Efficient Formulation of Hexapod Kinematics Enabling Real Time Adaptive Feedforward Control

1st Bram Seinhorst
Precision Engineering, Engineering technology
University of Twente
Enschede, the Netherlands
b.seinhorst@utwente.nl

2nd Wouter Hakvoort
Precision Engineering, Engineering technology
University of Twente
Enschede, the Netherlands
w.b.j.hakvoort@utwente.nl

2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) | 978-1-6654-4139-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/AIM46487.2021.9517402

***Abstract*—The tracking performance of manipulators can be improved considerably by adaptive feedforward control (AFFC). However, complex kinematics hinder the application to parallel kinematic manipulators (PKMs). This paper proposes a compact and efficient formulation of the full PKM kinematics enabling real-time application of AFFC to complex PKMs. The efficient kinematic formulation is the basis for the inverse dynamics used to compute the feedforward signal. A Kalman filter is used for online estimation of the parameters in the equations of motion. A parallel multi-rate implementation is used, which, together with the efficient kinematic formulation, allows for a feedforward sampling time as low as** $0.5\,\mathrm{ms}$. **The parameters are updated every** $30\,\mathrm{ms}$, **which suffices to track the slow parameter variations. The application to a highly repeatable flexure-based manipulator is considered. Experimental results for the manipulator show that the tracking error can be reduced by** 97.5% **compared to using feedback control only.**

I. INTRODUCTION

Parallel kinematic manipulators (PKMs) have several advantages over their serial counterparts due to their high rigidity and low moving mass [1]. The low moving mass typically allows for high accelerations and a relatively high payload mass. Recently the T-Flex, a fully flexure-based 6 degree of freedom (DoF) PKM, was developed [2]. The use of flexures instead of conventional bearings results in a highly repeatable system. Exploiting the high repeatability of the manipulator to improve its tracking accuracy would extend its applicability.

Accurate tracking for high accelerations can be realised by feedforward control using the inverse dynamics model. A variable payload mass can be handled by adaptive feedforward control (AFFC) [3]. It has been shown experimentally that AFFC improves the tracking of PKMs [4]. A complication in the application of feedforward control and real-time adaptation is the complex kinematics of PKMs; the use of multiple kinematic chains results in complex, or even absent, closed form kinematic expressions. Oftentimes, the required modelling effort is avoided by residing to non-physics based models for AFFC [5]. Nonetheless, a physics based model typically requires fewer parameters to model the 6DoF dynamics and thereby it is less prone to overfitting and better capable to extrapolate training data. Several examples of model-based real time AFFC applied to PKMs are available [4], [6], [7]. However, in these applications real time computation of the inverse dynamics typically requires significant simplifications of the actuator dynamics or neglecting dynamics of intermediate bodies. For a manipulator with highly reapeatable behaviour, such as the T-flex, these simplifications or neglected dynamics significantly limit the accuracy of the inverse dynamics and thereby limit the achievable tracking error. Moreover, in aforementioned publications systems with only 3 to 4 DoFs are considered. No applications of AFFC to 6DoF PKMs have been found in literature.

In this paper we present an efficient formulation of the hexapod kinematics, without simplification or neglected dynamics, which is used in a real-time adaptive feedforward control implementation for the T-flex manipulator. The dynamics of the manipulator are modelled using a rigid multibody approach, which employs Euler parameters (quaternions) [8] and solves the kinematics of each arm analytically, while the kinematic loop closure is solved numerically. This formulation allows computation of the kinematics at a high sampling rate, where the numerical correction step for loop closure provides accurate results with one iteration only. Multi-rate parallel computing is used to separate the feedback, feedforward and adaptive controllers and run each at the highest possible sample rate. The feedback controller and feedforward run at high sampling rates for stability and tracking purposes, while slow-parameter changes allow running the computationally heavy adaptation at lower sampling rate.

The use of very accurate absolute encoders in the T-Flex allows a simple filtering step to estimate the velocities and accelerations, allowing straightforward application of the classical version of adaptive feedforward control [3]. Although the T-Flex has no friction due to its flexure joints, they do introduce stiffness in the dynamics. Furthermore the actuators suffer from cogging and hysteresis. All these parameters and the relevant inertial parameters of each arm are estimated simultaneously.

After this introduction, the paper starts with the kinematic modelling in section II, which is used to formulate the parameter linear inverse dynamics in section III. In the subsequent section IV the multi-rate implementation, including the numerical solution of the kinematics, is described. Section V presents and discusses the experimental results. The paper ends with the conclusions in section VI.

