DEM. The experiments were carried out in a sand quarry and the global DEM was created by means of a UAV, while ground control points were used for georeference. The robot employed was Ral Space's Rimmer, as described in [43]. The resulting accuracy, which is confirmed by DGPS, is $1m$, equal to the resolution of the DEM and the convergence distance is at $58-78m$. The approach seems reasonable and well documented, however the examined area is quite limited. The uncertainty of rover location may reach up to 10ths of km , a range within which no convergence is guaranteed. Finally, the computational burden of the initialization of every point of DEM will lead to an unfeasible implementation.

3) *Interest Point Matching*: Di et al. [44] propose a novel approach for the "orbital-based rover localization" problem, by incorporating both rock detection/matching and Scale Invariant Feature Transform (SIFT) feature detection/matching in a system that handles outliers by employing a procedure similar to RANdom SAmple Consensus (RANSAC). The rock detection on the rover field of view is an intuitive algorithm that represents a 3D morphological filter, which is executed incrementally and leads to extraction of rock peak points [45]. The rock detection on the orbital image is a statistical intensity filter bearing constraints that ban shadows from being detected as a rock. The matching of the rover and orbital rocks employs a RANSAC-based algorithm, which through a random initial sample, is able to: (i) transform the rover rocks on the orbital image coordinate system and (ii) select the matches with a threshold over the Euclidean distance. Moreover, the system involves SIFT detection and matching among the orbital image and an orthophoria, which is created via the rover's stereo images. A similar to the aforementioned RANSAC based outlier detection algorithm is applied to the matched points. The remaining inliers are used to calculate the position of the rover inside the orbital image. The employed dataset does not contain a groundtruth as it includes real Mars rover images.

Carle et al. [46] proposed the employment of a LIGht Detection And Ranging (LIDAR) device in a system that is able to globally localize a rover laying within the imaging region of a georeferenced DEM. The approach follows a feature based approach, with features being the prominent peaks that appear on local and global DEMs. The local DEM is produced with a transformation on a single LIDAR scan. The extraction of the features, a peak detection in particular, is performed on the images, by filtering them with the external morphological gradient, i.e. the difference between the dilated image and the original one. The feature extraction is the same on orbital and rover DEMs. The correspondences between the rover and ground features are computed utilising a method named "Data-Aligned Rigidity-Constrained Exhaustive Search" (DARCES) [47]. Firstly, several groups of three matched points are considered and the group the resulting 2D transformation of which offers the least error is selected. This estimation is then fed into a "Simultaneous Localization and Mapping" (SLAM) module, namely the MOGA which includes a

RANSAC outlier detection part. The performance of the system is noteworthy with a $22.1 m$ error at a range of $1.5 km$ with a DEM resolution of $13m \times 13m$. The authors compare their system with VIPER (see Section II-B), which they outperform. The most significant issue of this work is the infeasibility of the LIDAR integration on a Mars Rover due to both heaviness and power consumption.

The last approach being presented here is the one developed by Hwangbo et al. [48]. The distinctiveness of this approach lays on the the employment of both full terrain and rock matching to hierarchically trace correspondences between orbital and rover images. Firstly, the 3D terrain matching is utilized to locate the region of interest in the global DEM that corresponds to the rover position. The rover DEM is created from stereo imagery and both DEMs are treated as images. With the purpose of having the same scale in elevation, the mean elevation is subtracted from both DEMs. Then, the matching is performed by identifying the region of the maximum weighted correlation of values and slopes within the global DEM. As soon as the region of interest is located the rock extraction and pattern matching is performed. The rock detection on rover imagery includes the removal of ground points and the identification of peaks, i.e. the highest points in an area having elevation of more than $25cm$. The rocks on orbital imagery are identified via an intensity thresholds technique. Furthermore the morphology of the rocks is examined, discarding those rocks with long axis longer than $2m$ and short axis shorter than $0.2m$. The rotation angles are considered to be zero and, thereby, the translation vector which produces the most matches is selected. Yet, it should be mentioned that this method is prone to local minimum on the terrain matching, which result in false localization estimation.

III. ASSESSMENT

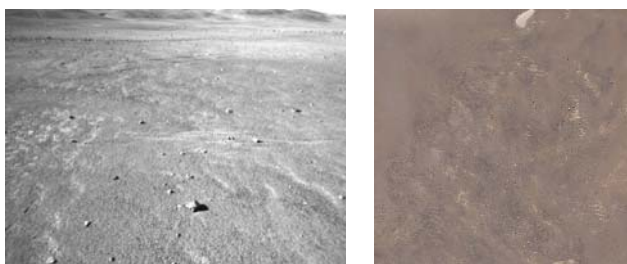
As rovers are called to fulfill more complicated functions the need for higher accuracy and autonomy correspondingly emerges. During the last years there has been an increasing interest in approaches for the localization of planetary rovers utilizing orbital imagery. Howbeit, a standard procedure for the assessment and comparison of a new reported method does not exist yet and neither does a standard dataset upon which the method can be valorized. Thus, taking an interest in robustly solving the localization problem, it is our belief that a benchmark framework should be setup with a view to define a common base-line for the coming out approaches. The main points of such a framework can be summarized as follows:

- **Real World Relevance**: In order to prove the relevance of a newly proposed method, one should test his/her approaches on specific and real world datasets. According to the authors, the most sufficient datasets are the ones produced by the ESA on the Chilean Atacama Desert (the ones provided by SEEKER [43] and SAFER activities). The selected areas in the Chilean desert are considered to be the most Mars-like regions on Earth

(Figure 2). Furthermore, these datasets contain both rover stereo images, registered with DGPS measurements for groundtruth and georeferenced aerial images, sampled appropriately to resemble the orbital imagery on Mars.

Another source of data could be the actual MER and MSL programs, but since the stereo images are sparse (at least sparser compared the actual rover's frame-rate) and no groundtruth exists, then the Atacama datasets are the most appropriate.

- **Accuracy:** The accuracy requirements of space rovers localization should be similar to the requirements on Earth. The accuracy should be measured both by: (i) the accumulated signed error, expressed as the difference between the measured and the desired position at the end point and (ii) the evolution of the error in all 6 Degrees of Freedom (DOF) of the robot's pose along its course. More specifically, in the case of utilization of orbital data, it is the the resolution of the respective imagery that defines the maximum resolution that can be achieved. Nowadays, the imagery of High Resolution Imaging Science Experiment (HiRISE) [49] is able to produce orthorectified images of 0.25m spatial resolution and DTMs with 1m resolution.
- **Repeatability:** The method should be tested at different scenarios with different trajectories and surrounding environments. This may be accomplished by the concurrent usage of simulators, such as the Pangu and 3DRov, [50], [51], for instance.
- **Feasibility:** The approach should be feasible, taking into consideration current space rover apparatus. Any addition to the current equipment should be accompanied by a feasibility assessment or, at least, a reasonable prediction of such future space qualified hardware.
- **Openness:** The performance should be tested publicly similar to the successful Middlebury evaluation for stereo correspondence algorithms [52], [53].



(a) Left image of a stereo pair (b) Section of orbital image

Fig. 2. Sample Data of Atacama dataset.

IV. DISCUSSION

We have presented the approaches that have been proposed during last decades for the problem orbital-based rover localization. We have classified the approaches into three categories, based on the type of information used for

localization: Descent Imagery, Skyline and Orbital Imagery ones. Along the last years, there has been observed an increasing attention to the problem and it is anticipated that more and more of novel approaches will emerge in the near future. Therefore, we have proposed a benchmark framework for the reliable and undisputed evaluation of such methods. One of the issues that arise from our study is the lack of robust techniques to overcome the manual tie point selection among the orbital and rover imagery. Towards this end the authors in [54] are examining the extraction of such commonly observed regions of interest.

ACKNOWLEDGMENT

This work was funded by the European Space Agency via the Network/Partnering Initiative under the contract No. 4000109064/13/NL/PA.

REFERENCES

- [1] B. A. Archinal, M. Ahearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, *et al.*, "Report of the iau working group on cartographic coordinates and rotational elements: 2009," *Celestial Mechanics and Dynamical Astronomy*, vol. 109, no. 2, pp. 101–135, 2011.
- [2] B. Semenov. (2013, May) Msl frames kernel. [Online]. Available: <http://naif.jpl.nasa.gov/pub/naif/MSL/kernels/fk/mslv08.tf>
- [3] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, "Geolocalization using skylines from omni-images," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 23–30.
- [4] —, "Skyline2gps: Localization in urban canyons using omni-skylines," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3816–3823.
- [5] T. Senlet and A. Elgammal, "A framework for global vehicle localization using stereo images and satellite and road maps," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2034–2041.
- [6] E. Boukas, I. Kostavelis, L. Nalpantidis, and A. Gasteratos, "Graph based localisation refinement by orbital images," 2012.
- [7] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard, "Large scale graph-based slam using aerial images as prior information," *Autonomous Robots*, vol. 30, no. 1, pp. 25–39, 2011.
- [8] M. Van Winnendael, P. Baglioni, and J. Vago, "Development of the esa exomars rover," in *Proc. 8th Int. Symp. Artif. Intell., Robot. Automat. Space*, 2005, pp. 5–8.
- [9] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, *et al.*, "Mars science laboratory mission and science investigation," *Space science reviews*, vol. 170, no. 1–4, pp. 5–56, 2012.
- [10] I. Kostavelis, E. Boukas, L. Nalpantidis, A. Gasteratos, and M. A. Rodrialvarez, "Spartan system: towards a low-cost and high-performance vision architecture for space exploratory rovers," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1994–2001.
- [11] L. Nalpantidis, I. Kostavelis, and A. Gasteratos, "Stereo-vision-based algorithm for obstacle avoidance," *Intelligent Robotics and Applications*, pp. 195–204, 2009.
- [12] L. Nalpantidis, G. Sirakoulis, and A. Gasteratos, "Non-probabilistic cellular automata-enhanced stereo vision simultaneous localization and mapping," *Measurement Science and Technology*, vol. 22, 2011.
- [13] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.
- [14] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.

- [15] I. Kostavelis, E. Boukas, L. Nalpantidis, and A. Gasteratos, "Visual odometry for autonomous robot navigation through efficient outlier rejection," *IEEE International Conference on Imaging Systems and Techniques*, 2013, *In Press*, 2013.
- [16] I. Kostavelis, L. Nalpantidis, E. Boukas, M. A. Rodrigalvarez, I. Stamoulias, G. Lentarlis, D. Diamantopoulos, K. Siozios, D. Soudris, and A. Gasteratos, "Spartan: Developing a vision system for future autonomous space exploration robots," *Journal of Field Robotics*, 2013.
- [17] A. H. Mishkin, J. C. Morrison, T. T. Nguyen, H. W. Stone, B. K. Cooper, and B. H. Wilcox, "Experiences with operations and autonomy of the mars pathfinder microrover," in *Aerospace Conference*, 1998 *IEEE*, vol. 2. IEEE, 1998, pp. 337–351.
- [18] R. Prakash, P. D. Burkhart, A. Chen, K. A. Comeaux, C. S. Guernsey, D. M. Kipp, L. V. Lorenzoni, G. F. Mendek, R. W. Powell, T. P. Rivellini, *et al.*, "Mars science laboratory entry, descent, and landing system overview," in *Aerospace Conference*, 2008 *IEEE*. IEEE, 2008, pp. 1–18.
- [19] P. Martella, M. Buonocore, D. Desiderio, M. Lovera, and S. Portigliotti, "Soft landing on mars: The gnc tasks in the exomars descent module mission," in *7th International ESA Conference on Guidance, Navigation and Control Systems*, Tralee, Ireland, 2008.
- [20] L. Matthies, C. Olson, G. Tharp, and S. Laubach, "Visual localization methods for mars rovers using lander, rover, and descent imagery," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, Tokyo, Japan. Citeseer, 1997, pp. 413–418.
- [21] F. Xu, F. Ma, R. Li, L. Matthies, and C. Olson, "Automatic generation of a hierarchical dem for mars rover navigation," in *Proceedings of the 3rd Mobile Mapping Conference*, Cairo, Egypt, 2001.
- [22] R. Li, F. Ma, F. Xu, L. Matthies, C. Olson, and Y. Xiong, "Large scale mars mapping and rover localization using descent and rover imagery," *International Archives of Photogrammetry and Remote Sensing*, vol. 33, no. B4/2; PART 4, pp. 579–586, 2000.
- [23] F. Ma, K. Di, R. Li, L. Matthies, and C. Olson, "Incremental mars rover localization using descent and rover imagery," in *ASPRS Annual Conference*, 2001, pp. 25–27.
- [24] R. Li, F. Ma, F. Xu, L. H. Matthies, C. F. Olson, and R. E. Arvidson, "Localization of mars rovers using descent and surface-based image data," *Journal of Geophysical Research Planets (1991–2012)*, vol. 107, no. E11, pp. F10D–4, 2002.
- [25] F. Xu, "Automation in mars landing-site mapping and rover localization," in *ISPRS XXth congress*. Citeseer, 2004, pp. 12–23.
- [26] R. Li, F. Ma, F. Xu, L. Matthies, C. Olson, and Y. Xiong, "Mars rover localization using descent and rover imagery-result of the field test at silver lake, ca," in *ASPRS Annual Conference*, 2000, pp. 22–26.
- [27] F. Cozman and E. Krotkov, "Robot localization using a computer vision sextant," in *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 1. IEEE, 1995, pp. 106–111.
- [28] P. Furgale, J. Enright, and T. Barfoot, "Sun sensor navigation for planetary rovers: Theory and field testing," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 47, no. 3, pp. 1631–1647, 2011.
- [29] F. Stein and G. Medioni, "Map-based localization using the panoramic horizon," in *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992, pp. 2631–2637.
- [30] —, "Map-based localization using the panoramic horizon," *Robotics and Automation, IEEE Transactions on*, vol. 11, no. 6, pp. 892–896, Dec 1995.
- [31] F. Cozman and E. Krotkov, "Automatic mountain detection and pose estimation for teleoperation of lunar rovers," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3. IEEE, 1997, pp. 2452–2457.
- [32] F. Cozman, E. Krotkov, and C. Guestrin, "Outdoor visual position estimation for planetary rovers," *Autonomous Robots*, vol. 9, no. 2, pp. 135–150, 2001.
- [33] R. Talluri and J. Aggarwal, "Position estimation for an autonomous mobile robot in an outdoor environment," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 573–584, 1992.
- [34] P. C. Naval Jr, M. Mukunoki, M. Minoh, and K. Ikeda, "Estimating camera position and orientation from geographical map and mountain image," in *38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*. Citeseer, 1997.
- [35] P. Furgale, P. Carle, and T. D. Barfoot, "A comparison of global localization algorithms for planetary exploration," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 4964–4969.
- [36] J. R. Guinn, "Mars surface asset positioning using in-situ radio tracking," in *Proceedings of the 11th Annual AAS/AIAA Space Flight Mechanics Meeting*, Santa Barbara, CA, 2001, pp. 45–53.
- [37] R. Li, R. E. Arvidson, K. Di, M. Golombek, J. Guinn, A. Johnson, M. Maimone, L. H. Matthies, M. Malin, T. Parker, *et al.*, "Opportunity rover localization and topographic mapping at the landing site of meridiani planum, mars," *Journal of Geophysical Research: Planets (1991–2012)*, vol. 112, no. E2, 2007.
- [38] R. Li, S. W. Squyres, R. E. Arvidson, B. A. Archinal, J. Bell, Y. Cheng, L. Crumpler, D. Marais, K. Di, T. A. Ely, *et al.*, "Initial results of rover localization and topographic mapping for the 2003 mars exploration rover mission," *Photogrammetric engineering and remote sensing*, vol. 71, no. 10, p. 1129, 2005.
- [39] R. Li, B. A. Archinal, R. E. Arvidson, J. Bell, P. Christensen, L. Crumpler, D. J. Des Marais, K. Di, T. Duxbury, M. Golombek, *et al.*, "Spirit rover localization and topographic mapping at the landing site of gusev crater, mars," *Journal of Geophysical Research: Planets (1991–2012)*, vol. 111, no. E2, 2006.
- [40] R. Li, K. Di, J. Hwangbo, and Y. Chen, "Integration of orbital and ground images for enhanced topographic mapping in mars landed missions," in *Proceedings of the Annual NASA Science Technology Conference (NTSC)*, College Park, MD, 2007.
- [41] R. Li, S. He, Y. Chen, M. Tang, P. Tang, K. Di, L. Matthies, R. E. Arvidson, S. W. Squyres, L. S. Crumpler, *et al.*, "Mer spirit rover localization: Comparison of ground image-and orbital image-based methods and science applications," *Journal of Geophysical Research: Planets (1991–2012)*, vol. 116, no. E7, 2011.
- [42] B. Van Pham, A. Maligo, and S. Lacroix, "Absolute map-based localization for a planetary rover," in *Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA)*, 2013.
- [43] M. Woods, A. Shaw, E. Tidey, B. Van Pham, U. Artan, B. Maddison, and G. Cross, "Seeker-autonomous long range rover navigation for remote exploration," *International Symposium Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, 2012.
- [44] K. Di, Z. Liu, and Z. Yue, "Mars rover localization based on feature matching between ground and orbital imagery," *Photogrammetric engineering and remote sensing*, vol. 77, no. 8, pp. 781–791, 2011.
- [45] R. Li, K. Di, A. B. Howard, L. Matthies, J. Wang, and S. Agarwal, "Rock modeling and matching for autonomous long-range mars rover localization," *Journal of Field Robotics*, vol. 24, no. 3, pp. 187–203, 2007.
- [46] P. J. Carle, P. T. Furgale, and T. D. Barfoot, "Long-range rover localization by matching lidar scans to orbital elevation maps," *Journal of Field Robotics*, vol. 27, no. 3, pp. 344–370, 2010.
- [47] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng, "Ransac-based darces: A new approach to fast automatic registration of partially overlapping range images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 11, pp. 1229–1234, 1999.
- [48] J. W. Hwangbo, K. Di, and R. Li, "Integration of orbital and ground image networks for the automation of rover localization," in *ASPRS 2009 Annual Conference*, 2009.
- [49] R. Kirk, E. Howington-Kraus, M. Rosiek, J. Anderson, B. Archinal, K. Becker, D. Cook, D. Galuszka, P. Geissler, T. Hare, *et al.*, "Ultrahigh resolution topographic mapping of mars with mro hirise stereo images: Meter-scale slopes of candidate phoenix landing sites," *Journal of Geophysical Research: Planets (1991–2012)*, vol. 113, no. E3, 2008.
- [50] S. Parkes, I. Martin, M. Dunstan, and D. Matthews, "Planet surface simulation with pangu," in *Eighth International Conference on Space Operations*, 2004, pp. 1–10.
- [51] P. Poulakis, L. Joudrier, S. Wailliez, and K. Kapellos, "3drov: A planetary rover system design, simulation and verification tool," in *Proceedings of the 10th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS-08)*, 2008.
- [52] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.
- [53] "Middlebury stereo evaluation [online]," <http://vision.middlebury.edu/stereo/eval/>, accessed: 2014-01-01.
- [54] E. Boukas, A. Gasteratos, and G. Visentin, "Extraction of common regions of orbital and rover (ground) acquired imagery," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (iSAIRAS)*, 2014.

A probabilistic distribution approach for the classification of urban roads in complex environments

Giovani B. Vitor^{1,2}, Alessandro C. Victorino¹ and Janito V. Ferreira²

Abstract—Navigation in urban environments has been receiving considerable attention over the past few years, especially for self-driving cars. Road detection for Autonomous Systems, and also for ADAS (Advanced Driving Assistance Systems) remains a major challenging in inner-city scenarios motivated by the high complexity in scene layout with unmarked or weakly marked roads and poor lightning conditions. This paper introduces a novel method that creates a classifier based on a set of probability distribution. The classifier, created using a Joint Boosting algorithm, aims at detecting semantic information in roads. This approach is composed of a set of parallel processes to calculate the superpixel using the Watershed Transform method and the construction of feature maps based on Textons and Disptons. As a result, a set of probability distribution is generated. It will be used as an input to model the weak classifier by our Joint Boosting algorithm. The experimental results using the Urban-Kitty benchmark are comparable to the state-of-the-art approaches and can largely improve the effectiveness of the detection in several conditions.

Index Terms—Road Detection, Computer Vision, Joint Boosting, Texton Map, Dispton Map, Watershed Transform.

I. INTRODUCTION

Autonomous Navigation for urban environments has been receiving considerable attention from the robotic community over the past few years, motivating researchers to propose approaches towards the detection of roads in challenging inner-city environments. The inner-city road detection usually helps an Intelligent Vehicle System to get a better understanding of the environment improving interlinked or dependent tasks such as path planning [1], road following [2], and visual servoing [3].

Applications for road detection using camera sensors must deal with a set of problems such as: (i) continuously changing backgrounds in different environments (inner-city, highway, off-road), (ii) different road types (shape and color), (iii) the presence of different objects (signs, vehicles, pedestrian) and (iv) differences in imaging conditions (variation of illumination and weather conditions).

Many researchers have addressed this problem using monocular or stereo vision [4][5]. Approaches using monocular vision aim at detecting lane marking [6], appearance cues [7] or the 3D aspect by using prior knowledge about the environment as an extra source of information [8]. The detection methods using lane marking approaches may fail in unmarked roads, and some approaches overcome

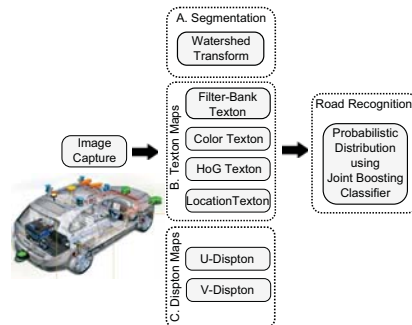


Fig. 1. Solution block diagram.

this assumption discriminating the overall road area with some appearance cues such as color feature [7][9][10], and texture feature [11][12]. Outcomes from the color feature may present poor results when there is high intra-class variability presents in the dynamic nature of the scenes. Texture is scale-dependent and is affected by the strong perspective in road image. Detection using prior knowledge of 3D aspect of the road may fail due to the car’s dynamic and road’s imperfections. In the case of stereo vision, the 3D information of the environment is typically used to estimate free spaces and obstacles, with specific techniques like the V-Disparity Map [13][14]. However, the detection of road in urban environments must deal with different sources of noise that makes it difficult to define the disparity map. Some approaches propose improvements to the model by merging color, texture and adding 3D information [15][16].

This work proposes a novel method for road detection in inner-city, using a set of probabilistic distribution to model the classifier of a Joint Boosting algorithm. Differently from others works, this approach creates a different set of features merging a technique called Dispton, proposed by previous works (3D information) with the Texton (2D texture and color) to compute a set of probabilistic distribution for each superpixel. The probabilistic distribution feature’s descriptor is used to model the weak classifier used in the Joint Boosting algorithm.

The rest of the paper is organized as follows: Section II presents the Image Segmentation, Texton Mapping and Dispton Mapping processes. Section III presents the formalism to model a set of probabilistic distribution using a Joint Boosting algorithm. Section IV presents the results of an experiment using the KITTI benchmark. Finally, Section V presents some conclusions about what we have learned from the study and proposals for future work.

Authors are members of ¹Heudiasyc UMR CNRS 7253 Université de Technologie de Compiègne, ²Universidade Estadual de Campinas (UNICAMP). Giovani B. Vitor holds a Ph.D. scholarship from CNPQ. Contact authors giovani.bernardes-vitor@hds.utc.fr

II. IMAGE PROCESSING

In this section is presented a description of three parallel processes namely Image Segmentation, Texton Map and Dispton Map. As can be seen in Figure 1, the resulting output of these processes will be used as a source for generating the set of probability distribution in a second step.

A. Image Segmentation

Among various approaches used to produce superpixels, the most used nowadays in literature is the mean-shift algorithm [17][18]. In this work however, it is explored another methodology based on Watershed Transform that since [19] has been applied to road detection. The combination of Watershed Transform with other filters has presented encouraging results as shown in [20].

In order to obtain a reasonable flexibility to determine the segmentation level, we have used the same approach described in [20]. Three pre-filters were added: (i) the *Morphological Gradient Adjusted* (MG_{adj}), (ii) the *AreaClose* and (iii) the *Hmin*. Filter (i) is applied to obtain the high frequency image, where an adjustment is done to improve the low-contrast of high frequency in shadow areas [21]. The formal definition can be seen in (1):

$$MG_{Adj} = \begin{cases} c[(f \oplus g_e) - (f \ominus g_i)]^\gamma & , \text{if } \{\forall x | f(x) < \rho\} \\ (f \oplus g_e) - (f \ominus g_i) & , \text{otherwise} \end{cases} \quad (1)$$

In equation (1) f is the image function, g_e and g_i are structuring elements centered at the origin, the operators \oplus and \ominus are respectively dilation and erosion. The non-linear transformation at low-contrast of high frequency has the factor γ setted to 0.45 and ρ is the threshold. The constant of normalization c is defined by the equation (2).

$$c = \frac{\max((f \oplus g_e) - (f \ominus g_i))}{\max((f \oplus g_e) - (f \ominus g_i))^\gamma} \quad (2)$$

The goal of the filters (ii) and (iii) is to control the segmentation level of the Watershed Transform by acting on the regional minimum of GM_{Adj} . The procedure is identical to [20], where the parameter λ of *AreaClose* determines the area of the regional minimum to be cut out, and the parameter h of *Hmin* determines the height from the regional minimum to be also cut out. The final outcome applied in a sample image can be observed in Figure 2. Notice that the classification's procedure is performed making a probability distribution for each super-pixel. Therefore, the analysis of these parameters is done to understand their sensibility in the final result of the segmentation.

B. Texton Map

In the last decade, *Textons* have been proven effective for generic feature representation of object [18][22], where a class demands different appearances to have a compact representation maintaining their efficiency. Thereby, the methodology applied for this block is to learn a dictionary of Textons using a textonization technique [23], which allows to perform a dense-texture-based feature extraction for all pixels. The

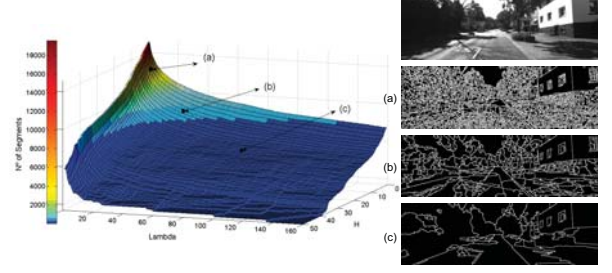


Fig. 2. Example of an influence surface for the parameters λ and h on the number of segments in an given image. In (a) $\lambda_1 = 5$, $h_1 = 2$ and 4090 segments; (b) $\lambda_2 = 30$, $h_2 = 5$ and 427 segments; (c) $\lambda_3 = 80$, $h_3 = 15$ and 73 segments.

process of *textonization* generates the *Texton Map*, having the same size of the image. The Textons contained in the dictionary have their value associated with all pixels in this map. It can be seen as a pre-classification or a transformation from feature's space to the texton's space. Thus, this process is done by applying the K-Means algorithm on a feature's space. Denoting a dictionary as D , each texton's element $x_j \in D = \{x_1, x_2, \dots, x_K\}$ represents a cluster generated by the algorithm, employing the Euclidian-distance as a metric. Finally, it is obtained the *Texton Map* $T \in \mathbb{N}^2$ with the pixel i having value $x_j \in D$.

In this work, the *textonization* executed in various feature's spaces, as it can be seen in [18]. The set includes 17-dimensional filter bank, 3-dimensional CIELAB color, 81-dimensional histograms of oriented gradient [24] and 2-dimensional normalized pixel location. All feature's descriptor are whitened (to give zero mean and unit covariance) to learn the dictionaries of textons in which their configuration were assigned to $D^b = 400$ clusters, $D^c = 128$ clusters, $D^g = 150$ clusters and $D^l = 144$ clusters, respectively. The output result for this module can be seen in Figure 3.

C. Dispton Map

Based on the approach explained in section II-B, *Texton Maps* are able to discriminate between class of similar textures. However this technique lacks spacial information. This section presents an approach to build two additional dictionaries over 3D information from Stereo Vision. This method is called *Dispton Map* and it aims at creating meaningful clusters based on the Disparity Map, denoted by I_Δ . Attempting to have the same functional advantage provided by the usage of U-Disparity and V-Disparity algorithm to filter and extract the navigable area and obstacles in literature [13][25], this work addresses another way to embed these information in a dictionary of Dispton, generating the *Dispton Maps* from I_Δ .

Firstly, the technique consists in putting in evidence the peaks of the U-V Disparity maps, which concentrates the relevant information to start the process of *Disptonization*. Defining U-Disparity as $I_u\Delta$ and V-Disparity as $I_v\Delta$, they are obtained from a histogram for each column, $I_u\Delta = \{hist(I_\Delta(:, u)) | \forall u \in \{0..width - 1\}\}$ and for each row,

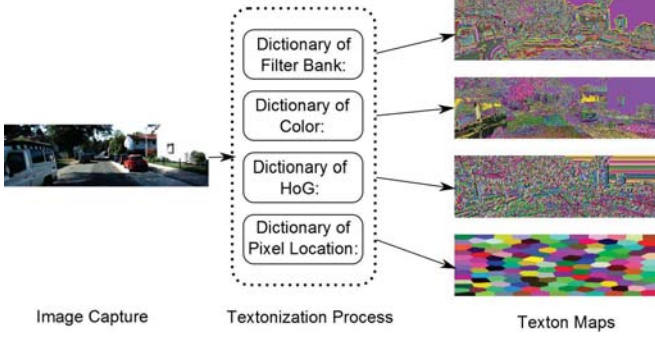


Fig. 3. The texton maps resulting from the textonization process using different features.

$I_v \Delta = \{hist(I\Delta(v, :)) | \forall v \in \{0..height - 1\}\}$. Like Watershed Transform, they can be seen as surfaces to apply the $Hmin$ and filter the regional minimum (in this case considered as noise) of these surfaces. The result maps, denoted $I_u^h \Delta$ and $I_v^h \Delta$ are calculated by the binarization. A Hough Transform is executed to detect line segments, characterized by l^u and l^v .

After that, to build the dictionary of $U-Dispton$ (D^u) is applied the clusterization where the points from each line segment l_j^u , supplies seeds to perform the clusterization of dispton's element $j \in \{1..NumberOfLines\}$. In equation (3), the clusterization process of a line segment l_j^u , denoted by $\Lambda(I_u^h \Delta)(l_j^u)$, is given by:

$$\Lambda(I_u^h \Delta)(l_j^u) = \begin{cases} C_j^\eta & , \text{if } \{x\} \in N_x(I_u^h \Delta) \neq 0 \\ 0 & , \text{otherwise} \end{cases} \quad (3)$$

Where the cluster C_j^η is defined in equation (4). Note that the variable (η) can be either u or v .

$$C_j^\eta = \{j | x_1 W + x_2 \in \{l_j^\eta\} \subset N_x(I_u^h \Delta)\} \quad (4)$$

Here, W is the number of image columns, and x_1 and x_2 are the row and column coordinates respectively, and the term $N_x(\cdot)$ represents the neighbors of the x element. There are two additional clusters given by $C_1^u = \{I_u \Delta(v, :)|v \in \{0, 1, \dots, \tau\}\}$ and $C_2^u = (I_u \Delta \cap C_1^u)'$, where $(\cdot)'$ represents its complement. Therefore, the D^u is given by the union of all clusters (5):

$$D^u = \left\{ \bigcup_{\forall j \in l^u} \Lambda(I_u^h \Delta)(l_j^u) \cup C_1^u \cup C_2^u \right\} \quad (5)$$

Finally, with the $U-Dispton$ dictionary, it is possible to obtain the $U-Dispton$ map (Dm_u) by the following equation (6):

$$Dm_\eta = \{D^\eta(I\Delta(x)) | \forall x; I\Delta(x) \neq 0\} \quad (6)$$

In order to build the $V-Dispton$ dictionary (D^v), the clusters are created separately. Following Hu and Uchimura [26], a road is modeled as a plane so that it can be represented by straight slope line segments in the V-disparity map. In this

sense, the goal of the first cluster is to curve fitting these line segments to represent the surface of navigable area. From l^v set, a subset l^{vs} is obtained filtering out the line segments with a given vertical orientation (7):

$$l^{vs} = \{l_i^v | \forall l_i^v; Ang(l_i^v) < 90^\circ - \psi\} \quad (7)$$

The $Ang(\cdot)$ represents the angle of inclination with the reference defined on the bottom-left image and ψ a parameter of input. Thus, the road surface can be formed by a succession of plane's parts, being projected as a piecewise linear curve [27]. In order to connect the line segments that represent the surface, the algorithm sorts the l^{vs} set based on the distance from the line segment to the reference. Starting from the first line segment l_0^{vs} to the last one l_n^{vs} , the constraints that define whether two line segments can be connected, are given by the follow equation (8), which l^{vc} is the set of connected line segments.

$$l^{vc} = \min(dist(l_i^{vs}, l_j^{vs})) \begin{cases} \forall l_j^{vs} \in \{l^{vs} > l_i^{vs}\} \text{ and} \\ \text{if } l_j^{vs} \subset AreaSupport(l_i^{s-}, l_i^{s+}) \end{cases} \quad (8)$$

Where the function $dist(\cdot)$ between two line segments is calculated considering the Euclidian distance from the nearest points of the current line segments, limited by a given maximum distance ε between them. The function $AreaSupport(\cdot)$ delimits the search area by two line segments as seen in (9).

$$AreaSupport(l_{s1}, l_{s2}) = \begin{cases} 1 & , \text{if } right(l, l_{s1}) \text{ and } left(l, l_{s2}) \\ 0 & , \text{otherwise} \end{cases} \quad (9)$$

This area is defined by a translation from l_i^{vs} given by the σ parameter, then $l_i^{s-} = l_i^{vs} - \sigma$ and $l_i^{s+} = l_i^{vs} + \sigma$. The other two functions in this equation return true case when the line segment is on right and left of the reference lines. As a result, the cluster C_1^v is obtained applying the equation (3) on the l^{vc} set ($\Lambda(I_v^h \Delta)(l_j^{vc})$) with one more constraint, where all pixels cannot cross out the line (l_{lim1}) formed by the first and last points of l^{vc} (added a small shift constraint). In addition, the second cluster is generated taking those pixels which cross out the first one and is restricted to another shifted line $l_{lim2} = l_{lim1} + \sigma_2$, resulting the equation (10):

$$C_2^v = \{\Lambda(I_v^h \Delta)(l_j^{vc}) | \text{if } AreaSupport(l_{lim1}, l_{lim2})\} \quad (10)$$

To finish the $Disptonization$, the last two clusters are generated by $C_3^v = \{I_v \Delta(:, u) | \forall u \in \{1, \dots, \tau\}\}$, where τ defines the max disparity to be considered as background or infinity, and $C_4^v = \{(I_v \Delta \cap (C_1^v \cup C_2^v \cup C_3^v))'\}$. The $V-Dispton$ dictionary (D^v) is defined as:

$$D^v = \{C_1^v \cup C_2^v \cup C_3^v \cup C_4^v\} \quad (11)$$

And the generation of $V-Diston$ map is obtained by the equation (6). Algorithm 1 summarizes the $Disptonization$ process.

Algorithm 1 Disptonization algorithm:

- 1: Process $I_u\Delta$ and $I_v\Delta$ from $I\Delta$;
 - 2: Apply the *Hmin* filter on $I_u\Delta$ and $I_v\Delta$;
 - 3: Binarize and obtain the line segments by *Hough Transf.* for l^u and l^v ;
 - 4: Determine the *U-Dispton* dictionary D^u by eq. (5):
 - Apply the clusterization on l^u , eq. (3);
 - 5: Determine the *V-Dispton* dictionary D^v by eq. (11):
 - Filter out the vertical lines to take l^{vs} , eq. (7);
 - Find out the connected lines l^{vc} , eq. (8);
 - Define the clusterization to $C_1^v, C_2^v, C_3^v, C_4^v$;
 - 6: Generate the *UV-Dispton map* by eq. (6)
-

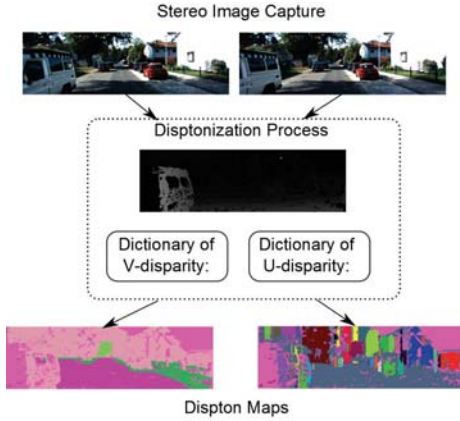


Fig. 4. The Dispton maps obtained from the disptonization process using the Disparity Map.

Note that the *V-Dispton map* has 4-dimensional clusters and the *U-Dispton map* has N-dimensional clusters. Intuitively, they aim at storing important information such as navigable area, sidewalk, obstacles and background. The N-dimensional structure from *U-Dispton map* dynamically retrieve the representation of all possible different obstacles in the scene, as can be seen in Figure 4. With the *Texton maps* and *Dispton maps*, the next section explains how they are combined with the superpixel to perform the classification.

III. ROAD RECOGNITION

This section presents an approach to represent and compute the classification of the road class, as shown in previous sections, where the road recognition can be executed using an adapted version of the Joint Boosting algorithm [28]. In fact, the algorithm is inspired by the *TextonBoost* approach [29], which iteratively builds a *strong classifier* as a sum of *weak classifiers*, simultaneously selecting discriminative features. We have improved the representation of weak classifiers using a specific shape filter. Thus, the novelty is to build a set of probability distribution of the Texton and Dispton maps from the decomposition of the scene into a number of semantically consistent regions, supplied by the segmentation result shown in section II-A, to model the *weak classifier*.

The process could be formally explained taking into account the maps $\{M^f : f \in \{F\}\}$ where $F =$

$\{b, c, g, l, v, u\}$ is the set of Textons and Disptons. Each element i in the map $M^f \in \mathbb{N}^2$ belongs to exactly one region, identified by its region-correspondence variable $S_r \in \{1, \dots, NumSegments\}$. The r -th region is then simply the set of elements i_r whose region-correspondence variable equals r , i.e., $i_r = \{i : M_i^f = r\}$. We use $X_i^f = \{X_1^f, X_2^f, \dots, X_N^f\}$ to denote the set of random variables corresponding to the f -th value of i -th element into M^f . Any possible assignment to the random variables $X_i^f = x_j^f$ takes values from $j \in D^f$, which D^f is defined by the constructed dictionary for each $f \in F$ generated in the sections II-B and II-C, .

The probability of the X_i^f if given by $P(X_i^f = x_j^f)$, and the associated set of probability distribution under the S_r is denoted by $P(X_r)$, as can be seen in the equation (12):

$$P(X_r) = \left\{ \bigcup_{f \in F} \left\{ \frac{1}{Z} \sum_{i_r} P(x_j^f) \right\} \mid \forall j \in D^f \right\} \quad (12)$$

In Equation (12), Z is a normalization factor for each probability distribution set. Using the probability representation of Textons and Disptons, the *weak classifiers* are modeled as comparisons of this probability distribution to a decision stump based on a threshold, where each *weak classifier* is *shared* between a set of classes, allowing a single probability to help classify several classes at once. They are defined by wc containing 2-tuples $[x_{rand}, P(x_{random})]$, where the first component represents a random possible assignment $\{x_{rand} : x_j^f \in D^f\}$ and its value of probability randomly defined. To express how well the probability distribution of $P(X_r)$ at a given x_j^f matches the weak classifier, a comparison response is given by equation (13):

$$d(wc, S_r) = 1 - \sqrt{[P(x_{rand}) - P(X_r = x_j^f)]^2} \quad (13)$$

Thereby, the Joint Boosting algorithm is an additive model of the form $H(c_l) = \sum_{m=1}^M h_m(c_l)$, that sum the classification confidence of M joint weak classifiers. In this case, $H(c_l)$ represents the strong learned classifier and the weak classifiers are extended to discriminate the share between classes. Therefore, each weak-learner is modeled as a decision stump of the form:

$$h(c_l) = \begin{cases} a\delta(d(wc, S_r) > \theta) + b & , \text{if } \{c_l \in L\} \\ \kappa_{c_l} & , \text{otherwise} \end{cases} \quad (14)$$

Where $\delta(\cdot)$ is a 0-1 indicator function. The share is given by those classes ($c_l \in L$), where the weak learner gives $h(c_l) \in \{a+b, b\}$ depending on the comparison of $d(wc, S_r)$ to a threshold θ . The constant κ_{c_l} ensures asymmetrical sets of positive and negative training examples for those classes that do not share the feature ($c_l \notin L$). Thus, the resulting classification output is defined by the probability conversion given by(15):

$$P = \frac{1}{Z} \exp^{-H(c_l)} \quad (15)$$

Here, the Z represents the normalization factor into the classes $c_l \in L$.

