

Human-Robot Co-Transportation using Disturbance-Aware MPC with Pose Optimization

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

Abstract—This paper proposes a new control algorithm for human-robot co-transportation using a robot manipulator equipped with a mobile base and a robotic arm. We integrate the regular Model Predictive Control (MPC) with a novel pose optimization mechanism to more efficiently mitigate disturbances (such as human behavioral uncertainties or robot actuation noise) during the task. The core of our methodology involves a two-step iterative design: At each planning horizon, we determine the optimal pose of the robotic arm (joint angle configuration) from a candidate set, aiming to achieve the lowest estimated control cost. This selection is based on solving a disturbance-aware Discrete Algebraic Riccati Equation (DARE), which also determines the optimal inputs for the robot’s whole body control (including both the mobile base and the robotic arm). To validate the effectiveness of the proposed approach, we provide theoretical derivation for the disturbance-aware DARE and perform simulated experiments and hardware demos using a Fetch robot under varying conditions, including different trajectories and different levels of disturbances. The results reveal that our proposed approach outperforms baseline algorithms.

I. INTRODUCTION

Collaborative human-robot systems can significantly reduce human workloads (Fig. 1). The capability of autonomous robots to adapt to disturbances (such as human behavioral uncertainties or robot actuation noise) is the key to determining system operational efficiency and safety [2]. One frequently encountered task in engineering settings is object transportation. To employ a human and a mobile manipulator to perform co-transportation, the key challenges arise from the disturbances [3], 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 [4].

To address these challenges, this paper formulates and solves a disturbance-aware Model Predictive Control (MPC) tracking problem enhanced with a pose optimization mechanism. The goal is to derive optimal control strategies by leveraging whole-body control, enabling the robot to estimate the impact of disturbances on costs in terms of tracking errors and energy consumption when simultaneously controlling the mobile base and robotic arm. Building on this, the pose optimization mechanism allows the robot to dynamically adjust its joint angle configurations, more efficiently compensating for trajectory deviations caused by disturbances and reducing control costs.

A. Mahmud, A. Raj, D. Nguyen, X. Xiao, and X. Wang are with George Mason University. W. Li is with the University of Tennessee, Knoxville. Supplementary material of this work can be found at [1].

This research is supported by NSF IIS-2153426 and ECCS-2332210.

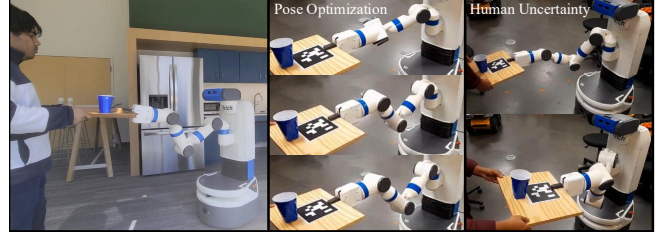


Fig. 1: Human robot co-transportation through disturbance-aware MPC and pose optimization (redundancy resolution for the same end effector position).

Statement of contribution: Our contributions in this paper are threefold. Firstly, we formulate a new problem that considers disturbances in an MPC tracking problem that incorporates pose optimization and the robot’s whole-body control. Secondly, we propose a dual-phase optimization strategy, which calculates the estimated control costs in the presence of disturbance 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 hardware demos using a Fetch robot. We use quantitative comparisons to highlight the advantages of our approach over existing algorithms that either do not consider disturbances or lack a pose optimization mechanism.

II. LITERATURE REVIEW

Trajectory Tracking: 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, learning-based nonlinear MPC [5], and adaptive planning can be employed. On the other hand, tracking using a robotic arm features algorithms such as Gaussian process-based MPC [6], adaptive time delay control [7], and model predictive path-following control [8]. These works focus on the high degree of freedom (DOF) and address control problems under complex task configurations and workspace constraints. When combining a robotic arm with a mobile base, many existing studies have addressed their comprehensive whole-body kinematics control [9]. These holistic control mechanisms can significantly expand the operational workspace of the mobile manipulator. Nonetheless, in many cases, the authors often overlook disturbances, particularly those introduced by human uncertainties in collaborative tasks.

