2016 14th International Conference on Control, Automation, Robotics & Vision
Phuket, Thailand, 13-15th November 2016 (ICARCV 2016)

Dynamic Sensor-Based Control

Sylvain Vandernotte*, Abdelhamid Chriette†, Philippe Martinet†, and Adolfo Suarez Roos‡
*IRT Jules Verne, Chemin du Chaffault, 44430 Bouguenais, France
Email: sylvain.vandernotte@irt-jules-verne.fr
†LUNAM-IRCCyN, 1 Rue de la Noe, 44300, Nantes, France
Email: philippe.martinet, abdelhamid.chriette@irccyn.ec-nantes.fr
‡Airbus Group Innovations, 13 Rue Pasteur, 92152 Suresnes, France
Email: adolfo.suarez@airbus.com

***Abstract*—Robots in industries are often used for repetitive tasks. Their motions rely on precise virtual model of their environment and they are not able to handle changes or unexpected events. This disqualifies them to perform tasks that would require precision in a non controlled environment such as an assembly task in real world. Visual servoing is a well known tool to control the robot using spatial sensors. It includes real world references at control law level. But, visual servoing and more generally sensor-based control schemes provide kinematic control law and do not consider robot dynamics. As consequences, tracking performances are poor and convergence behavior is hardly predictable. In this paper, we proposed a new control scheme considering second order sensor-based control law and robot dynamics. Our main goal is to enable full trajectory tracking in sensor-space. Additionally, the scheme is compatible with priority-ordered task sequencing and it can be also used within a hybrid control scheme where force control is considered. This new control scheme brings the possibility to make easier the robot task definition, dividing a complex positioning task into small easy-manageable ones. Multi-tasks operation has been validated in simulation by using MSC Adams software [1] and where the robot has to perform an engraving task on a surface.**

I. Introduction

Industrial companies intensively use robots for highly repetitive task such as welding or palletization. But most of robotic cells in manufacturing plants can be seen as automation cell, where only a part of the robot capacities is exploited. Robots can improve industrial competitiveness, increasing their productivity while reducing their functional costs. However, some tasks are still difficult to automate: pick-and-place for example requires an heavy calibration process. Moreover, performing assembly tasks in complex environment while keeping the robot safe for operators is still an active research field. Without a loss of generalities, it is possible to say that robots have low manipulation, mobility and perception capabilities out-of-the-box. These points has to be enhanced to exploit their full capacities.

One can imagine specifying a robot operation with a limited knowledge of the environment. When a human pick a bottle, he does not know anything about the pose of the bottle. The only information available are a view of his hand, a view of the bottle and the experience of how works his arm. The picking operation can be seen as decreasing the distance between its hand and the bottle. This is done through the control of his arm based on the visual information coming form his eyes. This can be transposed to all kind of manipulation.

In robotics, sensor-based control scheme is a common way to perform relative positioning taking into account a specific sensor signal, improving absolute precision with respect to local reference of the world. Many spatial geometric features has been studied : image points [2], depth [3], lines [4], etc. The main drawbacks is that most of sensor-based control schemes proposed only kinematic control laws. Robots can be controlled only though its end-effector kinematic screw or its joints velocities. These control laws ensure an exponential decrease of the monitored values in sensor-space. As consequences, the convergence behavior and the end-effector motion could not be predictable in operational space.

On the other side, task-based framework allows to split a complex problem into small priority-ordered tasks using a recursive null-space projector [5]. Kinematic multi-task sensor-based control scheme was successfully used to solve an assembly task [6], but no contact interaction has been considered.

Efforts were conducted to merge force contribution inside a sensor-based control scheme. Several methods can be listed here. The impedance control scheme [7] describes the robot as a virtual mechanical system Its parameters are chosen in order to limit dynamic performance in desired directions. Methods are proposed to merge impedance and admittance control scheme in one hybrid controller to benefits from both [8]. Parallel hybrid position force scheme or external hybrid position force scheme [9], [10] merge force control and spatial control using an inner-outer loop setup, where spatial control is the inner-loop. [11] used an hybrid force position control scheme where the position loop is based on a pose estimation sub-system and visual data. Concurrently, operational space control [12], [13], [14] was proposed to gather 6-DOF end-effector spatial constraints and force constraints in a unified task-based dynamic framework. The framework was applied to control complex redundant humanoids robot in specifying tasks for the hands and for the body posture.

