2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

# Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots

Stefan B. Liu and Matthias Althoff
Department of Informatics, Technical University of Munich, Germany, Email: [stefan.liu | althoff]@tum.de

***Abstract*— Model-based design of robotic systems has many advantages, among them faster development cycles and reduced costs due to early detections of design flaws. Approximate models are sufficient for many classical robotic applications; however, they no longer suffice for safety-critical applications. For instance, a dangerous situation which has not been detected by model-based testing might occur in a human-robot coexistence scenario since models do not exactly replicate behaviors of real systems—this problem arises no matter how accurate a model is, since even disturbances and sensor noise can cause a mismatch. We address this issue by adding non-determinism to robotic models and by computing the whole set of possible behaviors using reachability analysis. By using reachset conformance, we automatically adjust the required non-determinism so that all recorded behaviors are captured. For the first time this approach is demonstrated for a real robot.**

## I. Introduction

Formal methods require models of real physical systems. However, we only have formal correctness if models and real systems *conform* to each other. In [1] it is shown that for the formal verification of safety properties, *reachset conformance* is sufficient. As shown in Fig. 1, this means that the real behavior (red lines) must always lie within the reachable set prediction (gray area) of the model. Therefore, reachset conformance checking is a prerequisite for safety approaches such as verified controllers [2] or safe human-robot coexistence [3]. Further possible applications of reachset-conformant models are, e.g., the error bounding of feedback control and the formal analysis of open-loop scenarios, such as mechanical braking or sensor faults, where the robot's possible future behavior could quickly diverge. Here, reachable sets give us upper and lower-bound predictions of the robot position and velocity states, which helps us to formally avoid collisions with surrounding objects.

Bounding uncertainties have previously been addressed in set-membership approaches [4], [5], where one determines feasible parameter sets of dynamical systems such that the current measurement of a physical system is always contained within the output sets of its model. Set-membership approaches are useful for robot modeling [6], fault diagnosis [7], and state estimation [8]. Reachset conformance extends set-membership by the idea that not only current, but also the future behavior is considered in the uncertainties. The tool proposed in [9] monitors reachset conformance for a future time sequence at runtime for systems modeled as hybrid programs. In this work, we model our systems using differential equations. Previous reachset conformance checks can be found for human arms [10], [11] or pedestrians [12].

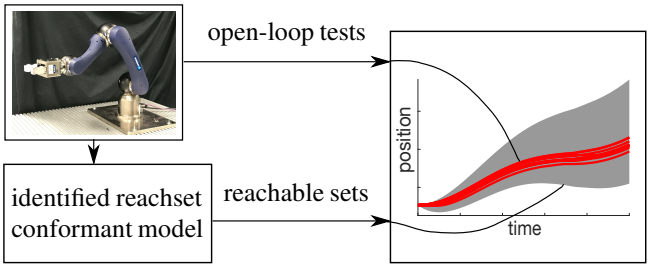


Fig. 1. We identify a reachset conformant model of the Schunk LWA-4P robot, such that the reachable sets enclose all recorded future behaviors of the robot.

Formal analysis tools for dynamical systems (e.g., SpaceEx [13], Flow* [14], HyLAA [15], XSpeed [16], or CORA [17]) require simple, yet conformant models, which are restricted to e.g., linear or polynomial terms. In contrast, the forward dynamics of robot arms are highly nonlinear and also hard to obtain symbolically, especially when the robot has many degrees of freedom (DOFs). In [18] the authors use automatic differentiation to generate fast forward dynamics abstractions up to the second order without explicitly generating the symbolic version of the forward dynamics. Higher-order approximations are often realized using Taylor polynomial arithmetics [19], [20] and are not only beneficial for formal techniques, but also for control design and optimal control in particular [18], [21].

In this paper we present the first work on reachset conformance of robot arms. We are aiming to find abstract models with a simple structure and consider unmodeled effects by adding non-determinism to achieve reachset conformance. Our approach creates reachset-conformant models in four steps:

1) We identify the nominal robot dynamics through experiments on the real counterpart.
2) We generate a global forward dynamics abstraction (linear or polynomial) using Taylor polynomial arithmetics and exploiting structural properties.
3) We perform open-loop testing using a fixed input trajectory.
4) We identify additive uncertainties using intervals of minimum size to ensure that all recorded behaviors lie within the reachable set of the abstract model.

Our approach is demonstrated experimentally on a 6-DOF Schunk LWA-4P robot arm shown in Fig. 1. We begin this paper in Sec. II by formalizing the problem at hand. In Sec. III we introduce the mathematical tools we use. Sec. IV

describes our main contribution, which is the identification of the reachset-conformant robot model. Experimental results on our Schunk LWA-4P robot arm are presented in Sec. V.

## II. Problem Statement

We consider a robot manipulator with rotary joints, whose $n$ joint positions and $n$ velocities $x = (q, \dot{q})^T \in \mathbb{R}^{2n}$ depend on joint torques $u \in \mathbb{R}^n$. To describe all possible behaviors of a robot manipulator, we use a first-order differential inclusion in state space form. Model uncertainties are captured by sets of uncertain initial states $\mathcal{X}_0 \subset \mathbb{R}^{2n}$ and sets of uncertain inputs $\mathcal{U} \subset \mathbb{R}^n$ imposed on the initial state $x_0 = x_m(0)$ and nominal input $u_m(t)$:

$$\dot{x} \in \Big\{ f(x,u) \Big| u(t) \in u_m(t) \oplus \mathcal{U} \Big\}, x(0) \in x_0 \oplus \mathcal{X}_0, \quad (1)$$

where the Minkowski sum is defined as $\mathcal{A} \oplus \mathcal{B} = \{a+b | a \in \mathcal{A}, b \in \mathcal{B}\}$. Next, we define reachable sets:

**Definition 1** (Reachable Set)**.** Given an initial set $\mathcal{X}_0$ and a time-dependent input trajectory $u_m(\tau)$, and the uncertain input set $\mathcal{U}$, the reachable set at time $t$ of system (1) is

$$\mathcal{R}(t, x_0, u_m(\tau)) = \Big\{ \int_0^t f(x(\tau), u(\tau)) d\tau + x(0) \Big| \\ x(0) \in x_0 \oplus \mathcal{X}_0, \forall \tau \in [0,t] : u(\tau) \in u_m(\tau) \oplus \mathcal{U} \Big\}.$$