Pose Optimization: As part of end-effector control with a high degree of freedom, our pose optimization arises from

the concept of redundancy resolution [10], where multiple alternative joint angle combinations can achieve the same end-effector pose. Redundancy resolution has been studied for robotic arm manipulation using optimization-based methods [11], Monte Carlo tree search, the stiffness model of the end-effector, and sampling-based methods [12]. Its main benefits include avoiding singularities, improving reachability in constrained spaces, and enhancing control accuracy [13]. These redundant robot poses are often used to overcome static or dynamic environmental constraints. The efforts to integrate pose optimization into co-manipulation problems are limited [14], and, similar to the tracking problem, they are not usually designed to better compensate for human uncertainties. In contrast, our formulation integrates pose selection with a disturbance-aware MPC formulation, leading to the best pose that minimizes future expected costs.

Human-Robot Collaboration: Adapting robot responses to align with human uncertainties during collaborative tasks is necessary, but it 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 [15], engaging in collaborative assembly tasks using Gaussian Process MPC [16], robust variable admittance control with unknown payload, sampling-based MPC, such as MPPI, for task and motion planning [17], handling task and joint space constraints etc. Some studies have considered the whole-body kinematics of the mobile manipulator during co-manipulation or transportation tasks with a reactive controller, an adaptive collaborative interface [18], and an environmental awareness module; as well as human-humanoid collaboration to reduce the human workload in carrying objects [19], etc. While reinforcement learning demonstrates effectiveness in handling unmodeled human uncertainties, it often lacks transparency to theoretically ensure performance guarantees. To address this, robust MPC provides a control-theoretic approach to handling uncertainties in MPC problems, including consideration of strict safety and physical constraints, as well as systems with varying parameters [20]. However, they usually do not integrate the MPC formulation with pose optimization to further improve control performance, which is a key difference from the problem considered in this paper.

III. PRELIMINARIES AND PROBLEM FORMULATION

We first introduce a trajectory with human uncertainties that the robot must follow and adapt to. Then, we detail the whole-body control 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 disturbances and incorporates pose optimization.

Notations: Let I_r denote the $r \times r$ identity matrix; $\text{diag}\{a_1, \dots, a_r\}$ denote a diagonal matrix. 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 that the matrix is positive (semi-) definite. We let $\|x\|_{M \succeq 0}^2 = x^\top M x$.

A. Trajectory Prediction with Human Uncertainty

As illustrated in Fig. 1, the task is to enable a team comprising a mobile manipulator and a human to collaboratively transport an object. We use $r = [r^x, r^y, r^z, r^\alpha, r^\beta, r^\gamma]^\top \in \mathbb{R}^6$ to represent the manipulator's end effector pose to be tracked in order to adapt to human movements, where the entries correspond to three-dimensional positions and orientations (roll, pitch, yaw), respectively. A trajectory is then defined as $r(k)$, $k \in \{0, 1, 2, \dots, H\}$ with H being the control time horizon for future time steps. In general, the trajectory $r(k)$ can be obtained by predicting future human behaviors based on their historic poses. Additionally, our formulation considers a disturbance term $\varpi(\tau)$ in the trajectory caused by human uncertainty, which yields

$$r(k) = \tilde{r}(k) + D \sum_{\tau=0}^k \varpi(\tau). \quad (1)$$

Here, $\tilde{r}(k)$ is the predicted human behavior, which can be obtained by various different methods in the literature, such as direct curve fitting [21], recurrent neural network [22], etc. If the co-transportation task has a desired nominal trajectory, $\tilde{r}(k)$ may also represent such a trajectory that is known to the robot. The term $\varpi(\tau) \sim \mathcal{N}(\mathbf{0}, \Sigma)$ represents the human positional uncertainty that functions as disturbances to $r(k)$. The distribution of ϖ is assumed to be zero mean and follows a covariance matrix $\Sigma \in \mathbb{R}^{3 \times 3}$ in x, y, z directions, where Σ encapsulates individual variations of humans. Different values of Σ can represent different human preferences, as they tend to introduce disturbances in varying directions and magnitudes. Given the positional disturbances caused by humans, we assume that the desired roll, pitch, and yaw remain unchanged. Thus, $D = \begin{bmatrix} I_3 & \mathbf{0}_{3 \times 3} \end{bmatrix}^\top \in \mathbb{R}^{6 \times 3}$. In addition, as will be discussed at the end of the next subsection, $\varpi(\tau)$ can also be equivalently used to represent the robot's actuation disturbances if an appropriate covariance matrix Σ can be quantified for the robot. Since this paper primarily focuses on how robots can adapt to human uncertainties rather than on human modeling itself, in the following, we assume $\tilde{r}(k)$ and Σ are known. Note that the use of different methods to obtain $\tilde{r}(k)$ and Σ does not impact the correctness of our results.