IV. EXPERIMENTAL RESULTS

In this subsection we present the results of our experiments using real driving situations. It is used the Urban Kitti-road dataset¹, which consists of $\simeq 600$ frames (375x1242 px) recording from five different days and containing relatively low traffic density [30]. The data are categorized in three sets having each one a subset of training and test images, representing a typical road scene in inner-city. The first set called the UU is formed by images taken from an urban unmarked area and has 98 images for training and 100 images for testing. The second set called the UM is formed by images taken from an urban marked two-way road and has 95 images for training and 96 images for testing. Finally, the final set called UMM is formed by images taken from an urban marked multi-lane road and has 96 images for training and 94 images for testing. The experiments in this work use the training set in the perspective space to learn the classifier and the metric space to calculate the complete evaluation of this approach. The evaluation process is done on the metric space in order to capture the fact that vehicle control happens in the 2D environment. Further, the evaluation in perspective space is biased by the fact that the pixel's value in near range is more homogenous and covers a larger area of the evaluated perspective pixels [30].

The learning process were executed separately for each category. Thus, a sample set was built for each one, extracting $\simeq 12.8E + 4$ samples from the UU image training set, $\simeq 12.6E + 4$ samples from the UM image training set and $\simeq 11.7E + 4$ samples from the UMM image training set. Table I shows the results of the quantitative evaluation of our approach applied in the test set. For comparison purposes, it also presents the baseline provided as a lower bound, by averaging all ground truth road maps from the present testing set, and also the results of the proposed road detection using Artificial Neural Network (ANN) [20]. As can be seen for the UU category, our approach reached an improvement of 26.12% if compared to ANN and 10.80% to Baseline, using the testing set on the metric space. With respect to the UM category, our approach reached a level of correctness of 87.60%, reaching a gain of 24.96% compared to ANN and overcoming the baseline approach in 5.07%. Using the UMM category, which is less complex if compared to others, we can highlight our approach, overcoming the ANN approach with 9.03% and 13.95% compared to the baseline. The qualitative result for this challenging dataset can be seen in Figure 5, presenting a classified image using the perspective space. The same images using the metric space can be seen in Figure 6.

To conclude the evaluation process, Table II presents the final results merging all categories. Our approach presents 87.21% of correctness for challenging urban Kitti-road benchmark. According to our experiments we believe that

¹http://www.cvlibs.net/datasets/kitti/eval_road.php

TABLE I
RESULTS [%] OF PIXEL-BASED FOR THE ALL CATEGORIES ON THE METRIC SPACE EVALUATION.

Urban Unmarked (UU)						
	F_{max}	AP	Prec.	Recall	FPR	FNR
Baseline [30]	69.49	73.84	65.73	73.70	12.78	26.30
ANN [20]	54.17	36.86	39.50	86.19	43.92	13.81
Our	80.29	69.05	85.58	75.61	4.24	24.39
Urban Marked (UM)						
	F_{max}	AP	Prec.	Recall	FPR	FNR
Baseline [30]	82.53	85.59	79.24	86.11	10.41	13.89
ANN [20]	62.64	46.80	50.18	83.34	38.21	16.66
Our	87.60	76.04	85.92	89.36	6.76	10.64
Urban Marked Multi-Lane (UMM)						
	F_{max}	AP	Prec.	Recall	FPR	FNR
Baseline [30]	76.17	78.42	65.02	91.95	57.89	8.05
ANN [20]	81.09	68.93	70.43	95.56	46.94	4.44
Our	90.12	85.04	88.15	92.12	14.50	7.82



Fig. 5. The resulting output of our approach in perspective space. The first, second and third rows show the UU, UM and UMM category image respectively.

our approach improved the detection of roads in scenarios with high complexity, significantly outperforming the baseline.

TABLE II
RESULTS [%] OF PIXEL-BASED FOR COMPLETE URBAN ROAD AREA EVALUATION.

	F_{max}	AP	Prec.	Recall	FPR	FNR
Baseline [30]	75.61	79.72	68.93	83.73	21.73	16.27
ANN [20]	68.12	51.52	54.85	89.85	42.59	10.15
Our	87.21	77.79	86.96	87.47	7.55	12.53

V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have proposed an approach for road recognition for inner-city based on appearance, shape and spacial model learned from training data. The main contribution of this work is the creation of a probabilistic distribution based on Texton and Dispton maps to model weak classifiers used in a Joint Boosting classifier.

Experiments conducted on real driving situations demonstrate the qualitative and quantitative evaluation of our algorithm to detect road despite the presence of shadows and other objects in the scene, inherent from the complexity of inner-city environments. The result also provides the benefits of our approach over existing methods.

We are still working on improvements to reduce the processing time using the GPU architecture, and also working

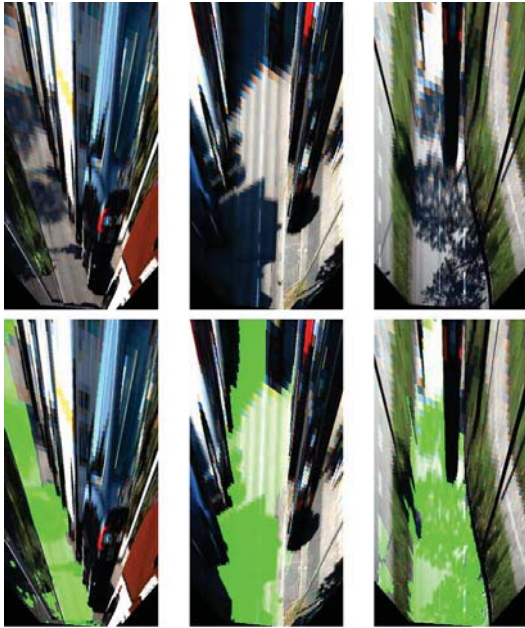


Fig. 6. The resulting output of our approach in metric space (highlighted in green). The first, second and third columns show the UU, UM and UMM category image respectively, from the images of Figure 5.

on Self-Organizing Maps [31], in order to better discriminate the road pattern and extending the recognition for different classes such as vehicles, builds, sidewalks, etc. This approach could improve the prediction of the classifier. The complete application will be embedded in a real car-like robot, sponsored by the project ROBOTEX, from the Heudiasyc laboratory, to perform autonomous driving in urban environments.

REFERENCES

- [1] N. Rawashdeh and H. Jasim, "Mult-sensor input path planning for an autonomous ground vehicle," in *Mechatronics and its Applications (ISMA)*, 2013, pp. 1–6.
- [2] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the carnegie-mellon navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, May 1988.
- [3] P. Karasev, M. Serrano, P. Vela, and A. Tannenbaum, "Depth invariant visual servoing," in *Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 4992–4998.
- [4] J. Alvarez, A. Lopez, and R. Baldrich, "Illuminant-invariant model-based road segmentation," in *Intelligent Vehicles Symposium, 2008 IEEE*, June 2008, pp. 1175–1180.
- [5] F. Dornaika, J. Alvarez, A. Sappa, and A. Lopez, "A new framework for stereo sensor pose through road segmentation and registration," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 954–966, Dec. 2011.
- [6] M. Felisa and P. Zani, "Robust monocular lane detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, June 2010, pp. 591–596.
- [7] C. Tan, T. Hong, T. Chang, and M. Shneier, "Color model-based real-time learning for road following," in *Intelligent Transportation Systems Conference, ITSC*, 2006, pp. 939–944.
- [8] P. Coulombeau and C. Lurgeau, "Vehicle yaw, pitch, roll and 3d lane shape recovery by vision," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, June 2002, pp. 619–625.
- [9] J. Alvarez, T. Gevers, and A. Lopez, "Learning photometric invariance from diversified color model ensembles," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 565–572.
- [10] J. Alvarez and A. Lopez, "Road detection based on illuminant invariance," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 184–193, 2011.
- [11] C. Rasmussen, "Grouping dominant orientations for ill-structured road following," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, 2004, pp. 1–470–1–477.
- [12] H. Kong, J.-Y. Audibert, and J. Ponce, "General road detection from a single image," *Image Processing, IEEE Transactions on*, vol. 19, no. 8, pp. 2211–2220, 2010.
- [13] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, vol. 2, 2002, pp. 646–651.
- [14] A. Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," in *Proceedings of the International IEEE Workshop on Machine Vision for Intelligent Vehicles*, 2005, pp. 1–8.
- [15] N. Soquet, D. Aubert, and N. Hautiere, "Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, 2007, pp. 160–165.
- [16] P. Sturgess, K. Alahari, L. Ladicky, and P. H. S. Torr, "Combining appearance and structure from motion features for road scene understanding," in *BMVC. British Machine Vision Association*, 2009.
- [17] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Transactions on*, vol. 24, no. 5, pp. 603–619, 2002.
- [18] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical crfs for object class image segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*, 2009, pp. 739–746.
- [19] S. Beucher and M. Bilodeau, "Road segmentation and obstacle detection by a fast watershed transformation," in *Intelligent Vehicles '94 Symposium, Proceedings of the*, 1994, pp. 296–301.
- [20] G. B. Vitor, D. A. Lima, A. C. Victorino, and J. V. Ferreira, "A 2d/3d vision based approach applied to road detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 952–957.
- [21] D. A. Lima, G. B. Vitor, A. C. Victorino, and J. V. Ferreira, "A disparity map refinement to enhance weakly-textured urban environment data," in *International Conference on Advanced Robotics (ICAR), 2013 IEEE*, 2013.
- [22] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *CoRR*, vol. abs/1210.5644, 2012.
- [23] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Texonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.
- [24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *In CVPR*, 2005, pp. 886–893.
- [25] N. Soquet, M. Perrollaz, R. Labayrade, and D. Aubert, "Free Space Estimation for Autonomous Navigation," in *5th International Conference on Computer Vision Systems, Bielefeld, Allemagne*, 2007. [Online]. Available: <http://hal.inria.fr/hal-00780658>
- [26] Z. Hu and K. Uchimura, "U-v-disparity: an efficient algorithm for stereo vision based scene analysis," in *Intelligent Vehicles Symposium (IV), IEEE*, 2005, pp. 48–54.
- [27] N. Hautiere, R. Labayrade, M. Perrollaz, and D. Aubert, "Road scene analysis by stereovision: a robust and quasi-dense approach," in *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*, 2006, pp. 1–6.
- [28] A. Torralba, K. Murphy, and W. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2004, pp. II–762–II–769.
- [29] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Texonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," 2007.
- [30] J. Fritsch, T. Kuehnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," in *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.
- [31] T. Kohonen, *Self-organizing maps*, 3rd ed., ser. Springer series in information sciences, 30. Berlin: Springer, December 2001.



Session III

Navigation, Control, Planning

- **Invited Talk: Dimi Apostolopoulos (Carnegie Mellon University, USA)**
Title: Important Considerations in the Design of All-terrain Wheeled Robots
- **Title: A Planner for All Terrain Vehicles on Unknown Rough Terrains based on the MPC Paradigm and D*- like Algorithm**
Authors: A. Tahirovic, M. Brkic, G. Magnani and L. Bascetta
- **Title: An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments**
Author: D. Alves de Lima and A. Correa Victorino
- **Title: Towards Lifelong Learning of Optimal Control for Kinematically Complex Robots**
Author: A. Dettmann¹, M. Langosz², Kai von Szadkowski¹, and S. Bartsch

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICRA 2014





Session III

Invited Talk: **Dimi Apostolopoulos**
(Carnegie Mellon University, USA)

Important Considerations in the Design of All-terrain Wheeled Robots

Abstract: Two considerations stand out when designing the locomotion subsystem of all-terrain robots. The first is the ability to handle a variety of ground conditions. The second is the capacity to react to forces generated from the robot's interaction with its environment. A great deal of research and development has been devoted to designing innovative drive, steering, and suspension assemblies that maximize a robot's inherent mobility and terrainability. In this talk, I will discuss key metrics in the design of wheeled robotic locomotion and share lessons and observations from the field validation of advanced robot prototypes that feature all-wheel-drive and simple planar suspension geometries.

Biography: Dr. Dimitrios (Dimi) Apostolopoulos is a Senior Systems Scientist at the Robotics Institute of Carnegie Mellon University and the National Robotics Engineering Center (NREC). His work focuses on using robotics for discovery and exploration, and as a workforce for hazardous duty and in demanding applications. Since 1998, Dr. Apostolopoulos has led recognized robotic programs for space and defense applications. Robots developed under his programs have made breakthroughs in some of the Earth's most extreme environments. His significant accomplishments include creating, developing, and successfully validating robots for autonomous search of Antarctic meteorites, high-mobility unmanned ground vehicles for US Marine Corps forward missions, systems for profiling and earthmoving in constrained underground environments, and legged robots for exploration of steep terrain and volcanoes.

Dr. Apostolopoulos' research interests are in robotic mobility, mechanisms, robot configuration and design, mechatronics, mechanics, dynamics, and control. Currently, he leads research and development of autonomous robots for mining of noble metals, novel robotic mechanisms for all-terrain mobility, and new methods for design and prototyping of unmanned ground vehicles. Dr. Apostolopoulos holds a Ph.D. in Robotics from Carnegie Mellon University.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



Important Considerations in the Design of All-terrain Wheeled Robots

Dimi Apostolopoulos

Robotics Institute

Carnegie Mellon University, Pittsburgh, PA, USA

Carnegie Mellon



Designing for superior all-terrain mobility



Major design considerations

Locomotion elements (*geometry, elastic vs. rigid*)

Wheel disposition and chassis geometry

Drive scheme (*independent, coordinated*)

Steering scheme (*skid, knuckle, explicit*)

Suspension (*from passive to active, 2D vs. 3D*)

Articulation (*passive vs. active, # & type of axes*)

Control (*speed, traction, slip*)

Actuation (*electric/hydraulic, etc.*)

Sensing (*external, proprioceptive*)

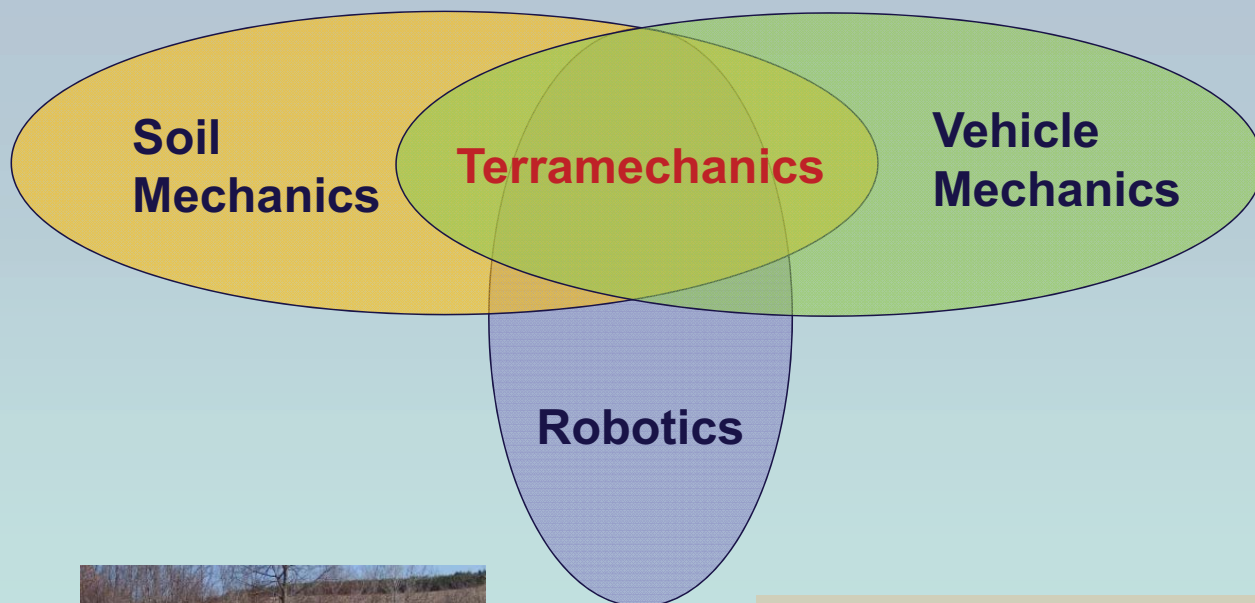
Inherent features for improved terrainability and ease of control



Focus of this presentation

- Terramechanics-based parametric design
- High-performance drive and suspension configurations
- Augmenting mobility with unique design features
- Systems and field experiences

Terramechanics and robotic locomotion design

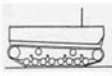
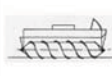

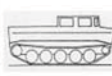



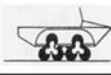

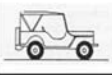


Designing using semi-empirical metrics

Mean Maximum Pressure

$$MMP = \frac{6.895W_{Gross}}{nab^{0.8}d^{0.8}\delta^{0.4}}$$

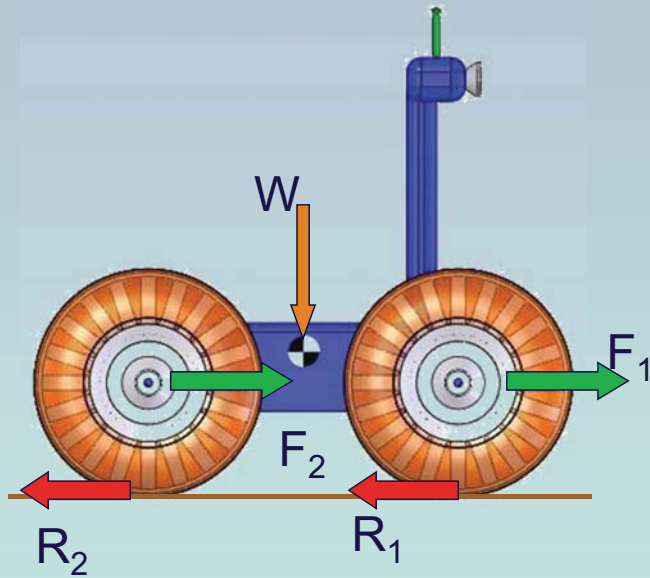
- n: Number of wheels per axle
- b: Average unloaded tire width (in)
- d: Average unloaded tire diameter (in)
- δ: Average tire deflection (in)
- a: Total number of axles
- W: Gross vehicle weight (lb)

Vehicle	Weasel	Marsh Screw	6x6 RST	M116 Husky	Link Track
MMP (kPa)	27	40	46	50	60
					
Vehicle	Spinner w/ CTIS	HMMWV w/ CTIS	Terrastar	Motorbike	Willy's Jeep
MMP (kPa)	< 100	< 150	139	180	226
					

Recommended MMP (kPa)	MMP Levels with skid steer		
	Ideal	Satisfactory	Maximum
Temperate climate, wet, fine-grain soils	120	160	240
Tropical, wet, fine-grain soil	72	112	192

Reference: JD Priddy, WE Willoughby

Study of motion-resisting and tractive forces critical to design



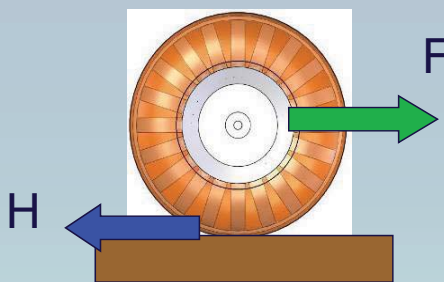
What are drive forces to overcome soil resistances?

Will soil support the loads?

$$F \geq \sum R$$

F: Thrust, R: Resistance, $F >$ soil's capacity to react to loads

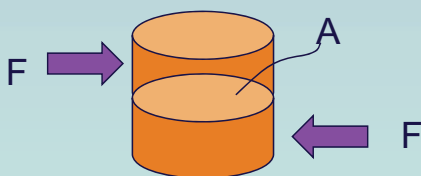
Soil thrust available to sustain locomotion



$$H = \underbrace{(cA + W \tan \phi)}_{\text{Ideal Soil Thrust or Tractive Effort}} \underbrace{\left(1 - e^{-\frac{J}{K}}\right)}_{\text{Slip Term}}$$

J: Shear displacement

K: Modulus of shear deformation

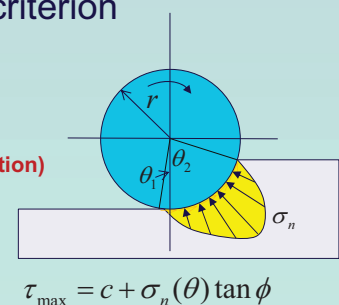


Mohr-Coulomb failure criterion

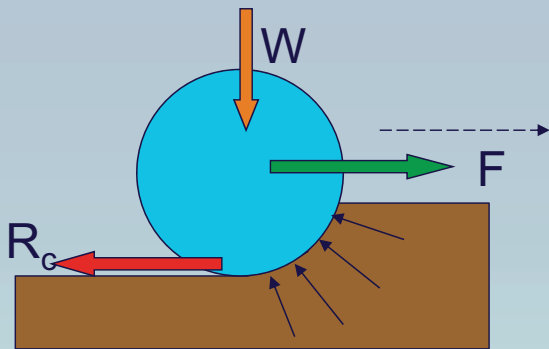
$$H = \tau A$$

$$\tau = c + \sigma \tan \phi$$

τ : Shear stress
 c : Soil cohesion
 σ : Normal stress (compression)
 ϕ : Soil angle of internal shearing resistance (friction)



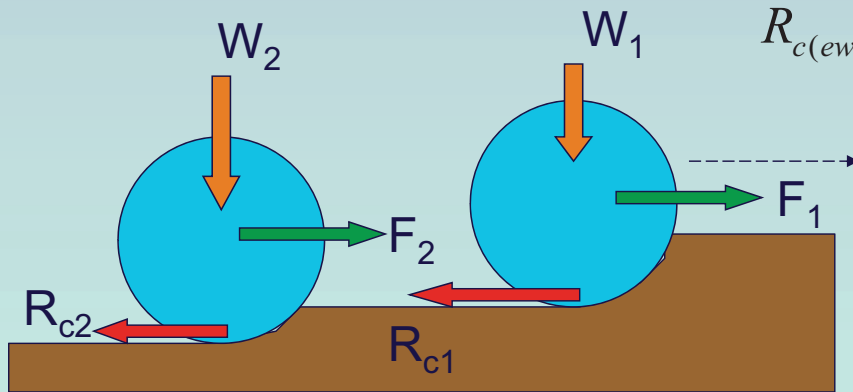
Resistance due to compaction of deformable terrain



Equations and analysis for deformable terrain

$$R_c = b \int_0^{z_{\max}} p dz \Rightarrow$$

$$R_{c(ew)} = b \int_0^{z_{\max}} \left(\frac{k_c}{b} + k_\phi \right) z^n dz$$



Equal to the work per unit length of pressing a block of width b to depth z .

Resistance due to compaction of deformable terrain

Flexible wheel:

$$R_{cew} = \frac{b^{\frac{(n-1)}{n}} p_{gr}^{\frac{(n+1)}{n}}}{(n+1)(k_c + bk_\phi)^{\frac{1}{n}}}$$

Contact/inflation pressure

Rigid wheel:

$$R_{crw} = \frac{(3W / \sqrt{d})^{(2n+2)}}{(3-n)^{\frac{(2n+2)}{(2n+1)}} (n+1)(k_c + bk_\phi)^{\frac{1}{(2n+1)}}}$$

Resistance due to compaction of deformable terrain

Flexible wheel:

$$R_{cew} = \frac{b^{(n-1)} p_{gr}^{(n+1)}}{(n+1)(k_c + bk_\phi)^n}$$

Contact/inflation pressure

Rigid wheel:

$$R_{crw} = \frac{(3W / \sqrt{d})^{(2n+2)}}{(3-n)^{(2n+1)} (n+1) (k_c + bk_\phi)^{(2n+1)}}$$

- Vehicle design parameter
- Soil geophysical parameters
- Force (performance parameter)

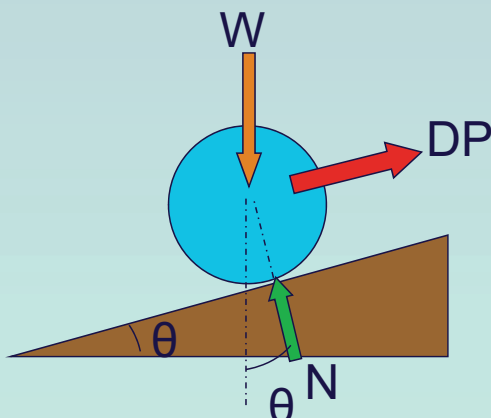
Drawbar pull is a key metric of mobility

Measure of how much tractive force is available after resistances are overcome and before the soil fails in shear.



$$DP = \text{Soil thrust} - \sum \text{Resistance}$$

Estimate max traversable slope (θ_{\max}):



$$\tan(\theta_{\max}) = \frac{DP}{N} \leftarrow N \approx W$$

Maximize slope by maximizing DP:
Increase contact length, decrease width

Terramechanics-based parametric design

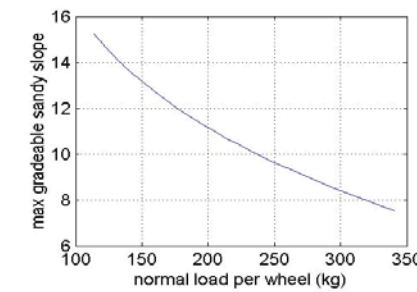
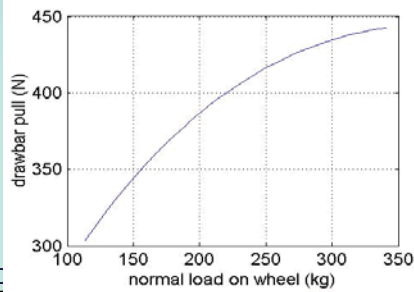
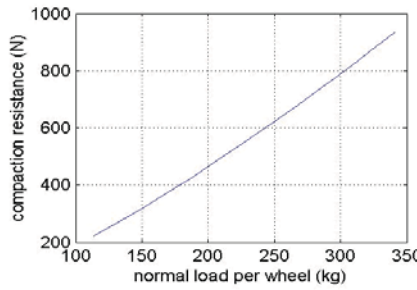
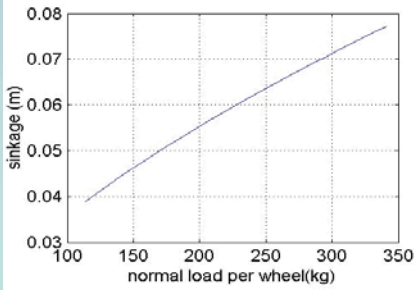
Parametric Equations
Sinkage,
Rolling Resistance,
Drawbar Pull
Torque
Energy/Power Dissipation

$$R_b = \left(\frac{b_w \sin(\alpha + \phi)}{2 \sin \alpha \cos \phi} \right) (2c K_c z_w + \gamma K_\gamma z_w^2) + \frac{\pi \gamma l_r^2 (90 - \phi)}{540} + \frac{\pi c l_r^2}{180} + c l_r^2 \tan \left(45 + \frac{\phi}{2} \right)$$

$$K_c = (N_c - \tan \phi) \cos^2 \phi, K_\gamma = \left(\frac{2N_\gamma}{\tan \phi} + 1 \right) \cos^2 \phi, \alpha = \arccos \left(1 - \frac{2z_w}{d_w} \right), l_r = z_w \tan^2 \left(45 - \frac{\phi}{2} \right)$$

$$R_{crw} = \frac{\left(\frac{3W_w \cos \theta}{\sqrt{d_w}} \right)^{\frac{(2n+2)}{(2n+1)}}}{(3-n)^{\frac{(2n+2)}{(2n+1)}} (n+1) (k_c + b_w k_\phi)^{\frac{(2n+1)}{(2n+1)}}} \Rightarrow$$

$$d_w^{(n+1)} (k_c + b_w k_\phi) = \frac{(3W_w \cos \theta)^{(2n+2)}}{R_{crw}^{(2n+1)} (3-n)^{(2n+2)} (n+1)^{(2n+1)}}$$

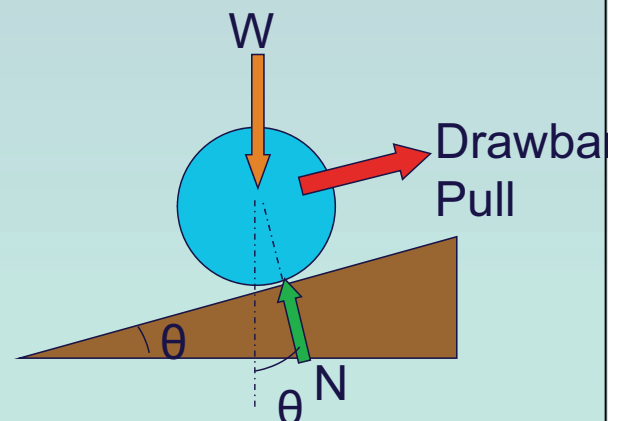
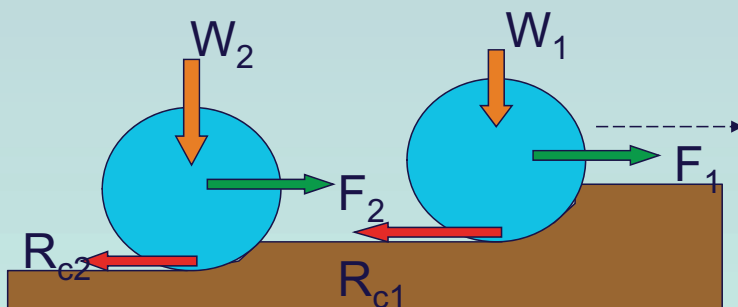
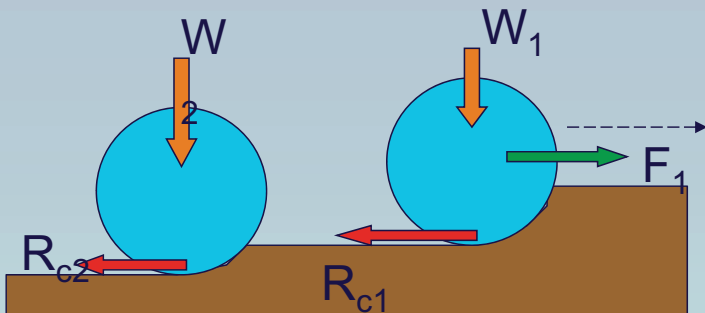


Number of wheels: 6
Wheelbase: 2.5 m x 1.5 m
Roll Radius: 0.25 m
Tire width: 0.20 m

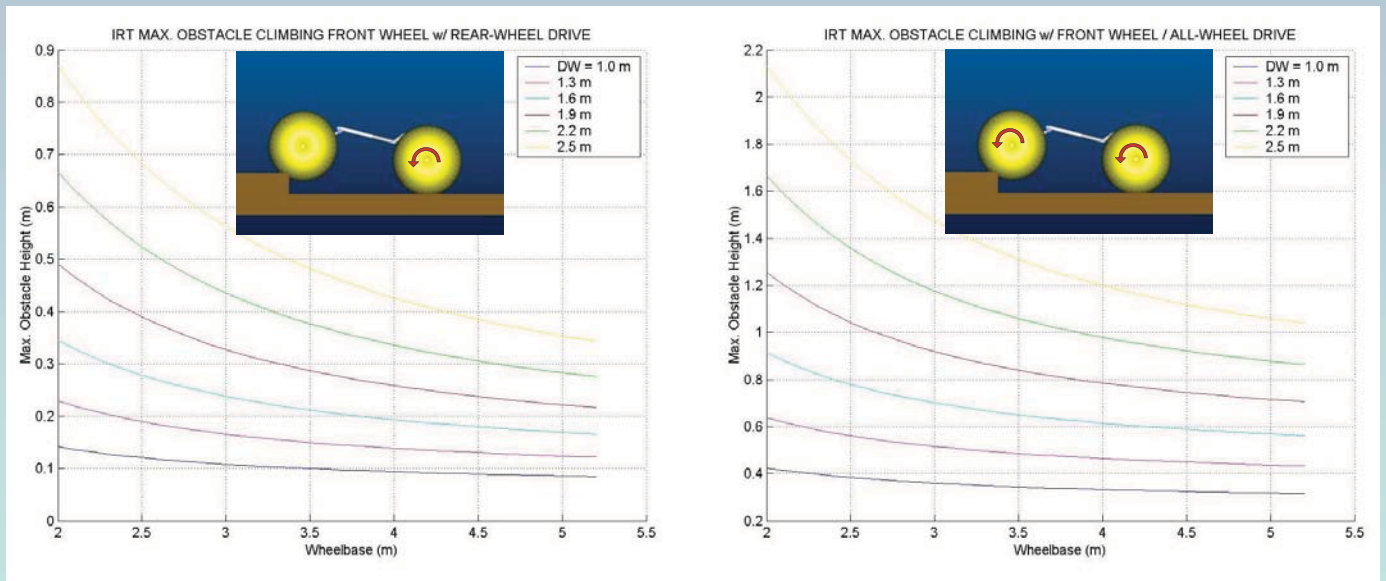


Benefits for all-wheel drive design

Greater pull
Better slope climbing
Better obstacle climbing
Smoother control



All-wheel drive needed for superior terrainability



Configuration of the drive scheme is a key design trade

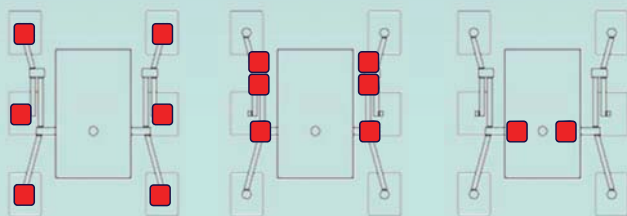
In-hub actuators

One-actuator per side

Distribute torque/power via belts, chains, gears, etc.

One-actuator per vehicle

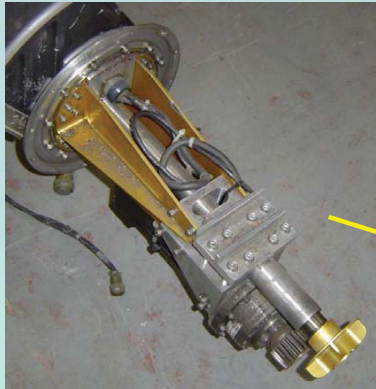
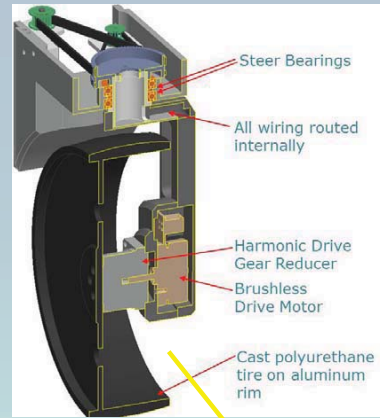
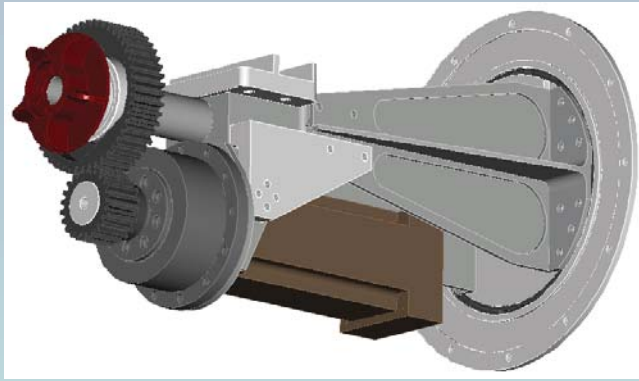
Distribute left/right through mechanical transmissions and clutching



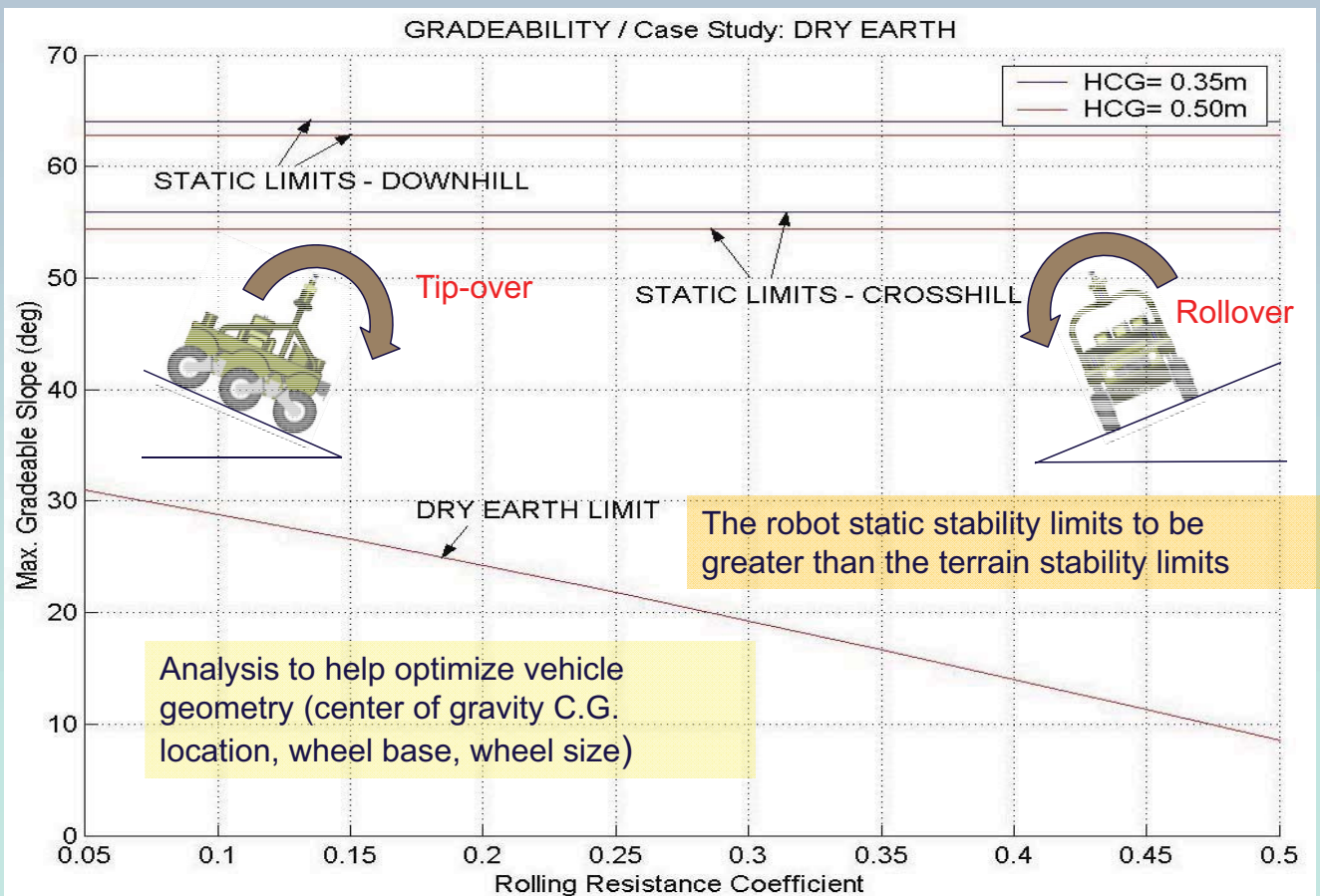
Metric:	Traction motor in wheel	Inboard motor per wheel	Inboard motor per 3 wheels
Energetic efficiency	●	●	○
Torque to weight ratio	○	●	●
Torque to any wheel	●	●	●
Traction control	●	●	○
Control complexity	○	●	●
Mech. complexity	●	●	○
Redundancy	●	●	○
Reliability	●	●	○
Cooling complexity	○	●	●
Volumetric efficiency	●	●	○

Good=● Fair=◐ Poor=○

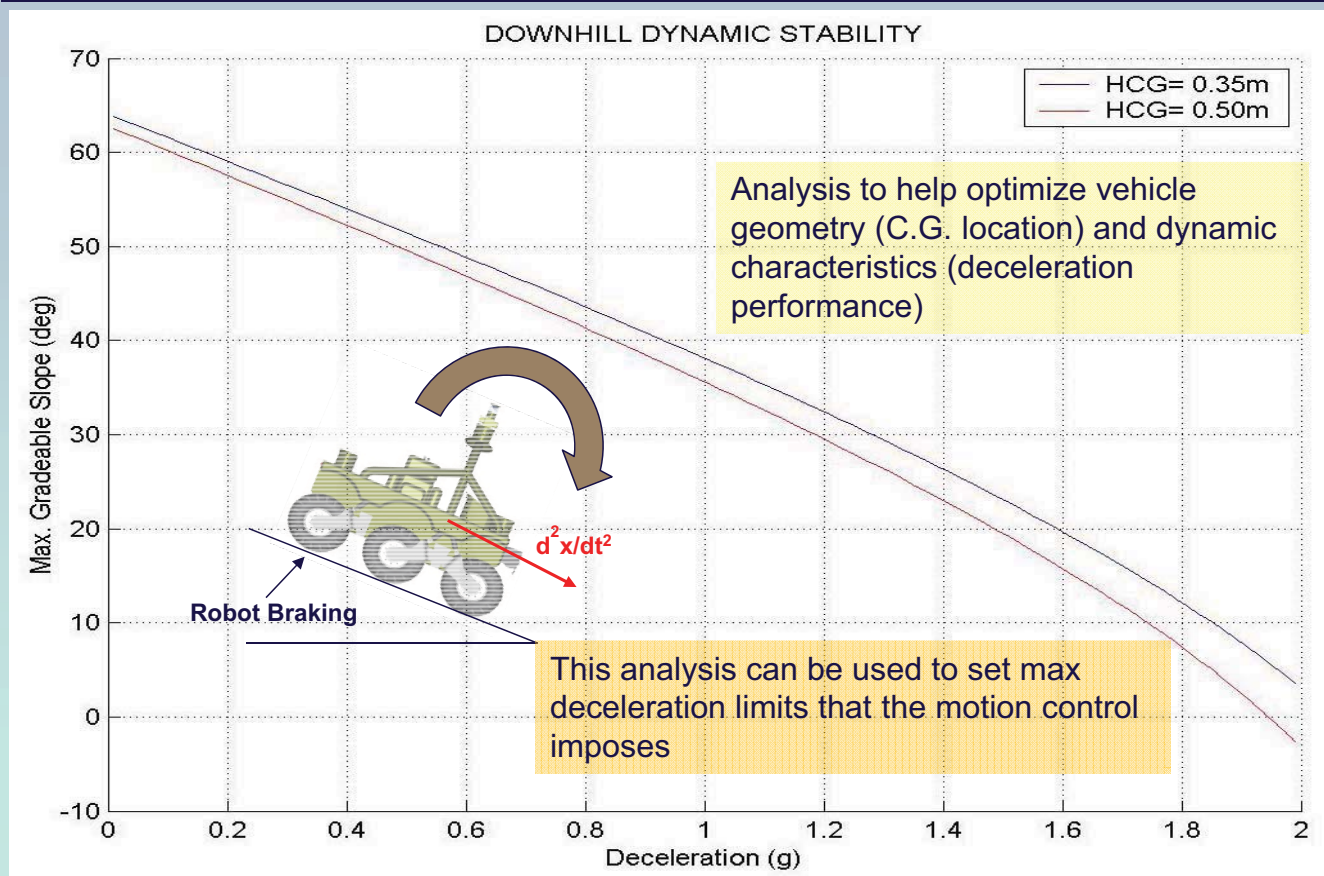
In-wheel electric actuators offer compact and efficient drive solutions



Considerations for slope climbing



Considerations for slope climbing (cont.)