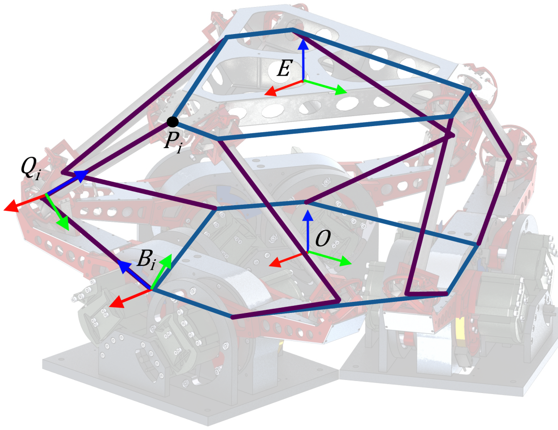

Fig. 1. Kinematic diagram of the T-flex manipulator. The reference frames B_i and Q_i are attached to the upper and lower arm respectively. The end-effector E is attached to the lower arms in the points P_i.

II. MANIPULATOR KINEMATICS

In this section the formulation of the kinematics of the hexapod-like PKM is discussed. Using the kinematic structure of the manipulator, several explicit and implicit constraints are introduced. Solving all constraint equations numerically requires large numerical inverses making it computationally expensive. A full symbolic solution either does not exist or is untractable. We propose a novel hybrid approach, where the explicit constraints are used to obtain a reduced set of coordinates, reducing the numerical effort to solve the remaining implicit constraints.

A. Manipulator kinematic structure

Figure 1 shows the kinematic diagram of the utilised manipulator. The manipulator consists of a base platform O, an end-effector platform E, the upper arms B_i and lower arm Q_i, with i being an index associated with each of the six arms. O an B_i are connected by an actuated revolute joint with associated coordinate θ_i. B_i and Q_i are connected by a universal joint and Q_i and E are connected at point P_i by a spherical joint. Since the position of B_i is fixed and the orientation is fully determined by θ_i, this coordinate suffices to describe the configuration of the upper arm bodies. These angles and all coordinates associated with each of the other bodies define the coordinate set

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{x}_{Q_1} & \boldsymbol{\lambda}_{Q_1} & \dots & \boldsymbol{x}_{Q_6} & \boldsymbol{\lambda}_{Q_6} & \boldsymbol{x}_E & \boldsymbol{\lambda}_E \end{bmatrix}^T, \tag{1}$$

where $\boldsymbol{\theta} \in \mathbb{R}^6$ contains all θ_i, $\boldsymbol{x}_{Q_1} \in \mathbb{R}^3$ denotes the position of coordinate frame Q_1 and $\boldsymbol{\lambda}_{Q_1} \in \mathbb{R}^4$ denote the Euler parameters associated with the orientation of Q_1.

B. Explicit constraints

The position of the lower arm, $\boldsymbol{x}_{Q_i}$, can be expressed in terms of θ_i as

$$\boldsymbol{x}_{Q_i} = \boldsymbol{x}_{B_i} + R(\theta_i)\boldsymbol{r}_{BQ_i}, \tag{2}$$

where $R(\theta_i)$ denotes the rotation matrix associated with θ_i and $\boldsymbol{r}_{BQ_i}$ denotes the vector from B_i to Q_i expressed in B_i. Eq. (2) can be used to express the total set of coordinates $\boldsymbol{x}$ in terms of a reduced set of coordinates $\tilde{\boldsymbol{x}}$

$$\boldsymbol{x} = \mathscr{F}(\tilde{\boldsymbol{x}}), \tag{3}$$

where $\tilde{\boldsymbol{x}}$ no longer contains the coordinates $\boldsymbol{x}_{Q_i}$.

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{\theta} & \boldsymbol{\lambda}_{Q_1} & \dots & \boldsymbol{\lambda}_{Q_6} & \boldsymbol{x}_E & \boldsymbol{\lambda}_E \end{bmatrix}^T. \tag{4}$$

C. Implicit constraints

Three so-called deformation coordinates $\boldsymbol{\varepsilon}_u(\tilde{\boldsymbol{x}}) \in \mathbb{R}^3$, associated with the bending and twist deformations of the universal joint, can be defined [9]. The full formulation of these deformations is not repeated here for the sake of brevity. The universal joint constraint is then simply expressed by considering a zero twist rotation, resulting in the third deformation coordinate being zero

$$\varepsilon_{u3}(\tilde{\boldsymbol{x}}) = 0. \tag{5a}$$

The constraint introduced by the spherical joint at P_i can be expressed as

$$\boldsymbol{x}_E + R(\boldsymbol{\lambda}_{\boldsymbol{E}})\boldsymbol{r}_{EP_i} = \boldsymbol{x}_{B_i} + R(\theta_i)\boldsymbol{r}_{BQ} + R(\boldsymbol{\lambda}_{Q_i})\boldsymbol{r}_{QP}. \tag{5b}$$

Finally the redundancy constraints in the definition of the Euler parameters are

$$|\boldsymbol{\lambda}_{Q_i}| = 0, \quad |\boldsymbol{\lambda}_E| = 0. \tag{5c}$$

