Task-Constrained Motion Planning Considering Uncertainty-Informed Human Motion Prediction for Human–Robot Collaborative Disassembly

Wansong Liu, Xiao Liang, and Minghui Zheng

***Abstract*—While the disassembly of high-precision electronic devices is a predominantly labor-intensive process, collaborative robots provide a promising solution through human–robot collaboration. To ensure efficient yet safe collaboration, this article presents a new way to generate task-constrained and collision-free motion for a collaborative robot operating in a dynamic environment involving human movement, which is traditionally challenging due to the high degree of freedom of the corobot and the uncertainty nature of human motion. We first establish a neural human motion prediction model with quantified uncertainty, and then optimize the configuration of the robot online by taking the human motion and uncertainties into consideration. While such rationale is straightforward in nature, our method explicitly quantified the uncertainty of the neural human prediction model to further enhance the collaboration safety, and integrated the quantified uncertainty into the task-satisfied motion planning in real time to efficiently conduct tasks. Extensive experimental tests and comparison studies have been conducted to validate the efficiency and effectiveness of the proposed planning method.**

***Index Terms*—Human motion prediction, human–robot collaboration (HRC), motion planning, robot planning and control.**

I. Introduction

THERE has been an increasing amount of used high-precision electronic products, such as hard drives, cell phones, and computers. While the disassembly and recycling of such used products are labor intensive, collaborative robots provide a promising solution for such processes through human–robot collaboration (HRC) [1]. To guarantee the safety of human operators and the efficiency of the operation, the motion planning algorithms need to be reliable and fast, and usually require the robot to meet two types of constraints, i.e., the workspace inequality constraints (e.g., maintaining human–robot safety distance) and the task equality constraints (e.g., following a particular path or completing an unscrewing task) [2]. It is very challenging to satisfy such two constraints in real-time planning due to the high degree of freedom (DOF) of the manipulator and the uncertain and complex human motions in HRC.

Motion planning is usually investigated based on girding or sampling over the space. The grid-based methods, such as the A* algorithm [3], usually suffer from exponentially increased computational time as the manipulator's DOF becomes higher. The sampling-based methods, such as the probabilistic roadmap [4] and the rapid exploring random tree [5], randomly generate samples in the configuration space. They are good at searching collision-free motion for the manipulator with high DOF but not good at simultaneously satisfying additional constraints [6] induced by tasks in real applications, e.g., following a particular task path [2] or maintaining a desired end-effector's orientation [7]. Their computational complexity would be significantly increased when incorporating such task constraints [8]. As abovementioned, planning task-constrained (TC) motion in configuration space for high-DOF manipulator is not an easy problem and it becomes even more challenging in HRC as the free configuration space is varying in real time with respect to complex human movements.

To plan the TC motion for manipulators, existing studies, e.g., [2] and [9], successfully formulated and solved such planning problems as optimization-based problems with constraints in both task and configuration spaces. Wang et al. [10] used artificial potential field methods to plan the manipulator trajectory while satisfying desired constraints. In addition, to better collaborate with human operators and enable the robot to take action in advance, robotic planning may leverage the future information of human motion. The authors in [11] and [12] showed that robots can react earlier based on the knowledge of the human behavior provided by the prediction model. Recently, network-based models have shown effectiveness in the prediction of long-term and complex motions. For example, Li et al. [13] employed a long-term encoder to assist the network in better learning the past's motion. Mao et al. [14] proposed

Manuscript received 15 January 2023; revised 20 March 2023; accepted 23 April 2023. Date of publication 29 May 2023; date of current version 16 August 2023. Recommended by Technical Editor Zheng (AIM FS TE) Chen and Senior Editor Qingze Zou. This work was supported by the USA National Science Foundation under Grant 2026533. *(Corresponding authors: Minghui Zheng; Xiao Liang.)*

This work involved human subjects or animals in its research. The author(s) confirm(s) that all human/animal subject research procedures and protocols are exempt from review board approval.

Wansong Liu and Minghui Zheng are with the Mechanical and Aerospace Engineering Department, University at Buffalo, Buffalo, NY 14260 USA (e-mail: wansongl@buffalo.edu; mhzheng@buffalo.edu).

Xiao Liang is with the Civil, Structural and Environmental Engineering Department, University at Buffalo, Buffalo, NY 14260 USA (e-mail: liangx@buffalo.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2023.3275316.

Digital Object Identifier 10.1109/TMECH.2023.3275316

a network with an attention module to effectively exploit the historical information of complex human actions. Furthermore, recurrent neural network (RNN) has been widely leveraged for human motion prediction considering its capability of capturing the dependencies of the sequential motion data [15], [16], [17]. The past human motion characteristics memorized in the hidden states of RNN are beneficial for improving the efficiency (e.g., the human moving intention [18]) and safety (e.g., the possible occupied area [19]) in HRC. However, the RNN-based prediction models employed in HRC do not explicitly quantify the prediction model uncertainties.

It is worth noting that the predicted human motion should not be blindly integrated into robotic planning [20]. For example, during a collaboration, human operator's motions have variations and may change in an irregular way due to its nonlinear and stochastic nature. In this case, inaccurate predictions may lead the robot to plan a dangerous trajectory. Therefore, existing studies using Gaussian mixture regression [21] or hidden Markov model (HMM) [22] generated probabilistic distribution over the future state of human operator based on the observation of human past states, and quantified the uncertainty of human motions as additional knowledge for robotic planning.

