Uncertainty-aware Navigation in Crowded Environment

Emmanuel ALAO[1], and Philippe MARTINET[1]

***Abstract*— Robots are now widely used around humans, in homes and public places like the museums, all due to their many benefits. These autonomous robots are called social or service robots and they always find it difficult to navigate in crowded environments; largely because of the high level of uncertainty in observing and predicting human behaviours in a highly dynamic environment. Uncertainty is propagated during prediction and might grow to levels that renders the whole environment unsafe for the robot leading to the so-called Freezing Robot Problem – FRP. This work presents our proposed approach to proactively account for various uncertainties during the robot's motion planning using a stochastic Nonlinear Model Predictive Controller (SNMPC). Additionally, using numerical optimization methods enables the planner to compute new control commands in realtime.**

I. INTRODUCTION

In recent years, there have been growing interests in service robots used explicitly in daily applications at homes, offices and public places such as indoor shopping malls, in museums, or in hospitals [1]. Most of the services they perform require safe and efficient navigation in crowded environments which is still an open problem in robotics [2] and [3]. However, results from a long-term study of the workflow in some public environment also show that the actions of state-of-the-art service robots often fail to fulfil people's expectations [4]. The state of the art approaches that have been employed in path planning with a focus on social awareness, uncertainty and solving the Freezing Robot Problem –FRP, are usually an extension of multi-agent collision avoidance and learning-based techniques with additional parameters that makes the algorithms socially compliant [5]. Classical motion planning algorithms generally employ the deterministic model by ignoring uncertainty, which may be sufficient in static environments or when an accurate information about future locations of the humans are available (an example is using motion capture cameras). A novel solution to the FRP problem is to use more informed models that constrain the uncertainty by assuming a low and constant predictive covariance in each agent's uncertainty, using a probabilistic model close to a near perfect prediction of the human motion [6]. In [7] it was shown that even with more accurate predictive models, FRP still occur given a dense enough crowd. The Social Force Model (SFM) [8], [9], is a state of the art model for representing human-human interaction and has been extended to human-robot interaction. It is based on the proxemics theory which defines the different invisible regions of intimacy around people. Humans tend to move towards their destination while avoiding obstacles and other pedestrians [10]. However, ESFM is a reactive planner that does not proactively predict the uncertainty in human motion [11], for instance the robot can easily collide with humans moving fast. Uncertainties in paths modelling and predictions are usually modelled as Gaussian processes and have been applied to pedestrian trajectory in [12]. A pioneering approach using an interacting Gaussian processes (IGP) was developed in [7] and [6] based on coupled output Gaussian Processes. FRP was perceived as a joint collision avoidance and cooperative planning problem amongst the crowd. An Hybrid Reciprocal Velocity Obstacle (HRVO) introduced in [13] was combined with ESFM to form a Proactive Social Motion Model (PSMM). A recent solution to the FRP by [14] uses a planner that efficiently captures the dynamics of obstacles in a crowded environment. The possible paths between the pedestrians were referred to as dynamic channels, which were formulated as a graph search in a triangulation space. In order to obtain socially better results, [15] uses various models of possible human scenarios such as group motion to augment the prediction and induce a cooperative motion behaviour in the robot. Cutting edge machine learning approach like inverse reinforcement learning (IRL) in [16] have also been employed to learn pedestrian decisions and interaction model from human trajectory data. They are limited in scale and will require huge amount of data to be generalised to several crowded environments. Our approach is closely related to [17] and [18] where the uncertainties in the system are modeled as a stochastic model predictive control problem, our methodology explicitly impose social behaviour and prevention of the freezing robot problem based on the social force model and the dynamic channel. The paper is organized as follows: section II presents some preliminaries, section III develops the proposed approach, section IV shows the results obtained, and finally section V concludes the article.

II. PRELIMINARIES

During path planning, most research assume that, there exist a sensor in the global (inertial) frame, while some perform a transformation of the state of the pedestrian from the robot frame to the Inertial frame before computing the optimal trajectory. Both approach lead to the propagation of the errors in the SLAM algorithm to the trajectory planner which is undesirable. In this work, a more realistic situation is considered, where the robot can only rely on sensors with limited range mounted on the robot, for instance a lidar scanner with about 5-10 meters range. Hence, the robot will try to estimate only the state of pedestrians within the sensor range, and then find an optimal path in a dynamic channel that leads to the goal as shown in figure 1.

