Human-Robot Co-Transportation with Human Uncertainty-Aware MPC and Pose Optimization

Al Jaber Mahmud, Amir Hossain Raj, Duc M. Nguyen, Xuesu Xiao, and Xuan Wang

***Abstract*— This paper proposes a new control algorithm for human-robot co-transportation based on a robot manipulator equipped with a mobile base and a robotic arm. The primary focus is to adapt to human uncertainties through the robot's whole-body dynamics and pose optimization. We introduce an augmented Model Predictive Control (MPC) formulation that explicitly models human uncertainties and contains extra variables than regular MPC to optimize the pose of the robotic arm. The core of our methodology involves a two-step iterative design: At each planning horizon, we select the best pose of the robotic arm (joint angle combination) from a candidate set, aiming to achieve the lowest estimated control cost. This selection is based on solving an uncertainty-aware Discrete Algebraic Ricatti Equation (DARE), which also informs the optimal control inputs for both the mobile base and the robotic arm. To validate the effectiveness of the proposed approach, we provide theoretical derivation for the uncertainty-aware DARE and perform simulated and proof-of-concept hardware experiments using a Fetch robot under varying conditions, including different nominal trajectories and noise levels. The results reveal that our proposed approach outperforms baseline algorithms, maintaining similar execution time with that do not consider human uncertainty or do not perform pose optimization.**

I. Introduction

Collaborative human-robot systems can significantly reduce human workloads (Fig. 1). The capability of autonomous robots to adapt to human uncertainties is the key to determining system operational efficiency and safety [1], [2]. One frequently encountered task in engineering settings is object transportation [3]. To employ a human and a mobile manipulator to perform co-transportation, the key challenges arise from the uncertainties of human behaviors [4], which may not adhere strictly to predefined trajectories, and from the increased control complexity due to the coupling of the robotic arm and its mobile base [5].

To address these challenges, this paper formulates and solves a human uncertainty-aware Model Predictive Control (MPC) tracking problem. Its goal is to derive optimal control strategies by using robots' whole-body dynamics and augmented with pose optimization. Unlike most existing uncertainty-aware MPC approaches that consider the source of uncertainty to be from the robot dynamics [6], [7], we explicitly model and consider human uncertainties in MPC tracking problems. This approach allows us to estimate their impact on costs in terms of tracking errors and energy consumption when controlling the mobile base and robotic arm simultaneously. Building on this, pose optimization enables

George Mason University. {amahmud5, araj20, mnguy21, xiao, xwang64}@gmu.edu

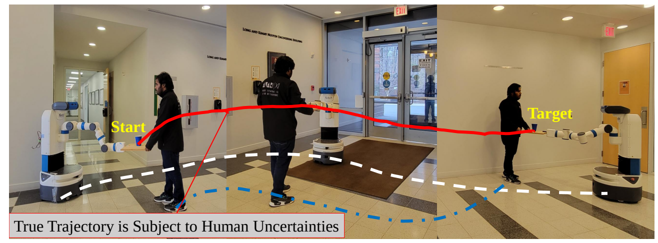


Fig. 1: Human robot co-transportation. The robot needs to adapt to human uncertainties added to a nominal trajectory. The board must be horizontal throughout the trajectory, guaranteeing the carried object's stability.

the robot to dynamically adjust its joint angles to better compensate for uncertainties and reduce predicted costs.

Statement of contribution: We study the collaborative transportation of objects between a human and a robot manipulator composed of a mobile base and a robotic arm. Our contributions in this paper are threefold. Firstly, we introduce a novel modeling approach that incorporates human uncertainties along with the robot's whole-body dynamics, leading to the development of a unique human-uncertainty-aware MPC tracking problem that includes pose optimization. Secondly, we propose a dual-phase optimization strategy. This strategy begins with calculating the estimated control costs in the presence of human uncertainties within a certain planning horizon, followed by optimization of the robot's pose through selection from a set of joint angle combinations. Lastly, we demonstrate the effectiveness of our method through a combination of theoretical derivation, simulated experiments, and a proof of concept hardware demo using a Fetch robot. We use quantitative comparisons to showcase the advantages of the approach over existing algorithms that either overlook human uncertainties or neglect pose optimization.

II. Literature review

There has been considerable existing literature investigating trajectory tracking using a mobile robot or the end effector of a robotic arm. For mobile robots, diverse algorithms such as MPC [8], learning-based nonlinear MPC [9], [10], sliding mode control [11], and adaptive planning can be employed. These works highlight the inherent challenges of real-time maneuvering in complex environments, such as terrain contact [11] and high-speed mobility [9]. On the other hand, tracking using a robotic arm features algorithms like Gaussian process-based MPC [12], adaptive time delay control [13], and model predictive path-following control [14]. These works focus on the high degree of freedom associated with the mechanical and kinematics characteristics of robotic arms, and address control problems under complex task configurations and workspace constraints [15], [16]. When combining a robotic arm with a mobile base, only a handful of studies have rigorously addressed their comprehensive whole-body dynamics control. These studies employ methodologies such as linear programming [17], constrained sequential linear quadratic control [18], end-to-end reinforcement learning [19], and nonlinear MPC [20]. These holistic control mechanisms can significantly expand the operational workspace of the mobile manipulator. Nonetheless, they often do not consider external disturbances, particularly the uncertainties introduced by humans in collaborative tasks.

As part of end-effector control, pose optimization has been extensively studied for robotic arm manipulation [21]. Instead of simply using forward dynamics to control the arm, pose optimization considers alternative joint angle combinations that can achieve the same end-effector position. This can help to avoid singularities [22], improve reachability in constrained spaces [23], and enhance control accuracy [24]. These pose optimization approaches are often used to overcome static or dynamic environmental constraints. The efforts to integrate pose optimization into co-manipulation problems are limited [25], and, similar to the tracking problem, they are not usually designed to better compensate for human uncertainties.

Adapting robot responses to align with human uncertainties during collaborative tasks is necessary and presents challenges in terms of control efficiency and safety. For this purpose, the integration of reinforcement learning and model-based control has been substantially used, such as using a robotic arm to assist humans in specific target activities through model-based reinforcement learning [26], wood sawing and surface polishing [27], and engaging in collaborative assembly tasks using Gaussian Process MPC [28]. Some studies have considered the whole-body dynamics of the mobile manipulator during co-manipulation or transportation tasks [29], [30]. While reinforcement learning shows effectiveness in handling unmodeled human uncertainties, it usually lacks transparency to theoretically ensure performance guarantees. To address this, robust MPC provides a control theoretic approach to address uncertainties in MPC problems, including the consideration of strict safety [31] and physical constraints [32], and systems with varying parameters [33]. However, these works mainly focus on uncertainties embedded in robot dynamics [6], [7], rather than those arising from humans. In addition, they do not consider integrating the MPC formulation with pose optimization to further improve control performance, which is a key difference from the problem considered in this paper.