Design for terrain adaptation and motion smoothing

Suspension geometries

Passive

- Dynamic (spring/damper type)
- Geometric, articulated



Passive with active adjustability

- Increase wheel travel
- Control suspension dynamic response
- Stow/deploy suspension
- Adaptively position chassis
- Shift center of mass

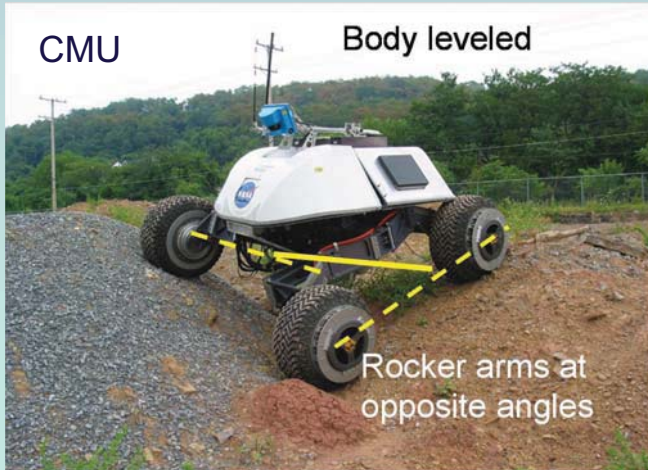
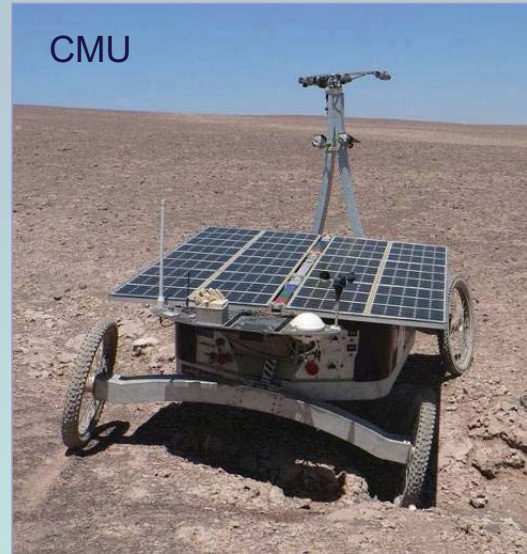


Active

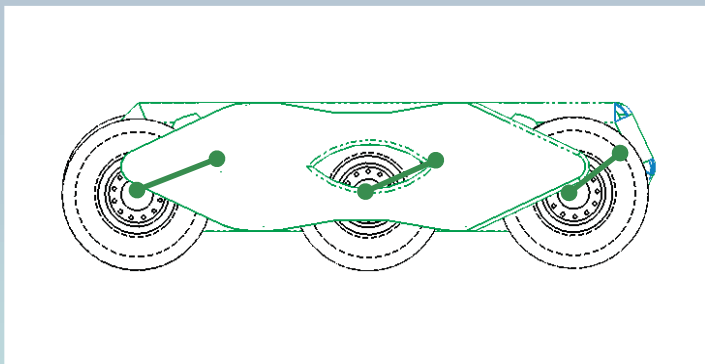
Elastic mobility elements



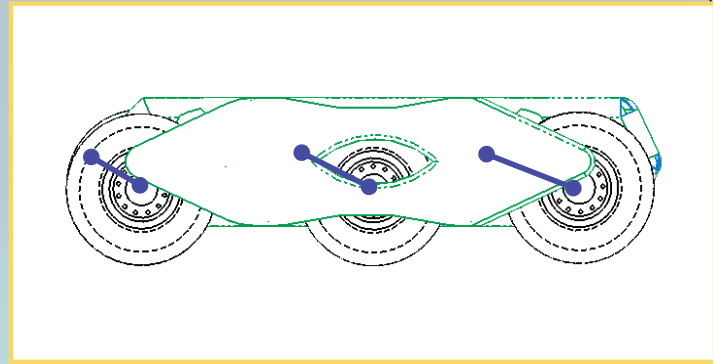
Passive geometric suspensions for balancing ground reaction forces



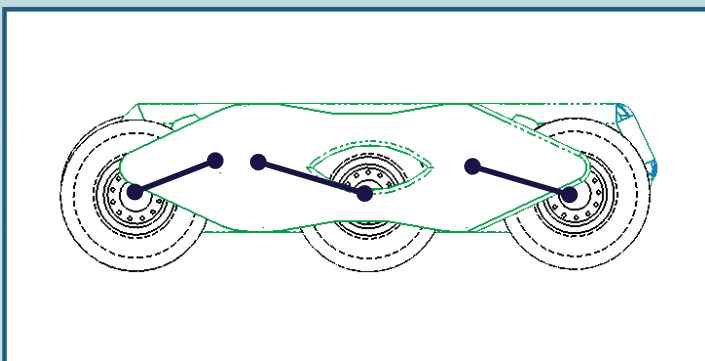
In-lane pivot-arm suspensions for speed and superior terrainability



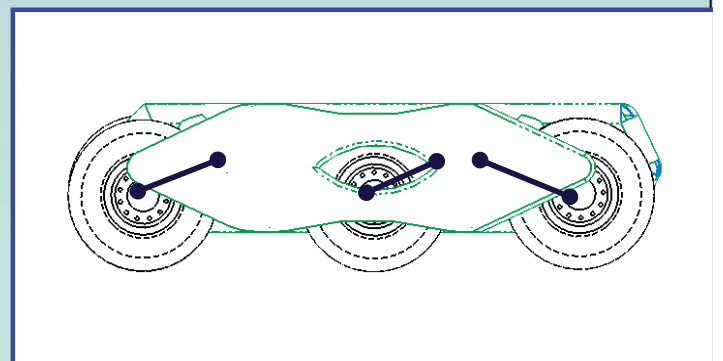
All Leading Arm



All Trailing Arm



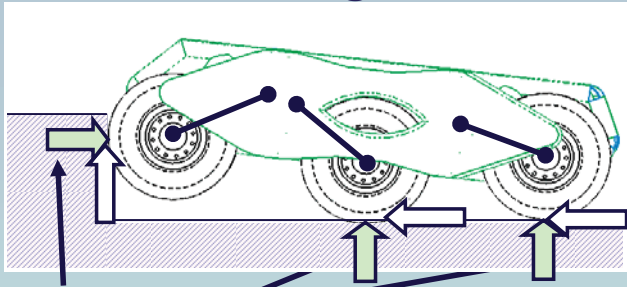
Leading/Trailing/Trailing



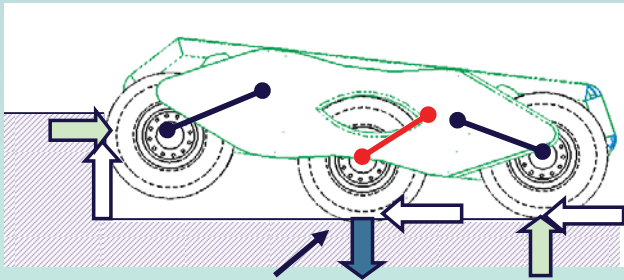
Leading/Leading/Trailing

Design for superior obstacle negotiation (cont.)

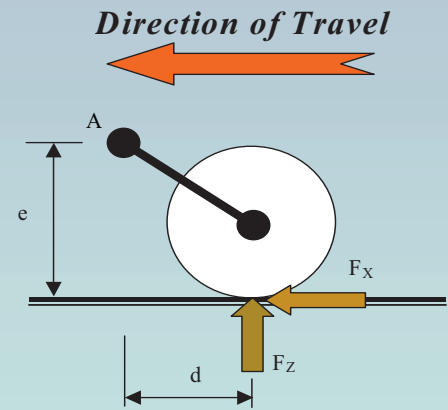
Front wheel climbing



Increased reaction force (provides better traction)



Reaction decreases

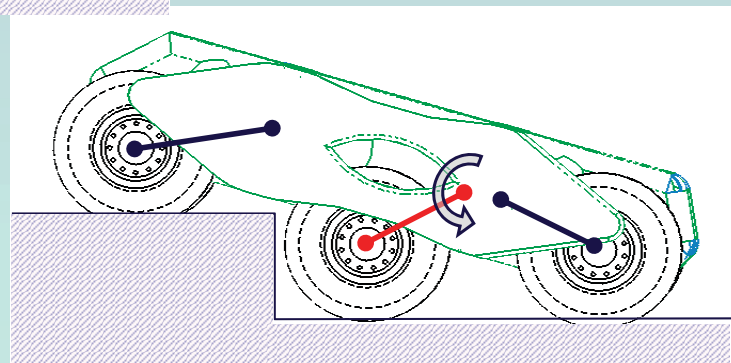
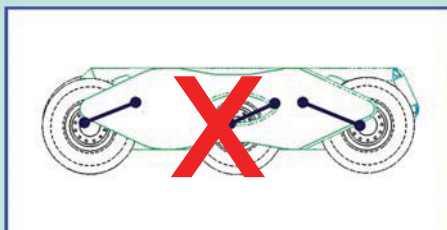
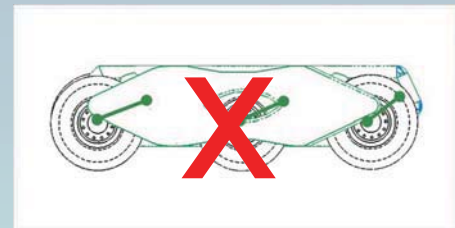
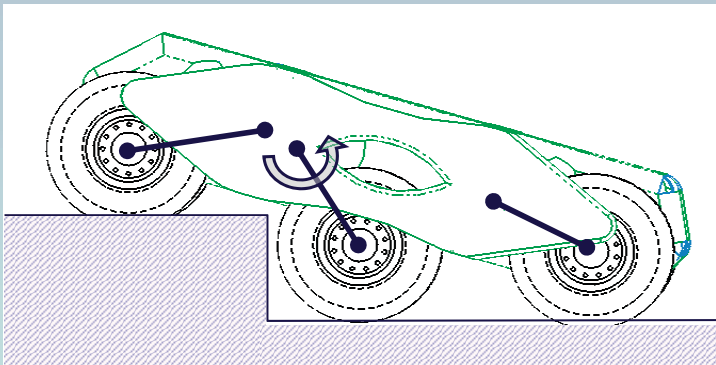


$$F_z = F_x \frac{e}{d}$$

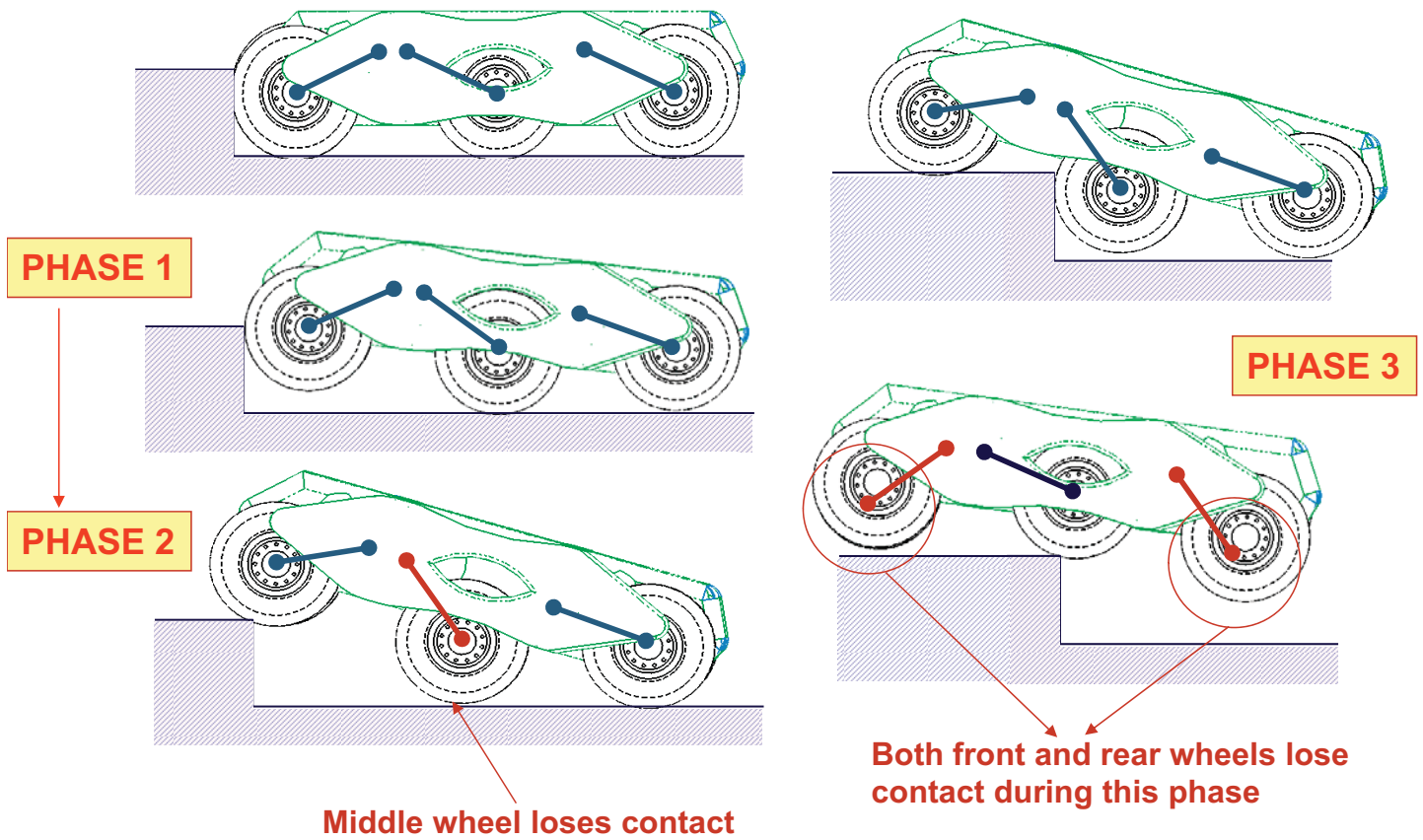
Design for large obstacle negotiation

Middle wheel climbing

Middle-axle trailing arm will easily lift over step corner



Synthesizing suspension geometries (cont.)



Relating geometry to forces

For Phases 1, 2, 3
Compute

1. Feasible initial:

$$\psi_{1i}, \psi_{2i}, \psi_{3i}, \phi_i$$

2. Feasible final:

$$\psi_{1f}, \psi_{2f}, \psi_{3f}, \phi_f$$

Fit functions:

$$\psi_1 = f_1(h, \alpha)$$

$$\psi_2 = f_2(h, \alpha)$$

$$\psi_3 = f_3(h, \alpha)$$

$$\phi = f_4(h, \alpha)$$

Derive static equations

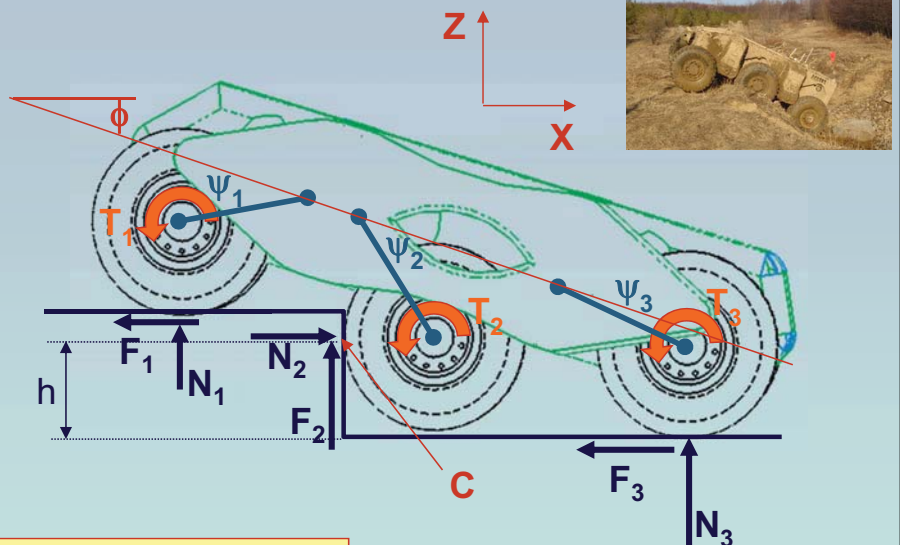
$$\sum F_x = 0, \sum F_z = 0, \sum M_C = 0$$

For wheels in contact with ground use spring/damper curve to derive: $N_j = g_j(f_1(h, \alpha))$

Solve system

for F_i and N_i

Compute required torques: $T_i = F_i R_w$



Considering feasible geometries

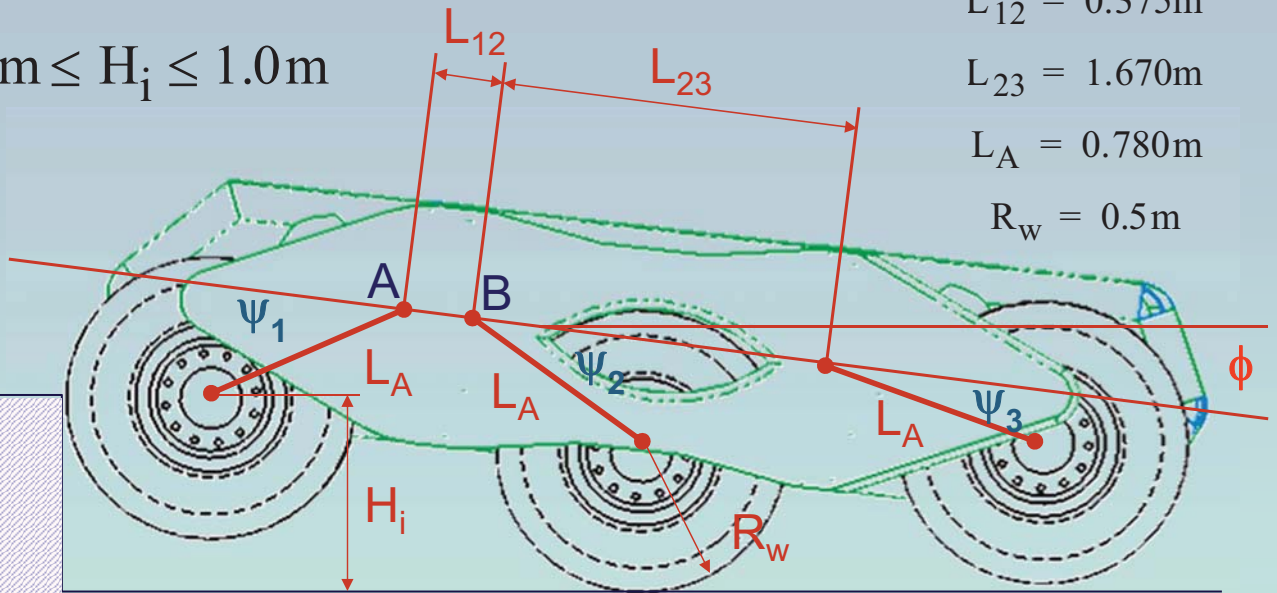
$$0.5\text{m} \leq H_i \leq 1.0\text{m}$$

$$L_{12} = 0.375\text{m}$$

$$L_{23} = 1.670\text{m}$$

$$L_A = 0.780\text{m}$$

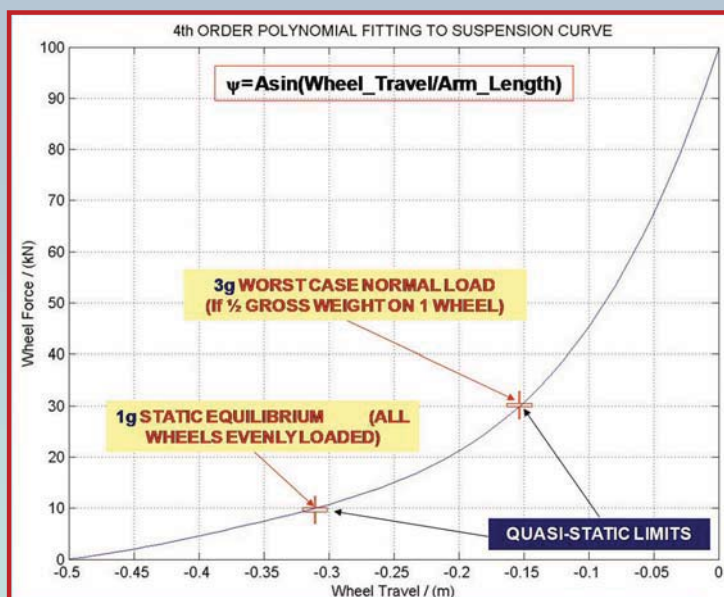
$$R_w = 0.5\text{m}$$



$$H_i + L_A \sin(\psi_1 - \phi) = L_{12} \sin \phi + L_A \sin(\psi_2 + \phi) + R_w$$

$$L_A \sin(\psi_2 + \phi) = L_A \sin(\psi_3 + \phi) + L_{23} \sin \phi$$

Deriving additional equations from suspension-specific spring/damper curves



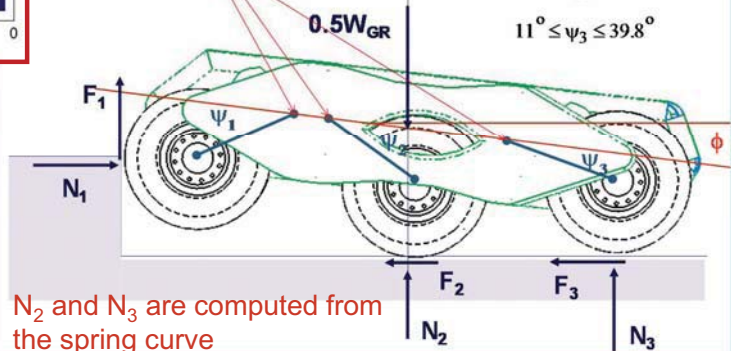
For wheels in contact with ground use spring/damper curve to derive: $N_j = g_j (f_1(h, \alpha))$

Torsional spring/damper response curves
Taken into account to derive feasible solutions

$$11^\circ \leq \psi_1 \leq 39.8^\circ$$

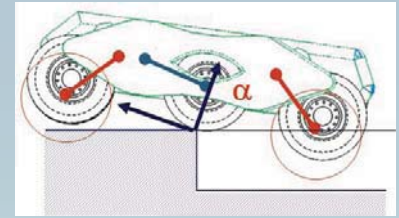
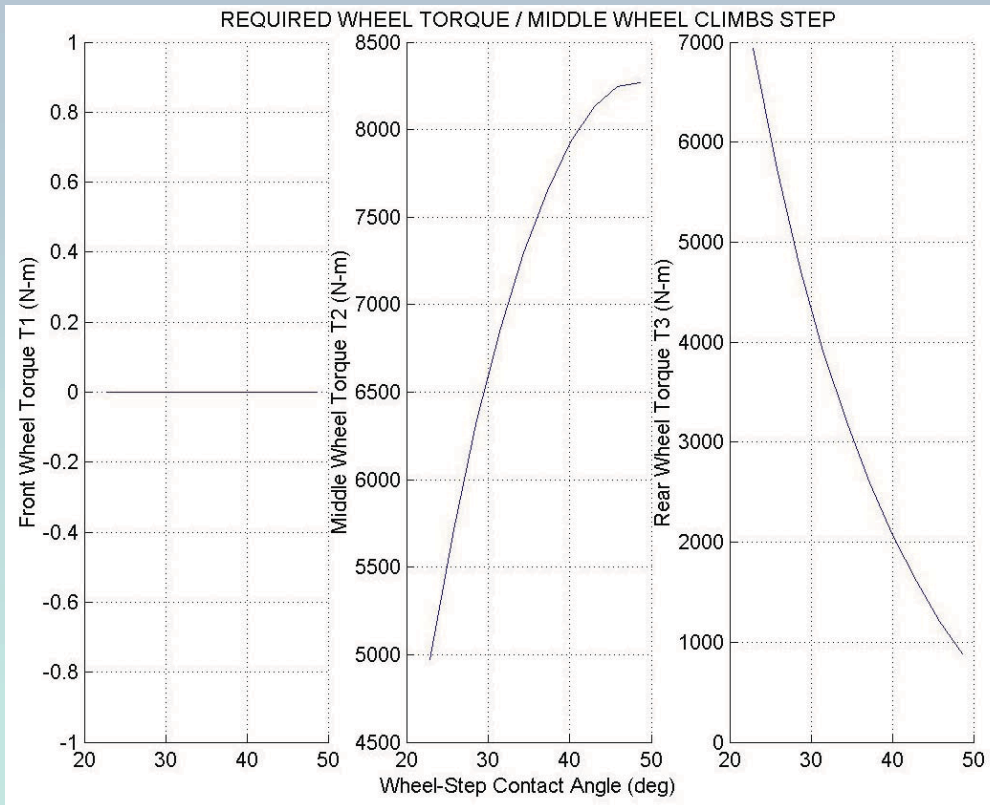
$$11^\circ \leq \psi_2 \leq 39.8^\circ$$

$$11^\circ \leq \psi_3 \leq 39.8^\circ$$



N_2 and N_3 are computed from the spring curve

Maximum instantaneous torque to climb obstacle



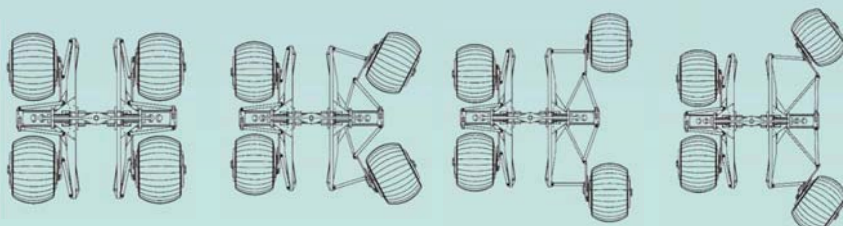
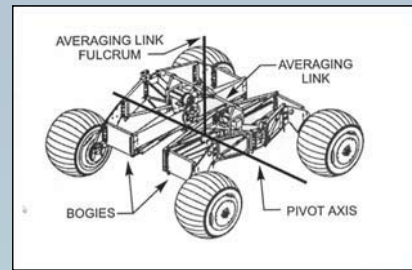
Solved equations of equilibrium for F_i and N_i Computed required torques: $T_i = F_i R_w$

Benefits from active posture adjustment

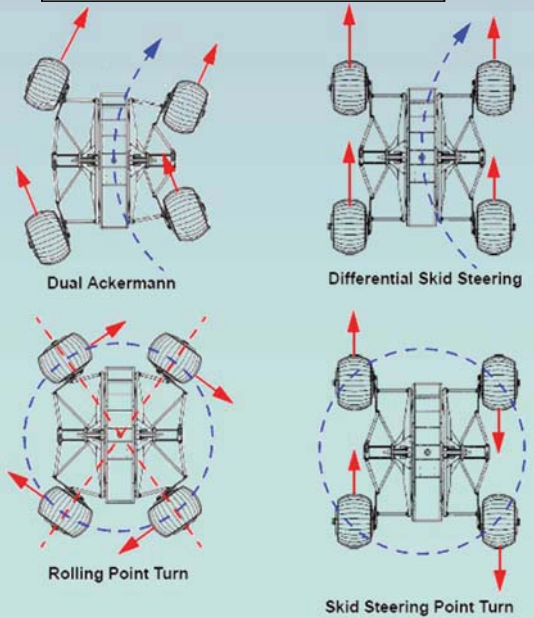


Dual-use mechanisms

CMU



Chassis reconfiguration



Thoughts and lessons from field deployments



Closing thoughts

Achieving high-mobility robotic designs

All-wheel drives

In-plane, long-travel suspensions

Progressive-spring behavior passive suspensions

Geometric suspensions excellent too

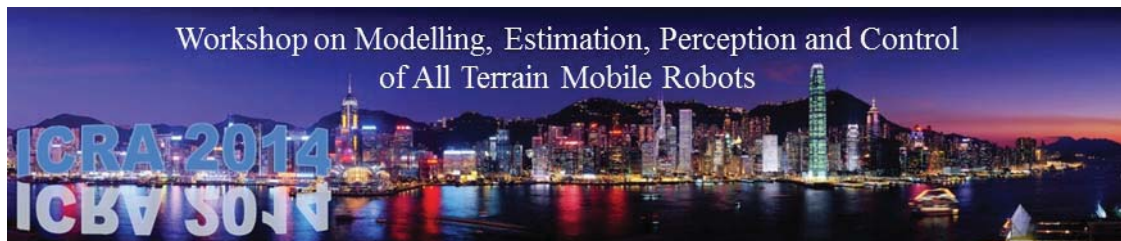
Maximum wheel diameter (fewer could be better)

Skid-steered or explicitly steered

Multiple wheels coupled to a single drivetrain (not always)

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots





Session III

Navigation, Control, Planning

• **Title: A Planner for All Terrain Vehicles on Unknown Rough Terrains based on the MPC Paradigm and D*- like Algorithm**

Authors: A. Tahirovic, M. Brkic, G. Magnani and L. Bascetta

• **Title: An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments**

Author: D. Alves de Lima and A. Correa Victorino

• **Title: Towards Lifelong Learning of Optimal Control for Kinematically Complex Robots**

Author: A. Dettmann¹, M. Langosz², Kai von Szadkowski¹, and S. Bartsch

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



A Planner for All Terrain Vehicles on Unknown Rough Terrains based on the MPC Paradigm and D*-like Algorithm

Adnan Tahirović, Mehmed Brkić, Gianantonio Magnani and Luca Bascetta

Abstract—A novel conceptual design of a planner for a mobile vehicle, operating on poorly traversable unknown rough terrains, is discussed. Finding a way to include a vehicle model into the planning stage, while coping with unknown or partially known terrains, is a challenging and rarely addressed optimization setup. The main advantages of a possible solution of such a problem would be twofold. First, the planner would give trajectories which are feasible to follow by the vehicle, which is not the case in many other state of the art planning algorithms especially for large vehicle speeds. Second, those trajectories would be the optimal ones in accordance to the current vehicle states and knowledge on its environment.

We propose a solution based on an MPC planning paradigm, wherein the planner solves a constrained optimal control problem at each time instant using the current knowledge on the terrain, which is caught appropriately by an objective function. Solving an optimal control problem allows for the vehicle model being included into the planning stage, while the repeated optimization allows for taking continuously into account new terrain information. To deal with the information given beyond the sensor range and to guarantee reaching a given goal position, we have adopted a D*-like algorithm for rough terrains being used as a cost-to-go term within the optimization setup.

I. INTRODUCTION

The popularity of the research on wheeled mobile robots has been recently increasing, due to their possible use in different outdoor environments. Planetary explorations, search and rescue missions in hazardous areas [1], surveillance, humanitarian de-mining [2], as well as agriculture works such as pruning vine and fruit trees, represent possible applications for autonomous vehicles in natural environments. Differently from the case of indoor mobile robotics, where only flat terrains are considered, outdoor robotics deals with all possible natural terrains. The unstructured environment and the terrain roughness, including dynamic obstacles [3], and poorly traversable terrains, make the development of an autonomous vehicle a challenging problem.

The aim of our research is to develop an All-Terrain Mobile Robot (ATMR), based on a commercial All-Terrain Vehicle (ATV), that is suitable for a wide range of different outdoor operations. The ATMR should be able to operate in any natural environment with a high level of autonomy.

The research leading to these results has received funding from Filas, the Regional Development Agency of the Lazio Region in Italy, under grant agreement No. FILAS-RS-2009-1290 - Project QUADRIVIO.

A. Tahirović and M. Brkić are with the Department of Automatic Control and Electronics, Faculty of Electrical Engineering, University of Sarajevo, Bosnia and Herzegovina (email: {atahirovic, mbrkic}@etf.unsa.ba).

L. Bascetta and G. Magnani are with Politecnico di Milano, Dipartimento di Elettronica e Informazione, Piazza Leonardo Da Vinci 32, 20133, Milano, Italy (email: {gianantonio.magnani, luca.bascetta}@polimi.it).

The advantage of using ATVs is represented by their good traversability potential for poorly traversable terrains and by the short time spent for reaching the goal, as well as by the possibility to operate in unsafe environments. On the other hand, the main disadvantage of ATVs is their low stability margin due to dynamic constraints, roll-over and excessive side slip [4].

An overview of motion planning algorithms has been presented in [5], [6], [7]. The interpretation of the state-of-the-art given in [7], related to the planning algorithms for mobile vehicles on rough terrains, is outlined in its compressed form in the sequel.

The research on motion planning evolved by adding the capability of taking into account the vehicle motion dynamics constraints within the well known dynamic window approach (DWA) [8], [9]. The DWA selects translational and rotational velocities by maximizing an objective function based on the vehicle heading to the goal position, distance to the closest obstacle, and velocity of the vehicle. The optimization is performed using arcs considering only reachable and safe velocities. This subject was extended to the high-speed navigation of a mobile robot in [10] by the global DWA, as the generalization of the DWA. A combination of the DWA with other methods yielded some improvements in long-term real-world applications [11]. Dubowsky and Iagnemma extended the DWA to rough terrains using the vehicle curvature-velocity space bounded by hazards as well as steering limits, wheel-terrain interaction, rollover and sideslip constraints. In this space the stability constraints of the vehicle, for instance, expressed by limit values of the roll-over and side slip indexes, can be easily described. The given algorithm was also suitable for high speed vehicles and appropriate for real-time implementation [12], [13], [14]. A convergent DWA was obtained for the unicycle mobile vehicle [15] exploiting the model predictive control combined with the direct Lyapunov function approach (MPC/CLF) [15], [16]. Even in the case when some of the aforementioned approaches might be used for unknown or partially known terrains, the continuous re-computation of the objective function term which attracts the vehicle toward the goal position, required when new information come from sensors, is limited to small scale terrains. In addition, all the approaches use the kinematic model of the vehicle, except the one presented in [15], [16], where the dynamic model of unicycle mobile vehicle has been used for planning purposes.

Sequences of motion primitives have been used to cover local planning search space since [17]. More recent works are given in [18] and [19], where the inverse trajectory

generation was used to navigate UAV and UGV, respectively. The importance of separation in a local planning search space is discussed in [20] and it was shown that the mutual separation of a set of paths is related to the relative completeness of the motions set. The planning approach proposed in [21] generate path sets to navigate an UGV. The planner considers global guidance, satisfies environmental constraints, and guarantees dynamic feasibility by the use of a model-predictive trajectory generator.

A grid based planning approach which takes into account the vehicle differential constraints is introduced in [22]. This planner uses the vehicle model to generate the state lattices assuring the feasible paths along the cost map edges. The heuristic cost estimate, which represents cost-to-go for each node of the grid needed by the A^* algorithm [23], is taken from a priori calculated heuristic look-up table (HLUT) [24],[25], which is based on the path length, speeding up the algorithm. In [22], the cost edges are calculated by solving two boundary value problem where the control action is parameterized converting the problem into a nonlinear programming one. The cost function represented the minimum slope-dwell performance index, selecting less difficult paths between the nodes (lattice states) that are considered in the overall optimization. Including the vehicle model into the motion planning stage provides a planner which generates trajectories that can be easily followed by a mobile robot. This especially comes to the fore when a vehicle moves with high speed and operates on rough terrains. Using a simpler planner that does not take into account the mobile vehicle model might cause a fatal error due to the difference between the planned and executed trajectories. For this reason, the gradient based algorithms such as the navigation function or a variant of the D^* [26], [27], in our case are not considered being an acceptable solution.

The sample-based technique for robot motion planning was introduced in [28]. The first sample-based motion planners were not computationally efficient for certain environments. In [29], [30], [31] the probabilistic roadmap method (PRM) was developed for path planning in configuration spaces with many degrees of freedom. A comprehensive overview and discussion about PRM is given in [32] and [33]. PRM method has prove to work well in static well-known environments and are considered computationally efficient for car-like vehicles [34]. In [35], the authors introduced quasi-PRM and lattice roadmap (LRM) algorithms. LRM was extended in [22] to allow the state lattice to represent the differential constraints of the mobile vehicle. However, PRM may not be suitable for planning in a dynamic environment, especially because it does not take into account the vehicle dynamics and might result in very sharp turning points.

Rapidly exploring random trees (RRT) is a type of probabilistic planners originally developed to cope with differential constraints [36], [37], [38]. A significant feature of the RRT-like algorithms is that the resulting trajectories are executable by the underlying dynamical system. The RRT algorithm has been proven probabilistically complete [38], meaning that the probability of finding a solution feasible path converges

to one if such a path exists. An improvement of the RRT algorithm was proposed in [39], where the obtained exponential convergence speed yielded a good performance. Several variants of the roughness-based RRTs are illustrated in [40], [41], [42], [43], while some recent results on the RRT-like planners have been introduced in [44] and [45]. Although the RRT may find a feasible solution it cannot be considered an optimal approach in accordance to the current vehicle states and knowledge on the terrain.

The RHC/CLF (Receding Horizon Control/Control Lyapunov Function) scheme developed in [46] used the concept of control Lyapunov function to obtain the stability of RHC scheme. The authors presented the generalization of the RHC/CLF scheme demonstrating its relation to the optimal controller. In [15], the authors have implemented the same scheme (MPC/CLF, Model Predictive Control/Control Lyapunov Function) for the navigation planning of a unicycle mobile vehicle. The approach developed and proposed in [47] used a passivity-based constraint to obtain an MPC scheme with guaranteed closed loop stability for nonlinear systems. Inspired by this control concept, a framework for mobile robot motion planning using the PB/MPC is presented both for flat and rough terrains in [48], [49], [50], where any dynamic model can be used to plan the vehicle trajectories. The main issue of this framework is computation of the cost-to-go term for the given objective function, which is required each time new information is acquired by the vehicle sensors. The idea of using optimal cost-to-go map computed by Dijkstra algorithm is presented in [51]. However, a continuous computing such a map and/or computing it for large scale terrains can be computationally very expensive, hence inappropriate for real time reactive motion planning.

The idea of using a faster algorithm to compute cost map for the purpose of cost-to-go term required by the MPC planning optimization framework, has been addressed in [7]. Therein, an approximate computation, named as Roughness-based Navigation Function (RbNF), has been presented, while some of the applications of using such a cost map have been examined in [52]. Although it has not been presented in [51], using a D^* -like algorithm as a cost-to-go term has been mentioned as a possible solution. Since the D^* -like algorithms inherently deal with unknown terrains, the focus of our current research is an MPC optimization setup which uses a D^* -like algorithm for planning purposes on unknown rough terrains.

