

Towards Using Fast Embedded Model Predictive Control for Human-Aware Predictive Robot Navigation

Till Hielscher¹, Lukas Heuer², Frederik Wulle³, Luigi Palmieri²

Abstract—Predictive planning is a key capability for robots to efficiently and safely navigate populated environments. Particularly in densely crowded scenes, with uncertain human motion predictions, predictive path planning, and control can become expensive to compute in real time due to the curse of dimensionality. With the goal of achieving pro-active and legible robot motion in shared environments, in this paper we present HuMAN-MPC, a computationally efficient algorithm for Human Motion Aware Navigation using fast embedded Model Predictive Control. The approach consists of a novel model predictive control (MPC) formulation that leverages a fast state-of-the-art optimization backend based on a sequential quadratic programming real-time iteration scheme while also providing feasibility monitoring. Our experiments, in simulation and on a fully integrated ROS-based platform, show that the approach achieves great scalability with fast computation times without penalizing path quality and efficiency of the resulting avoidance behavior.

I. INTRODUCTION

The usage of autonomous mobile robots (AMRs) is continuously increasing in various types of environments for accomplishing several tasks, e.g. pick and delivery in logistic centers, and cleaning of cluttered surfaces. In these environments, robots share spaces with humans. To accomplish this the robot needs to infer how surrounding humans will move in the near future and utilize those predictions in its navigation framework [1]. Moreover, to respond to dynamic changes in surrounding environments, robots need to compute actions in real time. Model predictive control (MPC) is an established method to compute robot trajectories in such settings [2], [3]. However, those techniques suffer from the curse of dimensionality in such environments, i.e., planning considering human motion predictions is quite expensive [2], [3], [4], [5].

With the goal of handling the introduced complexity and achieving fast and reliable robot motion generation in crowded environments, we present HuMAN-MPC – a system for Human Motion Aware Navigation using fast embedded Model Predictive Control and make the following contributions:

- We propose an MPC formulation that includes a novel cost definition to achieve collision avoidance of nearby

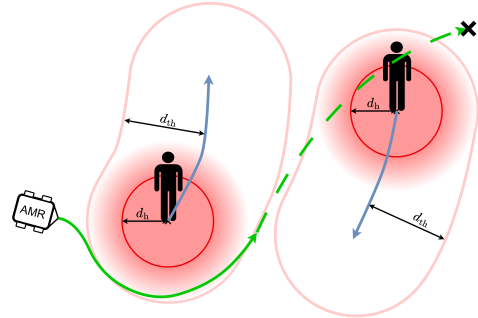


Fig. 1: Human awareness considering areas for cost (d_{th}) and safety constraint (d_h) with the human predictions (blue) and the robots path (green).

humans by considering their short-term predictions. The cost formulation implies a positive semi-definite Hessian that allows us to use recent fast embedded optimization tools, *acados* [6], for achieving quasi-real-time motion generation (i.e., a few milliseconds for solving an MPC cycle).

- To further enhance its reliability, we interface the motion generation to feasibility monitors developed by considering established safety norms for mobile robotic applications.

II. RELATED WORK

First approaches to collision avoidance include model-based techniques not considering predictions of humans, e.g. Dynamic Window Approach (DWA) [7], Elastic Band [8] or Optimal Reciprocal Collision Avoidance (ORCA) [9].

Lately, there has been an increasing interest in MPC techniques for generating robot motion. Differently from [7], [8], [9] they allow researchers to adopt more easy-to-use and customizable tools for numerical optimization. In this work, building on our previous results [1], [2], [4], [3], we shift our attention towards the usage of fast embedded solvers for generating robot motion considering human motion predictions. The idea of using a cost term in the MPC formulation to drive the trajectory away from dynamic obstacles has been explored in the past [5], [10], [11]. In particular, several approaches have considered human motion predictions for generating robot motion [5], [3], [11], [12], [13]. Schmerling et al. [11] and Schaefer et al. [5] use multi-modal prediction in their MPC formulations, that include interactions between the agents in the scene. Heuer et al. [3] also use multi-modal predictions by directly considering the spatial relation between the predicted modes of motion. Our approach also uses predictions, differently,

¹ T. Hielscher is with the Socially Intelligent Robotics Lab, Institute for Artificial Intelligence, University of Stuttgart, Germany, till.hielscher@ki.uni-stuttgart.de

² L. Heuer, L. Palmieri are with Bosch Corporate Research, Stuttgart, Germany, lukas.heuer@de.bosch.com, luigi.palmieri@de.bosch.com

³ F. Wulle is with ARENA2036 e.V., a research campus at the University of Stuttgart, Germany, frederik.wulle@arena2036.de.