III. Problem Statement and Formulation

In this section, we formulate the problem we aim to address. We start by introducing a trajectory with human disturbances, which the robot must follow and adapt to. Then, we detail the whole body dynamics of the mobile manipulator, composed of both the mobile base and the robotic arm. Lastly, we mathematically define an MPC tracking problem that is aware of human uncertainties and incorporates pose optimization.

Notations: Let I_r denote the $r \times r$ identity matrix. Let $\text{diag}\{a_1, a_2, \cdots, a_r\}$ denote a diagonal matrix with a_i being the ith diagonal entry. For a vector x, $|x|_2$ denotes its 2-norm. For a square matrix M, $\text{Tr}(M)$ denotes its trace. We use $M \succ 0$, $M \succeq 0$ to denote the matrix is positive definite and positive semi-definite, respectively. We let $\|x\|_M^2 = x^\top M x$ with $M \succeq 0$, which represents a quadratic evaluation of the vector based on matrix M.

A. Nominal Trajectory and Human Uncertainty

As illustrated in Figure 1, the task is to enable a team comprising a mobile manipulator and a human to collaboratively transport a board, adhering to a nominal trajectory. This trajectory defines the desired position and orientation of the robots' end-effector in the inertial frame at each time step t, expressed as $r_t = [r_t^x, r_t^y, r_t^z, r_t^\alpha, r_t^\beta, r_t^\gamma]^\top \in \mathbb{R}^6$ for $t \in \{0, 1, 2, \cdots, T\}$. Here, r_t^x, r_t^y, and r_t^z represent the end-effector's three-dimensional position, while r_t^α, r_t^β, and r_t^γ represent its orientation in terms of roll, pitch, and yaw, respectively. We assume the board must maintain a horizontal orientation throughout the cooperative transportation, i.e., $\forall t$, $r_t^\beta = r_t^\gamma = 0$.

Suppose human operators attempt to work with robots to transport the board following the nominal trajectory, but human actions are subject to uncertainties. As a result, the robot must dynamically adjust its movements to compensate for these disturbances to keep the board horizontal. To tackle this challenge, we implement a strategy based on receding horizon tracking, which allows the robot to adapt to human disturbances continuously [34]. At each control cycle, the robot aims to track a segment of the trajectory for future H steps, with each step defined by: $k \in \{0, 1, \cdots, H\}$,

$$\tilde{r}(k) = r(k) + D(h + \sum_{\tau=0}^{k} \varpi(\tau)), \tag{1}$$

with $\varpi(\tau) \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Here, $D = \begin{bmatrix} I_3 & \mathbf{0}_{3\times3} \end{bmatrix}^\top \in \mathbb{R}^{6\times3}$ is a matrix that maps human disturbances onto the nominal trajectory. We assume that the human only causes positional disturbances, without affecting the desired roll, pitch, and yaw of the reference trajectory. Therefore, the last three rows of D are kept as zeros. The vector $h = [h^x, h^y, h^z]^\top \in \mathbb{R}^3$ represents the positional disturbance created by the human up to the current time in the real world, which can be directly observed by the robot and should be added to the nominal trajectory. The term $\varpi \in \mathbb{R}^3$ represents the predicted human positional disturbance for future time steps. We assume the distribution of ϖ is zero mean and follows a covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ in x, y, z directions. The Σ is presumed to encapsulate individual human variations and is assumed to be known a priori. Equation (1) formulates a human-disturbance-aware trajectory. To track this trajectory in a receding horizon manner, we introduce the whole-body dynamics of the mobile manipulator as follows.

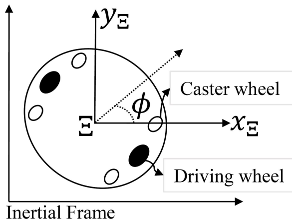


Fig. 2: Fetch robot mobile base in the inertial frame. A shifted frame Ξ based on current robot position.

B. Dynamics of a Mobile Manipulator.

We use a Fetch robot to present modeling details. However, a similar mechanism is generalizable to a wide class of mobile manipulator platforms.

Mobile Base. As shown in Fig. 2, the base of the Fetch mobile robot is equipped with four caster wheels and two driving wheels. Its discrete-time motion dynamics in the inertial frame can be represented by a differential drive model:

$$s_{base}(k+1) = s_{base}(k) + \tau \begin{bmatrix} cos(\phi(k)) & 0 \\ sin(\phi(k)) & 0 \\ 0 & 1 \end{bmatrix} u_{base}(k) \quad (2)$$

where $s_{base} = [x_{base},\ y_{base},\ \phi]^\top \in \mathbb{R}^3$ represents the x, y positions and the heading angle of the robot base, both in the in the inertial frame; $u_{base} = [v\ \eta]^\top$ represents the linear and angular velocities of the mobile base in its own body frame. τ is the discretization time interval.

Robotic Arm. The Fetch robot has a 7-DOF [35] robotic arm built on its mobile base. For the ease of combining the dynamics of the base and robotic arm, which will be discussed in the next subsection, we consider the robot base heading angle ϕ as one extra freedom for the robotic arm. This leads to an 8-DOF shown in Fig. 3 [36]. This definition allows us to represent the end-effector pose in the Ξ coordinate frame visualized in Fig. 2. Ξ has the same orientation as the inertial frame, thus, the two frames can be transformed without rotation. Furthermore, this definition also makes it easier for us to incorporate angle ϕ into pose optimization, together with 7 other joint angles.

We represent the end-effector pose of the robotic arm in Ξ frame by $s_{\text{arm}} = [{p_{\text{arm}}}^\top\ {\psi_{\text{arm}}}^\top] \in \mathbb{R}^6$, where $p_{\text{arm}} \in \mathbb{R}^3$ denotes the end-effector position in Cartesian coordinates, and $\psi_{\text{arm}} \in \mathbb{R}^3$ denotes the end-effector orientation in Euler angles. Note that our use of Euler angles is primarily for simplicity, although we are aware that Euler angles may potentially lead to singularities [37]. In our application, the target end-effector orientation is always horizontal to the ground, i.e., $r_t^\beta = r_t^\gamma = 0$, which significantly mitigates this issue. Furthermore, while alternative orientation representation methods like quaternions or $\mathbb{SO}(3)$ could be considered with complex dynamics matrices, we claim that our main results remain applicable if these alternative equations are substituted into the formulations in Section IV.