In order to use the MPC optimization setup using a D^* -like algorithm for unknown terrains, we properly extend the setup presented in [51] by defining varying state constraints for each MPC optimization cycle in accordance to the vehicle current information on the terrain. The planner may include any dynamic model into the optimization setup, and it can comply with any constraints such as those imposed on acceleration, velocity, roll and slip angles. By using D^* -like algorithm, such a planner deals with unknown terrains in a near-optimal manner.

After the description of the aim of our ongoing research (Section I), the state of the art relevant to the ATV planning

on rough terrains (Section I), the ATV which will be used in the experimental setup (Section II), the novelty of the ongoing research is presented in Section III throughout an MPC optimization setup, and it is validated within Section IV.

II. THE ATMR

The vehicle considered in this research (see Fig. 1) is a YAMAHA GRIZZLY 700, a commercial fuel powered All-Terrain Vehicle (ATV) equipped with an electric power steering (EPS).

The GRIZZLY 700 is a utility ATV and is thus specifically designed for agriculture work. As a result it has a total load capacity of 130 Kg, and it is equipped with a rear tow hook. The main characteristics of the vehicle are listed in Table I.



Fig. 1. The Yamaha Grizzly 700 ATV

For the purposes of the project, the original vehicle cover has been removed and substituted with an aluminium cover, that allows to easily accommodate for the control hardware and the sensors.

Main characteristics of the vehicle

Engine type	686cc, 4-stroke, liquid-cooled, 4 valves
Drive train	2WD, 4WD, locked 4WD
Transmission	V-belt with all-wheel engine braking
Brakes	dual hydraulic disc (both f/r)
Suspensions	independent double wishbone (both f/r)
Steering System	Ackermann
Dimensions (LxWxH)	2.065 x 1.180 x 1.240 m
Weight	296 Kg (empty tank)

TABLE I

VEHICLE CHARACTERISTICS

Control system and software architectures used on the ATMR have been described and presented in [51].

III. AN MPC-BASED PLANNER ON UNKNOWN TERRAINS

In this work we extend the idea presented in [51] from the ATV planning on known to unknown terrains. We use \mathbf{D}^* -like algorithm, which inherently deals with unknown terrains, as an input to form a properly differentiable function, providing for a cost-to-go term required by the MPC optimization setup. In order to count for the new information continuously coming into the vehicle sensor range, we define a local term of an objective function as a measure of roughness in the same way as in [51]. However, to locally deal with the new obstacles, we now define a varying constraint set for the MPC optimization setup.

The MPC optimization problem can be expressed as an initial value optimal control problem (OCP) with an end-free position eqs. (1-4). The task of this optimization is to find the input \mathbf{u} of the vehicle (the velocity and steering angle for a kinematic model, or the acceleration and steering momentum for a dynamic model) along the optimization horizon $t \in (t_0, t_0 + T)$, that is over all potential candidate paths, by minimizing the cost function $J(\mathbf{u})$ given in (1).

The integrand $\gamma(\mathbf{x}, \mathbf{u})$ represents a differentiable function as a measure of the local terrain roughness estimated by the vehicle within the sensor range. Many different terrain roughness measures can be used [53], [7] to compute the roughness cost map. To get such a differentiable function, we interpolate roughness map over the sensor range domain for each MPC optimization cycle.

The cost-to-go term $\Gamma(t_0 + T)$ represents a differentiable function as a measure of the estimated cost-to-go map in accordance with the current vehicle information on the terrain. This cost-to-go map is computed by \mathbf{D}^* using both obstacle and roughness maps acquired by the vehicle prior the beginning of each MPC optimization cycle. Hence, $\Gamma(t_0 + T)$ gives the cost of traversing the path on rough terrain from a position at the terminal optimization time, $t_0 + T$, to a goal position, by using \mathbf{D}^* algorithm. To get such a differentiable function, we now interpolate the cost-to-go map over the sensor range.

Eqs. (2-4) represent optimization constraints including the differential constraint related to the vehicle model (2), state constraints and control constraints (4). State constraints vary with each MPC optimization horizon such that they prevent the vehicle terminal state, $\mathbf{x}(t_0 + T)$, being beyond the sensor range. This constraint is important since functions $\gamma(\mathbf{x}, \mathbf{u})$ and $\Gamma(t_0 + T)$ are interpolated using the discrete values from the roughness and cost-to-go maps over that range, respectively. Obstacles, which are located within the sensor range, are also appropriately included in eq. (3). Other constraints such as those preventing the vehicle from the sideslip and rollover can also be easily accommodated into the optimization setup.

$$J(\mathbf{u}) = \int_{t_0}^{t_0+T} \gamma(\mathbf{x}, \mathbf{u}) dt + \Gamma(t_0 + T) \quad (1)$$

$$\frac{d}{dt} \mathbf{x} = f(\mathbf{x}, \mathbf{u}) \quad (2)$$

$$\mathbf{x}(t) \in \mathbf{X}, \quad (3)$$

$$\mathbf{u}(t) \in \mathbf{U} \quad (4)$$

The optimization is solved by using an OCP software GPOPS I [54], providing the planner eqs. (1-4) being nearly optimal due to the 'optimality principle' since the \mathbf{D}^* algorithm is a near optimal estimator of the cost-to-go map. This is equally true both for the known and unknown terrains. For the latter case, \mathbf{D}^* algorithm estimates the cost-to-go map nearly optimal in accordance to the vehicle current belief on the environment.

IV. SIMULATIONS

We illustrate two simulation setups in order to show two different ways of using the proposed planner on unknown terrains. Since the optimization framework allows for using any vehicle model, we use, without loss of generality, a double integrator as a model of the vehicle. For the purpose of simplicity, we assume that the vehicle perfectly follows the planning path for each MPC control horizon.

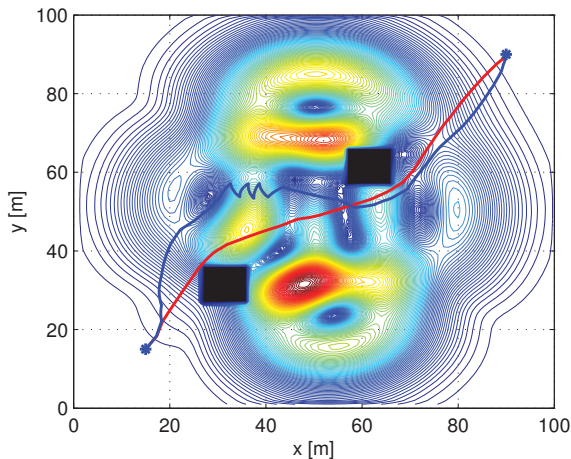


Fig. 2. The generated paths depicted on the contour plot of the roughness cost map. Blue: the complete planner. Red: \mathbf{D}^* gradient based planner. Initial position: (15,15). Goal position: (90,90).

The first setup, illustrated with Figs. 2 and 3, utilizes a rough terrain with obstacles. The path depicted with the blue line is generated by the planner given by eqs. (1-4) (complete planner), while the one depicted with the red line is generated without using local objective function term $\gamma(\mathbf{x}, \mathbf{u})$ (\mathbf{D}^* gradient based planner). Namely, in the latter case, we obtain a planner which is the result only of the cost-to-go term, $\Gamma(t_0 + T)$, meaning that the planner follows the gradient of the \mathbf{D}^* map at the end of the current optimization horizon, $t_0 + T$. Such a planner mimics the \mathbf{D}^* algorithm and it also includes the model of the vehicle into optimization setup. However, the model without using local objective function term does not take into account the terrain roughness within the sensor range. This fact can be seen in Figs. 2 and 3, where Fig. 2 illustrates the paths on the contour plot of the roughness cost map, and Fig. 3 depicts the paths on the

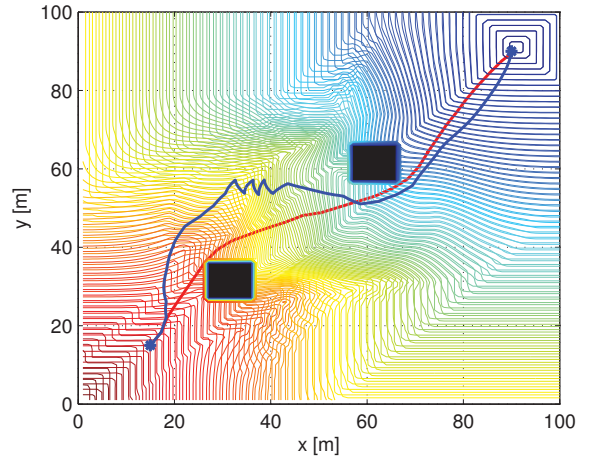


Fig. 3. The generated paths depicted on the contour plot of the cost-to-go map obtained by \mathbf{D}^* algorithm once the terrain is completely known. Blue: the complete planner. Red: \mathbf{D}^* gradient based planner. Initial position: (15,15). Goal position: (90,90).

contour plot of the cost-to-go map obtained by \mathbf{D}^* algorithm once the terrain is completely known.

The second simulation setup, illustrated with Figs. 4, 5 and 6 shows a different implementation possibility. In this case, the obstacles are considered as highly rough terrain parts, as shown in Fig. 4. Figs. 5 and 6 show the generated paths both for the complete and the \mathbf{D}^* gradient based planner. In all simulated cases, the proposed planner appropriately guides the vehicle through unknown terrain.

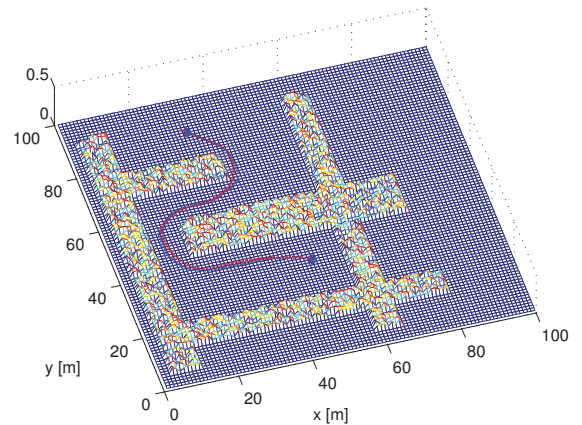


Fig. 4. The generated paths depicted on the roughness map, where the obstacles are considered as highly rough part of the terrains. Blue: the complete planner. Red: \mathbf{D}^* gradient based planner. Initial position: (50,35). Goal position: (30,90).

V. CONCLUSIONS

This paper describes part of the work devoted to the development of an All-Terrain Mobile Robot, based on a commercial All-Terrain Vehicle, for riding on unknown difficult terrains. Besides the developments of control and

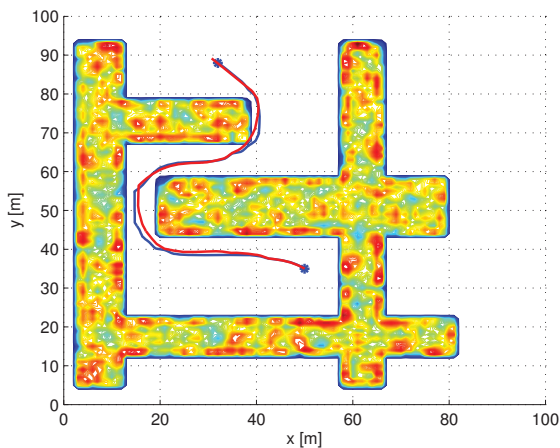


Fig. 5. The generated paths depicted on the roughness cost map, where the obstacles are considered as highly rough part of the terrains. Blue: the complete planner. Red: D^* gradient based planner. Initial position: (50,35). Goal position: (30,90).

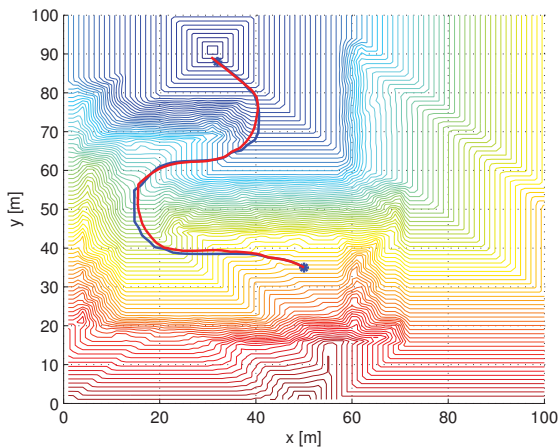


Fig. 6. The generated paths depicted on the contour plot of the cost-to-go map obtained by D^* algorithm once the terrain is completely known. Blue: the complete planner. Red: D^* gradient based planner. Initial position: (50,35). Goal position: (30,90).

software architecture for such a vehicle, which have been covered in our previous work, one of the main issues of our current research is the development of an ATV reactive planner for unknown terrains. We propose a novel near optimal planning technique based on an MPC combined with the D^* algorithm. An MPC based planner can account for the vehicle model during the planning stage and impose a wide range of constraints such as obstacles and those for preventing the vehicle from the sideslip and rollover. A thorough statistical analysis and experimental validation, using the vehicle described in Section II, is our ongoing research.

REFERENCES

[1] A. Garcia Cerezo, A. Mandow, J. Martinez, J. Gomez de Gabriel, J. Morales, A. Cruz, A. Reina, and J. Seron, "Development of

ALACRANE: a mobile robotic assistance for exploration and rescue missions," in *IEEE International Workshop on Safety, Security and Rescue Robotics*, 2007.

- [2] P. Debenest, E. Fukushima, and S. Hirose, "Proposal for automation of humanitarian demining with buggy robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 2003, pp. 329–334.
- [3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.
- [4] R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, "Backstepping observer dedicated to tire cornering stiffness estimation: application to an All Terrain Vehicle and a farm tractor," in *IEEE/RSJ International Conference on Intelligent Robotics Systems*, 2007, pp. 1763–1768.
- [5] J. Latombe, *Robot Motion Planning*. Boston: Boston, MA: Kluwer, 1991.
- [6] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.
- [7] A. Tahirovic and M. Gianantonio, *Passivity-Based Model Predictive Control for Mobile Vehicle Motion Planning*. London, UK: Springer London, 2013.
- [8] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proc. of the IEEE International Conference on Robotics and Automation*, 1996, pp. 3375–3382.
- [9] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics and Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar 1997.
- [10] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 1, 1999, pp. 341–346.
- [11] R. Philippsen and R. Siegwart, "Smooth and efficient obstacle avoidance for a tour guide robot," in *Proc. of the IEEE International Conference on Robotics and Automation*, vol. 1, 2003, pp. 446–451.
- [12] M. Spenko, Y. Kuroda, S. Dubowsky, and K. Iagnemma, "Hazard avoidance for high speed unmanned ground vehicles in rough terrain," *Journal of Field Robotics*, vol. 23, no. 5, pp. 311–331, 2006.
- [13] M. Spenko, "Hazard avoidance for high speed rough terrain unmanned ground vehicles," Ph.D. dissertation, Massachusetts Institute of Technology, MA, 2005.
- [14] S. Shimoda, Y. Kuroda, and K. Iagnemma, "High-speed navigation of unmanned ground vehicles on uneven terrain using potential fields," *Robotica*, vol. 25, no. 4, 2007.
- [15] P. Oegren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.
- [16] —, "A provably convergent dynamic window approach to obstacle avoidance," in *Proc. of IFAC World Congress*, 2001, pp. 595 – 600.
- [17] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497 – 516, 1957.
- [18] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *Journal of Guidance, Control, and Dynamic*, vol. 1, no. 25, 2002.
- [19] T. M. Howard and A. Kelly, "Optimal rough terrain trajectory generation for wheeled mobile robots," *Int. J. Rob. Res.*, vol. 26, no. 2, pp. 141–166, 2007.
- [20] C. J. Green and A. Kelly, "Toward optimal sampling in the space of paths," in *Proc. of the International Symposium of Robotics Research*, 2007, pp. 171 – 180.
- [21] T. M. Howard, C. J. Green, A. Kelly, and D. Ferguson, "State space sampling of feasible motions for high-performance mobile robot navigation in complex environments," *Journal of Field Robotics*, vol. 25, no. 10, pp. 325–345, 2008.
- [22] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *J. Field Robot.*, vol. 26, no. 3, pp. 308–333, 2009.
- [23] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [24] M. Pivtoraiko and A. Kelly, "Efficient constrained path planning via search in state lattices," in *Proc. of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, September 2005.

- [25] R. A. Knepper and A. Kelly, "High performance state lattice planning using heuristic look-up tables," in *Proc. of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2006, pp. 3375 – 3380.
- [26] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Proc. of the International Joint Conference on Artificial Intelligence*, 1995, pp. 1652–1659.
- [27] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354 – 363, 2005.
- [28] J. Barraquand and J. Latombe, "Motion planning: A distributed representation approach," *The International Journal of Robotics Research*, vol. 10, no. 6, 1991.
- [29] L. E. Kavraki, "Random networks in configuration space for fast path planning," Ph.D. dissertation, 1995.
- [30] M. H. Overmars and P. Švestka, "A probabilistic learning approach to motion planning," in *WAFR: Proceedings of the workshop on Algorithmic foundations of robotics*, 1995.
- [31] L. Kavraki, P. Švestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for fast path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, 1996.
- [32] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.
- [33] D. Hsu, J.-C. Latombe, and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," *Int. J. Rob. Res.*, vol. 25, no. 7, 2006.
- [34] G. Song and N. M. Amato, "Randomized motion planning for car-like robots with C-PRM," in *Proc. of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 37 – 42.
- [35] M. S. Branicky, S. M. Lavalley, K. Olson, and L. Yang, "Quasi-randomized path planning," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2001, pp. 1481–1487.
- [36] J. Kuffner and S. M. Lavalley, "Randomized kinodynamic planning," in *Proc. of the IEEE International Conference on Robotics and Automation*, 1999, pp. 473–479.
- [37] S. M. Lavalley and J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*. Wellesly, MA: AK Petetrs, Ltd., 2001, pp. 293–308.
- [38] —, "RRT-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2000, pp. 995–1001.
- [39] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," in *Algorithmic and Computational Robotics: New Directions*, 2001, pp. 247–264.
- [40] D. J. Spero and R. A. Jarvis, "Path planning for a mobile robot in a rough terrain environment," in *Third International Workshop on Robot Motion and Control*, 2002, pp. 9–11.
- [41] M. Kobilarov and G. S. Sukhatme, "Time optimal path planning on outdoor terrain for mobile robots under dynamic constraints." Unpublished research paper from the USC Center for Robotics and Embedded Systems Lab., 2004.
- [42] A. Ettlín and H. Bleuler, "Randomised rough-terrain robot motion planning," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 5798 – 5803.
- [43] A. Tahirovic and G. Magnani, "A roughness-based RRT for mobile robot navigation planning," in *Proc. of the 18th IFAC World Congress*, 2011, pp. 5944 – 5949.
- [44] D. Ferguson and A. Stentz, "Anytime rrt," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.
- [45] A. P. E. F. S. Karaman, M. R. Walter and S. Teller, "Anytime motion planning using the RRT," in *Proc. of the IEEE International Conference on Robotics and Automation*.
- [46] J. A. Primbs, V. Nevistic, and J. C. Doyle, "Nonlinear optimal control: A control Lyapunov function and receding horizon perspective," *Asian Journal of Control*, vol. 1, pp. 14–24, 1999.
- [47] T. Raff, C. Ebenbauer, and F. Allgoewer, *Nonlinear Model Predictive Control: Passivity-based Approach*. New York: Springer Berlin / Heidelberg, 2007.
- [48] A. Tahirovic, G. Magnani, and P. Rocco, "Mobile robot navigation using passivity-based MPC," in *Proc. of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2010, pp. 248–488.
- [49] A. Tahirovic and G. Magnani, "General framework for mobile robot navigation using passivity-based MPC," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, 2011.
- [50] —, "Passivity-based model predictive control for mobile robot navigation planning in rough terrains," in *Proc. of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2010.
- [51] L. Bascetta, G. Magnani, M. Matteucci, D. Osmankovic, and A. Tahirovic, "Towards the implementation of a mpc-based planner on an autonomous all-terrain vehicle," in *Workshop on Robot Motion Planning: Online, Reactive, and in Real-time, The IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012*, 2012.
- [52] A. Tahirovic, G. Magnani, and Y. Kuwata, "An approximate of the cost-to-go map on rough terrains," in *Proc. of the IEEE International Conference on Mechatronics 2013*.
- [53] K. Iagnemma and S. Dubowsky, *Mobile Robots in Rough Terrain: Estimation, Motion Planning and Control with Application to Planetary Rovers*. New York: Springer Berlin / Heidelberg, 2004.
- [54] A. Rao, D. Benson, C. Darby, M. Patterson, C. Francoín, I. Sanders, and G. Huntington, "Gpops: A matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method," *ACM Transactions on Mathematical Software*, vol. 37, no. 2, 2010.

An image based dynamic window approach for local navigation of an autonomous vehicle in urban environments

Danilo Alves de Lima and Alessandro Corrêa Victorino

Abstract—This paper presents a local navigation strategy for autonomous vehicles in urban environments with an Image Based Dynamic Window Approach (IDWA). Differently from the global navigation techniques, which requires the vehicle localization to perform its movement, the focus here was to solve the navigation problem in local navigation steps. For that, the environment features will be used, performing the road lane following e.g. The DWA performs a reactive obstacle avoidance while trying to reach a goal destination. In this case, reach the goal destination is based on the Image Based Visual Servoing equations for road lane following, which were incorporated into the DWA. The final solution takes into account the car kinematics/dynamics constraints to allow the vehicle to follow the road lane while avoiding obstacles. The results show the viability of the proposed methodology.

Index Terms—Dynamic Window Approach, Visual Servoing, Local Navigation, Obstacle Avoidance.

I. INTRODUCTION

Traditionally, a car-like robot navigates based on its perception of the environment and its localization related to the path previously planned. However, in urban environments, the localization systems based in GPS information must deal with signal losses and several noises caused by urban canyons, as reported by many DARPA's Challenges participants between 2004 and 2007 [1]. Due to the GPS drift, this problem also increases locally when a vehicle tries to follow a road based on GPS points. One way to deal with it is using local navigation techniques, which do not use the vehicle global position to calculate the control action. These techniques are normally based on local features extracted from exteroceptive sensors (like vision systems), which in urban environments there are useful ones available [2]. Thus, a global navigation task in urban environments can be divided in road following (branches), representing the local tasks, and road intersections maneuvers (nodes), connecting the next local task. To accomplish the global task, the global localization system can be limited only to the nodes, e.g. using techniques based on vehicle-to-infrastructure (V2I) communication [3].

Focusing on the local navigation, there are many approaches to perform vehicle control using visual data [4], [5], [2]. In addition to follow the desired features the vehicle must also consider reactive techniques for obstacle avoidance. A well-known reactive technique is the Dynamic Window Approach (DWA) [6], which searches for an optimal input

command between all possible commands in a short time interval. Its optimization function takes into account the final goal position (*heading*), the obstacles distance (*dist*), and the maximum linear velocity (*velocity*) during the calculation. It also considers the kinematics and some dynamics constraints of the robot. Due to the nature of its optimization function, it can be adapted to several techniques [7], [8], [9], to different robot types, like car-like robots [10], [11], as well as to dynamic environments [12]. However, these works using DWA are conceived when the robot and goal positions are known in the world frame, recalling the localization problems previously mentioned.

To avoid this problem, the proposed work presents a local navigation strategy for autonomous vehicles in urban environments with an Image Based Dynamic Window Approach (IDWA). Differently from the global DWA techniques [7], [13], [9], which requires the vehicle localization to perform its movement, the focus here is to solve the navigation problem in local navigation steps using the environment features acquired from a camera, performing e.g. the road lane following. In this case, reaching the goal destination is a task guaranteed by the Image Based Visual Servoing equations [4], [14], [15], incorporated into the DWA functions. This work also differs from the vision navigation approach proposed by [16], based on the tentacles technique of [1], once the image based task and the obstacle avoidance are in the same controller on the robot velocity space. The objective in mind is to perform Image Based Visual Servoing control tasks while validating their velocity outputs in a obstacle avoidance methodology. In the near future, it will allow electric vehicles, like the one from the project VERVE¹, to perform local navigation in road lanes with a safe behavior.

In the block diagram of the Figure 1 it is shown the present methodology, structured in two general layers: workspace perception and robot control. Its concepts are presented in this article as follow: Section II presents the robot model used and the problem definition; Section III presents the workspace perception layer, describing the environment perception strategy to features extraction and obstacle detection; the navigation control layer, with the proposed Image Based Dynamic Window Approach, is presented in the Section IV; an experimental analysis and validation of the method, using a simulated car-like robot, is in Section V; and, finally, Section VI presents some conclusions and perspectives for future works.

The authors are with Heudiasyc UMR CNRS 7253 Université de Technologie de Compiègne. Danilo Alves Lima holds a Ph.D scholarship from Picardi region. Contact authors danilo.alves-de-lima@hds.utc.fr

¹The project VERVE stands for *Novel Vehicle Dynamics Control Technique for Enhancing Active Safety and Range Extension of Intelligent Electric Vehicles*.

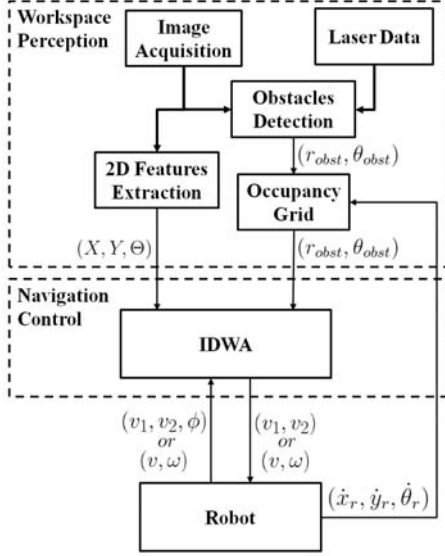


Fig. 1. Methodology block diagram.

II. GENERAL DEFINITIONS

The car-like robot used in this work is similar to the ones described in [15], [13]. It is considered to move in a planar workspace with a fixed pinhole camera directed to the front to perceive and follow the road lane center, which defines a path once differentiable in \mathbb{R}^2 . The vehicle is also considered to be over the road surface and able to always see a road lane. The kinematic model is based on a front wheel car, represented as [17]:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \phi \\ \sin \theta \cos \phi \\ \sin \phi/l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2, \quad (1)$$

where the vehicle configuration is given by $q = [x_r \ y_r \ \theta \ \phi]^T$, with the position (x_r, y_r) and orientation (θ) of the car's reference frame $\{\mathcal{R}\}$ in relation to a static world reference frame $\{\mathcal{O}\}$, and ϕ is the average steering angle of each front wheel by the Ackerman approximation [17]. The orientation and steering angles $(\theta$ and $\phi)$ are defined as $\theta \in [-\pi, \pi]$ and $\phi \in [-\phi_{max}, \phi_{max}]$, both positive counter-clockwise. The Figure 2 illustrates these variables. Note that the origin of $\{\mathcal{R}\}$ is located at the midpoint of the two rear wheels, which performs circular trajectories defined by the instantaneous center of curvature (ICC). The approximation for the steering angle ϕ is related to the x_r axis, pointed to the front of the vehicle.

For the vehicle model (1), the control input is $u = [v_1 \ v_2]^T$, where v_1 is the linear velocity of the front wheels and v_2 is the steering velocity. In this model, the robot linear velocity v is related to the front wheels velocity by $v = v_1 \cos(\phi)$, and the angular velocity $\dot{\theta} = v_1 \cos(\phi)/r_1 = \omega$ is directed related to the steering angle (see the Figure 2), which allows to chose the robot control input as $u_r = [v \ \omega]^T$. These inputs can be generalized for an unicycle robot, although the

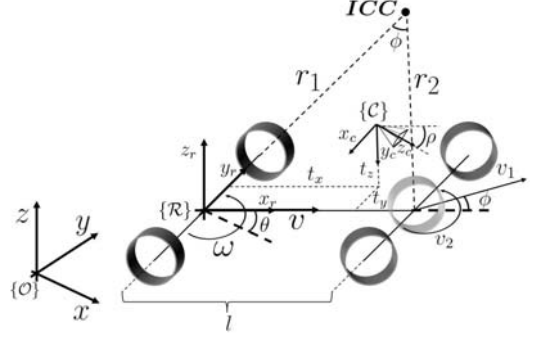


Fig. 2. Kinematic model diagram for a front wheel car-like robot. In this model the vehicle reference frame \mathcal{R} performs circular trajectories related to the instantaneous center of curvature (ICC). The pinhole camera frame is also represented in \mathcal{C} .

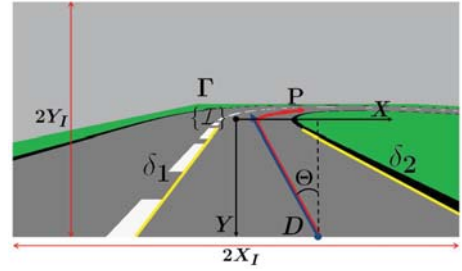


Fig. 3. Image frame $\{\mathcal{I}\}$ with the road lane center projection P (in red) related to the boundaries δ_1 and δ_2 (in yellow), its tangent Γ (in blue) at the point D and the angle offset Θ of Γ to the axis $-Y$.

robot frame $\{\mathcal{R}\}$, in the unicycle, be located in the projection of the wheel center on the ground. The main difference is regarding a constraint present in the car-like robot model, which limits the ICC (Figure 2) by ϕ , and that is not present in the unicycle robot.

Figure 2 also represents the camera frame $\{\mathcal{C}\}$ with optical center position in $(x_c, y_c, z_c) = (t_x, t_y, t_z)$ in the robot frame and a constant tilt offset $0 < \rho < \frac{\pi}{2}$ related to the x_r axis, required for the image based approach. The camera final position is in the robot sagittal plane ($t_y = 0$), which is not a limitation, but must be with a certain height from the floor ($t_z > 0$). Finally, the camera's image frame $\{\mathcal{I}\}$ is illustrated in the Figure 3, with a defined size of $(2X_I, 2Y_I)$.

III. WORKSPACE PERCEPTION

The workspace perception is the first step for the local navigation task proposed (Figure 1), that provides the environment information (calculated by on-boarded camera and laser scan) required to perform the Image Based Dynamic Window Approach (IDWA). It was divided in 2D features extraction, obstacle detection and occupancy grid representation.

The current implementation of the IDWA uses a similar features set presented by Cherubini et al [15], applied in a path reach and following strategy of a nonholonomic robot. It uses small path features set to navigate, defined as the projection in the image plane of a visible white line on the floor, with its features calculated in the image frame

$\{\mathcal{L}\}$ in an Image Based Visual Servoing scheme [14]. These features were adapted for the road lane following problem as described in Figure 3, where they are related to the tangent Γ of the path P (according to its direction) at the point $D = (X, Y)$, with an angular offset $\Theta \in]-\pi, \pi]$ from Γ to the axis $-Y$ (positive counterclockwise). P is the center of the road surface between the boundaries δ_1 and δ_2 , which are on the limit of the most right visible lane or, in case of non lane marks, are on the road limits.

An obstacle detection layer is also necessary in the IDWA to guarantee the right execution of obstacle avoidance maneuvers, and with an occupancy grid [18] the obstacles can be stored during the robot movement. Once no entire environment information must be on the grid, the occupancy grid can be reduced to a local window around the robot, actualized with its movement (see Figure 1). For more details about the obstacle detection and occupancy grid layers, see [13]. This implementation considers only static environments for validation purposes, which does not restrain a future implementation with dynamic environments as presented in [12].

IV. NAVIGATION CONTROL

The present controller was based on the integration of the Image Based Visual Servoing (IBVS) [15] equations with the Dynamic Window Approach (DWA) [6] to perform the obstacle avoidance while performing the road lane following. This technique was called by Image Based Dynamic Window Approach (IDWA), used by the navigation control layer of Figure 1, and will be presented in this section.

A. The Dynamic Window Approach

DWA is a reactive obstacle avoidance technique proposed originally by [6], adapted for car-like robots by [10], which selects in the velocity space an optimum control input around the current robot state. It regards some kinematics/dynamics conditions of the robot to construct a control search space, classified by the weighted sum of three functions. They are based on the goal position (*heading*), the obstacle distance (*dist*) and the final linear velocity (*velocity*), compounding the objective function (3):

$$DWA(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega), \quad (3)$$

to be optimized.

1) *The DWA Functions*: In the original formulation of the DWA [6], the function $\text{heading}(v, \omega)$ is responsible to guide the robot to a desired goal position, calculating high weights to the velocity inputs that lead the robot to a final orientation closer to the goal position in the world frame. It is frequently adapted when some specific navigation task is required [7], [13], [9]. Its improvements proposed for this work will be presented in the Subsection IV-B. The next function $\text{dist}(v, \omega)$ is the normalized distance to collision when performing circular movements, calculated for polygonal robots as proposed by [19]. It uses the obstacle information from the occupancy grid described in Section III.

To avoid unsafe conditions while performing the obstacle avoidance, a similar consideration from [1] was applied to expand the robot neighborhood in the *dist* evaluation. The last function, $\text{velocity}(v, \omega)$ is calculated based on the desired robot linear velocity v_d (which is constant regarding to the road speed limit), as follow:

$$\text{velocity} = \begin{cases} \frac{v}{(v_d - v_{min})} & \text{if } v \leq v_d, \\ \frac{(v_{max} - v)}{(v_{max} - v_d)} & \text{if } v > v_d. \end{cases} \quad (4)$$

The importance of these previous functions in the objective function is adjusted by the constant gains α , β and γ .

2) *The DWA Search Space*: Initially, for the current vehicle velocity (v_a, ω_a) , the Dynamic Window V_d is defined for all reachable velocities in a time interval Δt as:

$$V_d = \{(v, \omega) \mid v \in [v_a - \dot{v}\Delta t, v_a + \dot{v}\Delta t], \omega \in [\omega_a - \dot{\omega}\Delta t, \omega_a + \dot{\omega}\Delta t]\}, \quad (5)$$

with $u_r = [v \ \omega]^T$ the set of robot inputs (see section II), and \dot{v} and $\dot{\omega}$ are the robot accelerations.

Once defined the reachable velocities, they must be classified in admissible or not due to the obstacle distance (function $\text{dist}(v, \omega)$ defined previously and proposed by [19]) and the robot maximum breaking accelerations $(\dot{v}_b, \dot{\omega}_b)$. The resulting set is defined as:

$$V_a = \{(v, \omega) \mid v \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{v}_b}, \omega \leq \sqrt{2 \cdot \text{dist}(v, \omega) \cdot \dot{\omega}_b}\}. \quad (6)$$

Finally, the Dynamic Window search space is computed as:

$$V_{DW} = V_d \cap V_a \cap V_s, \quad (7)$$

where V_s is the set of points that satisfy the maximum acceleration constraints \dot{v}_{max} and $\dot{\omega}_{max}$. It considers the current speed of the vehicle, its accelerations/physicals limits, and the obstacles in the workspace. By discretization of the search space V_{DW} , a velocity must be selected following the criteria presented by the objective function (3).

B. The Image Based Dynamic Window Approach

Considering the objective function (3) and the search space (7), the main changes for the Image Based Dynamic Window Approach (IDWA) concern to the function $\text{heading}(v, \omega)$. As previously mentioned, it is responsible to guide the robot to a desired goal in the world frame. For the present formulation, the goal is to lead the features set $s = [X \ Y \ \Theta]^T$, defined in Section III by the tangent Γ in the image frame $\{\mathcal{L}\}$ (see Figure 3), to the final configuration $X^* = \Theta^* = 0$ and $Y^* = Y_I$, which means the vehicle in the center of the road.

Based on the Image Based Visual Servoing (IBVS) equations proposed by Cherubini et al [15], the *heading* function must estimate the features error:

$$e = \begin{bmatrix} X_{t+\Delta t} - X^* \\ Y_{t+\Delta t} - Y^* \\ \Theta_{t+\Delta t} - \Theta^* \end{bmatrix},$$

$$L_s = \begin{bmatrix} \frac{-\sin \rho - Y \cos \rho}{t_z} & 0 & \frac{X(\sin \rho + Y \cos \rho)}{t_z} & XY & -1 - X^2 & Y \\ 0 & \frac{-\sin \rho - Y \cos \rho}{t_z} & \frac{Y(\sin \rho + Y \cos \rho)}{t_z} & 1 + Y^2 & -XY & -X \\ \frac{\cos \rho \cos^2 \Theta}{t_z} & \frac{\cos \rho \cos \Theta \sin \Theta}{t_z} & -\frac{\cos \rho \cos \Theta (Y \sin \Theta + X \cos \Theta)}{t_z} & -(Y \sin \Theta + X \cos \Theta) \cos \Theta & -(Y \sin \Theta + X \cos \Theta) \sin \Theta & -1 \end{bmatrix} \quad (2)$$

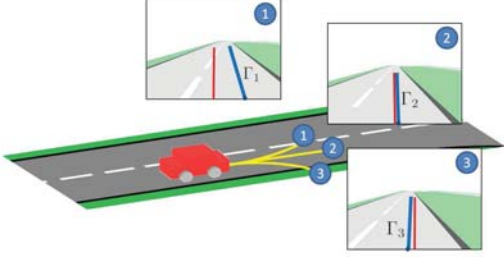


Fig. 4. Estimation of the features set Γ_i (blue line) in the frame $\mathcal{I}_{t+\Delta t}$ applying the control inputs (v_1, ω_1) , (v_2, ω_2) and (v_3, ω_3) . The reference position is also represented in red, which means the vehicle in the center of the road lane.

in the next image frame $\mathcal{I}_{t+\Delta t}$, considering (X^*, Y^*, Θ^*) as the set point. This is illustrated in the Figure 4. Thus, high weight are given to the inputs $(v_i, \omega_j) \in V_{DW}$ which reduce the final error e .

To this end, the controller must relate the image features velocity $\dot{s} = [\dot{X} \ \dot{Y} \ \dot{\Theta}]^T$ to the robot velocity $u_r = [v \ \omega]^T$. First of all, the image features velocity must be written in terms of the camera frame velocity $u_c = [v_{c,x} \ v_{c,y} \ v_{c,z} \ \omega_{c,x} \ \omega_{c,y} \ \omega_{c,z}]^T$. Using the interaction matrix $L_s(X, Y, \Theta)$ (2), expressed for a normalized perspective camera model, yields:

$$[\dot{X} \ \dot{Y} \ \dot{\Theta}]^T = L_s(X, Y, \Theta) u_c. \quad (8)$$

Note that each line of the matrix L_s are related to its respective image feature (L_X , L_Y and L_Θ). The robot velocity u_r can be expressed in the camera frame $\{C\}$ by (9) using the homogeneous transformation (10):

$$u_c = {}^C T_R u_r, \quad (9)$$

$${}^C T_R = \begin{bmatrix} 0 & -t_x \\ -\sin \rho & 0 \\ \cos \rho & 0 \\ 0 & 0 \\ 0 & -\cos \rho \\ 0 & -\sin \rho \end{bmatrix}. \quad (10)$$

The final features configuration can be acquired using the equations (8) and (9) to estimate the features velocity \dot{s} and integrating them over the time interval Δt . Cherubini et al has defined a row and column controller, depending of the $D = (X, Y)$ point location in the image frame $\{\mathcal{I}\}$ (see Figure 3). The row controller is applied when $Y = \text{const} = Y^*$ or the column one otherwise. Under this constraint, the function $\text{heading}(v, \omega)$ was divided in: $XY_{error}(v, \omega)$, responsible for the row/column error (X or

Y); and $\Theta_{error}(v, \omega)$ with the Θ error. The final values are calculated as:

$$XY_{error} = \begin{cases} 1 - \frac{|e_X|}{e_{Xmax}}, & \text{if row controller,} \\ 1 - \frac{|e_Y|}{e_{Ymax}}, & \text{otherwise.} \end{cases} \quad (11)$$

$$\Theta_{error} = 1 - \frac{|e_\Theta|}{\pi}. \quad (12)$$

where e_X , e_Y , and e_Θ are the features error in the image frame $\mathcal{I}_{t+\Delta t}$, and e_{Xmax} and e_{Ymax} are the maximum measurable error in X and Y . The final value is defined as:

$$\text{heading}(v, \omega) = \alpha_1 XY_{error}(v, \omega) + \alpha_2 \Theta_{error}(v, \omega). \quad (13)$$

V. EXPERIMENTAL RESULTS

To validate the navigation methodology proposed by the block diagram of Figure 1, a simulation environment using Matlab was created for some road configurations (see Figures 5a and 7a). The vehicle moves based on the kinematic model of the equation 1, respecting its kinematics constraints and some actuators dynamics. It simulates a monocular camera with a focal length $1.8mm$ and large field of view ($\simeq 140^\circ$). The camera tilt offset $\rho = 10^\circ$ and $(t_x, t_y, t_z) = (2.0, 0.0, 2.0)m$. The obstacles are detected by a simulated laser sensor with 180° of coverage in front of the vehicle and the information was stored in an occupancy grid [18], locally constructed around the robot. Each laser reading is represented in the occupancy grid by a bidimensional Gaussian model. The relative movement of the robot frame updates the grid information, using its proprioceptive data, like odometry (in real environments the visual odometry can be applied), velocity and steering angle, which is enough for the simulation purposes and low speed experiments.

