

Safe Human-Robot Coexistence Through Model Predictive Control Barrier Functions and Motion Distributions^{*}

Mohammadreza Davoodi^{*} Joseph M. Cloud^{**} Asif Iqbal^{*}
William J. Beksi^{**} Nicholas R. Gans^{*}

^{*} University of Texas at Arlington Research Institute, Fort Worth, TX, USA. (mdavoodi@uta.edu, asif.iqbal@uta.edu, nick.gans@uta.edu.)

^{**} Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, USA. (joe.cloud@mavs.uta.edu, william.beksi@uta.edu.)

Abstract: Future real-world applications will consist of robots and human workers collaborating with each other in a shared environment to increase productivity. In such scenarios, it is necessary to guarantee the safety of humans while maintaining precise control of the robots performing tasks. Probabilistic movement primitives (ProMPs) are a powerful tool for defining a distribution of trajectories for dynamic systems. However, they have been solely used for determining robot trajectories. In this paper, we utilize ProMPs to predict the probabilistic motion of humans in the environment. To achieve this, we propose a combination of model predictive control (MPC) and control barrier functions (CBFs) to guide a robot along a predefined trajectory while guaranteeing it always maintains a desired distance from a human worker motion distribution defined by a ProMP. A case study is provided to demonstrate the efficacy of our methods.

Copyright © 2021 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Probabilistic Movement Primitives; Control Barrier Functions; Model Predictive Control; Human-Robot Coexistence

1 INTRODUCTION

Cooperation between humans and robots has increased in many applications such as industry and in the home (Kanazawa et al. (2019); Fishman et al. (2020)). In these scenarios, robots may perform repetitive and simple jobs, as well as physically demanding tasks outside the abilities of humans. Currently, human workers are typically responsible for tasks that are challenging for robots, such as situation assessments. To have an efficient and united human-robot system, their advantages must be integrated in a collaborative manner. One of the main challenges for realizing a seamless, combined system is to guarantee the safety of human workers in the environment (Kimmel and Hirche (2017)). Two important aspects for accomplishing safe cooperation are the following: (i) modeling human-motion trajectories in a way that captures humans' relatively low-levels of repeatability in the environment; (ii) synthesizing an appropriate controller for a robot to avoid collisions with human trajectories.

1.1 Human Motion Prediction

Successful human-robot collaboration requires models of human behavior (e.g., motion). This problem has been

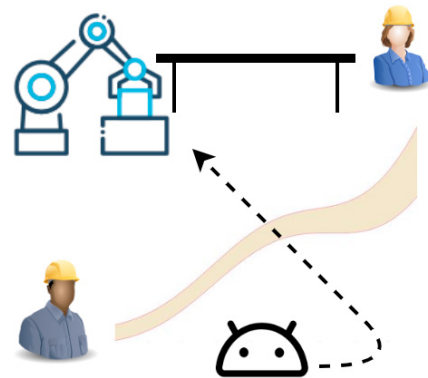


Fig. 1. A worker heading towards another worker. The motion of the moving worker is modeled as a distribution generated using a ProMP and transmitted to a mobile robot. Our controller guides the robot to its final position and simultaneously guarantees collision avoidance with other humans.

investigated in a variety of contexts (Bai et al. (2015); Zhou and Wachs (2018)). Human behavior, even when performing repetitive jobs, does not have a repeatable and deterministic pattern. It rather depends on many factors including fatigue, skill level, and adaptation to various tasks. Consequently, it's more effective to consider probabilistic models such as Gaussian mixture models (Wiest et al. (2012)) which can model the prediction uncertainty. Moreover, from a control theory point of view, it is important to have methods capable of predicting

^{*} This material is based upon work supported by the National Science Foundation (NSF) under grant No. 1728057 and the University of Texas at Arlington Research Institute. Joseph M. Cloud was supported by an NSF Graduate Research Fellowships Program grant No. 1746052.

human motion over a future time horizon instead of just delivering the current human dynamical states.

In this work, we use ProMPs for modeling human motion in the environment whereby a distribution of trajectories is learned from multiple demonstrations or observations (Calinon and Lee (2017)). In Paraschos et al. (2018a), the design of a stochastic ProMP feedback controller was studied by exploiting the property of the covariance derivatives which can be explicitly computed. A model-free ProMP controller that adapts movements to a force-torque sensor input was developed in Paraschos et al. (2018b). Unlike Paraschos et al. (2018a,b) we do not design a ProMP controller in this work, instead we utilize ProMPs to model human motion and for defining safety barriers.

1.2 Safe Robot Control Approaches

Real-time safety in safety-critical dynamical systems is a paramount issue (Wieland and Allgöwer (2007)). This problem has been investigated by designing polynomial barrier certificates/functions using offline iterative algorithms based on sum-of-squares optimization to verify safety for a given dynamical system (Sloth et al. (2012)). The concept of barrier certificates/functions was extended for synthesizing real-time safe control laws for dynamical systems via CBFs (Ames et al. (2016)).