To study the state transition of s_{arm}, we represent its forward kinematics equation $f(\cdot)$ using the Jacobian matrix

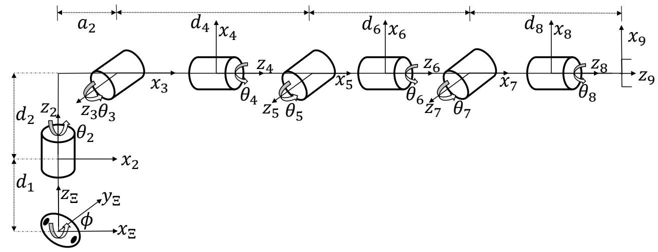


Fig. 3: Joint axes configurations of the 8-DOF robotic arm with DH parameters in frame Ξ.

$\mathbf{J}(\cdot) \in \mathbb{R}^{6\times 8}$ derived based on the DH-parameters [38] of the robotic arm configured in Fig. 3:

$$\begin{aligned} s_{\text{arm}}(k+1) &= f(\theta(k) + \tau\omega(k)) \\ &\approx s_{\text{arm}}(k) + \tau \mathbf{J}(\theta(k))\omega(k) \end{aligned} \quad (3)$$

where $\theta = [\phi,\ \theta_2,\ \theta_3, \cdots, \theta_8]^\top \in \mathbb{R}^8$ represents the joint angles including the mobile base heading angle ϕ and the seven robotic arm angles; and $\omega = \dot{\theta} = [\eta,\ \dot{\theta}_2,\ \dot{\theta}_3, \cdots,\ \dot{\theta}_8]^\top \in \mathbb{R}^8$ represents the corresponding angular velocities. The Jacobian matrix can be computed by $\mathbf{J}(\theta(k)) = \frac{\partial f(\theta)}{\partial \theta}\Big|_{\theta=\theta(k)}$.

Whole-Body Dynamics. We combine the dynamics for the base (Equation (2)) and for the robotic arm (Equation (3)) to obtain the linearized whole-body dynamics of the robot's end-effector pose in the inertial frame as $s \in \mathbb{R}^6$. It can be represented by

$$s = s_{\text{arm}} + \begin{bmatrix} x_{\text{base}} \\ y_{\text{base}} \\ 0_{4\times 1} \end{bmatrix},$$

where the two states can be directly added because s_{base} is defined in the inertial frame, s_{arm} is defined in the Ξ frame, and no rotation is needed for the transition between the two frames. Consequently, the state update is given by

$$s(k+1) = s(k) + B(\theta(k))u(k) \quad (4)$$

with

$$B(\theta(k)) = \tau \begin{bmatrix} cos(\phi(k)) & \\ sin(\phi(k)) & \mathbf{J}(\theta(k)) \\ 0_{4\times 1} & \end{bmatrix}$$

where $u = [v,\ \omega^\top]^\top \in \mathbb{R}^9$ is the control input combining the linear velocity of the base and all rotations of the robot. $B(\theta(k)) \in \mathbb{R}^{6\times 9}$ is the input matrix which depends on the joint angle combinations $\theta(k)$. The first column of $B(\theta(k))$ is derived from the linearized motion dynamics of the mobile base as in (2) and Jacobian matrix comes from dynamics of the 8-DOF arm model as (3). The second column of (2) is not used since it has been integrated into the Jacobian matrix.

C. Human Uncertainty Aware MPC with Pose Optimization

Our research problem focuses on effectively tracking a specified trajectory, denoted as equation (1), subject to the dynamics of the whole body dynamics (4). We solve this by introducing a special MPC formulation. Unlike regular MPC, which simply minimizes the objective function to obtain the optimal control input sequence within the planning horizon while tracking the trajectory, our formulation introduces two enhancements. First, we consider human uncertainty ϖ embedded in trajectory (1). We estimate its impact on the cost when optimizing the control strategies. Unlike the robust MPC in the literature, which considers uncertainties from the robot dynamics, our formulation considers external uncertainties coming from the human. Second, we consider pose optimization. In addition to the regular control inputs $u(k)$, we allow the robot to change its joint angle combination from θ_0 to a new combination $\bar{\theta}$ selected from a candidate set, if the new $\bar{\theta}$ leads to a lower predicted cost. The rationale behind this is that given the same end-effector pose, an 8-DOF robot can have infinitely many feasible joint angle combinations. If the robot is informed by the future trajectory and the human uncertainty distribution, it can choose a better $\bar{\theta}$, i.e., a more desired pose, which induces less future cost.

The discrete-time human uncertainty-aware MPC tracking with pose optimization can be formulated as follows:

$$\begin{aligned}
&\min_{\boldsymbol{u}(0:H-1),\bar{\theta}\in\Theta} \mathcal{J}(\boldsymbol{u}(0:H-1),\bar{\theta}) \\
&\triangleq \mathbb{E}_{\varpi}\left[\sum_{k=0}^{H}\left[s(k)-\tilde{r}(k)\right]^{\top} Q\left[s(k)-\tilde{r}(k)\right]\right] \\
&\quad + \sum_{k=0}^{H-1} u(k)^{\top} R u(k) + \kappa|\bar{\theta}-\theta_0|_2^2 \qquad (5)\\
&\text{s.t.}\quad s(k+1)=s(k)+B(\bar{\theta})u(k), \quad s(0)=s_0 \\
&\qquad \tilde{r}(k)=r(k)+D(h+\sum_{\tau=0}^{k}\varpi(\tau)), \quad \varpi\sim\mathcal{N}(\mathbf{0},\Sigma)
\end{aligned}$$

where $\boldsymbol{u}(0:H-1)=\{u(0),\cdots,u(H-1)\}$, $Q\in\mathbb{R}^{6\times 6}\succeq 0$, $R\in\mathbb{R}^{9\times 9}\succ 0$ are the weighting matrices for tracking and input costs, respectively; $\kappa\in\mathbb{R}_{+}$ is the cost weight for pose optimization. s_0 is the current end-effector pose to initialize each planning phase, and θ_0 is the current joint angle combination. For tractability, similar to prior works [39], [40], we consider the linearized system with a fixed $B(\cdot)$ matrix for the robot's end-effector dynamics throughout the entire planning horizon. The impact is small when the planning horizon is short [40]. Finally, we note that without pose optimization, i.e., $\bar{\theta}=\theta_0$ and ignoring human disturbance, i.e., $\Sigma=0$, then problem (5) degrades to a regular MPC tracking problem.