The main goal of this paper is to propose a new sensor-based control scheme using the second order task function model and dynamic decoupling. One can see as a merge between classic sensor-based control and operationnal space control. We wish to enable full trajectory control in sensor space and ensuring the convergence behavior in sensor space as in operational space. The new scheme takes full benefits from operational space control and sensor-based control schemes as

978-1-5090-3549-6/16/$31.00

it is compatible with multi-tasks framework and hybrid control scheme to handle interactions.

This paper is organized with two main sections. Section II describes mathematical development of the novel formulation of sensor-based control scheme : the second order sensor-based task function is detailed and instantiated with the inverse dynamic model of the robot to ensure dynamic decoupling. The dynamic multi-tasks framework is presented and used to implement a hybrid parallel force position control scheme. Section III presents validation results. An engraving task over a surface is used to illustrate the new dynamic sensor based control.

II. Dynamic Sensor Based-Control

A. Kinematic sensor-based control

Kinematic sensor-based control scheme is based on the relationship between the motion of a geometric feature $\mathbf{s}$ in sensor space to the kinematic screw $\mathbf{v}_{\mathrm{S}}$ of the sensor. The considered feature can be a 2D image point coming from a visual camera sensor, or a geometric 3D plane extracted from a depth camera output. The link between these two quantities is named the interaction matrix $\mathbf{L_s}$, defined as follows:

$$\dot{\mathbf{s}} = \mathbf{L_s}\mathbf{v}_{\mathrm{S}} \tag{1}$$

The sensor kinematic screw can be linked the joint velocities using the robot kinematic matrix.

$$\dot{\mathbf{s}} = \mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\dot{\mathbf{q}} \tag{2}$$

where

- $\mathrm{F_S}$ and $\mathrm{F_N}$ are images of the sensor frame and the frame of the last robot link, respectively.
- ${}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}$ is the velocity spatial transformation matrix from $\mathrm{F_S}$ to $\mathrm{F_N}$.
- $\mathbf{J}$ the kinematic matrix of the robot expressed in frame $\mathrm{F_N}$.

Figure 1 shows how frames are disposed with respect the end-effector. Please note that the sensor is rigidly attached to the end-effector. From (2), one can obtain the control law that ensure an exponential decrease on all components of feature vector $\mathbf{s}$:

$$\dot{\mathbf{q}} = -\lambda\left(\mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\right)^{+}\left(\ \mathbf{s}-\mathbf{s}^{\mathrm{d}}\ \right) \tag{3}$$

where $(\cdot)^{+}$ denotes the Moore-Penrose pseudo-inverse. The goal of (3) is to minimize (converge to zero) the error between $\mathbf{s}$ and its desired value $\mathbf{s}^{\mathrm{d}}$ by acting on joint velocities of the robot. The motion shape of the camera frame $\mathrm{S_S}$ is hardly predictable and depends strongly of the chosen features. Knowing that 2D point will constrain only 2 degrees of freedom (DOFs) of the camera motion, one would need at least four 2D points to constraints all of the DOFs of the robot. The complete interaction matrix is then obtained by stacking the interaction matrix of each feature.

B. Feature acceleration

The second order model is obtained by differentiation of (1). The feature acceleration is expressed as :

$$\ddot{\mathbf{s}} = \mathbf{L_s}\dot{\mathbf{v}}_{\mathbf{s}} + \mathbf{H_s}\mathbf{v}_{\mathbf{s}} \tag{4}$$

Or, considering (2),

$$\ddot{\mathbf{s}} = \mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\ddot{\mathbf{q}} + \mathbf{H_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\dot{\mathbf{q}} + \mathbf{L_s}{}^{\mathrm{S}}\dot{\mathbb{T}}_{\mathrm{N}}\mathbf{J}\dot{\mathbf{q}} + \mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\dot{\mathbf{J}}\dot{\mathbf{q}} \tag{5}$$