Equations (5) are contained in the set of constraint equations

$$\mathscr{D}(\tilde{\boldsymbol{x}}) = 0. \tag{6}$$

D. Solving for the coordinates $\tilde{\boldsymbol{x}}$

As explained in the next section, there is no need to explicitly compute the full set of coordinates $\boldsymbol{x}$. Furthermore, since the actuator angles $\boldsymbol{\theta}$ are measured, only the dependent coordinates, defined as the set difference between the reduced set of coordinates and the actuator coordinates $\boldsymbol{d} = \tilde{\boldsymbol{x}} \setminus \boldsymbol{\theta}$, have to be computed. Noting that $\boldsymbol{d}$ contains only 31 coordinates and since a good estimate of $\boldsymbol{d}$ is available from the previous time step, a Newton Raphson procedure can be used to solve the dependent coordinates in one iteration in real time (see subsection IV-E).

III. MANIPULATOR DYNAMICS

Using the manipulator kinematics, the equations of motion (EOM) can be formulated and projected onto the permissible degrees of freedom. Furthermore, the elastic joint forces as well as the actuator-induced cogging and hysteresis are included in the dynamics in this section.

A. Explicit projection

The equations for the inertial dynamics of a rigid body are obtained from [1], [8]. Stacking these equations for all bodies, yields the unconstrained inertial dynamics, which are expressed in terms of the full set of coordinates $\boldsymbol{x}$ as

$$M(\boldsymbol{x})\ddot{\boldsymbol{x}} + \boldsymbol{h}(\dot{\boldsymbol{x}}, \boldsymbol{x}) = \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}), \tag{7}$$

where $M(\boldsymbol{x})$ contains the inertial terms of all bodies, $\boldsymbol{h}(\dot{\boldsymbol{x}},\boldsymbol{x})$ represents the gyroscopic terms and $\boldsymbol{f}(\dot{\boldsymbol{x}},\boldsymbol{x})$ represents the gravitational forces. In section III-C and III-D the additional terms $\boldsymbol{g}$ and $\boldsymbol{\chi}$, representing the joint reaction forces and actuator induced forces will be introduced.

Using d'Alembert's principle, the EOM of one arm can be projected on the reduced set of coordinates as

$$\mathscr{F}_{\tilde{\boldsymbol{x}}}^T M \left(\mathscr{F}_{\tilde{\boldsymbol{x}}}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{\gamma}\right) + \mathscr{F}_{\tilde{\boldsymbol{x}}}^T \boldsymbol{h} = \mathscr{F}_{\tilde{\boldsymbol{x}}}^T \boldsymbol{f}, \tag{8}$$

where $\mathscr{F}_{\tilde{\mathbf{x}}} = \frac{\partial \mathscr{F}}{\partial \tilde{\mathbf{x}}}$ is the Jacobian matrix and $\boldsymbol{\gamma}$ can be computed using

$$\gamma_i = \sum_{j,k} \frac{\partial^2 \mathscr{F}_i}{\partial \dot{\tilde{x}}_j \partial \dot{\tilde{x}}_k} \dot{\tilde{x}}_j \dot{\tilde{x}}_k. \tag{9}$$

Since the formulation of the constraints is explicit, this projection can be performed symbolically. Together with the following derivatives the projected equations of motion can be rewritten as a function of the reduced set of coordinates only

$$\dot{\boldsymbol{x}} = \mathscr{F}_{\tilde{\boldsymbol{x}}}\dot{\tilde{\boldsymbol{x}}}, \tag{10}$$

$$\ddot{\boldsymbol{x}} = \mathscr{F}_{\tilde{\boldsymbol{x}}}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{\gamma}, \tag{11}$$

$$\tilde{M}(\tilde{\boldsymbol{x}})\ddot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{h}}(\dot{\tilde{\boldsymbol{x}}},\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{f}}(\dot{\tilde{\boldsymbol{x}}},\tilde{\boldsymbol{x}}), \tag{12}$$

where $\tilde{\square}$ refers to the projection.

B. Implicit projection

The implicit projection of the EOM is given analogous to the explicit projection as

$$J^T \tilde{M}\ddot{\tilde{\boldsymbol{x}}} + J^T \tilde{\boldsymbol{h}} = J^T \tilde{\boldsymbol{f}}, \tag{13}$$

$$J = \begin{bmatrix} I \\ -\mathscr{D}_{\boldsymbol{d}}^{-1}\mathscr{D}_{\boldsymbol{\theta}} \end{bmatrix}, \tag{14}$$

$$\dot{\tilde{\boldsymbol{x}}} = J\dot{\boldsymbol{\theta}}, \tag{15}$$

$$\ddot{\tilde{\boldsymbol{x}}} = J\ddot{\boldsymbol{\theta}} + \tilde{\boldsymbol{\gamma}} \tag{16}$$