The Image based Dynamic Window Approach (IDWA) must considers the road limits, obstacles and linear velocity variations when adjusting the gains α , β , and γ from equation 3. When no obstacle obstructs the robot path, the movement is similar to the one in Figure 5b. To visualize their influence in the final navigation, the Figure 6(a-c) combines them one by one. In the Figure 6a only the function velocity was applied, resulting in a movement with no direction that stops at the first visible obstacle. Enabling the function dist , it gives the Figure 6b with a movement over regions free of obstacles but without any goal, better observed in the curves. Adding the heading functions $vs1$ and $vs2$ to the previous ones, it results the Figure 6c. Here the robot avoid the obstacles and follow when possible the

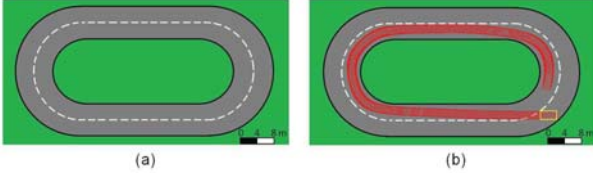


Fig. 5. Environment for car-like robot simulation (a) and its navigation in this condition (b). The car initial pose is represented in yellow, and in red are the car instantaneous positions for a clockwise movement.

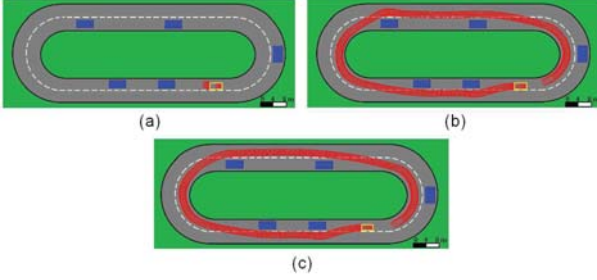


Fig. 6. Evaluation of the IDWA functions in the navigation task for: *velocity* (a), *velocity + dist* (b), and the complete objective function with the *heading* functions *vs1* and *vs2* (c). The gains where set to $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

right road lane, guaranteed by our image based task. For this final configuration: $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. It is important to observe that even using only one function in the objective function, the final movement will be safe from collisions, once we are using only the velocities from the DW search space in 7. An extended experiment is presented in Figure 7.

The influence of changing the road lane setpoint when overtaking an obstacle was verified in Figure 8. Comparing the Figures 8a, with the setpoint ways on the right lane, and 8b, where the setpoint changes due to the lane obstruction, the second one presents a smoothest trajectory and farther from the obstacles than the first one. This is better for real world applications, providing more safety for the vehicle movement.

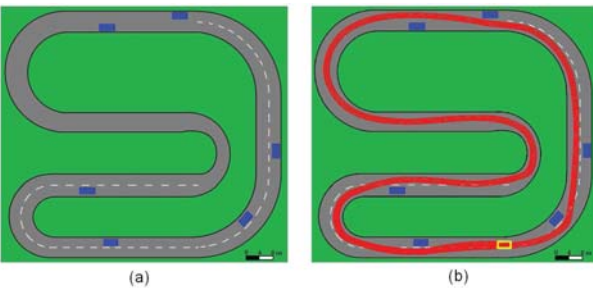


Fig. 7. Environment for car-like robot simulation (a) and its navigation in this condition (b), with $\alpha_1 = \alpha_2 = 0.01$, $\beta = 0.2$, and $\gamma = 0.3$. In the cases where do not have the road markers, the vehicle follows the road center. The car initial pose is represented in yellow and starts the movement to the left. The obstacles are in blue, and in red are the car instantaneous positions.

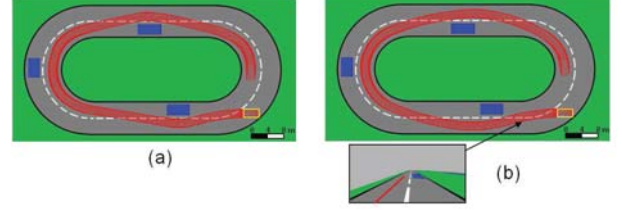


Fig. 8. Vehicle movement with the setpoint defined only on the right lane (a) and switching due to the lane obstruction (b), illustrated by the red line in the camera image from the car point of view. The car initial pose is represented in yellow, the obstacles are in blue, and in red are the car instantaneous positions for a clockwise movement.

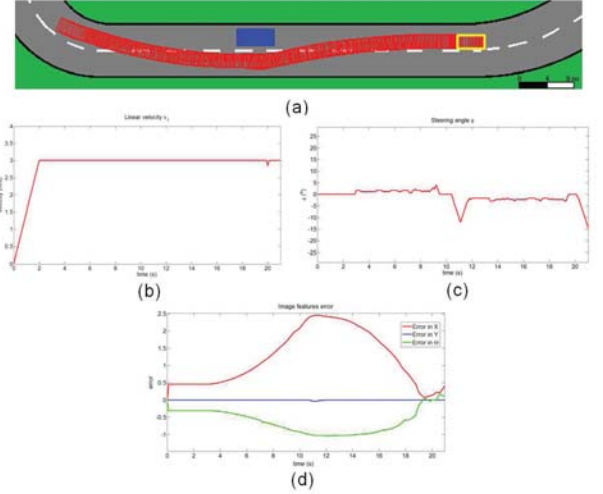


Fig. 9. IDWA controller outputs and image error evolution for the trajectory represented in (a). In (b) is the linear velocity and (c) the steering angle commands. The image features errors are in (d).

Finally, the Figure 9 shows the resulting commands and the image features error evolution during the first 21 seconds of simulation using the IDWA controller. Note that, even if the IDWA is a discrete technique, the output commands are smooth for real cars actuators. This can be better observed in the steering angle (Figure 9c), rarely reaching values higher than $\pm 10^\circ$ in the complete circuit, which results in more comfort for a final user. The image features error converge smoothly to zero when there are no obstacles preventing the vehicle movement, as seen in Figure 9d.

VI. CONCLUSIONS AND FUTURE WORKS

This work presented an image based local navigation approach applicable to car-like robots in urban environments among obstacles. For that, it was integrated the Image-Based Visual Servoing (IBVS) equations, originally conceived to follow lines on the floor with a small features set, in the Dynamic Window Approach (DWA), resulting in a local navigation system independent of the vehicle global localization and the final destination. The Image Based Dynamic Window Approach (IDWA), as it was called here, allowed the robot to perform the path reaching, to follow the road lane center avoiding obstacles. The methodology was tested in a simulation environment over Matlab considering the car

kinematics and some dynamics constraints, and sensors limitations, which provided a solid validation for the proposed solution in a static environment.

The optimization function of the DWA taken in to account the path reaching and following problem, the obstacles avoidance, and the linear velocities variations, in order to find the control inputs that best attend the gains setup. However, find the best adjustment for these gains depends of which element must be considered preferentially, if is to follow the road lane, to keep the distance from the surrounding obstacles, or moves with higher velocities. For the current configuration, the vehicle was able to complete its navigation task with smooth control inputs. The present approach also left us switch the setpoint between the lanes during the overpass maneuver. It is important to mention that other IBVS equations could be integrated in the present solution to allow different tasks.

Thanks to the nature of the elements analyzed, the application of this solution in a real car-like robot could be done with low cost sensors. The experiments are being prepared to the autonomous car Iris of the projects VERVE and ROBOTEX, from the Heudiasyc laboratory. Considering the real environments, more robust techniques must be applied to detect the image features, like the one proposed in [20], as well as the moving obstacles like in [12]. Other applications in mind are related to the human machine interface, improving the interaction between the vehicle and its conductor.

REFERENCES

- [1] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *J. Field Robot.*, vol. 25, no. 9, pp. 640–673, Sep. 2008. [Online]. Available: <http://dx.doi.org/10.1002/rob.v25:9>
- [2] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008. [Online]. Available: <http://dx.doi.org/10.1007/s10846-008-9235-4>
- [3] O. Hassan, I. Adly, and K. Shehata, "Vehicle localization system based on ir-uwv for v2i applications," in *Computer Engineering Systems (ICCES), 2013 8th International Conference on*, Nov 2013, pp. 133–137.
- [4] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313–326, 1992.
- [5] S. Lee, K. Boo, D. Shin, and S. Lee, "Automatic lane following with a single camera," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, May 1998, pp. 1689–1694 vol.2.
- [6] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, 1997.
- [7] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 341–346.
- [8] P. Ogren and N. Leonard, "A convergent dynamic window approach to obstacle avoidance," *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 188–195, April 2005.
- [9] P. Saranrittichai, N. Niparnan, and A. Sudsang, "Robust local obstacle avoidance for mobile robot based on dynamic window approach," in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, May 2013, pp. 1–4.
- [10] K. Rebai, O. Azouaoui, M. Benmami, and A. Larabi, "Car-like robot navigation at high speed," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, 2007, pp. 2053–2057.
- [11] K. Rebai and O. Azouaoui, "Bi-steerable robot navigation using a modified dynamic window approach," in *Proceedings of the 6th International Symposium on Mechatronics and its Applications*, 2009, pp. 1–6.
- [12] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2007, pp. 1986–1991.
- [13] D. Lima and G. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *Journal of Control, Automation and Electrical Systems*, pp. 1–11, 2013. [Online]. Available: <http://dx.doi.org/10.1007/s40313-013-0006-5>
- [14] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.
- [15] A. Cherubini, F. Chaumette, and G. Oriolo, "Visual servoing for path reaching with nonholonomic robots," *Robotica*, vol. 29, pp. 1037–1048, 12 2011. [Online]. Available: http://journals.cambridge.org/article_S0263574711000221
- [16] A. Cherubini, F. Spindler, and F. Chaumette, "A new tentacles-based technique for avoiding obstacles during visual navigation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, May 2012, pp. 4850–4855.
- [17] A. D. Luca, G. Oriolo, A. De, and C. Samson, *Robot Motion Planning and Control*. Springer Berlin / Heidelberg, 1998, vol. 229, ch. Feedback Control Of A Nonholonomic Car-Like Robot, pp. 171–253.
- [18] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.
- [19] K. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-time obstacle avoidance for polygonal robots with a reduced dynamic window," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, 2002, pp. 3050–3055.
- [20] G. B. Vitor, D. A. Lima, A. C. Victorino, and J. V. Ferreira, "A 2d/3d vision based approach applied to road detection in urban environments," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 952–957.

Towards Lifelong Learning of Optimal Control for Kinematically Complex Robots*

Alexander Dettmann¹, Malte Langosz², Kai von Szadkowski¹, and Sebastian Bartsch²

Abstract—Robots intended to perform mobile manipulation in complex environments are commonly equipped with an extensive set of sensors and motors, creating a wide range of perception and interaction capabilities. However, to exploit all theoretically possible abilities of such systems, a control strategy is required that allows to determine and apply the best solution for a given task within an appropriate time frame. In this paper, a lifelong self-improving control scheme for kinematically complex robots is presented, which uses simulation-based behavior generation and optimization procedures to create a library of well-performing solutions for varying tasks and conditions, and combines it with case-based selection, evaluation, and online adaptation methods.

I. INTRODUCTION

Behavior-based systems are best suited for changing environments, where fast response and adaptivity are crucial [1]. Their distributed nature increases fault tolerance and promotes component reuse as well as distributed development [2], [3]. However, coordination effort of behaviors increases with system complexity, making the control of behavior-based systems a challenging task. This is especially true for kinematically complex robots such as walking machines, whose different locomotion patterns, postures, or reflexes will be more or less efficient depending on the context, i.e., the external environment, the internal state, and the actual task. Current systems react to context changes by tuning parameters of their existing behaviors [4], [5], [6], but work on online integration of new behaviors for completely new contexts is rather sparse.

As future robots will be required to act more and more autonomously in well-known as well as in novel environments, it is important to equip robotic systems with tools to efficiently adapt to unknown contexts. For mobile platforms it is thus desired to use machine learning algorithms and/or simulation methods to create new sets of behaviors for situations in which none of the predefined locomotion behaviors is well suited. Such newly-derived behaviors will then have to be directly included in the running robot control to be utilized on the spot.

*The presented work was carried out in the project LIMES, a collaboration between the DFKI Robotics Innovation Center and the University of Bremen, funded by the German Space Agency (DLR, Grant numbers: 50RA1218, 50RA1219) with federal funds of the Federal Ministry of Economics and Technology (BMWi) in accordance with the parliamentary resolution of the German Parliament.

¹Alexander Dettmann and Kai von Szadkowski are with the Faculty of Mathematics and Computer Science, University of Bremen, 28359 Bremen, Germany `firstname.lastname@uni-bremen.de`

²Sebastian Bartsch and Malte Langosz are with the German Research Center for Artificial Intelligence - Robotics Innovation Center (DFKI RIC), 28359 Bremen, Germany `firstname.lastname@dfki.de`

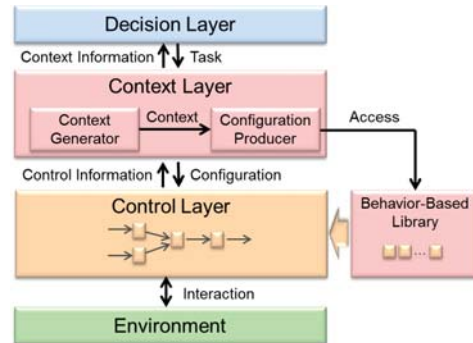


Fig. 1. Context-based control approach

Commonly a *decision layer* is used on top of a *control layer*, with the former generating tasks derived from deliberative planning processes and the latter executing these by producing appropriate target values for the actuators. This structure leaves a gap, as higher deliberative planning algorithms cannot take all configuration possibilities of the control layer into consideration. Consequently, the full potential of a robot is not used or an operator has to be included in the loop to tune the system according to the changing contexts in which the robot finds itself in.

As a solution for these problems, an intermediate layer is proposed which configures the control layer autonomously according to the inputs from higher and lower layers (Fig. 1). This *context layer* receives commands specifying the desired task from the decision layer, e.g., “move forward at a certain speed and as stable as possible”, as well as data defining the current state from the control layer, e.g., “hard ground with small scattered obstacles”, and builds up a *context* representation. Based on this *context*, the best-known configuration of the control layer is retrieved from a *behavior-based library* (BBL) and applied on the robot. This encapsulation of the control layer provides a robot-independent interface to higher levels, allowing a far more abstracted design of the deliberative control layer.

Section II describes the BBL and its components followed by a description of how optimization in simulation can generate additional competence (Section III). The selection principle to choose the best configuration of the control layer according to the current context and the data available in the BBL is described in Section IV. Section V details a simple example for how the proposed approach can be used to improve the performance of the robot by integrating externally generated knowledge. In the last section, a brief conclusion and an outlook are provided.

II. BEHAVIOR-BASED LIBRARY

The BBL basically represents the memory of a robot. On the one hand, it consists of solutions in form of algorithms and parameterizations, which both together result in *configurations* of the control layer. On the other hand, it holds all information required for proper selection or even creation of new solutions, i.e. performance evaluations of configurations in diverse contexts.

The BBL can easily be used in conjunction with robotic frameworks such as ROS¹ and Rock², and thus its functionality can be integrated in a large number of diverse robotic systems.

A. Behavior Representation

In reactive control approaches, the overall robot behavior emerges from the interaction of several behavior producing modules, simply called *behaviors*. In the proposed approach two behavior types are available, *graph behaviors* and *parameter behaviors*.

Graph behaviors are defined as nodes, as described in [7], mapping a defined number of input ports to a defined number of output ports (Fig. 2). This mapping is realized via a graph of behavior producing modules, which are either atomic transfer functions or themselves graph behaviors. Since both are using the same interfaces, there is no need for special handling in the different layers, thus allowing a hierarchical decomposition of behaviors. Possible atomic transfer functions include direct mapping of input to output, various mathematical such as trigonometric functions or conditional branching.

The signals used in these graphs are tuples of values and accompanying weights, allowing flexible interactions of the modules. Both outputs and inputs may be connected to multiple other ports, however in the latter case, one of various available merge functions is used to calculate a resulting input value for each input port. If an input is not connected a default value is used instead.

Behavior modules and their individual input and output ports can be named and annotated with additional meta information in the form of textual descriptions. This allows to attach a description or information on a behavior's suitability for a certain scenario, providing higher control layers or a human operator with necessary information to select an appropriate behavior for a given context. Annotations for ports can take the form of units and type information as well as expected input and output ranges. The latter allow the usage of interval analysis [8] for consistency checking.

The flexibility of this design and the common interface of graph behaviors, no matter how complex internally, opens many possibilities. For instance, any algorithm for which numerical inputs and outputs can be specified can be wrapped in a graph behavior module and thus be integrated in the overall behavior of a robot on any level of the graph hierarchy. Tools for designing new and altering existing behaviors

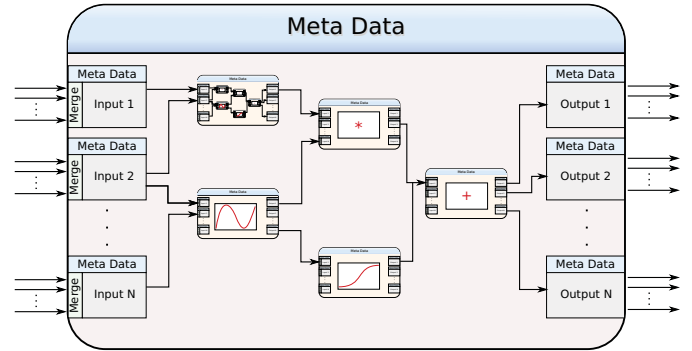


Fig. 2. Graph behavior consisting of a network of atomic transfer functions and another graph behavior

<pre> nodes: - id: 1 type: "[PIPE]" inputs: idx: 0 type: "[PRODUCT]" default: 1.0 outputs: idx: 0 - id: 6 type: "[OUTPUT]" inputs: idx: 0 type: "[SUM]" </pre>	<pre> default: 1.0 outputs: idx: 0 networkInputIds: - 1 edges: - fromNodeId: 1 fromNodeOutputIdx: 0 toNodeId: 6 toNodeInputIdx: 0 weight: 2.5 </pre>
--	--

Fig. 3. An example of a graph behavior in YAML (www.yaml.org) format is shown. First, two nodes, one pipe and one output node are defined, followed by the declaration of the network's input nodes and the definition of an edge from the input node to the output node.

can be built with little effort. One resulting advantage is the possible application of machine learning algorithms to change or optimize an algorithm by modifying the structure and parameters of the corresponding behavior graph; similar work is successfully done in the field of neuroevolution or genetic algorithms [9], [10], [11]. The modular structure simplifies testing, as a generic test suite can be used to validate that a behavior operates in a given output range and does not contain singularities such as divisions by zero. Finally, due to the annotation with meta information, even complex graph behavior can be saved in human readable files such as shown in Fig. 3.

The second concept of behaviors used in the BBL is that of parameter behaviors which parameterize the algorithmic graph behaviors, significantly influencing the emergent robot behavior; e.g., the same walking pattern generator can be switched from generating patterns for obstacle covered slopes to patterns suitable for plain soft soil simply by assigning different parameters. A parameter behavior can be imagined as a behavior module having no inputs but outputs with default values or as a parameter list as depicted in Fig. 4(a).

B. Context-Based Behavior Evaluation

To be able to use the best-suited behaviors in the right situation, their performance in diverse contexts needs to be

¹<http://www.ros.org>

²<http://rock-robotics.org>

<pre> name: planar_pattern type: parameter behavior parameters: walking_speed: SpeedX: 50 LengthFactor: 0.51 LiftTime: 200 ShiftTime: 1400 TouchdownTime: 200 PhaseShift: 0.0 posture: BodyHeight: 0.26 BodyShiftX: 0.1 BodyLean: 10.0 LegWidth: 0.42 </pre>	<pre> name: planar_rigid_test3 topology: topology_0.yml merge type: wta behaviors: planar_rigid_pattern3: 1 context evaluations: - setup: real evaluations: 53 state: planar_rigid_noSlope.yml performance: velocity x: [45, 3, mm/s] velocity y: [5, 1, mm/s] turn rate: [0, 0, °/s] body height: [250, 0, mm] body width: [890, 0, mm] ssm: [145, 0, mm] epd: [0.73, 0.1, Wh/m] power consumption: [119, 7, W] </pre>
(a) Parameter behavior	(b) Configuration evaluation

Fig. 4. Example descriptions of components of the BBL

known. Therefore, this information is stored in the BBL as well (Fig. 4(b)). *Configuration evaluations* can be generated from experiments with the real robot or derived from optimization results in simulation. They hold information about:

- the overall behavior graph (topology of behaviors)
- applied parameter behaviors and their merge method
- evaluation information for each encountered context
 - setup information (useful to filter out undesired experiences)
 - number of evaluations
 - all state context features
 - mean and standard deviation of all performance features

III. SIMULATION-BASED BEHAVIOR GENERATION AND OPTIMIZATION

The efficacy of choosing context-based behaviors strongly depends on the quality of the underlying behavior database. This refers to both its extent, i.e. the number of contexts for which a suitable behavior is available, and its elaborateness, i.e. how well-tested stored behaviors are and how reliably they perform. As it is virtually impossible to optimize behaviors over a large number of experimental cycles on the actual robotic system, the BBL is built not only by the robot itself, but is also fed with simulation results. For this, the open-source physical simulation environment MARS³ is utilized which is based on the Open Dynamics Engine (ODE)⁴. Using MARS together with an integrated framework for behavior learning, testing of vast sets of possible behavioral solutions to problems posed by various contexts and refinement of the resulting behaviors with sophisticated optimization algorithms is made possible within reasonable constraints of time and resources. This is further assisted by the use of the meta information provided by the behaviors being optimized, which allows to restrict the search space by pre-defining dependencies, parameter ranges and other properties of the respective modules.

³MARS (Machina Arte Robotum Simulans) is available on <https://gitorious.org/rock-simulation/mars>

⁴<http://www.ode.org/>

In order to yield useful results, the simulation model of the robot model as well as the characteristics of the simulated environments have to be sufficiently accurate representations of their real-world counterparts. This necessitates measurements of the robot’s single components’ physical behavior as well as including environmental factors such as soil dynamics in the simulation. Still in most cases, behaviors evolved in simulation will have to be adapted to be used on real robots due to the remaining simulation-reality gap. Even given these difficulties, developing behaviors in simulation still constitutes a dramatically reduced effort when new strategies are required in novel or changed contexts as compared to developing said strategies on the real system from scratch. Moreover, simulation allows to discard erroneous or unfit parameter sets that might result in malfunction or damage of the robot, providing a mechanism of safety-relevant quality control. This is true for both offline-development of novel behaviors in simulation as well as for online-testing of newly-derived behaviours in the current context of a robot, an approach becoming more and more feasible with the constant increase in processing power.

IV. ONLINE BEHAVIOR SELECTION AND ADAPTATION

Kinematically complex robots in real world scenarios have a tremendous amount of possibilities to solve certain tasks. Consequently, especially at the beginning of the lifetime of a robot, only sparse domain knowledge and anecdotal experience are available. Thus, case-based reasoning (CBR) is used to infer the best-suited configuration of the control layer for a certain context. This artificial intelligence paradigm solves problems by reusing experiences from similar, previously solved problems [12]. A *case* consists of two parts. The first part represents a *solution*, which is in the proposed application a configuration of the control layer. The second part holds a list of E performed configuration evaluations, where each entry describes the number of evaluations, the evaluation setup, the state context, and the evaluated performance.

The BBL will contain several case bases which are used in different scenarios, e.g., a six-legged locomotion scenario and a manipulation scenario have different case bases. This refines the case retrieval step described in the following section. In addition, it can be beneficial to introduce new case bases when something unpredictable occurs (as proposed in [13]), e.g., a malfunction of a leg. Then, a corresponding five-legged locomotion case base could be created with some initial cases from the six-legged case base as a starting point. These cases could then be adapted and optimized to fit the current conditions. The BBL has the opportunity to load, use, and store cases or switch entire case-bases according to the current task.

In the proposed application of adaptive robot control, the input problem is described by two feature vectors describing a robot’s current state \mathbf{S}^{cur} with M state features ($s_1^{cur}, \dots, s_M^{cur}$) and task \mathbf{T}^{cur} with N performance features ($p_1^{cur}, \dots, p_N^{cur}$), where each feature is a tuple consisting

of value and weight. This format matches the description of configuration evaluations in the BBL, with the notable difference that the library contains mean and standard deviation values for the performance features p_n^{ref} . To decide which behavior configuration to deploy, the CBR algorithm processes this input in multiple steps (Fig. 5) which are very common for many CBR systems and are motivated from [12]. First, the similarity of the provided context to previously tested cases is evaluated, resulting in a number of candidate behaviors. Then, a *ballpark solution* for the given problem is derived from this set and adapted if necessary according to the reigning conditions. The resulting behavior configuration is checked one last time before execution to avoid mistakes from the past. During application, the performance is evaluated. Finally, the gained experiences are stored in the corresponding case base. The algorithmic details are described in the following subsections.

A. Case Retrieval

In the first step of CBR, all configuration evaluations in the BBL are rated according to the input query. First, the state and performance features are normalized according to robot-specific limits, before a similarity measure is used to determine how well a case from the case base matches the input query. Since the input query consists of two feature vectors of variable length, a multi-stage approach is proposed: First, the state similarity Sim_e^{State} between the current state features s_m^{cur} and stored reference state features s_m^{ref} is calculated for each evaluation of each case, then the task similarity Sim^{Task} is computed for the library entry with the highest state similarity, and finally the overall case similarity Sim is calculated.

In the first step, the weighted mean square error is used, as it is more sensitive to large differences of one single feature than to small differences of several features compared to the weighted mean absolute error. The error is subtracted from one (since normalized values are used) to get the state similarity for each evaluation $e \in E$ (1).

$$Sim_e^{State} = 1 - \frac{\sum_{m=1}^M (s_m^{cur} - s_m^{ref})^2 \cdot w_m^S}{\sum_{m=1}^M w_m^S} \quad (1)$$

The weights for each feature variable w_m^S are used to include the confidence of the corresponding context feature estimation. Alternatively, they could be used to model the features' importance, which can be learned to improve the case retrieval [14]. If a state similarity has to be calculated for a current state feature not listed in the evaluation, two solutions are possible. The safest way is to set the corresponding feature state similarity to zero. A more curious strategy would be to set it to one. Finally, the state similarity for the entire case Sim^{State} is represented by the maximum state similarity of the E evaluations (2).

$$Sim^{State} = \max(Sim_1^{State}, \dots, Sim_E^{State}) \quad (2)$$

The task similarity is calculated for the evaluation with the highest state similarity, also using the weighted mean square

error (3),

$$Sim^{Task} = 1 - \frac{\sum_{n=1}^N (p_n^{cur} - p_n^{ref})^2 \cdot w_n^P}{\sum_{n=1}^N w_n^P} \quad (3)$$

where w_n^P are the weights of a performance ratio, which the operator or higher layers can define to influence the robot behavior. Therefore, the vector of performance features p_n^{cur} contains the task-depending features (desired motion, ...) and the evaluation features (stability, energy efficiency, ...). The latter stay constant at their best value, e.g., the best stability value would be one whereas the power consumption would be zero. Finally, the overall similarity Sim of a case is the product of both single similarities (4).

$$Sim = Sim^{State} \cdot Sim^{Task} \quad (4)$$

After determining the similarity of each case, the k nearest neighbors (K-NN) are chosen, yielding a limited set of candidates for the next processing stage. The K-NN are limited by number and also have to reach a pre-defined relative similarity threshold.

B. Ballpark Solution Proposal

In this step, a case solution or parts of several solutions are extracted to form a temporary solution to be used in subsequent processing stages. A number of methods are available for this, including choosing the most similar case [15], drawing randomly from K-NN [16], or drawing from K-NN according to the degree of similarity [17]. While the first method simply selects the best known solution, the second and third method avoid overusing one particular solution, which might be beneficial in situations where the solution with the highest similarity does not necessarily result in the best performance. Deriving the average of K-NN solutions is also common in case-based regression. Beyond these methods, other merging techniques such as the application of machine learning to explore new solutions can be imagined as well.

C. Case Adaptation

Once a ballpark solution has been obtained, it is adapted to fit the current needs. Here, two adaptation variants exist: constant adaptation and continuous adaptation. Constant adaptation can be applied to vary the ballpark solution by taking the dissimilarity between input query and ballpark solution into consideration to enhance the expected performance. Some rule-based approaches exist which use deep domain knowledge to infer adaptation rules. However, this leads to losing the generality of the overall approach. Instead, so-called "knowledge-light" approaches use the implicit knowledge of the case base to infer adaptations. For instance, case difference heuristic approaches build adaptation rules by comparing pairs of cases and identifying their context and solution differences. The resulting mappings between incoming context difference and resulting solution adaptation are then scored according to some gradient [18], [19] or covariance metrics [20].

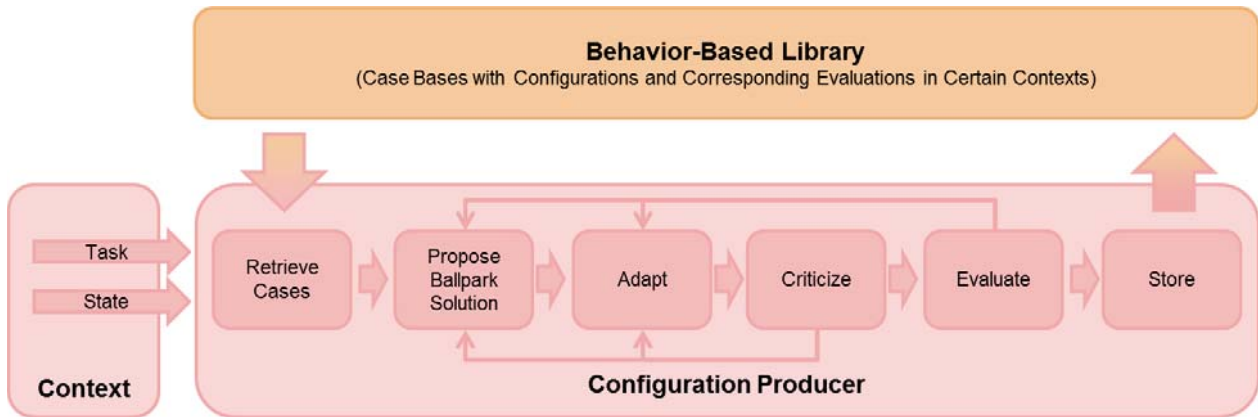


Fig. 5. Processing steps of the case-based reasoning system

Continuously adapting algorithms on the other hand can be applied in environments with low fluctuations to find local performance maxima. Simply adding noise [15], [16] or crouching and inverting a randomly initialized adaptation vector [17] can lead to increasing system performance. Some more sophisticated machine learning approaches such as CMA-ES [21], REPS [22] or PSO [23] could certainly improve the results. In addition, they could be used to handle the simulation-reality gap. Imagine a behavior configuration evolved through optimization in simulation is chosen as a ballpark solution and finally applied. Because of the simulation-reality gap, the performance will most certainly differ, but probably still be close to the actual (local) optimum. Thus, a lazy, fast-converging learning algorithm could most likely be used to adapt the simulation solution to reality. The required feedback would in this case be generated in the evaluation step.

D. Case Criticism

Before the adapted solution is applied on the system, it is *criticized*. In this processing step, solutions can be rejected which have already been chosen and tested before but did not perform well, thus avoiding known failures. If this happens, a new solution has to be provided, repeating the algorithm's previous subroutines. This step is sparsely used in literature. In [17] a case switching tree is used to recover from overused cases which do not improve the situation. Here, it is also advantageous to predict the performance of the chosen solution. This information can help in later steps to analyze the outcome of applying the generated solution.

E. Evaluation

The evaluation of a solution itself is separated from the CBR algorithm since computation of the performance features is robot-specific. The resulting performance vector only has to match the case description. As mentioned before, the results of the evaluation are needed in the other processing steps. Useful performance metrics are energy efficiency, stability, precision of task realization, processing time, or other control-specific information.

F. Memory Storage

The last stage of a CBR system is the memory update, which incorporates updating performance values of known cases (configuration evaluations) or creating new cases (configurations and their evaluations) if a new solution was applied. Through gathering of new experiences, the robot gets the opportunity to learn, i.e. increasing its performance and competence. In addition, constantly updating the case base incorporates wear out of the system.

V. PROOF OF CONCEPT

In order to show the possibility to include externally generated knowledge in a robot control, the following experiment was conducted in simulation and reality. The SpaceClimber [24] robot was set up to walk on a plane for 30 s as energy-efficient as possible at a given speed of 50 mm/s. In a first step, SpaceClimber's BBL consisted of one configuration evaluation holding performance information of one graph behavior in combination with one parameter behavior. The latter (Table I) was created with expert knowledge and has shown good results in previous experiments [24]. During context-dependent configuration of the control layer, this solely available configuration evaluation was of course most similar to the experimental context, which led to the application of the corresponding graph and parameter behavior.

In the second step of the experiment, a new parameter behavior was created in simulation, using a CMA-ES optimization aiming for energy efficiency of the previously-tested graph behavior in the same context of walking on plane ground. The resulting parameter behavior (Table I) was stored in the BBL along with the corresponding configuration evaluation (performance based on optimization fitness). When repeating the 30 s walk, the optimized walking pattern was selected by the configuration producer and applied on the robot control because of its better performance in the same context.

The resulting power consumption in both scenarios, simulation and reality, are depicted in Fig. 6. It is visible that the power consumption in simulation is higher than on the real system indicating a gap between simulation and reality.

TABLE I

PARAMETERS OF USED WALKING PATTERNS (ALL OTHER PARAMETERS ARE KEPT AT THEIR DEFAULT VALUE AND ARE OMITTED FOR CLARITY)

pattern	plane, handcrafted	plane, optimized
body shift x in mm	0	100
body height in mm	250	275
speed x in mm/s	50	50
length factor [0...1]	0.5	0.5
lift time in ms	200	240
shift time in ms	1400	1560
touchdown time in ms	200	1600
phase shift [0...1]	0.0	0.0
swing amplitude in mm	100	100

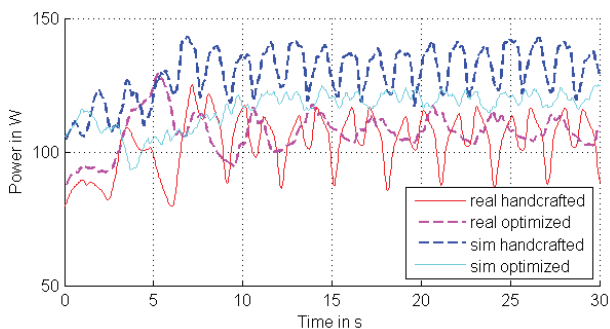


Fig. 6. Power consumption while walking

While in simulation, the power consumption was lower with the optimized locomotion pattern, this was not the case with the real system. However, the optimized walking pattern traversed 1,38 m during the 30 s while the handcrafted traversed 1,23 m. Consequently, the resulting energy per distance was lower for the optimized locomotion pattern. The main reason is that with the optimized pattern the feet are placed more smoothly due to higher touchdown time resulting in less slippage. Although the movement was improved, the walking behavior on the real system was not optimal since the power consumption was not less as indicated from the simulation comparison.

VI. CONCLUSIONS AND OUTLOOK

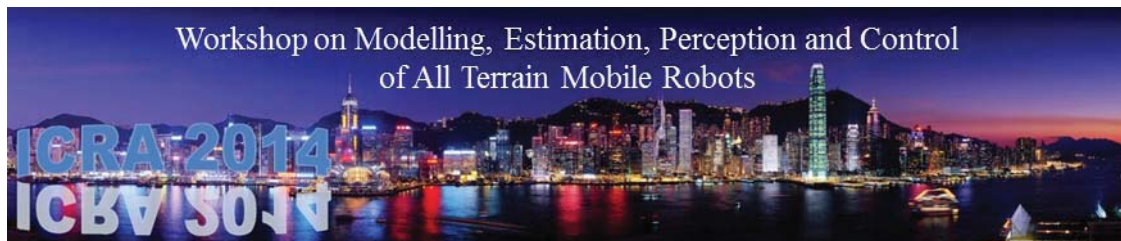
In this paper, a control scheme for kinematically complex robots is presented, which uses a BBL to store possible control configurations and their performances in varying contexts. Real world experiences and optimization results build the knowledge base which is continuously growing during life-time increasing the robot's performance and competence. A case-based reasoner is used to find the best-known control configuration for a given context. The given example of increasing the energy efficiency of a walking pattern is a rather simple problem. In future, the scalability of this approach for complex problems have to be discussed. In addition, the generation of new solutions through intelligent case merging and adaptation as well as the storage of experiences need to be analyzed. Though, in simulation optimized behaviors can improve the performance of the real system, an online optimization on the real system will be needed to handle the simulation reality gap.

ACKNOWLEDGMENT

Special thanks are due to all team members of the project LIMES.

REFERENCES

- [1] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2008.
- [2] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14–23, 1986.
- [3] R. Arkin, *Behavior-based robotics*. MIT press, 1998.
- [4] J. Albiez, *Verhaltensnetzwerke zur adaptiven Steuerung biologisch motivierter Laufmaschinen*. GCA-Verlag, 2007.
- [5] B. Gassmann, "Modellbasierte, sensorgestützte navigation von laufmaschinen im gelände," Ph.D. dissertation, University Karlsruhe (TH), 2007.
- [6] F. Michaud, "Selecting behaviors using fuzzy logic," in *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*. IEEE, 1997, pp. 585–592.
- [7] M. Langosz, L. Quack, A. Dettmann, S. Bartsch, and F. Kirchner, "A behavior-based library for locomotion control of kinematically complex robots," *Proceedings of the 16th International Conference on Climbing and Walking Robots, (CLAWAR-2013)*, pp. 495–502, Aug. 2013.
- [8] R. E. Moore, *Interval Analysis*. Prentice Hall, 1966.
- [9] L. Cardamone, D. Loiacono, and P. L. Lanzi, "Learning to drive in the open racing car simulator using online neuroevolution," *IEEE Trans. Comput. Intellig. and AI in Games*, pp. 176–190, 2010.
- [10] Y. Kassahun, J. de Gea Fernández, M. Römmermann, and F. Kirchner, "On applying neuroevolutionary methods to complex robotic tasks," in *IEEE IROS Workshops on Exploring new horizons in Evolutionary Design of robots*, 2009, pp. 26–30.
- [11] M. Römmermann, M. Ahmed, L. Quack, and Y. Kassahun, "Modeling of leg soil interaction using genetic algorithms," in *Proceedings of International Conference of the International Society for Terrain-Vehicle Systems*, 2011.
- [12] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.
- [13] D. Leake and R. Sooriamurthi, "When two case bases are better than one: Exploiting multiple case bases," *Case-Based Reasoning Research and Development*, pp. 321–335, 2001.
- [14] S. Gunawardena, R. Weber, and J. Stoyanovich, "Learning feature weights from positive cases," *Case-Based Reasoning Research and Development*, pp. 134–148, 2013.
- [15] A. Ram and R. Arkin, "Case-based reactive navigation: a method for on-line selection and adaptation of reactive robotic control parameters," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, no. 3, pp. 376–394, 1997.
- [16] M. Likhachev and R. Arkin, "Spatio-temporal case-based reasoning for behavioral selection," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1627–1634, 2001.
- [17] M. Likhachev, M. Kaess, and R. Arkin, "Learning behavioral parameterization using spatio-temporal case-based reasoning," *IEEE International Conference on Robotics and Automation*, 2002.
- [18] N. McDonnell and P. Cunningham, *A knowledge-light approach to regression using case-based reasoning*. Springer Berlin Heidelberg, 2006, vol. 4106.
- [19] V. Jalali and D. Leake, "A context-aware approach to selecting adaptations for case-based reasoning," *Modeling and Using Context*, pp. 101–114, 2013.
- [20] —, "Extending case adaptation with automatically-generated ensembles of adaptation rules," in *Case-Based Reasoning Research and Development*. Springer, 2013, pp. 188–202.
- [21] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, pp. 159–195, 2001.
- [22] M. P. Deisenroth, G. Neumann, J. Peters, et al., "A survey on policy search for robotics," *Foundations and Trends in Robotics*, 2013.
- [23] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995.
- [24] S. Bartsch, "Development, control, and empirical evaluation of the six-legged robot spaceclimber designed for extraterrestrial crater exploration," Ph.D. dissertation, University of Bremen, 2013.