[1]ACENTAURI team - Centre Inria d'Université Côte d'Azur

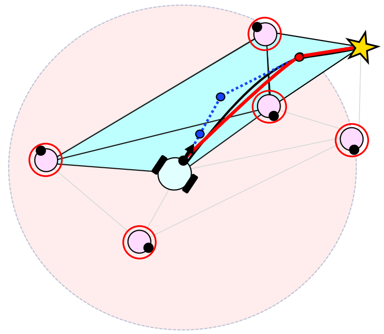


Fig. 1: Uncertainty-aware Dynamic Channel

A. Robot

The state of the robot in the environment can be described as $\mathbf{x}_r(t) = [x_r(t), y_r(t), \theta_r(t)]^T$ and the control inputs $\mathbf{u}(t) = [\boldsymbol{v}(t), \omega_r(t)]^T$, with $x_r(t)$ and $y_r(t)$ representing the position of the robot and $\theta_r(t)$ its orientation, $\boldsymbol{v}(t)$ is its linear velocity vector and $\omega_r(t)$ is the angular velocity all as a function of time. Besides the robot state, another important property of the robot to be considered in the path optimization problem, are its kinematic limits, such as its velocity and acceleration during motion. For the velocity we have,

$$\mathbf{u}_{min} \leq \mathbf{u}(t) \leq \mathbf{u}_{max}, \quad \forall t \in [t_0, \quad T], \tag{1}$$

where t_0 and T are the current time and the total motion time.

B. Pedestrian Model

For safe and efficient navigation, the robot needs to predict the motion of the pedestrians. The motion of humans in a crowd cannot be predicted precisely. Good results can be obtained using a Bayesian filter like the Kalman filter to track pedestrian position and estimate their states with the Constant velocity (CV) motion model:

$$\begin{aligned} \mathbf{x}_p(t) &= A\mathbf{x}_p(t) + \mathbf{w}(t) \\ \boldsymbol{z}_p(t) &= C(t)\mathbf{x}_p(t) + \boldsymbol{\xi}(t) \end{aligned} \tag{2}$$

where $\mathbf{x}_p$ represents the pedestrian state and $\boldsymbol{z}_p(t)$ the measurement at time t. Also, $\mathbf{w} \sim \mathcal{N}(0, Q)$ is the process noise and $\boldsymbol{\xi} \sim \mathcal{N}(0, R)$ is the measurement noise. Both are assumed independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance Q and R respectively. However, this model fails when the estimation is done in the robot frame. A model that considers the motion of the moving robot is proposed in this work.

C. Uncertainty and Optimization Problem

Robot, pedestrians, and other obstacles in the environment that influence the motion planning problem are required to formulate the constraints and variables of the optimization problem. A typical optimal control problem that satisfy the earlier mentioned conditions can be formulated as:

$$\min_{\{\mathbf{x},\mathbf{u},\mathbf{p}\}} \int_{t_0}^{T} [l(t, x(t), u(t), p) + E(T, x(T), p) \tag{3a}$$

$$\text{s.t } x(t_0) = x_0, \tag{3b}$$

$$\dot{x}(t) = f(t, x(t), u(t), p) \qquad \forall t \in [t_0, T], \tag{3c}$$

$$y(t) = h(t, x(t), u(t), p) \qquad \forall t \in [t_0, T], \tag{3d}$$

$$0 \geq c(t, y(t), u(t), p) \qquad \forall t \in [t_0, T], \tag{3e}$$

$$0 \geq c_{end}(T, x(T), p) \tag{3f}$$

here, $\mathbf{x} \in \mathbb{R}^{n_x}$ denotes the n_x dimensional state vector of the robot, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the n_u dimensional vector of control inputs, $p \in \mathbb{R}^{n_p}$ is the n_p dimension of free parameters in the system equation. The OCP aims to minimise an objective function (3a) that contains a running-cost (or *Lagrange term*) l and an end-cost (or *Mayer term*) E. The optimization problem can be subjected to an initial constraint on the states x_0, a dynamic constraint (3c) (usually the system ODE), algebraic constraint (3d), as well as initial and end boundary constraints on the states, controls and parameters.

III. METHODOLOGY

This section dives into the presentation of the proposed model for estimating the pedestrian motion, followed by the formulation of the additional objective and constraint functions for the Stochastic NMPC, then the creation of the uncertainty-aware dynamic channel. We assume a discrete time system in the rest of the paper.

A. Pedestrian State Estimation Model

The failure of the Constant Velocity (CV) model in Eqn. 2, comes from the changing frame of the robot during motion, which is why we propose a model that takes into account the motion of the robot:

$$\mathbf{x}^i_{p,k+1} = \tilde{\mathcal{R}}(\delta\theta) \left[\begin{bmatrix} \mathbf{p}^i_{p,k} + \mathbf{v}^i_{p,k}\Delta t - \boldsymbol{v}_{r,k+1}\Delta t \\ \mathbf{v}^i_{p,k} \\ r^i_{p,k} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\mathrm{w}_k \Delta t^2 \\ \mathrm{w}_k \Delta t \\ \mathrm{w}_{r,k} \end{bmatrix} \right] \tag{4}$$

where

$$\tilde{\mathcal{R}}(\delta\theta) \triangleq \begin{bmatrix} \mathcal{R}(\delta\theta) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}(\delta\theta) & \mathbf{0} \\ 0 & 0 & 1 \end{bmatrix} \quad \mathcal{R}(\delta\theta) = \begin{bmatrix} \cos(\delta\theta) & \sin(\delta\theta) \\ -\sin(\delta\theta) & \cos(\delta\theta) \end{bmatrix}$$