$$\tilde{\boldsymbol{\gamma}} = \begin{bmatrix} \mathbf{0}, \\ -\mathscr{D}_{\boldsymbol{d}}^{-1} \sum_{j,k} \frac{\partial^2 \mathscr{D}}{\partial \tilde{\boldsymbol{x}}_j \partial \tilde{\boldsymbol{x}}_k} \dot{\tilde{\boldsymbol{x}}}_j \dot{\tilde{\boldsymbol{x}}}_k \end{bmatrix}, \tag{17}$$

with $\mathscr{D}_{\boldsymbol{d}} = \frac{\partial \mathscr{D}}{\partial \boldsymbol{d}}$. The implicit nature of $\mathscr{D}$ makes the symbolic computation of $\mathscr{D}_{\boldsymbol{d}}^{-1}$ untractable and therefore the implicit projection has to be computed in real time.

C. Elastic joint forces

The joints of the manipulator are realised using flexures, resulting in elastic force under bending. Similar to the universal joint constraint, these joint forces are best expressed in terms of the twist and bending of the spherical and universal joints [9]. The deformations of all joints are collected in the vector

$$\boldsymbol{\varepsilon} = \mathscr{E}(\tilde{\boldsymbol{x}}). \tag{18}$$

The bending forces are then given by

$$\boldsymbol{\sigma} = -K\boldsymbol{\varepsilon}. \tag{19}$$

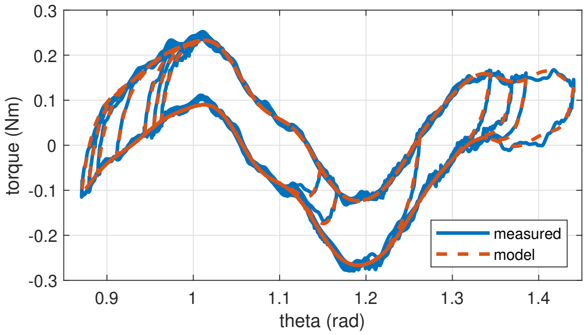

Fig. 2. Actuator torque τ plotted against the actuator angle θ. The data has been generated using a slow reference trajectory and has been compensated for linear stiffness and a constant offset. The model includes 9 cogging frequencies and a Dahl hysteresis model.

The joint reaction forces dual to $\tilde{\boldsymbol{x}}$, denoted as $\tilde{\boldsymbol{g}}$, can be computed using the coordinate transformation

$$\tilde{\boldsymbol{g}} = -\mathscr{E}_{\tilde{\boldsymbol{x}}}^T K \mathscr{E}(\tilde{\boldsymbol{x}}). \tag{20}$$

$\tilde{\boldsymbol{g}}$ can be added to the equations of motion similar to $\tilde{\boldsymbol{f}}$ as they both represent forces dual to $\tilde{\boldsymbol{x}}$

$$J^T \left(\tilde{M}(\tilde{\boldsymbol{x}})\ddot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}\,) - \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}\,) - \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}\,)\right) = \mathbf{0}. \tag{21}$$

D. Actuator dynamics

The manipulator is driven by Tecnotion ironcore motors powered by Kollmorgen drives, which do not provide ideal torque sources, since they introduce hysteresis and cogging. This is clearly visible by plotting the actual actuator torque τ versus the actuator angle θ for slow motions. Fig. 2 shows that the torque is position dependent and an offset between forward and backward movement exists. The actuator behaviour is described using

$$\begin{aligned} \chi_i(\theta_i, z_i) = & o_i + k_i\theta_i + d_i\dot{\theta}_i + c_i z_i + \\ & \sum_{n=1}^{9} a_{in}\cos(n\omega\theta) + b_{in}\sin(n\omega\theta), \end{aligned} \tag{22}$$

where o_i compensates for offset, k_i for the flexure stiffness, d_i for damping, a_{in} and b_{in} for the cogging and c_i for the hysteresis. z_i a Dahl state given by [10]

$$\dot{z} = \frac{\dot{\theta}}{\sigma}\left(1 - \operatorname{sign}(\dot{\theta})z\right), \tag{23}$$

with σ being $1.5 \cdot 10^{-2}$ rad. The model torque prediction is visualised by the dashed lines in Fig. 2.

Since $\boldsymbol{\chi}(\boldsymbol{\theta}, \boldsymbol{z})$ is already dual to the independent coordinates $\boldsymbol{\theta}$, it can simply be added to the EOM as

$$J^T \left(\tilde{M}\ddot{\tilde{\boldsymbol{x}}} + \tilde{\boldsymbol{h}} - \tilde{\boldsymbol{f}} - \tilde{\boldsymbol{g}}\right) + \boldsymbol{\chi} = \boldsymbol{\tau}. \tag{24}$$

This is the final formulation of the equations of motion.