Importantly, besides the uncertainty derived from human motions, network-based prediction models also have uncertainties. As highlighted in [23] and [24], uncertainty is a natural feature of network-based models and needs to be explicitly considered especially in human–robot close collaboration. It can be interpreted as the confidence level of these models regarding their outputs [25]. If a human prediction model with large uncertainty is treated accurately and blindly taken into robotic motion planning, it can be a potential threat to the human operator's safety. Cheng et al. [24] proposed a network to adaptively predict the time-varying human behaviors and estimate the prediction uncertainty. Franchi et al. [26] employed variational autoencoders (VAEs) to learn the interaction and the latent distribution of network parameters such that the uncertainty can be quantified. Lakshminarayanan et al. [27] trained an ensemble of network models, and measured the uncertainty based on the variance between model outputs. However, VAEs may sacrifice accuracy to impose the latent distribution prior while ensemble methods may require significant computational costs to train, especially with high-dimensional data [28]. Different from the ensemble methods that generate an ensemble of network models, Monte Carlo dropout sampling (MCDS) methods train a single model and use random dropout masks to output a set of predictions [29]. Furthermore, such an approach is applicable to sequence prediction problems (i.e., human motion prediction problems).

In this article, we present a new TC motion planning algorithm considering uncertainty-informed human motion prediction. Compared with the existing network-based human motion prediction for the robotic planning in HRC, to the best of authors' knowledge, instead of blindly relying on the prediction model, this article is among the very first to explicitly quantify the uncertainty of the neural prediction model as a danger zone around the human arm and incorporate it into the high-DOF collaborative manipulator planning when working with human. We have conducted extensive experimental studies to validate two advantages of our method: 1) larger safety margin with earlier manipulator's response, and 2) convenient incorporation regarding quantified prediction uncertainties.

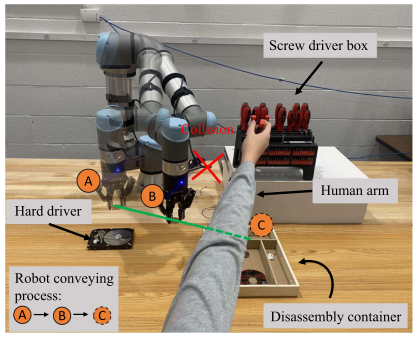

Fig. 1. Risky collaborative disassembly scene in which the manipulator and the human operator are conducting disassembly tasks collaboratively in close proximity. The manipulator is picking and placing a disassembled component while the human operator is switching the disassembly tool.

The rest of this article is organized as follows. Section II presents an overview of the proposed task-constrained planner (TC-Planner). Section III details the human motion prediction model and its uncertainty quantification. Section IV describes our online reconfiguration algorithm. Section V presents extensive experimental studies and validation. Finally, Section VI concludes this article.

II. Overview of the Proposed TC-Planner

In this section, we will

1) briefly explain the collision risk when a human and a robot are working closely on disassembly tasks,
2) introduce variable notations and definitions used in this article, and
3) give an overview of the proposed TC-Planner.

Fig. 1 illustrates a risky scene in which the manipulator and the human operator are performing disassembly tasks collaboratively in close proximity. The human arm locates in the manipulator's moving direction, which would trigger a collision. To follow the given path as well as avoid physical injury to the human operator, the manipulator needs to find a safe alternative configuration with the same end-effector's position to replace the current dangerous configuration. If the manipulator could know the human arm's motion in advance, the manipulator would respond earlier to the future human motion resulting in larger safety margin.

The position of the manipulator's end-effector in e-dimensional task space is denoted as $x \in \mathbb{R}^e$. We use $\theta \in \mathbb{R}^q$ to denote the manipulator configuration in q-dimensional joint space, and $\mathcal{M}(\theta) \in \mathbb{R}^e$ to stand for the area occupied by the manipulator with the configuration θ in the task space. Forward kinematics maps the manipulator configuration θ in the joint space to the end-effector's position x in the task space using the following equation:

$$x = F(\theta) \tag{1}$$

where $F \in \mathcal{F}$ denotes the forward kinematics. Suppose the maniulator's end-effector follows a given path $\mathbf{x} = [x_1, \ldots,$

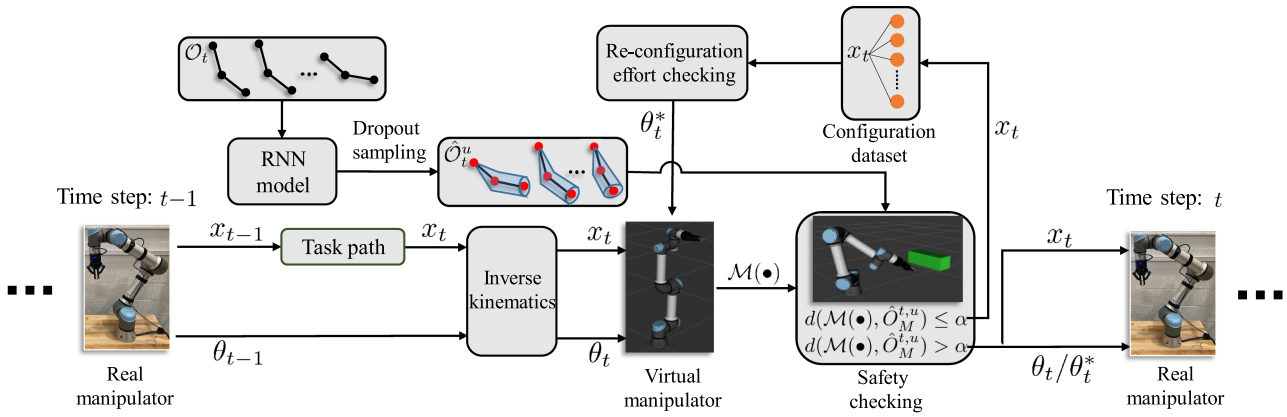

Fig. 2. Overview of the proposed TC-Planner and its implementation. The inverse kinematics is first used to initialize the next-step configuration of the manipulator. Meanwhile, the uncertainty-informed human motion prediction is obtained and the minimum distance between the robot and the human operator is checked. If the manipulator configuration is dangerous to the human operator in the near future, our online manipulator reconfiguration algorithm would select a safe alternative configuration. Finally, the next collision-free configuration, either θ_t or θ_t^*, is sent to the real manipulator.

