

# Adaptive Model Predictive Control for Differential-Algebraic Systems towards a Higher Path Accuracy for Physically Coupled Robots

Xin Ye<sup>1</sup>, Karl Handwerker<sup>1</sup> and Sören Hohmann<sup>2</sup>

**Abstract**—The physical coupling between robots has the potential to improve the capabilities of multi-robot systems in challenging manufacturing processes. However, the path tracking accuracy of physically coupled robots is not studied adequately, especially considering the uncertain kinematic parameters, the mechanical elasticity, and the built-in controllers of off-the-shelf robots. This paper addresses these issues with a novel differential-algebraic system model which is verified against measurement data from real execution. The uncertain kinematic parameters are estimated online to adapt the model. Consequently, an adaptive model predictive controller is designed as a coordinator between the robots. The controller achieves a path tracking error reduction of 88.6% compared to the state-of-the-art benchmark in the simulation.

## I. INTRODUCTION

Robot manipulators are becoming increasingly important in manufacturing processes. Apart from the conventional tasks, such as object handling, welding, or surface painting, it has been attempted to utilize robots in more challenging processes such as milling and sheet metal bending [1]. The goal is to replace the expensive Computerized Numerical Control (CNC) machines with robots by exploiting their versatility and low cost [17]. Due to the limitation of robots in stiffness and load capability under process forces, the idea of physically coupling multiple robots to a single end-effector (also named as coupler below) is proposed [14], as illustrated in Fig. 1. This idea inherits the research work in the cooperative manipulation [8] since the 1970s, but brings about new aspects in stiffness [23] and accuracy [22] for physically coupled robots which are essential for manufacturing processes.

Our previous work [22] analyses the cause of inaccuracy in the unknown kinematic parameters of robot base placements and the coupler geometry, as well as in the imperfect synchronization of joint trajectories. These factors lead to internal stress between the coupled robots, deformation of robots' joints, and consequently the deviation of the tool center point (TCP) from the path. As the tracking accuracy along a pre-defined path is improved by an offline compensation method using the measured data from a non-compensated trial execution in [22], neither the offline compensation nor the trial execution is needed in the present work. Instead, the accuracy enhancement is brought to a potentially online-capable control framework. In order to make our method

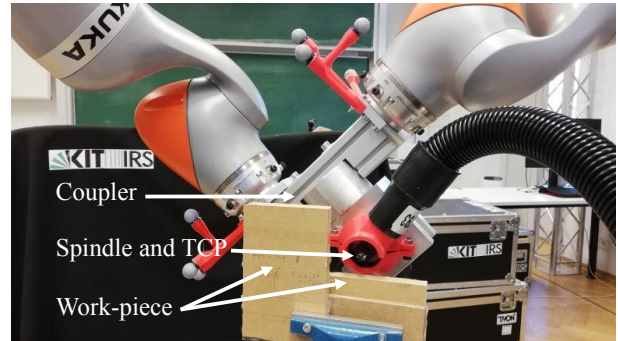


Fig. 1: Two robots are coupled to a coupler. The spindle and TCP are mounted on the coupler to achieve a high stiffness while cutting the work-piece.

applicable to the manufacturing practice, the following requirements should be satisfied:

- A given path with arbitrary complexity should be tracked by the TCP with high accuracy;
- Deviations in the joint trajectories should be corrected instantly during the execution;
- The uncertain kinematic parameters should be estimated with actual sensor measurements and updated to the utilized model for the adaptive control instantly;
- The method should be applicable to real robots with non-negligible elasticity, particularly in joints [1];
- The built-in controller of off-the-shelf robots should not be circumvented;
- Additional objectives such as the load distribution and the limitation of internal stress between robots should be handled optimally.

In the literature of control regarding cooperative manipulation, the above requirements cannot be satisfied simultaneously, because the accuracy of physically coupled robots is not studied adequately. Preliminary works [2, 13] solve the position and internal force control problem for constrained multibody systems, where passive joints or joints with elasticity are considered. These methods have not been applied to real robots because the control is not adaptable to uncertain kinematic parameters, upon which the internal force and joint loads are highly sensitive. Adaptive control is applied in [3, 9] for uncertain kinematic parameters to achieve a minimal internal stress or positioning accuracy. However, these methods have difficulty on real robots because the limited joint stiffness cannot be considered in the controllers. Black- or grey-box adaptive control methods based on neural networks [16, 21] or inverse dynamics learning [18] have the potential

<sup>1</sup>Xin Ye and Karl Handwerker are with FZI Research Center for Information Technology, 76131 Karlsruhe, Germany ye, handwerker@fzi.de

<sup>2</sup>Sören Hohmann is with Institute of Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany soeren.hohmann@kit.edu

to approximate the robots' behavior in physical coupling, but the high computational load and the limited ability of generalization still hinder their utilization. Furthermore, the aforementioned studies assume a direct access to the torque input of joints, which is only available for robot prototypes instead of off-the-shelf robots.

In view of the limitation in state-of-the-art methods, the contributions of the present work are as follows:

- A novel model of differential algebraic equations (DAE) for the physically coupled robots which accounts for the joint stiffness and the uncertain kinematic parameters
- The reduction of the differentiation order of the DAE model by considering the dynamics of the built-in impedance controller of off-the-shelf robots
- The estimation of uncertain kinematic parameters using real measurement data based on the sensitivity analysis of the DAE
- An adaptive DAE model predictive controller (MPC) that corrects the joint trajectories instantly towards a higher path tracking accuracy

## II. MODELING OF PHYSICALLY COUPLED ROBOTS

This section describes the modeling of the physically coupled robots in DAE as the basis of the adaptive MPC. The model incorporates the rigid-body dynamics (RBD) of each individual robots, the joint elasticity in addition to the RBD, the coupling constraints, as well as the characteristics of the built-in controller of an off-the-shelf robot type KUKA LBR iiwa 14 R820, abbreviated as LBR.

