



**HAL**  
open science

# Towards autonomous robot navigation in human populated environments using an Universal SFM and parametrized MPC

Enrico Fiasché, Philippe Martinet, Ezio Malis

## ► To cite this version:

Enrico Fiasché, Philippe Martinet, Ezio Malis. Towards autonomous robot navigation in human populated environments using an Universal SFM and parametrized MPC. IROS 2023 - IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct 2023, Detroit (MI), United States. hal-04210032

**HAL Id: hal-04210032**

<https://inria.hal.science/hal-04210032v1>

Submitted on 19 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Towards autonomous robot navigation in human populated environments using an Universal SFM and parametrized MPC

Enrico Fiasché<sup>1,2</sup>, Philippe Martinet<sup>2</sup> and Ezio Malis<sup>2</sup>

**Abstract**—Autonomous mobile robot navigation in a human populated and encumbered environment is recognized as a hard problem to be solved in real-time. Most of the time, robots face the so-called ‘Freezing Robot Problem’, that occurs when the robot stops because no feasible and safe motion can be found. In order to provide to the robot the capability of proactive navigation, in this work we generalize the classical Social Force Model into a Universal Social Force Model (USFM) that attributes to any object surrounding the robot (humans, robots, obstacles) a social behavior. Nonlinear Model Predictive Control (MPC) can be used to solve the autonomous navigation problem since it can take into account all the possible constraints coming from the interaction model between the robot and the different surrounding objects. However, to be effective, MPC requires a sufficiently large prediction horizon, which generally implies a high computational cost. In order to considerably reduce the computational cost, we propose a new control parametrisation based on *Thin Plate Spline Radial Basis Functions* that allow us to have a large prediction horizon with fewer parameters. The global control framework is validated in simulation with virtual pedestrians, and in real world environments.

## I. INTRODUCTION

The utilization of robots in human populated environment is becoming increasingly effective and significant. Some applications include floor cleaning (such as in supermarkets, subway stations, etc.), package transportation and delivery, and orientation guidance (such as in historical centers, museums, fairs, etc.). To achieve autonomous navigation in such environments, it is necessary to estimate the behavior of actors (including humans, moving obstacles, and other robots), compute real-time control while ensuring safety, and potentially perform proactive navigation (i.e., compute actions that are compatible and acceptable to humans while avoiding the Freezing Robot Problem [1]).

Understanding human behavior is likely the most complex issue that must be addressed. Firstly, understanding Human to Human interactions is essential. In [2], the author presents a categorization of the surrounding space into different types (Intimate, Personal, Social, Public) from which proxemics were defined for Human to Human interactions (H2H). In [3], the Social Force Model (SFM) is described. Pedestrian motion is modeled as a result of the application of a set of forces called “social forces”, which are applied to the pedestrian from different sources. Several versions of this model have been developed, including the Extended Social Force Model (ESFM) [4], the Headed Social Force Model

(HSFM) [5], and the Collision Prediction Extension of Social Force Model (CPESFM) [6]. All these extensions consider interactions with Humans (H2H), Obstacles (H2O), Robots (H2R), and, more generally, everything (H2X). Based on the H2X Social Force model, the SPACiSS simulator [7] was developed and recently used to simulate pedestrian behavior in the presence of an autonomous vehicle operating in a shared and open space in a proactive manner [8]. The accumulated knowledge of H2X interaction modeling can be beneficial to the modeling of Robot to everything (R2X) interaction. However, predicting human intentions in particular environments remains challenging. For example, in [9], the authors propose a Vehicle to Human (V2H) interaction model where the pedestrian’s cooperation with the vehicle is estimated online. In this work, we propose a further extension of the SFM by defining the Universal Social Force Model (USFM) that can be used by the robot controller for autonomous navigation.

Computing real-time control while ensuring safety over a fixed time horizon is a challenging task. One approach to address this challenge is Model Predictive Control (MPC), which involves computing optimal control while anticipating future events within a time horizon. However, state-of-the-art works often fail to mention the computational efficiency of MPC in real-time applications. A major drawback of using an MPC controller is that it requires considerable time and computational resources to perform an online optimization problem at each time step, which limits its application in real-world scenarios [10]. Control parametrization is an effective solution [11], which significantly reduces the number of control variables in the optimization problem while minimizing performance loss. Recent works have applied control parametrization on real-time simple systems employing linear parameterizations [12] and B-spline [13], and on real-time dynamic vehicles [10]. In this work, we propose a new control input parametrization that outperforms previous ones and allows for an efficient real-time implementation on a real robot in complex scenarios.

The remainder of this paper is organized as follows. Section II introduces the Universal Social Force Model used in the description of the interaction with the environment. Section III presents the proposed non-linear MPC control framework including the parametrization of the control Input. Section IV describes the evaluation methodology and the comparison results obtained in different simulated and real world scenarios. The last section recalls the main contributions of the work and presents the future perspectives.