B. Whole-body Control of a Mobile Manipulator

We use a Fetch mobile manipulator to present modeling details. However, a similar mechanism is generalizable to a wide class of mobile manipulator platforms.¹ The Fetch mobile manipulator has a differential drive base and a 7-DOF robotic arm. We consider the heading angle of the base ϕ as an additional degree of freedom, which yields an 8-DOF configuration. With this, the end-effector pose $s \in \mathbb{R}^6$ expressed in the robot body frame can be transformed from the inertial frame without rotation. Detailed modeling of the

¹Here, for simplicity, we only present the kinematics considering joint velocities. A dynamics model with torque control is given in the supplementary material [1]. The following results are directly applicable to the dynamics model.

mobile base, robotic arm, and whole-body kinematics of the robot's end-effector is in supplementary material [1]. The state update equation of the end-effector is given by

$$s(k+1) = s(k) + B(\theta(k))u(k) \quad (2)$$

with

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

where $u = [v, \omega^\top]^\top \in \mathbb{R}^9$ is the control input combining the linear velocity of the base v and all rotations of the robot ω . $B(\theta(k)) \in \mathbb{R}^{6 \times 9}$ is the input matrix depending on the joint angle combinations $\theta(k)$. $\theta = [\phi, \theta_2, \theta_3, \dots, \theta_8]^\top \in \mathbb{R}^8$ includes the mobile base heading angle ϕ and the seven robotic arm angles; and $\omega = \dot{\theta} = [\eta, \dot{\theta}_2, \dot{\theta}_3, \dots, \dot{\theta}_8]^\top \in \mathbb{R}^8$ is the corresponding angular velocities. The Jacobian matrix is computed by $\mathbf{J}(\theta(k)) = \left. \frac{\partial f(\theta)}{\partial \theta} \right|_{\theta=\theta(k)} \in \mathbb{R}^{6 \times 8}$ where $f(\cdot)$ is the forward kinematics equation which maps the end-effector pose from the joint angle combinations. Δt is the time interval.

C. Disturbance Aware MPC with Pose Optimization

Our research problem focuses on effectively adapting to disturbances, subject to the whole body kinematics of the robot, as per Equation (2). We solve this by introducing a special MPC formulation, which features a **pose optimization** mechanism. Specifically, 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 $\bar{\theta}$ before the execution of control inputs, if the new $\bar{\theta}$ leads to a lower predicted cost over the planning horizon. The rationale behind this stems from redundant solutions that, given the same end-effector pose realization, 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 pose with the advantage of inducing less future cost. The disturbance-aware MPC tracking with pose optimization can be formulated as follows:

$$\begin{aligned} & \min_{u(0:H-1), \bar{\theta} \in \Theta} \mathcal{J}(u(0:H-1), \bar{\theta}) \\ & \triangleq \mathbb{E}_\varpi \left[\sum_{k=0}^H [s(k) - r(k)]^\top Q [s(k) - r(k)] \right] \\ & + \sum_{k=0}^{H-1} u(k)^\top R u(k) + \kappa |\bar{\theta} - \theta_0|_2^2 \\ & \text{s.t. } s(k+1) = s(k) + B(\bar{\theta})u(k), \quad s(0) = s_0 \\ & r(k) = \tilde{r}(k) + D \sum_{\tau=0}^k \varpi(\tau), \quad \varpi \sim \mathcal{N}(\mathbf{0}, \Sigma) \end{aligned} \quad (3)$$

where $u(0:H-1) = \{u(0), \dots, u(H-1)\}$, $Q \in \mathbb{R}^{6 \times 6} \succeq 0$, $R \in \mathbb{R}^{9 \times 9} \succ 0$ 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 [23], [24], we consider the linearized system with a fixed $B(\cdot)$ matrix for the robot's end-effector kinematics for each planning horizon. The impact is small when the planning horizon is short [24].

