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 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 com-
prehensive whole-body dynamics control. These studies employ
methodologies such as linear programming [17], constrained
sequential linear quadratic control [18], end-to-end reinforce-
ment learning [19], and nonlinear MPC [20]. These holistic
dynamics 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 po-

tion. This can help to avoid singularities [22], improve
reachability in constrained spaces [23], and enhance con-
tral accuracy [24]. These pose optimization approaches are
often used to overcome static or dynamic environmental
constraints. The efforts to integrate pose optimization into
collaboration 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 uncer-
tainties 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 saw-
ing 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 transpor-
tation tasks [29], [30]. While reinforcement learning shows effec-
tiveness in handling unmodeled human uncertainties, it usu-
ally lacks transparency to theoretically ensure performance
guarantees. To address this, robust MPC provides a control
theoretical 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 embed-
ed 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 manipu-
lator, 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 \( i \)-th 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
\( M \succ 0, M \preceq 0 \) 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 collabo-
ratively 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:

\[
\mathbf{r}_t = [x_t, y_t, z_t, \alpha_t, \beta_t, \gamma_t]^\top \in \mathbb{R}^6
\]

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 \( r_t^\alpha, r_t^\beta, \) and \( r_t^\gamma \)
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., \( \forall t, r_t^\alpha = 0 \).

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:

\[
r(k) = r(k) + D(h + \sum_{\tau=0}^k \omega(\tau)), \quad (1)
\]

with \( \omega(\tau) \sim \mathcal{N}(0, \Sigma) \). Here, \( D = [I_3 \quad \mathbf{0}_{3 \times 3}]^\top \in \mathbb{R}^{6 \times 3} \)
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 \( \omega \in \mathbb{R}^3 \) represents the
predicted human positional disturbance for future time steps.
We assume the distribution of \( \omega \) is zero mean and follows
a covariance matrix \( \Sigma \in \mathbb{R}^{3 \times 3} \) in \( x, y, z \) directions. The \( \Sigma \)
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.
\[ s_{\text{base}}(k + 1) = s_{\text{base}}(k) + \tau \begin{bmatrix} \cos(\phi(k)) & 0 \\ \sin(\phi(k)) & 0 \\ 0 & 1 \end{bmatrix} u_{\text{base}}(k) \]  

where \( s_{\text{base}} = [x_{\text{base}}, y_{\text{base}}, \phi]^T \in \mathbb{R}^3 \) represents the \( x, y \) positions and the heading angle of the robot base, both in the inertial frame; \( v_{\text{base}} = [v \ \eta]^T \) represents the linear and angular velocities of the mobile base in its own body frame. \( \tau \) 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 \( \phi \) 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 \( \Xi \) coordinate frame visualized in Fig. 2. \( \Xi \) 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 \( \phi \) into pose optimization, together with 7 other joint angles.

We represent the end-effector pose of the robotic arm in \( \Xi \) frame by \( s_{\text{arm}} = [p_{\text{arm}}^T \ \psi_{\text{arm}}^T] \in \mathbb{R}^6 \), where \( p_{\text{arm}} \in \mathbb{R}^3 \) denotes the end-effector position in Cartesian coordinates, \( \psi_{\text{arm}} \in \mathbb{R}^3 \) 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., \( r_t^3 = r_t^7 = 0 \), 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 \( s_{\text{arm}} \), we represent its forward kinematics equation \( f(\cdot) \) using the Jacobian matrix \( J(\cdot) \in \mathbb{R}^{6 \times 8} \) derived based on the DH-parameters [38] of the robotic arm configured in Fig. 3:

\[ s_{\text{arm}}(k + 1) = f(\theta(k) + \tau \omega(k)) \]

\[ \approx s_{\text{arm}}(k) + \tau J(\theta(k))\omega(k) \]

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

**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 \( s_{\text{base}} \) is defined in the inertial frame, \( s_{\text{arm}} \) is defined in the \( \Xi \) frame, and no rotation is needed for the transition between the two frames. Consequently, the state update is given by

\[ s(k + 1) = s(k) + B(\theta(k))u(k) \]  

with

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

where \( u = [v, \omega]^T \in \mathbb{R}^3 \) 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 \( \varpi \) 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 \( \theta_0 \) to a new combination \( \theta \) selected from a candidate set, if the new \( \theta \) 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 \( \theta \), 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:

\[
J^*(\bar{u}(0 : H - 1), \bar{\theta}) = \min_{u(0 : H - 1), \bar{\theta}} \mathbb{E}_{\varpi} \left[ \sum_{k=0}^{H-1} \left[ s(k) - \bar{r}(k) \right]^\top Q \left[ s(k) - \bar{r}(k) \right] + \sum_{k=0}^{H-1} u(k)^\top Ru(k) + \| \bar{\theta} - \theta_0 \|^2 \right] \quad \text{s.t.} \quad s(k+1) = s(k) + B(\bar{\theta})u(k), \quad s(0) = s_0
\]