CBFs can provide verifiable control laws with safety guarantees. They integrate seamlessly with control Lyapunov functions (CLFs) to offer stability while respecting limits and safe regions of the state space (Ames et al. (2019)). Additionally, CBF and CLF controllers typically solve a constrained quadratic program (QP) to find an optimal controller at runtime. This allows the system to minimize the control effort while ensuring stability and safety. Other tasks formulated as cost functions or constraints can be included as well. One downside to CBFs and CLFs is the complexity in defining the barriers and trajectories. Efforts to automate the definition of CBFs and CLFs include mapping temporal logic statements with respect to performance requirements (Srinivasan and Coogan (2020)) and fitting piecewise-linear barrier functions to trained obstacles or regions (Saveriano and Lee (2020)).

Although CBFs are effective for safe control design in engineering systems, most existing techniques only assure myopic short-term safety constraints for simple nonlinear dynamical systems. On the other hand, MPC is a well-established method for real-time control of dynamical systems subject to system constraints and has a long history in industrial applications (Qin and Badgwell (2003)). While MPC is a useful tool for addressing various types of system constraints, there exists some safety constraints that cannot be adequately handled. However, MPC and CBFs can be unified together into one optimization problem to address such challenges (Zeng et al. (2020)).

1.3 Contributions

This work investigates the concept of human-robot collaboration from the viewpoint of having an efficient level of safe coexistence (Magrini et al. (2020)). For example, this may occur when a robot and human work close to each other in a shared workspace without requiring mutual contact or the coordination of actions and intentions

(De Luca and Flacco (2012)). We present a safe human-robot coexistence system for anticipating the surrounding human workers' motions based on the assumption that there exists a centralized high-level task assignment and observer (HLTAO) module for assigning tasks to and monitoring all humans and robots in the environment.

Our goal is to design effective control strategies for robots such that they are capable of achieving their assigned tasks while avoiding collisions with other robots and humans. To accomplish this objective, the possible motions of the human workers in the environment are modeled as a set of ProMPs. Using HLTAO, each individual robot can receive the current location of the neighboring robots as well as the position and distribution (represented by ProMPs) of the surrounding humans. These human distributions and robot positions are then used to define two sets of CBFs. We propose an MPC/CBF control method that simultaneously guarantees the control performance and the safety of the system.

2 BACKGROUND

In this section we provide the essential background information on ProMPs, CBFs, and MPC.

Notation: Given a matrix A , let $A^\top$ denote its transpose. The number of axes of the workspace is represented by n . We denote the sample number for a discrete-time state or parameter by subscript, e.g., A_k . Given a vector $x \in \mathbb{R}^n$, we indicate its j th element at the k th time step with a subscript $x_{j,k}$. j is used as a subscript throughout Section 2.1 to refer to the axis of the workspace. The superscript j in x^j is used to denote the j th robot, human, or task, and is also utilized throughout Section 3 in this context for other variables.

2.1 Probabilistic Movement Primitives

ProMPs provide a parametric representation of trajectory distributions. These parameters facilitate modulation in both space and time, combination and co-activation with other movement primitives, and coupling between joints or axes to support coordinated movements. Basis functions are used to reduce model parameters and aid learning over demonstrated trajectories. The trajectory distribution can be defined and generated in any space that accommodates the system (e.g., configuration/joint or work/task spaces). In this work, we consider tasks in the workspace.

Within a ProMP, a trajectory is represented as a set of Cartesian positions $\zeta_j = \{x_{j,k}\}$, with state variable $x_{j,k} \in \mathbb{R}$, axis $j \in [1, \dots, n]$, and k is the time step. Let $w_j \in \mathbb{R}^{1 \times L}$ be a weight matrix with L terms. A linear basis function model is then given by

$$\varrho_{j,k} = \begin{bmatrix} x_{j,k} \\ \dot{x}_{j,k} \end{bmatrix} = \Phi_k w_j + \xi_{x_j},$$

where $\Phi_k = [\phi_k \ \dot{\phi}_k]^\top \in \mathbb{R}^{2 \times L}$ is the time-dependent basis function matrix, and L is the number of basis functions. Gaussian noise is described by $\xi_{\varrho_j} \sim \mathcal{N}(0, \Sigma_{\varrho_j})$. Thus, a ProMP trajectory is represented by a Gaussian distribution over the weight vector w_j and the parameter vector $\theta_j = \{\mu_{w_j}, \Sigma_{w_j}\}$ which simplifies the estimation of the parameters. We marginalize out w_j to create the trajectory distribution, i.e.,

$$p(\zeta_j, \theta_j) = \int p(\zeta_j | w_j) p(w_j; \theta_j) dw_j. \quad (1)$$