Session IV

Cooperative Robots, Software architecture

- **Invited Talk: Roland Lenain (IRSTEA, France)**

Title: Formation control of off-road fleet of UGVs: issues, advances and applications

Co-authors: A. Guillet, B. Thuilot, C. Cariou

- **Title: Integrating ABSYNTHE autonomous navigation system into ROS**

Authors: A. Llamazares, E. Molinos, M. Ocana and F. Herranz

- **Title: The Artemis Rover as an Example for Model Based Engineering in Space Robotics**

Author: J. Schwendner, T. M. Roehr, S. Haase, M. Wirkus, M. Manz, S. Arnold and J. Machowinski

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICRA 2014





Session IV

Invited Talk: **Roland Lenain**

(IRSTEA, France)

Formation control of off-road fleet of UGVs: issues, advances and applications

Co-Authors: A. Guillet, B. Thuilot, C. Cariou

Abstract: The control of several mobile robots in a coordinated global motion has several potential applications, especially in an off-road context: defense, environment monitoring or agriculture... Nevertheless, despite the important progress in robotics and autonomous driving, some key issues are still challenging and subjected to development. In particular, the varying conditions encountered in natural environment, influences perception, communications and robot dynamics, and have then to be addressed. The needed accuracy, reliability and safety indeed require developing and using robust and adaptable control architecture. This talk first introduces to problems arising in off-road context when achieving the control of several robots in cooperation. Some control strategies and ongoing development are then presented in order to preserve the accuracy and reliability. Experimental illustrations permit to consider short and long term applications. Finally, open issues and perspectives close the talk.

Biography: Roland LENAIN is a research fellow in Irstea on the topic of off-road mobile robotics. His research interests include the modeling and the control of mobile robots, submitted to uncertain and dynamical effects. This covers the motion control as well as safety aspects (rollover, instability). These works are applied in various areas: agriculture and environment, driver assistance or military aspects.

Currently in Irstea since 2006, Roland Lenain has been in charge of several National Projects in the topic of mobile robot and vehicle control in hazardous context (FAST, ActiSurTT, ...). He achieved a Post Doctoral Position in Lund University (dpt Automatic Control). He received is Ph.D. degree in Robotics from the Blaise Pascal University in 2005. He was graduated from IFMA (French Institute for Advanced Mechanics) in 2002.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



Formation control of off-road fleet of UGVs: issues, advances and applications

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICBA 2014

Roland Lenain

ICRA14 Workshop on Modelling, Estimation, Perception and Control of All Terrain Mobile Robots
Hong-Kong, 2014, June 1st

2

Introduction

Mobile robots – from a single autonomous robot to a fleet control

Multi robots control

- Formalism and objectives
 - General principles
 - Modeling
 - Basic assumptions
- **Lateral dynamic control**
 - Extension of a single robot approach
 - Desired set point computation
 - Predictive approach
- Longitudinal dynamic control
 - Reactive approach
 - Relation with other robots
 - Predictive approach and motion anticipation

Extension to target tracking

- Common framework
- Algorithm generalization



Introduction

Autonomous vehicle nowadays

- From dream to actual applications



1986 movie

- Several solved problem

- navigation
- Perception
- Control aspects



2012 google car

Some open issues remaining

- Human machine interaction
- Sensor and material costs
- Related to All-terrain context

- Control and positioning accuracy
- Harsh conditions (terrain geometry, grip conditions)
- Extending area to be covered



2006 Stanley Darpa
 Challenge (stanford
 Univ)

Extend speed

Extend number of robots



Introduction

Extend speed of off-road robots

- Face high dynamical effects
 - Inertial effects
 - Low grip conditions
- Important Safety issues
 - Risk of controllability loss
 - Risk of rollover pending on terrain conditions
 - Obstacle avoidance becomes critical



Several robots under cooperation

- Permits to preserve limited speed
- High dynamics phenomena neglected
- Several point of views to preserve efficiency



Swarm
 robotics



Formation Control



Number of robots

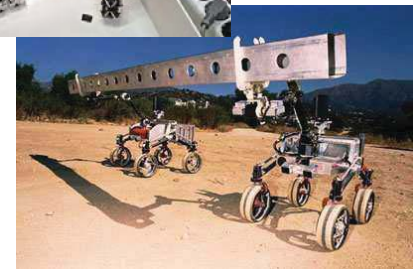
scale of robots



Formation control objectives

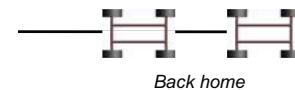
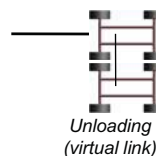
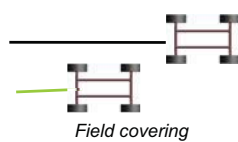
Several points of view

- Independent control
 - All of robots has its own area
 - Few interaction
 - Separate work
- Robots work together
 - May be associated to achieve a bigger task
 - May be controlled or supervised
 - May achieved complementary works



Coordinated control under some formation

- Preserve repeatability
- Ensure a high level of accuracy
- Preserve its integrity and safety
- Have to be adaptable (variable shape)



Formation control issues in off-road context

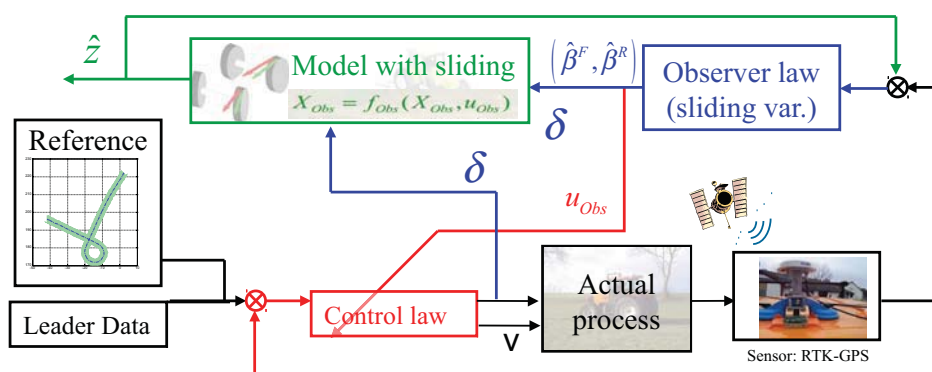
Harsh and variable conditions

- Environments and terrain properties
- Robot design and capabilities
- Perception and communication state
- Phenomena to be accounted?



Overview of the control approach

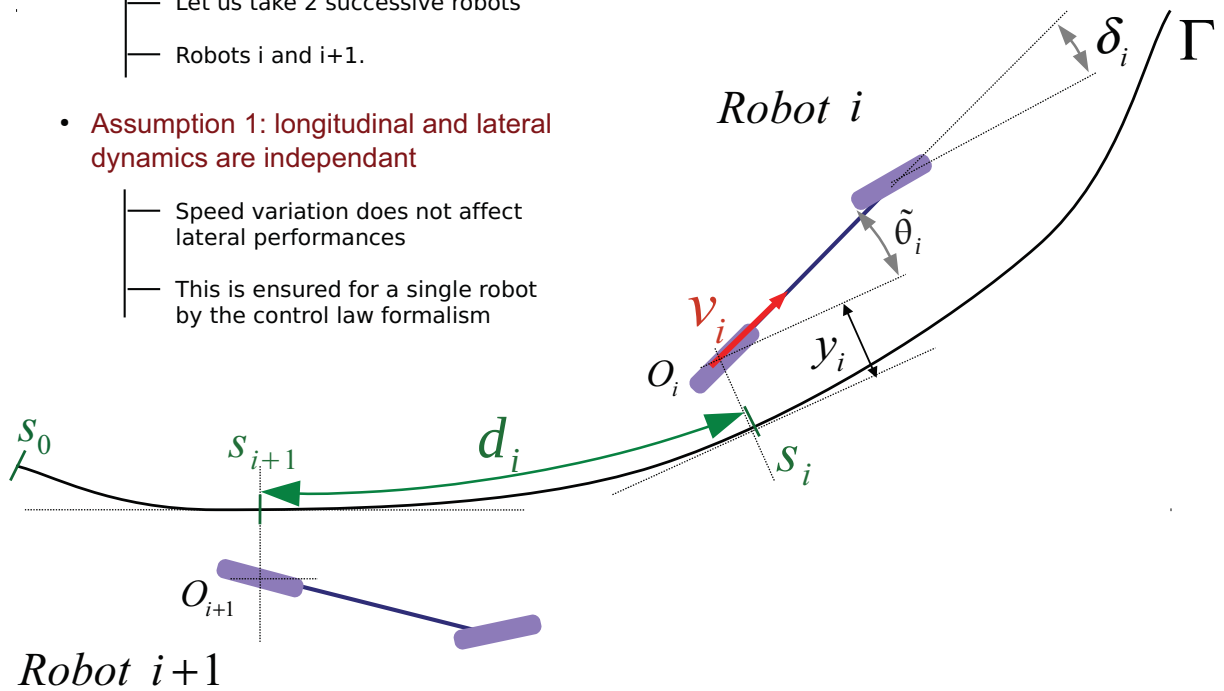
- Modeling of the fleet
- Adaptive layer
- Predictive layer



Modeling of robots formation

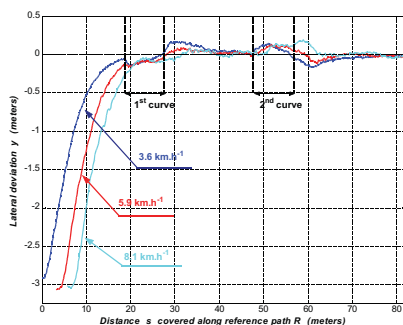
Formalism in the path tracking framework

- Considers $n > 1$ robots
 - Let us take 2 successive robots
 - Robots i and $i+1$.
- **Assumption 1: longitudinal and lateral dynamics are independent**
 - Speed variation does not affect lateral performances
 - This is ensured for a single robot by the control law formalism

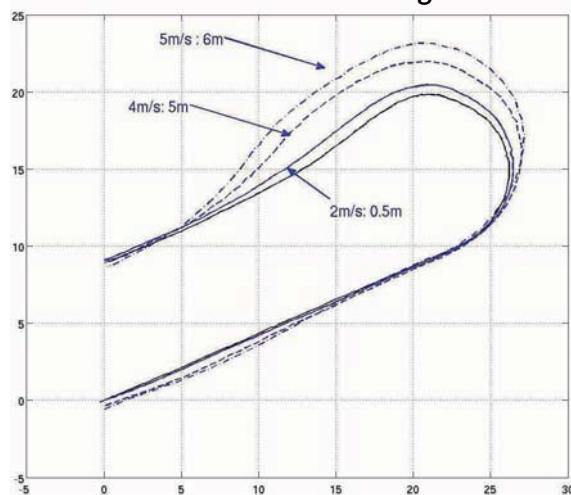


Classical path tracking control

Satisfactory when assumptions are valid



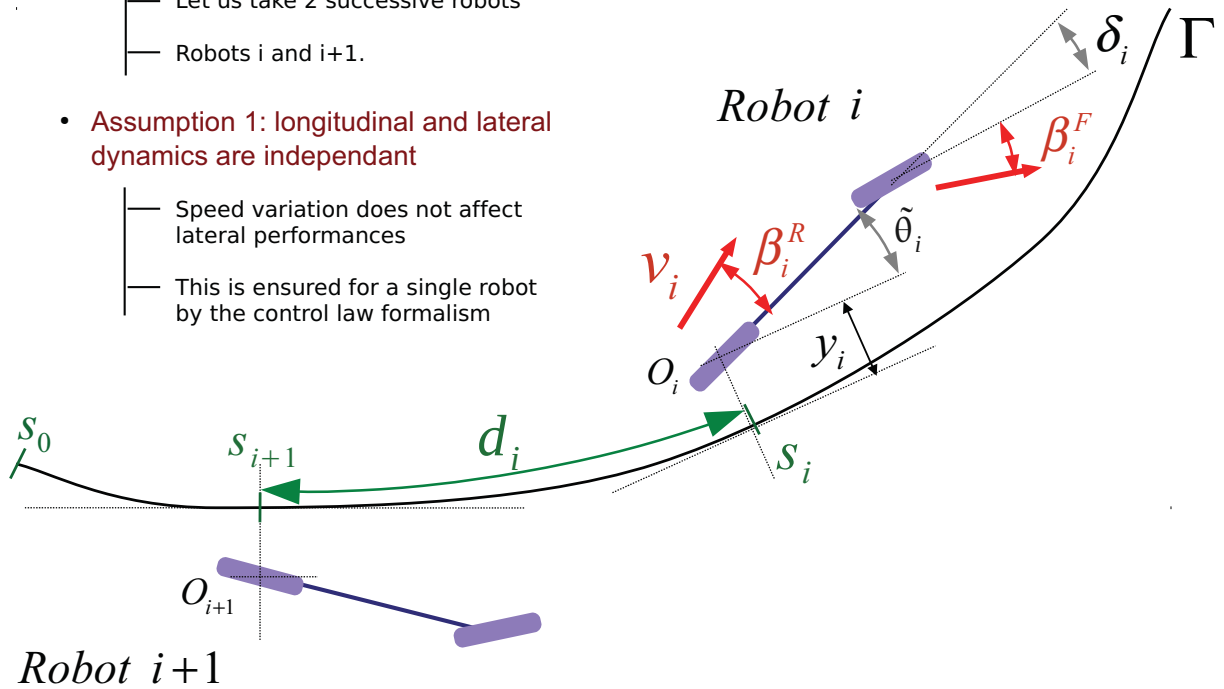
Unsuitable when running on natural ground (depends on speed!!)



Modeling of robots formation

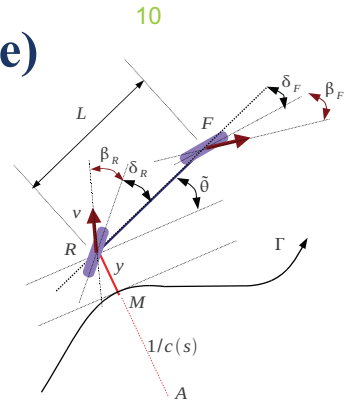
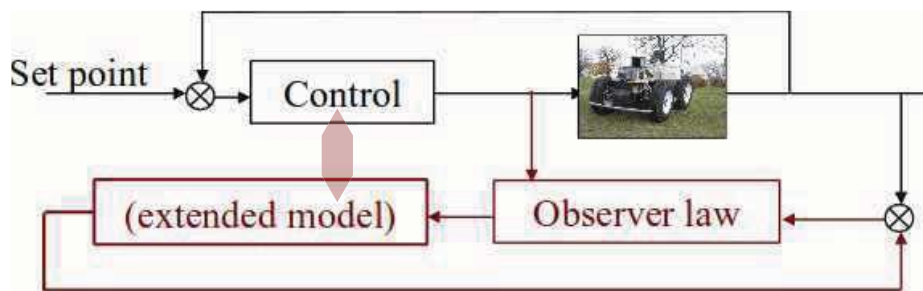
Extended model

- Considers $n > 1$ robots
 - Let us take 2 successive robots
 - Robots i and $i+1$.
- **Assumption 1: longitudinal and lateral dynamics are independant**
 - Speed variation does not affect lateral performances
 - This is ensured for a single robot by the control law formalism



Sideslip angle observation (virtual measure)

General scheme of observation



- A second loop is designed
 - Convergence of model output to measure
 - Using sideslip angles computation
- Assumptions for reconstruction
 - Convergence of model output to measure
 - Using sideslip angles computation

Observer design

Measured state $X = \begin{bmatrix} y \\ \theta \end{bmatrix}$ derivative $\dot{X} = \begin{bmatrix} \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \sin(\tilde{\theta} + \delta_R + \beta_R) \\ v \cos(\delta_R + \beta_R) \frac{\tan \delta_F + \beta_F - \tan \delta_R + \beta_R}{L} - \frac{c(s) v \cos(\tilde{\theta} + \delta_R + \beta_R)}{1 - c(s) y} \end{bmatrix}$



Introduction – Lecture overview

Callback trajectory tracking control

Multi robots control

- Formalism and objectives
 - General principles
 - Modeling
 - Basic assumptions
- Lateral dynamic control
 - Extension of a single robot approach
 - Desired set point computation
 - Predictive approach
- Longitudinal dynamic control
 - Reactive approach
 - Relation with other robots
 - Predictive approach and motion anticipation

Extension to target tracking

- Common framework
- Algorithm generalization



Modeling of robots formation

Motion model stay unchanged

- For the i^{th} robot
 - Controls : velocity, steering angles(s)
 - State: Cirv. Absc. lateral + angular deviations
- Kinematic model is let unchanged

$$\dot{X} = \begin{bmatrix} \dot{s} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \frac{\cos(\tilde{\theta} + \delta_R + \beta_R)}{1 - c(s)y} \\ v \sin(\tilde{\theta} + \delta_R + \beta_R) \\ v \cos(\delta_R + \beta_R) \frac{\tan \delta_F + \beta_F - \tan \delta_R + \beta_R}{L} - \frac{c(s)v \cos(\tilde{\theta} + \delta_R + \beta_R)}{1 - c(s)y} \end{bmatrix}$$

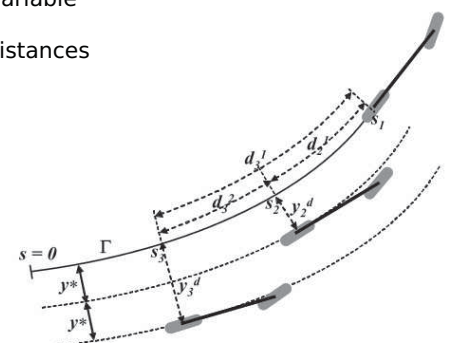
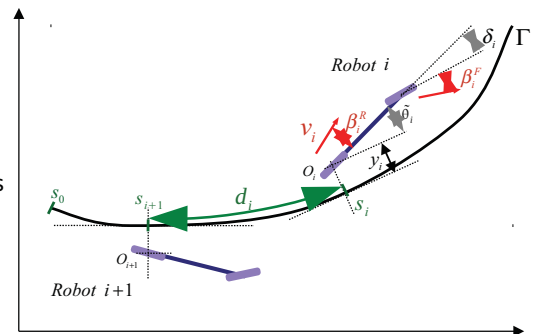
- In counterpart the set point changes to describe relative positions
 - Lateral deviation may defined different from zero and variable
 - The velocity may be computed to ensure longitudinal distances

- Control objectives

- Lateral :

$$y_{i+1} \rightarrow y^d(y_i)$$
- Longitudinal : servo the curvilinear distance

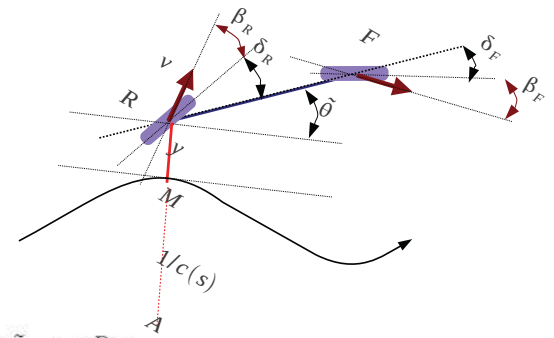
$$s_i - s_{i+1} \rightarrow d_i$$



Lateral dynamic control

The same principle as for a single robot may be used

- Control objective $y_i \rightarrow y_i^d$
 - Whatever the speed
 - Whatever the potential rear steering angle
 - Knowing the grip conditions
- Using exact linearization



Variable transformation

$$\begin{aligned} [s_i, y_i, \tilde{\theta}_i] &\rightarrow [a_{1i}, a_{2i}, a_{3i}] = \\ &\quad [s_i, y_i, (1 - c(s_i) y_i) \tan(\tilde{\theta}_i + \beta_i^R)] \\ [v_i, \delta_i] &\rightarrow [m_{1i}, m_{2i}] = \left[\frac{v_i \cos(\tilde{\theta}_i + \beta_i^R)}{1 - c(s_i) y_i}, \frac{da_{3i}}{dt} \right] \end{aligned}$$

Chained system form \rightarrow derivative w.r.t curvilinear abscissa

$$\begin{cases} \dot{a}_{1i} = \frac{da_{1i}}{dt} = m_{1i} \\ \dot{a}_{2i} = \frac{da_{2i}}{dt} = a_{3i} m_{1i} \\ \dot{a}_{3i} = \frac{da_{3i}}{dt} = m_{2i} \end{cases} \quad \begin{cases} a'_{2i} = \frac{da_{2i}}{ds_i} = a_{3i} \\ a'_{3i} = \frac{da_{3i}}{ds_i} = m_{3i} = \frac{m_{2i}}{m_{1i}} \end{cases}$$

Virtual control law

$$m_{3i} = -K_d(a_{3i} - y_i^d) - K_p(a_{2i} - y_i^d) + y_i^{d'} \quad (K_d, K_p > 0)$$

$$\epsilon_i^{''y} + K_d \epsilon_i^{'y} + K_p \epsilon_i^y = 0 \quad \text{with} \quad \epsilon_i^y = a_{2i} - y_i^d$$

Lateral dynamic control

Computation for a given robot i

- Control design

Expression

$$m_{3i} = -K_d(a_{3i} - y_i^d) - K_p(a_{2i} - y_i^d) + y_i^{d'} \quad (K_d, K_p > 0)$$

Reverse transformation

$$\begin{aligned} [s_i, y_i, \tilde{\theta}_i] &\rightarrow [a_{1i}, a_{2i}, a_{3i}] = \\ &\quad [s_i, y_i, (1 - c(s_i) y_i) \tan(\tilde{\theta}_i + \beta_i^R)] \\ [v_i, \delta_i] &\rightarrow [m_{1i}, m_{2i}] = \left[\frac{v_i \cos(\tilde{\theta}_i + \beta_i^R)}{1 - c(s_i) y_i}, \frac{da_{3i}}{dt} \right] \end{aligned}$$

Steering angle control law

$$\delta_i = \arctan \left\{ \tan(\beta_i^R) + \frac{l}{\cos(\beta_i^R)} \left(\frac{c(s_i) \cos \tilde{\theta}_{2i}}{\alpha_i} + \dots \right) \right. \\ \left. \dots \frac{A_i \cos^3 \tilde{\theta}_{2i}}{\psi \alpha_i^2} \right\} + \beta_i^F$$

$$\begin{cases} \tilde{\theta}_{2i} = \tilde{\theta} + \beta_i^R \\ \alpha_i = 1 - c(s_i) y_i \\ A_i = -K_p \epsilon_i^y - K_d \alpha_i \eta + c(s_i) \alpha_i \eta \tan \tilde{\theta}_{2i} \\ \eta = \left(\tan \tilde{\theta}_{2i} - \frac{\dot{y}_i^d}{v_i \cos \tilde{\theta}_{2i}} \right) \\ \psi = 1 + \tan^2(\tilde{\theta}_{2i}) - \frac{\dot{y}_i^d \tan \tilde{\theta}_{2i}}{v_i \cos \tilde{\theta}_{2i}} \end{cases}$$

- Based on the same assumption and methodology than for a single robot with set points

- Settling distance is theoretically independent from velocity
- Convergence to a desired lateral distance
- Actuator settling time and delays are neglected

How the relationship with others is ensured ?????

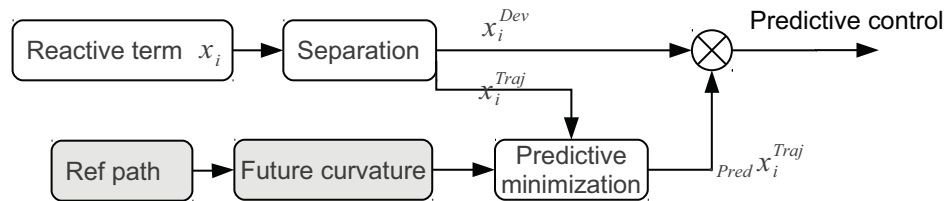
Lateral dynamic control

Control strategy

- Path tracking based approach using a common reference
- A non null desired distance is defined
- Lateral control law is slightly modified, but methodology is preserved

Extensions

➤ Predictive control may be used if future trajectory is available (see next part)



➤ Introduction of a composite error pending on other robots deviation

- Let us define several errors with other robots
- Define a global error as a linear combination of elementary errors

➤ Key issue: defined a rule for varying coefficient



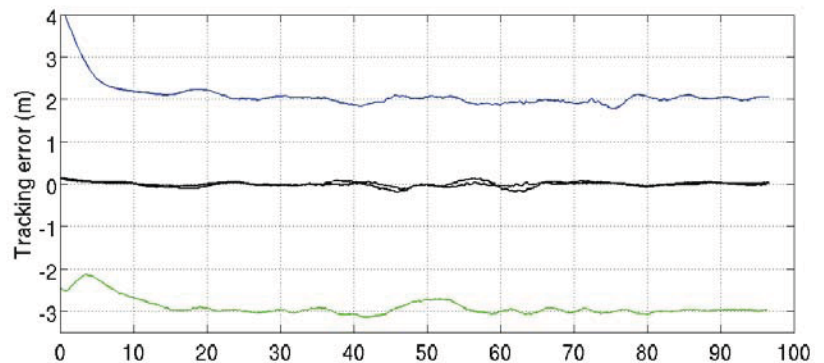
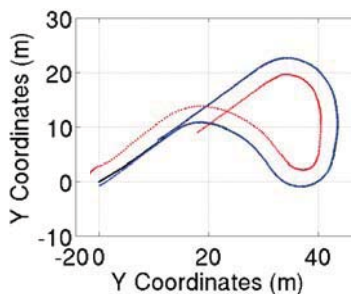
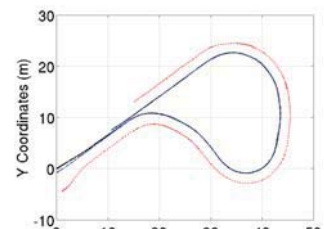
Lateral dynamic control

Actual results

- B/ Prediction addon interest

— speed : 2m/s — Communication WIFI

1. Constant desired deviation at 2m, and -3m



Introduction – Lecture overview

Callback trajectory tracking control

Multi robots control

- Formalism and objectives
 - General principles
 - Modeling
 - Basic assumptions
- Lateral dynamic control
 - Extension of a single robot approach
 - Desired set point computation
 - Predictive approach
- Longitudinal dynamic control
 - Reactive approach
 - Relation with other robots
 - Predictive approach and motion anticipation

Extension to target tracking

- Common framework
- Algorithm generalization



Longitudinal dynamic control

Velocity control

- Velocity servoing objective : $s_i - s_{i+1} \rightarrow d_i$
 - Ensure a curvilinear distance
 - Whatever the lateral position
 - Whatever the grip condition
- With respect to the previous robot

- Error definition

$$e_{i+1} = s_i - s_{i+1} - d_i$$
- Desired behavior for the convergence of the defined error to zero

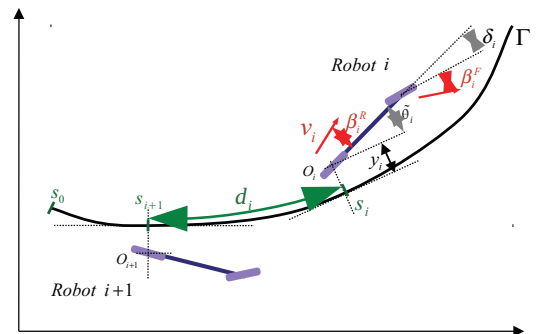
$$\dot{e}_{i+1} = K_{i+1}^l e_{i+1}$$

- Time derivative (or derivative w.r. To curvilinear abscissa)

$$\dot{e}_{i+1} = v_i \frac{\cos(\tilde{\theta}_{2i})}{1 - c(s_i) y_i} - v_{i+1} \frac{\cos(\tilde{\theta}_{2(i+1)})}{1 - c(s_{i+1}) y_{i+1}} - \dot{d}_i$$

- Control law expression for velocity may be deduced (w.r to robot i) :

$$v_{i+1} = \frac{1 - c(s_{i+1}) y_{i+1}}{\cos(\tilde{\theta}_{2(i+1)})} \left[v_i \frac{\cos(\tilde{\theta}_{2i})}{1 - c(s_i) y_i} - K_{i+1}^l e_{i+1} \right]$$



Longitudinal dynamic control

Velocity control

- Other distance to be defined
 - If control is referred only with respect to previous robot
 - Errors may be propagated over the formation
 - If previous robots stop, the follower will stop too
 - If the last one stops ???

Control with respect to robot i

- Same methodology, defining the distance with first robot considered as a leader
 - Only index and desired distance change
 - error

$$e_{i+1}^i = s_i - s_{i+1} - d_i$$

$$e_{i+1}^1 = s_1 - s_{i+1} - \sum d_m$$

$$v_{i+1} = \frac{1 - c(s_{i+1})y_{i+1}}{\cos(\tilde{\theta}_{2(i+1)})} \left[v_A \frac{\cos(\tilde{\theta}_{21})}{1 - c(s_1)y_1} - K'_{i+1} e_{i+1}^1 \right]$$
 - Control only with respect to the leader
 - No error propagation (a unique reference)
 - If a previous robot (except the leader) stops
 - If the last robot stops?



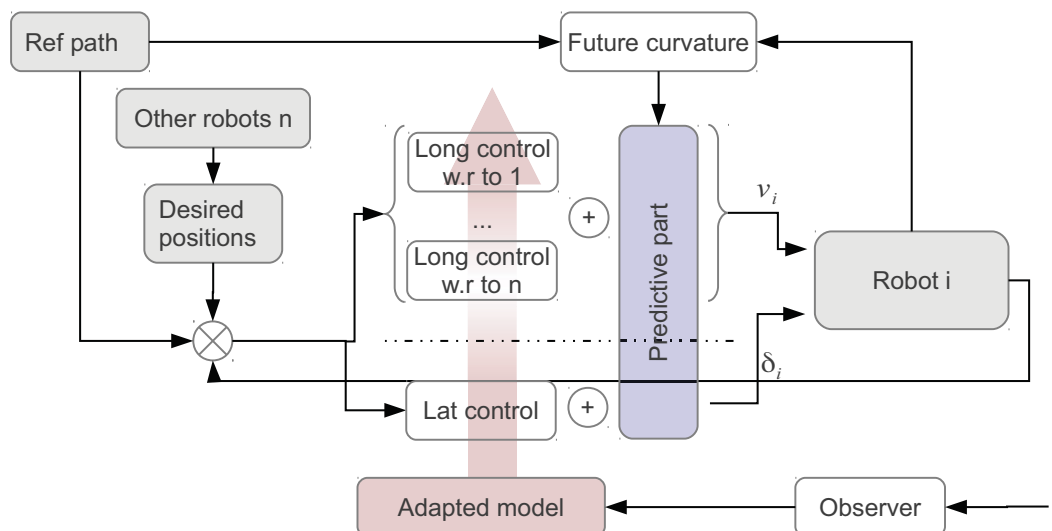
Longitudinal dynamic control

Hybrid distance regulation

- The velocity control may regulate several distance between robots

- Control of i^{th} robot with respect to $n-1$ others

$$v_i^k = \frac{1 - c(s_i)y_i}{\cos(\tilde{\theta}_i + \beta_i^R)} \left(\frac{\cos(\tilde{\theta}_k + \beta_k^R)}{1 - c(s_n)y_n} + k_l^k e_i^k \right)$$
- Collection of $n-1$ control law (or $n-1$ relative distance) $e_i^k = s_i - s_k - \sum_{j=k}^i d_j$
- Selection/weighting process
- Different functions: Safety/formation control/insertion...

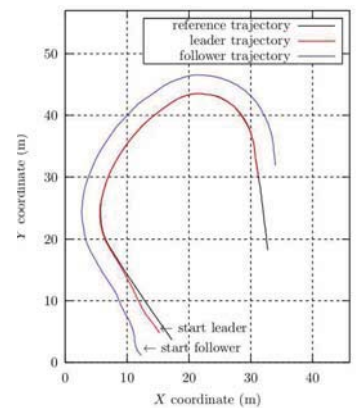
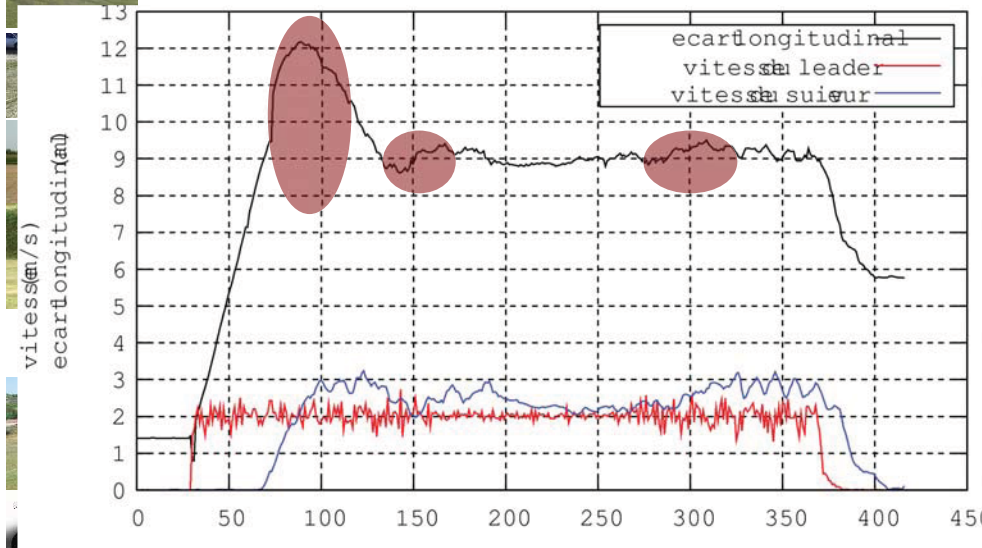


Longitudinal dynamic control

Results without prediction

- Parameters

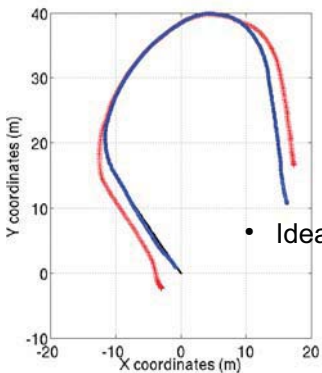
- Leader : 2m/s
- Desired dist.i: constant = 9m
- Lateral dev. : -2m



Longitudinal dynamic control

Control law drawback and extension

- Purely reactive strategy (speed variations may generate overshoot)
 - Does not account for curvature variations
 - Does not take into account actuator settling time
 - Robots of the fleet may be heterogeneous
- Velocity may change even in steady state period



- Reference path curvature modifications
- Set points may change
- Leader may have velocity change

- Ideal actuators effects

- Overshoots
- Oscillations
- Instability

➡ Gain may be tuned to improve stability (accuracy may be depreciated)

➡ Predictive and feedforward control may be developed



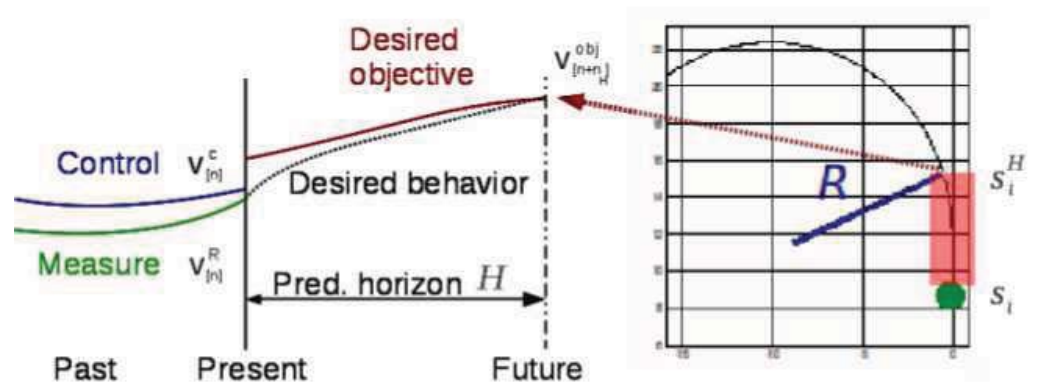
Predictive longitudinal dynamic control

Control law extension

$$v_i^k = \frac{1-c(s_i)y_i}{\cos(\tilde{\theta}_i+\beta_i^R)} \left(\frac{\cos(\tilde{\theta}_k+\beta_k^R)}{1-c(s_n)y_n} + k_l^k e_i^k \right)$$

- Predictive control 1. Curvature servoing

— Trajectory in front of the I robot may be known



— Control law separation

— Expected future position is derived

— A desired behavior is chosen

— Control structure is chosen

— Minimization in the future give the control to be applied

— The term pending on curvature is substituted with predictive term

Predictive longitudinal dynamic control

Control law extension

- Predictive control 2 : Feedforward and prediction

— Step 1 : Feedforward - Future distance may be computed

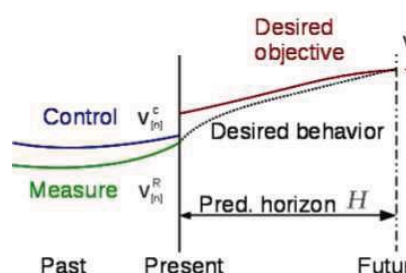
$$e_{i+1}^i = s_i - s_{i+1} - d_i$$

$$e_{i+1}^i(t+H) = s_i(t+H) - s_{i+1}(t+H) - d_i$$

— Step 2: Future desired velocity is derived

$$v_i^k = \frac{1-c(s_i)y_i}{\cos(\tilde{\theta}_i+\beta_i^R)} \left(\frac{\cos(\tilde{\theta}_k+\beta_k^R)}{1-c(s_n)y_n} + k_l^k e_i^k \right)$$

— Step 3: Model predictive control is applied using future set point for velocity



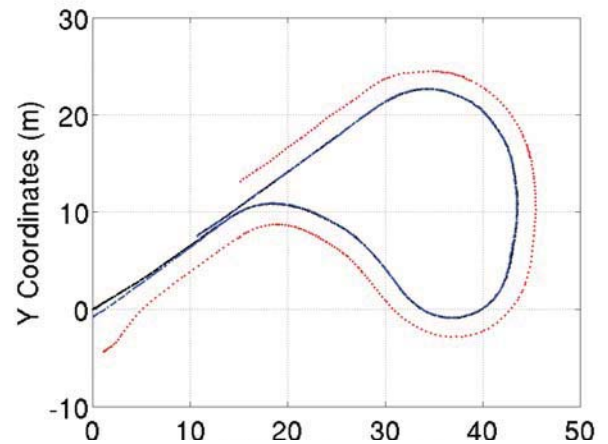
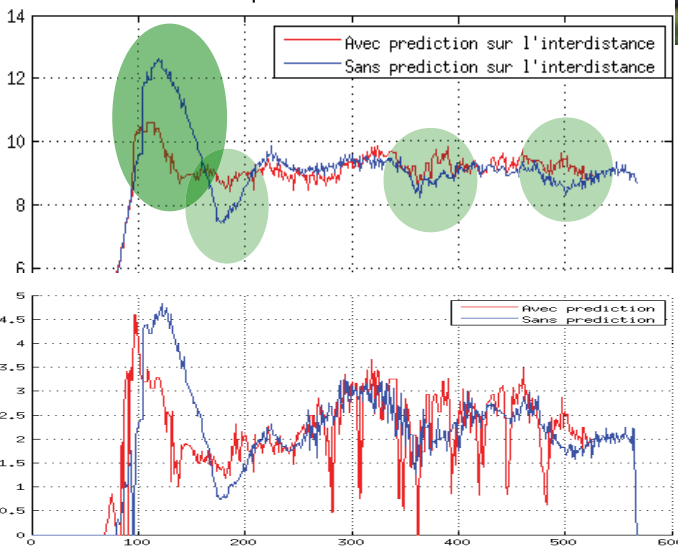
➡ Error and curvature are anticipated

