AURT: A Tool for Dynamics Calibration of Robot Manipulators*

Emil Madsen[1,†], Daniella Tola[1], Carlos Hansen[1], Cláudio Gomes[1], Peter Gorm Larsen[1]

***Abstract*— We introduce AURT, an open-source software for modeling and calibration of robot manipulator dynamics. AURT includes rigid-body dynamics and a selection of joint dynamics models. The separation of the robot dynamics into rigid-body dynamics and joint dynamics allows for fast re-calibration of the robot dynamics using different joint dynamics models. We evaluate the calibration performance using a Universal Robots UR5e robot manipulator. We also demonstrate how the joint dynamics models embedded in AURT outperforms the standard approach of the state of the art.**

I. Introduction

Robot manipulators are predicted to replace many manufacturing machines and it is expected that more manufacturing floors are connected to digital twins [1], [2]. As a consequence, there is a need for tools that enable the creation and calibration of robot manipulator dynamics models by non-specialists. These can be used at the robot control level and for digital twins. At the control level, they are paramount for safety, trajectory accuracy, and lead-through programming [3]. At the digital twin level, these models can be part of cloud computing solutions where anomaly detection, predictive maintenance, and self-adaptation tasks can be performed.

There has been extensive work on the theoretical techniques that facilitate the calibration of dynamics models for robot manipulators. Noteworthy is the pioneering work in [4] and [5], which discuss how to obtain a linear robot dynamics model and eliminate redundant parameters symbolically and numerically. Since then, multiple tools have been developed which facilitate creation and calibration of robot dynamics models. However, to the best of the knowledge of the authors, no tool exists that allows for experimentation using different joint dynamics models, which are essential for achieving dynamics models of high fidelity. Additionally, most tools do not include the entire workflow comprising dynamics modeling, calibration, and validation but rather focus on a single subject.

In this paper, we present the Aarhus University Robotics Toolbox (AURT) [6]: An open source tool licensed under the *INTO-CPS Association Public License*, which allows for dynamics modeling, calibration, and validation of robot dynamics models. The separation of the robot dynamics into rigid-body dynamics and joint dynamics allows the user to quickly try out different joint dynamics models without having to re-compile the rigid-body dynamics each time – a computationally demanding process.

The paper is structured as follows. In Section II we provide an overview of AURT and describe how users create and calibrate their own robot dynamics models. Section III describes how AURT achieves a linear and minimal dynamics description using a numerical technique based on a QR decomposition of the observation matrix. Section IV details the Weighted Least Squares (WLS) calibration procedure. In Section V, AURT is applied to achieve a full robot dynamics calibration of a Universal Robots UR5e collaborative robot. We also demonstrate the model prediction capabilities using the joint dynamics models embedded in AURT. Compared to the state of the art described in Section VI, the tool enables fast and easy experimentation with different joint dynamics models, which are common in calibration experiments. Finally, Section VII concludes on the work and presents our future developments.

II. Tool Overview

In the following, we present the main commands of AURT [6] following the workflow illustrated in Fig. 1. The robot dynamics model is compiled through two steps (by compiling separately the rigid-body dynamics) because this allows the user to efficiently switch between different joint dynamics models without having to re-compile the rigid-body dynamics each time, which is a computationally demanding process. AURT uses a command line interface for its simplicity, ease of composition with other tool chains, and low maintenance. In the following, we detail each command of Fig. 1.

A. Compile Rigid-Body Dynamics Model

The `compile-rbd` command reads the Modified Denavit-Hartenberg (MDH) [7] parameters in `mdh.csv` and outputs the rigid-body dynamics model to `rigid_body_dynamics`. The components of the gravity vector determine the orientation of the robot base. The generated model does not include the joint dynamics. An example description of `mdh.csv` is provided in Section V.

B. Compile Robot Dynamics Model

The `compile-rd` command reads the rigid-body dynamics model created in Section II-A, and generates the robot dynamics model, from the joint dynamics configuration. The parameter `friction-load-model` is either `none`, `square`, or `absolute` as described in Section III-B.1 and `friction-viscous-powers` is a set of positive integers defining the structure of the viscous friction model as described in Table I.