For conformance checking, reachable sets are compared against test suites obtained from the real robot (see Fig. 1):

**Definition 2** (Test suite)**.** Given are measured trajectories $x_{m,1}(\cdot), x_{m,2}(\cdot), ...$ of a physical system starting at the same initial state $x_0$ and receiving the same input trajectory $u_m(\cdot)$ via open-loop control. A test suite is the set

$$X_m(t, x_0, u_m(\cdot)) = \{x_{m,1}(t), x_{m,2}(t), ...\}.$$

For establishing reachset conformance, the sets $\mathcal{X}_0$, $\mathcal{U}$ are chosen such that reachable sets always overapproximate all test suites, regardless of input or initial state. To formalize our goal, we introduce the volume operator $Vol()$ and the time horizon $t_e$. The goal of this paper is to derive a robot model in the form of system (1), where $f(x,u)$ is linear or polynomial and where the uncertainty sets $\mathcal{X}_0$ and $\mathcal{U}$ are chosen such that a reachset-conformant model is obtained whose reachable set has a minimized volume:

$$\begin{aligned} &\min_{\mathcal{X}_0, \mathcal{U}} \int_0^{t_e} \mathrm{Vol}\Big(\mathcal{R}(t, x_0, u_m(\cdot))\Big) dt \\ &\text{subject to} \quad \forall x_0, u_m(\cdot), t \in [0, t_e] : \\ &\mathcal{R}(t, x_0, u_m(\cdot)) \supseteq X_m(t, x_0, u_m(\cdot)). \end{aligned} \quad (2)$$

## III. Preliminaries

To obtain reachset-conformant models, we use Taylor polynomials and interval arithmetics, which are introduced subsequently.

### A. Taylor polynomial arithmetics

We use Taylor polynomials to locally approximate a continuous function $f(z)$ with variables $z =\in \mathbb{R}^k$ at the expansion point $a \in \mathbb{R}^k$.

**Definition 3** (Taylor polynomial (see Sec. 3 in [22]))**.** Let us first introduce the multi-index set

$$\mathcal{L}^p = \big\{ (l_1, l_2, ..., l_k) \big| l_i \in \mathbb{N}, \sum_{i=1}^k l_i \leq p \big\}.$$

We define $T_f^p(z-a)$ as a p-th order Taylor polynomial of $f(z)$ around $a$:

$$T_f^p(z-a) = \sum_{l \in \mathcal{L}^p} \frac{\prod_{i=1}^k (z_i - a_i)^{l_i}}{l_1! ... l_k!} \left( \frac{\partial^{l_1 + ... + l_k} f(z)}{\partial z_1^{l_1} ... \partial z_k^{l_k}} \right) \Bigg|_{z=a}.$$

One way to create $T_f^p(z)$ (short notation) is to obtain $f(z)$ symbolically and subsequently compute its derivatives. A second way is to perform numerical differentiation, which often yields high inaccuracies [18].

A third way to build Taylor polynomials is via composition of simpler Taylor polynomials. In fact, coefficients of Taylor polynomials form a commutative algebra [19], [20] with well-defined arithmetic operators such as '+','·' and '/'. Via operator overloading we can use the same algorithms that are used for the numerical evaluation of $f(z)$ (i.e., recursive Newton-Euler and Featherstone's algorithm in robotics) to compose $T_f^p(z)$ up to an arbitrary degree $p$. For our application, this approach is faster and more accurate than symbolical or numerical derivation. For details on our implementation of Taylor polynomial arithmetics, please see [23].

### B. Interval arithmetics

We use intervals to describe model uncertainties. An interval is defined by an upper and a lower limit $[a] := [\underline{a}, \overline{a}], \underline{a} \in \mathbb{R}, \overline{a} \in \mathbb{R}, \underline{a} \leq \overline{a}$. Set-based operations $* \in \{+, -, \cdot\}$ are defined as

$$[a] \circledast [b] := \{a * b | a \in [a], b \in [b]\}.$$

The functions $\inf([\underline{a}, \overline{a}]) := \underline{a}$ and $\sup([\underline{a}, \overline{a}]) := \overline{a}$ return the infimum and supremum, respectively.

A $k$-dimensional interval is called *hyperrectangle* and is defined by the Cartesian product of intervals in each dimension $[\underline{z}_1, \overline{z}_1] \times \cdots \times [\underline{z}_k, \overline{z}_k]$. For an arbitrary set $\mathcal{Z} \in \mathbb{R}^k$ the function $\inf(\mathcal{Z}) := \underline{z} \in \mathbb{R}^k$ and $\sup(\mathcal{Z}) := \overline{z} \in \mathbb{R}^k$ return the infimum and supremum of the smallest hyperrectangle overapproximation of $\mathcal{Z}$.

## IV. Forward dynamics modeling

We aim to create a reachset-conformant robot model in the form of (1), consisting of the nominal model $f(x,u)$ and uncertainty sets $\mathcal{X}_0, \mathcal{U}$. We first introduce our robot and friction model. Afterwards we present our main contribution, which is the forward dynamics abstraction and the identificiation of uncertain sets based on conformance testing.

### A. Robot model

The standard inverse dynamics model of a robot arm is

$$\tau_l = \Big(M(q) + diag(k_r^2 I_m)\Big)\ddot{q} + c(q,\dot{q}) + g(q), \quad (3)$$

where $\tau_l$ is the link-side torque; $q, \dot{q}, \ddot{q}$ are the joint positions, velocities, and accelerations, respectively; $M$ is the mass matrix; $c$ are the Coriolis forces; $g$ is the gravity vector; $k_r$ is the gear ratio; and $I_m$ is the motor inertia. These terms can be obtained by using the recursive Newton-Euler algorithm (see Ch. 7 in [24]). We present two possible ways to obtain the nominal part of (1): The first one is to solve (3) for $\ddot{q}$, which results in:

$$\ddot{q} = M_m(q)^{-1}(\tau_l - c(q,\dot{q}) - g(q)), \quad (4)$$

where $M_m(q) = M(q) + diag(k_r^2 I_m)$. The second way is to compute $\ddot{q}$ directly using Featherstone's algorithm for rigid-body dynamics [25], which bears a result equal to (4). Featherstone's algorithm is generally more accurate and is faster for many DOFs [26]. Using our computer setup (see Sec. V), however, neither algorithm is able to terminate when trying to obtain (4) symbolically for DOFs greater than four. Therefore we use Taylor polynomial arithmetics.