<sup>1</sup>Université Côte d’Azur DS4H, Nice, France, [enrico.fiasche@inria.fr](mailto:enrico.fiasche@inria.fr)

<sup>2</sup>INRIA Centre at Université Côte d’Azur, Sophia Antipolis, France, [{philippe.martinet, ezio.malis}@inria.fr](mailto:{philippe.martinet, ezio.malis}@inria.fr)

## II. UNIVERSAL SOCIAL FORCE MODEL

The objective of this section is to introduce a suitable Universal Social Force Model for describing interactions between humans and their dynamic environment, denoted as H2X interactions. To enable the robot to behave in a human-like manner, the H2X interaction model can be extended for use in R2X interactions, thereby making the interaction more sociable. Various approaches have been proposed to model R2H and R2O interactions between robots and humans, and between robots and obstacles. For instance, the reactive navigation of a robot treating pedestrians as static obstacles is described in [14]. This is because in the Social Force Model (SFM) used, humans are not aware of the robot. In contrast, [15] proposes a solution based on a modified version of the SFM that takes into account the interaction force between pedestrians and the robot. This knowledge enables the robot to navigate through crowds using various algorithms. However, most of these works are developed for open environments, and the forces generated by obstacles and other robots are not well managed in indoor environments. To obtain a model capable of accurately describing the dynamic environment and pedestrian behavior in different scenarios, the human model proposed in [4] is considered. In particular the forces that influence the pedestrian  $i$  are:

$$\mathbf{f}_i(t) = \mathbf{f}_i^0(\mathbf{v}_i) + \sum_{j \neq i}^{N_{ped}} \mathbf{f}_{ji}(\mathbf{x}_i, \mathbf{x}_j) + \sum_o^{N_{obs}} \mathbf{f}_{oi}(\mathbf{x}_i) + \sum_a \mathbf{f}_{ai}(\mathbf{x}_i, \mathbf{x}_a) \quad (1)$$

where  $\mathbf{f}_i^0(\mathbf{v}_i)$  is the driving force that attracts the pedestrian towards the goal,  $\mathbf{f}_{ji}(\mathbf{x}_i, \mathbf{x}_j)$  is the repulsive interaction force with other pedestrian  $j$ ,  $\mathbf{f}_{oi}(\mathbf{x}_i)$  is the repulsive force generated by obstacles and  $\mathbf{f}_{ai}(\mathbf{x}_i, \mathbf{x}_a)$  is the attraction force generated by points of interest.

Most existing works define forces based on the center point of an object, making it unsuitable for long walls. Moreover, critical forces such as the Along Wall Force and the Social Slip Force are missing, which are essential for accurately simulating human behavior in crowded environments.

To address these limitations, the proposed Universal Social Force Model (USFM) considers the forces affecting a pedestrian from nearby obstacles based on the closest points rather than the center point of the nearest obstacle. This approach results in smoother and more realistic navigation. The force is inversely proportional to the distance between the pedestrian and the obstacle and is activated when the distance is less than a threshold  $d_o$ .

To prevent local minima and enable pedestrians to bypass obstacles, an Along Wall Force is added to the model, as illustrated in Figure 1. This force is computed as follows:

$$\mathbf{f}_o^{AW}(\mathbf{v}_i) = A_o \begin{cases} \mathbf{n}_o & \text{if } \mathbf{dir}_i \cdot \mathbf{n}_o \geq 0 \\ -\mathbf{n}_o & \text{else} \end{cases} \quad (2)$$

where  $\mathbf{n}_o$  is the unit directional vector of the wall,  $\mathbf{dir}_i$  is the direction of the pedestrian and  $A_o$  is the strength of the

force. This force is activated when the distance between the pedestrian and the wall is less than a threshold  $d_o^{AW}$ .

To encourage more collaborative behavior among pedestrians during frontal interactions, the Social Slip Force is introduced in the USFM. This force can be seen as a magnetic field that causes pedestrians to “slip” along each other, as illustrated in Fig.1. The strength of this force is also based on the distance between pedestrians and is calculated as follows:

$$\mathbf{f}_{ij}^S(\mathbf{x}_i, \mathbf{x}_j) = A_{S,ij} \mathbf{t}_{ji} \quad (3)$$

where  $\mathbf{t}_{ji}$  is the unit directional vector of the new force composed by  $[v_{iy}; v_{ix}]$ , which are the directional velocities of the pedestrian, and  $A_{S,ij}$  is the strength of the force. This force is activated when the interaction angle between the direction of the two pedestrians is  $\theta_{ij} \in [\alpha, \beta]$  and the distance between them is less than a threshold  $d_{ij}^S$ . This force also allows faster pedestrians to surpass slower ones, promoting a smoother movement.

In addition to the previous described forces, the proposed Universal Social Force Model (USFM) incorporates a pedestrian’s perception zone, which defines the region in which the pedestrian can perceive obstacles and other pedestrians. This zone is represented as an ellipse and the interaction forces are only computed if the obstacles or other pedestrians are within the pedestrian’s perception zone.