### A. Dynamics in the Physical Coupling

A part of the error analysis from [22] is recapitulated and slightly modified for the novel DAE model. Consider a system of  $n_R$  physically coupled robots, each having  $n_J$  joints. An example of two robots is shown in Fig. 2. The dash-double-dotted lines depict the nominal configurations of robots at the measured motor driving side joint angles  $q_{mi} \in \mathbb{R}^{n_J}, i = 1, \dots, n_R$ . Due to the elasticity and the physical coupling, the configurations at the driven link side, depicted in solid lines, are not identical to the motor driving side. The joints of robot  $i = 1, \dots, n_R$  undergo elastic deformation  $\delta q_i \in \mathbb{R}^{n_J}$  that complies to Hooke's Law by:

$$\tau_{mi} = K_J \delta q_i, \quad (1)$$

where the joint torque  $\tau_{mi} \in \mathbb{R}^{n_J}$  is measured by the torque sensors.  $K_J$  is a diagonal matrix with constant joint-stiffness parameters in diagonal entries.

The coupling compels the robots to share a common TCP frame  $\{t\}$ , which is deviated away from the reference path point  $\{r\}$ . To enhance precision, it is crucial to identify and immediately mitigate the factors contributing to this deviation. To achieve that, the source of errors, namely the uncertain kinematic parameters in the coupling, is analysed as follows. In the world frame  $\{w\}$ , we now observe robot  $i = 2, \dots, n_R$ , which corresponds to the one on the right-hand-side of Fig. 2. From the nominal placement frame  $\{b_i\}$ , the real base  $\{b'_i\}$  is deviated by a small error  $\Delta p_{bi} \in \mathbb{R}^6$

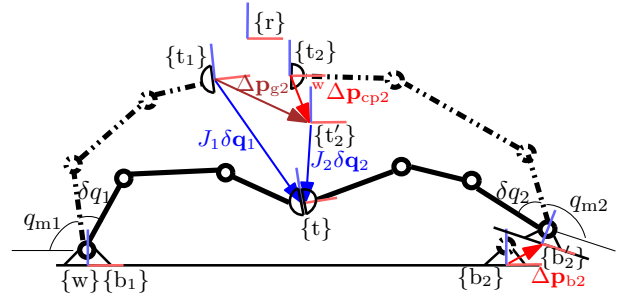


Fig. 2: The loop closure condition of a two-robot example

in rotational and translational dimensions. From  $\{b'_i\}$  to the robots' nominal TCP frame  $\{t_i\}$ , the transformation is determined by the forward kinematics  $T_{kini}(q_{mi})$ . The geometric error of the coupler  ${}^w\Delta p_{cpi} \in \mathbb{R}^6$  (after its transformation from the coupler frame constant vector  ${}^{body}\Delta p_{cpi}$  to the world frame) is superposed upon  $\{t_i\}$ , resulting in  $\{t'_i\}$ . If the uncertain kinematic parameters are summarized in the vector  $p_i = [\Delta p_{bi}^T, {}^{body}\Delta p_{cpi}^T]^T$ , then the transformation from  $\{w\}$  to  $\{t'_i\}$  is denoted as:

$${}^wT_{t'_i}(q_{mi}, p_i) = {}^wT_{b_i} {}^{b_i}T_{b'_i}(\Delta p_{bi}) T_{kini}(q_{mi}) {}^{t_i}T_{t'_i}({}^{body}\Delta p_{cpi}) \quad (2)$$

W.l.o.g., the base of the first robot  $\{b_1\}$  is defined on the origin of world frame  $\{w\}$  without placement error. Also as in [22], the first robot is the only one that is assumed to be well calibrated w.r.t. the TCP. Therefore, it applies  ${}^wT_{t1}(q_{m1}) = T_{kin1}(q_{m1})$ . As such, the gap vector  $\Delta p_{gi}(q_{m1}, q_{mi}, p_i) \in \mathbb{R}^6, \forall i = 2, \dots, n_R$  can be computed as the difference between  ${}^wT_{t'_i}(q_{mi}, p_i)$  and  ${}^wT_{t1}(q_{m1})$ , denoted as  $\Delta p_{gi} = \text{diff}({}^wT_{t'_i}, {}^wT_{t1})$  where the operator  $\text{diff}(\cdot, \cdot)$  is defined as follows:

$$\text{diff}(T_1, T_2) = \begin{bmatrix} \frac{1}{2} (R_2^T R_1 - R_1^T R_2)^{\vee} \\ t_1 - t_2 \end{bmatrix} \quad (3)$$

Whereby, the rotation matrices  $R$  and the translation vectors  $t$  are extracted from the corresponding matrix  $T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$ . According to [6, 12], the *vee map*  $\vee : so(3) \rightarrow \mathbb{R}^3$  is the inverse of the hat map for a skew-symmetric matrix. This convention is adopted to facilitate a symbolic computation in solving the subsequent estimation and control problem.

As depicted in Fig. 2, the coupling condition can finally be described by the loop closure equation:

$$J_1 \delta q_1 = \Delta p_{gi}(q_{m1}, q_{mi}, p_i) + J_i \delta q_i, i = 2, \dots, n_R \quad (4)$$

where  $J_i$  is the Jacobian matrix.