The distribution $p(\zeta_j, \theta_j)$ defines a hierarchical Bayesian model over the trajectories ζ_j (Paraschos et al. (2018a)), and $p(w_j | \theta_j) = \mathcal{N}(w_j | \mu_{w_j}, \Sigma_{w_j})$. In a movement primitive representation, the parameters of a single primitive must be easy to obtain from demonstrations. The distribution of the state $p(\varrho_{j,k}; \theta_j)$ is

$$p(\varrho_{j,k}; \theta_j) = \mathcal{N}(\varrho_{j,k} | \Phi_k \mu_{w_j}, \Phi_k \Sigma_{w_j} \Phi_k^\top + \Sigma_{\varrho_j}). \quad (2)$$

A trajectory can be generated from the ProMP distribution using w_j , the basis function Φ_k , and (2). The basis function is chosen based on the type of robot (or human) movement, which can be either rhythmic or stroke-based. From (2), the mean $\mu_{j,k} \in \mathbb{R}^2$ of the ProMP trajectory at k is $\Phi_k \mu_{w_j}$ and the covariance $\Sigma_{j,k} \in \mathbb{R}^{2 \times 2}$ is $\Phi_k \Sigma_{w_j} \Phi_k^\top + \Sigma_{\varrho_j}$.

Multiple demonstrations are needed to learn a distribution over w_j . We use a combination of radial basis and polynomial basis functions for training. From the demonstrations, the parameters θ_j can be estimated using maximum likelihood estimation (Lazaric and Ghavamzadeh (2010)). However, this may result in unstable estimates of the ProMP parameters when there are insufficient demonstrations. Our method uses a regularization to estimate the ProMP distribution similar to Gomez-Gonzalez et al. (2020). We maximize θ_j for the posterior distribution over the ProMP using expectation maximization, $p(\theta_j | x_{j,k}) \propto p(\theta_j) p(\varrho_{j,k} | \theta_j)$.

For systems with multiple degrees of freedom, where $n > 1$, coupling between axes of the workspace enables the robot (or human) to exhibit coordinated behavior. To realize this with ProMPs, we calculate the covariance of the axes (Paraschos et al. (2018b)). In addition, we use a Normal-Inverse-Wishart as a prior distribution $p(\theta_j)$ to increase stability when training the ProMP parameters (Gomez-Gonzalez et al. (2020)). Assuming two, single-axis ProMPs ($\mu_{j,k}, \Sigma_{j,k}$), $j \in \{1, 2\}$, the joint distribution is defined as ($\tilde{\mu}_k \in \mathbb{R}^4, \tilde{\Sigma}_k \in \mathbb{R}^{4 \times 4}$). Finally, by removing the rows and columns related to the axis velocities, ($\mu_k \in \mathbb{R}^2, \Sigma_k \in \mathbb{R}^{2 \times 2}$) are obtained and used as the 2D model of human motion in the workspace as described in Section 3.

2.2 Control Barrier Functions

Consider the following discrete-time nonlinear system

$$x_{k+1} = f(x_k, u_k), \quad (3)$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ denotes the state, $u \in \mathcal{U} \subset \mathbb{R}^p$ is the control input, and $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a locally Lipschitz vector field. It is assumed that the system in (3) is controllable.

Assumption 1: All system states x_k can be measured or estimated at each time step k .

Define a set C for which we wish to verify that $x(t) \in C, \forall t$. C then defines a *safe set*. A smooth function $h(x): \mathbb{R}^n \rightarrow \mathbb{R}$ is defined to encode a constraint on the state x of the system. The constraint is satisfied if $h(x) \geq 0$, and violated if $h(x) < 0$. Consider the set C defined by

$$\begin{aligned} C &= \{x \in \mathbb{R}^n : h(x) \geq 0\}, \\ \partial C &= \{x \in \mathbb{R}^n : h(x) = 0\}, \\ \text{Int}(C) &= \{x \in \mathbb{R}^n : h(x) > 0\}, \end{aligned} \quad (4)$$

where $\text{Int}(C)$ and ∂C denote the interior and boundary of C , respectively.

Existing approaches to define CBFs include exponential CBFs, zeroing CBFs, and reciprocal CBFs (Nguyen and Sreenath (2016); Ames et al. (2016)). These methods have trade-offs in terms of ease of definition, boundedness of velocities, speed of convergence, etc. In this work we investigate discrete-time exponential CBFs.

Definition 1. Given the set C , the smooth function h is a CBF if there exists a constant scalar γ such that (Agrawal and Sreenath (2017)),

1. $h(x_0) \geq 0$,
2. There exists a control input u_k such that

$$\Delta h(x_k, u_k) \geq -\gamma h(x_k), \quad 0 < \gamma \leq 1, \quad (5)$$

where $\Delta h(x_k, u_k) = h(x_{k+1}) - h(x_k)$. From (5) it is clear that $h(x_{k+1}) \geq (1 - \gamma)h(x_k)$. Hence, the barrier function $h(x_k)$ always has a lower bound $(1 - \gamma)^k h(x_0)$, and is an exponential function in k . Due to this fact, it is referred to as a discrete-time exponential CBF.