$\mathbf{x}^i_{p,k} = [\mathbf{p}^i_{p,k}, \mathbf{v}^i_{p,k}, r^i_{p,k}]^T \in \mathcal{X}_p \subset \mathbb{R}^5$ denotes the state (position, velocity and radius) of pedestrian $i \in N_p$, $\mathbf{u}_{k+1} = [\boldsymbol{v}_{r,k+1}, \omega_{r,k+1}]^T$ are the robot control inputs at time step $k+1$, and $\delta\theta = \omega_{r,k+1}\Delta t$ is the instantaneous change in robot orientation. The robot control inputs can be obtained from odometry or directly from the NMPC algorithm, note that we get back the constant velocity (CV) model when $\mathbf{u}_{k+1} = \mathbf{0}$.

The Extended Kalman Filter (EKF) is used to estimate the state of the pedestrians. The EKF is a parametric Bayesian filter that can estimate the state of nonlinear differential functions.

B. Pedestrian Tracking

The fact that the sensor is onboard the robot and their limited range implies that the observed pedestrians will quickly enter and leave the range of the robot. Hence, pedestrian tracking is done via a three (3) steps approach:

- Prediction: using the proposed motion model and the EKF, predict the next state of each tracked pedestrian and their covariance estimate.

$$\forall i \in N_p, \text{ predict} \begin{cases} \hat{\mathbf{x}}^i_{p,k+1|k} \\ \mathbf{P}^i_{k+1|k} \end{cases}$$

- Gating: use the measurement residual $\tilde{\mathbf{y}}_{k+1}$ and the residual covariance $\mathbf{S}_{k+1}$ from EKF, to compute the mahalanobis distance between the predicted state $\hat{\mathbf{x}}_{i,k+1}$ and the set of observations (Obv).

$$d^2_{Mh}(i,j) = \tilde{\mathbf{y}}_{k+1}(i,j) \quad \mathbf{S}^{-1}_{k+1}(i) \quad \tilde{\mathbf{y}}^T_{k+1}(i,j) \qquad (5)$$
$$\forall i \in N_p, \forall j \in Obv$$

A validation gate is then applied to prune the observations $d^2_{Mh} < \mathbf{G}$, where $\mathbf{G} \triangleq \mathcal{X}^2_{k,\alpha}$ is gotten from a chi-square distribution with kth degree of freedom and a significance level of α. α is typically set to accept 3σ of the normal distribution around the predicted state $\hat{\mathbf{x}}_{k+1|k}$.
- Update: the observation inside the validation gate and closest to the predicted state is then used to update the pedestrians state. Observations with no matching tracks are then assigned a new track e.g. $\mathbf{x}_{n,k+1}$ in figure 2.

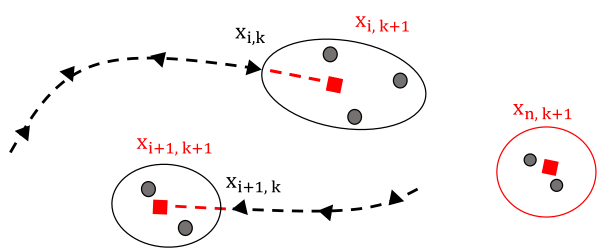


Fig. 2: Pedestrian tracking using proposed model

C. *Uncertainty-aware Proactive Collision Avoidance*

For a successful navigation to the goal, the robot is expected to avoid collision with the pedestrians (including obstacles) while moving to it's goal. Collision avoidance is achieved by adding a constraint to the NMPC problem, defined as:

$$\boldsymbol{\mathcal{C}}_{p_i} : r_{rob} + r_{p_i} + d^i \leq \delta^i_k, \qquad (6)$$
$$\forall k \in \{0, 1, ..., N\}, \forall i \in \{0, 1, ..., N_p\}$$

here r_{rob} and r_{p_i} are the radius of the robot and pedestrian

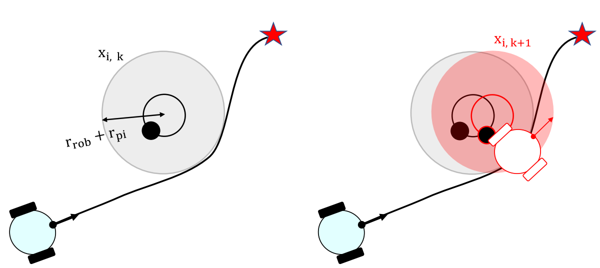


Fig. 3: Pedestrian Proxemics constraint is violated in the next time step due to hard constraint leading to the Freezing Robot Problem-*FRP*