While the elasticity between the driving motor side and the driven link side of each joint is considered, the links are assumed rigid. Therefore, RBD still applies to the driven link side of the joints for each individual robot  $i = 1, 2, \dots, n_R$ :

$$M_i \ddot{q}_i + h_i + g_i = J_i^T \lambda_i + \tau_i + e_{di} \quad (5)$$

Whereby,  $q_i \in \mathbb{R}^{n_J}$  is the link side joint angle obtained from  $q_i = q_{mi} + \delta q_i$ ,  $M_i$  is the joint space inertia matrix,  $h_i \in \mathbb{R}^{n_J}$

includes Coriolis and centrifugal torques,  $g_i \in \mathbb{R}^{n_j}$  is the gravity torque.  $\lambda_i \in \mathbb{R}^6$  is the constraint wrench between the coupler and robot  $i$ .  $\tau_i \in \mathbb{R}^{n_j}$  is the torque vector provided by the motors and the corresponding reduction gears in joints, whereas torque sensors measure the vector  $\tau_{mi} = -\tau_i$ . Frictions and other errors in the dynamics are combined in  $e_{di} \in \mathbb{R}^{n_j}$ . By combining the first three terms of (5) into  $\tau_{\text{RBD}i} \in \mathbb{R}^{n_j}$  and neglecting  $e_{di}$ , one obtains:

$$\tau_{\text{RBD}i} + \tau_{mi} = J_i^T \lambda_i \quad (6)$$

The RBD of the coupler is described as follows:

$$w_{\text{cp}} = w_{\text{ext}} - \sum_{i=1}^{n_R} \lambda_i \quad (7)$$

Whereby,  $w_{\text{ext}} \in \mathbb{R}^6$  is the external process wrench exerted on TCP.  $w_{\text{cp}} \in \mathbb{R}^6$  is the combination of inertial and gravity wrench of the coupler.

Combining (1) (4) (6) and (7), one obtains the algebraic equations in matrix form:

$$Fx_a = b(p, w_{\text{ext}}), \quad (8)$$

where the algebraic variables are  $x_a = [\tau_{m1}^T, \dots, \tau_{mn_R}^T, \lambda_1^T, \dots, \lambda_{n_R}^T]^T$ . On the right hand side of (8),  $b(p, w_{\text{ext}}) = [\tau_{\text{RBD}1}^T, \dots, \tau_{\text{RBD}n_R}^T, w_{\text{cp}}^T - w_{\text{ext}}^T, \Delta p_{g2}(p_2)^T, \dots, \Delta p_{gn_R}(p_{n_R})^T]^T$ .  $p$  is a concatenation of  $p_i, i = 2, \dots, n_R$ . For matrix  $F$  in (8), only the formulation for two robots are given to keep the matrix tidy:

$$F = \begin{bmatrix} -I & O & J_1^T & O \\ O & -I & O & J_2^T \\ O & O & -I & -I \\ J_1 K_J^{-1} & -J_2 K_J^{-1} & O & O \end{bmatrix} \quad (9)$$

This matrix is easily extendable to any number of robots.

We do not build the differential equations w.r.t.  $q_i$  and its derivatives as in (5), because  $q_i = q_{mi} + \delta q_i = q_{mi} + K_J^{-1} \tau_{mi}$  is readily obtained from sensor measurements and  $\tau_{\text{RBD}i}(q_i, \dot{q}_i, \ddot{q}_i)$  is computed from the inverse dynamics. Therefore,  $\tau_{\text{RBD}i}$  are regarded as known parameters that are updated in each model prediction loop and placed in the corresponding entries of  $b$ . The same reason applies to the known parameters  $w_{\text{cp}}$ . Also  $w_{\text{ext}}$  is available in run time through a conversion from the measurements  $\tau_{mi}$ .

### B. Characterization of the Built-in Controller

Since the built-in controller of most off-the-shelf robots cannot be circumvented, it is only possible to design a superimposed controller that coordinates all the decentralized built-in controllers of each individual robot. It is therefore necessary to characterize the behaviors of the built-in controller for a complete DAE model.

For the studied LBR, three modes are available, namely joint specific position control, joint specific impedance control, and Cartesian impedance control [19]. Since the modeling in Section II-A is in the joint space, the Cartesian space control mode is rejected because it brings about complications in the transformation between the spaces. The joint

specific position control exhibits the highest control stiffness, and behaves similar to a proportional–integral–derivative (PID) controller. When there is a deviation of joint position from a given set-point and the deviation cannot be eliminated immediately due to the coupling constraint, the integral term of the PID controller accumulates, resulting in a gradual increase in motor torque output over time. Although this integral term can be seen as a state variable in the differential equation, the built-in controller does not provide its value to the user. The joint specific impedance control is chosen as the baseline controller upon which DAE-MPC is designed, because it lacks the integral term, RBD is compensated in its control law, and the stiffness and damping constants of the controller can be set by the user. The controller stiffness is limited to  $2000 \text{ Nm rad}^{-1}$  across all joints, rendering it lower than the achievable stiffness offered by the joint specific position control. However, a superposed controller is capable to achieve a higher overall controller stiffness, as long as the joint position set-point is moved in the direction against the joint torque, proportional to the effects of the external process wrench  $w_{\text{ext}}$ .

According to [19], the control law of the joint specific impedance control for robot  $i = 1, \dots, n_R$  is approximately rewritten with the symbols of the present paper as follows:

$$\tau_i = K_P(q_{\text{cmd}i} - q_{mi}) - K_D \dot{q}_{mi} + \tau_{\text{FR}li} + \tau_{\text{RBD,comp},i} \quad (10)$$

Whereby,  $K_P \in \mathbb{R}^{n_j \times n_j}$  is the diagonal controller stiffness matrix that is set to  $2000 \text{ Nm rad}^{-1}$  for all joints in the present study.  $q_{\text{cmd}i} \in \mathbb{R}^{n_j}$  is the joint position command. The damping term is a linear approximation with the constant diagonal matrix  $K_D \in \mathbb{R}^{n_j \times n_j}$ . Although the diagonal entries of  $K_D$  are variable w.r.t.  $\dot{q}_{mi}$ , the nominal values are identified as  $10 \text{ Nm/(rad/s)}$  for all joints at low and moderated magnitude of  $\dot{q}_{mi}$ .  $\tau_{\text{FR}li}$  is an optional torque overlay given by the user. The exploitation of this feature is regarded as a future work to enhance the stiffness of the multi-robot system in milling process, but  $\tau_{\text{FR}li} = 0$  is set in the current paper.  $\tau_{\text{RBD,comp},i}$  is the compensation of RBD with  $\tau_{\text{RBD,comp},i} = \tau_{\text{RBD}i}$ .