E. Parameter linear form

It can be shown that the inertial part of the equations of motion (7) is linear in 10 inertial parameters per body [8]. The stiffness relations in (19) are linear in the stiffness parameters. Similarly, the actuator dynamics in (22) is linear in the parameters $o_i, k_i, d_i, c_i, a_{in}, b_{in}$. The projection does not change this parameter linearity, making the equations of motion (24) linear in the parameters.

Since the equations of motion are linear in the parameters, they can be rewritten into a parameter linear form

$$\Phi \boldsymbol{p} = \boldsymbol{\tau}, \tag{25}$$

where Φ is defined as

$$\Phi_{lk} = \sum_{ij} J^T_{li} \left(\frac{\partial \tilde{M}_{ij}}{\partial p_k} \ddot{x}_j + \frac{\partial \tilde{h}_i}{\partial p_k} - \frac{\partial \tilde{f}_i}{\partial p_k} - \frac{\partial \tilde{g}_i}{\partial p_k} \right) + \frac{\partial \chi_l}{\partial p_k}. \tag{26}$$

After excluding parameters that have become redundant due to the projections, 162 parameters remain.

IV. Implementation

AFFC has been implemented for the manipulator in Simulink Real-Time. The processes are executed on a Speedgoat Baseline real-time target machine which supports multi-core and multi-rate processing. Hereafter, the overall control structure is discussed first and followed by a discussion of the computational routines used for the AFFC scheme.

A. Control structure

An overview of the AFFC control structure, as implemented in Simulink, is shown in Fig. 3. P is the interface to the actuators, which outputs the measured torques $\hat{\boldsymbol{\tau}}(t) \in \mathbb{R}^6$ and actuator angles $\boldsymbol{\theta}(t) \in \mathbb{R}^6$. The input torque of P, $\boldsymbol{\tau}(t) \in \mathbb{R}^6$, is computed by the feedback controller C and the feedforward controller F. The actuator angles $\boldsymbol{\theta}(t)$ are controlled by the feedback controller C and feedforward controller F to the reference signal, denoted by $\boldsymbol{r}(t) \in \mathbb{R}^6$. First and second derivatives are denoted as $\dot{\boldsymbol{r}}(t)$ and $\ddot{\boldsymbol{r}}(t)$. The Kalman filter K estimates the parameter set $\boldsymbol{p}(t) \in \mathbb{R}^{162}$ for the feedforward controller. E computes the parameter linear form of the EOM, which is used by K. Since E and F are functions of the dependent coordinates, the Newton Raphson procedures N_F and N_E are used to determine the dependent coordinates. The filter and derivative estimator W provides the time derivatives of the actuator angles for N_E.

As can be seen in Fig. 3, the computational routines are grouped into 3 different processes and executed at different execution rates.

B. Plant communication (P)

In P the bidirectional EtherCAT communication with the Kollmorgen AKP-P00309 servo drives is implemented. The Kollmorgen drives run in current control mode and interface with the Tecnotion QTR-A-133-60-N motors as well as the Heidenhain LIC 4007-411 encoders.

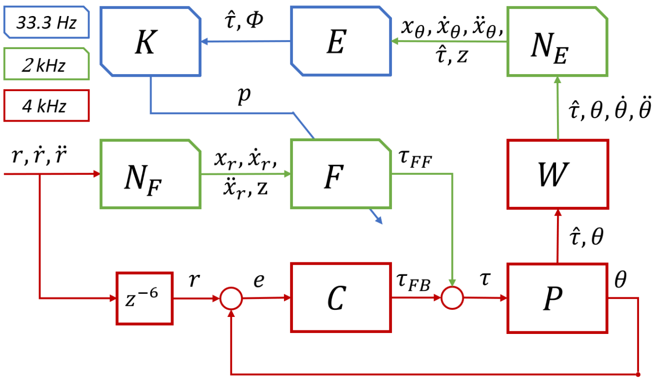

Fig. 3. An overview of all the compuational processes. P is the interface to the T-flex drives, which outputs the measured torques $\tilde{\tau}$ and angles θ. C represents the feedback controller. N_F and N_E are Newton-Raphson procedures used to determine the dependent coordinates. F computes the feedforward signal. W filters the incoming signals and estimates derivatives. E computes the regression matrix. K is a Kalman filter that estimates $\boldsymbol{p}$. z^{-6} is a delay of 1.5 ms used to synchronise the feedback and feedforward reference signals.

C. Feedback controller (C)

The 6 PID controllers, one independent controller for each actuator, are used for disturbance suppression, The PID controllers have a gain cross over frequency and phase margin of 10 Hz and 45 degrees respectively.

D. Filtering and derivative estimation (W)

In order to reduce noise, the measured $\boldsymbol{\theta}$ and $\tilde{\tau}$ are first filtered by a 101 point Hann-window based FIR filter.

$$W(z) = \sum_{n=0}^{100} sin^2 \left(\frac{\pi n}{101} \right) z^{-n} \tag{27}$$

The derivatives $\dot{\boldsymbol{\theta}}$ and $\ddot{\boldsymbol{\theta}}$ are computed using the central difference stencil $[-1, 0, 1]/(2t_s)$ and the 3 point second derivative stencil $[-1, 2, -1]/t_s^2$, with sample time t_s.