This work was partly supported by the EU Horizon 2020 research and innovation program under grant agreement No. 101017274 (DARKO).

we adopt a formulation that is convex for all relevant values and thereby suited for fast embedded optimization using sequential quadratic programming (SQP) techniques [14].

Other approaches adopt data-driven technologies (i.e., deep reinforcement learning (DRL)) for the generation of robot motion among people: GA3C-CADRL [15] or SA-CADRL [16]. While being quite efficient in inferring robot actions, they often use a limited amount of predefined discrete actions due to complex training, and thus do not completely account for the dynamics of the robot and the environment leading to difficulties in ensuring safe behavior.

III. BACKGROUND

A. Safety Norms

Existing safety standards must be taken into account when deploying novel planning and control algorithms to real robotic applications. Relevant in this regard is DIN EN ISO 3691-4 [17] for driverless industrial trucks and DIN EN ISO 13482:2014-11 [18] for personal care robots. The norms state that navigation controllers have to be designed to provide collision avoidance and risk reduction functionality. To this end, the controller has to constrain operational scenarios and the workspace around surrounding humans. If there is a safety-relevant object, i.e., a human, in close proximity a protective stop is a sufficient action.

Remark 1: In this work, the term “safety” is regarded as compliant with the relevant safety norms [17] and [18].

B. Human Motion Prediction

As discussed in our previous work [1], the literature does not provide a consistent definition of motion predictions.

Remark 2: In this work, we assume the availability of all $i \in [1, N_h]$ humans positions $\mathbf{h}_{i,n}$ at time instant n up to the prediction horizon N , i.e., predictions are reported as discrete trajectories $\mathbf{T}_{i,1:N} = \{\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \dots, \mathbf{h}_{i,N}\}$.

C. Fast Embedded Optimization with *acados*

We adopt *acados* [6], a framework (i.e., a collection of solvers) for fast embedded optimal control (i.e., solving optimization problems with limited computational resources). Based on BLASFEO [19], which is a library specifically designed for high-performance linear algebra techniques for embedded optimization, it offers several computationally efficient algorithms and components written in the C programming language.

Remark 3: In our approach, we will use the SQP Real-Time Iteration (RTI) scheme [14] and the HPIPM solver with partial condensing [20]. With the RTI scheme subsequent optimization problems are initialized and aligned to previous solutions, thus leading to reduced computation times [14]. In order to achieve fast computations and generate feasible solutions with SQP-RTI, we need to ensure that the cost-function derived Hessian stays positive semi-definite [6].

IV. APPROACH

Our approach consists of a combination of a novel human-aware collision avoidance formulation for fast embedded MPC together with safety-feasibility monitors. We describe

the novel MPC formulation, the cost function, and the monitors in Sec. IV-A and IV-B.

A. Fast Embedded MPC Formulation

We construct the motion planner as a model predictive controller that solves a discretized optimal control problem (OCP) at each iteration. The OCP has the following structure:

$$\underset{\mathbf{x}, \mathbf{u}}{\text{minimize}} \quad \sum_{n=0}^{N-1} \left(\mathcal{J}_S(\mathbf{x}_n, \mathbf{u}_n, \mathbf{p}_n) \right) + \mathcal{J}_T(\mathbf{x}_N, \mathbf{p}_N) \quad (1a)$$

$$\text{subject to} \quad \mathbf{x}_n \in \mathbb{X} \quad n \in [0, N], \quad (1b)$$

$$\mathbf{u}_n \in \mathbb{U} \quad n \in [0, N-1], \quad (1c)$$

$$d(\mathbf{x}_n, \mathbf{h}_{i,0}) \geq d_h \quad n \in [0, N]; i \in [1, N_h], \quad (1d)$$