For the joint friction we choose to model the load-dependency and nonlinearity of joint friction. The resulting link-side torque is

$$\tau_{l,i} := \tau_{m,i} - \tau_{c,i} - \underbrace{(v_1\dot{q}_i + v_2\dot{q}_i^2 + v_3\dot{q}_i^3)}_{\tau_{\nu,i}}, \quad (5)$$

where $i$ denotes the joint number, $\tau_m$ is the motor torque, $\tau_c$ is the Coulomb friction, and $\tau_\nu$ is the viscous friction modeled as a cubic function with constants $v_1, v_2, v_3$.

For the Coulomb friction we use the model in [27] which considers different constants $a$ for each motor quadrant, such that $\tau_c(\tau_l, \dot{q})$ is for each joint (subscripts omitted):

$$\tau_c = a_1 + a_2\tau_l, \quad \text{if } sgn(\tau_l) \neq sgn(\dot{q}) \wedge \dot{q} < 0, \quad (6)$$
$$\tau_c = a_3 + a_4\tau_l, \quad \text{if } sgn(\tau_l) = sgn(\dot{q}) \wedge \dot{q} < 0, \quad (7)$$
$$\tau_c = a_5 + a_6\tau_l, \quad \text{if } sgn(\tau_l) \neq sgn(\dot{q}) \wedge \dot{q} > 0, \quad (8)$$
$$\tau_c = a_7 + a_8\tau_l, \quad \text{if } sgn(\tau_l) = sgn(\dot{q}) \wedge \dot{q} > 0. \quad (9)$$

Employing a load-dependent friction model has one caveat: when inserting (5) into (4), acceleration $\ddot{q}$ appears on both sides such that forward dynamics becomes implicit and would need to be solved iteratively [28]. We avoid acceleration to appear on the right side by setting $\tau_l = g(q)$ in (6)–(9), because gravity usually dominates $\tau_l$ at low speeds. For the identification of our robot and friction model we refer to the Appendix.

### B. Abstracting the forward dynamics

We abstract the following forward dynamics:

$$f(\hat{x}, u) = \begin{pmatrix} \hat{x}_2 \\ M_m(\hat{x}_1)^{-1}(u - c(\hat{x}_1, \hat{x}_2) - \tau_\nu(\hat{x}_2)) \end{pmatrix}, \quad (10)$$

where $\hat{x}_1 = q, \hat{x}_2 = \dot{q}$ and $u_i$ is the input of the $i^{th}$ joint

$$u_i := \tau_{m,i} - \tau_{c,i}(g_i(x_1), x_{2,i}) - g_i(x_1), \quad (11)$$

The input $u$ represents the motor torque, but with added gravity and feed-forward Coulomb friction compensation since this drastically simplifies the obtained model and only requires small uncertainty sets. In addition, we avoid mixed discrete/continuous dynamics by considering the discontinuities of $\tau_c$ inside $u$ instead of $f(\hat{x}, u)$. Additionally, we exploit three structural properties of robot dynamics:

**Property A** (Trigonometric $\hat{x}_1$): The generalized coordinates $\hat{x}_1$ of revolute joints only appear as trigonometric functions $\sin(\hat{x}_1)$ and $\cos(\hat{x}_1)$ in (3) as shown in [29]. By introducing $x_3 = q_s = \sin(\hat{x}_1)$ and $x_4 = q_c = \cos(\hat{x}_1)$ as new variables, (3) becomes a polynomial in $q_s, q_c, \dot{q}, \ddot{q}, \tau_l$. This reduces the number of operations and therefore reduces the model error when applying Taylor polynomial arithmetics. Using this property increases size of the state-space: $x = (q, \dot{q}, q_s, q_c)^T \in \mathbb{R}^{4n}$.

**Property B** (Near diagonal mass matrix): For high gear ratios the mass matrix $M_m$ is dominated by the constant term $diag(k_r^2 I_m)$. This also propagates to the inverse of $M_m$.

**Property C** (Omitting Coriolis terms): The Coriolis and centripetal term $c(x_1, x_2)$ can be written as

$$c(x_1, x_2) := \begin{pmatrix} x_2^T C_1(x_1) x_2 \\ ... \\ x_2^T C_N(x_1) x_2 \end{pmatrix}, \text{ see [30]}, \quad (12)$$

where $C_i \in \mathbb{R}^{N\times N}$ are matrices that depend only on $q$ and its coefficients are $c_{ijk} := \frac{\partial M_{ij}}{\partial x_{1,k}} - \frac{1}{2}\frac{\partial M_{jk}}{\partial x_{1,i}}$ (as shown in [24], Ch. 7) . In (12) it is shown that velocities $x_2$ only appear as squared terms in the forward dynamics, such that small velocities can be neglected and high velocities may let $c(x_1, x_2)$ dominate the robot dynamics. We propose omitting $c(x_1, x_2)$ for slow moving robots and using Taylor polynomials of $C_i(x_1)$ for high velocities.

In the following we list three useful models which apply the above properties to a varying degree and are evaluated subsequently. For a global approximation, we use the expansion point $(q_a, \dot{q}_a, \sin(q_a), u_a)^T = \vec{0}$ and $\cos(q_a) = \vec{1}$. Model 1 is the simplest model, considering properties B and C and only depends on input $u$ and velocity $x_2 = \dot{q}$:

$$\dot{x} = f_1(x, u) := \begin{pmatrix} x_2 \\ M_m^{-1}(q_a)(u - T^p_{\tau_\nu}(x_2)) \end{pmatrix}, \quad (13)$$

where the subscript of $T$ denotes the function that is Taylor-approximated. Model 2 uses assumptions A and B, and therefore considers Coriolis effects:

$$\dot{x} = f_2(x, u) := \begin{pmatrix} x_2 \\ M_m^{-1}(q_a)(u - c_a(x_2, x_3, x_4) - T^p_{\tau_\nu}(x_2) \\ x_2 x_4 \\ -x_2 x_3 \end{pmatrix} \quad (14)$$

$$c_a(x_2, x_3, x_4) := \begin{pmatrix} x_2'^T T^{p-2}_{C_1}(x_3, x_4) x_2 \\ ... \\ x_2'^T T^{p-2}_{C_N}(x_3, x_4) x_2 \end{pmatrix}, \quad (15)$$