By combining (10) and (6) with  $\tau_{mi} = -\tau_i$ , the following differential equation is obtained:

$$K_D \dot{q}_{mi} = K_P(q_{\text{cmd}i} - q_{mi}) + J_i^T \lambda_i \quad (11)$$

It is further investigated how the built-in controller of each individual robot  $i = 1, \dots, n_R$  processes the joint position set-point provided by the user which is denoted as  $q_{\text{s}pi}$ . It is discovered that  $q_{\text{cmd}i}$  used in the control law (10) is not equal to the user input  $q_{\text{s}pi}$ , but results from a command filter, characterized as follows:

$$q_{\text{cmd}i} = q_{\text{s}pi} - K_C \dot{q}_{\text{s}pi} \quad (12)$$

The diagonal entries of the diagonal matrix  $K_C$  are identified in Table I.

TABLE I: Identified parameters of the command filter

Joint	1	2	3	4	5	6	7
Parameter in $10^{-3} \text{ s}$	11.5	11.4	7.8	13.1	7.2	6.8	7.1

Given that  $q_{spi}$  is transmitted to the built-in controller from the user, it is essential for a DAE-MPC solver operating from the user side to generate  $q_{spi}$  as a result. Alternatively, when the solver generates  $\dot{q}_{spi}$ , then  $q_{spi}$  can be transmitted to the built-in robot controllers immediately after an integration. For the sake of building a complete DAE model, the control input vector is selected to be  $u = [\dot{q}_{sp1}^T, \dots, \dot{q}_{spn_R}^T]^T$ . The differential variables are selected as  $x_d = [q_{m1}^T, \dots, q_{mn_R}^T, q_{sp1}^T, \dots, q_{spn_R}^T]^T$  because the derivatives of these variables occur in (11) and (12)

The differential equations can thus be formulated by combining (11) and (12) in a matrix form:

$$\dot{x}_d = Ax_d + Bu + Ex_a \quad (13)$$

The matrices in (13) are as follows for the case of  $n_R = 2$ :

$$A = \begin{bmatrix} -K_D^{-1}K_P & O & K_D^{-1}K_P & O \\ O & -K_D^{-1}K_P & O & K_D^{-1}K_P \\ O & O & O & O \\ O & O & O & O \end{bmatrix} \quad (14)$$

$$B = \begin{bmatrix} -K_D^{-1}K_C & O \\ O & -K_D^{-1}K_C \\ I & O \\ O & I \end{bmatrix} \quad (15)$$

$$E = \begin{bmatrix} O & O & K_D^{-1}J_1^T & O \\ O & O & O & K_D^{-1}J_2^T \\ O & O & O & O \\ O & O & O & O \end{bmatrix} \quad (16)$$

The DAE model of the physically coupled robots is comprised of the differential and algebraic equations (13) (8). Thanks to the built-in joint specific impedance control that compensates RBD of robots, the DAE model is of index one without any derivatives of an order higher than one.

### III. ADAPTIVE MODEL PREDICTIVE CONTROL

This section firstly handles the estimation of  $p$ . Afterwards, an MPC based on the model with the updated  $p$  is formulated and solved to provide instant correction to the joint trajectory for an enhanced path accuracy.

#### A. Estimation of the Uncertain Kinematic Parameters

The basic idea of estimating  $p$  is to compare the prediction of the DAE model with the sensor measurements of the real system and to update  $p$  iteratively. The difference from the comparison is used in the following Newton-Raphson Root-Finding Method [20]:

$$p_{k+1} = p_k - \alpha \left( \frac{\partial}{\partial p} \tau_m \right)^{-1} (\tilde{\tau}_m - \tau_m) \quad (17)$$

Whereby,  $k$  is the index of the iteration that updates  $p$ .  $0 < \alpha \leq 1$  is the step length of the update.  $\tau_m = [\tau_{m1}^T, \dots, \tau_{mn_R}^T]^T$  is a concatenation of torque measurements of all robots, whereas  $\tilde{\tau}_m$  is the prediction of the DAE model.  $\frac{\partial}{\partial p} \tau_m$  is the sensitivity matrix of  $\tau_m$  w.r.t.  $p$  which is obtained from a sensitivity analysis [4]. Among  $x_d$  and  $x_a$ , only  $\tau_m$ , a part of  $x_a$ , is compared with its model prediction  $\tilde{\tau}_m$  in (17) due

to the following reasons. In the DAE model,  $p$  only occurs in the algebraic equations (8) instead of the differential equations (13). W.r.t. changes of  $p$ , the sensitivity of  $x_a$  is much higher than that of  $x_d$  because of the high magnitude of joint stiffness  $K_J$  that only appears inside  $F$  in (8) (9). The exact values of  $K_J$  are provided in [7]. The comparison is further restricted to  $\tau_m$  within  $x_a$ , because  $\tau_m$  is measured directly by the torque sensors, which is not the case for  $\lambda$ .