IV. Main Result

To solve the problem formulated in Equation (5), we observe that the optimal control input sequence, $\boldsymbol{u}(0:H-1)$, is influenced by the input matrix $B(\bar{\theta})$, which means it also relies on the robot's pose optimization variable, $\bar{\theta}$, as in Equation (4). Since there are no closed-form solutions for $\boldsymbol{u}(0:H-1)$ for general MPC tracking problems, optimizing both $\boldsymbol{u}(0:H-1)$ and $\bar{\theta}$ at the same time presents a challenge [41]. Our approach to address this is a dual-phase method. First, we generate a set of candidate joint angle combinations or $\bar{\theta}$ values. For each $\bar{\theta}$, we theoretically compute optimal control inputs, $\boldsymbol{u}(0:H-1)$, within the planning horizon and estimate the cost-to-go associated with it, considering uncertainties caused by humans, referred to as ϖ. Then, in the second step, we go through candidate joint angle combinations and choose the one that resulted in the lowest estimated cost. This will combine the best of both tracking cost optimization and pose optimization to find the most efficient $\bar{\theta}$ and $\boldsymbol{u}(0:H-1)$.

Following this approach, we start by presenting the result to solve the optimal control input sequence $\boldsymbol{u}(0:H-1)$ and the optimal cost $\mathcal{J}^*$ with a fixed $\bar{\theta}$. For presentation simplicity, let $\bar{B}=B(\bar{\theta})$, and define the following error dynamics for (4) by subtracting $\tilde{r}(k+1)$ from both sides of the equation:

$$\begin{aligned}
e(k+1) &= e(k)+\bar{B}u(k)+\tilde{r}(k)-\tilde{r}(k+1) \qquad (6)\\
&= e(k)+\bar{B}u(k)+r(k)-r(k+1)-D\varpi(k+1)
\end{aligned}$$

with $e(k)=s(k)-\tilde{r}(k)$ being the tracking error. We hypothesize that the optimal cost-to-go function follows:

$$\mathcal{J}^*(e(k),k)=\|e(k)\|_{P(k)}^2+2e(k)^{\top}p(k)+c(k) \qquad (7)$$

where, $P(k)\in\mathbb{R}^{6\times 6}, p(k)\in\mathbb{R}^{6}, c(k)\in\mathbb{R}$, are unknown matrices, vectors, and scalars to be determined. The following result shows that the assumed solution form is valid, and the parameters can be computed from a Discrete Algebraic Ricatti Equation (DARE) [42]. The proof of the theorem is given in the Appendix.

Theorem 1. *Given a $\bar{B}$, assuming the optimal solution $\boldsymbol{u}^*$ of (5) yields an optimal cost $\mathcal{J}^*$ with the form of (7). Then $P(k)$, $p(k)$, and $c(k)$ can be computed by the following uncertain-aware DARE:*

$$P(k)=Q+P(k+1)-P(k+1)\bar{B}MP(k+1) \qquad (8a)$$

$$\begin{aligned}
p(k) &= p(k+1)+P(k+1)(r(k)-r(k+1)) \\
&\quad - P(k+1)\bar{B}MP(k+1)(r(k)-r(k+1)) \\
&\quad - P(k+1)\bar{B}Mp(k+1) \qquad (8b)
\end{aligned}$$

$$\begin{aligned}
c(k) &= c(k+1)+\|r(k)-r(k+1)\|_{P(k+1)}^2 \\
&\quad + Tr(\Sigma D^{\top}P(k+1)D)-\|P(k+1)(r(k) \\
&\quad - r(k+1))+p(k+1)\|_{BM}^2+2(r(k) \\
&\quad - r(k+1))^{\top}p(k+1) \qquad (8c)
\end{aligned}$$

with $M=(R+\bar{B}^{\top}P(k+1)\bar{B})^{-1}\bar{B}^{\top}$, terminal conditions:

$$P(k=H)=Q, \quad p(k=H)=0, \quad c(k=H)=\kappa|\bar{\theta}-\theta_0|_2^2$$

The corresponding control input

$$u^*(k)=-M(P(k+1)(e(k)+r(k)-r(k+1))+p(k+1)) \qquad (9)$$

gives the cost in (7) with parameters in (8). □

From Theorem 1 and (9), it can be observed that the optimal control input $u^*(k)$ does not depend on $c(k)$. However, $c(k)$ contributes to the computation of optimal cost $\mathcal{J}^*$, which impacts the pose selection among the candidates. More specifically, our two-step solver is summarized in

Algorithm 1: Human-Uncertainty-Aware MPC Tracking with Pose Optimization

1 **Input** Nominal trajectory r_t; human uncertainty covariance matrix Σ; current accumulated human disturbance h; current joint angles θ_0.
2 Formulate $\tilde{r}(k=0:H)$ based on equation (1).
3 Create the joint angles candidate set Θ by sampling around θ_0, also add θ_0 to Θ.
4 **for** *each* $\bar{\theta} \in \Theta$ **do**
5   Compute Jacobian matrix, $\mathbf{J}(\bar{\theta}) = \frac{\partial f(\theta)}{\partial \theta}\Big|_{\theta=\bar{\theta}}$.
6   Compute matrix $\bar{B} = B(\bar{\theta})$ with (4).
7   Solve the MPC by computing solutions for the uncertain-aware DARE in *Theorem 1*.
8   Compute the optimal cost $\mathcal{J}^*$ associated with the current $\bar{\theta}$ using (7).
9 **end**
10 Compare the costs $\mathcal{J}^*$ for all candidate $\bar{\theta}$ and find the optimal $\bar{\theta}^*$ as pose optimization.
11 Apply the pose optimization to the robot.
12 Reuse the results in step 7 for the selected $\bar{\theta}^*$ and equation (9) to generate optimal control input sequence $\boldsymbol{u}^*(0:H-1)$, and apply them to the robot.