where (15) is only evaluated for $p \geq 2$ (else $c_a = 0$) and we replace $x_1$ by the trigonometrical variables $x_3$ and $x_4$ when evaluating the Coriolis matrix. From property A we

know that $c(x_2, x_3, x_4)$ is a polynomial, of which we denote its order as $p_{max}$. We conclude that for $p \geq p_{max} - 2$ : $c_a(x_2, x_3, x_4) = c(x_2, x_3, x_4)$. Model 3 only considers property A:

$$\dot{x} = f_3(x, u) := \begin{pmatrix} x_2 \\ T_f^p(x_2, x_3, x_4, u) \\ x_2 x_4 \\ -x_2 x_3 \end{pmatrix}, \tag{16}$$

where $f$ is the second row of (10) and, e.g., can be computed using a modified Featherstone's algorithm that evaluates $q_s, q_c$ instead of $q$. Note that for $p = 1$ the model in (13) and the first two rows of (14) and (16), respectively, are identical and linear.

### C. Identifying the uncertainty sets

After obtaining the nominal part of (1), we identify the sets $\mathcal{X}_0$ and $\mathcal{U}$ by solving the optimization problem in (2). We first consider the case of linear systems ($p = 1$), which can also be written in the standardized form $\dot{x} = Ax + Bu_m$.

Given an initial state $x_0$ and an input trajectory $u_m(\cdot)$, the solution of a linear system is known to be

$$x(t, x_0, u_m(\cdot)) = e^{At} x_0 + \int_0^t e^{A(t-\tau)} v(\tau) d\tau,$$
$$v(\tau) = B u_m(\tau).$$

If the linear system has the uncertainty sets $\mathcal{X}_0$ and $\mathcal{U}$, the reachable set is

$$\mathcal{R}(t, x_0, u_m(\cdot)) =$$
$$e^{At}(x_0 \oplus \mathcal{X}_0) \oplus \int_0^t e^{A(t-\tau)}(v(\tau) \oplus \mathcal{V}) d\tau, \quad \mathcal{V} = B\mathcal{U}.$$

We consider the conformance constraint in (2) and subtract $x(t, *) = x(t, x_0, u_m(\cdot))$ from both sides to obtain $\forall t \in [0, t_e], \forall x_0, \forall u_m(\cdot)$:

$$X_m(t, *) - x(t, *) \subseteq \mathcal{R}(t, *) - x(t, *)$$

where on the left-hand side $x$ is substracted from every element of $X_m$, and thus

$$X_m(t, *) - x(t, *) \subseteq e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-\tau)} \mathcal{V} d\tau, \tag{17}$$

and observe that the right side is now independent of $x_0, u_m$.

**Proposition 1.** *By moving a constant set $\mathcal{V}$ out of the convolution integral, the result becomes an underapproximation*

$$\left\{ \int_0^t e^{A(t-\tau)} d\tau v \middle| v \in \mathcal{V} \right\} \subseteq$$
$$\left\{ \int_0^t e^{A(t-\tau)} v(\tau) d\tau \middle| \forall \tau : v(\tau) \in \mathcal{V} \right\}.$$

*The proof is trivial, because the notation already shows that on the right-hand side more solutions are present.* □

After introducting

$$X_{all}(t) := \bigcup_{i=1}^{I} \Big( X_{m,i}(t, x_{0,i}, u_{m,i}(\cdot)) - x(t, u_{m,i}(\cdot)) \Big),$$

where $I$ is the number of test suites, we infer from proposition 1 and (17)

$$X_{all} \subseteq e^{At}\mathcal{X}_0 \oplus \int_0^t e^{A(t-\tau)} d\tau \mathcal{V}, \tag{18}$$

which is a stricter constraint on $\mathcal{V}$ and thus it subsumes (17). For easier reading we introduce $E_1, E_2$ to replace the matrix-valued terms in (18):

$$X_{all}(t) \subseteq E_1(t)\mathcal{X}_0 \oplus E_2(t)\mathcal{V} = \begin{pmatrix} E_1(t) & E_2(t) \end{pmatrix} \begin{pmatrix} \mathcal{X}_0 \\ \mathcal{V} \end{pmatrix}.$$

We overapproximate both sides by hyperrectangles (multidimensional intervals). We then know that the following must hold true for $t \in [0, t_e]$:

$$\sup(X_{all}(t)) \leq \sup \left( \begin{pmatrix} E_1(t) & E_2(t) \end{pmatrix} \begin{pmatrix} \mathcal{X}_0 \\ \mathcal{V} \end{pmatrix} \right),$$
$$\inf(X_{all}(t)) \geq \inf \left( \begin{pmatrix} E_1(t) & E_2(t) \end{pmatrix} \begin{pmatrix} \mathcal{X}_0 \\ \mathcal{V} \end{pmatrix} \right).$$

Without loss of generality we assume that the origin is contained in $\mathcal{X}_0$ and $\mathcal{V}$. Hence $\inf(\mathcal{X}_0, \mathcal{V})^T$ is a $4n \times 1$ vector with only negative elements, and $\sup(\mathcal{X}_0, \mathcal{V})^T$ is a $4n \times 1$ vector with only positive elements. Usually, $t$ is sampled. We stack the vectors and matrices for all $m$ samples in time $0 \leq t_i \leq t_e$:

$$\sup(X_M) \leq |E_M| \sup \begin{pmatrix} \mathcal{X}_0 \\ \mathcal{V} \end{pmatrix} \tag{19}$$

$$\inf(X_M) \geq |E_M| \inf \begin{pmatrix} \mathcal{X}_0 \\ \mathcal{V} \end{pmatrix} \tag{20}$$

$$X_M = \begin{pmatrix} \cdots \\ X_{all}(t_k) \\ \cdots \end{pmatrix} \subset \mathbb{R}^{n \cdot m}, k = 0, ..., m$$

$$E_M = \begin{pmatrix} \cdots & \cdots \\ E_1(t_k) & E_2(t_k) \\ \cdots & \cdots \end{pmatrix} \in \mathbb{R}^{n \cdot m \times 2 \cdot n}, k = 0, ..., m.$$