$$d(\mathbf{x}_n, \mathbf{o}) \geq d_s \quad n \in [0, N] \quad (1e)$$

where \mathbf{x}_n and \mathbf{u}_n denote respectively the state and the control at a specific time step n on the horizon which is split into N shooting nodes, with \mathbb{X} and \mathbb{U} being the allowed state and control spaces. The additional parameters \mathbf{p}_n include information on the goal \mathbf{g}_n , the predicted position of the human $\mathbf{h}_{i,n}$ for all regarded N_h humans and the position of the current closest static obstacle \mathbf{o} . The distance function $d(\cdot, \cdot)$ calculates the Euclidean distance. d_h is the minimum safe distance from a human, and d_s the allowed distance to static obstacles. The cumulative stage cost is comprised of:

$$\begin{aligned} \mathcal{J}_S(\mathbf{x}_n, \mathbf{u}_n, \mathbf{p}_n) = & \mathcal{J}_g(\mathbf{x}_n, \mathbf{p}_n) \\ & + \mathcal{J}_u(\mathbf{u}_n) \\ & + \mathcal{J}_{\text{col}}(\mathbf{x}_n, \mathbf{p}_n). \end{aligned} \quad (2)$$

For the final shooting node, the terminal cost is:

$$\begin{aligned} \mathcal{J}_T(\mathbf{x}_N, \mathbf{p}_N) = & \mathcal{J}_g(\mathbf{x}_N, \mathbf{p}_N) \\ & + \mathcal{J}_{\text{col}}(\mathbf{x}_N, \mathbf{p}_N). \end{aligned} \quad (3)$$

The goal cost term penalizes the distance to the given goal

$$\mathcal{J}_g(\mathbf{x}_n, \mathbf{p}_n) = \|\mathbf{x}_n - \mathbf{p}_n\|_{W_g}^2 \quad n \in [0, N] \quad (4)$$

weighted by the diagonal matrix W_g .

The control is penalized with the control cost term

$$\mathcal{J}_u(\mathbf{u}_n) = \|\mathbf{u}_n\|_{W_u}^2 \quad n \in [0, N-1] \quad (5)$$

weighted by the diagonal matrix W_u . The collision cost term $\mathcal{J}_{\text{col}}(\mathbf{x}_n, \mathbf{p}_n)$ is described in Sec. IV-B.

The optimization is constrained by various inequality constraints. General collision avoidance is achieved with Equation 1e. Aligned with the state of the art [3] this is implemented as a soft constraint that limits the reachable space around the position of the closest detected obstacle by not allowing the distance between the robot and the obstacle to be less than the distance d_s . Since the cost formulation does not provide guarantees the OCP has to be extended by a hard constraint to assure the safety of the surrounding humans. The state space is constrained by calculating the distance between the robot and the current position of the human for all surrounding humans and requiring this distance to be greater than a minimal distance d_h which respects uncertainties of both the human perception system and the robot localization system in order to provide guarantees [18].

B. Human Awareness and Feasibility

1) *Human Awareness*: General human awareness is achieved by using motion prediction trajectories of the humans in the robot's surroundings, i.e., $\mathbf{T}_{:,1:N}$.

To handle this complexity we propose a cost term based on a logistic function generating a potential field [10]. However, the original term in [10] is not C^2 (i.e., twice continuously differentiable) with a positive semi-definite Hessian and therefore not suitable for the efficient and reliable usage of SQP-RTI with *acados* for fast embedded optimization since feasibility can not be guaranteed with a negative definite Hessian. Therefore, we adopt the following cost function:

$$\mathcal{J}_{\text{col}}(\mathbf{x}_n, \mathbf{p}_n) = \sum_{i=1}^{N_h} f(d(\mathbf{x}_n, \mathbf{h}_{i,n})) \quad (6)$$

$$f(d(\mathbf{x}_n, \mathbf{h}_{i,n})) = \begin{cases} \left(-\frac{\kappa q}{4}\right) d(\mathbf{x}_n, \mathbf{h}_{i,n}) & \text{if } d(\mathbf{x}_n, \mathbf{h}_{i,n}) \leq d_{\text{th}} \\ + \left(\frac{q}{2} + \frac{\kappa q}{4} d_{\text{th}}\right) & \\ \frac{q}{1 + e^{\kappa(d(\mathbf{x}_n, \mathbf{h}_{i,n}) - d_{\text{th}})}} & \text{if } d(\mathbf{x}_n, \mathbf{h}_{i,n}) > d_{\text{th}} \end{cases} \quad (7)$$

Tuning of this cost function is possible by adjusting the smoothness with $\kappa > 0$ and by changing the threshold distance d_{th} where the produced cost is $\frac{q}{2}$ which influences the distance the robot tries to keep from surrounding humans. By using this cost function we aim to find optimal robot motion that maximizes the distance to humans considering their motion predictions (see Figure 1). The novel cost definition is convex for all mathematically relevant and reachable values, and therefore the requirement for the optimization calculation is met (i.e., the optimizer is pushed to find feasible solutions). The linear function lays greater stress on the penalization of close distances between the robot and surrounding humans compared to a logistic potential field.