$x_t, \ldots, x_T] \in \mathbb{R}^{e\times T}$, where x_t is the end-effector's position at t time step. The corresponding manipulator configurations are denoted as $\vartheta = [\theta_1, \ldots, \theta_t, \ldots, \theta_T] \in \mathbb{R}^{q\times T}$.

Denote the area occupied by the human arm in the task space as $O = H(v^a, v^b, p_h) \in \mathbb{R}^e$, where $v^a \in \mathbb{R}^e$ and $v^b \in \mathbb{R}^e$ are normalized bone vectors standing for the upper arm and forearm directions, respectively, p_h indicates the anthropometric parameters, e.g., bone length and radius of upper arm and forearm, and $H(\bullet)$ is a nonlinear mapping function to calculate the occupied area in the task space. Note that the prediction in this article is denoted with $\wedge$. The human arm motion in the task space is defined as follows.

1) The observed human motion at time t: $\mathcal{O}_t = [O_{t-N+1}, \ldots, O_t] \in \mathbb{R}^{e\times N}$, where $O^t = H(v_t^a, v_t^b, p_h)$, and N is the time horizon of the observation.
2) The predicted human motion at time t: $\hat{\mathcal{O}}_t = [\hat{O}_1^t, \ldots, \hat{O}_m^t, \ldots, \hat{O}_M^t] \in \mathbb{R}^{e\times M}$, where $\hat{O}_m^t = H(\hat{v}_{t+m}^a, \hat{v}_{t+m}^b, p_h)$, m is the prediction index, and M is the time horizon of the prediction.
3) The predicted human motion informed by the model uncertainty at time t: $\hat{\mathcal{O}}_t^u = [\hat{O}_1^{t,u}, \ldots, \hat{O}_m^{t,u}, \ldots, \hat{O}_M^{t,u}] \in \mathbb{R}^{e\times M}$, where u indicates the prediction model uncertainty.

Moreover, we use $d(\bullet)$ to denote the task-space-based minimum distance between the manipulator and the human arm, which will be used to assess the collision risk and serve as the safety index for human. The minimum safety distance is denoted as α.

Fig. 2 illustrates the overview of our proposed approach. First, we use the inverse kinematics to initialize the next-step configuration of the manipulator. Meanwhile, the uncertainty-informed prediction $\hat{\mathcal{O}}_t^u$ is obtained and the minimum distance is checked using $d(\bullet)$. If the manipulator configuration θ_t is dangerous to the human operator in the near future, our online manipulator reconfiguration algorithm would select an alternative configuration θ_t^* based on the reconfiguration effort and safety checking. Finally, the next collision-free configuration, either θ_t or θ_t^*, is sent to the real manipulator.

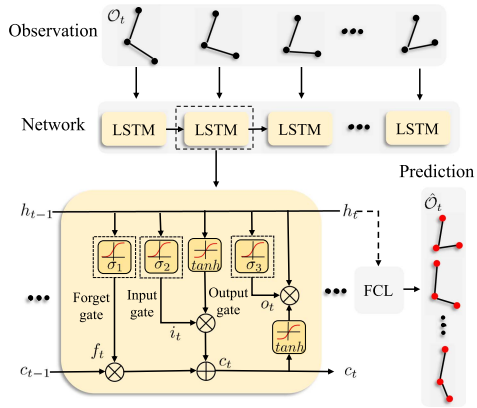

Fig. 3. Human motion prediction model: the temporal arm pose of observation O_t is the input of the LSTM cell, FCL indicates the fully connected layer, and FCL outputs the predicted arm motion $\hat{O}_t$.

III. TC-Planner: Uncertainty-Informed Human Motion Prediction

This section presents that we 1) use an RNN with long short-term memory (LSTM) structure to predict the arm motion, and 2) quantify the prediction model uncertainty based on Bayesian interface.

A. Human Motion Prediction Model

In this section, we will present details on the architecture of the human motion prediction model. We use the RNN with LSTM structure to capture the temporal coherence of the human motion during collaborative disassembly tasks due to the advantage of LSTM gates in storing and forgetting motion's characteristics.

Fig. 3 illustrates the prediction framework, including the structure of the LSTM cell. The cell state c memorizes the motion information and is updated iteratively. The sigmoid function σ and the tanh function tanh are applied to determine what motion characteristics need to be ignored, added, or outputted. The

hidden state h is the input of a fully connected layer (FCL). The future human motion $\hat{\mathcal{O}}_t$ is obtained using the following equation:

$$\hat{\mathcal{O}}_t = \text{LSTM}(\mathcal{O}_t, \mathbf{W}) \tag{2}$$

where $\mathbf{W}$ is the weights of the prediction model.