By utilizing the USFM, the robot can estimate the position and velocity of pedestrians in the environment. This information can be utilized by the Model Predictive Control (MPC) to predict future trajectories of the agents in the environment. The MPC can also manage R2O interactions between the robot and obstacles as constraints in the optimization problem. Since the robot is behaving like a pedestrian, generating socially acceptable trajectories, the USFM can manage R2R interactions between the robot and other robots as R2H interactions.

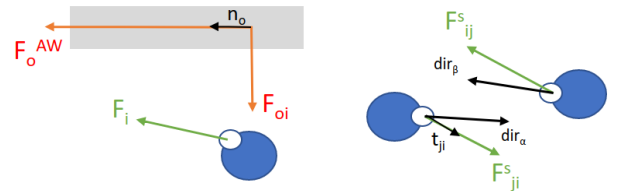


Fig. 1: Along Wall Force and Social Slip Force

The final forces that influence the pedestrian behavior in the USFM are:

$$\mathbf{f}_i(t) = \mathbf{f}_i^0(\mathbf{v}_i) + \sum_{j \neq i}^{N_{ped}} \left( \mathbf{f}_{ji}(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{f}_{ji}^S(\mathbf{x}_i, \mathbf{x}_j) \right) + \sum_a \mathbf{f}_{ai}(\mathbf{x}_i, \mathbf{x}_a) + \sum_o^{N_{obs}} \left( \mathbf{f}_{oi}(\mathbf{x}_i) + \mathbf{f}_o^{AW}(\mathbf{v}_i) \right) \quad (4)$$

## III. NONLINEAR MODEL PREDICTIVE CONTROL FRAMEWORK

In Model Predictive Control, several types of mathematical formulations can be used to find a solution to the optimization problem [12]. In order to achieve real-time feasibility,

direct methods are widely used. With this method the original continuous system is modeled as a discrete-time system; given the discretized state vector  $\mathbf{x}_k \in \mathbb{R}^n$  and the discretized control input  $\mathbf{u}_k \in \mathbb{R}^m$  at the  $k$ -th time step, a transition function  $\mathbf{f}_d$  returns the state vector that should be reached at the next time step.

$$\mathbf{x}_{k+1} = \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k) \quad (5)$$

Finally, the Euler's forward integration is used to approximate the state vector  $\mathbf{x}_{k+1}$  by a first-order Taylor expansion:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k) \quad (6)$$

where  $\Delta t$  is the time sampling step. The general formulation of the nonlinear control problem is usually in the form of a quadratic cost function. The Model Predictive Control optimization problem can be stated by the following:

$$\min_{\mathbf{u}, \mathbf{x}} \left( \sum_{k=0}^{N_p} \|\mathbf{x}_k - \mathbf{x}^*\|_Q^2 + \sum_{k=0}^{N_p-1} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_R^2 \right) \quad (7a)$$

$$\text{s.t.: } \mathbf{x}_{k+1} - \mathbf{f}_d(\mathbf{x}_k, \mathbf{u}_k) = 0 \quad k = 0, \dots, N_p - 1 \quad (7b)$$

$$\mathbf{x}_k \in \mathcal{X} \quad k = 1, \dots, N_p \quad (7c)$$

$$\mathbf{u}_k \in \mathcal{U} \quad k = 0, \dots, N_p - 1 \quad (7d)$$

where  $\mathbf{x} = [\mathbf{x}_0; \dots; \mathbf{x}_{N_p}]$  is the sequence of the states,  $\mathbf{x}_0$  is assumed known a priori,  $\mathbf{u} = [\mathbf{u}_0; \dots; \mathbf{u}_{N_p-1}]$  is the sequence of control inputs,  $\mathbf{x}^*$  is the desired state vector and  $N_p$  is a constant that replaces the prediction horizon  $T_p$  with the following relation  $T_p = \Delta t N_p$ . The constraints (7c) and (7d) can be added in order to enforce explicit constraints to the state or the control input. The constraint (7b) can be enforced in two different ways depending on the chosen decision variables [12].

#### A. Parameterization of the control input

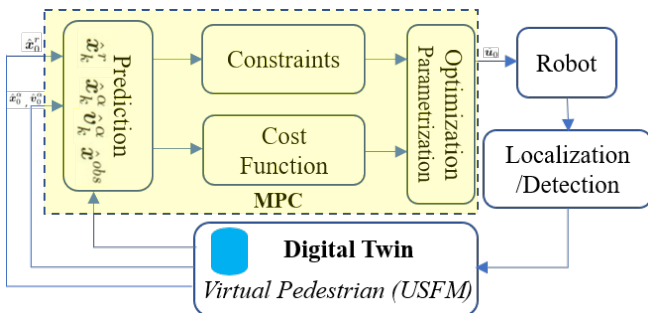


Fig. 2: NLMPC framework

The effectiveness of the Model Predictive Control (MPC) is influenced by the length of the prediction horizon. A longer horizon can consider more optimal trajectories, but the complexity of the problem should be kept manageable. To extend the prediction horizon, one can use a longer sampling time instead of increasing the number of decision variables. However, some systems may require a shorter sampling time for stability in closed [16]. Consequently, there is a trade-off between increasing the prediction horizon and keeping the number of decision variables manageable. Control input