We overapproximate all reachable sets $\mathcal{R}(t, *)$ with hyperrectangles $\mathcal{H}_{\mathcal{R}(t,*)}$. Because $x(t, *)$ is a vector, $\text{Vol}(\mathcal{H}_{\mathcal{R}(t,*)}) = \text{Vol}(\mathcal{H}_{\mathcal{R}(t,*)} - x(t, *)) = \text{Vol}(E_1(t)\mathcal{X}_0 \oplus E_2(t)\mathcal{V})$, where the last expression is evaluated via matrix interval multiplication (see Sec. 2.2 in [31]). We therefore simplify the optimization task in (2) to an optimization problem, that minimizes the sum of the edge lengths of $\mathcal{H}_{\mathcal{R}(t)}$:

$$\min_{y1, y2} \quad j^T \sum_{k=0}^{m} \begin{pmatrix} E_1(t_k) & E_2(t_k) \end{pmatrix} (y_1 - y_2), \tag{21}$$

where $j$ is a $2n \cdot m \times 1$ column vector of ones, $y_1 = (\sup(\mathcal{X}_0), \sup(\mathcal{V}))^T$ and $y_2 = (\inf(\mathcal{X}_0), \inf(\mathcal{V}))^T$. The advantage of (21) is that together with (19) and (20) a linear program is formed which can be efficiently solved. $\mathcal{U}^*$ is obtained by using the pseudo-inverse $B^{\#}$:

$$\mathcal{U} = B^{\#}\mathcal{V}^*,$$

where $\mathcal{U}^*$ is a hyperrectangle, when evaluated using matrix interval multiplication.

TABLE I

SIMULATION ERROR OF MODEL ABSTRACTIONS

| polyn. ord. $p$ | Model 1 slow | Model 1 fast | Model 2 slow | Model 2 fast | Model 3 slow | Model 3 fast |
|---|---|---|---|---|---|---|
| 1 | 0.0040 | 0.0408 | 0.0040 | 0.0408 | 0.0040 | 0.0408 |
| 3 | 0.0034 | 0.0243 | 0.0042 | 0.0294 | $\infty$ | $\infty$ |
| 5 | −− | −− | 0.0026 | 0.0161 | 0.0031 | $\infty$ |
| 7 | −− | −− | 0.0025 | 0.0162 | −− | −− |

For polynomial and nonlinear systems, we first linearize the dynamics and then use binary search for each dimension to find the sets $\mathcal{X}_0, \mathcal{U}$.

## V. EXPERIMENTAL RESULTS

In this section we present the experimental results of our approach. We carry out the experiments on a 6-DOF Schunk LWA 4P robot (Fig. 3), which is controlled by Simulink Real-Time OS on a Core i7 Speedgoat machine. The results for the identification of the nominal model can be found in the Appendix. Subsequently, we first evaluate the model abstractions in a simulation study, and then provide the results of conformance testing.

### A. Evaluation of the model abstractions

In this section we evaluate the effectiveness of the three different model abstractions proposed in Sec. IV-B. These are computed using our MATLAB reachability analysis tool CORA [17], which already contains an implementation of Taylor polynomial arithmetics. As can be inferred from their model structures models 1 and 2 have maximal polynomial degrees; for our robot these are 3 and 12, respectively.

We compare these models in simulations by generating a slow and a fast trajectory, where the top speeds of each axis are 0.4 and 1.2 rad/s (max. velocity from the robot's data sheet), respectively. In Tab. I we show the mean of velocity errors of each abstracted model versus the standard numerical simulation using Featherstone's algorithm.

We observe that for the slow trajectory, there is almost no difference between the abstracted models and the numerical simulation. For fast trajectories the errors are larger. For model 2, the error decreases below the error of model 1 for higher orders, because of the improved modeling. For model 3, however, the errors frequently diverge from the simulation.

We observe that the linear model already has a decent approximation performance, although one would not expect this for a single expansion point. Our simulations have shown that the model 1 abstractions only start to diverge from the original rigid-body dynamics at velocities much higher than the robot's capability. The improvements of model 2 are not that significant for our robot. In fact, as will be shown later in the experiments, the uncertainty of friction has a higher effect on the dynamics than the Coriolis terms, which have been considered in model 2. Model 3 is not suitable for global approximation.

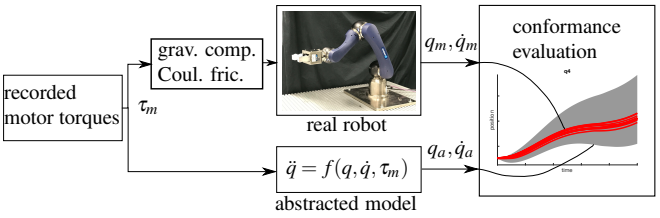


Fig. 2. Open-loop control with gravity and Coulomb friction compensation result in non-deterministic robot behavior (red). The same motor torques influence the abstracted model, which generates the reachable sets (gray).

### B. Conformance testing and optimal uncertainty sets

In this subsection we present the results from conformance testing and obtained optimal uncertainty sets. We focus on the linear model 1, which turned out to be sufficiently accurate in the previous subsection.

For conformance testing we have recorded 152 test suites, where each test suite consists of a fixed series of motor torques that are applied via open-loop control to the real robot 15 times from the same initial state, as shown in Fig. 2. The motor torques are pre-recorded from closed-loop point-to-point (PTP) motions. We generate 38 uniformly random PTP motions; from each motion, we choose four initial points, as shown in Fig. 3. The robot moves to these initial points via closed-loop control, and then immediately switches to open-loop by applying the pre-recorded motor torques, such that the resulting trajectories diverge. Each test suite is up to $t_e = 5$ seconds long.

Using the data from all test suites, we determine the optimal uncertainty bounds via Sec. IV-C such that the reachable sets enclose all measurements, as shown in Fig. 2. The results are shown in Tab. II for two cases: In the first case we aim for conformance of all states (position and velocity). In the second case we only aim for conformance of the robot position by excluding the velocity constraints in (19) and (20) from the linear program (21). Fig. 4 shows the reachability analysis of both cases for an exemplary test suite.