B. Uncertainty Quantification of Prediction Model Based on Bayesian Approximation

This section describes the details of the uncertainty quantification using MCDS. We interpret applying dropout in a neural network as a Bayesian approximation of a Gaussian process model over the network weights [29]. A Gaussian prior distribution is applied over the weights of our prediction model, i.e., $p(\mathbf{W})$, and a likelihood $p(\hat{\mathcal{O}}_t|\mathcal{O}_t, \mathbf{W})$ is defined to capture the prediction process. In this case, we could acquire the predictive distribution $p(\hat{\mathcal{O}}_t|\mathcal{O}_t)$ once the posterior distribution $p(\mathbf{W}|\mathcal{O}_t, \hat{\mathcal{O}}_t)$ is obtained. Unfortunately, the posterior $p(\mathbf{W}|\mathcal{O}_t, \hat{\mathcal{O}}_t)$ is intractable. But we could use a variational inference to approximate a distribution $q(\mathbf{W})$ that is close to the true posterior distribution, and this is achievable by minimizing the Kullback–Leibler divergence between these two distributions. To connect the approximation inference with dropout training, the variational distribution $q(\mathbf{W})$ for each layer of the prediction model is defined with some parameters that can be optimized. Sampling from $q(\mathbf{W})$ would be equivalent to applying dropout on each layer of the prediction model. Considering that the network training is also beneficial for optimizing $q(\mathbf{W})$ [30], the prediction model uncertainty at test time using MCDS is denoted as follows [25], [29]:

$$\mathbf{u} \approx \frac{1}{K}\sum_{k=1}^{K} \text{LSTM}(\mathcal{O}_t, \overline{\mathbf{W}}_k)^T \text{LSTM}(\mathcal{O}_{t,k}, \overline{\mathbf{W}}_k) - E^T E \tag{3}$$

where $\mathbf{u} = [u_1, \ldots, u_m, \ldots, u_M]$ is the predictive variance, which reveals the less confidence of the prediction and is regarded as the uncertainty of the prediction model, $\overline{\mathbf{W}}$ represents the model weights, which are based on a certain dropout probability and fitted to $q(\mathbf{W})$, K is Monte Carlo sampling size, $\mathcal{O}_{t,k}$ indicates the kth sample at time t, and E stands for the predictive mean, where $E \approx \frac{1}{K}\sum_{k=1}^{K}\text{LSTM}(\mathcal{O}_t, \overline{\mathbf{W}}_k)$.

Fig. 4 shows the framework of the uncertainty quantification. First, the observed human motion at time t is the input of the motion prediction model. Next, an LSTM-based prediction model outputs K samples using MCDS. Then, we quantify the variance of the future bone directions. Eventually, we use the predictive mean E to stand for the predicted arm poses, and the quantified uncertainty is represented by the blue dangerous area around the human arm. The next m-step uncertainty-informed prediction at time t is denoted as follows:

$$\hat{\mathcal{O}}_m^{t,u} = H^*(\hat{v}_{t+m}^a, \hat{v}_{t+m}^b, p_h, u_m, \beta)$$

where $H^*(\bullet)$ is the uncertainty-informed mapping function, and β is a user-defined z-score [31] used in experimental tests. Theoretically, the quantified dangerous area would disappear if the prediction model uncertainty is zero (i.e., $\overline{\mathbf{W}}_k$ are the same all the time). However, in real cases, the essential point of the uncertainty quantification, i.e., model weights $\overline{\mathbf{W}}_k$, cannot be constant due to the random dropout. Therefore, the uncertainty is necessary to be quantified and embraced into robot motion planning.

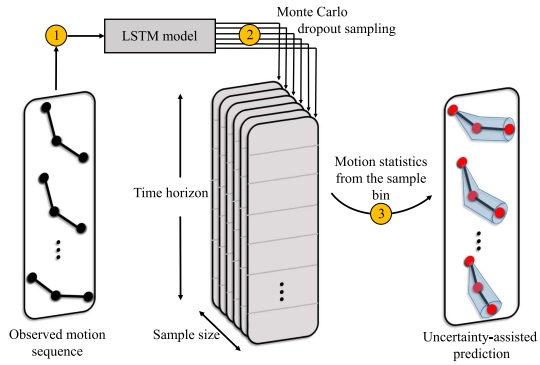

Fig. 4. Uncertainty quantification of the human prediction model and refinement of human motion prediction based on the modeling uncertainty: the black dots indicate the observed joint positions, the red dots stand for the mean predicted joint positions, and the blue areas are the dangerous zones defined based on the network uncertainty.

IV. TC-Planner: Manipulator Reconfiguration

This section presents the details of our reconfiguration algorithm. We 1) employ forward kinematics to construct a database that is capable of providing configuration candidates based on the end-effector's position, and 2) integrate the uncertainty-informed future motion into a reconfiguration algorithm such that the robot can take reactions in advance.

A. Reconfiguration Database Construction

In this section, we describe how the reconfiguration database is constructed. A manipulator can reach the same end-effector's position with different (infinite) configurations due to its kinematic redundancy. We utilize such a property of the manipulator to construct a database offline. The database is capable of providing a set of configuration candidates based on the end-effector's position x_t in real time when facing a potential collision.

To generate sufficient robot configurations, we first limit each joint's range, and then divide each joint range based on a specific interval δ. In this case, the total number of the generated configuration depends on the value of δ. Equation (1) is applied to calculate the end-effector's position based on different joint angle combinations. The last link's transformation is calculated using Denavit–Hartenberg matrix. By classifying the configurations that lead to the same end-effector's position within a task space error tolerance Δ, different configuration candidate sets Θ are generated. The number of configuration candidates may be different for each end-effector's position. The database is utilized in the way shown in Fig. 5. If a potential collision is identified, the current end-effector's position x_t would be the input of the database. All the elements of the corresponding

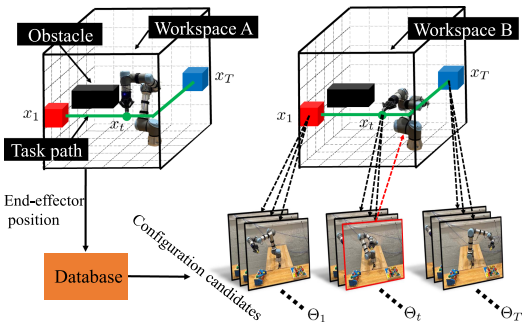

Fig. 5. Construction of the reconfiguration database: each Θ stands for the set of configuration candidates that leads to the same x in the task space, and the manipulator selects one configuration to replace the previous one.