Predictive longitudinal dynamic control

Results with 2 prediction levels

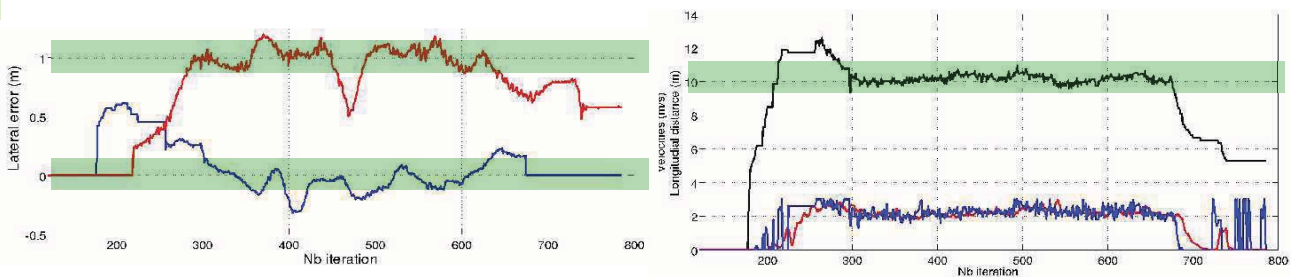
- Parameters

- Leader : 2m/s
- Desired distance: constant = 9m
- Lateral set point: 3m

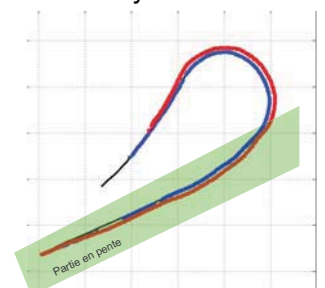


Motion control – 2 – Formation control results

Fixed configuration, robustness w.r to terrain geometry

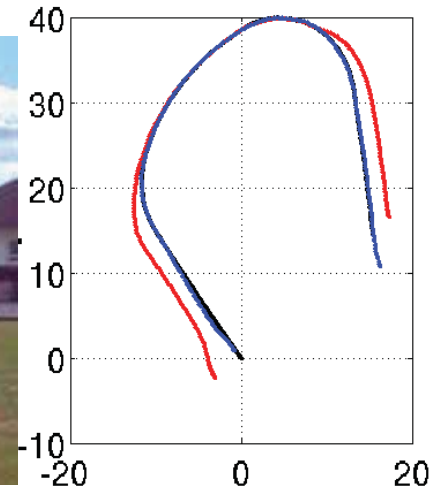
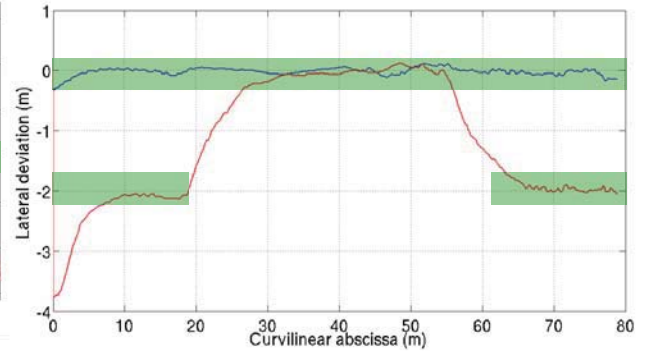
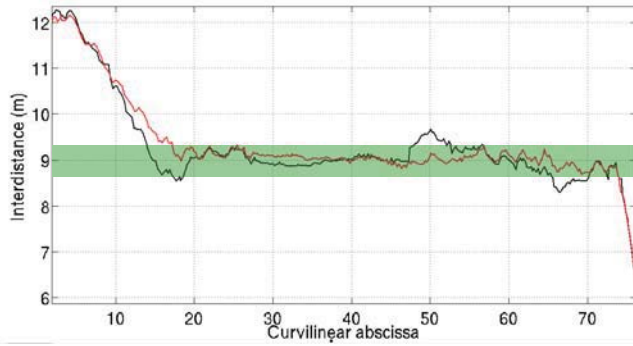


- Leader at 2m/s
- Desried Long: 9m
- Desired Lat: 1m
- Inclination of 15°
- Accuracy <15cm



Motion control – 2 – Formation control results

Variable configuration and predictive control interest



Introduction – Lecture overview

Callback trajectory tracking control

Multi robots control

- Formalism and objectives
 - General principles
 - Modeling
 - Basic assumptions
- Lateral dynamic control
 - Extension of a single robot approach
 - Desired set point computation
 - Predictive approach
- Longitudinal dynamic control
 - Reactive approach
 - Relation with other robots
 - Predictive approach and motion anticipation

Extension to target tracking

- Common framework
- Algorithm generalization



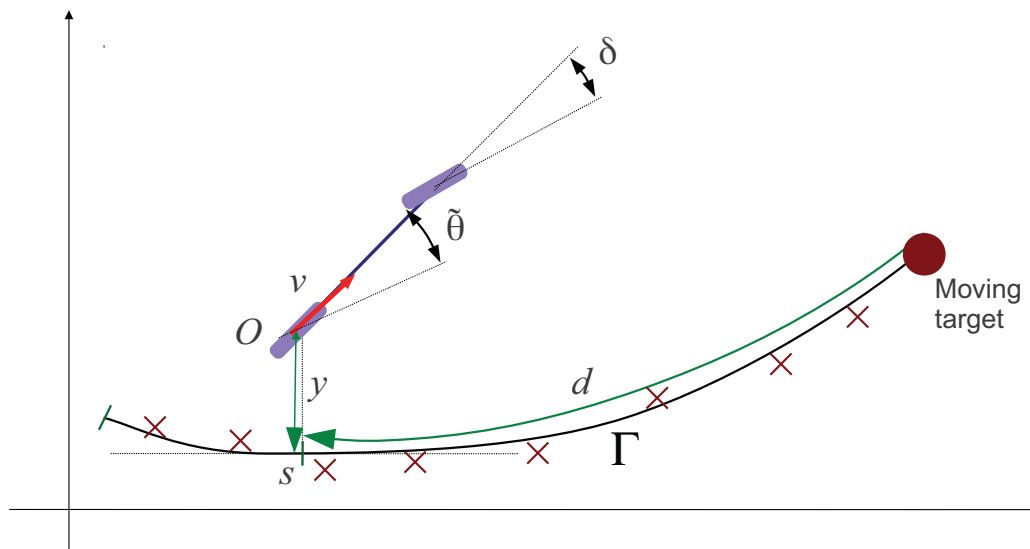
Formation control with a free leader

Target may be manually controlled or directly a moving object

- Trajectory is not necessarily achievable
- Variations may be impossible to follow

The same control strategy may be used

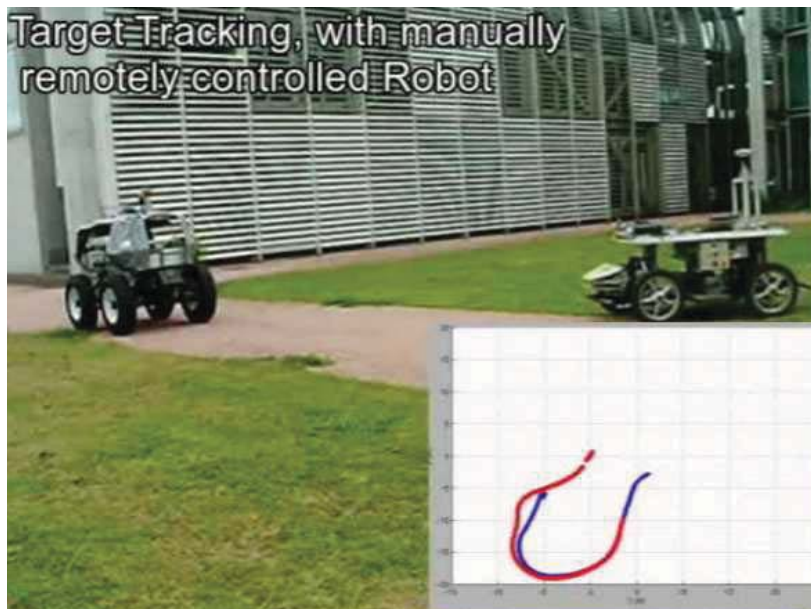
- Similar model may be used
- Reference trajectory is viewed as successive target positions



Formation control with a free leader

Results with an achievable trajectory target

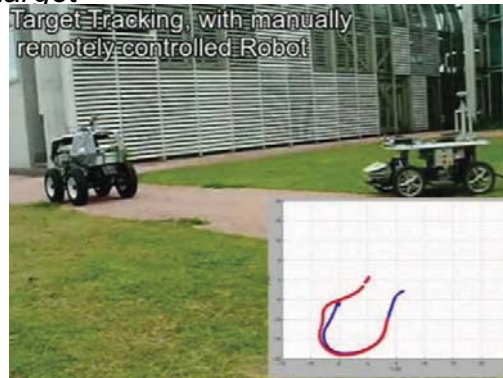
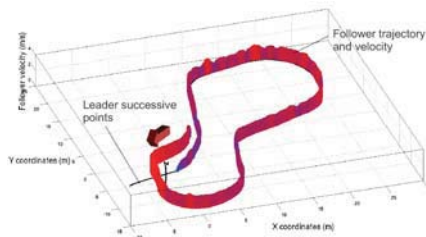
- target = remotely controlled robot
- Using RTK-GPS sensors
- All-terrain conditions



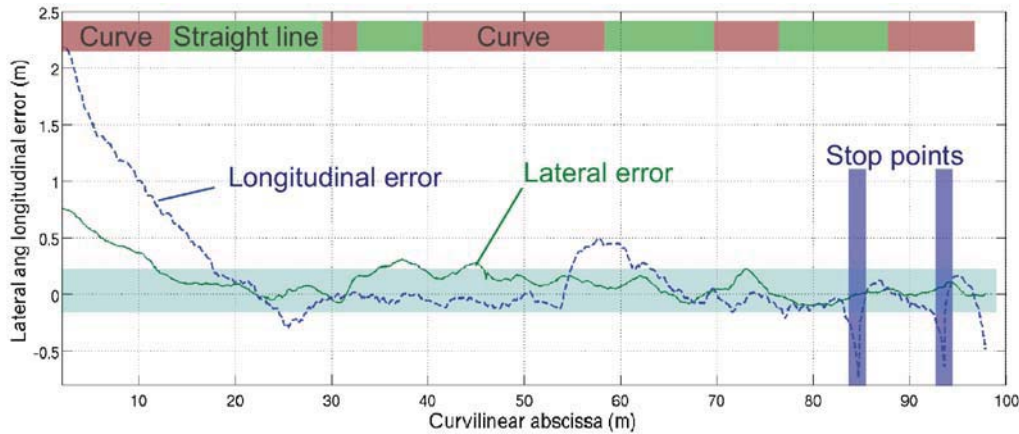
Formation control with a free leader

Results with an achievable trajectory target

- Trajectory followed



- Tracking errors

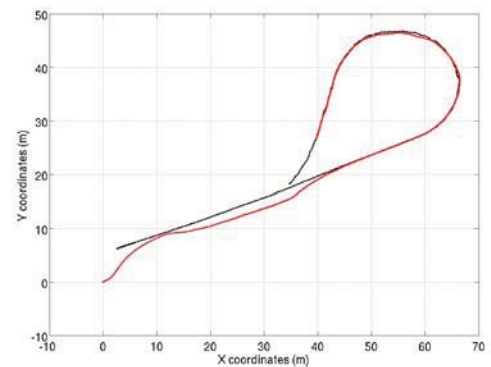
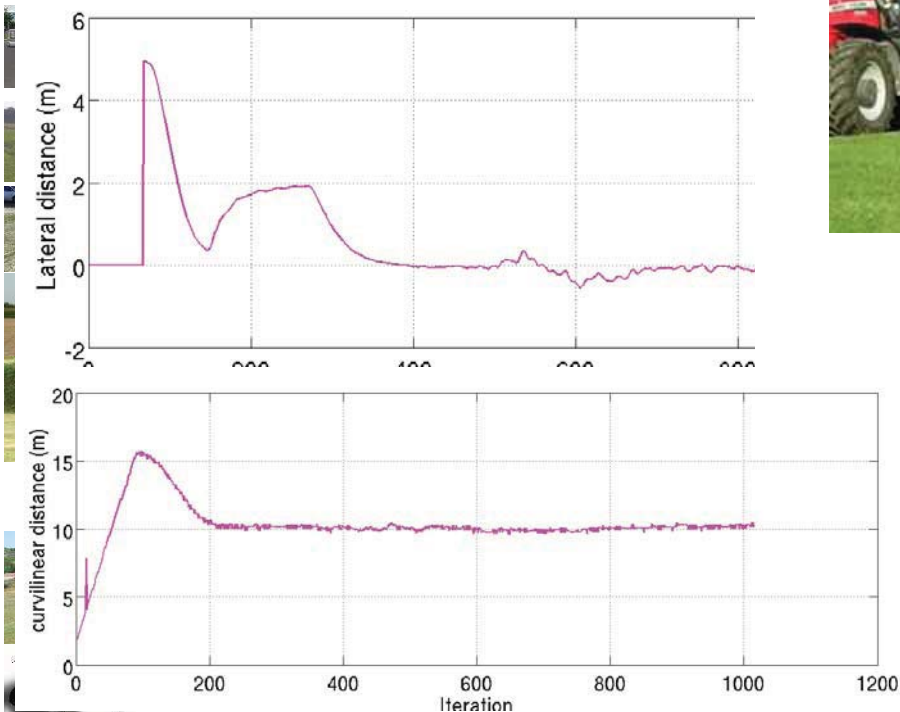


Formation control with a free leader

Actual results

- Without predefined reference Traj
- speed : 2m/s — WIFI communication

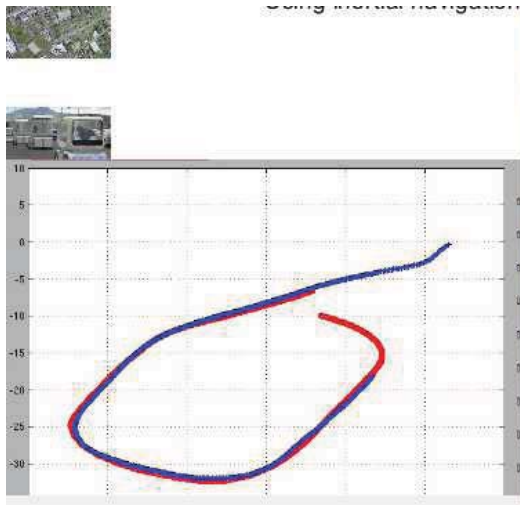
1. Variable deviation 0 → 2m without complete trajectory



Formation control with a human target

Pedestrian following with a desired distance

- Using inertial navigation



- Using another type of robot



Car like (v, δ)



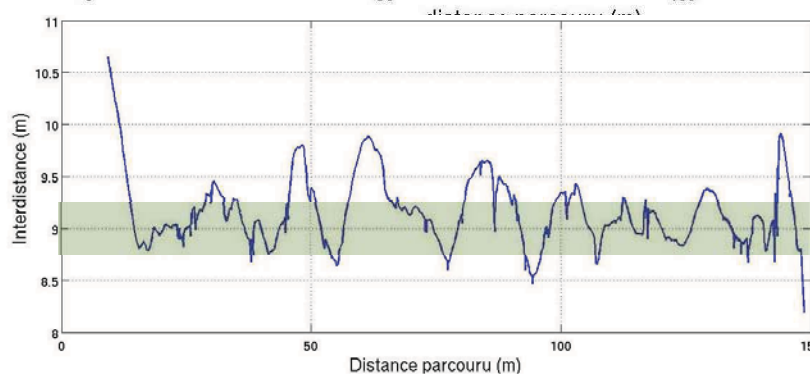
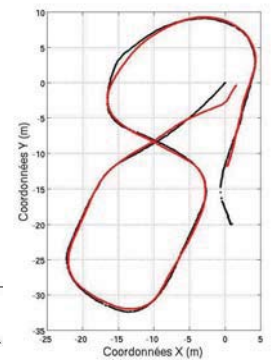
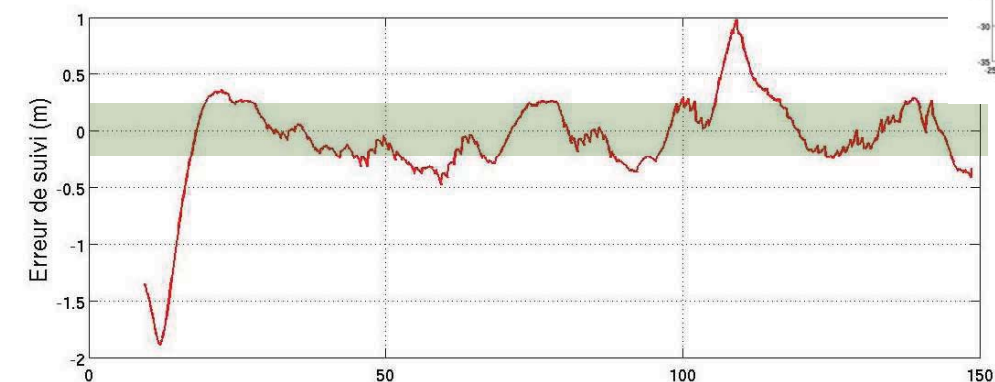
Skid steering

(v, θ)

Formation control with a human target

Pedestrian following with a desired distance

- Using inertial navigation



Formation control (remaining issues)

Regarding robots an terrain interaction

- Grip conditions are addressed one by one
- Skidding estimation to be shown in next lecture

Regarding fleet management

- Obstacle avoidance and management
 - Perception problem w.r. to Numeric Terrain Model
 - Avoidance may be addressed using non null lateral set point
- Robot addition and removal
 - On-line modifications of control laws number
 - On-line modification of weighting functions

Regarding robots safety

- Safety distance must be ensured
 - Not necessarily compatible with desired set point (pending on speed)
 - Ability to compute safety distance?
 - Control must account for localization accuracy and availability
- ➔ Supervision process must manage desired distances and control laws



Robot cooperation in the framework of formation

Viewed as an extension of path tracking

- Longitudinal and lateral dynamics have decoupled
 - Using exact linearization of steering angle control (see course 1)
 - This may be true whatever the control strategy
- Lateral control
 - Steering angle or yaw rate, stay unchanged
 - But the set point is modified (relative lateral position)
- Longitudinal control (velocity)
 - Using a defined curvilinear distance
 - Elementary control law with respect to one robot
 - Extension and weighting of n-1 elementary control laws
- Predictive principles may be added to improve robustness with respect to settling times
 - 1. Feedforward control in order to anticipate for errors
 - 2. The future set point may be derived
 - 3. Model predictive control is then applied

In off-road conditions, it uses extended kinematic model...





Session IV

Cooperative Robots, Software architecture

• **Title:** Integrating ABSYNTHÉ autonomous navigation system into ROS

Authors: A. Llamazares, E. Molinos, M. Ocana and F. Herranz

• **Title:** The Artemis Rover as an Example for Model Based Engineering in Space Robotics

Author: J. Schwendner, T. M. Roehr, S. Haase, M. Wirkus, M. Manz, S. Arnold and J. Machowinski

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots



Integrating ABSYNTHÉ autonomous navigation system into ROS

Ángel Llamazares, Eduardo Molinos, Manuel Ocaña
and Fernando Herranz
Department of Electronics. University of Alcalá, Madrid (Spain)
Email: allamazares, emolinos, mocana, fherranz@depeca.uah.es

Abstract—ABSYNTHÉ, which stands for **Abstraction, Synthesis, and Integration of Information for Human-Robot Teams**, is an interdisciplinary project which aims to develop basic concepts and structures for the abstraction of partial views of the environment and the actions and intentions of teams, as well as the integration of this information into situation assessments. One of our key objectives with this project is to develop the autonomous navigation system to be used in different platforms (big all terrain outdoor and small indoor robots) that can contain a variety of heterogeneous sensors. It represents a challenging topic because we have to develop a robust and safety navigation system that can be used by these different robots. The goal of this paper is to show the integration of the navigation system into Robot Operating System (ROS) platform. Some experimental results and conclusions will be presented.

I. INTRODUCTION

Robots teams are being developed to support a variety of human activities, including planning and decision making. In recent years, these teams have started to include humans as active components that sense, act, and collaborate with their automated counterparts. These human-robot teams are characterized by their ability to acquire vast amounts of data, generally heterogeneous in nature, reflecting the variety of available member platforms and sensing devices. These data must be properly processed, analysed, interpreted, and fused to ensure the right functioning of human-robot teams in terms of coordination, negotiation, distribution, and cooperation.

The competency of humans to collaborate on the solution of complex problems, while exchanging information that facilitates the understanding of system behaviour, may be explained by the use of cognitive tools that produce summarizations and descriptions of complex objects, events, and relations in terms that are easy to comprehend by other humans. In addition, achieving an effective collaboration of human and robots in mixed human-robot environments, is especially important to ensure a robust navigation for complex tasks, such as rescue and guiding, in terms of ensuring human integrity.

The autonomous navigation systems have been studied deeply in the literature [1] [2]. To achieve an autonomous navigation, the system should take into account the following stages: perception, localization, cognitive processes and motors modulation. While the first one uses the on-board sensors to provide information about the surroundings of the robot, the localization stage is usually based on this sensory information and a priori map to determine the position. Once it has been determined the robot position, the cognitive process stage determines what things are needed to do to ensure that the robot gets the target in a safety way. Finally, cognitive

processes will provide the output for the motors modulation stage.

When the robot is navigating in outdoors, usually Global Positioning System (GPS) or its enhanced version Differential GPS (DGPS) [3] are used in the localization stage because they provide enough accuracy (a few centimetres in DGPS case). On the contrary, when GPS receiver is in urban environments with high buildings or trees, the signal can suffer multipath fading or even Line-Of-Sight (LOS) blockage. In addition, it is important to remark that GPS signal is not strong enough to penetrate inside tunnels or buildings, then this problem discards this technique in indoor environments.

To solve this kind of problem, robot navigation systems use a combination of a previous map with perception information that comes from a sensor fusion [4] [5] to guide the robot. Maps are usually obtained in a semi-autonomous process known as mapping [6] [7].

Localization and mapping are two processes with similar features and a common problem. It is not possible to build a map if the localization process does not work well, and it is impossible to locate a device with high precision without an accurate map. Some many times, SLAM (Simultaneous Localization And Mapping) techniques are used to fix this problem doing both processes simultaneously and using a variety of sensors, such as GPS and vision [4], vision and laser [8] even using RF range only sensors [9] [10].

While the map, generated by mapping or SLAM algorithms, can be used by the global planner to reach the target, if we want to get a robust and safety navigation it is needed to include an obstacle avoidance system. It ensures that the robot always navigates avoiding obstacles and keeping robot-human integrity [11] [12]. Obstacle avoidance systems can be divided into global or local, depending of the available information about the environment. While the first ones assume a complete model of the environment such as potential field methods [13], local methods can be considered faster and can be programmed like reactive tasks. These reactive methods control the robot when an obstacle is detected to avoid the collision. They use the nearest portion of the environment and update the world model according to the current sensor observation.

In this work, we review some several available navigation, mapping and localization packages in ROS. We describe our own algorithms and how we have integrated them into this platform. We test the combination of several algorithms to achieve a robust and safety autonomous navigation using heterogeneous robots and sensors. Real and simulated experiments are developed to test the performance of these systems.

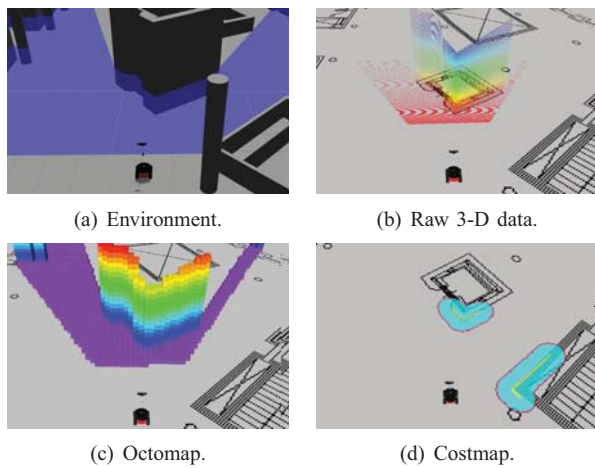


Fig. 1. From perception to mapping

In addition, we test the effect of dynamic obstacles on our navigation system.

The rest of the paper is organized as follows: section 2 shows a selection of related works; proposed method is shown in section 3; section 4 describes the test-bed and some experimental results; and finally, section 5 enumerates conclusions and future works.

II. RELATED WORKS

This section provides a description of available packages and some previous algorithms developed by the authors that have been integrated in ROS platform. We describe a briefly introduction about several algorithms related to the next stages: mapping, localization and navigation.

A. Mapping stage

The mapping is used to obtain a model of the real world based on the data provided by the sensors. These maps can be global or local depending of the knowledge about the actual robot position (absolute or relative). The following methods have been evaluated in this paper:

- Octomap [14] builds a probabilistic 3D representation of the world based on Octree structures using a Point Cloud data. This Point Cloud can be directly obtained from 3D range sensors (3D laser ranger or Kinect depth camera) or using 2D range sensors (laser or sonar) and the appropriate information about the pose of the robot. Octomap is available in ROS under BSD License.
- Costmap [15] obtains a probabilistic 2D representation based on occupancy grids. It needs 2D range data or 3D Point Cloud projected to the floor. It is also able to enlarge any occupied cell to a specified radius, which is recommended for safety navigation purposes. It is available in ROS under BSD License in the navigation package.
- DMAP (Dynamic Occupancy Mapping) [16] is a dynamic occupancy grid where it is stored the probability of occupation of each cell and an estimation of object's velocity. While the occupancy is obtained by a Bayesian

Occupancy Filter (BOF), the velocity estimation is based on a Kalman Filter tracking and a detection of the obstacles movement using the pyramidal implementation of Lukas Kanade optical Flow [17]. The algorithm is able to obtain not only a more robust position of the obstacle but also its velocity. This algorithm was developed by the authors and has been integrated into ROS.

Figure 1 shows an example of mapping using Octomap and Costmap. The environment 1(a) is perceived by a depth camera and in this way the 3D range raw data 1(b) is obtained. Using this Point Cloud, Octomap obtains the map shown in 1(c) and projecting this Point Cloud to the ground Costmap obtains the map shown in 1(d).

B. Localization stage

Global localization is the process that provides the position of a robot in a map, while local localization is relative to a starting position and increments the position of the robot by mean of integrating the odometry information. This is obtained by the encoders of the robot or by fusing information from encoders with Inertial Measurement Unit (IMU). A package to fuse this information using an extended Kalman filter is available in ROS (Robot_Pose_EKF) and can use odometry, IMU or even GPS (if available) to improve the localization of the robot.

If there is a priori good map of the environment it is possible to use an Adaptive Monte-Carlo Localization (AMCL) system [18] to obtain the global position of the robot in the map. This package is available in ROS under BSD License in the navigation package. On the contrary, if a map is not available, it is also possible to perform the Localization and Mapping stages simultaneously to obtain a map and the localization of the robot at the same time. SLAM processes can be found in the ROS package GMapping [7].

C. Navigation stage

The navigation stage must be divided into global and local stages. While the first one obtains the best path to get the target, the second one tries to avoid the possible obstacles of the route.

Global navigation calculates the best path from one point to another that the robot should be able to follow. This stage can be tackled by the Dijkstra algorithm [19] to calculate the best path into a static map given. Dijkstra is a graph search algorithm that solves the single source shortest path problem.

The local navigation controls the robot velocity to follow the path created by the global navigation. In addition, it avoids the obstacles found in the environment that are not included in the static map. There are so many algorithms to tackle this stage, some examples of them:

- 1) Dynamic window approach [20] tries to deal with the dynamic of the robot in a real world. This algorithm builds a search space of possible velocities without collision and search the best velocity to follow. It is available in ROS under BSD License in the navigation package.

- 2) Elastic band approach [21] fuses control and navigation into the same stage. It builds a map of interconnected spheres between the robot and the goal that can adapt their volume to the environment and selects the best path to travel from one sphere to another. It is available in ROS under BSD License in the navigation package.
- 3) VFH+ (Vector Field Histogram Plus) [22] is a reactive only algorithm. It was developed to use with laser sensors and divide the environment into angular sectors and choose the best sector to navigate according to a cost function with different parameters such as the angular difference from the actual position or distance to the goal. Smooth Nearness Diagram [12] also divides the environment into angular sectors but in this case can have different sizes. This algorithm maximizes the security and usually navigates at the middle of the best sector selected by a cost function. Both algorithms were released under BSD License for the Player Stage Project and were integrated in ROS by the authors.
- 4) Curvature Velocity Method [23] works with velocity states instead of position states. This algorithm builds all the curvature arcs that the robot can follow and selects the best arc to follow based on a cost function that depends on parameters such as free distance to obstacle or angular difference to the orientation of the robot. Lane Curvature Method [24] and Beam Curvature Method [11] are two extensions of the previous method that add a previous stage where the environment is divided (into rectangular lanes or angular sectors) and select a middle goal based on another cost function that CVM should reach. These algorithms were first released under Carmen Project and they have been integrated in ROS by the authors.

One of the main problems of these methods is that most of them do not take into account the dynamic information of the environment, they only consider that all obstacles are static. This characteristic is especially important when the robot deals with a high uncertainty over the position, shape and velocity of the obstacles and it is still a challenge for real world applications [25] [26]. In addition, these approaches only take into account the two-dimensional information and do not take care of the height of them.

III. PROPOSED METHOD

In this section we show our proposal of autonomous navigation avoiding dynamic obstacles into the ABSYNTHÉ project framework. We focus on an improvement of the local mapping stage to tackle with dynamic obstacles using DOMap, as it has been presented before.

The system is divided into several stages which are connected with each other as Figure 2 shows. It is important to notice that the main goal of our system is to navigate through an environment that is composed of human-robot teams. Therefore, the robot has to reach the goal introduced by a human on a mobile device application and it has to do autonomously and safely, keeping the integrity of both

human and robots. Figure 2 shows the stages involved in the navigation and the data flow between them.

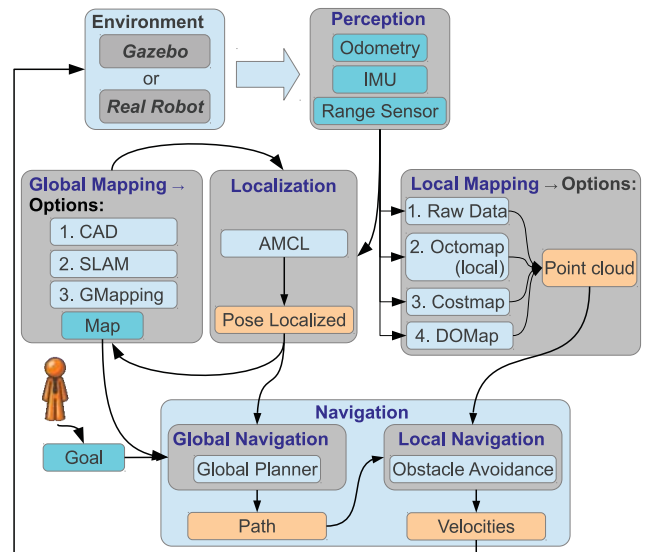


Fig. 2. ABSYNTHÉ's autonomous navigation flow diagram

In this work we are going to test the combination of obstacle avoidance algorithms with the four local mapping methods: raw data from the sensors, Octomap based mapping, Costmap and DOMap. In order to do that, we process the data provided by the mapping systems to give the same type of input to all avoidance algorithms. We have selected a Point Cloud structure because it represents 3D information but easily can be modelled like 2D information. The 3D information will be:

- 1) Raw data directly obtained from a 3D range laser or a depth camera. Also it can be obtained from a 2D range sensor and transform into 3D if the position of the sensor into the robot is known.
- 2) Costmap and Octomap divides the environment into cells (bi-dimensional or three-dimensional) so we transform the centre of each occupied cell into a 3D point.
- 3) DOMap also divides the world into cells of regular size, but instead of transforming occupied cells into a point, we transform the future estimation of the occupied cells in order to predict the movement of dynamic obstacles. Figure 3 shows how the environment is clustered into different objects, their relative velocities to the robot calculated and a Point Cloud built from this information.

IV. IMPLEMENTATION AND RESULTS

This section describes some implementation features and the experimental results obtained with the designed tests. Firstly we describe the test-bed and the platforms that we are using to test our proposal.

A. Test-bed

The environment to test our system was established indoor. We use several robots with different capabilities to test the whole system: Pioneer 3-DX, Pioneer 3-AT and a big outdoor

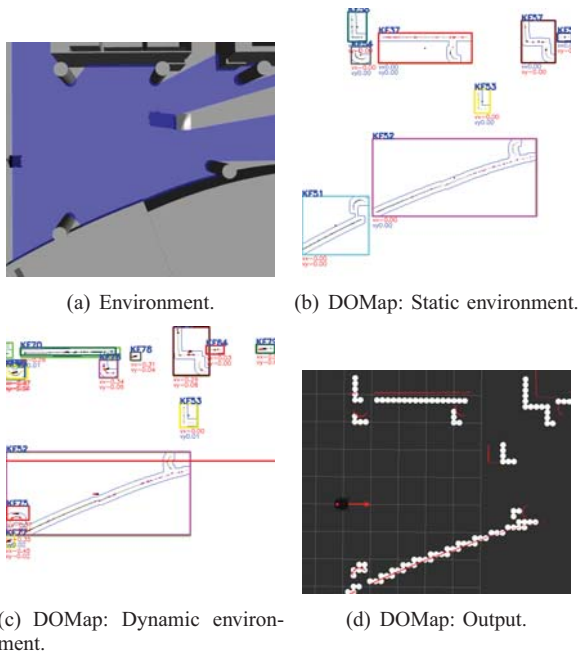


Fig. 3. DOMap: Perception stages. Fig. 3(a) Show the environment. Fig. 3(b) Show how the objects are detected, classified in blobs and tracking with Kalman Filters. Fig. 3(c) Show how the velocities is obtained and the Fig. 3(d) show the output of the system with the prediction poses of the objects.

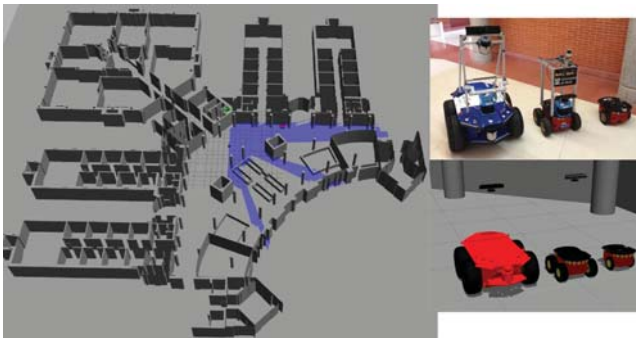


Fig. 4. Environment and robots used

robot Seekur Jr., all of them made by MobileRobots. The indoor environment dimensions are approximately 60x60 meters. Figure 4 shows the environment and the different robots that has been modelled into Gazebo/ROS for simulation and test purposes.

The platforms are equipped with different configurations: Pioneer 3-DX has 6 sonar and a Hokuyo URG-04LX laser parallel to the ground; Pioneer 3-AT is equipped with 12 sonar rangers, a Sick LMS 200 laser parallel to the ground, a Hokuyo URG-04LX laser that can be used parallel or angled to the ground, a Colibri IMU attached to the laser to obtain its attitude, and a Kinect sensor that can provide 3D information; Seekur Jr. is equipped with two SICK LMS 151 laser range finders, one of them angled to the ground, bumpers, and an IMU to reduce the uncertainty of the odometry. The control of each robot is made from a laptop with Ubuntu 12.04 LTS distribution and ROS Hydro. We also have a database server to provide the different targets and tasks that the robots should

execute.

B. Results

We have tested different scenarios: local mapping, local obstacle avoidance and global navigation.

1) *Local Mapping*: Octomap and Costmap present some issues in the map building when moving obstacles appear due to these two methods uses a similar system to mark a cell as occupied or free: if there is a number of sensor impacts into a cell it is marked as occupied, and to mark this cell again as clear an impact behind this cell is needed. Figure 5 represents an obstacle moving perpendicularly to the robot and an obstacle getting closer to it, as there are no walls behind the obstacle in the field of view of the sensor, each cell marked as occupied can not be marked again as free. This problem produces a trail of the object that marks as occupied cells that are really free and it can affect to the local navigation. Due to the DOMap decreases the occupancy probability of the cells without laser impacts, and the lack of memory of Raw Data, they have not these issues.

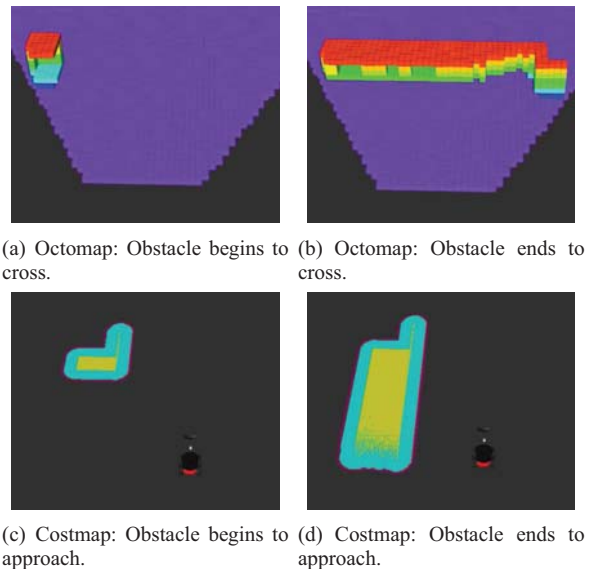


Fig. 5. Octomap and costmap mapping moving obstacles

2) *Local Obstacle Avoidance*: We have tested the behaviour of local obstacle avoidance with moving obstacles. In order to do that, we select an obstacle avoidance algorithm (Curvature Velocity Method) and tests four different mapping systems in three different scenarios: a moving obstacle getting closer to the robot, a moving obstacle crossing the robot's path and a moving obstacle overtaking the robot. The algorithm has the same configuration at each scenario, with its maximum velocities limited to 0.4 m/s and 40 degrees/s. The moving obstacle's size is 0.7 meters long, 0.4 meters width and 1.5 meters height, and moves at 0.4m/s when it is crossing the robot's path and 0.5m/s when it is overtaking the robot. On each scenario the robot should reach a target situated 6 meters ahead of it. We use a Pioneer 3-AT with a parallel laser to the floor.

For the obstacle getting closer to the robot scenario, figure 6 shows paths followed by the robot. The paths followed were similar but the algorithm's behaviour was slightly different. The Raw Data mapping does not have any memory of the obstacle, so the obstacle is avoided when it comes near the robot, what is potentially dangerous. Octomap and Costmap present the obstacle trail issue in the map building as we have probed before, so the obstacle is transformed into a wall separating the path from the ideal path. DOMap predicts the position of the obstacle in the future and its avoidance manoeuvre begins before Octomap or Costmap and follows a path closer to the ideal path.

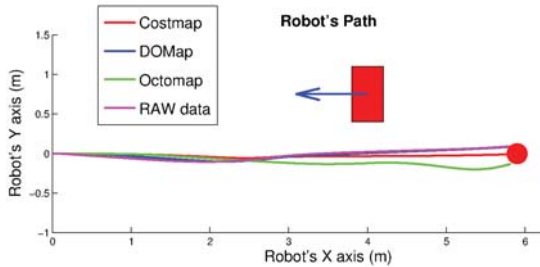


Fig. 6. Obstacle getting closer to the robot scenario: paths followed

For the obstacle crossing robot's path scenario, The Costmap and Octomap problem with trail obstacles is more acute in this case. Octomap is not able of clearing the trail and the algorithm avoid the path followed by the obstacle as if it was a wall. DOMap improves the path followed by Raw Data Mapping: the avoidance manoeuvre starts before because the algorithm predicts that the obstacle is not going to be in the path of the robot. Figure 7 shows the paths followed by the robot

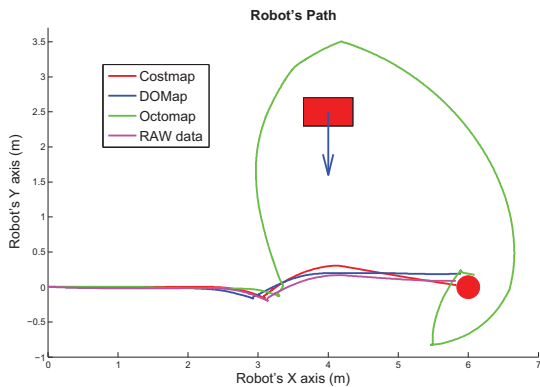


Fig. 7. Obstacle crossing robot's path scenario: paths followed

For the obstacle overtaking the robot scenario, Raw Data Mapping can be dangerous because it can crash with the obstacle due to its lack of memory. Octomap and Costmap have a better performance in this case because an obstacle moving away from the robot does not generate tail (cells are clearing as the obstacle moves away). DOMap predicts the future position of the obstacle and gets away from the trajectory of the obstacle making the obstacle avoidance manoeuvre longer but safer. Figure 8 shows the paths followed by the robot.

3) *Global Navigation:* We have tested the capabilities of the system to navigate into a full environment simulating a real ABSYNTH scenario: robot starts in a known position and has to reach an office situated in a small corridor. The environment is static so we choose costmap local mapping stage and test seven different obstacle avoidance algorithms.

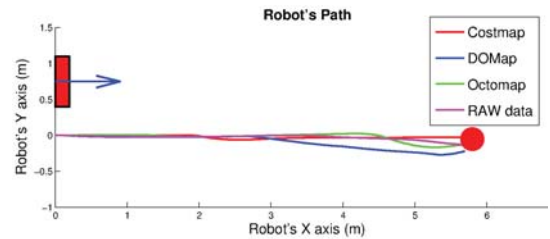


Fig. 8. Obstacle overtaking the robot scenario: paths followed

Figure 9 shows the path planned using a Dijkstra algorithm and the path followed by each obstacle avoidance algorithm. Every algorithm were able to ensure the security of the robot but only three of them were able to reach the target into the corridor: SND, Dynamic Window Approach and CVM.

Reactive algorithms like VFH+ can easily fall into local minimum and were not able to move, in this case it was stuck into a U-shape obstacle. The localization of the robot was not perfect and it affected the behaviour of the algorithm, especially to Elastic Band. This algorithm tried to follow the path planned but due to the robot was too close to an obstacle the algorithm failed and stopped the robot.