*We are grateful to the Poul Due Jensen Foundation, which has supported the establishment of a new Centre for Digital Twin Technology at Aarhus University.

[1] Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

† Corresponding author: `ema@ece.au.dk`

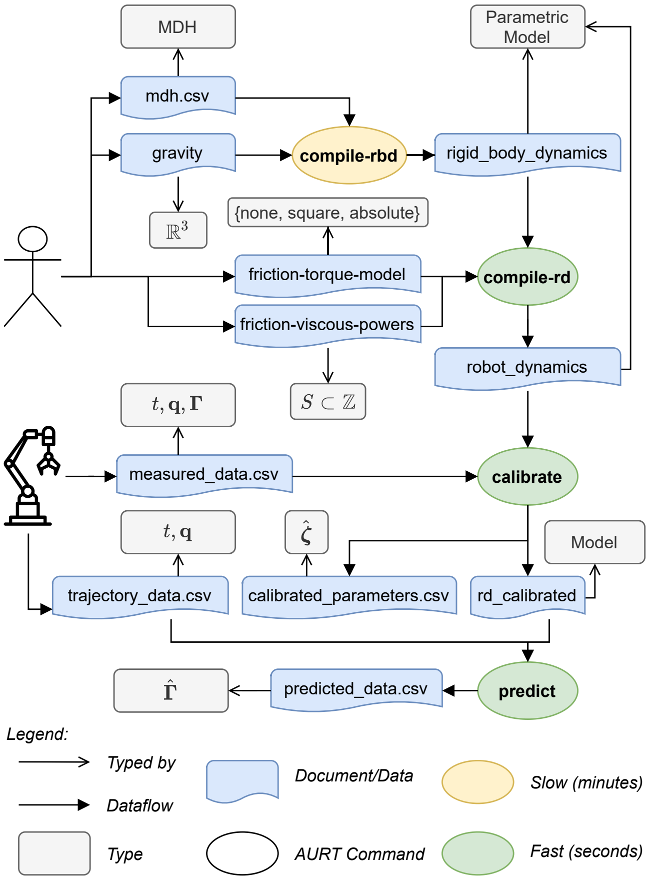


Fig. 1: Typical calibration workflow and main commands of the AURT tool. The mathematical symbols are defined in Section III.

C. Calibrate and Validate

The `calibrate` command reads the robot dynamics model produced in Section II-B and the measured data in `measured_data.csv`, and writes the values of the calibrated dynamics coefficients to `calibrated_parameters.csv` and the calibrated robot dynamics model to `rd_calibrated`. `measured_data.csv` should contain the time t, the measured angular positions $\mathbf{q}$, and the measurements $\mathbf{y}$ (e.g. actuator currents, joint torques, etc.).

The `calibrate-validate` command (not shown in Fig. 1), in addition to calibrating on a subset of the data, also performs a validation prediction on the remaining data.

D. Predict

The `predict` command reads the calibrated robot dynamics model `rd_calibrated` produced in Section II-C and `trajectory_data.csv`, and writes `predicted_data.csv` comprising the prediction $\hat{\mathbf{y}}$ – usually the joint torques or electric actuator currents.

III. Robot Description

The robot is assumed to be an open kinematic chain having $N+1$ rigid bodies (base and N links), interconnected by N revolute joints. The MDH parameters are used to describe the geometry. In this notation, frame $\{j\}$ is attached to link j and has its origin attached to joint axis j and the generalized coordinate of joint j is q_j. The inverse dynamics is the sum of rigid-body dynamics (RBD) related to the motions of the rigid links and joint dynamics (JD) related to the drive actuator of each joint. The robot dynamics are linear in the dynamic coefficients $\boldsymbol{\zeta}$, hence