Algorithm 1. At the beginning of each MPC horizon, we first create the joint angles candidate set Θ by randomly changing multiple joint angles of θ_0 with a small radian value. This should lead to small pose optimization cost $\kappa|\bar{\theta}-\theta_0|_2^2$ and small changes to the end-effector pose of the robot. For every $\bar{\theta} \in \Theta$, we compute $\bar{B}$ that defines the control input matrix of the system dynamics. Then, we use Theorem 1 to compute the estimated optimal cost $\mathcal{J}^*$ for the current $\bar{\theta}$. This process is parallelizable to improve computational efficiency. By exploiting all $\bar{\theta} \in \Theta$ and the associated $\mathcal{J}^*$, we select the best $\bar{\theta}$ and use (9) to obtain the associated control inputs. Finally, we apply both the pose optimization and the control inputs to the robot.

Remark 1. *In general, the highly non-linear relation between $B(\bar{\theta})$ and $\bar{\theta}$ makes it computationally infeasible to systematically find the optimal pose $\bar{\theta}$ for estimated control cost $\mathcal{J}^*$. Instead, in Algorithm 1, we employ a sample-based approach to select a candidate set for pose optimization. This allows us to numerically search for a pose $\bar{\theta}$ that is better than θ_0 in terms of future control cost. A similar technique has been used in [43]. Here, increasing the cardinality of Θ can potentially lead to a better $\bar{\theta}^*$, but also incurs more computation. As previously mentioned, since the evaluation of $\bar{\theta}$ can be performed in parallel, the cardinality of Θ can be chosen based on the robot's local computational power. Furthermore, as we will demonstrate in the experiments, if computation resources are limited, pose optimization does not necessarily have to be performed at every step. Optimizing poses periodically over fixed intervals can also improve tracking performance. Lastly, to make the pose optimizing more efficient, one may leverage deep learning methods to determine when a pose optimization is needed [44], and how a candidate set should be chosen [45].*

V. Experiments

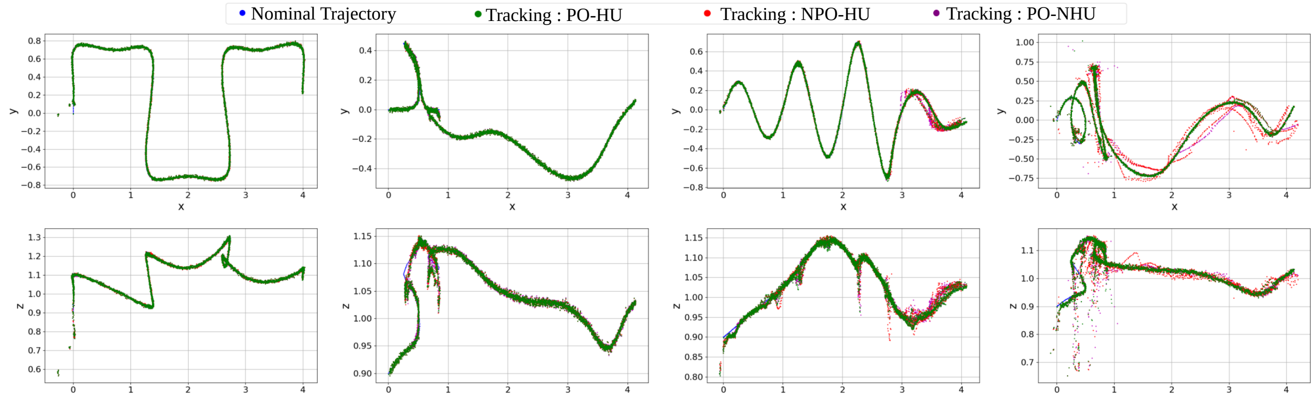
In this section, we evaluate our proposed Human-Uncertainty-Aware MPC Tracking with Pose Optimization algorithm through simulation experiments in Gazebo and perform proof of concept demonstration using a real fetch robot. We build the model of a Fetch robot using methods described in Sec. III with DH parameters [46] for specifying link lengths, offsets, and twist angles of rotation. The creation of the robot in the Gazebo environment is based on its URDF (Unified Robot Description Format) [46]. To simulate human disturbances, we assume $\varpi(k) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where $\Sigma = q \cdot \mathrm{diag}(0.015, 0.025, 0.015)$ (meters) and $q \in \{0.4,\ 0.7\}$. This reflects the tendency for disturbances to be more pronounced along the y-axis compared to the x and z-axes, with q being the strength of these disturbances. Despite our problem-solving approach being based on the linearized dynamics of the system, during the simulations, we continuously update the robot's state using real physics and determine the end-effector's position through forward kinematics calculations.

We test our algorithm using four different nominal trajectories, denoted as $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$ visualized in Fig. 4(a-d), respectively. Each trajectory is discretized into 500 discrete points, with a time interval of $\tau = 0.1$ seconds. We assume the robot and the human positions are initialized by holding a board horizontally. The parameters used for solving the problem (5), are selected to be $H = 8$, $R = I_9$, and $\kappa = 1$. We experiment with two different settings for the Q parameter, choosing either $Q = 1000 \cdot I_6$ or $Q = 500 \cdot I_6$, to reweigh the importance of tracking error on the overall cost. These parameter variations help to validate our proposed algorithm's performance under different conditions over other baseline algorithms.

Our experiments follow Algorithm 1, we repeat the MPC planning every step using the current end-effector pose as the initial state, and apply the first control signal in the planned trajectory to actuate the system. In each planning horizon, we chose twelve candidate poses in Θ. While the proposed MPC-based algorithm seeks to minimize an expected cost over a horizon H, we define the true system cost over the entire trajectory as:

$$\mathcal{C}_{\text{total}} = \sum_{t=1}^{T} e(t)^\top Q e(t) + u(t)^\top R u(t) + \kappa|\bar{\theta}(t) - \theta(t)|_2^2,$$

which takes into account the costs for the robot's end effector tracking error, control input, and pose optimization, T is the total number of time steps. We compare our proposed approach (PO-HU: considering pose optimization and human uncertainty) with two baselines: one approach with No Pose Optimization but considering Human Uncertainty (NPO-HU) and another one with Pose Optimization but Not considering Human Uncertainty (PO-NHU). Note that we do not need to evaluate the no pose optimization and no human uncertainty



(a) Trajectory $\mathcal{A}$ (b) Trajectory $\mathcal{B}$ (c) Trajectory $\mathcal{C}$ (d) Trajectory $\mathcal{D}$

Fig. 4: All trajectories with human disturbances, $Q = 1000 \cdot I_6$, $q = 0.4$. Tracking performance is represented by dot clouds over 10 trials.