#### B. Formulation of Model Predictive Control

The objective of the MPC is to enhance the path tracking accuracy. For a manufacturing process, the path is usually given in Cartesian space with up to six dimensions of translation and rotation. The path is input into the trajectory planning algorithm from the previous work [23]. It returns the optimal joint trajectories of all coupled robots without considering the uncertainties of kinematic parameters,  $p = 0$ . In the present paper, the generated joint trajectories are used as reference, denoted as  $q_{ref}$ . However, in the physically coupled configuration with uncertain kinematic parameters  $p \neq 0$ , if robot  $i$  tracks its own reference joint trajectory  $q_{refi}$ , the other robots can no longer do the same. Since robot  $i = 1$  is well calibrated and has its base on the origin, we assume that the error free forward transformation of its optimized reference trajectory  $T_{kin1}(q_{ref1})$  has been well planned in [23] and overlaps exactly with the manufacturing path, so the path accuracy problem is converted into the problem for robot  $i = 1$  to track  $q_{ref1}$  with its link side joint positions. Therefore, we define the system output of the DAE as  $y = q_{m1} + \delta q_1 = q_{m1} + K_J^{-1} \tau_{m1}$ , in matrix form as follows:

$$y = Cx_d + Gx_a \quad (18)$$

For two robots, the matrices can be shown in a tidy form:

$$C = [I \ 0 \ 0 \ 0] \quad (19)$$

$$G = [K_J^{-1} \ 0 \ 0 \ 0] \quad (20)$$

It is worth noting that selecting  $q_{ref1}$  as the only tracking target is not a master-slave control approach, because the control variables  $u$  are generated simultaneously for all robots, and the built-in controllers of robot  $i = 2, \dots, n_R$  do not follow that of robot  $i = 1$ . In this way, the physically coupled system maintains the highest stiffness against  $w_{ext}$  in manufacturing processes.

As such, the DAE-MPC with a receding horizon of  $T$  beginning from  $t_0$  can be formulated completely as follows:

$$\min_u \int_{t_0}^{t_0+T} \|u - u_{ref}\|_Q^2 + \|x_a\|_R^2 + \|y - q_{ref1}\|_P^2 dt \quad (21a)$$

$$+ V(y_{end} - q_{ref1,end})$$

s.t. DAE system dynamics (13) (8) (18)

$$\dot{q}_{min} \leq u \leq \dot{q}_{max} \quad (21b)$$

$$q_{min} \leq x_d \leq q_{max} \quad (21c)$$

$$x_d(t_0) = x_{d0} \quad (21d)$$

Whereby,  $Q, R$ , and  $P$  are weight matrices for the corresponding quadratic penalty terms, defined for any vector  $z$  and

any weight matrix  $L$  as  $\|z\|_L^2 = z^T L z$ . The trajectory error  $y - q_{\text{ref}1}$  is penalized with heavy weights because accuracy is the major objective of the controller.  $u_{\text{ref}} = \dot{q}_{\text{ref}}$  is obtained by a pre-processing of the reference trajectory  $q_{\text{ref}}$ . The penalty on  $u - u_{\text{ref}}$  serves to let the optimizer provide a solution in the vicinity of the pre-processed reference  $u_{\text{ref}}$ , instead of finding a distant one that requires an abrupt joint motion to reach. The penalty on  $x_a$  optimizes the distribution of internal and external loads among the joints and over the coupler.  $V(y_{\text{end}} - q_{\text{ref}1,\text{end}})$  is the penalty function of the tracking error at the end of the horizon  $t_0 + T$ .  $u$  is bounded by joint velocity limits (21b), and  $x_d$  is bounded by joint position limits (21c). Joint acceleration limits do not appear in the MPC, but can be taken into consideration in the trajectory planning and the built-in controllers. Finally in (21d),  $x_{d0}$  stands for the initial state of  $x_d$  that can be measured.

Problem (21) is solved with the direct collocation [5].

#### IV. RESULTS

In this section, the parameter estimation in Section III-A is tested on a pre-planned path with two physically coupled LBR robots in the real execution,  $n_R = 2$ . The adaptive MPC in Section III-B is tested in the simulation based on the results from the real execution.

##### A. Experimental Setting

As shown in Fig. 3a, the two LRB robots are rigidly connected to the half-transparently visualized coupler. The base of robot  $i = 1$  is at the origin, while robot  $i = 2$  has the nominal base position (1.365 m, 0, 0). In this preliminary study, the work-piece and the spinning spindle are not yet involved. A nominal Cartesian path with six straight segments and five corners is placed between the two robots. The corresponding reference joint trajectories are generated [23]. To investigate the response of the parameter estimation and the DAE-MPC to disturbances and vibrations both in simulation and real execution, the reference joint trajectories are superposed with a vibration in the second, fourth, and sixth segment. So the resulting modified reference joint trajectories  $q_{\text{ref}}$  with the vibration are no longer synchronized on the TCP, as shown in Fig. 3b. Meanwhile,  $q_{\text{ref}}$  incorporates rotations of the coupler in the Cartesian space as shown in Fig. 4. This is necessary because for the estimation of the rotational and translational kinematic parameters in  $p$ , the condition of persistent excitation must be fulfilled [9, 15].

In open-loop control, the built-in controllers of robots  $i = 1, 2$  receive the corresponding  $q_{\text{ref}i}$  without coordination. A cyclic internal stress will occur in the coupler due to the superposed vibration. In the closed-loop control, the adaptive DAE-MPC should achieve the coordination between the built-in controllers by processing  $q_{\text{ref}}$ , outputting  $u = \dot{q}_{\text{sp}}$ , and sending  $q_{\text{sp}}$  to the built-in controllers.