IV. MAIN RESULT

To solve problem (3), we observe that the optimal control input sequence, $u(0:H-1)$, is influenced by the input matrix $B(\bar{\theta})$. Thus, it also depends on the robot pose $\bar{\theta}$, which leads to our pose optimization. Since there are no closed-form solutions for $u(0:H-1)$ for general tracking problems in MPC, optimizing both $u(0:H-1)$ and $\bar{\theta}$ at the same time presents a challenge. 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, $u(0:H-1)$, and estimate the cost-to-go associated with it, considering human uncertainties ϖ . Then, in the second step, we examine candidate joint angle combinations and select the one that yields the lowest estimated cost. This combines the best of both tracking cost optimization and pose optimization to find the most efficient $\bar{\theta}$ and $u(0:H-1)$. We start by presenting the result to solve the optimal control input sequence $u(0:H-1)$ 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 (2) by subtracting $r(k+1)$ from both sides of the equation:

$$\begin{aligned} e(k+1) &= e(k) + \bar{B}u(k) + r(k) - r(k+1) \\ &= e(k) + \bar{B}u(k) + \tilde{r}(k) - \tilde{r}(k+1) - D\varpi(k+1) \end{aligned} \quad (4)$$

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

$$\mathcal{J}^*(e(k), k) = \|e(k)\|_{P(k)}^2 + 2e(k)^\top p(k) + c(k) \quad (5)$$

where, $P(k) \in \mathbb{R}^{6 \times 6}$, $p(k) \in \mathbb{R}^6$, $c(k) \in \mathbb{R}$, are unknown quantities to be determined. The following result shows that the assumed solution form is valid, and the parameters can be computed from a Discrete Algebraic Riccati Equation (DARE).

Theorem 1. *Given a \bar{B} , assuming the optimal solution u^* of (3) yields an optimal cost \mathcal{J}^* with the form of (5). Then $P(k)$, $p(k)$, and $c(k)$ can be computed by the following disturbance-aware DARE:*

$$P(k) = Q + P(k+1) - P(k+1)\bar{B}M P(k+1) \quad (6a)$$

$$\begin{aligned} p(k) &= p(k+1) + P(k+1)(\tilde{r}(k) - \tilde{r}(k+1)) \\ &\quad - P(k+1)\bar{B}M P(k+1)(\tilde{r}(k) - \tilde{r}(k+1)) \\ &\quad - P(k+1)\bar{B}M p(k+1) \end{aligned} \quad (6b)$$

$$\begin{aligned} c(k) &= c(k+1) + \|\tilde{r}(k) - \tilde{r}(k+1)\|_{P(k+1)}^2 \\ &\quad + \text{Tr}(\Sigma D^\top P(k+1)D) - \|P(k+1)(\tilde{r}(k) \\ &\quad - \tilde{r}(k+1)) + p(k+1)\|_{B M}^2 + 2(\tilde{r}(k) \\ &\quad - \tilde{r}(k+1))^\top p(k+1) \end{aligned} \quad (6c)$$

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

$$u^*(k) = -M(P(k+1)(e(k) + \tilde{r}(k) - \tilde{r}(k+1)) + p(k+1)) \quad (7)$$

gives the cost in (5) with parameters in (6). \square

Algorithm 1: Human-disturbance-aware MPC Tracking with Pose Optimization

- 1 **Input** Nominal trajectory $\tilde{r}(k)$; human uncertainty covariance matrix Σ ; current joint angles θ_0 .
 - 2 Formulate $r(k=0:H)$ 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 (2).
 - 7 Solve the MPC by computing solutions for the disturbance-aware DARE in *Theorem 1*.
 - 8 Compute the optimal cost \mathcal{J}^* associated with the current $\bar{\theta}$ using (5).
 - 9 **end**
 - 10 Compare the costs \mathcal{J}^* for all candidates $\bar{\theta}$ and find the optimal $\bar{\theta}^*$ as pose optimization. Compute the control for pose optimization as $\bar{u} = \frac{\bar{\theta}^* - \theta_0}{\tau}$.
 - 11 Reuse step 7 for the selected $\bar{\theta}^*$ and equation (7) to generate control input sequence $u^*(0:H-1)$.
 - 12 Add \bar{u} and $u^*(0)$ then apply them to the robot.
-