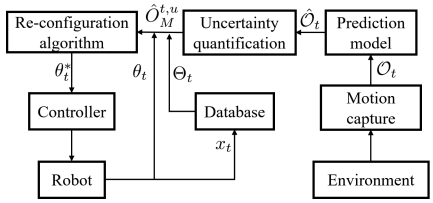

Fig. 6. Uncertainty-informed reconfiguration framework.

configuration candidates set Θ_t lead to x_t, and one of them would be the new manipulator configuration.

B. Manipulator Reconfiguration Incorporating Future Human Motion and Uncertainty

The previous section briefly introduces the construction and capability of the configuration database. The provided configuration candidates enable the manipulator to follow the given path. However, since not every configuration candidate for the human worker is safe, selecting a collision-free configuration from the database to protect the human worker is still a challenge. Therefore, we develop an algorithm to select the optimal configuration θ_t^* from Θ_t based on the joint change effort of the manipulator and the minimum safety distance α. To enable the manipulator to change the configuration in advance, the future motion obtained by the prediction model needs to be considered in the selection of the alternative configuration. In addition, given the prediction model may show less confidence (i.e., high predictive variance $\mathbf{u}$) in forecasting arm's motion, the quantified uncertainty is taken account into the selection as well.

Fig. 6 presents the framework of uncertainty-informed reconfiguration. The details are illustrated as follows.

1) When the reconfiguration is required, the database provides the configuration candidates Θ_t based on the current end-effector's position x_t.
2) To minimize the joint change effort, our algorithm first evaluates the root-mean-square error (RMSE) between the risky joint configuration θ_t and each element of Θ_t, and next rank elements of Θ_t starting from the candidate with the minimum RMSE.

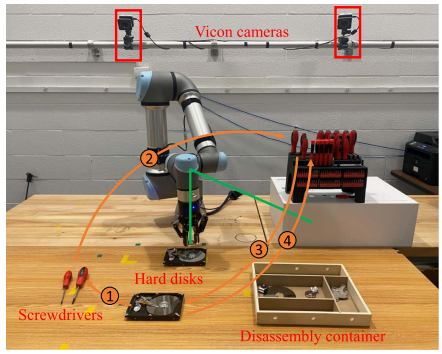

Fig. 7. Experimental setup. The experimental test consists of a collaborative robot UR5e, a human operator, a Vicon motion capture system, two used hard disks, a tool box, and a container. The human operator's motion is represented by the orange curves: 1) grasping one screwdriver from the desk, 2) moving over the robot and dropping the screwdriver to the tool box, 3) picking a new screwdriver and disassembling the hard disk on the desk, and 4) returning the screwdriver back to the tool box. Simultaneously, the manipulator picks a disassembled component and places it to the container following the given green path. The Vicon cameras are used to capture human motion in real time.

3) Simultaneously, the prediction model takes the observed motion $\mathcal{O}_t$ at the current time step, and generates the predicted motion $\hat{\mathcal{O}}_t$.
4) By applying MCDS, the predictive variance $\mathbf{u}$ regarding as the prediction uncertainty is quantified based on the Bayesian inference, and converted to the dangerous zone around the future human arm pose.
5) Eventually, our algorithm successively checks the safety of elements in Θ_t using the dangerous area of the next M step (i.e., $\hat{O}_M^{t,u}$). The optimal configuration θ_t^* is selected as long as the safety distance requirement is satisfied, i.e.,

$$d(\mathcal{M}(\theta_t^*), \hat{O}_M^{t,u}) > \alpha. \tag{4}$$

Algorithm 1 describes detailed implementation steps of our algorithm. Overall, the proposed algorithm has two advantages: 1) Our algorithm selects the robot configuration based on the end-effector's position, which inherently satisfies the task constraint. 2) The efficiency is improved by eliminating the necessity of the online task-configuration-space conversion such that the expensive conversion cost is replaced by the cheap selection cost.

V. Experimental Tests and Results

A. Experimental Setup

1) Experimental Platform: Fig. 7 illustrates the experimental setup and a disassembly scenario, in which a human operator is doing tasks following the orange curves, and a collaborative robot, UR5e, is doing tasks following the green end-effector's path. We use the Vicon camera system to capture the movement of the human arm. The minimum safety distance is defined as $\alpha = 0.01$ m.