##### B. Estimation of the Uncertain Kinematic Parameters

As stated in Section II-A, the uncertain kinematic parameters only include  $p = [\Delta p_{b2}^T, {}^{\text{body}}\Delta p_{cp2}^T]^T$  in case of two robots. Considering the structure where the bases of

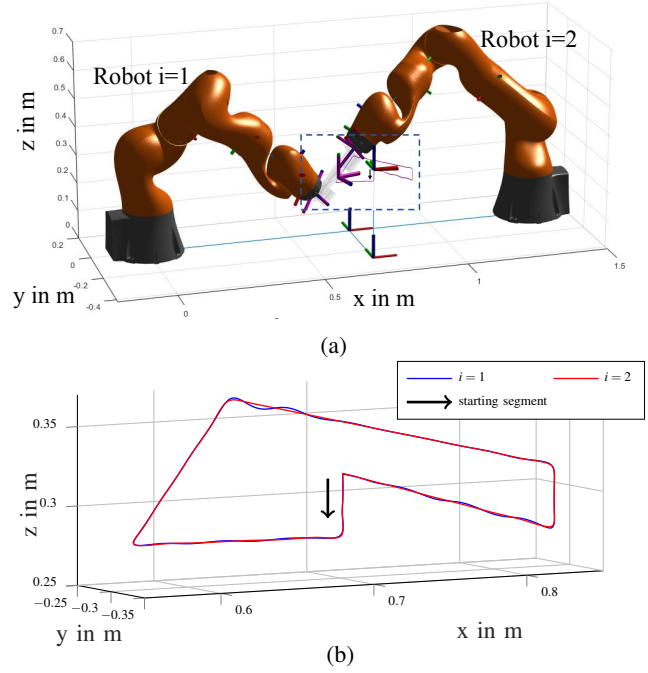


Fig. 3: Subfigure (a) shows the overall scenario of the experiment both in simulation and real execution. Subfig. (b) is the zoomed view of the enframed area in Subfig. (a). The modified reference trajectories  $q_{\text{ref}}$  for robot  $i = 1$  and  $i = 2$  are transformed into TCP positions in Cartesian space with the forward kinematics. The TCP positions start with the arrow, move clockwise, and finish a loop in 14.69 s

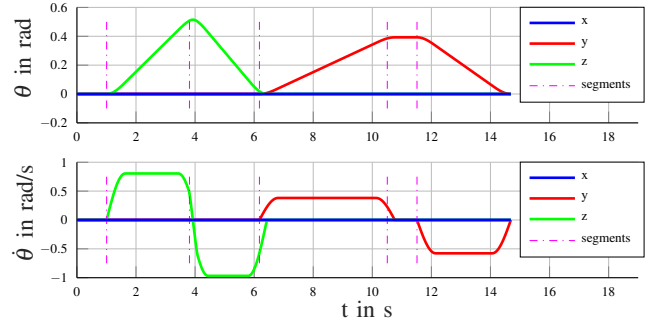


Fig. 4: Reference Euler angles  $\theta$  and angular velocities  $\dot{\theta}$  of the coupler. The dash-dotted lines are the boundaries of segments where TCP comes to corners in Fig. 3b.

the robots are mounted, the greatest uncertain kinematic parameters in  $\Delta p_{b2}$  are the translational x- and z-components which can be over millimeters, whereas the other four components of  $\Delta p_{b2}$  are at a lower order of magnitude. As of the uncertainty in the coupler geometry  ${}^{\text{body}}\Delta p_{cp2}$ , the rotational x- and z-components as well as the translational z-component are more critical. Therefore, we reduce the dimension of estimated kinematic parameters to five and leave the rest seven components zero.

The estimation method is tested in a real execution with the open-loop control without MPC. The DAE system (13) (8) is solved by the integrator IDAS [11]. To compute

the sensitivity matrix  $\frac{\partial}{\partial p} \tau_m$  in (17), IDAS provides the functionalities of sensitivity analysis within the framework of a CasADi plug-in [10]. The update of (17) is conducted at a rate of 10Hz with a step length  $\alpha = 0.1$ .

In Fig. 5, it can be seen that before the error estimation is started at  $t = 1$  s, there are significant differences between the torques from the measurement  $\tau_m$  and the model prediction  $\tilde{\tau}_m$  in both robots  $i = 1, 2$ . After starting the estimation, the model prediction converges towards the measurement within 3s. This happens because the torque differences at the beginning are used to update the kinematic parameters of the DAE model, as shown in Fig. 6.

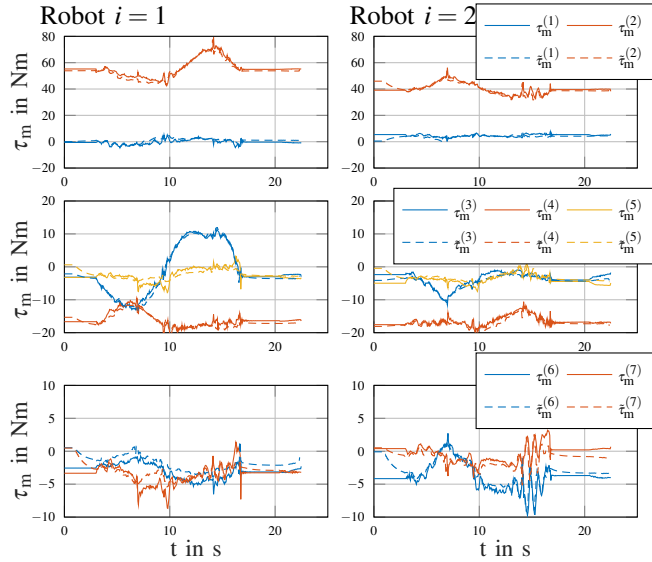


Fig. 5: Comparing the sensor measurement  $\tau_m$  and model prediction  $\tilde{\tau}_m$ . They converge after starting the estimation of uncertain kinematic parameters from  $t = 1$  s. Joint movements start at  $t = 3$  s. In the legends, the joint numbers are given in superscript brackets.