A complete proof of Theorem 1 can be found in [1]. From Theorem 1 and equation (7), 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. Specifically, our two-step solver is summarized in Algorithm 1. For 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. For every $\bar{\theta} \in \Theta$, we compute the control matrix $\bar{B}(\bar{\theta})$. 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 (7) to obtain the associated control inputs. For implementation, to precisely follow our definition of control cost, we should first apply pose optimization and then apply arm control u^* . However, this separation can be challenging for the hardware runtime. To address this, we fuse the pose optimization (if it exists) and the arm control inputs into a single step. Based on the triangle law of vector addition, the combined implementation will always result in no greater cost than the separated implementations. Furthermore, the cost difference is minimal when the angle change for pose optimization is sufficiently small.

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 costs. A similar technique has been used in [12]. Increasing the cardinality of Θ can potentially lead to a better $\bar{\theta}^*$ using more computation, and the evaluation of $\bar{\theta}$ can be performed in parallel. Furthermore, as we will demonstrate in experiments, optimizing poses periodically over fixed intervals (not every step) can also improve tracking performance. Lastly, to make the pose optimization more efficient, one may leverage deep learning methods to determine when a pose optimization is needed [25], and how a candidate set should be chosen [26].

V. EXPERIMENTS

In this section, we evaluate our proposed algorithm through simulation experiments in Gazebo and a demonstration using a Fetch mobile manipulator. In the following, we assume human uncertainties are the source of disturbance, which is mathematically equivalent if the robot has actuation disturbances. To simulate human uncertainties, we assume $\varpi(k) \sim \mathcal{N}(\mathbf{0}, \Sigma)$, where $\Sigma = q \cdot \text{diag}(0.015, 0.025, 0.015)$ (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. We test our algorithm by considering two trajectories under two cases. For one case, we assume human future poses are unknown and use LSTM [27] for pose prediction to obtain \tilde{r} in (1). We use two layers of LSTM, each containing 50 units and a dense layer (output layer). The LSTM layers are fed with the sequences of historic poses, and the output of the layers is the predicted poses. Details of LSTM implementation are given in the supplementary material [1]. The experiments are denoted as \mathcal{A}_P and \mathcal{B}_P . We also test the algorithm assuming human future poses are known as a nominal trajectory \tilde{r} , although the trajectory is still subject to human uncertainties. The experiments are denoted as \mathcal{A}_N and \mathcal{B}_N . Each trajectory has 500 discrete points, with a time interval of $\tau = 0.1$ seconds. The parameters used for solving the problem (3), are selected to be $H = 3$ (MPC horizon) for \mathcal{A}_P and \mathcal{B}_P ; $H = 8$ for \mathcal{A}_N and \mathcal{B}_N . We let $R = I_9$, and $\kappa = 1$. We test two different settings for Q , with $Q = 1000 \cdot I_6$ or $Q = 500 \cdot I_6$ to reweigh the tracking error on the overall cost.

Our experiments follow Algorithm 1, and we repeat the MPC planning every time step. In each planning horizon, we chose twelve candidate poses in Θ . While the proposed algorithm seeks to minimize an expected cost, 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