2) Prediction Model Training and Uncertainty Quantification: We collected 130 trajectories for the human arm motion. All trajectories are converted to normalized bone vectors. In total, 70% is used to train the RNN-based prediction model, 15% is

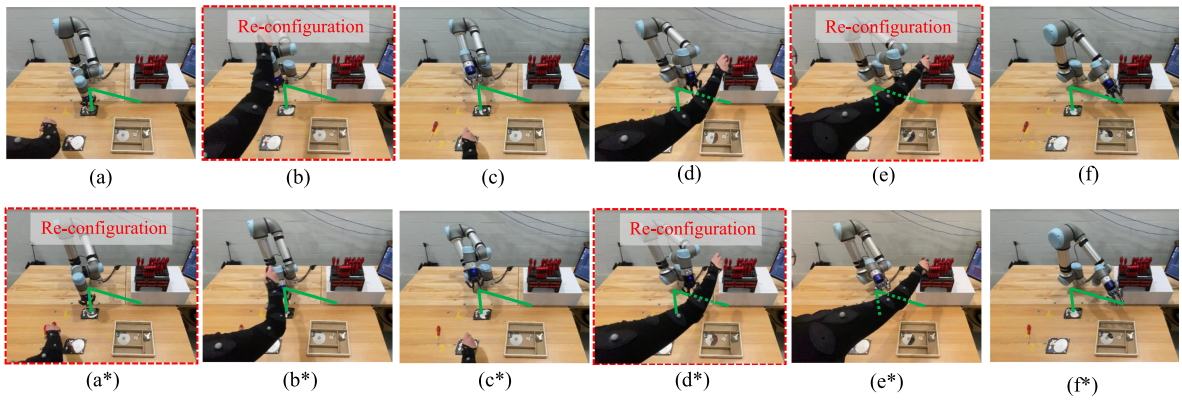

Fig. 8. Experimental test result. The subfigures in the top show the task execution based on the current human arm motion (TE-A), (b) shows the human arm is over the robot while the robot moves vertically, which triggers the first reconfiguration, and (e) displays the robot moves horizontally to the human arm, which triggers the second reconfiguration; the subfigures in the bottom present the task execution based on the future uncertainty-informed arm motion (TE-B), (a*) indicates that a collision has been detected based on the future arm's position, which triggers the first reconfiguration, and (d*) indicates the start moment of the second-reconfiguration. The experimental video is available at: http://zh.eng.buffalo.edu/PaperDemo/TCPlanner.mp4.

Algorithm 1: Online Reconfiguration.

- Obtain $\hat{O}_M^{t,u}$ based on the prediction.
- Get Θ_t from the database based on x_t.
- Get the total element number of Θ_t, denoted as C.
- Calculate RMSE of each element in Θ_t.
- Rank elements of Θ_t based on the RMSE.
- Set iteration number $j = 1$.
- Assign the j th element of the ranked Θ_t as a temporal θ_t^*.

while $d(\mathcal{M}(\theta_t^*), \hat{O}_M^{t,u}) \leq \alpha$ **do**
- Increment j.
- **if** $j \leq C$ **then**
  - Assign the j th element of the ranked Θ_t as a temporal θ_t^*.
- **else**
  - No alternative configuration exist.
- **end**

end

- Return the optimal manipulator configuration θ_t^*.

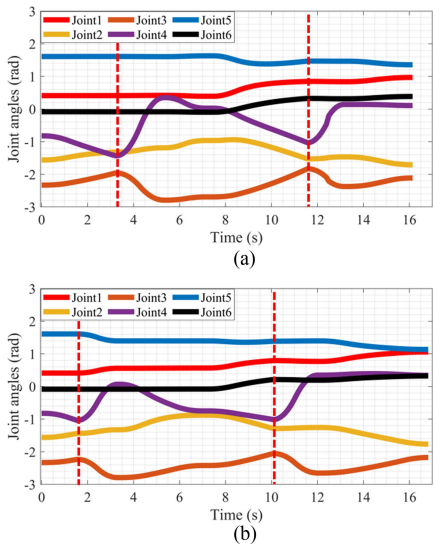

Fig. 9. Joint angle comparison between the two cases (TE-A and TE-B), and the red dash line indicates the start instant of the manipulator reconfiguration. (a) Joint angles of the UR5e during the TE-A. (b) Joint angles of the UR5e during the TE-B.

used for validation, and the remaining is used for testing. The observation horizon N and prediction horizon M are both set to be 50 implying that we predict the human motion of the next 2 s. Using MCDS with 10% dropout probability, the variance of the predicted bone vectors is obtained. We combine the variance with the anthropometric data p_h and z-score $\beta = 2$ (covering 95% of samples) to generate the dangerous area around the human arm. The dangerous area is varying during the experimental tests due to the update of the observed motion.

3) Reconfiguration Database: To construct the database offline, we first specify the joint range of the UR5e from -180° to 180°, and then determine the value of the joint change interval δ to be 4°, and the configurations that lead to the same end-effector's position based on the tolerance $\Delta = 0.01$ m are classified and stored as a cell. Note that a small δ indicates that the waypoint cell would have more configuration candidates to be selected. But if the value of δ is too small, it may require more computation when constructing the configuration database offline and the huge database may not be loaded due to the limitation of the computer memory. Therefore, we choose $\delta = 4°$. In this case, the average number of configuration candidates in the waypoint cell is more than 1000, and the database can be successfully loaded when doing online planning. In addition, to reduce database size and lookup time, we specify the range of the end-effector's position based on the specific task, allowing for the elimination of unneeded cells. Eventually, a total of 357 911 waypoint cells are included in the final database, and the final database size is 6.9GB.