2.3 Model Predictive Control

The main goal of MPC is to find a sequence of control actions by solving a finite-horizon optimization problem at each sampling instant. Concretely, the MPC problem is formulated as (Grandia et al. (2020); Zeng et al. (2020))

$$\underset{u}{\text{argmin}} V(x_{k+N|k}) + \sum_{t=0}^{N-1} L(x_{k+t|k}, u_{k+t|k}) \quad (6a)$$

subject to

$$x_{k+t+1|k} = f(x_{k+t|k}, u_{k+t|k}), \quad t = 0, \dots, N-1, \quad (6b)$$

$$x_{k+t|k} \in \mathcal{X}, \quad u_{k+t|k} \in \mathcal{U}, \quad t = 0, \dots, N-1, \quad (6c)$$

$$x_{k+N|k} \in \mathcal{X}_f, \quad (6d)$$

$$x_{k|k} = x_k, \quad (6e)$$

where $V(\cdot)$ and $L(\cdot)$ are, respectively, the terminal and stage costs, and N is the time horizon. The state and input constraints are given by (6c), and the terminal constraint is enforced in (6d). To solve an MPC problem, at each time step k , (6) is solved and a sequence of control actions $\mathbf{u}^* = [u_{k|k}, \dots, u_{k+N-1|k}]$ is generated. Then, only the first element of $\mathbf{u}^*$ is applied (as a feedback law) to the system dynamics and the next measured state x_{k+1} is estimated. Finally, the estimated state is considered as a new initial condition for the next time step and the optimization problem is repeated.

3 MPC/CBF FORMULATION

3.1 Problem Formulation

Consider an environment (denoted by $\mathcal{Q} \subset \mathbb{R}^n$), with t tasks designated by Task^o , $o \in \mathcal{T} = \{1, \dots, t\}$, m human workers $\mathcal{H}^h$, $h \in \mathcal{H} = \{1, \dots, m\}$, and l heterogeneous robots $\mathcal{R}^r$, $r \in \mathcal{R} = \{1, \dots, l\}$ that work in close proximity to humans to complete a set of predefined tasks. Similar to (3), each robot is modeled as a nonlinear dynamical system, i.e., $x_{k+1}^i = f^i(x_k^i, u_k^i)$, $i \in \mathcal{R}$, with state vector $x^i \in \mathbb{R}^{n_i}$, and control input $u^i \in \mathbb{R}^{p_i}$.

Problem: Our primary objective is to control the robots in real-time such that (i) they avoid collisions with human

robots. The Mahalanobis distance, as a measure of the distance between the current robot position $p_{\mathcal{R}^i_k}$ and the current distribution of a human $\mathcal{N}^j_k = (\mu^j_k, \Sigma^j_k)$, is used to constrain the minimal distance to the human and avoid collisions. It is calculated as

$$D_M(p_{\mathcal{R}^i_k}, \mu^j_k, \Sigma^j_k) = \sqrt{(p_{\mathcal{R}^i_k} - \mu^j_k)^\top \Sigma^{j-1}_k (p_{\mathcal{R}^i_k} - \mu^j_k)}.$$

The Mahalanobis distance is used to define the CBFs h_1^{ij}

$$h_1^{ij} = (p_{\mathcal{R}^i_k} - \mu^j_k)^\top \Sigma^{j-1}_k (p_{\mathcal{R}^i_k} - \mu^j_k) - \epsilon_{th}^2, \quad j \in \mathcal{N}_{\mathcal{H}^i}, \quad (7)$$

where ϵ_{th} acts as a safety factor by specifying a bound for the minimal distance to the humans in the workspace. Thus, ϵ_{th} can be chosen by the designer (mission planner) based on the safety requirements of the tasks in the environment. For example, if the job is dangerous (e.g., the robot is carrying a hazardous payload as part of its task) larger values can be considered for ϵ_{th} to create more distance between the robot and the humans.

Neighboring robots are modeled as circles with centroids $p_{\mathcal{R}^j} = (p_{\mathcal{R}^j_x}, p_{\mathcal{R}^j_y})$ and fixed radii r^j . Thus, the CBFs h_2^{ij} that represent the safety constraint between robot i and its neighboring robots are defined as

$$h_2^{ij} = (p_{\mathcal{R}^i_x} - p_{\mathcal{R}^j_x})^2 + (p_{\mathcal{R}^i_y} - p_{\mathcal{R}^j_y})^2 - r^{j2}, \quad j \in \mathcal{N}_{\mathcal{R}^i}. \quad (8)$$