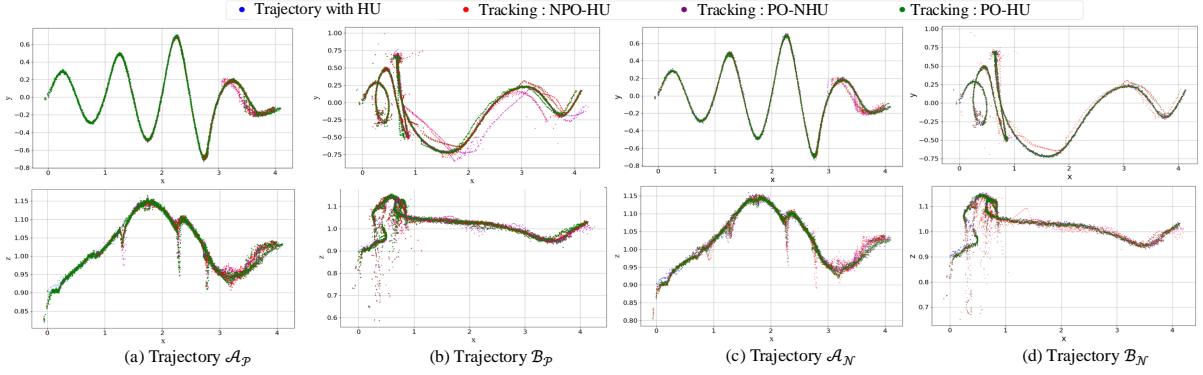


Fig. 2: All trajectories with human uncertainties, $Q = 1000 \cdot I_6$, $q = 0.4$. The results of trials are represented by dot clouds.

the total number of time steps. We compare our proposed approach (PO-HU, which considers pose optimization and human uncertainty) with two baselines: one approach that considers no pose optimization but human uncertainty (NPO-HU) and another that considers pose optimization but not human uncertainty (PO-NHU). Note that we do not evaluate the no pose optimization and no human uncertainty 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 (7).

The dot clouds in Fig. 2 compare tracking performance across 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 C_{total} of different algorithms over 100 trials. Additionally, 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. For hardware validation, all the trajectories computed from Gazebo visualized in Fig. 2 have been executed and successfully reproduced on our hardware platform to justify their feasibility. The effectiveness of the proposed algorithm is justified by Fig. 2. In all cases, the result of PO-HU closely follows the trajectories, whereas the other two baseline algorithms deviate a lot. This is further

reinforced by the numerical results in Table I. Specifically, in the table, we use \dagger in NPO-HU, and \ddagger 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). Finally, from both figures and tables, we observe that in general, the robot performs better in \mathcal{A}_N , \mathcal{B}_N . This is because robots have access to nominal trajectories of human future poses, which is more informative than human prediction cases \mathcal{A}_P , \mathcal{B}_P . This is especially true for trajectory \mathcal{B} because it is more complex and has sharp turns, where the predicted human future poses can be inaccurate.

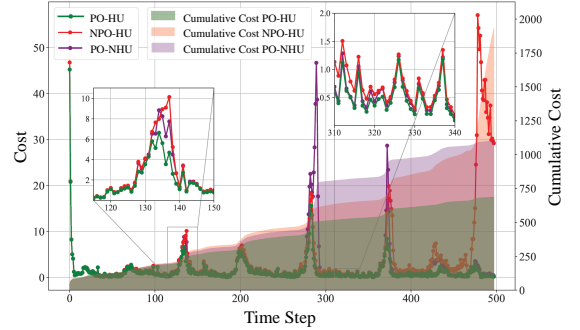


Fig. 3: Accumulated cost comparison for trajectory \mathcal{A}_N , with $Q = 1000 \cdot I_6$, $R = I_9$, $q = 0.4$.

Fig. 3 visualizes the accumulated cost over time and the cost at each time step for trajectory \mathcal{A}_N . 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 with pose optimization remain relatively stable. Similar trends are seen for other cases.

Fig. 4 shows the average execution time of the proposed algorithm for different planning horizons, averaging over 500

Traj	Q	q	PO-HU	pPO-HU	NPO-HU	PO-NHU
\mathcal{A}_P	$1000I_6$	0.4	744.04	841.06	2364.77 \dagger	1339.781
		0.7	1273.42	1303.88	2394.74	2427.38
	$500I_6$	0.4	846.64	847.77	901.58	898.16
		0.7	999.66	1003.27	1077.64	1056.52
\mathcal{B}_P	$1000I_6$	0.4	2436.97	2671.56	5546.44 \dagger	3172.04
		0.7	3831.34	4083.78	9731.61 \dagger	9234.85 \ddagger
	$500I_6$	0.4	2030.57	2138.46	8752.65 \dagger	2409.09
		0.7	2324.85	2499.22	9589.51 \dagger	7509.28 \ddagger
\mathcal{A}_N	$1000I_6$	0.4	686.94	769.55	1932.83 \dagger	1102.95
		0.7	1151.33	1292.57	2342.11 \dagger	2384.16 \ddagger
	$500I_6$	0.4	829.59	835.64	851.60	852.53
		0.7	978.52	986.46	1013.69	1017.27
\mathcal{B}_N	$1000I_6$	0.4	2306.64	2540.73	3759.09 \dagger	2968.25
		0.7	3782.70	3943.75	8694.57 \dagger	7220.29 \ddagger
	$500I_6$	0.4	2013.31	2041.27	2261.32	2209.01
		0.7	2293.11	2481.39	5612.76 \dagger	4301.46 \ddagger