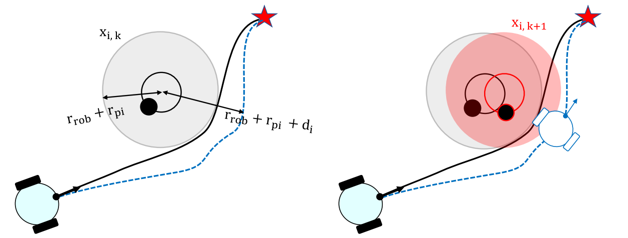


Fig. 4: Pedestrian Proxemics with probabilistic constraints prevents Freezing Robot Problem-*FRP*

i respectively, while $\delta^i_k = ||\mathbf{p}_{r,k} - \hat{\mathbf{p}}^i_{p,k}||_2$ is the euclidean distance between the position of the robot $\mathbf{p}_{r,k}$ and the pedestrian $\hat{\mathbf{p}}^i_{p,k}$, d^i is a small margin programmed by the user. The Non-linear MPC accounts for the radius as well as the estimated future states of the pedestrians, this look ahead behaviour leads to proactive collision avoidance. However, to minimize the objective cost function, the optimization algorithm usually returns a path that is near the collision region (see thick line in figure 3). Therefore, this hard constraint is easily violated in a real-world scenario due to the uncertainty in robot localization and the predicted states of the crowds, which leads to the freezing robot problem-FRP. To solve this problem, the proposed methodology is to model the uncertainty in the planner as a stochastic NMPC problem, with an additional cost function for each pedestrian and a collision probabilistic constraint that models the possibility of the robot colliding with any of the pedestrians over the prediction horizon, making the distance margin d^i a soft constraint:

$$\mathcal{J}_p(\boldsymbol{d}_i, \boldsymbol{\delta}_{prox}) = \sum_{k=1}^{N} ||d^i_k - \delta_{prox}||_2 \qquad (7)$$

$$\boldsymbol{\mathcal{C}}_{prob_i} : \mathbb{P}r_{COLL}(\mathbf{p}_{r,k}, \hat{\mathbf{p}}^i_{p,k}) \leq 1 - \boldsymbol{\epsilon}_P, \quad \boldsymbol{\epsilon}_P \in [0, 1] \qquad (8)$$

$$\mathbb{P}r_{COLL} \triangleq \frac{1}{1 + \boldsymbol{f}^p_{COLL}(d^i_k, \delta^i_k)} \qquad (9)$$

$$\boldsymbol{f}^p_{COLL}(d^i_k, \delta^i_k) = \Lambda_p e^{-(d^i_k - \delta^i_k)/\sigma_v} \qquad (10)$$

here δ_{prox} is a user defined proxemics distance and Λ_p is a tunable parameter. $\mathbb{P}r_{COLL}$ is the probability of collision between robot and pedestrian. $\boldsymbol{\epsilon}_P$ is the collision avoidance safety factor, for a high safety factor (e.g. $\boldsymbol{\epsilon}_A = 0.9$), a low collision probability is obtained. $\boldsymbol{f}^p_{COLL}(d^i_k, \delta^i_k)$ is a colision force derived from the social force model that models the interaction between the robot and the pedestrians. σ_v accounts for the uncertainty in the velocity of the moving pedestrians. This allows for a flexible path (see dashed line in figure 4) that accounts for the uncertainty in the pedestrians motion. The stochastic NMPC will therefore minimize the expected objective function $\mathcal{J}_p$ while satisfying the constraint with a high probability of $\boldsymbol{\epsilon}_A$.

D. *Boundary Constraints*

The boundary obstacles are both structured and unstructured nonhuman objects in the environment. This often consist of the walls in the scene and are hereby modelled as line segments, since they do not posses an exact geometry. For simplicity, boundary lines are defined by the two points at their extremities:

$$\mathcal{B}l_i \triangleq \{\mathbf{p}_a^i, \mathbf{p}_b^i\} = \{(x_a^i, y_a^i), (x_b^i, y_b^i)\}$$

To avoid collision with the boundary (walls and other large obstacles), the robot must ensure it maintains a safe distance between itself and the closest point on the boundary:

$$\boldsymbol{\mathcal{C}}_{p_i} : r_{rob} + d_{B,k}^i \leq \delta_{B,k}^i, \tag{11}$$
$$\forall k \in \{0, 1, ..., N\}, \forall i \in \{0, 1, ..., N_B\}$$