2) *Feasibility and Safety Monitors*: We implement monitors that make sure that the controls sent to the robot actuators do not harm humans. The safety system has to react to two critical possible situations that can occur during operation:

- C1) The solver does not return a solution in time for real-time decision-making (e.g., overly complex formulation, numerical sub-optimality, infeasibility).
- C2) The solver does not find a safe solution. This can occur if there are numerical issues or calculations end up in constrained state space (e.g., a human is too close).

Regarding case C1) we implement a running time check for the underlying solver. If a solution is not feasible and the optimization exceeds the allowed time frame provided by the real-time control loop appropriate measures can be initiated. To catch case C2) we use the optimization constraint based on the position of all surrounding humans (see Equation 1d and Figure 1). Regarding the relevant safety norms [17] and [18] this is sufficient as long as the source is reliable

which is to be assumed and an adequate safety margin is included. Following the safety norms [17] and [18] the system commands a protective stop in both cases.

C. System Integration

To facilitate easy integration across several robot platforms, we implement *HuMAN-MPC* into the Robot Operating System (ROS). The algorithm is embedded into a plugin for the local planner (ROS 2: Controller; ROS: move_base). Required for operation are two additional parts. One is a node responsible for the formulation of the OCP and the export of the library which is interfaced by *acados* during runtime. The other one is an interface to connect the plugin to the prediction data of the surrounding humans.

V. EXPERIMENTS

We design a set of experiments for evaluating *HuMAN-MPC*. Most of the experiments use Gazebo to simulate the environment and a TurtleBot3 with the ROS integration of *HuMAN-MPC* (see Figure 2). The specifications of the PC used for the experiments are: Intel® Core™ i7-10850H, 32 GB RAM, NVIDIA Quadro RTX 3000.

A. Scenarios, Metrics, and Experiments

We design two main classes of experiments: one focused on the computation capabilities of the approach and the other on the analysis of the robot's behavior performance. While computational performance is the main aspect of the presented approach, latter experiments are conducted to show that the adaptations made to the formulation for fast embedded optimization include all the known advantages that come with using predictions.

a) *For computation efficiency*: We design an experiment to show that *HuMAN-MPC* is efficient, even when considering a large number of surrounding humans and thereby a large number of optimization parameters. In this experiment, an increasing number of up to 30 randomly simulated humans are considered in the optimization.

b) *For behavior performance*: We consider two scenarios: The *random crowded and cluttered* scenario simulates ten human actors with random start and goal poses, which act according to the social force model, and 5 randomly placed static obstacles. In the *crossing group* scenario the robot has to pass a group of nine simulated humans (i.e., actors) walking back and forth through the environment. To show advantages that arise from using predictions we also run the above-described scenarios to compare against the baseline: the open-source local planner Dynamic Window approach (DWB) [21]. Predictions are provided for the length of the optimization horizon using the constant velocity model [22].

B. Parameters and Model

In the experiments, we set the following values for the defined parameters: prediction horizon 5.0 s, number of shooting nodes 50, stage goal cost weights [0.5, 0.5, 0.0, 250.0], terminal goal cost weights [40.0, 40.0, 2.0, 0.0], control cost weights [0.0, 0.0], collision cost: $q = 2.0$ ($d_{\text{th}} = 1$), $\kappa = 5.0$.