CVM based algorithms (CVM, LCM and BCM) were able to reach the corridor's entrance, but due to the intermediate planning stage of LCM and BCM, the entrance of the corridor was never the best selected as a target due to the size of its corridor. SND was able to reach the goal but its velocities were very low due to the complexity of the environment (especially in the corridor). Table I shows the smoothness and lengths of the paths and prove that Dynamic Window approach obtained the best results following the smoothest and fastest path.

TABLE I
GLOBAL NAVIGATION DATA

Algorithm	BCM	CVM	DW	Eband	LCM	SND	VFH
Lin. speed avg (m/s)	0.336	0.319	0.386	0.282	0.192	0.134	0.260
Rot. speed avg (rad/s)	0.019	0.010	0.011	0.021	0.043	0.004	0.050
Distance	53.28	51.59	49.52	12.61	40.75	48.12	15.12
Reach the goal	No	Yes	Yes	No	No	Yes	No

V. CONCLUSIONS AND FUTURE WORKS

In this work we have presented a revision of some several available navigation, mapping and localization packages in ROS, and we have described the integration of our own algorithms. The autonomous navigation of several platforms in simulated and real environments combining local and global algorithms have been tested in order to demonstrate that is robust and safe. Nowadays we are doing more real tests to increase the number of experiments and for generating more multimedia content that will be available in our Robesafe

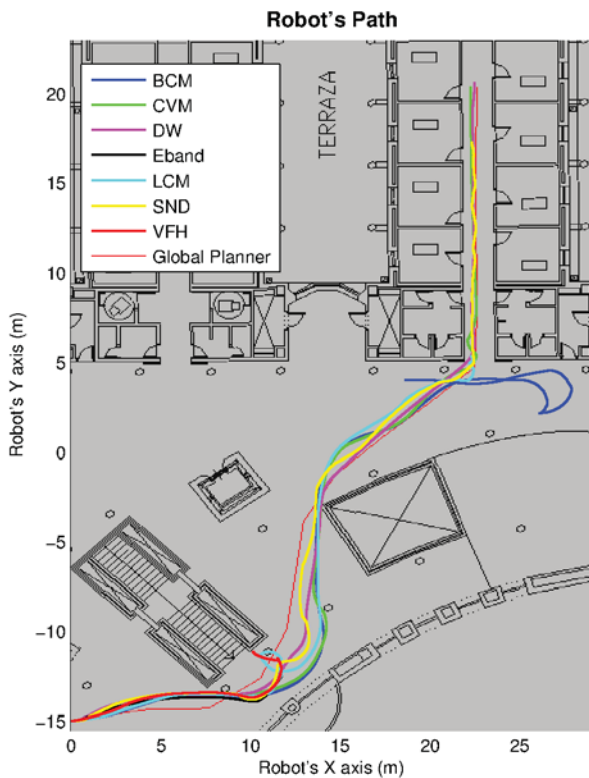


Fig. 9. Global navigation: paths followed by the robot

YouTube Channel (www.youtube.com/Robesafe). We are also working on adapting the algorithms to work in 3D environments distinguishing between navigable and non-navigable areas using a 3D range scanner.

VI. ACKNOWLEDGEMENTS

This work has been funded by TIN2011-29824-C02-01 and TIN2011-29824-C02-02 (ABSYNTHÉ project) from the Ministerio de Economía y Competitividad.

REFERENCES

- [1] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhne, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, 2008.
- [2] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Winning the darpa grand challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [3] P. Enge and P. Misra, "Special issue on global positioning system," *Proceedings of the IEEE*, vol. 87, no. 1, pp. 3–15, jan 1999.
- [4] D. Schleicher, L. M. Bergasa, M. Ocaña, R. Barea, and E. López, "Low-cost GPS sensor improvement using stereovision fusion," *Signal Processing*, vol. 90, no. 12, pp. 3294 – 3300, December 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0165168410002197>
- [5] M. Hentschel, O. Wulf, and B. Wagner, "A gps and laser-based localization for urban and non-urban outdoor environments," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, sept. 2008, pp. 149–154.

- [6] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.
- [7] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, p. 2007, 2007.
- [8] P. Newman, D. Cole, and K. Ho, "Outdoor slam using visual appearance and laser ranging," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, may 2006, pp. 1180–1187.
- [9] K. E. Bekris, M. Click, and E. E. Kavraki, "Evaluation of algorithms for bearing-only slam," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'06)*, 2006, pp. 1937–1943.
- [10] J. L. Blanco, J. A. Fernandez-Madriral, and J. Gonzalez, "Efficient probabilistic range-only slam," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'08)*, 2008, pp. 1017–1022.
- [11] J. L. Fernández, R. Sanz, J. A. Benayas, and A. R. Diéguez, "Improving collision avoidance for mobile robots in partially known environments: the beam curvature method," *Robotics and Autonomous Systems*, vol. 46, no. 4, pp. 205–219, 2004.
- [12] J. W. Durham and F. Bullo, "Smooth nearness-diagram navigation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, September 22–26, 2008, Acropolis Convention Center, Nice, France*. IEEE, 2008, pp. 690–695.
- [13] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 1, pp. 23–32, 1992. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=127236
- [14] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *In Proc. of the ICRA 2010 workshop*, 2010.
- [15] S. Thrun and A. Buecken, "Integrating grid-based and topological maps for mobile robot navigation," in *Proceedings of the AAAI Thirteenth National Conference on Artificial Intelligence*, 1996.
- [16] Á. Llamazares, V. Ivan, E. Molinos, M. Ocaña, and S. Vijayakumar, "Dynamic Obstacle Avoidance Using Bayesian Occupancy Filter and Approximate Inference," *Sensors*, vol. 13, no. 3, pp. 2929–2944, mar 2013. [Online]. Available: <http://www.mdpi.com/1424-8220/13/3/2929>
- [17] J. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," *Intel Corporation, Microprocessor Research Labs*, 2000.
- [18] P. Pfaff, W. Burgard, and D. Fox, "Robust monte-carlo localization using adaptive likelihood models," in *EUROS, 2006*, pp. 181–194.
- [19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [20] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *ICRA*. IEEE, 2007, pp. 1986–1991. [Online]. Available: <http://dblp.uni-trier.de/db/conf/icra/icra2007.html>
- [21] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *In Proceedings of the International Conference on Robotics and Automation*, 1993, pp. 802–807.
- [22] I. Ulrich and J. Borenstein, "VFH+: reliable obstacle avoidance for fast mobile robots," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, may 1998, pp. 1572–1577 vol.2.
- [23] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *Robotics Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, mar 1997.
- [24] N. Y. Ko and R. Simmons, "The lane-curvature method for local obstacle avoidance," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 3, oct 1998, pp. 1615–1621 vol.3.
- [25] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application," *Int. Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006, voir basilic : <http://emotion.inrialpes.fr/bibemotion/2006/CPLFB06/>. [Online]. Available: <http://hal.inria.fr/inria-00182004/en/>
- [26] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grid," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Rome, April 2007. [Online]. Available: <http://emotion.inrialpes.fr/bibemotion/2007/FSL07>

The Artemis Rover as an Example for Model Based Engineering in Space Robotics

Jakob Schwendner¹ and Thomas M. Roehr¹ and Stefan Haase¹ and Malte Wirkus¹ and Marc Manz¹
and Sascha Arnold¹ and Janosch Machowinski¹

Abstract— Future application of robotic missions in the space context will require the systems to have both mobility and manipulation capabilities. The limited direct communication with the systems due to visibility, and severe time delays also make it a requirement for the system to perform its actions mainly autonomously. The increasing complexity of the task, as well as the strict requirements for reliability and fault tolerance pose a significant challenge to both engineering and research activities. The SpaceBot Cup was held in November 2013 to probe those capabilities in the context of a competition. In this paper we present the Artemis rover and its software architecture as well as the competition results and lessons learned. Special attention is given to the modular design based on the Robot Construction Kit (Rock) framework – a component based software framework, which uses a component model based on the Orocos Real-Time-Toolkit (RTT).

I. INTRODUCTION

Since the days crewed space exploration was at its pinnacle with the Apollo program, robotic systems have replaced humans as the agents for the exploration of our solar system. The main advantage of these systems is that they are much more applicable with regards to the type of environment they can sustain. Even though artificial intelligence development has come a long way since the days of the Lunokhod [1], which was purely remote controlled, the latest systems to explore planets [2] or the return to the lunar surface [3] are still limited in what they can do on their own. In general it is preferable to have a human in the loop. It will be a fair while – if ever – until computers will be able to outmatch humans in coping especially with unforeseen situations. A difficult communication environment with delays and only limited connection windows pose limitations to the mission design. Advancing the level of autonomy of space systems improves the options available for difficult missions – usually the ones that are scientifically most interesting [4], [5], [6].

One way of supporting the advance of technologies is to use competitions. These type of events are suitable to foster creative ways of solving current problems, and generating new questions and engineering challenges [7]. One such competition was held by the German Space Agency (DLR) in Rheinbreitbach, Germany in 2013 [8]. The challenge of the SpaceBot Cup was to develop an autonomous mobile manipulation system within 8 months, and then show its capabilities in a 1 h mission. The task was to find and collect two objects in an unknown 21 m by 21.5 m area,

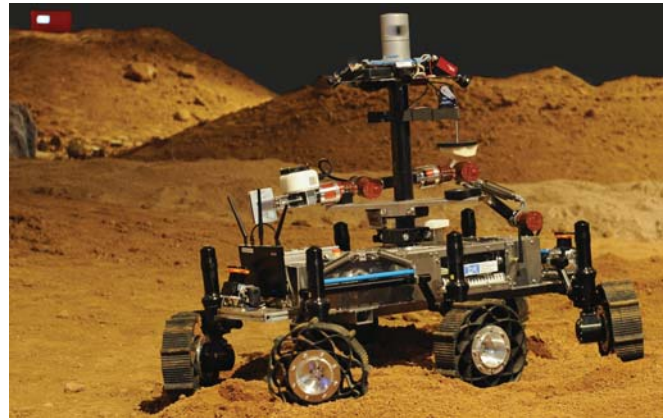


Fig. 1. The Artemis rover – shown at the SpaceBot Cup competition area – is a six wheeled system with a mass of 87 kg (including lander setup) and a size of 830 mm x 1300 mm x 500 mm (width x length x height).

and transport them to a third location and then return to the origin. Communication with the robot was severely limited, with a 2 s communication delay and multiple scheduled communication outages.

In this paper we describe the design process of Artemis (see Figure 1) to fulfill the functional requirements of hardware and software in order to perform the full mission scenario of the SpaceBot Cup.

In a first step the functional decomposition for each of the tasks was detailed. Subsequently, this decomposition allowed to distribute the work and to assign it to corresponding system experts in the fields of mechanical, electronic and software engineering. A fundamental requirement for the mission was autonomous navigation. In order to achieve this capability a further decomposition was performed into the following functional modules: 1) locomotion, 2) mapping, 3) and navigation comprising path planning and trajectory following. Since a coarse map was provided before the competition an exploration module has been considered optional – forwarding an a-priori list of waypoints for exploration was a valid approach that satisfied the needs of the mission scenario.

A critical functional element was the management of the autonomous activities, i.e. the integration layer for all functional components that were needed to fulfil the mission requirements. This management of Artemis is performed by the supervision [9] – a component dedicated to manage the activities of Artemis based on previously modelled high-level functionality which relies on a set of functional single-

¹German Research Center for Artificial Intelligence, DFKI Bremen, Robotics Innovation Center (RIC), Germany
firstname.lastname@dfki.de

purpose modules.

This paper presents Artemis as one approach of solving the complexity of the SpaceBot Cup scenario both on a hardware and software level. In addition to the description of Artemis and its design process, we describe the competitions results and provide a selection of the lessons learnt throughout the development and the competition.

II. SYSTEM DESCRIPTION

The SpaceBot Cup scenario requests capabilities in multiple fields of robotics, which led to three parallel lines of development: 1) navigation – the ability to get to a specific location 2) manipulation – the ability to manipulate objects 3) exploration and object detection – finding the target locations for navigation and manipulation. Each of these capabilities depends on development of hardware and software, while eventually both need to be managed by the supervision component. The following sections describe the approaches for each of the three development lines:

A. HARDWARE

The first main design driver for Artemis were the expected terrain characteristics with slopes up to 30° , loose surfaces with a variety of sand and stone fields. The second design driver was the required manipulation capability. The SpaceBot Cup participants had to find, identify and manipulate three different objects. The strategy of the Artemis team was to collect both mobile objects and directly transport them to the stationary third object. To cope with the demands the hardware development targeted a highly mobile platform to enable the system to traverse each part of the contest area. Additionally, the platform was equipped with a six degree of freedom manipulator and several sensors (Figure 2). In order to implement these capabilities with only one arm, the system was further equipped with storage devices for the cup and the battery.

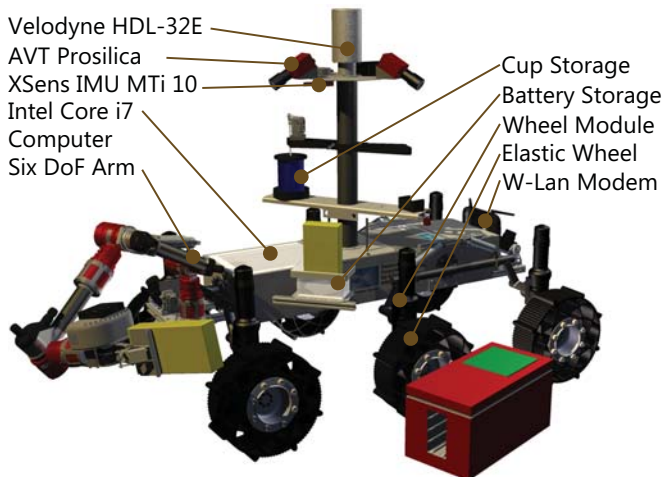


Fig. 2. The Rover with all subsystems (CAD drawing).

Passive suspension: Artemis locomotion platform consists of three single rockers, each is equipped with two fully actuated wheel modules. This enables the rover to drive in every direction from within any orientation and using fully actuated wheels facilitates the manipulation of objects with a six degree of freedom manipulator whilst retaining the possibilities offered by a seven degree of freedom manipulator. The chosen suspension is based on the 3-Bogie design which was proposed for the ExoMars rover [10]. This concept allows for a relatively light weight as well as for higher static stability compared with other concepts like the CRAB, RCL-E, and the Rocker-Bogie [11]. High static stability was a crucial precondition for the placement of a sensor mast at the top of the system. The final system reaches a static stability of $> 45^\circ$ in each direction.

Sensors: The sensor setup for the system is geared towards the individual requirements of the different software modules. The odometry requires the wheel encoder readings as well as an AHRS. Local obstacle avoidance is performed using a tilting laser range finder unit located before the front axis of the system to reduce sensor shadowing. The same unit is used – together with a small camera for color information – for the near range object identification and positioning system. The rotating laser range finder located on the mast of the system is used for the mapping subsystem. The three cameras which are also located on the mast are used for the long range identification of the objects. See Figure 2 for the exact sensor placements and models.

B. SOFTWARE COMPONENTS & SUPERVISION

The software development approach for Artemis has been model-based and component-based. Components in Rock are so-called oroGen components, i.e. Orocos components which have been generated from a specification file which describes the component's interface. The specification mainly defines input and output ports, operations and configuration parameters of a task. Based on this specification a binary - a so-called deployment - can be composed using various task models. This strategy is useful in multiple ways. Firstly, it speeds up the development process since framework specific code is automatically generated and a code skeleton is provided that allows to easily embed functionality that resides in framework independent libraries. Secondly, components are designed modularly and for reuse, e.g., a component for retrieving images from a camera or a path planning component can be easily used in different contexts.

Eventually, since each component comes with an explicit specification further modeling strategies can be applied. The previously mentioned supervision component allows to create high-level functionality using composition of multiple components. This additional modeling involves defining dependencies and data connections between components. Hence, compositions represent subnetworks of components and the supervision can manage these subnetwork during runtime to minimize side-effects of having subnetworks run in parallel and to optimize resource usage. Furthermore, the supervision also allows to perform model based vali-

dition of connections and can automatically compute the size of connection buffers using information given on a components update frequency. The supervision provides an abstract modeling layer and thus does not only apply to oroGen models, but also for ROS Nodes. ROS does not explicitly provide a specification for nodes but since ROS uses well defined interfaces the specifications have been extracted from existing nodes. Having interface specifications for both component types allowed to manage oroGen and ROS Nodes in parallel in the supervision.

C. NAVIGATION

The capability to accurately perform localization and mapping is crucial for a good performance in navigation. To model the environment and localize the robot, a pose-graph based approach for Simultaneous Localization And Mapping (SLAM) is used. As graph SLAM back-end we rely on the g2o graph optimization framework [12].

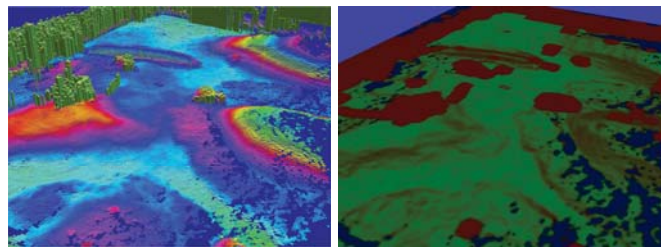
Since we do not model environment features separately, the graph only consists of pose vertices and edges representing constraints between poses. Each pose is associated with a full static 360° laser scan of the environment. New poses will be added depending on the euclidean distance the robot travels from its last known pose in the graph. Possible sensor movements during scan acquisition are corrected using odometry based transformation, so that a static 360° scan can be created for a single robot pose. Due to the high speed of the Velodyne laser scanner (10 Hz) the odometry error is acceptably small.

The transformations between the vertices, represented by edges, are optimized by an Iterative Closest Point (ICP) algorithm, using the known odometry based transformation as a starting point. In particular we use Generalized-ICP (GICP) [13], which has proven to perform quite well on 3D lidar data. After adding a new vertex additional edge candidates to the existing vertices are identified, depending on a maximum euclidean distance (d_{\max}). The edge candidates are prioritized and processed continuously from the top of the list. If GICP produces a valid solution for a candidate edge this edge will be added to the graph. To achieve a sufficiently connected graph d_{\max} should be at least three times the size of d_{\min} , in which d_{\min} is the euclidean distance between consecutive poses. The size of d_{\max} on the other hand is limited by the ability of GICP to perform valid matches in time.

The global graph optimization is executed every time a predefined number of new edges has been added to the graph. Through the strongly connected graph it is possible to reduce the impact of poor ICP alignments. Adding an outlier detecting approach like [14] or [15] can reduce this impact even more. To limit the memory consumption, older laser scans are deleted on basis of a 2D grid in which the scans are indexed based on the position where they were taken. Deleting older entries in an index cell limits the total number of scans, but assures sufficient number of scans in less covered areas.

As a runtime safety feature it is also possible to continue the SLAM from the latest known location. The current state of the graph is stored at shutdown of the SLAM module and loaded again on startup. If the robot has only moved slightly ($< d_{\min}$), while the SLAM module was stopped, it is possible to continue mapping without losing information or increasing pose uncertainty.

As an additional benefit the SLAM module is real-time capable and can operate nearly independently of the environment type. However, it cannot be applied to strongly ambiguous environments where the ICP algorithm tends to fail. To integrate well with the overall navigation an abstract map out of the aligned point clouds is generated on demand, e.g., when the global planner needs a new map, though this cannot be done in real-time. A result from the mapping module created during the competition run is shown in Figure 3. The generated map uses an extended version of the multi-level surface map (MLS) [16], [17] for representation.



(a) Multi-level surface map with colorized z-height. (b) Traversability map, increasing costs are visualized by a color changing from green to red.

Fig. 3. Maps of the SpaceBot Cup competition area.

To effectively use the generated maps for navigation, we use a hierarchical approach for path planning, i.e. the developed path planning consists of a global planner using D* Lite [18] and a local planner using VFH* [19].

The global planner creates the shortest trajectory from start to goal avoiding high slopes and obstacles. It is a grid based planner that works on traversability maps, which are generated from MLS maps by dividing the cells into cost-classes according to their slopes. Steeper slopes correspond to higher costs and slopes with an inclination above 31° are regarded as obstacles. In the global context the dimensions of the robot is assumed to be a circle. Therefore obstacle growing can be applied to provide a computational fast way to implement safety areas around non-traversable regions. For performance reasons the movement model of the robot is also simplified. It is assumed that the robot can move directly into all eight neighbor cells, next to cell representing the robot's position, at uniform movement cost. Although the global planner should be triggered at a low frequency, the D* Lite algorithm is used for fast and efficient replanning.

The local planner is designed to loosely follow the trajectory generated by the global planner. In contrast to the global planner it is not grid based, uses a complex motion model and takes the shape of the robot into account. To be able to perform the trajectory generation in real time, the

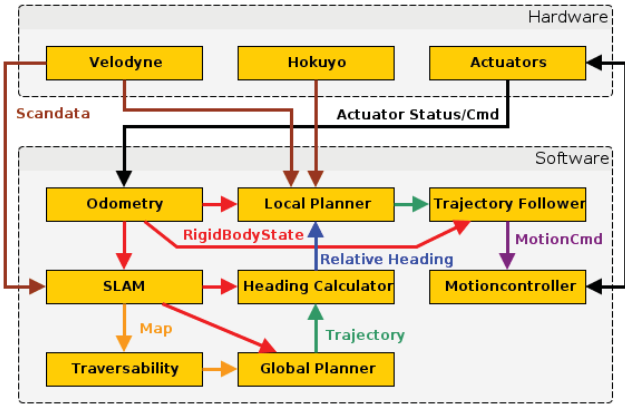


Fig. 4. Modular architecture of the navigation stack.

local planner only computes a trajectory to a horizon that is perpendicular to the vector from the robot to a target location on the global trajectory. The target location is computed by finding the nearest location to the robot on the global trajectory and advancing a certain distance on it. Additional speedup is achieved by only computing the traversability map for the local surrounding of the robot and by using the VFH algorithm to reduce the sample space of the local planner.

To avoid oscillations during the trajectory execution, caused by a 'jumpy' position provider, the local trajectory is transformed into the odometry coordinate system (OCS). An additional benefit this approach allows for a higher update rates of the local trajectory, which results in a smoother movement for the trajectory following.

The global planner is only run if the SLAM map changed, the user changed the goal position, or if the local planner failed. The failure of the local planner is a nominal case, as the global planner may plan through unknown terrain, which might turn out to be non traversable. In this case the global planner is rerun and should change the global trajectory in a way that it can be followed by the local planner. It should be noted, that both local and global planner have to be properly configured to avoid lockouts (see Section IV). The local planner runs almost continuously as it is triggered by position changes of the robot and new sensor input to the local traversability map. If it is triggered, the horizon is determined from the global trajectory and subsequently the local motion planning is performed. In contrast to the global planner the local planner treats unknown areas as obstacles. In the case that repeatedly no motion towards the horizon could be performed the local planner reports a failure to the global planner. Figure 4 shows the integration of all components of the navigation stack.

D. EXPLORATION & OBJECT-DETECTION

The strategy for solving the SpaceBot Cup challenge was to use different modules for the identification of objects from a distance, and an extraction of the objects full pose in the near field. The rationale for this is that searching for items is performed more effectively in the visual domain, while

extracting a precise pose for manipulation is better solved using 3D localization methods. The object detection method uses the images from the mast camera on which a blob extraction algorithm is applied. Regions with colors similar to the target color are marked as candidates. After simple consistency checking of the blobs using their apparent size, the candidates are projected into a grid map of the environment. Each occurrence of candidate leads to incrementing a candidate related counter in the cell. The cell with the highest count is used as the most likely position of the object.

Once the object is located, the 3D-pointclouds from the front laser scanner are used to find the pose of the object in the scene. This is done by looking for parts of the shape, e.g. planes or curved surfaces and then rate how these elements could be part of the wanted object. The extracted pose is then forwarded to the manipulation subsystem.

E. MANIPULATION

The SpaceBot Cup scenario requires manipulation skills to handle two types of objects: a block-like battery of 1 kg and cylindrical drinking cup holding approx. 0.2 kg of distilled water. The objects have to be grasped and put into their respective stow position on the rover. At the final location the objects need to be retrieved from their storage position respectively and assembled with the so-called base object – a custom-made scale. Figure 2 shows the battery in the manipulator hand and the cup in its stow positions.

We decomposed the different manipulation tasks into five basic abilities: *a)* motion planning to Cartesian and joint space goals, *b)* following joint trajectories, *c)* move the end-effector towards an attractor pose in Cartesian space, and *d)* execution of different grasps suitable for the different objects involved in the task.

For each of those abilities we separately developed components in Rock, mainly by integrating functionalities from existing software libraries, e.g. such as Reflexxes[20] for developing a trajectory controller.

In order to pick or place an object, a motionplanner has to compute a path which is free from self-collisions and collisions with its environment. For the motion planning task MoveIt! [21] software is used, which runs on the ROS framework. The motionplanner uses the current state of the robot and the environment information for generating the collision free path. These informations are given to the motionplanner node from Rock components, since the supervision component can manage ROS nodes and Rock components. Figure 5 shows the software components in ROS and in Rock in use on Artemis.

The gripper component provides the interface to two different grasps types, a flat and a spherical grasp for round objects. A grasp is defined by its type, the hand opening diameter and a reference force to be applied to the object. By observing motor torques and thresholding them, we detect successful grasps.

We implemented a Cartesian control component that uses the weighted damped least squares (WDLS) inverse kinematics solver of the KDL library [22] to generate joint motion

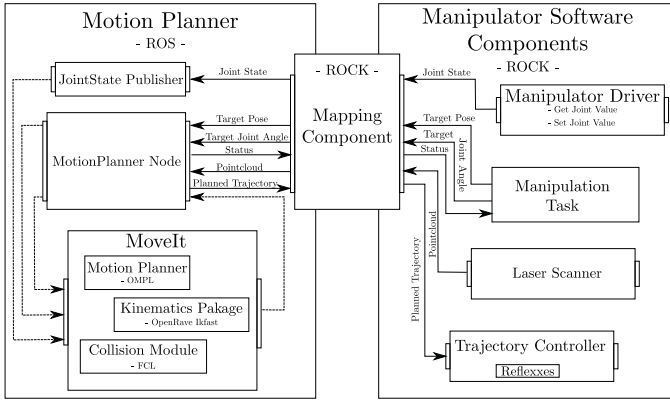


Fig. 5. Motionplanning as complementary activity of ROS nodes and Rock components

commands towards a Cartesian target.

Based on these components, we implemented the core abilities as parametrizable actions. As a bridge between the data-driven component networks and actions we need a mapping of the component network state (defined by the data that is on the network’s ports) to discrete events. In addition, we require to influence system behavior at a given moment in time. To achieve this, we used the following synchronization primitives: *a) port writers* trigger a behaviour of the component network by writing data to a specific port, *b) port readers* read from a port of component network and store data for later reuse in a state-overlapping memory, and *c) monitors* trigger an event when a configurable condition, which is a description of data on the ports of a component network, is fulfilled.

The discrete action for Cartesian control ”move_arm_cart” serves as example to illustrate the application of the synchronization primitives. This action takes the goal pose, tolerance boundaries, and a validation time as arguments. It instantiates the corresponding component network and triggers the desired behavior by writing the goal pose to the setpoint port of the Cartesian control component. In order to determine whether the desired pose is reached a monitor is attached to the control error port of the same component. It emits a success event when the absolute control error is inside the given tolerance boundaries for the given validation time. The algorithm used for Cartesian control can get stuck in local minima and a monitor is applied to observe the joint positions and detect this situation during execution. A failure event is triggered if joint positions do not change significantly for some time.

Artemis manipulation strategy builds upon fixed, taught-in movements wherever possible. To do so, we defined a home configuration – i.e. a joint configuration that is used as start-and/or end-point for most of our actions. Picking up objects in our approach is a sequence of approaching the object, preparing the grasp and after moving further towards the object closing the grippers. Similarly, the assembly of objects is performed as a sequence of planned movements to two target poses defined relative to the base object. Table I gives

Name of action	Dependencies	Explanation
Primitive actions		
exec_arm_traj	–	Execute a given trajectory
move_arm_cart	–	Move to a given target pose using Cartesian controller
move_arm_cart_p	exec_arm_traj, move_arm_cart	Plan and execute a trajectory to a given goal pose. Optionally use move_arm_cart if planning fails.
Planned movements with pre-defined goals		
move_home	move_arm_jnt_p	Plan and execute movement to home configuration
Taught in movements		
manip_store_arm	exec_arm_traj	Move arm to its store pose
manip_unstore_arm	exec_arm_traj	Move arm from store pose to home
gripper_open	–	Open the gripper
Complex operations		
execute_grasp	move_arm_jnt_p, gripper_open, move_arm_cart_p, gripper_grasp, move_home	Pre-grasp and grasp pose are given as arguments. Move arm to a pre-grasp pose, open gripper, move to grasp pose, grasp and lift object.
move_bat_to_store	move_home, exec_arm_traj, gripper_open	Transports object in hand to battery holder using a pre-defined trajectory
unstore_bat	exec_arm_traj, gripper_grasp	Remove object from battery holder using a predefined trajectory
Top-level operations		
store_bat	move_home, execute_grasp, move_bat_to_store	Grasp and store battery
assemble_bat	move_home, unstore_bat, move_arm_cart_p, gripper_open	Unstore battery and insert into base object.

TABLE I
HIERARCHY OF ACTIONS TO MODEL MANIPULATION TASKS FOR HANDLING THE BATTERY

an example of how manipulation tasks have been created for handling the battery.

III. COMPETITION RESULTS

The SpaceBot Cup was held over two consecutive days and the teams were given one day before to prepare at the given competition location and test their communication infrastructure. Teams were not allowed to test their robot in the competition site before the competition, but were bound to test on a small scale test-bed nearby. In the final competition each team got a time-slot of one hour to perform the full mission. Each team had three checkpoints where communication with the system was available for 5 minutes with a 4 s round-trip latency.

No team was able to complete the full mission, and the jury decided to not announce a winner. However, Artemis demonstrated outstanding locomotion capabilities. Though manipulation and object detection had also been prepared, we focus on locomotion and navigation in this result section. Since the exploration site was roughly known through low-resolution maps provided by the organizers, an exploration strategy had been predefined. The main strategy was to explore the site and meanwhile use the robot’s and operators’ object detection capabilities to locate the target objects. A coarse waypoint sequence was given to the robot in order to perform exploration and after reaching a waypoint the robot tried to advance to the next waypoint. Artemis started off by autonomously traversing large parts of the exploration



Fig. 6. Artemis negotiating an obstacle while navigating autonomously.

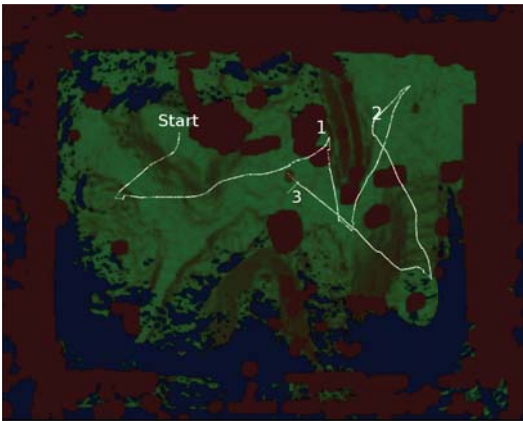


Fig. 7. Traversability map built during the run, with the travelled path during competition: (1) location of getting first time stuck, (2) location of getting a second time stuck and official end, and (3) end of autonomous navigation after exceeding the official competition time.

site (also cf. [23]). The site consisted of sandy terrain with rocky sections. Artemis easily overcame a section of loose soil (cf. Figure 1) where the flexible wheels showed their advantage. Subsequently, Artemis moved over a stone with a size of about two thirds of a wheel’s diameter (cf. Figure 6) – compensated by the passive locomotion platform. Finally, Artemis reached a trench. While the global planner computed a plan which led through the trench, the local planner did not allow traversing of the trench and eventually Artemis became stuck since the global planner did repeatedly lead the robot into a situation where the local planner prevented further movement.

Previous to the competition Artemis had been tested to climb up to 35° of inclination. Experiments after the competition using manual operation confirmed the outstanding locomotion capabilities. However, to fully exploit the capabilities the planning parameter required tuning which needed to be done during the run. Issues with the communication infrastructure eventually prevented parameter adaptation and showing off the full capabilities of the system during the competition.

IV. LESSONS LEARNT

There is much to be learnt from a competition and host as well as participants gain experience and can reflect on de-

velopment strategies, design choices and the implementation with respect to the final performance. Thus, the following discussion will take a critical look at the performance of Artemis in the competition and the decisions made during and prior to the competition.

Incremental goals and the critical path: The decisions made prior to the competition were influenced by an optimistic and ambitious attitude and the goal to complete the full mission – not just parts of it. While resource limitations existed only small time buffers could be accounted for, reducing integration and test time for the fully integrated system to a minimum. The initial focus was put on designing the high-level functionality of Artemis to allow for autonomous operation. However, the competition showed that manual interaction with the operator remains a critical and substantial element for error handling. Eventually, remote operation of the system was a single point of failure of Artemis. While the development targeted a fully autonomous system the risk of not achieving this goal was high. A dedicated approach for risk mitigation would have identified operations as an element of the critical development path leading to a (re)prioritization of the implementation task for operations.

Maintain a robust development procedure: The model-based development approach throughout all development phases proved to be highly beneficial – thanks to a well structured and proven workflow of Rock. Package management facilitated integration of external packages as well as the management of existing ones, and the general encapsulation of functionality in libraries served for good reusability of existing functionality. Furthermore, auto-generation of component’s framework code allowed to easily maintain the framework specific (oroGen) components and create interface contracts using well defined input and output types. Finally, creating components with standardized interfaces using Rock allowed to apply a proper system management tool.

Rock facilitates many tasks when developing a robotic system, but additional complexity arose using Rock components and ROS nodes in parallel. The supervision module was capable of handling both component types, yet this functionality was a recent development and as such did not have a perfect integration in the existing development workflow. We did not enforce the workflow early on all developers for testing and smoothing the process and suffered debugging efforts in later stages of the development. Thus, maintaining a robust and reliable workflow when developing complex systems should be given high priority.

The human factor in a component-based development approach: The theoretical benefit of a component-based system is the ease of integration. Artemis development showed that this assumption does directly depend on the maturity of the components, i.e. when interfaces including configuration properties require frequent updates integration becomes much harder. In contrast to the previously mentioned robust workflow for the component design, Artemis’ integration workflow to create high-level functionality from these components as part of the supervision showed some

weaknesses with respect to communication between system specialists and system integrators. System specialists used a different set of tools for performing small integration testing than the system integrator which led to some redundant work and a communication gap. The workflow for creating and testing high-level functionality should be homogeneous and allow to communicate and propagate requirements and semantics of components more clearly – ideally in a model-based fashion to allow verification.

Top down versus bottom up: A top down approach seems to be desirable for developments that focus only on a given mission scenario. Developing for Artemis started with a top down approach, but soon turned into a mixture of bottom up and top down approach. The mixed approach originated from reusing existing components with new experimental components where details of the implementation were unforeseen. From our experience, an experimental development approach seems to favour a bottom-up strategy in combination with an agile development approach. However, componentization and modularization is motivated by fostering reusability and a top down view will still be beneficial to identify generic, reusable parts. From our experience as soon as the development turns experimental the impact of a top down approach is severely limited and top down development efforts can stay on an abstract level without detailing interfaces precisely.

Testing: At the time of development we missed advanced offline unit-test facilities in the supervision to evaluate high-level functionality. Since these runtime tests had to be performed either in simulation or on the real system this led to a development slow-down. This stressed that unit-tests should not be missed at any level of hardware and software. Main parts of the initial development for Artemis relied on software simulation and the complementary application of simulation and real world testing led to an increase of the team's efficiency and allowed to cope with the short time frame for integration testing, since faulty behaviour of the real system could be fixed using simulation.

V. CONCLUSIONS & FUTURE WORK

The result of the competition showed that handling a complex navigation and manipulation scenario autonomously is more than putting the parts together. Although the individual parts required for the activities have been extensively researched in the past, the application of the individual skills in an integrated scenario provides additional challenges with many interesting open questions. On the software level the relation between task components, operations and failure management is likely a key element in the advancement towards robust autonomy in complex real world settings. Explicitly setting up the layout of the components of a system does not scale very well in terms of complexity and robustness. As an alternative, functional models of the components and their connections can be used for decomposition, validation and reconfiguration of the component networks. The Rock framework already provides many tools for the support of this approach, however there is much work to be

done in order to make it more accessible and robust. Events like the SpaceBot Cup are a great way to probe the abilities and force the evaluation of fully integrated systems and all the problems that come with it.

ACKNOWLEDGMENT

The work presented here is part of the Project "SpaceBot", which is funded by the Federal Ministry for Economics and Technology (BMWi) through the German Space Agency (DLR) grant number 50RA1318.

REFERENCES

- [1] B. Harvey, *Soviet and Russian Lunar Exploration*, ser. Springer Praxis Books. New York, NY: Praxis, 2007.
- [2] J. P. Grotzinger, "Exploring martian habitability. Habitability, taphonomy, and the search for organic carbon on Mars. Introduction." *Science (New York, N.Y.)*, vol. 343, no. 6169, pp. 386–7, Jan. 2014.
- [3] E. Lakdawalla, "China lands on the Moon," *Nature Geoscience*, vol. 7, no. 2, pp. 81–81, Jan. 2014.
- [4] C. Kunz, C. Murphy, and H. Singh, "Toward extraplanetary under-ice exploration: Robotic steps in the Arctic," *Journal of Field Robotics*, 2009.
- [5] S. Bartsch and T. Birnschein, "Development of the six-legged walking and climbing robot spaceclimber," *Journal of Field Robotics*, vol. 29, no. October 2008, pp. 506–532, 2012.
- [6] R. J. Léveillé and S. Datta, "Lava tubes and basaltic caves as astrobiological targets on Earth and Mars: A review," *Planetary and Space Science*, vol. 58, no. 4, pp. 592–598, Mar. 2010.
- [7] J. Schwendner and S. Joyeux, "Classifying Autonomy for Mobile Space Exploration Robots," in *i-SAIRAS 2010*, Sapporo, Japan, 2010.
- [8] T. Kaupisch and D. Noelke, "DLR SpaceBot Cup 2013 - A Space Robotics Competition," *Künstliche Intelligenz*, 2014.
- [9] S. Joyeux, F. Kirchner, and S. Lacroix, "Managing plans: Integrating deliberation and reactive execution schemes," *Robotics and Autonomous Systems*, vol. 58, no. 9, pp. 1057 – 1066, 2010, hybrid Control for Autonomous Systems.
- [10] S. Michaud, A. Gibbesch, T. Thuerer, A. Krebs, C. Lee, B. Despont, B. Schäfer, and R. Slade, "Development of the exomars chassis and locomotion subsystem," in *i-SAIRAS 2008*, 2008.
- [11] N. Patel, R. Slade, and J. Clemmet, "The exomars rover locomotion subsystem," *Journal of Terramechanics*, vol. 47, no. 4, pp. 227 – 242, 2010.
- [12] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [13] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Proc. of Robotics: Science and Systems (RSS)*, vol. 25, 2009, pp. 26–27.
- [14] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph slam," in *IEEE/RSJ Intelligent Robots and Systems (IROS)*. IEEE, 2012, pp. 1879–1884.
- [15] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2013.
- [16] R. Triebel, P. Pfaff, and W. Burgard, "Multi-level surface maps for outdoor terrain mapping and loop closing," in *IEEE/RSJ Intelligent Robots and Systems*. IEEE, 2006, pp. 2276–2282.
- [17] J. Schwendner, "Embodied Localisation and Mapping," Ph.D. dissertation, Universität Bremen, 2013.
- [18] S. Koenig and M. Likhachev, "D*lite." in *AAAI/AAAI*, R. Dechter and R. S. Sutton, Eds. AAAI Press / The MIT Press, 2002, pp. 476–483.
- [19] I. Ulrich and J. Borenstein, "Vf*: Local obstacle avoidance with look-ahead verification." in *ICRA*. IEEE, 2000, pp. 2505–2511.
- [20] Reflexxes, "The reflexxes motion libraries," 2013.
- [21] A. Sucan and S. Chitta, "Moveit!" 2013. [Online]. Available: <http://moveit.ros.org>
- [22] R. Smits, "Kdl: Kinematics and dynamics library." [Online]. Available: <http://www.orocos.org/kdl>
- [23] MrRheingold, "Spacebot Cup: Robotik Wettbewerb," retrieved March 25 2014, from <http://www.youtube.com/watch?v=vaDo4eMk2Go>.

Workshop on Modelling, Estimation, Perception and Control
of All Terrain Mobile Robots

ICRA 2014
ICRA 2014