Based on the results of Grandia et al. (2020) and Zeng et al. (2020), the MPC/CBF problem is formulated as

$$\underset{u^i}{\operatorname{argmin}} V^i(x_{k+N|k}^i) + \sum_{t=0}^{N-1} L^i(x_{k+t|k}^i, u_{k+t|k}^i) \quad (9a)$$

subject to

$$x_{k+t+1|k}^i = A^i x_{k+t|k}^i + B^i u_{k+t|k}^i, \quad t = 0, \dots, N-1, \quad (9b)$$

$$x_{k+t|k}^i \in \mathcal{X}^i, \quad u_{k+t|k}^i \in \mathcal{U}^i, \quad t = 0, \dots, N-1, \quad (9c)$$

$$x_{k+N|k}^i \in \mathcal{X}_f^i, \quad x_{k|k}^i = x_k^i, \quad (9d)$$

$$\Delta h_1^{ij}(p_{\mathcal{R}^i_{k+t|k}}, u_{k+t|k}^i, \mathcal{N}_{k+t|k}^j) \geq \quad (9e)$$

$$- \gamma_1 h_1^{ij}(p_{\mathcal{R}^i_{k+t|k}}, \mathcal{N}_{k+t|k}^j), \quad j \in \mathcal{N}_{\mathcal{H}^i},$$

$$\Delta h_2^{ij}(p_{\mathcal{R}^i_{k+t|k}}, u_{k+t|k}^i, p_{\mathcal{R}^j_{k|k}}) \geq \quad (9f)$$

$$- \gamma_2 h_2^{ij}(p_{\mathcal{R}^i_{k+t|k}}, p_{\mathcal{R}^j_{k|k}}), \quad j \in \mathcal{N}_{\mathcal{R}^i}.$$

where $x_{k+1}^i = A^i x_k^i + B^i u_k^i$ is the linear discrete-time counterpart to the system in (3), corresponding to i th robot system dynamics. The nonlinear distance constraints for the safety criteria are given by conditions (9e) and (9f). One option for the stage cost is $L^i(x_k^i, u_k^i) = (x_k^i - x_{\text{ref}_i}^i)^\top Q^i (x_k^i - x_{\text{ref}_i}^i) + u_k^{i\top} R^i u_k^i$. In this work, $x_{\text{ref}_i}^i = [p_x^{\text{Task}_i}, p_y^{\text{Task}_i}, 0_{1 \times (n_i-2)}]$ such that the i th robot will reach its corresponding task, Task i . Note that this type of cost function minimizes the system input similar to $u_k^{i\top} H(x^i) u_k^i$ in a typical CLF/CBF control design.

In the MPC/CBF control design problem, the terminal cost V^i and the terminal set $\mathcal{X}_f^i \subseteq \mathcal{X}^i$ can be utilized to guarantee the stability of the system along the closed-loop trajectory. Instead of using the CLF constraints in the conventional CLF/CBF-based QP optimization (Ames et al. (2019)), in our formulation the terminal cost minimizes the CLF.

The following simple procedure can be used to guarantee stability using the terminal cost (Christofides et al. (2013)). We assume a linear stabilizing feedback control

law $u_k^i = K^i x_k^i$ exists for the unconstrained case, i.e., $A^i + B^i K^i$ is stable. Such a controller can be computed using the solution of an infinite-horizon linear quadratic regulator problem with the same weights Q^i and R^i used in the MPC optimization problem. Then, letting P^i be the solution of the Lyapunov equation

$$(A^i + B^i K^i)^\top P^i (A^i + B^i K^i) - P^i = -(Q^i + K^{i\top} R^i K^i),$$

the terminal cost and set are chosen to be $V^i(x_N^i) = x_N^{i\top} P^i x_N^i$ and $\mathcal{X}_f^i = \{x^i | x^{i\top} P^i x^i \leq c^i\}$, where c^i is a small positive value chosen such that $u_k^i = K^i x_k^i \in \mathcal{U}^i$ for any $x^i \in \mathcal{X}_f^i$. More details on the stability analysis can be found in Borrelli et al. (2017).

Remark 1. Based on the considered stage cost in (9), it is guaranteed that each robot safely achieves its final position ($x_{\text{ref}_i}^i$) from its current position. However, it might be practically interesting to have scenarios in which robots need to follow some predefined distributions while avoiding collisions with humans. The proposed method can be modified in such a way that it forces robots to move inside predefined distributions, delivered from pre-trained ProMPs. To this end, the cost function in (9) should behave as a ProMP-based controller for each robot. In this case, the stage cost of the MPC is replaced with $L^i(x_k^i, u_k^i) = (x_k^i - \mu_k^i)^\top \Sigma_k^{-1} (x_k^i - \mu_k^i) + u_k^{i\top} R^i u_k^i$, where μ^i and Σ^i come from pre-trained ProMPs for the i th robot. Moreover, to avoid collisions between robot and human distributions, the Mahalanobis distance should be replaced with the Bhattacharyya distance. The Bhattacharyya distance, D_B , measures the similarity of two probability distributions. For two normal distributions $\mathcal{N}^i = (\mu^i, \Sigma^i)$ and $\mathcal{N}^j = (\mu^j, \Sigma^j)$, the distance is defined as

$$D_B = \frac{1}{8} (\mu^i - \mu^j)^\top \Sigma^{-1} (\mu^i - \mu^j) + \frac{1}{2} \ln \left(\frac{\det \Sigma}{\det \Sigma^i \det \Sigma^j} \right),$$

where $\Sigma = \frac{\Sigma^i + \Sigma^j}{2}$.