E. Newton Raphson (N_F and N_E)

The dependent coordinates $\boldsymbol{d}$ can be solved using the given $\boldsymbol{\theta}$ in real time using a Newton-Raphson (NR) procedure. The procedure starts with a time update of the dependent variables $\boldsymbol{d}$.

$$\boldsymbol{d}^t = \boldsymbol{d}^{t-1} + \dot{\boldsymbol{d}}^{t-1} \Delta t + \frac{1}{2} \ddot{\boldsymbol{d}}^{t-1} (\Delta t)^2 \tag{28}$$

After the time update, a correction step is performed such that the equations satisfy the implicit constraint equations (6).

$$\boldsymbol{d}^t \leftarrow \boldsymbol{d}^t - \mathscr{D}_{\boldsymbol{d}}^{-1} \mathscr{D} \tag{29}$$

Where $\leftarrow$ is the assignment operator. The time derivatives $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are then computed using (15) and (17). The correction step (29) can be repeated to reach better accuracy, however this comes with an greater computational cost. For the utilised reference trajectory, using only one correction step yields $||\mathscr{D}|| < 10^{-9}\,\forall t$, which is sufficiently small.

For implementation convenience the Dahl states $\boldsymbol{z}$ are also tracked by the NR procedure. The state is approximated using the following numerical approximation of (23).

$$z_i^t = z_i^{t-1} + \frac{\theta_i^t - \theta_i^{t-1}}{\sigma}\left(1 - sign(\theta_i^t - \theta_i^{t-1}) z^{t-1}\right) \tag{30}$$

F. Feedforward and regressor computation (F and E)

The equations of motion in the regular and parameter linear form have been derived in terms of a set of elementary functions, being

$$\tilde{M}(\tilde{\boldsymbol{x}},\boldsymbol{p}),\ \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}},\boldsymbol{p}),\ \tilde{\boldsymbol{h}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}},\boldsymbol{p}),\ \mathscr{D}(\tilde{\boldsymbol{x}}),\ \mathscr{D}_{\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}}),\ \mathscr{E}(\tilde{\boldsymbol{x}}),\ \mathscr{E}_{\tilde{\boldsymbol{x}}}(\tilde{\boldsymbol{x}}),\ \mathscr{S}(\boldsymbol{\varepsilon}),$$
$$\tilde{\boldsymbol{\gamma}}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}),\ \boldsymbol{\chi}(\boldsymbol{\theta},z,\boldsymbol{p}),\ \frac{\partial \tilde{\boldsymbol{f}}_i}{\partial \boldsymbol{p}_k}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}),\ \frac{\partial \tilde{\boldsymbol{h}}_i}{\partial \boldsymbol{p}_k}(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}}),\ \frac{\partial \tilde{M}_{ij}}{\partial \boldsymbol{p}_k}(\tilde{\boldsymbol{x}}),\ \frac{\partial \tilde{\mathscr{S}}_i}{\partial \boldsymbol{p}_k}(\tilde{\boldsymbol{x}}),\ \frac{\partial \chi_l}{\partial \boldsymbol{p}_k}(\tilde{\boldsymbol{x}},\boldsymbol{z}).$$

These functions have been precomputed and simplified using the Matlab symbolic toolbox. Using these functions, the routines F and E have been written to evaluate the feedforward torque and the parameter linear form:

$$\boldsymbol{\tau}_{FF} = F(\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}},\ddot{\tilde{\boldsymbol{x}}},\boldsymbol{z}) \tag{31}$$

$$\Phi = E(\hat{\boldsymbol{\tau}},\tilde{\boldsymbol{x}},\dot{\tilde{\boldsymbol{x}}},\ddot{\tilde{\boldsymbol{x}}},\boldsymbol{z}) \tag{32}$$

G. Kalman Filter (K)

Since the EOM are linear in the parameters, the estimation problem is convex. A Kalman filter is used for real time estimation of the parameters and implemented as in [11]. The adaptive algorithm is used to track short timescale variations, caused by changing end-effector mass, and long timescale variations, for example due to changing temperature. The parameters are normalised by values obtained from an initial least squares estimation procedure. Thereafter, the normalised parameter drift R_1 is taken to be a diagonal matrix with 10^{-5} at the entries corresponding the inertial properties of the end-effector and 10^{-7} at the other entries. The measurement noise covariance is set at $R_2 = 10^{-4} I \ \ (\mathrm{Nm})^2$, with I being the identity matrix. This covariance corresponds to the observed flat and equal torque measurement noise for each of the actuators.