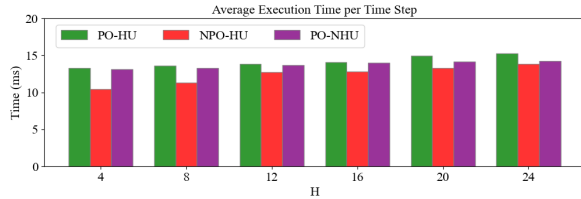


Fig. 4: Execution time comparison with different planning horizon, $|\Theta| = 12$. Test computer uses AMD 5975WX.

trials. Steps (4-9) of the proposed algorithm can be fully parallelized, thus the computation time for pose optimization only increases slightly, as long as the size of the set $|\Theta| = 12$ is smaller than the number of computing threads. Furthermore, 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 is due to more iterations in solving the DARE (cf. equation (6)). This concludes that our proposed PO-HU method does not introduce significant extra execution time compared with other baseline algorithms.

VI. CONCLUSIONS AND FUTURE WORK

We studied the control of a mobile manipulator to perform human-robot co-transportation tasks. By modeling human uncertainties and the whole-body kinematics of the robot, we formulated a new human-disturbance-aware MPC tracking problem with pose optimization. We proposed an algorithm with a two-step iterative design, equipped with an inner loop that computes a disturbance-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 both theoretically and experimentally. Future work will include the generalization of the algorithm to multi-human multi-robot collaborative tasks.

REFERENCES

- [1] A. J. Mahmud, A. H. Raj, D. M. Nguyen, W. Li, X. Xiao, and X. Wang, "Supplementary: Human-robot co-transportation using disturbance-aware mpc with pose optimization," https://mason.gmu.edu/~xwang64/papers/25_IROS_MPCPO_Supp.pdf.
- [2] A. J. Mahmud, D. M. Nguyen, F. Veiga, X. Xiao, and X. Wang, "Human uncertainty-aware mpc for enhanced human-robot collaborative manipulation," in *2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 1–6, 2024.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] J. Cavacanti Santos and M. Martins da Silva, "Redundancy resolution of kinematically redundant parallel manipulators via differential dynamic programming," *Journal of Mechanisms and Robotics*, vol. 9, no. 4, p. 041016, 2017.
- [11] F. Tassi, E. De Momi, and A. Ajoudani, "An adaptive compliance hierarchical quadratic programming controller for ergonomic human-robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 78, p. 102381, 2022.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] Y. Zhang, C. Pezzato, E. Trevisan, C. Salmi, C. H. Corbato, and J. Alonso-Mora, "Multi-modal mppi and active inference for reactive task and motion planning," *IEEE Robotics and Automation Letters*, 2024.
- [18] 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.
- [19] D. J. Agravante, A. Cherubini, A. Sherikov, P.-B. Wieber, and A. Kheddar, "Human-humanoid collaborative carrying," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 833–846, 2019.
- [20] M. Zanon and S. Gros, "Safe reinforcement learning using robust mpc," *IEEE Transactions on Automatic Control*, vol. 66, no. 8, pp. 3638–3652, 2020.
- [21] V. Parque and T. Miyashita, "Smooth curve fitting of mobile robot trajectories using differential evolution," *IEEE Access*, vol. 8, pp. 82855–82866, 2020.
- [22] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of human full-body movements with motion optimization and recurrent neural networks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1792–1798, IEEE, 2020.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] B. X. Nie, P. Wei, and S.-C. Zhu, "Monocular 3d human pose estimation by predicting depth on joints," in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3467–3475, IEEE, 2017.