4 SIMULATION RESULTS

In this section a proof of concept example is provided to highlight the effectiveness of our methodology for a safe human-robot coexistence system. The system models and proposed real-time controller were simulated using MATLAB 2019a. All computations were done on a Dell OptiPlex 7050 machine with an Intel Core i7-7700X CPU and 8 GB of memory.

Consider an environment $\mathcal{Q}$ with two robots ($\mathcal{R}^1, \mathcal{R}^2$), one human ($\mathcal{H}^1$), and three tasks (Task¹, Task², Task³). The positions of the tasks in the environment are, respectively, (0.3270, 0.2535), (−0.3813, 1.6545), (−0.15, 1.8). The initial positions of the robots are $\mathcal{R}^1 = (−0.1, 1.6)$, $\mathcal{R}^2 = (0.4, 0.1)$. The human is initially at Task¹. The HLTAO module has assigned Task² to the human and Task³ to the second robot. It's assumed that the first robot does not move during the mission and acts as a static obstacle to the second robot. The goal is to control the second robot ($\mathcal{R}^2$) to move from its initial position to Task³ without colliding with the human ($\mathcal{H}^1$) and the other robot ($\mathcal{R}^1$).

The human motion between Task¹ and Task² is modeled as a ProMP. We generated 49 different trajectories that arrive

in a neighborhood of Task² with a fixed starting position. Using this dataset, we trained the ProMP with Algorithm 1 from Gomez-Gonzalez et al. (2020). We used $L = 5$ basis functions consisting of five radial basis parameters. The results of the ProMP training are presented in Fig. 4 with the mean of the human demonstration shown in green, and the mean $\pm$ variance bounds shown with dotted black lines. We assume the human moves with a constant speed and it takes 8 seconds for the human to reach Task² from Task¹.

The robots are modeled as a linear discrete-time 2D double integrator system $x_{k+1}^i = A^i x_k^i + B^i u_k^i$, where

$$A^i = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B^i = \begin{bmatrix} 0.5dt^2 & 0 \\ 0 & 0.5dt^2 \\ dt & 0 \\ 0 & dt \end{bmatrix}. \quad (10)$$

The sampling time is set to $dt = 0.05$. The weighting parameters of the MPC/CBF controller are selected as $Q^i = 0.1I_4$, $R^i = I_2$ and $P^i = 100I_4$. The CBF parameters γ_1 and γ_2 are initialized to $\gamma_1 = \gamma_2 = 0.4$. The fixed radius in the second CBF is set to $r^1 = 0.05$. Additionally, we assume the robot model (10) is subject to the following state and input constraints

$$\mathcal{X}^i = \{x_k^i \in \mathbb{R}^{n_i}, -5I_{4 \times 1} \leq x_k^i \leq 5I_{4 \times 1}\}, \quad (11)$$

$$\mathcal{U}^i = \{u_k^i \in \mathbb{R}^{p_i}, -0.5I_{2 \times 1} \leq u_k^i \leq 0.5I_{2 \times 1}\}. \quad (12)$$

IPOPT, an open-source software package for large-scale nonlinear optimization (Wächter (2009)), is used as the solver for the proposed optimization problem.

Three different scenarios are considered. In *scenario 1*, the safety factor $\epsilon_{th} = 0.1$ and prediction horizon $N = 4$. For *scenario 2*, $\epsilon_{th} = 0.85$ and $N = 4$. In *scenario 3*, $\epsilon_{th} = 0.1$ and $N = 8$. The results are presented in Figs. 4, 5, 6. By comparing *scenario 1* and *scenario 2* in Fig. 4, it is clear that increasing the safety factor ϵ_{th} increases the distance of the robot to the human distribution. It can be seen that a larger horizon N (*scenario 3*) causes the robot to avoid obstacles (human $\mathcal{H}^1$, and robot $\mathcal{R}^1$) earlier. Specifically, among all scenarios, the robot trajectory in *scenario 3* is farthest from the obstacles. Moreover, the designed controller in *scenario 3* is faster than the controllers in the other scenarios and the system starts to avoid the obstacle earlier. This indicates that the robot is operating within a smaller safe set, i.e., the system tends to be safer. Furthermore, we can see that with a larger horizon N , the system has noticeable obstacle avoidance behavior when it is near obstacles. As can be seen from Fig. 6, in *scenario 3* robot $\mathcal{R}^2$ reaches the static obstacle (robot $\mathcal{R}^1$) after approximately 4.15 seconds (note that $h_2 \approx 0$), which is sooner than the other scenarios, with $h_2 \approx 0$ at about 9 seconds. Consequently, having a larger horizon time (*scenario 3*) is more efficient for cases where the robot should reach its final task (goal position) in a limited amount of time.