Using the generalized inverse, it is possible to write (5) as follows:

$$\ddot{\mathbf{q}} = \left(\mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\right)^{+}\left(\ \ddot{\mathbf{s}}-\mathbf{b}\ \right) \tag{6}$$

where $\mathbf{b}$ is equal to:

$$\mathbf{b} = \mathbf{H_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\dot{\mathbf{q}} + \mathbf{L_s}{}^{\mathrm{S}}\dot{\mathbb{T}}_{\mathrm{N}}\mathbf{J}\dot{\mathbf{q}} + \mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\dot{\mathbf{J}}\dot{\mathbf{q}} \tag{7}$$

(4) and (5) introduce the hessian $\mathbf{H_s}$ that corresponds to the differentiation of the interaction matrix and takes into account the Coriolis acceleration. One can find in [15] a recursive computation of the quantity $\dot{\mathbf{J}}\dot{\mathbf{q}}$ based on the recursive Newton-Euler algorithm.

C. Dynamic decoupling

Hereafter is recalled the dynamic behavior of the robotic system, also known as the inverse dynamic model of the robot [15]:

$$\begin{aligned}\tau &= \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{Q}(\mathbf{q})\\ &= \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{H}(\mathbf{q},\dot{\mathbf{q}})\end{aligned} \tag{8}$$

$\mathbf{A}(\mathbf{q})$ is the $n\times n$ symmetric positive definite inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ the vector of Coriolis and centrifugal torques and $\mathbf{Q}(\mathbf{q})$ the vector of gravity contributions acting on the system.

By mixing (6) in (8), one obtain the new control law for one spatial constraint:

$$\tau = \mathbf{A}\left(\mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\right)^{+}\left(\ \mathbf{w}-\mathbf{b}+\mathbf{L_s}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\mathbf{A}^{-1}\mathbf{H}\ \right) \tag{9}$$

By taking $\mathbf{w}$ as a command input, one can ensure the system is fully linearizable. Hence, the system is equivalent to a double integrator in ideal case where $\mathbf{w} = \ddot{\mathbf{s}}$ in sensor space. It takes fully benefits from the non linear dynamic decoupling approach and allows to specify task function to constraint motion of the sensor frame $\mathrm{S_S}$. One can then define a tracking control scheme in sensor-space by setting $\mathbf{w}$ equal to:

$$\mathbf{w} = \ddot{\mathbf{s}}^{\mathrm{d}} + \mathbf{K}_v\left(\dot{\mathbf{s}}^{\mathrm{d}}-\dot{\mathbf{s}}\right) + \mathbf{K}_p\left(\mathbf{s}^{\mathrm{d}}-\mathbf{s}\right) \tag{10}$$

where $\mathbf{s}^{\mathrm{d}}$, $\dot{\mathbf{s}}^{\mathrm{d}}$ and $\ddot{\mathbf{s}}^{\mathrm{d}}$ are the desired position, velocity and acceleration of the selected features in sensor space. $\mathbf{K}_p$ and $\mathbf{K}_v$ are positive diagonal matrices for tuning linear behavior of the system: $\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = 0$, $(\mathbf{e} = (\mathbf{s}-\mathbf{s}^{*}))$

D. Multi-tasks: using the orthogonal projector

In the task function approach [16], the control law is expressed as:

$$\dot{\mathbf{q}} = \mathbf{J}^{+}\dot{\mathbf{e}} + \left(\mathbf{I} - \mathbf{J}^{+}\mathbf{J}\right)\mathbf{z}_{\dot{\mathbf{q}}_2} \tag{11}$$

where the first term represent the minimization of the primary task and the second term, the projection of the motion induced by the secondary task in the kernel of the first one. This has been generalized with the efficient task sequencing scheme [5]. The recursive function are shown in (12). It helps to divide the whole problem of controlling a complex kinematic chain into several simple tasks organized following a priority order.