B. Experimental Test Results

1) Task Execution Comparison: The manipulator is executing a one-piece disassembly task on a used hard disk. To better explain the experimental test results, we denote the task execution based on the current arm motion as task execution (TE)-A,

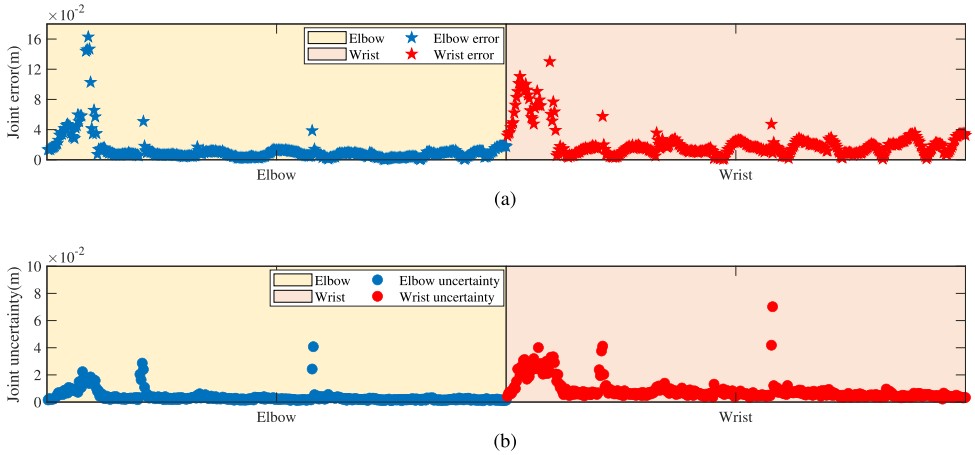

Fig. 10. Actual prediction errors and the quantified joint uncertainties, and they show high correlation. (a) Prediction errors of elbow and wrist joints. (b) Quantified uncertainties of elbow and wrist joints prediction.

TABLE I
DETAILS OF TC-PLANNER COMPUTATIONAL PROCESS

Task	Solution time/s
Prediction with MCDS (K=5)	0.23
Prediction with MCDS (K=10)	0.48
Prediction with MCDS (K=20)	1.12
Waypoint cell lookup	2.56×10^{-3}
RMSE calculation and rank	1.35×10^{-4}
Solution search upon safety-checking	1.51×10^{-3}

and the task execution based on the future uncertainty-informed arm motion as TE-B. TE-A is illustrated in Fig. 8(a)–(g). The manipulator first starts changing its configuration in Fig. 8(b) as the manipulator moves to the arm vertically. The second reconfiguration is shown in Fig. 8(e) in which the manipulator moves to the arm horizontally. TE-B is illustrated in Fig. 8(a*)–(g*). The subfigure (a*) indicates the start moment of the first reconfiguration since the collision has been detected based on the future arm's position, and the subfigure (d*) shows the moment of the second reconfiguration. In general, the manipulator is capable of responding earlier to avoid close proximity by integrating future human motion into the robot motion planning.

The average computational time of each section in TC-Planner is given in Table I. We use the Flexible Collision Library in MoveIt to quickly provide collision-checking results, and our reconfiguration algorithm only requires a cheap selection cost since the connection between the task space and the configuration space is built by the database offline. Table I also gives the computational time of the uncertainty quantification based on different values of sampling size K. A larger K (e.g., $K = 20$) implies that more prediction samples would be covered in the uncertainty quantification, which enhances the accuracy of the uncertainty estimation during the online planning. But the corresponding computational time would also increase (e.g., 1.12 s), which limits the efficiency of the online planning. Therefore, we select $K = 10$ to balance the accuracy of the prediction estimation and the computational cost. The 0.48 s computational time could be appropriately utilized in robot motion planning since the prediction horizon is 2 s. Fig. 9(a) and (b), respectively, presents the joint angles of the UR5e during TE-A and TE-B. The red dash line indicates the start instant of the manipulator reconfiguration. With human motion prediction, the manipulator is able to act in advance, i.e., 1.9 instead of 3.4 s and 10.1 instead of 11.6 s.

In general, as long as the prediction model is trained based on diverse motions and the training hyperparameters are well tuned, the network prediction and the quantified uncertainties could be reasonably accurate and reliable when facing unseen human motions. Therefore, the proposed method could still be utilized to plan collision-free manipulator trajectories based on different human motions in human–robot collaboration scenarios.

2) Prediction Errors and Uncertainties Comparison: To better understand and highlight the significance of the uncertainty quantification, we compare the joint uncertainties and the prediction errors regarding the positions of the elbow and wrist joints. The joint uncertainty is defined as the β-based variance with respect to the mean predicted joint position, and the prediction error is defined as the distance between the mean predicted joint position and the ground truth.

Fig. 10 illustrates the prediction errors and the joint uncertainties of eight motions (400 arm poses) randomly selected from the test dataset. The prediction error and joint uncertainty are represented using star and dot symbols, and they are highly related to each other. Actually, given that we cannot obtain future ground truth of arm pose at the current time step, it is impossible to obtain the prediction error during a task execution. On the other hand, the uncertainty can serve as an alternatively useful information due to the high correlation of them. Therefore, it is beneficial for quantifying the uncertainty of the prediction model to prevent human from physical injury from those predicted arm poses with less accuracy.

VI. CONCLUSION

This article presents a new TC-Planner for the manipulator such that it can work closely, safely, and collaboratively with the human operator on the disassembly of high-precision electronic devices. The TC-Planner explicitly incorporates the human motion prediction model and the modeling uncertainties into our manipulator reconfiguration algorithm, and efficiently generates

collision-free motions while keeping the original end-effector's path. The quantified uncertainty can be an additional safety indicator in HRC. Such a planner is experimentally validated in a collaborative disassembly scenario: it shows that the robot can respond to the human's motion earlier, and avoid collision with human with enlarged safety margin, while simultaneously working on disassembly tasks.

Considering that the current and future human arm positions can be, respectively, obtained by the Vicon system and the prediction model, future studies can focus on developing repulsive potential fields based on the danger zone, and incorporating the developed repulsive potential fields into the online planning of the robot trajectory to guarantee the safety in human–robot collaboration scenarios.