parametrization can be a useful approach to addressing this trade-off [12]. It aims to minimize performance loss while reducing the number of control variables. With control input parametrization, it is possible to decouple the prediction horizon ( $T_p$ ) from the control horizon ( $T_c$ ) allowing the system to have a large prediction horizon. Control input parametrization generates a sequence of control inputs from a set of parameters, which can be either pre-selected constants or tuned online during the optimization process. The approach uses a set of free control inputs as decision variables to define the entire control sequence.

$$\mathbf{u} = \boldsymbol{\pi}(\boldsymbol{\eta}) \quad \boldsymbol{\pi} : \mathbb{R}^{N_\eta} \rightarrow \mathbb{R}^{N_p} \quad (8)$$

Thanks to this method it is possible to reduce the number of optimization variables and of constraints, making the problem solvable even with limited computational resources. Different works at the state-of-the-art have already highlighted the advantages of using parametrizations in Model Predictive Control. We consider two distinct parameterization techniques, namely linear parameterization [12] and cubic B-spline parameterization [13], to approximate control inputs in our system. In [12] various linear parameterizations were investigated. The linear parameterization establishes a direct linear relationship between the parameters  $\boldsymbol{\eta}$  and the corresponding control inputs  $\mathbf{u}$ , which is expressed as follows:

$$\mathbf{u} = \mathbf{\Pi} \boldsymbol{\eta} \quad (9)$$

where  $\boldsymbol{\eta} \in \mathbb{R}^{N_\eta}$  are the free control inputs and  $\mathbf{\Pi} \in \mathbb{R}^{m \times N_\eta}$  is the combination matrix. On the other hand [13] explored the implementation and characteristics of the cubic B-spline parameterization, which is expressed as follows:

$$\mathbf{u}_i = \sum_{i=-1}^{N+1} \boldsymbol{\eta}_i \mathbf{B}_{i,3}(t) \quad (10)$$

where  $\mathbf{B}_{i,3}(t)$  are the cubic B-spline basis functions. Reducing the control horizon  $N_c$  to be smaller than the prediction horizon  $N_p$  is a common strategy in control system. The first  $N_c$  control inputs are freely allocated and optimized online, while the remaining  $N_p - N_c$  control inputs are predetermined, often holding the same value as  $\mathbf{u}_{N_c-1}$  [17]. This approach reduces the optimization problem's complexity but may lead to instability due to prolonged periods of constant control input. To address stability concerns, alternative strategies known as Move Blocking have been proposed. One such strategy is the Zero Order Hold (ZOH) method, where the control input is held constant between consecutive control points [12]. Another technique is the linear interpolation, called the LERP method, which spreads the "degree of freedom" uniformly along the prediction horizon [12]. Both Move Blocking methods decrease online computation time by reducing complexity, thus promoting system convergence and maintaining solution accuracy and control performance. However, an even more flexible approach that provides smoother control trajectories is the cubic B-spline parameterization [13]. The cubic B-spline utilizes

higher-order polynomial interpolation, allowing for continuous and smooth control input changes between points. We propose a novel control parametrization, which is smoother and more compact compared to the LERP parameterization. Additionally, it operates in a single dimension, making it less complex than the B-spline approach. The objective of this parametrization is to reduce the computational time required by the optimization to solve the problem. The Radial Basis Function (RBF) is an excellent technique for interpolating data and functions. A Radial Basis Function depends on the Euclidean distance between given points  $\mathbf{x}$ . Approximation with RBF is grid-free and easy to implement [18]. Considering a set of points  $x_1, x_2, \dots, x_N$ , the Radial Basis Function centered at  $x_j$  has the following form:

$$\varphi_j = \varphi(\|\mathbf{x} - x_j\|) \in \mathbb{R}^n, \quad j = 1, \dots, N \quad (11)$$

where  $\|\mathbf{x} - x_j\|$  is the Euclidean norm.

The Radial Basis Function approximation of the control input  $\mathbf{u}$  is defined as:

$$\mathbf{u}(t) = \sum_{j=0}^{N_b} \omega_j \varphi(\|\mathbf{t} - c_j\|) \quad (12)$$

where  $N_b$  is the number of free control inputs,  $\omega_j$  is the weight of the thin plate spline function computed to satisfy some boundary conditions,  $c$  is the set of corresponding time instants of the free control inputs and  $\mathbf{t}$  is the time sampled from 0 to the prediction horizon  $N_P$ . The thin plate spline function  $\varphi(r)$  is defined as:

$$\varphi(r) = r^2 \log(r) \quad (13)$$

One of the main advantages of Thin Plate Spline RBF is the flexibility of the space, making the implementation of this method much easier than other functions [19]. This type of parametrization has been already used in the literature to parametrize surfaces [20], and has been shown to provide stable results even in presence of noisy information [21].