$$\begin{aligned} \dot{\mathbf{q}}_i &= \dot{\mathbf{q}}_{i-1} + \left(\mathbb{J}_i\mathbf{P}_{i-1}\right)^{+}\left(\dot{\mathbf{e}}_i - \mathbb{J}_i\dot{\mathbf{q}}_{i-1}\right) \\ \mathbf{P}_i &= \mathbf{P}_{i-1} - \left(\mathbb{J}_i\mathbf{P}_{i-1}\right)^{+}\left(\mathbb{J}_i\mathbf{P}_{i-1}\right) \end{aligned} \tag{12}$$

for $i = 1, ..., k$, where $\mathbb{J}_i = \mathbf{L}_{\mathbf{s}_i}{}^{\mathrm{S}_i}\mathbb{T}_{\mathrm{N}}\mathbf{J}$ represents the task jacobian, $\mathbf{P}_i$ its corresponding null-space projector. Initial conditions are $\dot{\mathbf{q}}_0 = \mathbf{0}$, $\mathbf{P}_0 = \mathbf{I}$. The joint velocity vector realizing all the tasks is $\dot{\mathbf{q}}_k$.

We extend the recursive equation (12) to dynamic using the following recursion:

$$\begin{aligned} \tau_i &= \tau_{i-1} + \mathbf{A}\left(\mathbb{J}_i\mathbf{P}_{i-1}\right)^{+} \\ &\qquad \left(\ddot{\mathbf{e}}_i - \mathbf{b}_i + \mathbb{J}_i\mathbf{A}^{-1}\mathbf{H} - \mathbb{J}_i\mathbf{A}^{-1}\tau_{i-1}\right) \\ \mathbf{P}_i &= \mathbf{P}_{i-1} - \left(\mathbb{J}_i\mathbf{P}_{i-1}\right)^{+}\left(\mathbb{J}_i\mathbf{P}_{i-1}\right) \end{aligned} \tag{13}$$

for $i = 1, ..., k$. Initial conditions can be transposed from kinematic case. The torque vector realizing all the tasks is τ_k.

The use of a secondary objective is useful in both cases. It helps to control redundancy of the system and avoid motions as non controlled internal behavior.

E. Dynamic sensor-based control scheme in a hybrid control scheme

As said previously, dynamic sensor-based control scheme can be used within a hybrid control scheme to merge force and spatial constraints. Two hybrid scheme are available : the hybrid parallel which is based on a selection matrix to isolate contributions of the both scheme, and the hybrid external where the force control scheme modifies the input of the spatial one.

The equation (14) describes the control law of the hybrid parallel force position scheme where dynamic sensor-based scheme is responsible of the position part :

$$\begin{aligned} \tau = \mathbf{A}\mathbf{J}^{+}\mathbf{S}^{\mathrm{N}}\mathbb{T}_{\mathrm{S}}\mathbf{L}_{\mathbf{s}}^{+}\big(\mathbf{w}_{\mathbf{s}}(t) - \mathbf{b} \\ + \mathbf{L}_{\mathbf{s}}{}^{\mathrm{S}}\mathbb{T}_{\mathrm{N}}\mathbf{J}\mathbf{A}^{-1}\mathbf{H}\big) + \mathbf{J}^{\intercal}(\mathbf{I} - \mathbf{S})\mathbf{w}_{\mathbf{f}}(t) \end{aligned} \tag{14}$$

where $\mathbf{f}^{*}$ is the reference behavior of the force task, defined as:

$$\mathbf{f}^{*} = \mathbf{f}^{\mathrm{d}} + \mathbf{K}_{pf}\left(\mathbf{f} - \mathbf{f}^{\mathrm{d}}\right) + \mathbf{K}_{if}\int_{t_0}^{t}\left(\mathbf{f} - \mathbf{f}^{\mathrm{d}}\right)dt - \mathbf{K}_{vf}\mathbf{v} \tag{15}$$

$\mathbf{S}$ acts as an selection matrix and isolate contributions from the two different controllers. Note that due to the noise coming from force sensors, velocities are used as derivative counteraction rather than the derivative of forces.

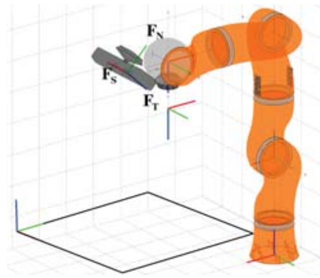


Fig. 1: Robot in stand-by configuration: Matlab/Simulink view

III. Simulation and Validation

The concept presented in the previous section has been validated in a co-simulation environment using Matlab/Simulink and MSC Adams softwares. An experiment has been performed in order to illustrate the new possibilities brought by this novel dynamic sensor-based control with sensor-space formulation, derived from the operational space formulation [12]. The simulation considered concern an example of engraving task. This can be generalized to a generic case where the robot has to apply a force along a specific direction while following a trajectory with respect to a local reference. By extension, this application will cover most real industrial cases such as sanding, tracing or assembly tasks. This concept can be applied to a torque-driven robot.