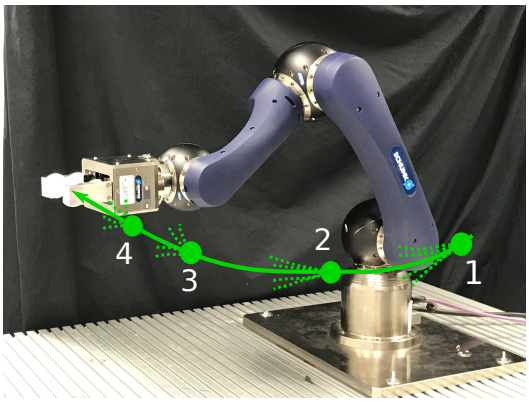


Fig. 3. In each random point-to-point motion the conformance testing starts from four different points indicated by dotted lines, where the controller is switched from closed-loop to open-loop

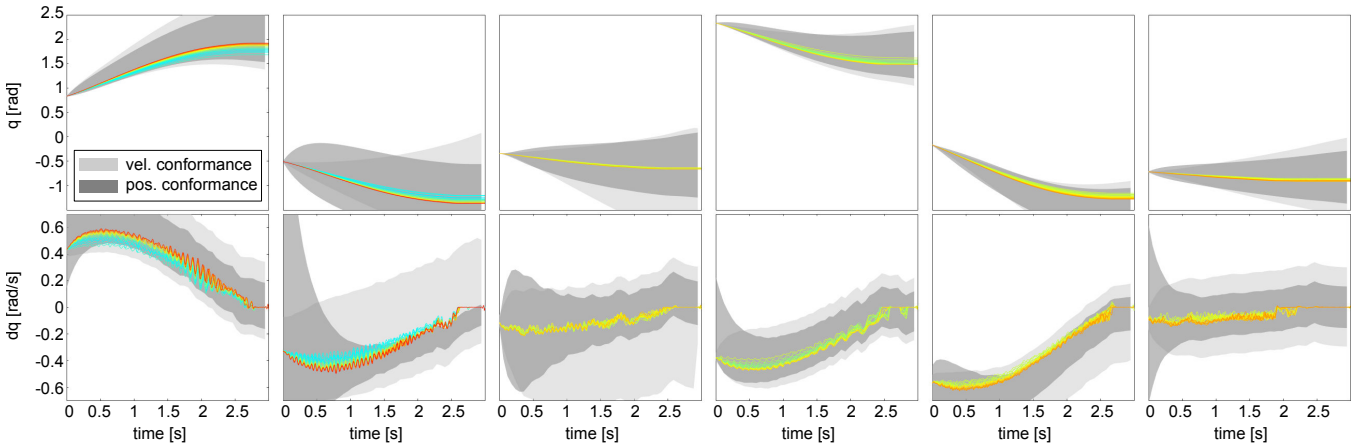


Fig. 4. Reachable set predictions of model 1 for axis 1-6 (left to right) of the Schunk LWA-4P robot. Light gray: both position and velocity are reachset conformant. Dark gray: only the position is reachset conformant. Colored lines are the measured test trajectories. The colors indicate the temperature relative to the temperature range of each axis, where red is hot and blue is cold.

TABLE II

IDENTIFIED OPTIMAL UNCERTAINTY SETS OF MODEL 1 FOR EACH JOINT

| | | **Velocity conformant** | |
|---|---|---|---|
| Jnt. | $\mathcal{X}_0 : q$ | $\mathcal{X}_0 : \dot{q}$ | $\mathcal{U}$ |
| 1 | $[-0.0030, 0.0030]$ | $[-0.0317, 1.2140]$ | $[-2.7201, 3.6017]$ |
| 2 | $[-0.0017, 0.0017]$ | $[-0.6586, 0.2550]$ | $[-2.3225, 7.1559]$ |
| 3 | $[-0.0025, 0.0025]$ | $[-0.0154, 0.0463]$ | $[-6.8833, 2.6287]$ |
| 4 | $[-0.0027, 0.0027]$ | $[-0.0184, 0.0331]$ | $[-1.7374, 2.0317]$ |
| 5 | $[-0.0075, 0.0075]$ | $[-0.1179, 0.0551]$ | $[-1.4919, 0.5486]$ |
| 6 | $[-0.0063, 0.0063]$ | $[-0.0765, 0.0765]$ | $[-1.0060, 1.0060]$ |
| | | **Position conformant** | |
| Jnt. | $\mathcal{X}_0 : q$ | $\mathcal{X}_0 : \dot{q}$ | $\mathcal{U}$ |
| 1 | $[-0.0030, 0.0030]$ | $[-0.2785, 1.1309]$ | $[-1.9257, 1.8615]$ |
| 2 | $[-0.0017, 0.0017]$ | $[-2.0695, 2.3636]$ | $[-0.8212, 1.0395]$ |
| 3 | $[-0.0025, 0.0025]$ | $[0, 0]$ | $[-1.8045, 1.5717]$ |
| 4 | $[-0.0027, 0.0027]$ | $[0.0000, 0.6020]$ | $[-1.2072, 1.2231]$ |
| 5 | $[-0.0075, 0.0075]$ | $[-0.3491, 0.1180]$ | $[-0.6622, 0.2146]$ |
| 6 | $[-0.0063, 0.0063]$ | $[-0.6369, 0.7051]$ | $[-0.5366, 0.5187]$ |

We observe that in both cases the test suite is enclosed, which means that the model shown in this evaluation is indeed reachset conformant. By color-coding the test trajectories according to the joint temperature measurement, we observe that temperature is one of the main reasons why the trajectories diverge. We have not included a temperature model in this work, but this would further improve the open-loop prediction.

## VI. CONCLUSIONS

We present an approach to create reachset-conformant models of robot manipulators. To this end, we abstract the identified forward dynamics to linear or polynomial systems and optimize the required uncertainty sets to achieve reachset conformance. Experimental results demonstrate the effectiveness of our approach on a real robot. Reachset-conformant models are useful for the formal analysis of uncertain behavior, such as to avoid collisions. We wish to apply our model to the formal analysis of mechanical braking (STOP 0 and STOP 1).

During the experiments it became apparent that friction has a large effect on the dynamics and that an accurate friction model is very important. Especially the highly uncertain stiction in the case of crossing zero velocity has not been addressed by this paper and is the subject of future work. We also plan to consider temperature dependency of friction in the future to reduce the uncertainty bounds.