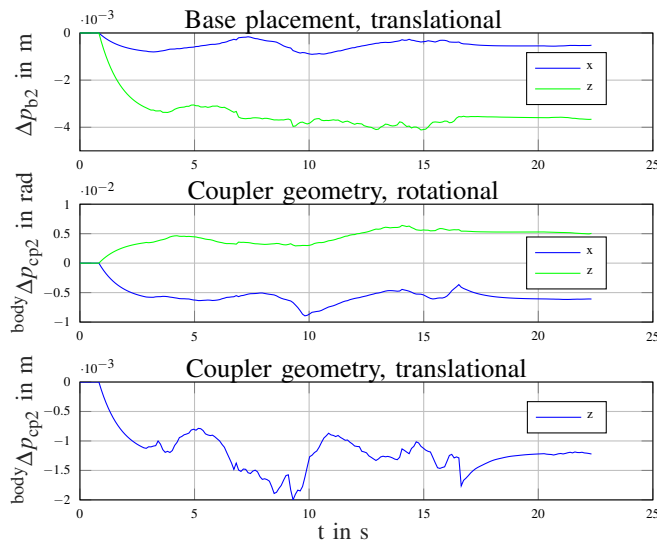


Fig. 6: Estimation of rotational and translational components of  $\Delta p_{b2}$  and  ${}^{\text{body}}\Delta p_{cp2}$ , starting from  $t = 1$  s

From Fig. 5 it can be further observed that the joint

torques from the measurement and the model prediction exhibit the same trend and overlap mostly. The proposed DAE model (13) (8) simulates the load behavior of the real multi-robot system at a decent quality after an effective parameter estimation. However, the discrepancy in torques does exist as the measurement skips up and down over the model prediction. This phenomenon occurs with a consistent pattern, only when the corresponding joint velocity changes direction. Therefore, the skipping of the torque measurement is attributed to the Coulomb friction in the joints that is neglected in the model. Further causes of the discrepancy include the neglected elasticity of the coupler and the robotic links. All the above simplifications in the model limit its prediction ability to the quasi-statics region of the system dynamics. This leads to the fluctuation of the estimated parameters in Fig. 6. For a more powerful parameter estimation, one can either reduce the step length  $\alpha$  or perform a dynamic identification of the robots so as to obtain more accurate RBD to be used in vector  $b(p, w_{\text{ext}})$  of (8).

### C. Adaptive Model Predictive Control

With the parameter estimation in place, the adaptive MPC is simulated on  $q_{\text{ref}}$  that has been superposed with vibrations as in Fig. 3b. The resulting joint torques are shown in Fig. 7. Before starting the adaptive MPC at  $t = 1$  s, the joint torques are close to the measured values in Fig. 5. But afterwards, a more even distribution is achieved. A significant load reduction can be observed among the joints that originally bear high torques. More torques are shifted to joint 1, 6, and 7 which originally bear smaller loads. Through a further comparison between Fig. 5 and Fig. 7, the adaptive MPC has a damping effect on the cyclic loads that are originally induced by the superposed vibration of  $q_{\text{ref}}$ .

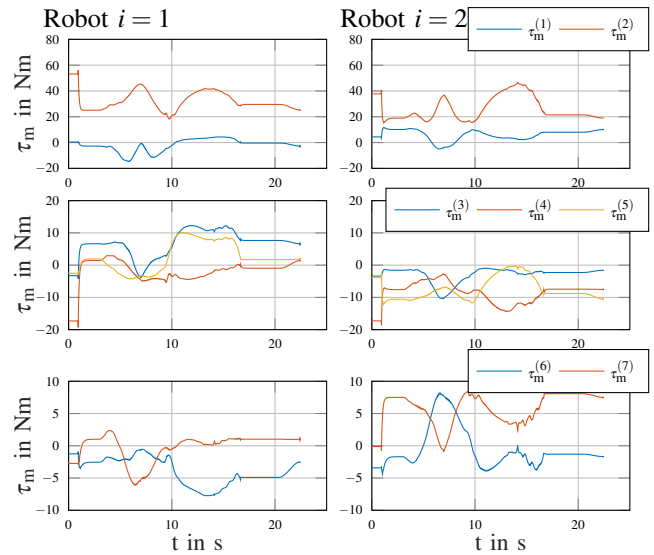


Fig. 7: Joint torques with the adaptive MPC started at  $t = 1$  s and the joint movements started at  $t = 3$  s.

The multi-robot system is able to track the still fluctuating  $q_{\text{ref}}$  with a high accuracy, as shown in Fig. 8. As the

superposed vibration is at the highest amplitude of 2.4 mm at  $t = 9.75$  s, shown at the beginning of the fourth segment in Fig. 3b, the greatest error of the adaptive MPC appears. Its translational component is 0.463 mm which is reduced by 84.8% compared to the open-loop control, and the rotational component  $8.93e-3$  rad is reduced by 91.6%. Over the whole trajectory, the average translational error is 0.0663 mm, and rotational  $0.159e-3$  rad, with a reduction of 93.5% and 96.5% respectively. The only state-of-the-art benchmark of a comparable scenario is the previous work [22], compared to which the average translational error is reduced by 88.6%.

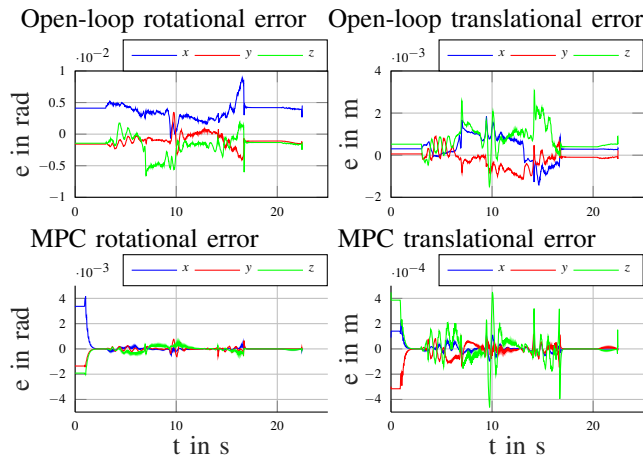


Fig. 8: Rotational and translational errors of TCP under open-loop control and closed-loop adaptive MPC.