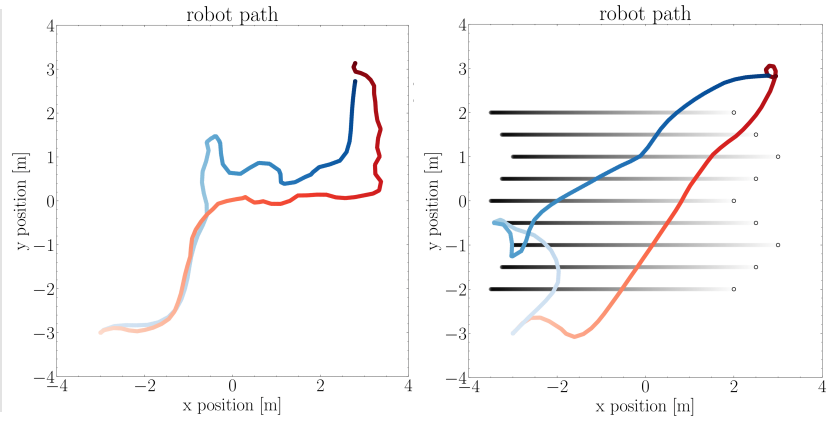
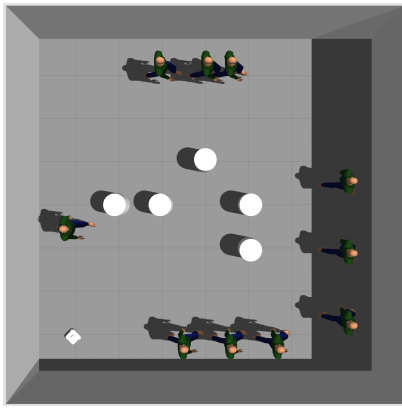


Fig. 2: Left: Scene of the simulation environment with robot, human actors (start poses) and static obstacles; Center: Paths in random crowded and cluttered scenario; Right: Paths in crossing group scenario (**red**: HuMAN-MPC, **blue**: DWB)

Constraint distances $d_s = 0.5$ m, $d_h = 0.5$ m (including safety margin). The model is a differential drive robot [23].

VI. RESULTS

A. Real-Time Performance and Scalability

We observe from our results (mean iteration times are given in Table I) that the optimization steps on average do not take more than 5 ms, thus being well within the control frequency limits and thereby theoretically allowing for control frequencies even over 100 Hz. Furthermore, the embedded optimization scales notably well when increasing the number of parameters. Table I

		Mean iteration time [ms]
Num. of humans – (Num. of parameters per shooting node)	5 – (27)	2.47
	10 – (47)	3.08
	20 – (87)	3.92
	30 – (127)	4.75

TABLE I: Iteration times considering a differing number of simulated humans.

Referring to the comparison experiments carried out in [6] (see Table II) it is furthermore shown that the *acados* based optimization is superior in terms of computational efficiency (i.e., two orders of magnitude) to other relevant optimization solvers. Aligned to this iteration times are much higher in our previous works [3], [4] and [2] with 34 ms, 17.26 ms and 64 ms respectively. Note that the settings are not identical but only similar. Nevertheless, this still shows great progress in computational performance.

Iteration time [ms]		acados	IPOPT	ACADO	GRAMPC
	Mean	1.05	59.84	1.97	1.06
	Min.	0.87	49.06	1.90	0.81
	Max.	2.23	384.90	3.45	1.31
RCSO		1.01e-04	0.00e+00	1.01e-04	7.17e-02

TABLE II: Case study results for *acados* obtained in [6]. *acados* achieves the best mean MPC iteration time, with only minor relative cumulative sub-optimality (RCSO). IPOPT achieves the optimal solution while being on average 60 times slower than *acados*.

B. Robot Behavior Performance

Evaluating both scenarios shows that making use of predictions allows HuMAN-MPC to navigate the environment more intuitively (i.e., passing behind the group, giving space to the humans), smoother and more legible (i.e., not getting stuck, not coming too close to humans). The resulting paths are shown in Figure 2.

C. Feasibility and Safety Monitors

The monitoring system initiated a protective stop at every harmful occasion and at every occasion where optimization running times exceeded the limit. This goes to show that the implemented monitors (described in subsection IV-B) are able to handle critical scenarios in compliance with the relevant safety norms [17] and [18]. Furthermore, it is to note that our approach triggered the safety monitors less often for reasons of faster computation and development of fewer safety critical situations.

D. Real Robot

Additionally, we show functionality on a real robot using the robot platform of the research project DARKO¹. The approach has been tested in several experiments which resulted in smooth movements without collisions. This also shows the advantages of the ROS integration. See Figure 3.