(NPO-NHU) case, as it is equivalent to NPO-HU. This equivalence arises because human uncertainty with zero mean does not affect the control inputs (but only the expected control cost), as shown in Equation (9).

The dot clouds in Fig.4 illustrate the comparative tracking performance across four distinct trajectories on the x-y and x-z planes, with the weighting matrix $Q = 1000 \cdot I_6$ and $q = 0.4$. A numerical comparison is further provided in Table I, which shows the average total cost $\mathcal{C}_{\text{total}}$ of different algorithms over 10 trials. Additionally, different from PO-HU, which performs pose optimization at every time step, we also introduce a periodic pose optimization (pPO-HU) that performs pose optimization every 5 time steps. This helps to reduce the computational burden when applied to low-cost devices. In terms of hardware implementation, due to the lack of global localization, we haven't yet finished the complete hardware implementation. However, for proof of concept purposes, all the trajectories computed from Gazebo visualized in Fig. 4, have been executed and successfully reproduced on our hardware platform to justify their feasibility. The complete and independent hardware implementation is our direct future work.

TABLE I: Comparison of $\mathcal{C}_{\text{total}}$ across Different Algorithms

Traj	Q	q	PO-HU	pPO-HU	NPO-HU	PO-NHU
$\mathcal{A}$	$1000I_6$	0.4	**1019.81**	1023.15	1119.29	1057.35
		0.7	**1325.96**	1337.11	1431.20	1370.91
	$500I_6$	0.4	**856.87**	863.12	1075.43	914.19
		0.7	**1027.91**	1032.42	1160.69	1070.57
$\mathcal{B}$	$1000I_6$	0.4	**970.99**	972.71	1189.59	1231.87
		0.7	**1161.31**	1378.59	1520.03	1500.54
	$500I_6$	0.4	**856.07**	927.54	5875.60†	2123.53‡
		0.7	**906.92**	1140.84	1938.88 †	2515.04‡
$\mathcal{C}$	$1000I_6$	0.4	**686.94**	769.55	1932.83†	1102.95
		0.7	**1151.33**	1292.57	2342.11†	2384.16‡
	$500I_6$	0.4	**829.59**	835.64	851.60	852.53
		0.7	**978.52**	986.46	1013.69	1017.27
$\mathcal{D}$	$1000I_6$	0.4	**2306.64**	2540.73	3759.09†	2968.25
		0.7	**3782.70**	3943.75	8694.57†	7220.29‡
	$500I_6$	0.4	**2013.31**	2041.27	2261.32	2209.01
		0.7	**2293.11**	2481.39	5612.76†	4301.46

It can be read from Fig. 4 that in all cases, the proposed approach (PO-HU) outperforms the other two in terms of tracking error. Especially for more complex trajectories $\mathcal{B}, \mathcal{C}, \mathcal{D}$, the result of PO-HU closely follows the reference trajectories, whereas the other two baseline algorithms deviate a lot. This verifies the effectiveness of the proposed algorithm. Regarding the total cost, the advantage of the proposed algorithm is justified by the results in Table I. Specifically, when comparing the columns, we use † in NPO-HU, and ‡ in PO-NHU to highlight the entries where the differences are significant. The fewer highlighted entries in PO-NHU imply that pose optimization has a greater impact on the total cost than the characterization of human uncertainty. Additionally, when performing pose optimization every five time steps (pPO-HU), it performs better than the other two, although having a small gap compared to performing pose optimization at every time step (PO-HU). We also observe that the cost associated with trajectory $\mathcal{D}$ is higher than that for other trajectories. This can be attributed to the inherent complexity and the sharp turns in trajectory $\mathcal{D}$, as depicted in Fig. 4.

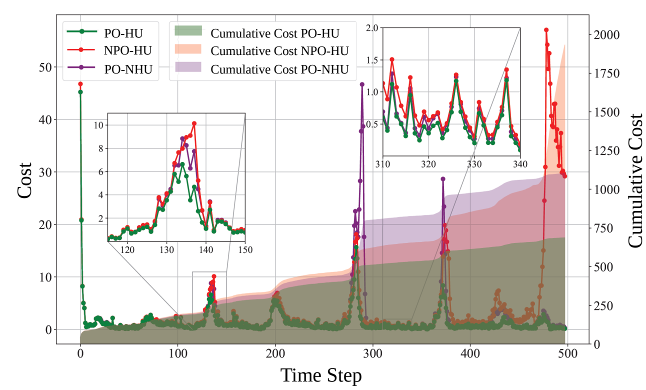


Fig. 5: Accumulated cost comparison for trajectory $\mathcal{C}$, with $Q = 1000 \cdot I_6$, $R = I_9$, $q = 0.4$.

Figure 5 visualizes the accumulated cost over time and the cost at each time step for trajectory $\mathcal{C}$, with $Q = 1000 \cdot I_6$

and $q = 0.4$. A comparison of cost and peaks reveals the effectiveness of our algorithm in navigating complex trajectory segments, particularly during sharp turns. Furthermore, it is observed that towards the end of the trajectory, the costs associated with NPO-HU explode because the robot admits a bad pose and can hardly reach and track the remaining trajectory. In contrast, the costs for the methods incorporating Pose Optimization, namely PO-HU and PO-NHU, remain relatively stable.

We use Fig. 6 to record the average execution time of the proposed algorithm for different planning horizons, averaging over 100 trials. Steps (4-9) of the proposed algorithm can be fully parallelized, thus the computation time for pose optimization only increases slightly compared to the case without pose optimization, as long as the size of the set $|\Theta| = 12$ is smaller than the number of computing threads. (The test computer uses an AMD 5975XW.) Furthermore, we observe that the execution time for each algorithm increases only mildly across different values of H. The main computation time is spent on the one-time computation of the Jacobian matrix. The minor increase in execution time is due to more iterations in solving the DARE (cf. equation (8)). This concludes that our proposed PO-HU method does not introduce significant extra execution time compared with other baseline algorithms. This further justifies the feasibility of the complete hardware real-time implementation in our future work.