$d_{B,k}^i$ is a distance margin to the boundary, $\delta_{B,k}^i = ||\mathbf{p}_{r,k} - \hat{\mathbf{p}}_{B,k}^i(\lambda_B)||_2$ is the euclidean distance between the position of the robot $\mathbf{p}_{r,k}$ and the estimated position of the closest point on the boundary line $\hat{\mathbf{p}}_{B,k}^i(\lambda_B)$. It is computed from the equation of a line given two points, such that:

$$\hat{\mathbf{p}}_{B,k}^i(\lambda_B) \triangleq \mathbf{p}_a^i + \lambda_B(\mathbf{p}_b^i - \mathbf{p}_a^i) \tag{12}$$

with

$$\lambda_B = \frac{\overrightarrow{\mathbf{p}_a^i \mathbf{p}_{r,k}} \cdot \overrightarrow{\mathbf{p}_a^i \mathbf{p}_b^i}}{||\overrightarrow{\mathbf{p}_a^i \mathbf{p}_b^i}||_2^2}, \lambda_B \in [0, 1] \tag{13}$$

where λ_B is the normalised distance from the robot center to the closest point on the boundary line. Notice that the value λ_B is bounded to range $[0, 1]$ ensuring that only points on the line are considered. As in the case of pedestrian collision avoidance, boundary collisions and the freezing robot problem$-FRP$ that often arise due to the hard constraints are accounted for by adding a target objective function:

$$\mathcal{J}_B(\boldsymbol{d}_B^i, \boldsymbol{\delta}_{prox}) = \sum_{k=1}^{N} ||d_{B,k}^i - \delta_B^{prox}||_2 \tag{14}$$

and a boundary probability constraint defined as:

$$\boldsymbol{\mathcal{C}}_{\mathcal{B}l_i} : \mathbb{P}r_{COLL}(\mathbf{p}_k, \hat{\mathbf{p}}_{B,k}^i(\lambda_B)) \leq 1 - \boldsymbol{\epsilon}_B, \quad \boldsymbol{\epsilon}_B \in [0, 1] \tag{15}$$

$$\mathbb{P}r_{COLL}(\cdot, \cdot) \triangleq \frac{1}{1 + \boldsymbol{f}_{COLL}^B(d_{B,k}^i, \delta_{B,k}^i)} \tag{16}$$

$$\boldsymbol{f}_{COLL}^B(d_k^i, \delta_{prox}) = \Lambda_B e^{-(d_{B,k}^i - \delta_{B,k}^i)/\sigma_{vB}} \tag{17}$$

similar to the pedestrian collision avoidance, $\mathbb{P}r_{COLL}(\mathbf{p}_{r,k}, \hat{\mathbf{p}}_{B,k}^i(\lambda_B))$ is a measure of the probability of the robot colliding with the boundary, while $\boldsymbol{f}_{COLL}^B(d_k^i, \delta_{B,k}^i)$ is the collision force that models the proactive interaction between the robot and boundary.

E. Uncertainty-aware Dynamic Channel Convex Hull Constraint

Dynamic channel is a novel geometric-interval based global optimal and heuristically efficient path planner. Planning a path through the crowd $\mathcal{P}_\tau$ is achieved using the Delaunay Triangulation to generate the light gray graph in (figure 1) denoted as $\mathcal{T}_\tau$ with pedestrian positions p_i as its vertices. A new graph $\mathcal{T}_\tau^*$ is formed with nodes containing the union of the start, goal and triangle centroid position. Pedestrians that form the vertices of a gate between two triangles are all assumed to be cooperative until the robot is near the gate and the gate is still closed. This uncertain behaviour is defined by a cost function f_{coop}, hence the A^* Algorithm returns the path that minimizes:

$$\mathcal{T}_d^* = \underset{\boldsymbol{n} \in \mathcal{T}^*}{\arg\min} \sum ||g(\boldsymbol{n}) + h(\boldsymbol{n}) + f_{coop}(\boldsymbol{n})|| \tag{18}$$

where $g(\boldsymbol{n})$ is the cost to move from current node to the next node, $h(\boldsymbol{n})$ is the heuristic cost towards the goal, while the cooperative cost function $f_{coop}(\boldsymbol{n})$ is defined by a monotonically increasing function:

$$f_{coop} = \begin{cases} Ve^{(d_{rg}+d_g)} & d_{rg} < \delta_{prox} \ \& \ d_g < 2||r_{rob} + r_p|| \\ 0 & otherwise \end{cases} \tag{19}$$

where d_g is the distance between two pedestrians (nodes) that form a gate, d_{rg} is the distance between the robot and the centre of the gate, and V is a tunable parameter. Performing the A* algorithm on the graph $\mathcal{T}_\tau^*$ produces the set of triangles that form the shortest safe path through the crowd.