A. Simulation setup

The software MSC Adams is a multibody dynamics software that helps to evaluate and manage interactions between motion, structure and control, based on finite element analysis (FEA) solutions. It comes with a control integration plugin that brings the possibility to incorporate control problematic from other software into mechanical problems.

The robot used in simulation is the KUKA LWR4+. The robot is a redundant arm with seven degrees of freedom. It is equipped with a depth camera sensor mounted attached to its end effector. The camera and the tool are symbolized in figure 1 respectively by frames $\mathrm{F_S}$ and $\mathrm{F_T}$.

Matlab/Simulink is in charge of the scene definition, sensors interaction and the control of the robot based on development described in section II. The KUKA LWR4+ is modeled using the modified Denavit-Hartenberg convention. The frame $\mathrm{F_S}$ at the end effector of the robot represents the optical frame of the depth camera. The rigid body transformations are known. MSC Adams simulates the dynamic behavior of the robot.

B. Definition of the tasks

The experimental task has to fulfill three different objectives: align the z-axis of the tool with respect to the normal of the surface, apply certain amount of force along this axis and perform a trajectory with respect to a specific feature of the surface. Considering the framework presented in II, the problem is divided in two spatial constraints and a third force

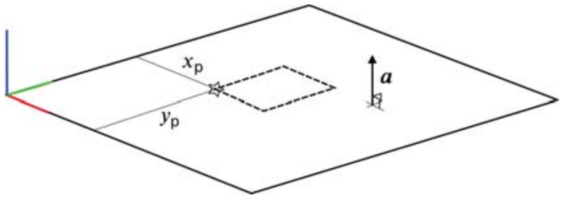


Fig. 2: Definition of features

constraint. One of the spatial constraints plays a specific sensor space trajectory using a fifth-order interpolation with constant acceleration blend [15].

The first spatial constraint is to align the z-axis (see figure 2). This task has the highest priority. The normal vector of the surface will be extracted from the depth camera flux. The control point is expressed in sensor frame S. The feature and its corresponding interaction matrix is defined as follows:

$$\mathbf{s}_{\text{axe}} = \mathbf{a}^{\intercal} = [\ x_{\text{axe}}\ y_{\text{axe}}\ z_{\text{axe}}\ ]^{\intercal} \tag{16}$$

where $||\mathbf{s}_{\text{axe}}|| = 1$.

$$\mathbf{L}_{\mathbf{s}_{\text{axe}}} = [\ \mathbf{0} \quad -[\mathbf{a}]_{\times}\ ] \tag{17}$$

The second spatial constraint is defined as follow. From the point cloud given by a 3D sensor (Kinect), the image plane of the table is extracted and one of a corner of the table, using a dedicated marker, is used as 3D point feature. Finally, the first two components of this point are considered. The feature and its interaction matrix are defined as follows:

$$s_{\text{p}} = [\ x_{\text{p}}\ y_{\text{p}}\ ]^{\intercal} \tag{18}$$

$$\mathbf{L}_{\mathbf{s}_{\text{p}}} = \begin{bmatrix} -1 & 0 & 0 & 0 & -z_p & y_p \\ 0 & -1 & 0 & z_p & 0 & -x_p \end{bmatrix} \tag{19}$$

The trajectory is defined to make a 5cm square around a reference point of the table with maximum velocities and maximum acceleration equal respectively to 5cm.s^{-1} and 0.5m.s^{-2}. Since the axis constraint has stronger priority upon the 2D point trajectory, the trajectory would be perform at least in a plane parallel to the table.

Force objective $\mathbf{f}^{\text{d}}$ along the z-axis is set to 10N. Spatial constraints does not fix the z-translation of the end effector, so that effector frame can move freely based on force control loop contribution.