H. Execution Rates

The time to calculate the feedforward (F, N_f and N_E) is 0.49 ms at most, therefore an execution rate of 2 kHz can be chosen. The majority of this time is used by the Newton-Raphson procedures N_E and N_F. Updating all 162 variables (E and K) takes 11 ms. An execution rate of 33.33 Hz is chosen, which is sufficiently fast to track the expected parameter changes. The computation of the parameter linear form E takes less that 1 ms and the majority of the computation time is required for the Kalman filter.

V. Experimental Results

The performance of the implemented AFFC scheme is evaluated on the T-flex manipulator [2]. This section presents the procedure, the results and a discussion of the experiments that show AFFC performance without changing end-effector mass. In the last section V-C the ability to adapt to changing end-effector mass is considered.

A. Experimental procedure

A 50 s trajectory is used. A new setpoint $\boldsymbol{r} \in \mathbb{R}^6$ is generated every second by a seeded random number generator. The setpoints are connected using a 0.6 s transition trajectory with piecewise linear acceleration. The trajectory is repeated three times, such that the non-repeatable behaviour of the manipulator can be determined. The tracking error and feedback torque will be determined for the following cases:

1) Feedback control only.
2) Feedback and feedforward control with a predetermined parameter set $\hat{\boldsymbol{p}}$. This parameter set was generated from a linear least squares fit to a previous dataset.
3) Feedback and adaptive feedforward control. The Kalman filter is initialised using the same $\hat{\boldsymbol{p}}$.
4) The non-repeatable part of the manipulator behavior, determined from the data obtained in case 2).

Furthermore, in order to evaluate the performance of the actuator model separately, the experiments are performed twice. Once with and once without the the upper part of the manipulator, consisting of the bodies Q_i and E.

B. Results and discussion

Table I shows the RMS tracking error and feedback torque of the experiments. The RMS tracking error at the joints has been translated to the end-effector tracking error using the nominal kinematic model.

From these results it can be seen that feedforward control yields a reduction of 96.4% of the tracking error when compared to using feedback control only and a reduction of 97.9% of the feedback torque. Note, perfect feedforward yields zero feedback torque. Introducing the adaptive estimation leads to an additional decrease in tracking error of 33.2% and an additional decrease of 26.8% in feedback torque. Fig. 5 shows the resulting feedforward and feedback torque. Further improvement is still possible, as 86.3% of the remaining feedback torque is still repeatable and could thus in principle be compensated if the feedforward model is properly extended.

The tracking error obtained with only the upper arms is similar to the tracking error for the total manipulator. This indicates that the actuator model is the limiting factor. Furthermore, it has been observed that the contribution to the RMS tracking error of frequencies higher than the feedback bandwidth is only marginal. This indicates that the rigid body approximation of the manipulator dynamics suffices for the utilised reference trajectory.

C. Adaptive performance

The tracking performance under changing end-effector inertial load is tested using the same random setpoint trajectory. However in this case the end-effector mass is varied with time, as shown in Fig. 4. Only the parameters associated with end-effector are estimated online. The adaptive algorithm maintains a similar performance when the end-effector mass is changed, whereas the performance of the non-adaptive algorithm becomes visibly worse, particularly for the highest change in mass.

TABLE I

THE RMS TRACKING ERRORS AND FEEDBACK TORQUE FOR THE EXPERIMENTS WITHOUT CHANGING END-EFFECTOR MASS.

	Total Manipulator								Actuators only	
RMS values	$e\,(\mu rad)$	$\tau_{FB}\,(Nm)$	$\Delta x\,(\mu m)$	$\Delta y\,(\mu m)$	$\Delta z\,(\mu m)$	$\Delta\phi\,(\mu rad)$	$\Delta\theta\,(\mu rad)$	$\Delta\psi\,(\mu rad)$	$e\,(\mu rad)$	$\tau_{FB}\,(Nm)$
Feedback only	1325	0.8767	197.3	166.2	137.8	1439.7	1816.4	316.5	565	0.9608
Non-adaptive	47.6	0.0185	8.5	10.1	8.0	43.0	53.7	22.7	34.6	0.0131
Adaptive	31.8	0.0117	5.2	5.4	4.7	24.7	26.0	13.4	27.5	0.0097
Repeatability	6.5	0.0016	0.9	1.0	0.9	4.7	5.0	2.7	6.4	0.0019