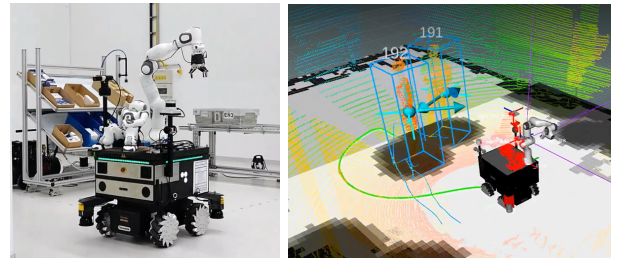


Fig. 3: Left: DARKO robot. Right: Generated path.

VII. CONCLUSIONS

To achieve efficient robot motion planning considering human motion predictions we propose HuMAN-MPC. The approach presents a novel formulation for fast embedded model predictive control. The optimization-based motion planning incorporates predictions of surrounding humans into an SQP-RTI scheme, while additionally verifying reliable and safe operation with monitors for feasibility and safety. Simulated and real-world experiments show that our approach is able to achieve substantially faster computation compared to other state-of-the-art baselines and provide great scalability while keeping predictive avoidance behavior. In future work, we plan to use more contextual information about the surroundings (e.g., semantics, human activities) leveraging the great scalability.

¹EU Project DARKO, <https://darko-project.eu/>

REFERENCES

- [1] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *Int. Journal of Robotics Research*, vol. 39, no. 8, 2020.
- [2] T. Schoels, P. Rutquist, L. Palmieri, A. Zanelli, K. O. Arras, and M. Diehl, "Ciao*: Mpc-based safe motion planning in predictable dynamic environments," *IFAC-PapersOnLine*, vol. 53, no. 2, 2020.
- [3] L. Heuer, L. Palmieri, A. Rudenko, A. Mannucci, M. Magnusson, and K. O. Arras, "Proactive model predictive control with multi-modal human motion prediction in cluttered dynamic environments," in *Int. Conf. Intell. Robot. Sys. (IROS)*. IEEE, 2023.
- [4] T. Schoels, L. Palmieri, K. O. Arras, and M. Diehl, "An NMPC approach using convex inner approximations for online motion planning with guaranteed collision avoidance," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 2020.
- [5] S. Schaefer, K. Leung, B. Ivanovic, and M. Pavone, "Leveraging neural network gradients within trajectory optimization for proactive human-robot interactions," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 2021.
- [6] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, Oct 2021. [Online]. Available: <https://doi.org/10.1007/s12532-021-00208-8>
- [7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. & Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.
- [8] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 1993.
- [9] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 2008.
- [10] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2017, pp. 236–243.
- [11] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 2018.
- [12] Y. Chen, U. Rosolia, C. Fan, A. Ames, and R. M. Murray, "Reactive motion planning with probabilistic safety guarantees," in *Conf. Robot Learn. (CoRL)*, vol. 155, 2020, pp. 1958–1970.
- [13] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic mpc with multi-modal predictions for traffic intersections," in *2022 IEEE 25th Int. Conf. on Intelligent Transportation Systems (ITSC)*, 2022, pp. 635–640.
- [14] M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, "Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations," *J. Process Control*, vol. 12, no. 4, pp. 577–585, 2002.
- [15] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10 357–10 377, 2021.
- [16] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *Int. Conf. Robot. & Autom. (ICRA)*. IEEE, 2017.
- [17] "DIN EN ISO 3691-4:2020-11; industrial trucks - safety requirements and verification - part 4: Driverless industrial trucks and their systems (ISO 3691-4:2020)," Nov. 2020.
- [18] "DIN EN ISO 13482:2014-11; robots and robotic devices - safety requirements for personal care robots (ISO 13482:2014)," Nov. 2014.
- [19] G. Frison, D. Kouzoupis, T. Sartor, A. Zanelli, and M. Diehl, "Blasfeo: basic linear algebra subroutines for embedded optimization," *ACM Transactions on Mathematical Software*, vol. 44, no. 4, pp. 1–30, 2018.
- [20] G. Frison and M. Diehl, "Hpipm: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.
- [21] DWB controller. [Online]. Available: https://github.com/ros-planning/navigation2/tree/main/nav2_dwb_controller
- [22] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1696–1703, 2020.
- [23] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.