Before proceeding, the robot starts from an closed pose above the point of interest on the table. Then, we apply the designed control (14) and the spatial constraints only are activated. The robot is expected to align the tool z-axis with respect to the measured surface normal due to the first task, while it keeps centered above the point of interest thanks to the second task.

The force task is activated once the first transition phase is completed. This is done by changing the weight in the selection matrix in (14) corresponding the the z axis from 1 to 0. Thus, the robot is expected to move forward the table and stabilizes itself at the desired applied force.

Finally, trajectory starts after the second transition phase is completed. Then, robot is expected to maintain the axis aligned as same as force objective, along the trajectory.

C. Simulation results

As said previously, the robot operation is divided in three phases. This paragraph highlights details about the features evolution. The surface has been placed with an small arbitrary orientation. This corresponds to the real case when model is confronted to the real world.

The first phase concerns the axis alignment task using the feature and interaction matrix defined in (16) and (17). The features evolution with respect to time is shown in figure 3. Since the model of environment is not well modeled, one can consider that the robot is in an uncertain pose. Then, the first task helps to compensate this default and error converges quickly to zero. The convergence time depends on specified proportional and derivative gains of the control law.

Note that the second task has been disturbed by the first (figure 3c). This is due the use of the orthogonal projector defined in (13) that puts stronger priority on the axis task over the point task. The second task converges slower than the first, meaning that the priority order is respected.

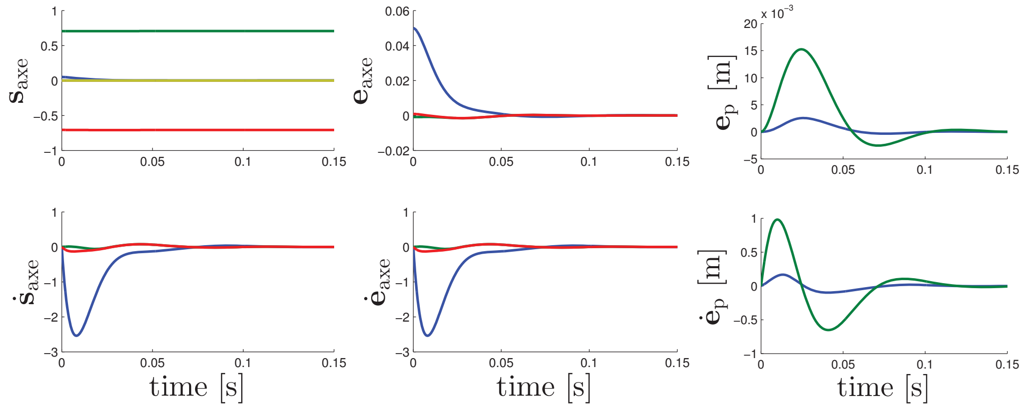
Force task is activated during the second phase. Figure 4 shows the feature evolution. A desired force is set at $t = 0.25s$. As robot is above the table, it starts to move downward against the table. Contact is established at $t = 0.3s$. A vertical line is plotted to symbolize the contact event. Following the definition of the force contribution through equations (14) and (15), the force task acts directly on the joint torques and therefore and its action is parallel to the spatial constraints contribution. A direct consequence is that it strongly perturbs the spatial tasks as long as the contact is not established.

Once contact is done, the force task quickly converges to its desired value. The convergence behavior is linked to proportional and integral gains. A less-oscillatory response can be obtained at price of a slower convergence. The others tasks stop to diverge and converges again to zero. Phase 2 has finished when the force stabilizes around its desired value.

At this point, the robot is in contact with the surface and applies its desired force along the normal axis of the feature.