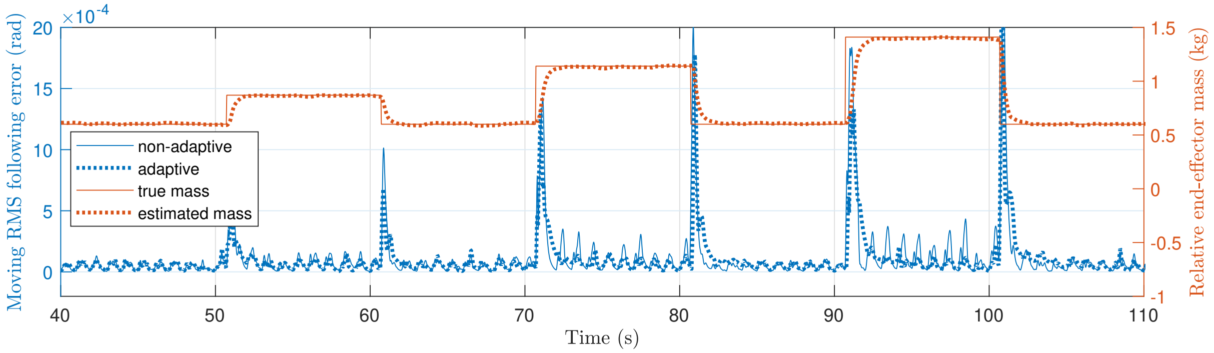

Fig. 4. The moving window RMS tracking error and estimated mass parameter. The utilised moving window is a Hann window with a width of 0.25 s. The test has been performed using a non-adaptive feedforward and an adaptive feedforward.

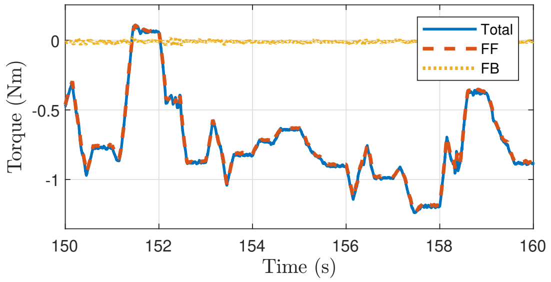

Fig. 5. Feedforward (FF), feedback (FB) and total torque of actuator 1 for the random setpoint trajectory when using adaptive feedforward control for the total manipulator.

VI. Conclusions

The compact kinematic formulation, the parametric approach and the parallel multi-rate processing resulted in a feedforward model for the T-flex PKM that can be computed within 0.5 ms. An estimation update is realised every 30 ms. For the utilised random setpoint trajectory the feedback current can be reduced to less than 1.4% and the tracking error to below 2.5% of the system without feedforward. When the actuators are tested separately, only a slightly better tracking error is obtained. This gives confidence in the accuracy of the derived kinematic model. Further improvement for the utilised reference trajectory should be achievable using a better actuator model.

The adaptive processing results in the manipulator being able to maintain its performance under changing end-effector mass. Further research is required to investigate the long term variations of the other parameters. Tests with faster and more exiting reference trajectories can also be performed, further utilising the fast execution rates of the implemented AFFC scheme.

References

[1] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, dec 2017.

[2] M. Naves, W. Hakvoort, M. Nijenhuis, and D. Brouwer, “T-flex: A large range of motion fully flexure-based 6-dof hexapod,” in *Proceedings of the 20th International Conference of the European Society for Precision Engineering and Nanotechnology, EUSPEN 2020*, pp. 205–208, 2020.

[3] J. J. Craig, P. Hsu, and S. Sastry, “Adaptive control of mechanical manipulators.,” *International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.

[4] H. Saied, A. Chemori, M. El Rafei, C. Francis, and F. Pierrot, “From non-model-based to model-based control of pkms: A comparative study,” *Mechanisms and Machine Science*, vol. 58, pp. 153–169, 2019.

[5] J. Escorcia-Hernández, H. Aguilar-Sierra, O. Aguilar-Mejia, A. Chemori, and J. Arroyo-Núñez, “A new adaptive rise feedforward approach based on associative memory neural networks for the control of pkms,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 100, no. 3-4, pp. 827–847, 2020.

[6] M. Bennehar, A. Chemori, F. Pierrot, and V. Creuze, “Extended model-based feedforward compensation in 1 adaptive control for mechanical manipulators: Design and experiments,” *Frontiers in Robotics and AI*, vol. 2, p. 32, 2015.

[7] G. Natal, A. Chemori, and F. Pierrot, “Dual-space control of extremely fast parallel manipulators: Payload changes and the 100g experiment,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1520–1535, 2015.

[8] T. Hardeman, R. Aarts, and J. Jonker, “A finite element formulation for dynamic parameter identification of robot manipulators,” *Multibody System Dynamics*, vol. 16, no. 1, pp. 21–35, 2006.

[9] A. Schwab and J. Meijaard, “The belt, gear, bearing and hinge as special finite elements for kinematic and dynamic analysis of mechanisms and machines,” in *Proceedings of the Tenth World Congress on the Theory of Machines and Mechanisms, IFToMM*, pp. 1375–1386, Oulu University Press Oulu, Finland, 1999.

[10] L. Huang, W. Wang, and Y. Huang, “Design of dynamic model with adaptive nonlinear friction compensation for vacuum parallel manipulator,” in *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 33–38, 2016.

[11] A. Uncini, *Fundamentals of Adaptive Signal Processing*. Springer Publishing Company, Incorporated, 1st ed., 2016.