## APPENDIX: DYNAMIC PARAMETER IDENTIFICATION

The identification of our robot is based on the works in [32] and [33]. The standard DH parameters can be found with the help of CAD files available from the Schunk website. Gear ratios are taken from the robot's data sheets ($k_r = [160, 160, 160, 160, 100, 100]$). We estimate the gravity model from 1000 static positions. Subsequently, we use the gravity torques as load torque to identify our friction model. As an example, we display the curve fitting results of the friction models for joint 2 in Fig. 5. Lastly, we determine our inertial parameters through linear regression. The results are shown in Tab. III.

## ACKNOWLEDGMENT

The authors gratefully acknowledge financial support by the Central Innovation Programme of the German Federal Government under grant ZF4086004LP7 and by the German Research Foundation (DFG) under grant AL 1185/5-1.

## REFERENCES

[1] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Reachset conformance testing of hybrid automata," in *Hybr. Systems: Comp. and Contr.*, 2016, pp. 277–286.

[2] B. Schürmann and M. Althoff, "Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space," in *Proc. 20th IFAC World Congress*, 2017, pp. 12 020–12 027.

[3] A. Pereira and M. Althoff, "Safety control of robots under computed torque control using reachable sets," in *Proc. ICRA*, 2015, pp. 331–338.

[4] M. Milanese and C. Novara, "Set Membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.

[5] M. Kieffer, E. Walter, and I. Simeonov, "Guaranteed nonlinear parameter estimation for continuous-time dynamical models," in *Proc. 14th IFAC World Congress*, vol. 39, no. 1, 2006, pp. 843–848.

[6] N. Ramdani and P. Poignet, "Robust dynamic experimental identification of robots with set membership uncertainty," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 2, pp. 253–256, 2005.

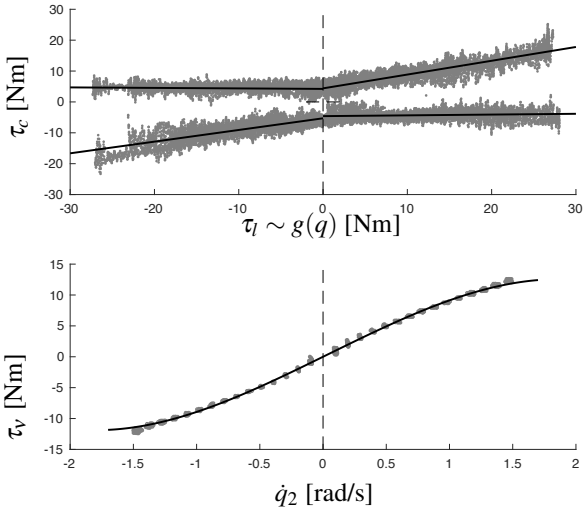


Fig. 5. Fitting results for load-dependent Coulomb friction (up) and viscous friction (down) of joint 2.

TABLE III

IDENTIFIED MODEL OF THE SCHUNK LWA-4P (SI BASE UNITS)

| **DH parameters** | | | | | | |
|---|---|---|---|---|---|---|
| Par. | Link 1 | Link 2 | Link 3 | Link 4 | Link 5 | Link 6 |
| a | 0 | 0.35 | 0 | 0 | 0 | 0 |
| $\alpha$ | $-\frac{\pi}{2}$ | $\pi$ | $-\frac{\pi}{2}$ | $\frac{\pi}{2}$ | $-\frac{\pi}{2}$ | 0 |
| $\theta$ | $q_1$ | $q_2 - \frac{\pi}{2}$ | $q_3 - \frac{\pi}{2}$ | $q_4$ | $q_5$ | $q_6$ |
| d | 0 | 0 | 0 | 0.3012 | 0 | 0.1548 |
| **Friction parameters** | | | | | | |
| Par. | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6 |
| $a_1$ | 6.08 | 4.46 | 6.27 | 6.08 | 1.71 | 2.33 |
| $a_2$ | 0 | 0.38 | 0.25 | 0 | 0.58 | 0 |
| $a_3$ | −6.08 | −5.46 | −6.50 | −6.08 | −1.95 | −2.33 |
| $a_4$ | 0 | 0.32 | 0.25 | 0 | 0.46 | 0 |
| $a_5$ | −6.08 | −4.64 | −5.95 | −6.08 | −2.10 | −2.33 |
| $a_6$ | 0 | 0.041 | −0.029 | 0 | 0.21 | 0 |
| $a_7$ | 6.08 | 4.08 | 6.38 | 6.08 | 1.67 | 2.33 |
| $a_8$ | 0 | −0.0019 | 0.0025 | 0 | −0.081 | 0 |
| $v_1$ | 11.52 | 10.08 | 10.68 | 8.41 | 3.61 | 3.53 |
| $v_2$ | 0.088 | 0.10 | 0.93 | −0.0098 | 0.024 | 0.11 |
| $v_3$ | −0.79 | −1.02 | −1.56 | 0.099 | −0.24 | −0.11 |

| **Gravity model** | |
|---|---|
| Parameter | Value |
| $m_2 + m_3 + m_4 + m_5 + m_6 + 2.8571c_{x2}m_2$ | 6.36 |
| $0.3012(m_4 + m_5 + m_6) + c_{y4}m_4 + c_{z3}m_3$ | 1.04 |
| $c_{z4}m_4 - c_{y5}m_5$ | −0.0661 |
| $0.1548m_6 + c_{z5} * m_5 + c_{z6}m_6$ | 0.189 |

| **Inertial parameters** | |
|---|---|
| Parameter | Value |
| $I_{1yy} + I_{2yy} - I_{2zz} + I_{3zz} + k_{r2}^2 I_{m1} - k_{r2}^2 I_{m2}$ | −0.0162 |
| $m_2 - 8.163(k_{r2}^2 I_{m2} - I_{2zz}) + m_3 + m_4 + m_5 + m_6$ | −7.45 |
| $0.301(m_4 + m_5 + m_6) + c_{y4}m_4 + c_{z3}m_3$ | 0.94 |
| $I_{3xx} - I_{3zz} + I_{4zz} + 0.0907(m_4 + m_5 + m_6) + 0.602c_{y4}m_4$ | 0.171 |
| $I_{3yy} + I_{4zz} + 0.0907(m_4 + m_5 + m_6) + 0.602c_{y4}m_4$ | 0.371 |
| $I_{5xx} + I_{6yy} - I_{5zz} + 0.024m_6 + 0.31c_{z6}m_6$ | 0.0555 |
| $I_{5yy} + I_{6yy} + 0.024m_6 + 0.31c_{z6}m_6$ | 0.126 |
| $2.857(I_{2zz} + k_{r2}^2 I_{m2}) + c_{x2}m_2$ | 5.02 |