In the simulations, the main computational cost with respect to the time of our controller comes from the fact that it has to solve a nonlinear optimization problem using IPOPT at every time step. To visualize the effect of the prediction horizon on the computational complexity of our proposed method, we compared the computational times in *scenario 2* (Sc2) and *scenario 3* (Sc3). The

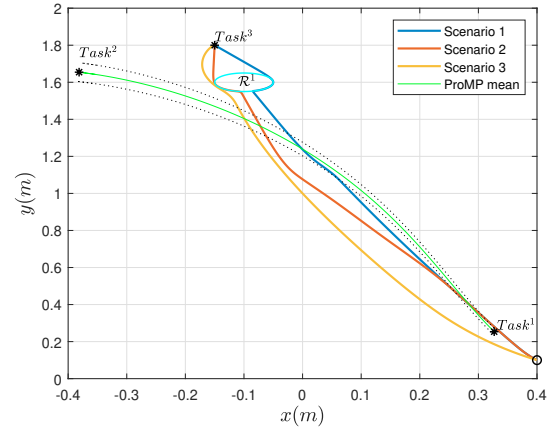


Fig. 4. A visualization of the trajectories of robot $\mathcal{R}^2$ in the presence of a 2D human ProMP in the workspace. Increasing the safety factor ϵ_{th} (*scenario 2*) expands the distance of the robot to the human distribution. Furthermore, by considering a larger horizon N (*scenario 3*) the robot avoids obstacles earlier.

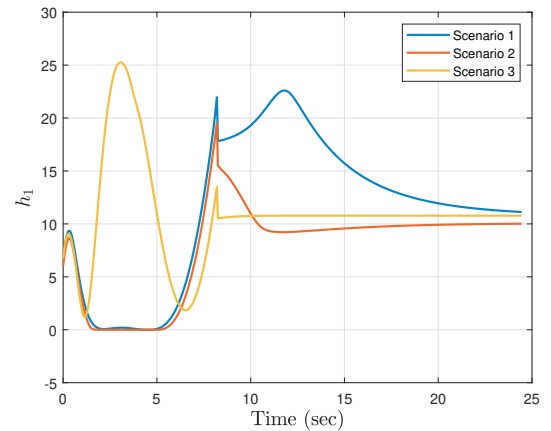


Fig. 5. The first CBF represents the safety constraint between the robot ($\mathcal{R}^2$) and neighboring human ($\mathcal{H}^1$). Note that the distance between robot and human never crosses zero thus assuring collision avoidance.

average required time (T_{ave}), maximum time (T_{max}), and the standard deviation (std) for solving the optimization problem of these scenarios are as follows: $T_{ave}^{Sc2} = 0.0107$, $T_{max}^{Sc2} = 0.0717$, $std^{Sc2} = 0.0036$, $T_{ave}^{Sc3} = 0.0112$, $T_{max}^{Sc3} = 0.1571$, $std^{Sc3} = 0.0074$, where the units are seconds. Based on these results, it can be concluded that a larger prediction horizon in the proposed MPC/CBF controller leads to higher computational time. Furthermore, it is clear that the expected execution time of the optimization problems is very small (in the range of 10 ms). The large maximum times are each a single outlier. Hence, the controller is appropriate for a real-time implementation.

5 CONCLUSIONS

In this paper, we investigated the design of a distributed, real-time controller for safe human-robot coexistence. We assumed there exists a HLTAO module capable of monitoring and assigning tasks to all humans and robots in the environment. Furthermore, the extent of possible human motions as they move from one task to another were mod-

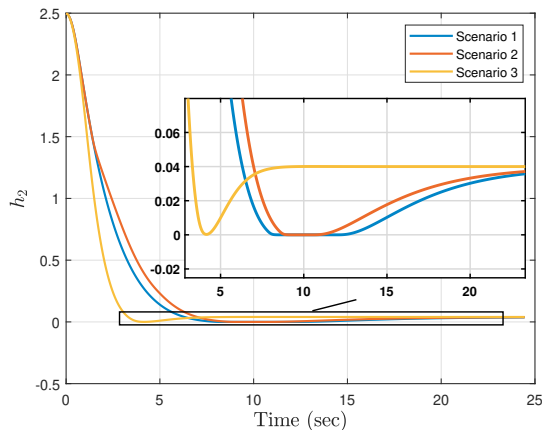


Fig. 6. The second CBF represents the safety constraint between the robot ($\mathcal{R}^2$) and neighboring robot ($\mathcal{R}^1$). Note that the minimal distance between robots never crosses zero thus assuring collision avoidance.

eled as a set of ProMPs. By transmitting these ProMPs and the current location of the humans to the robots through HLTAO, an MPC/CBF controller was designed for each robot. The controller can guide the robots to reach their assigned tasks and simultaneously avoid collisions with other robots and humans in the workspace.