The optimization problem is to find the set of robot control inputs, whose predicted trajectory lies in the convex hull of the dynamic channel. This is achieved by considering the vertices of the Dynamic channel as a convex combination of points that form a convex set. That is for each point $\{Conv(DC) | c_i \in \mathbb{R}^2, i = 0, \cdots, n\}$ in the convex hull of the dynamic channel, with linear parameters $\{\boldsymbol{\theta} : \theta_0, \cdots, \theta_n\}$, the optimal trajectory of the robot must satisfy the condition:

$$Conv(DC)^T \cdot \boldsymbol{\theta}_{i,k} = \mathbf{p}_{r,k}^i \qquad \forall k \in \{0, 1, ..., N\}, \tag{20}$$
$$\theta_i \geq 0 \qquad \forall i \in \{0, 1, ..., n\}, \tag{21}$$
$$\sum \theta_i = 1 \qquad \forall i \in \{0, 1, ..., n\} \tag{22}$$

F. Receding Horizon Implementation

The stochastic NMPC planner computes the trajectory using numerical optimization methods. Therefore, only the first control input $\mathbf{u}_0^*$ is applied to the system after obtaining the optimal trajectory. This allows for a feedback loop to check and correct appropriately for the uncertainties in the motion executed by the robot. Hence, given:

$$\mathcal{J}(\mathbf{x}, \mathbf{x}_{ref}) = ||\mathbf{x}_k - \mathbf{x}_{ref}||_{Q_x}^2 \quad \text{state cost} \tag{23}$$
$$\mathcal{J}(\mathbf{u}, \mathbf{u}_{ref}) = ||\mathbf{u}_k - \mathbf{u}_{ref}||_{Q_u}^2 \quad \text{control cost} \tag{24}$$
$$\mathcal{J}(\mathbf{x}_N, \mathbf{x}_{ref}) = ||\mathbf{x}_N - \mathbf{x}_{ref}||_{Q_N}^2 \quad \text{terminal cost} \tag{25}$$
$$\mathcal{J}_{COLL}(\boldsymbol{d}, \boldsymbol{\delta}_{prox}) = ||d_k^i - \delta_{prox}||_2 \quad \text{collision cost} \tag{26}$$

where $\mathbf{x}$ and $\mathbf{x}_{ref}$ are the robot state and reference state with a positive design weight $Q_x \in \mathbb{R}_{\geq 0}^{n_x \times n_x}$. $\mathbf{u}$ and $\mathbf{u}_{ref}$ are the robot control input and reference control with positive design weight $Q_u \in \mathbb{R}_{\geq 0}^{n_u \times n_u}$. The terminal cost ensures the trajectory final pose $\mathbf{x}_N$ minimizes the distance to the reference (goal) $\mathbf{x}_{ref}$ with positive design weight $Q_N \in \mathbb{R}_{\geq 0}^{n_x \times n_x}$. While the collision cost ensures the proxemics distance is ensured. Therefore, the stochastic NMPC problem is to minimize in a receding horizon fashion the objective function:

$$\mathcal{J}^*(\mathbf{x},\mathbf{u}) = \min_{\mathbf{x}_{0:N},\mathbf{u}_{0:N-1}} \sum_{k=1}^{N} \mathcal{J}(\mathbf{x}_k,\mathbf{x}_{ref}) + \sum_{k=0}^{N-1} \mathcal{J}(\mathbf{u}_k,\mathbf{u}_{ref}) + \mathcal{J}(\mathbf{x}_N,\mathbf{x}_{ref}) + \sum_{k=1}^{N} \mathcal{J}_{COLL}(\boldsymbol{d},\boldsymbol{\delta}_{prox}) \quad (27)$$

$$\text{s.t } \mathbf{x}(k_0) = \mathbf{x}_0, \quad (28)$$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k,\mathbf{u}_k) \qquad \forall k \in \{0,1,...,N\}, \quad (29)$$

$$\mathbf{y}_k = h(\mathbf{x}_k,\mathbf{u}_k,p) \qquad \forall k \in \{0,1,...,N\}, \quad (30)$$

$$\mathbf{0} \geq \mathcal{C}(\mathbf{y}_k,\mathbf{u}_k,p) \qquad \forall k \in \{0,1,...,N\}, \quad (31)$$

$$1-\boldsymbol{\epsilon}_P \geq \mathbb{P}r_{COLL}(\mathbf{p}_{r,k},\hat{\mathbf{p}}^i_{p,k}) \qquad \forall i \in \{0,1,...,Np\} \quad (32)$$

here $\hat{\mathbf{x}}^i_{p,k}$ is the estimated state of the pedestrian, Eqn. 28 is a constraint that ensures the beginning of the trajectory $\mathbf{x}(k_0)$ is the same as the robot current pose $\mathbf{x}_0$. Eqn. 29 satisfy the system dynamics, while Eq. 30 - 32 are the equality constraints, inequality constraints and probabilistic constraints that must be satisfied for safe navigation as defined in the previous sections.