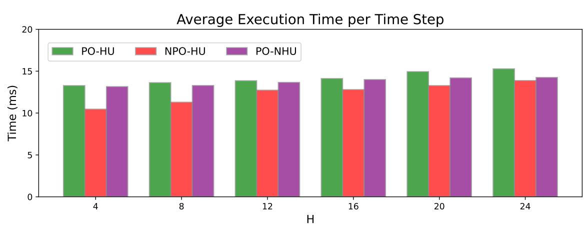


Fig. 6: Execution time comparison with different planning horizon, with $Q = 1000 \cdot I_6$, $R = I_9$, $q = 0.4$, $|\Theta| = 12$.

VI. Conclusions and Future Works

We studied the control of a mobile manipulator to perform human-robot co-transportation tasks. By modeling human uncertainties and the whole-body dynamics of the robot, we formulated a new human-uncertainty-aware MPC tracking problem with pose optimization. The key challenge arose from the need to simultaneously optimize the joint angle combination for pose optimization and the control inputs to minimize the cost. To address this, we proposed an algorithm with a two-step iterative design, equipped with an inner loop that computes an uncertainty-aware DARE to estimate the control cost, and an outer loop that selects the best pose with the minimum cost from a candidate set. The correctness and effectiveness of the proposed approach have been validated through both theoretical derivation and simulated experiments, respectively. Our simulation used a Fetch robot to perform co-transportation tasks under varying conditions. The results showed that the proposed approach has advantages in terms of tracking accuracy and energy consumption while maintaining similar execution time, over baseline algorithms. Future work will include the complete implementation of our algorithm on hardware and the generalization of the algorithm to multi-human multi-robot collaborative tasks.

Appendix

Proof of Theorem 1:

Integrating the error update (6) into the optimal cost (7), one has

$$
\begin{aligned}
&\mathcal{J}^{\star}(e(k),k) = \min_{u(k)} \mathcal{J}(e(k),k)) \\
&= \min_{u(k)} [\mathbb{E}_{\varpi}[\|e(k)\|_Q^2 + \|u(k)\|_R^2 + \mathcal{J}^{\star}(e(k+1)), k+1)]] \\
&= \min_{u(k)} [\mathbb{E}_{\varpi}[\|e(k)\|_Q^2 + \|u(k))\|_R^2 \\
&\quad + \|e(k) + \bar{B}u(k) + r(k) - r(k+1)\|_{P(k+1)}^2 \\
&\quad + \mathrm{Tr}(\Sigma D^{\top} P(k+1) D) + 2(e(k) + \bar{B}u(k) \\
&\quad + r(k) - r(k+1))^{\top} p(k+1) + c(k+1)]] \qquad (10)
\end{aligned}
$$

where the elimination of terms follows from $\mathbb{E}(\varpi(k+1)) = 0$. Since the control input minimizes cost at each time step, the optimality condition $\frac{\partial \mathcal{J}^{*}(e(k),k)}{\partial u(k)} = 0$ yields

$$
\begin{aligned}
Ru(k) + \bar{B}^{\top} P(k+1)(e(k) + \bar{B}u(k) + r(k) - r(k+1)) \\
+ \bar{B}^{\top} p(k+1) = 0 \qquad (11)
\end{aligned}
$$

Thus, for each time-step k, the control inputs follows:

$$
u^{*}(k) = -M(P(k+1)(e(k) + r(k) - r(k+1)) + p(k+1)) \qquad (12)
$$

with M defined in Theorem (1). Bring this back to (10) and reusing condition (11) by left multiplying $u(k)^{\top}$ yields

$$
\begin{aligned}
\mathcal{J}^{*}(e(k),k) = \ & \|e(k)\|^2_{[Q+P(k+1)-P(k+1)\bar{B}MP(k+1)]} \\
& + 2e(k)^{\top}\big[p(k+1) - P(k+1)\bar{B}(Mp(k+1) \\
& \quad - P(k+1)\bar{B}MP(k+1)(r(k) - r(k+1) \\
& \quad + P(k+1)(r(k) - r(k+1))\big] \\
& + c(k+1) + \|r(k) - r(k+1)\|_{P(k+1)} \\
& \quad + \mathrm{Tr}(\Sigma D^{\top} P(k+1) D) - \|P(k+1)(r(k) \\
& \quad - r(k+1)) + p(k+1))\|^2_{BM} + 2(r(k) \\
& \quad - r(k+1))^{\top} p(k+1) \qquad (13)
\end{aligned}
$$

Comparing (13) with (7), we have the DARE for $P(k)$, $p(k)$, and $c(k)$. The terminal conditions are obtained by considering $\mathcal{J}^{*}(e(H), k = H)$ for (5), where $P(H) = Q$ and $c(H)$ is the constant pose optimization cost. $\square$

References

[1] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," *learning*, vol. 8, no. 9, 2013.

[2] C. Brosque, E. Galbally, O. Khatib, and M. Fischer, "Human-robot collaboration in construction: Opportunities and challenges," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1–8, IEEE, 2020.

[3] A.-N. Sharkawy and P. N. Koustoumpardis, "Human–robot interaction: A review and analysis on variable admittance control, safety, and perspectives," *Machines*, vol. 10, no. 7, p. 591, 2022.

[4] S. Li, H. Wang, and S. Zhang, "Human-robot collaborative manipulation with the suppression of human-caused disturbance," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 4, p. 73, 2021.

[5] S. Erhart, D. Sieber, and S. Hirche, "An impedance-based control architecture for multi-robot cooperative dual-arm mobile manipulation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 315–322, IEEE, 2013.

[6] F. Cursi, V. Modugno, L. Lanari, G. Oriolo, and P. Kormushev, "Bayesian neural network modeling and hierarchical mpc for a tendon-driven surgical robot with uncertainty minimization," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2642–2649, 2021.

[7] G. P. Incremona, A. Ferrara, and L. Magni, "Mpc for robot manipulators with integral sliding modes generation," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 3, pp. 1299–1307, 2017.

[8] C. Wang, X. Liu, X. Yang, F. Hu, A. Jiang, and C. Yang, "Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy," *Applied Sciences*, vol. 8, no. 2, p. 231, 2018.

[9] X. Xiao, J. Biswas, and P. Stone, "Learning inverse kinodynamics for accurate high-speed off-road navigation on unstructured terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 6054–6060, 2021.

[10] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, and J. Collier, "Learning-based nonlinear model predictive control to improve vision-based mobile robot path tracking," *Journal of Field Robotics*, vol. 33, no. 1, pp. 133–152, 2016.