| Parameter | Value | Par. | Val. |
|---|---|---|---|
| $0.155m_6 + c_{z5}m_5 + c_{z6}m_6$ | 0.174 | $k_{r3}^2 I_{m3}$ | 1.78 |
| $c_{z2}m_2 - c_{y3}m_3 - 2.857I_{2xz}$ | −0.183 | $I_{4yz}$ | 0.0726 |
| $I_{2xx} - I_{2yy} + I_{2zz} + k_{r2}^2 I_{m2}$ | 1.64 | $k_{r4}^2 I_{m4}$ | 1.63 |
| $c_{z4}m_4 - c_{y5}m_5$ | −0.0388 | $I_{5yz}$ | 0.0352 |
| $I_{4xx} - I_{4zz} + I_{5zz}$ | 0.135 | $k_{r5}^2 I_{m5}$ | 0.541 |
| $I_{4yy} + I_{5zz}$ | 0.0269 | $I_{6zz}$ | 0.00569 |
| $I_{6xx} - I_{6yy}$ | −0.0352 | $k_{r6}^2 I_{m6}$ | 0.637 |
| $I_{3yz}$ | 0.106 | | |


[7] V. Reppa and A. Tzes, "Fault detection based on orthotopic set membership identification for robot manipulators," in *Proc. 17th IFAC World Congress*, vol. 41, no. 2, 2008, pp. 7344–7349.

[8] L. Jaulin, "Range-only SLAM with occupancy maps: A set-membership approach," *IEEE Trans. Robotics*, vol. 27, no. 5, pp. 1004–1010, 2011.

[9] S. Mitsch and A. Platzer, "Modelplex: verified runtime validation of verified cyber-physical system models," *Form. Methods Syst. Des.*, vol. 49, no. 1, pp. 33–74, Oct. 2016.

[10] A. Pereira and M. Althoff, "Overapproximative human arm occupancy prediction for collision avoidance," *IEEE Trans. Autom. Sci. and Engin.*, vol. 15, no. 2, pp. 818–831, 2018.

[11] C. Stark, A. Pereira, and M. Althoff, "Reachset conformance testing of human arms with a biomechanical model," in *Proc. IRC*, 2018, pp. 209–216.

[12] S. B. Liu, H. Roehm, C. Heinzemann, I. Lütkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *Proc. IROS*, 2017, pp. 1351–1357.

[13] G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *Proc. CAV*, 2011, pp. 379–395.

[14] X. Chen, E. Ábrahám, and S. Sankaranarayanan, "Flow*: An analyzer for non-linear hybrid systems," in *Proc. CAV*, 2013, pp. 258–263.

[15] S. Bak and P. S. Duggirala, "HyLAA: A tool for computing simulation-equivalent reachability for linear systems," in *Hybr. Systems: Comp. and Contr.*, 2017, pp. 173–178.

[16] A. Gurung, A. Deka, E. Bartocci, S. Bogomolov, R. Grosu, and R. Ray, "Parallel reachability analysis for hybrid systems," in *Proc. Formal Methods and Models for System Design*, 2016, pp. 12–22.

[17] M. Althoff, "An introduction to CORA 2015." in *Proc. 2nd Int. Workshop on Appl. Verif. of Cont. and Hybr. Syst.*, 2015, pp. 120–151.

[18] M. Neunert, M. Giftthaler, M. Frigerio, C. Semini, and J. Buchli, "Fast derivatives of rigid body dynamics for control, optimization and estimation," in *Proc. SIMPAR*, 2016, pp. 91–97.

[19] M. Berz, "Differential algebraic description of beam dynamics to very high orders," *Part. Accel.*, vol. 24, pp. 109–124, 1988.

[20] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *Int. J. Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.

[21] G. Rigatos, P. Siano, and G. Raffo, "A nonlinear H-infinity control method for multi-DOF robotic manipulators," *Nonlin. Dynamics*, vol. 88, no. 1, pp. 329–348, 2017.

[22] R. Neidinger, "Directions for computing truncated multivariate Taylor series," *Math. of computation*, vol. 74, no. 249, pp. 321–340, 2005.

[23] M. Althoff, D. Grebenyuk, and N. Kochdumper, "Implementation of taylor models in CORA 2018," in *Proc. 5th Int. Workshop on Appl. Verif. of Cont. and Hybr. Syst.*, 2018.

[24] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics*. London: Springer-Verlag, 2009.

[25] R. Featherstone, "The calculation of robot dynamics using articulated-body inertias," *Int. J. Robotics Research*, vol. 2, no. 1, pp. 13–30, 1983.

[26] R. Featherstone and D. Orin, "Chapter 2: Dynamics," in *Springer handbook of robotics*. Springer-Verlag Berlin Heidelberg, 2008.

[27] P. Hamon, M. Gautier, P. Garrec, and A. Janot, "Dynamic modeling and identification of joint drive with load-dependent friction model," in *Proc. Advanced Intelligent Mechatronics*, 2010, pp. 902–907.

[28] P. E. Dupont, "The Effect of Friction on the Forward Dynamics Problem," *Int. J. Robotics Research*, vol. 12, no. 2, pp. 164–179, 1993.

[29] M. Townsend and S. Gupta, "Automated modeling and rapid solution of robot dynamics using the symbolic polynomial technique," *J. Mech., Transm., and Autom. in Des.*, vol. 111, no. 4, pp. 537–544, 1989.

[30] V. D. Tourassis and C. P. Neuman, "Properties and structure of dynamic robot models for control engineering applications," *Mechanism and Machine Theory*, vol. 20, no. 1, pp. 27–40, 1985.

[31] S. M. Rump, "Intlabinterval laboratory," in *Developments in reliable computing*. Springer, 1999, pp. 77–104.

[32] A. H. Memar and E. T. Esfahani, "Modeling and dynamic parameter identification of the Schunk powerball robotic arm," in *ASME Int. Des. Eng. Techn. Conf. and Comp. and Info. in Eng. Conf.*, vol. 5C, 2015.

[33] C. Gaz, F. Flacco, and A. De Luca, "Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach," in *Proc. ICRA*, 2014, pp. 1386–1392.