REFERENCES

[1] M.-L. Lee, S. Behdad, X. Liang, and M. Zheng, "Task allocation and planning for product disassembly with human–robot collaboration," *Robot. Comput. Integr. Manuf.*, vol. 76, 2022, Art. no. 102306.

[2] W.-Y. Zhao, S. He, C. Wen, and C. Liu, "Contact-rich trajectory generation in confined environments using iterative convex optimization," in *Proc. Dyn. Syst. Control Conf.*, 2020, Art. no. V002T31A002033604.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[4] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[5] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Comput. Sci. Dept., Iowa State Univ., Ames, IA, USA, Tech. rep. TR 98-11, 1998.

[6] J. Schulman et al., "Motion planning with sequential convex optimization and convex collision checking," *Int. J. Robot. Res.*, vol. 33, no. 9, pp. 1251–1270, 2014.

[7] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 3074–3081.

[8] Z. Kingston, M. Moll, and L. E. Kavraki, "Sampling-based methods for motion planning with constraints," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 159–185, 2018.

[9] R. Liu, R. Chen, and C. Liu, "Task-agnostic adaptation for safe human-robot handover," 2022, *arXiv:2209.09418*.

[10] W. Wang, M. Zhu, X. Wang, S. He, J. He, and Z. Xu, "An improved artificial potential field method of trajectory planning and obstacle avoidance for redundant manipulators," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 5, 2018, Art. no. 1729881418799562.

[11] X. Meng and R. Weitschat, "Dynamic projection of human motion for safe and efficient human-robot collaboration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 3765–3771.

[12] Y. Wang, Y. Sheng, J. Wang, and W. Zhang, "Optimal collision-free robot trajectory generation based on time series prediction of human motion," *IEEE Robot. Automat. Lett.*, vol. 3, no. 1, pp. 226–233, Jan. 2018.

[13] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5226–5234.

[14] W. Mao, M. Liu, M. Salzmann, and H. Li, "Multi-level motion attention for human motion prediction," *Int. J. Comput. Vis.*, vol. 129, no. 9, pp. 2513–2535, 2021.

[15] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4346–4354.

[16] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep learning on spatio-temporal graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5308–5317.

[17] W. Liu, X. Liang, and M. Zheng, "Dynamic model informed human motion prediction based on unscented Kalman filter," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 5287–5295, Dec. 2022.

[18] Y. Cheng, L. Sun, C. Liu, and M. Tomizuka, "Towards efficient human-robot collaboration with robust plan recognition and trajectory prediction," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2602–2609, Apr. 2020.

[19] P. Kratzer, M. Toussaint, and J. Mainprice, "Prediction of human full-body movements with motion optimization and recurrent neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1792–1798.

[20] M. Faroni, M. Beschi, and N. Pedrocchi, "Safety-aware time-optimal motion planning with uncertain human state estimation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12219–12226, Oct. 2022.

[21] A. Kanazawa, J. Kinugawa, and K. Kosuge, "Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 817–832, Aug. 2019.

[22] D. Fridovich-Keil et al., "Confidence-aware motion prediction for real-time collision avoidance1," *Int. J. Robot. Res.*, vol. 39, no. 2-3, pp. 250–265, 2020.

[23] S. Sajedi, W. Liu, K. Eltouny, S. Behdad, M. Zheng, and X. Liang, "Uncertainty-assisted image-processing for human-robot close collaboration," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4236–4243, Apr. 2022.

[24] Y. Cheng, W. Zhao, C. Liu, and M. Tomizuka, "Human motion prediction using semi-adaptable neural networks," in *Proc. Amer. Control Conf.*, 2019, pp. 4884–4890.

[25] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[26] G. Franchi, A. Bursuc, E. Aldea, S. Dubuisson, and I. Bloch, "Encoding the latent posterior of Bayesian neural networks for uncertainty quantification," 2020, *arXiv:2012.02818*.

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6405–6416.

[28] C. Doersch, "Tutorial on variational autoencoders," 2016, *arXiv:1606.05908*.

[29] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[30] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," 2015, *arXiv:1506.02158*.

[31] D. G. Zill, *Advanced Engineering Mathematics*. Maynard, MA, USA: Jones & Bartlett Publishers, 2020.

Wansong Liu received the B.S. degree in materials processing and control engineering from the China University of Mining and Technology, Xuzhou, China, in 2017, and the M.S. degree in 2020 in mechanical and aerospace engineering from the University at Buffalo, Buffalo, NY, USA, where he is currently working toward the Ph.D. degree in mechanical engineering.

His research interests include planning, learning, and the control of robotic manipulators in collaboration with human.

Xiao Liang received the B.S. degree from Hunan University, Changsha, China, in 2010, and the M.S. and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA, in 2011 and 2016, respectively, all in civil engineering.

In 2018, he joined the Department of Civil, Structural, and Environmental Engineering, University at Buffalo, Buffalo, NY, USA. His research interests include health monitoring and autonomous inspection of infrastructure systems through advanced data analytics, model based, and machine learning.

Minghui Zheng received the B.S. degree in engineering mechanics and the M.S. degree in control science and engineering from Beihang University, Beijing, China, in 2008 and 2011, respectively, and the Ph.D. degree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2017.

In 2017, she joined University at Buffalo, Buffalo, NY, USA, where she is currently an Assistant Professor of mechanical and aerospace engineering. Her research interests include learning, planning, and control with applications to several areas that are of vital importance to manufacturing, and robotics.

Dr. Zheng was the recipient of the NSF Career Award in 2021.