[11] H. Taghavifar and S. Rakheja, "A novel terramechanics-based path-tracking control of terrain-based wheeled robot vehicle with matched-mismatched uncertainties," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 67–77, 2019.

[12] A. Carron, E. Arcari, M. Wermelinger, L. Hewing, M. Hutter, and M. N. Zeilinger, "Data-driven model predictive control for trajectory tracking with a robotic arm," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3758–3765, 2019.

[13] J. Baek, S. Cho, and S. Han, "Practical time-delay control with adaptive gains for trajectory tracking of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5682–5692, 2017.

[14] T. Faulwasser, T. Weber, P. Zometa, and R. Findeisen, "Implementation of nonlinear model predictive path-following control for an industrial robot," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1505–1511, 2016.

[15] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3229–3236, Ieee, 2007.

[16] A. B. Clark and N. Rojas, "Design and workspace characterisation of malleable robots," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9021–9027, IEEE, 2020.

[17] J. J. Quiroz-Omaña and B. V. Adorno, "Whole-body kinematic control of nonholonomic mobile manipulators using linear programming," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 263–278, 2018.

[18] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient kinematic planning for mobile manipulators with non-holonomic constraints using optimal control," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3411–3417, IEEE, 2017.

[19] J. Kindle, F. Furrer, T. Novkovic, J. J. Chung, R. Siegwart, and J. Nieto, "Whole-body control of a mobile manipulator using end-to-end reinforcement learning," *arXiv preprint arXiv:2003.02637*, 2020.

[20] M. Osman, M. W. Mehrez, S. Yang, S. Jeon, and W. Melek, "End-effector stabilization of a 10-dof mobile manipulator using nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9772–9777, 2020.

[21] L. Stadelmann, T. Sandy, A. Thoma, and J. Buchli, "End-effector pose correction for versatile large-scale multi-robotic systems," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 546–553, 2019.

[22] L. Huo and L. Baron, "The joint-limits and singularity avoidance in robotic welding," *Industrial Robot: An International Journal*, vol. 35, no. 5, pp. 456–464, 2008.

[23] H. Cheong, M. Ebrahimi, and T. Duggan, "Optimal design of continuum robots with reachability constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3902–3909, 2021.

[24] A. Mohammed, B. Schmidt, L. Wang, and L. Gao, "Minimizing energy consumption for robot arm movement," *Procedia Cirp*, vol. 25, pp. 400–405, 2014.

[25] M. Soleimani Amiri and R. Ramli, "Intelligent trajectory tracking behavior of a multi-joint robotic arm via genetic–swarm optimization for the inverse kinematic solution," *Sensors*, vol. 21, no. 9, p. 3171, 2021.

[26] L. Roveda, J. Maskani, P. Franceschi, A. Abdi, F. Braghin, L. Molinari Tosatti, and N. Pedrocchi, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 2, pp. 417–433, 2020.

[27] L. Peternel, N. Tsagarakis, D. Caldwell, and A. Ajoudani, "Robot adaptation to human physical fatigue in human–robot co-manipulation," *Autonomous Robots*, vol. 42, pp. 1011–1021, 2018.

[28] K. Haninger, C. Hegeler, and L. Peternel, "Model predictive control with gaussian processes for flexible multi-modal physical human robot interaction," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 6948–6955, IEEE, 2022.

[29] D. Sirintuna, A. Giammarino, and A. Ajoudani, "An object deformation-agnostic framework for human–robot collaborative transportation," *IEEE Transactions on Automation Science and Engineering*, 2023.

[30] D. Sirintuna, I. Ozdamar, J. M. Gandarias, and A. Ajoudani, "Enhancing human-robot collaboration transportation through obstacle-aware vibrotactile feedback," *arXiv preprint arXiv:2302.02881*, 2023.

[31] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.

[32] M. B. Saltık, L. Özkan, J. H. Ludlage, S. Weiland, and P. M. Van den Hof, "An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects," *Journal of Process Control*, vol. 61, pp. 77–102, 2018.

[33] J. Mohammadpour and C. W. Scherer, *Control of linear parameter varying systems with applications*. Springer Science & Business Media, 2012.

[34] T. M. Howard, C. J. Green, and A. Kelly, "Receding horizon model-predictive control for mobile robot navigation of intricate paths," in *Field and Service Robotics: Results of the 7th International Conference*, pp. 69–78, Springer, 2010.

[35] K. Kreutz-Delgado, M. Long, and H. Seraji, "Kinematic analysis of 7-dof manipulators," *The International journal of robotics research*, vol. 11, no. 5, pp. 469–481, 1992.

[36] J. Sudharsan and L. Karunamoorthy, "Path planning and co-simulation control of 8 dof anthropomorphic robotic arm," *International journal of simulation modelling*, vol. 15, no. 2, pp. 302–312, 2016.

[37] R. Campa and H. De La Torre, "Pose control of robot manipulators using different orientation representations: A comparative review," in *2009 American Control Conference*, pp. 2855–2860, IEEE, 2009.

[38] S. Kucuk and Z. Bingul, *Robot kinematics: Forward and inverse kinematics*. INTECH Open Access Publisher London, UK, 2006.

[39] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Trajectory tracking of autonomous vessels using model predictive control," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 8812–8818, 2014.

[40] Q. Shi, J. Zhao, A. El Kamel, and I. Lopez-Juarez, "Mpc based vehicular trajectory planning in structured environment," *IEEE Access*, vol. 9, pp. 21998–22013, 2021.

[41] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Linear tracking mpc for nonlinear systems—part ii: The data-driven case," *IEEE Transactions on Automatic Control*, vol. 67, no. 9, pp. 4406–4421, 2022.

[42] A. Ran and R. Vreugdenhil, "Existence and comparison theorems for algebraic riccati equations for continuous-and discrete-time systems," *Linear Algebra and its applications*, vol. 99, pp. 63–83, 1988.

[43] J. Wang, T. Zhang, Y. Wang, and D. Luo, "Optimizing robot arm reaching ability with different joints functionality," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1778–1785, IEEE, 2023.

[44] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[45] T. Li, K. Sun, Y. Jin, and H. Liu, "A novel optimal calibration algorithm on a dexterous 6 dof serial robot-with the optimization of measurement poses number," in *2011 IEEE international conference on robotics and automation*, pp. 975–981, IEEE, 2011.

[46] F. R. Inc., "Fetch & freight manual," 2014.