REFERENCES

- Agrawal, A. and Sreenath, K. (2017). Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Proc. of Robotics: Science and Systems*.
- Ames, A.D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P. (2019). Control barrier functions: Theory and applications. In *Proc. of the European Control Conference*, 3420–3431.
- Ames, A.D., Xu, X., Grizzle, J.W., and Tabuada, P. (2016). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8), 3861–3876.
- Bai, H., Cai, S., Ye, N., Hsu, D., and Lee, W.S. (2015). Intention-aware online POMDP planning for autonomous driving in a crowd. In *Proc. of the IEEE International Conference on Robotics and Automation*, 454–460.
- Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press.
- Calinon, S. and Lee, D. (2017). Learning control. *Humanoid Robotics: A Reference*, 1261–1312.
- Christofides, P.D., Scattolini, R., de la Pena, D.M., and Liu, J. (2013). Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51, 21–41.
- De Luca, A. and Flacco, F. (2012). Integrated control for pMRI: Collision avoidance, detection, reaction and collaboration. In *Proc. of International Conference on Biomedical Robotics and Biomechatronics*, 288–295.
- Fishman, A., Paxton, C., Yang, W., Fox, D., Boots, B., and Ratliff, N. (2020). Collaborative interaction models for optimized human-robot teamwork. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 11221–11228.
- Gomez-Gonzalez, S., Neumann, G., Schölkopf, B., and Peters, J. (2020). Adaptation and robust learning of probabilistic movement primitives. *IEEE Transactions on Robotics*, 36(2), 366–379.
- Grandia, R., Taylor, A.J., Ames, A.D., and Hutter, M. (2020). Multi-layered safety for legged robots via control barrier functions and model predictive control. *arXiv preprint arXiv:2011.00032*.
- Kanazawa, A., Kinugawa, J., and Kosuge, K. (2019). Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. *IEEE Transactions on Robotics*, 35(4), 817–832.
- Kimmel, M. and Hirche, S. (2017). Invariance control for safe human-robot interaction in dynamic environments. *IEEE Transactions on Robotics*, 33(6), 1327–1342.
- Lazaric, A. and Ghavamzadeh, M. (2010). Bayesian multi-task reinforcement learning. In *Proc. of the International Conference on Machine Learning*, 599–606.
- Magrini, E., Ferraguti, F., Ronga, A.J., Pini, F., De Luca, A., and Leali, F. (2020). Human-robot coexistence and interaction in open industrial cells. *Robotics and Computer-Integrated Manufacturing*, 61, 101846.
- Nguyen, Q. and Sreenath, K. (2016). Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *Proc. of the American Control Conference*, 322–328.
- Paraschos, A., Daniel, C., Peters, J., and Neumann, G. (2018a). Using probabilistic movement primitives in robotics. *Autonomous Robots*, 42(3), 529–551.
- Paraschos, A., Rueckert, E., Peters, J., and Neumann, G. (2018b). Probabilistic movement primitives under unknown system dynamics. *Advanced Robotics*, 32(6), 297–310.
- Qin, S.J. and Badgwell, T.A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.
- Saveriano, M. and Lee, D. (2020). Learning barrier functions for constrained motion planning with dynamical systems. *arXiv preprint arXiv:2003.11500*.
- Sloth, C., Wisniewski, R., and Pappas, G.J. (2012). On the existence of compositional barrier certificates. In *Proc. of the IEEE Conference on Decision and Control*, 4580–4585.
- Srinivasan, M. and Coogan, S. (2020). Control of mobile robots using barrier functions under temporal logic specifications. *IEEE Transactions on Robotics*, 363–374.
- Wächter, A. (2009). Short tutorial: getting started with ipopt in 90 minutes. In *Proc. of the Dagstuhl Seminar*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
- Wieland, P. and Allgöwer, F. (2007). Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12), 462–467.
- Wiest, J., Höfken, M., Kreßel, U., and Dietmayer, K. (2012). Probabilistic trajectory prediction with gaussian mixture models. In *Proc. of the IEEE Intelligent Vehicles Symposium*, 141–146.
- Zeng, J., Zhang, B., and Sreenath, K. (2020). Safety-critical model predictive control with discrete-time control barrier function. *arXiv preprint arXiv:2007.11718*.
- Zhou, T. and Wachs, J.P. (2018). Early prediction for physical human robot collaboration in the operating room. *Autonomous Robots*, 42(5), 977–995.