We propose to use parametrized MPC in a more complex constrained environment than the one proposed in [12] and [13]. The problem is formalized using the single-shooting method, where the Optimal Control Problem (OCP) is rewritten into a smaller optimization problem since only the control inputs are considered as decision variables. Thanks to the USFM, the MPC is able to extract the current positions and velocities of the pedestrians, and it is able to predict their future trajectories ( $x_k^p$  and  $y_k^p$ ). The OCP of the parametrized MPC used to navigate in human populated environment is the following:

$$\min_{\mathbf{u}} \left( \sum_{k=0}^{N_p} \|\mathbf{x}_k - \mathbf{x}^*\|_Q^2 + \sum_{k=0}^{N_p-1} \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_R^2 \right) \quad (14a)$$

$$\text{s.t.: } v_{min} \leq v_k \leq v_{max} \quad (14b)$$

$$\omega_{min} \leq \omega_k \leq \omega_{max} \quad (14c)$$

$$\sqrt{(x_k - x_{obs})^2 + (y_k - y_{obs})^2} \geq r_{robot} + r_{obs}; \quad (14d)$$

$$k = 1, \dots, N_p + 1$$

$$\sqrt{(x_k - x_k^p)^2 + (y_k - y_k^p)^2} \geq r_{robot} + r_{ped}; \quad (14e)$$

$$k = 1, \dots, N_p + 1$$

The control input used in the OCP is approximated with the control parametrizations previously presented.

## IV. EXPERIMENTS

In this section, we focus on assessing the effectiveness of the proposed parametrization and validating the global non-linear model predictive control (MPC) framework outlined in section III. To simulate the kinematic evolution of the mobile platform, which is defined as  $f_d(\cdot)$ , we use a non-holonomic model (a classical  $[2, 0]$  robot). We conducted experiments in two main environments at INRIA, using both a virtual representation and the real site in each case. The first environment, shown in Figure 3 (left), is a workshop with offices, corridors, and open area for robot experimentation. The second environment, shown in Figure 3 (right), is a conference hall. We used virtual pedestrians modeled with the USFM model and defined waypoints to create different frontal, lateral, and back interactions with the mobile robot.

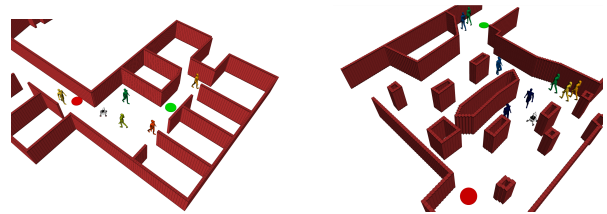


Fig. 3: Workshop and Conference Hall scenarios

### A. Evaluation Methodology

Defining the evaluation of the concept of efficiency, natural and social behavior in shared spaces poses a challenge as there are no fixed rules that limit the possible use cases. Prior studies have assessed autonomous navigation based on three primary metrics: collision rate, success rate, and time to goal [22]. However, recent works propose to include time efficiency and boundary violation as main metrics [23]. It is also possible to evaluate the motion safety of the robot in proximity to pedestrians and the quality of autonomous vehicle motion by considering the traveled distance, selected trajectory, and traveled time, as suggested by [24]. Navigation can be evaluated using two main categories: motion safety and trajectory quality.

*a) Motion Safety:* Motion safety in proximity to pedestrians is evaluated by analyzing the collision rate ( $CR$ ), which measures the collisions occurring between the robot and pedestrians during navigation. A collision occurs when there is an overlap between the footprints of the robot and the pedestrian. A near-zero collision rate is required to validate autonomous navigation.

*b) Trajectory Quality:* Trajectory quality is another critical factor used to evaluate autonomous navigation and can be divided into two categories: energy and efficiency. Energy can be defined using two different metrics: Path Energy ( $E_P$ ) and Dynamic Energy ( $E_T$ ). Path Energy is based on the robot's maneuvers along the trajectory and measures the smoothness of the path. Energy increases when there are several maneuvers in the path, while a zero energy ( $E_P = 0$ ) occurs when the path is straight. Dynamic Energy

estimates trajectory energy based on speed, measuring the average instantaneous variation of the speed from the preferred driving speed. Energy increases when instantaneous velocity variation occurs. Efficiency can generally be expressed in terms of time or distance. Traveled distance ( $C_L$ ) from the starting pose of the robot to the goal pose is an important metric for Trajectory Quality. Another important factor in efficiency is the Relative Time To Goal ( $TT_G$ ), which evaluates the traveled time to reach the goal at the average time.