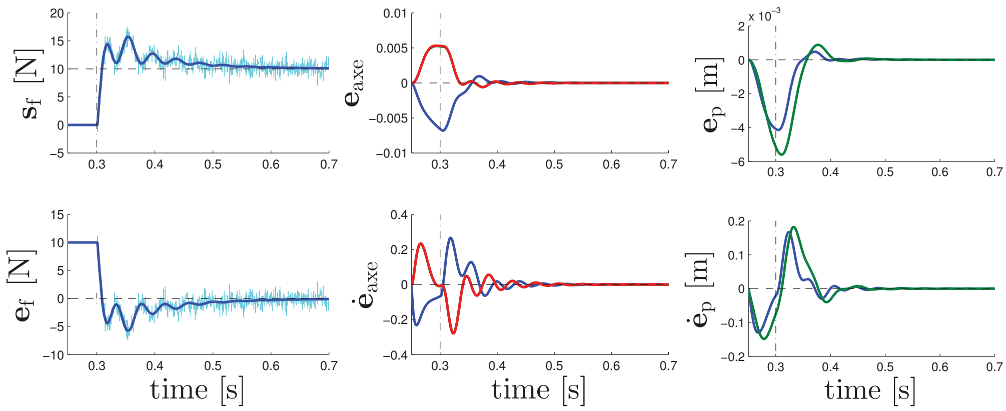
The next phase is about performing the trajectory defined around a point of interest on the table. Figure 5a and 5b show the trajectory in sensor space. The acceleration profile of the feature is not shown here. All along the trajectory, the position error is less than $1mm$, while the velocity maximum error is about $0.2mm$ during the linear phase. Spikes in velocity error plot are due to the $\ddot{\mathbf{s}}^{\text{d}}$ term in the reference behavior of the task (10).

It has to be noticed that the operation has been simulated with and without sensor noise. We especially set a Gaussian noise with a standard deviation equal to $1N$, which represents 10% of the objective. Results of the simulation with noise have been added to all figures. In general, control law is robust to noise and perform as expected. On can see that spatial constraints are not too much affected by noisy measures.



(a) (b) (c)

Fig. 3: Phase 1: axis alignment while maintaining the tool above the point of interest ; (a) Position and velocity component of the first task (axis alignment) ; (b) Position and velocity error of the first task with respect to desired ; (c) Position and velocity error of the second task (2D point).



(a) (b) (c)

Fig. 4: Phase 2: activation of force task at $t = 0.25s$; (a) Force profile and error with respect to desired ; (b) Position and velocity error of the first task (axis) ; (c) Position and velocity error of the second task (point).

Finally, the entire simulation shows that the robot operation performs correctly all along the three phases. The dynamic sensor-based control scheme, even using multi-task, fulfills the dynamic decoupling as expected.

IV. Conclusions

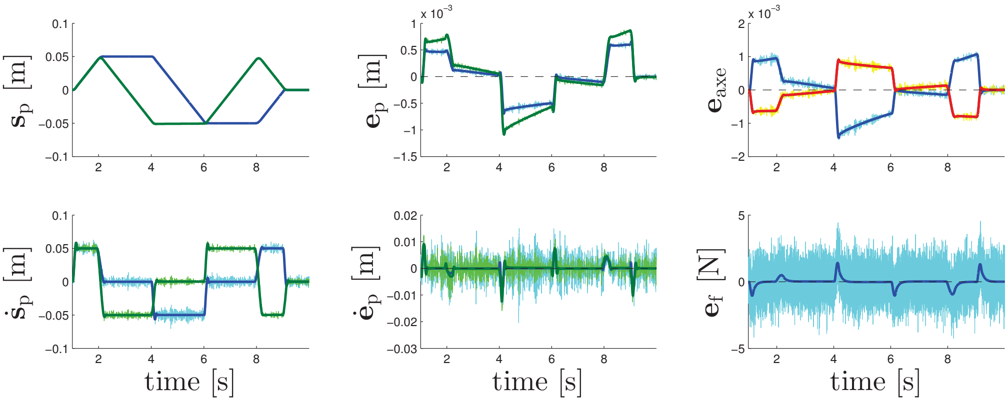
In this paper, a new dynamic sensor-based control scheme is presented. The purpose is to develop a framework able to perform trajectory tracking in sensor space. It has been shown that the scheme is compatible with hybrid controller when force control is required. The scheme is also compatible with multi-task prioritization. The framework brings us the possibility to perform accurate relative positioning with chosen features of the real world.

The simulation has been built based on real cases where a specific force has to be applied following one direction while performing a trajectory. Most industrial process require these needs such as engraving and can be transposed to others kind of operations as sanding or assembly. The framework now allows to specify an entire robot process with simple and easy-to-handle elementary tasks. The simulation shown the robustness of our approach to sensor noise and especially with heavy noisy force measurements.