IV. RESULTS

This section describes the implementation and the performance of the methodology in various scenarios. The results were obtained from numerous simulations. Our method was implemented using python and the Robotic Operating System(ROS). The pedestrians were simulated using an Extension of the Social force model.

A. Probability of Collision Parameter Evaluation

The most important parameters in our method are the values of the probability of collision $\mathbb{P}r_{COLL}$ and the safety distance δ_{prox} . We check the effects of the parameter by setting the safety distance $\delta_{prox} = 0.5m$, while the various values of $\mathbb{P}r_{COLL} < [0.1, 0.3, 0.5, 0.7]$ were considered. Figure 5 shows the scenario as seen in the world frame, to ensure a safety distance above δ_{prox} the probability of collision $\mathbb{P}r_{COLL} < 0.3$.

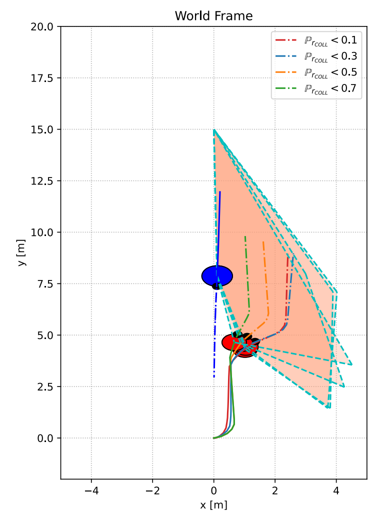


Fig. 5: The effects of various Probability of collision as seen in the World Frame with the robot (red) avoiding collision with a pedestrian(blue)

B. Performance in Test Scenarios

The proposed stochastic NMPC with dynamic channel (SNMPC-DC) method was evaluated using various metrics to measure its safety and efficiency. The safety distance measures the distance between the boundary of the robot's body and the pedestrians. The collision rate is the ratio of the number of collisions to the number of attempts (opposite of the success rate in [5]). The trajectory length and the time taken corresponds to the total displacement and time it took the robot to complete the task.

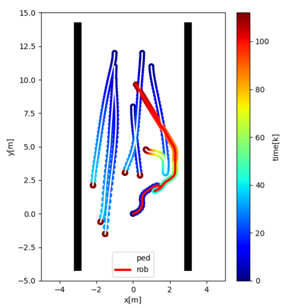
- **Corridor test Scenario**: In this test the robot is expected to move in an environment with two walls each on the left and right side while avoiding collision with different number of agents ($N_p \in [1, 2, 4, 6]$) Figure 6a. The results shows that our SMPC-DC is able to navigate successfully with a minimum safety distance of $0.22m$ as compared to the deterministic MPC that has a high collision rate as the density of the pedestrians increases (Table I). The cooperative constrain introduced by the Dynamic Channel force the robot to prioritize moving towards the left hand side of the pedestrian. Also, the probabilistic constraint in SNMPC-DC enables the robot to dynamically adapt to the small space.
- **Crossing test Scenario**: Here the robot is expected to move towards a goal that is some distance across an intersection of perpendicular paths Figure 6c. The collision rate of the deterministic MPC is better in this scenario as compared to the Corridor test scenario, this was achieved by waiting for the pedestrians to pass-by, before crossing the intersection. This is an automata behaviour that is arguably safe except that the robot would have remain in the freezing robot state if the flow of the crowd was continuous. However, our proposed methodology solve this problem by cooperatively moving in the crowd and following the dynamic channel. This in the long run lead to a collision free navigation with a minimum safety distance of $0.11m$ and an efficient displacement over time (Table I) as the pedestrians also actively cooperate with the robot.

V. CONCLUSIONS

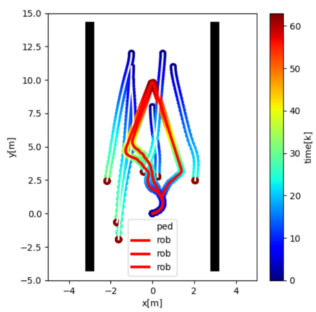
This paper presented our proposed approach for solving navigation problems in a crowd with uncertainty by formalizing the problem as a stochastic model predictive problem enclosed in a dynamic channel. We first proposed a model that reduces the estimation errors in the state of the pedestrians in the robot local frame by accounting for the motion of the robot. Then we proceed to show how we formalise the control problem as a stochastic NMPC to reduce the occurrence of the freezing robot problem$-FRP$ by adding a proxemics objective function, a probability