where \( u(0 : H - 1) = \{ u(0), \ldots, u(H-1) \} \), \( Q \in \mathbb{R}^{6 \times 6} \geq 0, R \in \mathbb{R}^{9 \times 9} \geq 0 \) 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 \( \theta_0 \) 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 \( u(0 : H - 1) \), 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 \( u(0 : H - 1) \) for general MPC tracking problems, optimizing both \( u(0 : H - 1) \) 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, \( u(0 : H - 1) \), within the planning horizon and estimate the cost-to-go associated with it, considering uncertainties caused by humans, referred to as \( \varpi \). 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 \( u(0 : H - 1) \).

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

\[
e(k + 1) = e(k) + \bar{B}u(k) + \bar{r}(k) - \bar{r}(k + 1) = e(k) + \bar{B}u(k) + r(k) - r(k + 1) - D\varpi(k + 1) = e(k) + s(k) - \bar{r}(k) \quad \text{being the tracking error. We hypothesize that the optimal cost-to-go function follows:}
\]

\[
J^*(e(k), k) = \| e(k) \|^2_{P(k)} + 2e(k)^\top p(k) + c(k)
\]

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 \( u^* \) of (5) yields an optimal cost \( J^* \) with the form of (7). Then \( P(k), p(k), \) and \( c(k) \) can be computed by the following uncertain-aware DARE:

\[
P(k) = Q + P(k + 1) - P(k + 1)\bar{B}MP(k + 1) \quad (8a)
\]

\[
p(k) = p(k + 1) + P(k + 1)(r(k) - r(k + 1)) - P(k + 1)\bar{B}MP(k + 1)(r(k) - r(k + 1)) - P(k + 1)\bar{B}Mp(k + 1) \quad (8b)
\]

\[
c(k) = c(k + 1) + \| r(k) - r(k + 1) \|^2_{P(k + 1)} + Tr(\Sigma D^\top P(k + 1)D) - \| P(k + 1)(r(k) - r(k + 1)) + p(k + 1) \|^2_{BM} + 2(r(k) - r(k + 1))^\top p(k + 1) \quad (8c)
\]

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

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

The corresponding control input

\[
u^*(k) = -M(P(k + 1)(c(k) + r(k) - r(k + 1)) + p(k + 1))
\]

(9)

gives the cost in (7) with parameters in (8). \( \square \)

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 \( J^* \), 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 $\Sigma$; current accumulated human disturbance $h$; current joint angles $\theta_0$.
2 Formulate $\tilde{f}(k = 0 : H)$ based on equation (1).
3 Create the joint angles candidate set $\Theta$ by sampling around $\theta_0$, also add $\theta_0$ to $\Theta$.
4 for each $\tilde{\theta} \in \Theta$ do
5   Compute Jacobian matrix, $\mathbf{J}(\tilde{\theta}) = \frac{\partial \tilde{f}(\theta)}{\partial \theta}|_{\theta = \tilde{\theta}}$.
6   Compute matrix $\mathbf{B} = \mathbf{B}(\tilde{\theta})$ with (4).
7   Solve the MPC by computing solutions for the uncertain-aware DARE in Theorem 1.
8   Compute the optimal cost $J^*$ associated with the current $\tilde{\theta}$ using (7).
9 end
10 Compare the costs $J^*$ for all candidate $\tilde{\theta}$ and find the optimal $\theta^*$ as pose optimization.
11 Apply the pose optimization to the robot.
12 Reuse the results in step 7 for the selected $\theta^*$ and equation (9) to generate optimal control input sequence $u^*(0 : H - 1)$, and apply them to the robot.

Algorithm 1. At the beginning of each MPC horizon, we first create the joint angles candidate set $\Theta$ by randomly changing multiple joint angles of $\theta_0$ with a small radian value. This should lead to small pose optimization cost $\kappa(\theta - \theta_0)^2$ and small changes to the end-effector pose of the robot. For every $\tilde{\theta} \in \Theta$, we compute $\mathbf{B}$ that defines the control input matrix of the system dynamics. Then, we use Theorem 1 to compute the estimated optimal cost $J^*$ for the current $\tilde{\theta}$. This process is parallelizable to improve computational efficiency. By exploiting all $\tilde{\theta} \in \Theta$ and the associated $J^*$, we select the best $\tilde{\theta}$ 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 $\mathbf{B}(\tilde{\theta})$ and $\tilde{\theta}$ makes it computationally infeasible to systematically find the optimal pose $\theta$ for estimated control cost $J^*$. 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 $\tilde{\theta}$ that is better than $\theta_0$ in terms of future control cost. A similar technique has been used in [43]. Here, increasing the cardinality of $\Theta$ can potentially lead to a better $\theta^*$, but also incurs more computation. As previously mentioned, since the evaluation of $\tilde{\theta}$ can be performed in parallel, the cardinality of $\Theta$ 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 $\omega(k) \sim N(0, \Sigma)$, 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. 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 $A$, $B$, $C$, and $D$ 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 $H = 8$, $R = I_6$, 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 $\Theta$. 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:

$$C_{\text{total}} = \sum_{t=1}^{T} e(t)^T Q e(t) + u(t)^T R u(t) + \kappa(\bar{\theta}(t) - \theta(t))^2,$$

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.
Table I, which shows the average total cost $C_{\text{total}}$ across 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 $C_{\text{total}}$ across Different Algorithms

<table>
<thead>
<tr>
<th>Traj</th>
<th>Q</th>
<th>$q$</th>
<th>PO-HU</th>
<th>pPO-HU</th>
<th>NPO-HU</th>
<th>PO-NHU</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1000$I_6$</td>
<td>0.4</td>
<td>1019.81</td>
<td>1023.15</td>
<td>1119.29</td>
<td>1057.35</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>1323.96</td>
<td>1337.11</td>
<td>1431.20</td>
<td>1370.91</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>886.87</td>
<td>896.12</td>
<td>1075.34</td>
<td>914.19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>1027.91</td>
<td>1032.42</td>
<td>1160.69</td>
<td>1070.57</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>970.99</td>
<td>972.71</td>
<td>1189.39</td>
<td>1231.87</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>1161.31</td>
<td>1178.59</td>
<td>1350.03</td>
<td>1500.54</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1000$I_6$</td>
<td>0.4</td>
<td>886.87</td>
<td>927.54</td>
<td>5875.60</td>
<td>2123.53</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>906.92</td>
<td>1140.84</td>
<td>1938.88</td>
<td>2515.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>866.94</td>
<td>769.55</td>
<td>1932.83</td>
<td>1102.95</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>1151.33</td>
<td>1292.57</td>
<td>2342.11</td>
<td>2384.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>829.59</td>
<td>855.64</td>
<td>821.60</td>
<td>832.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>978.82</td>
<td>986.46</td>
<td>1013.69</td>
<td>1017.27</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>1000$I_6$</td>
<td>0.4</td>
<td>2306.64</td>
<td>2540.73</td>
<td>3759.09</td>
<td>2968.25</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>3782.70</td>
<td>3943.75</td>
<td>8694.57</td>
<td>7220.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>2013.31</td>
<td>2041.27</td>
<td>2261.32</td>
<td>2209.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>2293.11</td>
<td>2481.39</td>
<td>5612.76</td>
<td>4301.46</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>1000$I_6$</td>
<td>0.4</td>
<td>2306.64</td>
<td>2540.73</td>
<td>3759.09</td>
<td>2968.25</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>3782.70</td>
<td>3943.75</td>
<td>8694.57</td>
<td>7220.29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>2013.31</td>
<td>2041.27</td>
<td>2261.32</td>
<td>2209.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>2293.11</td>
<td>2481.39</td>
<td>5612.76</td>
<td>4301.46</td>
<td></td>
</tr>
</tbody>
</table>

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 B, C, 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 D is higher than that for other trajectories. This can be attributed to the inherent complexity and the sharp turns in trajectory D, as depicted in Fig. 4.

Figure 5 visualizes the accumulated cost over time and the cost at each time step for trajectory C, with $Q = 1000 \cdot I_6$, $R = I_9$, $q = 0.4$.

![Fig. 5: Accumulated cost comparison for trajectory C, with $Q = 1000 \cdot I_6$, $R = I_9$, $q = 0.4$.](image-url)
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 adopts 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.

![Average Execution Time per Time Step](image)

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

$$J^*(e(k), k) = \min_{u(k)} J(e(k), k)$$

$$= \min_{u(k)} \left[ \mathbb{E}_\mathbb{E} \left[ \left( \left\| e(k) \right\|^2_Q + \left\| u(k) \right\|^2_R + J^*(e(k+1), k+1) \right) \right] \right]$$

$$= \min_{u(k)} \left[ \mathbb{E}_\mathbb{E} \left[ \left( \left\| e(k) \right\|^2_Q + \left\| u(k) \right\|^2_R \right. \right. \right. \right.$$  

$$+ \left. \left. \left. \left( \left( e(k) + B_u(k) + r(k) - r(k+1) \right)^2 \right) + \right. \right. \right.$$  

$$\left. \left. \left. \left. \left. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \right. \r