### B. Control input parametrization comparison

The OCP utilized in this study is the one presented in equation (14). The optimization solver employed is CasADi [25], a symbolic framework for algorithmic differentiation and non-linear optimization that is available as an open-source software package. To solve the optimization problem, CasADi will utilize the Interior Point Optimizer (IPOPT), an open-source software package designed for large-scale non-linear optimization. To evaluate the performance of the parametrized Model Predictive Controller (MPC), it was tested in a constrained environment that included both static and dynamic obstacles. The simulation and experimental parameters used in the study are described in Table I. In particular, the pedestrian values were tuned to produce smooth trajectories and interactions in a cluttered environment.

| Parameter     | Value                       | Parameter      | Value                  |
|---------------|-----------------------------|----------------|------------------------|
| $\theta_{ij}$ | $\in [-40^\circ, 40^\circ]$ | $v_{min}$      | 0.0 m/s                |
| $d_o$         | 0.7 m                       | $v_{max}$      | 0.3 m/s                |
| $d_o^{AW}$    | 0.4 m                       | $\omega_{min}$ | $-\frac{\pi}{8}$ rad/s |
| $d_{ij}^S$    | 1.5 m                       | $\omega_{max}$ | $\frac{\pi}{8}$ rad/s  |
| $r_{ped}$     | 0.4 m                       | $r_{robot}$    | 0.4 m                  |

TABLE I: Parameters used in the simulations

The dynamic agents in the simulation are modeled using the presented USFM model, and they follow specific trajectories to increase the complexity of the simulation. Three different types of control input approximations are considered, while the number of free control inputs  $N_\eta$  is fixed at 10, and the number of prediction points  $N_p$  and sampling time  $\Delta t$  are varied. During the simulation process, we implemented and tested the cubic B-spline parameterization. However, it was not able to accurately approximate the control input function, rendering the robot unable to navigate among pedestrians in complex environments. It proved inadequate in accurately modeling the control input function due to the limited number of free control points, especially when  $N_p$  exceeded the number of free control points  $N_\eta$ . Table II and Table III show the simulation results obtained with  $\Delta t$  fixed at 0.1 s and 0.2 s, respectively. The Thin Plate Spline RBF always provides a faster and smoother solution with lower energy consumption than the LERP approximation, as indicated by the Time metric, which represents the average time required by the MPC to solve the OCP, and the other performance metrics used. Moreover, when  $N_p \gg N_\eta$ , the LERP approximation leads to high values of Path Energy  $E_P$  and Dynamic Energy  $E_T$ , indicating that the robot's navigation is not smooth and it must stop when the environment becomes unsafe with no

feasible solutions, leading to the Freezing Robot Problem. In contrast, the Thin Plate Spline RBF consistently produces satisfactory results even when  $N_p$  is ten times greater than  $N_\eta$ . The time required to solve the OCP is always small enough to allow the robot to update the position of the dynamic obstacles in the environment. The Thin Plate Spline RBF outperforms the current parameterizations found in state-of-the-art in the simulations of autonomous navigation in human-populated environments. Note that, in real-world experiments, it was not feasible to compare the two parameterizations as navigation with the LERP approximation was computationally impractical due to the high complexity of the environment.

| Dynamic environment $\Delta t = 0.1$ s  |              |              |                |                |              |          |
|---|--------------|--------------|----------------|----------------|--------------|----------|
| Linear Interpolation                    |              |              |                |                |              |          |
| $N_p$                                   | $E_P$        | $E_T$        | $C_L$          | $TT_G$         | Time         | CR       |
| 50                                      | <b>2.3</b>   | 0.576        | 161.891        | 221.972        | 0.240        | 0        |
| 60                                      | <b>1.908</b> | 0.563        | <b>52.736</b>  | <b>101.522</b> | 0.328        | 0        |
| 70                                      | 46.009       | 1.524        | 136.483        | 188.666        | 0.491        | 0        |
| 80                                      | 49.256       | 0.557        | 138.152        | 299.234        | 0.573        | 0        |
| 90                                      | 68.478       | 0.355        | 150.358        | 811.318        | 0.602        | 0        |
| 100                                     | 112.884      | 0.517        | 168.762        | 983.407        | 1.215        | 2        |
| Thin Plate Spline Radial Basis Function |              |              |                |                |              |          |
| $N_p$                                   | $E_P$        | $E_T$        | $C_L$          | $TT_G$         | Time         | CR       |
| 50                                      | 3.722        | <b>0.167</b> | <b>93.103</b>  | <b>90.791</b>  | <b>0.121</b> | 0        |
| 60                                      | 5.346        | <b>0.164</b> | 89.416         | 108.407        | <b>0.147</b> | 0        |
| 70                                      | <b>4.191</b> | <b>0.161</b> | <b>110.211</b> | <b>154.335</b> | <b>0.173</b> | 0        |
| 80                                      | <b>6.872</b> | <b>0.158</b> | <b>87.615</b>  | <b>150.962</b> | <b>0.207</b> | 0        |
| 90                                      | <b>5.122</b> | <b>0.129</b> | <b>83.188</b>  | <b>160.230</b> | <b>0.242</b> | 0        |
| 100                                     | <b>8.802</b> | <b>0.131</b> | <b>84.131</b>  | <b>207.345</b> | <b>0.313</b> | <b>0</b> |