CORRIDOR	SNMPC-DC				DMPC			
No. Agents	$S_{min}(\mathbb{E})$	cr	L	T	$S_{min}(\mathbb{E})$	cr	L	T
1	0.32(1.04)	0%	10.20	26.50	-0.25(1.05)	30%	11.46	31.80
2	0.37(1.22)	0%	10.4	27.00	-0.16(1.00)	50%	11.15	32.35
4	0.37(1.15)	0%	10.97	29.00	-0.27(0.97)	60%	13.52	47.80
6	0.22(0.99)	0%	12.22	56.00	-0.21(0.98)	90%	11.52	38.50
CROSSING	SNMPC-DC				DMPC			
No. Agents	$S_{min}(\mathbb{E})$	cr	L	T	$S_{min}(\mathbb{E})$	cr	L	T
1	0.55(1.14)	0%	9.75	24.50	1.00(1.51)	0%	10.06	27.70
2	0.58(1.13)	0%	11.80	29.50	0.72(1.39)	0%	11.90	32.60
4	0.11(1.02)	0%	11.63	32.00	-0.04(1.21)	10%	13.06	38.60
6	0.11(1.04)	0%	11.75	33.00	-0.53(1.16)	60%	13.74	41.75

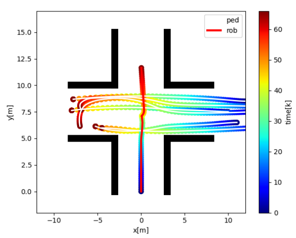
TABLE I: Corridor and Crossing Test Scenario Results, showing minimum safety distance and the mean ($S_{min}(\mathbb{E})$), collision rate (cr), trajectory length (L) and time taken to reach goal (T)



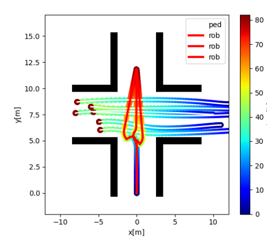
(a) Corridor SNMPC-DC: robot in red always move towards the left hand side of pedestrian



(b) Corridor DMPC: robot in red follow random shortest path leading to collision



(c) Crossing SNMPC-DC: finds a path through the crowd instead of waiting indefinitely for pedestrians to pass



(d) Crossing DMPC: the robot wait indefinitely for pedestrians to pass

Fig. 6: Time series plot of robot and pedestrian motion in various scenarios

constraint and a dynamic channel to make the robot follow a collision free path in the crowd. The implementation results shows that the method performs better than the deterministic MPC. Future works will consider extending the approach to unstructured environments.

ACKNOWLEDGMENT

This work is funded by MOBI-DEEP project (ANR-17-CE33-0011).

REFERENCES

[1] T. Fan, X. Cheng, J. Pan, D. Manocha, and R. Yang, "Crowdmove: Autonomous mapless navigation in crowded scenarios," *ArXiv*, vol. abs/1807.07870, 2018.

[2] A. Vemula, K. Muelling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1685–1692.

[3] A. Bera, T. Randhavane, R. Prinja, and D. Manocha, "Sociosense: Robot navigation amongst pedestrians with social and psychological constraints," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 7018–7025.

[4] B. Mutlu and J. Forlizzi, "Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*. ACM, 2008, pp. 287–294.

[5] M. Kabtoul, A. Spalanzani, and P. Martinet, "Proactive and smooth maneuvering for navigation around pedestrians," *Proceedings - IEEE International Conference on Robotics and Automation*, 05 2022.

[6] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: the case for cooperation," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 2153–2160.

[7] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 797–803.

[8] X. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 4, pp. 1743–1760, Oct 2017.

[9] D. Helbing, I. Farkas, and T. Vicsek, "Simulating dynamical features of escape panic," *Nature*, vol. 407, no. 6803, p. 487, 2000.

[10] G. Ferrer, A. G. Zulueta, F. H. Cotarelo, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous robots*, vol. 41, no. 4, pp. 775–793, 2017.

[11] F. Zanlungo, T. Ikeda, and T. Kanda, "Social force model with explicit collision prediction," *EPL (Europhysics Letters)*, vol. 93, no. 6, p. 68005, 2011.

[12] D. Ellis, E. Sommerlade, and I. Reid, "Modelling pedestrian trajectory patterns with gaussian processes," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, Sep. 2009, pp. 1229–1234.

[13] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[14] C. Cao, P. Trautman, and S. Iba, "Dynamic channel: A planning framework for crowd navigation," *ArXiv*, 02 2019.

[15] W. Jin, P. Salaris, and P. Martinet, "Proactive-cooperative navigation in human-like environment for autonomous robots," 2020, pp. 412–419.

[16] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, and S. Levine, "Uncertainty-aware reinforcement learning for collision avoidance," *arXiv preprint arXiv:1702.01182*, 2017.

[17] T. A. N. Heirung, J. Paulson, J. O'Leary, and A. Mesbah, "Stochastic model predictive control — how does it work?" *Computers & Chemical Engineering*, vol. 114, 11 2017.

[18] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.