Although the adaptive MPC shows high performance in the simulation, an implementation on the real robots is not yet accomplished due to the real-time capability issue of the collocation method on the processor Intel Core i7-10700K CPU@3.80GHz $\times$ 16. For a horizon of  $T = 0.1$  s and a step interval of 0.01 s, the collocation is solved in 0.404 s in Matlab and 0.294 s in C++.

## V. CONCLUSION AND OUTLOOK

The present work proposes an adaptive DAE-MPC method to achieve a high path accuracy online. The DAE model incorporates the uncertain kinematic parameters, the limited stiffness of robots, and the built-in robot controller that cannot be circumvented. The uncertain kinematic parameters are estimated effectively with measurement data from real execution, resulting in a successful model prediction. The MPC achieves an even load distribution and a high tracking accuracy on disturbed reference trajectories in simulation.

In the future work, a more efficient solution method should be applied to the adaptive DAE-MPC, so as to facilitate an online execution on the real robots. The torque overlay of the built-in robot controller can be exploited for a higher stiffness in manufacturing processes.

## ACKNOWLEDGMENT

The authors thank Stefan Schwab for insightful discussions, Tobias Schürmann and Max Grobbel for valuable comments and suggestions to improve the manuscript.

## REFERENCES

- [1] Eberhard Abele, Matthias Weigold, and Stefan Rothenbücher. Modeling and identification of an industrial robot for machining applications. *CIRP annals*, 56(1):387–390, 2007.
- [2] Farhad Aghili. A unified approach for inverse and direct dynamics of constrained multibody systems based on linear projection operator: applications to control and simulation. *IEEE Transactions on Robotics*, 21(5):834–849, 2005.
- [3] Farhad Aghili. Adaptive control of manipulators forming closed kinematic chain with inaccurate kinematic model. *IEEE/ASME Transactions on Mechatronics*, 18(5):1544–1554, 2012.
- [4] Joel A.E. Andersson and James B. Rawlings. Sensitivity analysis for nonlinear programming in casadi. *IFAC-PapersOnLine*, 51(20):331–336, 2018.
- [5] John T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [6] Francesco Bullo and Andrew D. Lewis. *Geometric control of mechanical systems*, 2005.
- [7] D. Busson, R. Bearee, and A. Olabi. Task-oriented rigidity optimization for 7 dof redundant manipulators. *IFAC-PapersOnLine*, 50(1):14588–14593, 2017.
- [8] F. Caccavale and M. Uchiyama. Cooperative manipulation. In *Springer handbook of robotics*, pages 989–1006. Springer, 2016.
- [9] Sebastian Erhart. *Cooperative multi-robot manipulation under uncertain kinematic grasp parameters*. PhD thesis, Technische Universität München, 2016.
- [10] Jonathan Frey, Jochem De Schutter, and Moritz Diehl. Fast integrators with sensitivity propagation for use in casadi. In *2023 European Control Conference (ECC)*, pages 1–6. IEEE, 2023.
- [11] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward. Sundials: suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):363–396, 2005.
- [12] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Geometric tracking control of a quadrotor uav on se(3). In *49th IEEE conference on decision and control (CDC)*, pages 5420–5425. IEEE, 2010.
- [13] X. Meng, M. Keppler, and C. Ott. Passivity-based motion and force tracking control for constrained elastic joint robots. *IEEE Control Systems Letters*, 7:217–222, 2022.
- [14] E. Mühlbeier, P. Gönnheimer, L. Hausmann, and J. Fleischer. Value stream kinematics. In *Congress of the German Academic Association for Production Technology*, pages 409–418. Springer, 2020.
- [15] Kumpati S. Narendra and Anuradha M. Annaswamy. *Stable adaptive systems*. Courier Corporation, 2012.
- [16] Vikas Panwar, Naveen Kumar, Nagarajan Sukavanam, and Jin-Hwan Borm. Adaptive neural controller for cooperative multiple robot manipulator system manipulating a single rigid object. *Applied Soft Computing*, 12(1):216–227, 2012.
- [17] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. R. Hansen, et al. Smerobotics: smart robots for flexible manufacturing. *IEEE Robotics & Automation Magazine*, 26(1):78–90, 2019.
- [18] M. Reuss, N. van Duijkeren, R. Krug, P. Becker, V. Shaj, and G. Neumann. End-to-end learning of hybrid inverse dynamics models for precise and compliant impedance control. *arXiv preprint arXiv:2205.13804*, 2022.
- [19] G. Schreiber, A. Stemmer, and R. Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE workshop on innovative robot control architectures for demanding (Research) applications how to modify and enhance commercial controllers (ICRA 2010)*, pages 15–21. Citeseer, 2010.
- [20] Endre Süli and David F. Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.
- [21] Fan Xu, Jin Wang, and Guo-dong Lu. Adaptive robust neural control of a two-manipulator system holding a rigid object with inaccurate base frame parameters. *Frontiers of Information Technology & Electronic Engineering*, 19(11):1316–1327, 2018.
- [22] Xin Ye, Manuel Schwartz, and Sören Hohmann. Enhancement of path tracking accuracy for physically coupled industrial robots by hybrid position-torque compensation. In *2023 European Control Conference (ECC)*, pages 1–8. IEEE, 2023.
- [23] Xin Ye, Manuel Schwartz, and Sören Hohmann. Stiffness optimized multi-robot behavior planning using reduced hessian method. *IFAC-PapersOnLine*, 55(38):55–60, 2022.