TABLE II: Comparison between LERP and Thin Plate Spline RBF in a dynamic environment with  $\Delta t = 0.1$  s

| Dynamic environment $\Delta t = 0.2$ s  |               |              |                |                |              |          |
|---|---------------|--------------|----------------|----------------|--------------|----------|
| Linear Interpolation                    |               |              |                |                |              |          |
| $N_p$                                   | $E_P$         | $E_T$        | $C_L$          | $TT_G$         | Time         | CR       |
| 50                                      | <b>6.297</b>  | <b>0.034</b> | 186.855        | <b>253.237</b> | 0.325        | 0        |
| 60                                      | <b>15.99</b>  | <b>0.033</b> | <b>167.842</b> | <b>224.391</b> | 0.422        | 0        |
| 70                                      | 59.98         | 0.038        | 251.091        | 424.99         | 0.500        | 0        |
| 80                                      | 48.43         | 0.040        | 246.096        | 433.147        | 1.139        | 1        |
| 90                                      | 319.47        | 0.044        | 170.920        | 675.694        | 0.893        | 1        |
| 100                                     | 320.778       | 0.051        | 221.656        | 624.205        | 1.382        | 2        |
| Thin Plate Spline Radial Basis Function |               |              |                |                |              |          |
| $N_p$                                   | $E_P$         | $E_T$        | $C_L$          | $TT_G$         | Time         | CR       |
| 50                                      | 8.153         | <b>0.034</b> | <b>168.786</b> | 274.627        | <b>0.148</b> | 0        |
| 60                                      | 29.064        | 0.038        | 240.068        | 433.988        | <b>0.193</b> | 0        |
| 70                                      | <b>25.631</b> | <b>0.037</b> | <b>176.553</b> | <b>353.722</b> | <b>0.244</b> | 0        |
| 80                                      | <b>24.881</b> | <b>0.035</b> | <b>175.061</b> | <b>430.188</b> | <b>0.340</b> | 0        |
| 90                                      | <b>27.543</b> | <b>0.036</b> | <b>166.551</b> | <b>479.076</b> | <b>0.276</b> | <b>0</b> |
| 100                                     | <b>32.015</b> | <b>0.037</b> | <b>163.701</b> | <b>519.293</b> | <b>0.406</b> | <b>0</b> |

TABLE III: Comparison between LERP and Thin Plate Spline RBF in a dynamic environment with  $\Delta t = 0.2$  s

### C. Real world experiments

Real experiments were conducted in both environments using virtual pedestrians simulated on ROS in their respective digital twins. During the experiments, the SCOUT MINI AgileX robot, controlled as a classical  $[2, 0]$  robot with non-holonomic kinematic constraint, utilized ROS to interface with the scenarios and detect virtual pedestrians as they entered its perception view. By employing the Thin Plate Spline RBF approximation, the OCP's dimension was significantly reduced, resulting in the model's real-time feasibility in the

dynamic environment. In these experiments, the prediction horizon time steps were  $N_p = 24$ , with  $N_b = 9$  free control variables. The robot model's sampling time was  $\Delta t = 0.4$  s, resulting in a prediction horizon of  $T_p = 9.6$  s. Several virtual pedestrians were considered in both environments, with non-collaborative agents included in the second environment, standing in front of vending machines and sofas. Twenty simulations per environment were carried out to extract critical parameters for evaluating autonomous navigation. One simulation was conducted without pedestrians to understand the parameter values.

The proposed Model Predictive Control was successful in avoiding the Freezing Robot Problem by collaborating with other pedestrians. As shown in Table IV and Table V, the Collision Rate was always 0. During the optimal solution without pedestrians, the robot followed a straight line since no interactions occurred during the navigation, resulting in a low Path Energy value. Due to the presence of several agents in the environment, the robot had to turn several times to enable a safe and secure navigation, resulting in a higher Path Energy value. The mean value of the Dynamic Energy  $E_T$  approached zero, indicating that the model actively collaborated with other pedestrians without decreasing its velocity. Despite several interactions, the robot was always near the optimal path during the experiments, as evidenced by the Relative Traveled Distance being close to the optimal value obtained in the empty environment. Regarding the Relative Time to Goal, the robot took longer to reach the final goal during the experiments, as it had to generate socially acceptable trajectories to ensure secure and safe navigation.

| Experiments |       |          |         |           |
|-------------|-------|----------|---------|-----------|
|             | Empty | Mean     | Max     | Std. dev. |
| $CR$        | 0     | 0        | 0       | 0         |
| $E_P$       | 3.629 | 26.422   | 30,718  | 1.665     |
| $E_T$       | 0     | 0.000845 | 0.00428 | 0.00116   |
| $C_L$       | 1.347 | 1.368    | 1.393   | 0.0109    |
| $TT_G$      | 0.887 | 3.109    | 3.542   | 0.202     |