The next step is to validate this result in a real robot cell. The main difficulty is being able to perform torque control which is not always possible on industrial robots.

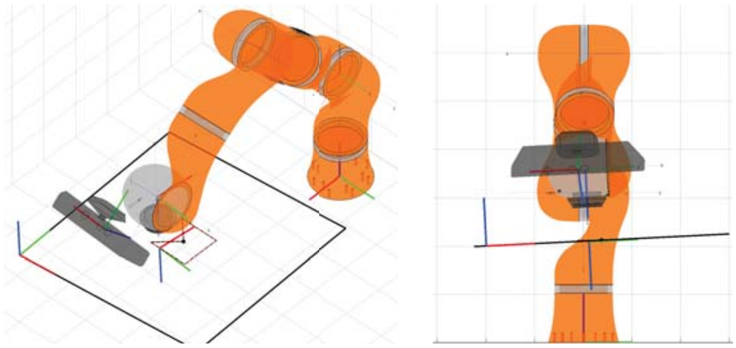
Acknowledgment

This work is supported by the ASIMOV project managed by IRT Jules Verne (French Institute in Research and Technology in Advanced Manufacturing Technologies for Composite, Metallic and Hybrid Structures).



(a) (b) (c)

Fig. 5: Phase 3: trajectory execution of 2D point task; (a) Position and velocity profile of the 2D point trajectory ; (b) Position and velocity error of the point during trajectory ; (c) Error profile of first task and force task along trajectory.



(a) (b)

Fig. 6: Robot view during process at $t = 2.64s$; (a) Overview: tool is on its trajectory in contact with the table ; (b) Front view: the tool axis is correctly aligned with the table one. ;

REFERENCES

[1] Adams, "Msc adams: Multibody dynamics simulation software," http://www.mscsoftware.com/fr/product/adams, 2016, [Online; last access in 2016].

[2] F. Chaumette and S. Hutchinson, "Visual servo control: I. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[3] E. Cervera, A. Del Pobil, F. Berry, and P. Martinet, "Improving image-based visual servoing with three-dimensional features," *The International Journal of Robotics Research (IJRR)*, vol. 22, no. 10-11, pp. 821–839, 2003.

[4] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.

[5] B. Siciliano and J.-J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Proceedings of the Fifth International Conference on Advanced Robotics (ICAR): Robots in Unstructured Environments*, 1991, pp. 1211–1216.

[6] S. Vandernotte, A. Chriette, A. Suarez Roos, and P. Martinet, "Performing assembly task under constraints using 3d sensor-based control," in *Proceedings of the 13th International Conference on Intelligent Autonomous System (IAS13)*, 2014.

[7] L. Villani and J. De Schutter, "Force control," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 563–583.

[8] C. Ott, R. Mukherjee, and Y. Nakamura, "A hybrid system framework for unified impedance and admittance control," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3-4, pp. 359–375, 2015.

[9] V. Perdereau and M. Drouin, "A new scheme for hybrid force-position control," *Robotica*, vol. 11, no. 05, pp. 453–464, 1993.

[10] Y. Mezouar, M. Prats, and P. Martinet, "External hybrid vision/force control," in *Proceedings of the International Conference on Advanced Robotics (ICAR?07)*, 2007.

[11] V. Lippiello, B. Siciliano, and L. Villani, "Interaction control of robot manipulators using force and vision," *International Journal of Optomechatronics*, vol. 2, no. 3, pp. 257–274, 2008.

[12] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[13] N. Mansard, O. Khatib, and A. Kheddar, "A unified approach to integrate unilateral constraints in the stack of tasks," *IEEE Transaction on Robotics*, vol. 25, no. 3, June 2009.

[14] C. Ott, A. Dietrich, and A. Albu-Schäffer, "Prioritized multi-task compliance control of redundant manipulators," *Automatica*, vol. 53, pp. 416–423, 2015.

[15] W. Khalil and E. Dombre, *Modeling, identification and control of robots*. Butterworth-Heinemann, ISBN:9780080536613., 2002.

[16] C. Samson, B. Espiau, and M. L. Borgne, *Robot Control: The Task Function Approach*. Oxford University Press. ISBN:0198538057, 1991.