TABLE IV: Experiments in workshop scenario

| Experiments |        |         |         |           |
|-------------|--------|---------|---------|-----------|
|             | Empty  | Mean    | Max     | Std. dev. |
| $CR$        | 0      | 0       | 0       | 0         |
| $E_P$       | 28.210 | 184.339 | 189.488 | 2.352     |
| $E_T$       | 0.0013 | 0.040   | 0.052   | 0.0053    |
| $C_L$       | 1.472  | 1.485   | 1.492   | 0.0038    |
| $TT_G$      | 0.243  | 2.600   | 2.843   | 0.101     |

TABLE V: Experiments in conference hall scenario

## V. CONCLUSIONS

This work discusses parameterized Model Predictive Control (MPC) for autonomous navigation in a human-populated environment, specifically addressing the Freezing Robot Problem. The Universal Social Force Model (USFM) was introduced to describe pedestrian behavior, and the MPC was optimized for real-time feasibility and quick convergence during autonomous navigation. Comparisons were made between different parameterizations in simulations and real-world experiments, demonstrating the superiority of the proposed approach in constrained environments. The

robot successfully avoided the Freezing Robot Problem and interacted safely with pedestrians in indoor settings. We plan to continue our work, improving the MPC by also optimizing the trajectories provided by the USFM with an optimization problem to enable proactive navigation.

## ACKNOWLEDGMENT

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

## REFERENCES

- [1] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *IEEE/RSJ IROS*, 2010.
- [2] E. Hall, *The Hidden Dimension*. Garden City N.Y: Doubleday, 1966.
- [3] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *PRE*, vol. 51, no. 5, 1995.
- [4] G. Ferrer, A. Zulueta, F. Cotarel, and A. Sanfeliu, "Robot social-aware navigation framework to accompany people walking side-by-side," *Autonomous Robots*, vol. 41, no. 4, 2017.
- [5] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Praticchizzo, "Walking ahead: The headed social force model," *PLOS ONE*, vol. 12, no. 1, 2017.
- [6] F. Zanlungo, T. Ikeda, and T. Kanda, "Social force model with explicit collision prediction," *Europhysics Letters*, vol. 93, no. 6, 2011.
- [7] M. Prédhumeau, "Simulating Realistic Pedestrian Behaviors in the Context of Autonomous Vehicles in Shared Spaces." IFAAMAS, 2021.
- [8] M. Kabtoul, A. Spalanzani, and P. Martinet, "Proactive and smooth maneuvering for navigation around pedestrians," in *IEEE ICRA*, 2022.
- [9] —, "Towards proactive navigation: A pedestrian-vehicle cooperation based behavioral model," in *IEEE ICRA*, 2020.
- [10] Z. R. M. Júnior, A. M. De Almeida, and R. V. Lopes, "Vehicle stability upper-level-controller based on parameterized model predictive control," *IEEE Access*, vol. 10, 2022.
- [11] M. Muehlebach and R. D'Andrea, "Parametrized infinite-horizon model predictive control for linear time-invariant systems with input and state constraints," in *ACC*, 2016.
- [12] F. Fusco, G. Allibert, O. Kermorgant, and P. Martinet, "Benchmarking nonlinear model predictive control with input parameterizations," in *MMAR22*, 2022.
- [13] B. Rohal'-Ilkiv, M. Gulan, and P. Minarčík, "Implementation of continuous-time mpc using b-spline functions," in *International Conference PC19*, 2019.
- [14] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli, "Reactive planning for assistive robots," *IEEE RA-L*, vol. 3, no. 2, 2018.
- [15] J. Miura and Y. Shirai, "Parallel scheduling of planning and action of a mobile robot based on planning-action consistency," *JSAI*, vol. 15, no. 6, 2000.
- [16] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *BMSC*, vol. 11, 2002.
- [17] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE T-RO*, vol. 26, no. 5, 2010.
- [18] J. Yoon, "Spectral approximation orders of radial basis function interpolation on the sobolev space," *J. SIAM*, vol. 33, no. 4, 2001.
- [19] K. Liew, X. Chen, and J. Reddy, "Mesh-free radial basis function method for buckling analysis of non-uniformly loaded arbitrarily shaped shear deformable plates," *CMAME*, vol. 193, 2004.
- [20] E. Malis, "An efficient unified approach to direct visual tracking of rigid and deformable surfaces," in *IEEE/RSJ IROS*, 2007.
- [21] G. Silveira and E. Malis, "Real-time visual tracking under arbitrary illumination changes," in *IEEE CVPR*, 2007.
- [22] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *IEEE ICRA*, 2019.
- [23] Y. Kobayashi *et al.*, "Robot navigation based on predicting of human interaction and its reproducible evaluation in a densely crowded environment," *Int. J. Soc. Robot.*, 2022.
- [24] M. Kabtoul, "Proactive and socially-aware navigation for autonomous vehicles in spaces shared with pedestrians," Ph.D. dissertation, Université Grenoble Alpes, 2018-2021.
- [25] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *MPC*, vol. 11, no. 1, 2019.