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\mathrm{RBD}} + \boldsymbol{\tau}_{\mathrm{JD}} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\, \boldsymbol{\zeta} \tag{1}$$

where $\boldsymbol{\zeta}$ contains parameters related to the RBD and JD for each joint,

$$\boldsymbol{\zeta} = \begin{bmatrix} \boldsymbol{\zeta}_1^\mathsf{T} & \cdots & \boldsymbol{\zeta}_N^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_p}, \tag{2}$$

$$\boldsymbol{\zeta}_j = \begin{bmatrix} \boldsymbol{\zeta}_{\mathrm{RBD},j}^\mathsf{T} & \boldsymbol{\zeta}_{\mathrm{JD},j}^\mathsf{T} \end{bmatrix}^\mathsf{T} \in \mathbb{R}^{n_{p,j}}, \tag{3}$$

with $n_p \equiv \sum_j n_{p,j}$ and the regressor matrix $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{N \times n_p}$ contains nonlinear functions in the generalized coordinates $\mathbf{q}$ and time-derivatives $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$.

Let the vector of Base Inertial Parameters (BIP) $\boldsymbol{\zeta}_{\mathrm{RBD}} \triangleq \begin{bmatrix} \boldsymbol{\zeta}_{\mathrm{RBD},1}^\mathsf{T} & \cdots & \boldsymbol{\zeta}_{\mathrm{RBD},N}^\mathsf{T} \end{bmatrix}^\mathsf{T}$ and the vector of joint dynamics parameters $\boldsymbol{\zeta}_{\mathrm{JD}} \triangleq \begin{bmatrix} \boldsymbol{\zeta}_{\mathrm{JD},1}^\mathsf{T} & \cdots & \boldsymbol{\zeta}_{\mathrm{JD},N}^\mathsf{T} \end{bmatrix}^\mathsf{T}$, then the RBD and JD subsystems can be expressed linearly in $\boldsymbol{\zeta}_{\mathrm{RBD}}$ and $\boldsymbol{\zeta}_{\mathrm{JD}}$ as

$$\boldsymbol{\tau}_{\mathrm{RBD}} = \mathbf{Y}_{\mathrm{RBD}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\, \boldsymbol{\zeta}_{\mathrm{RBD}}, \tag{4}$$

$$\boldsymbol{\tau}_{\mathrm{JD}} = \mathbf{Y}_{\mathrm{JD}}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\, \boldsymbol{\zeta}_{\mathrm{JD}}. \tag{5}$$

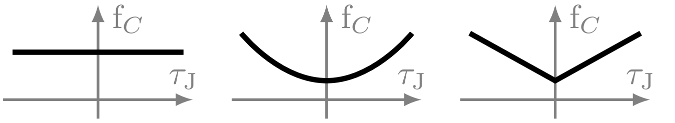
A. Rigid-Body Dynamics

The Recursive Newton-Euler Algorithm [8] is used to describe the RBD of the robot. To allow for the linear description in (4), the inertia tensors are defined relative to the joint Center of Rotation (CoR). The reference frames of the inertia tensors are then translated to the center-of-mass (CoM) positions of the links using the Parallel Axes Theorem (Steiner's Law). Each rigid body is parameterized dynamically by the set of standard inertial parameters (six inertia components, three first moments of mass, and the mass), i.e. $\{XX_j \ XY_j \ XZ_j \ YY_j \ YZ_j \ ZZ_j \ mX_j \ mY_j \ mZ_j \ m_j\}$.

1) Base Inertial Parameters: The minimum set of inertial parameters, commonly referred to as the set of Base Inertial Parameters (BIP), is determined using a two-step numerical procedure based on a QR decomposition of the observation matrix. The numerical procedure is based on the ideas suggested in [9] and used for instance in [10]. Our procedure consists of two steps:

Step 1. The total number of base parameters is determined. Inspired by the work in [11], dummy observations are used to iteratively construct an observation matrix. The number of BIP are obtained as the maximum rank of the observation matrix in the iterative procedure.

Step 2. The specific set of BIP is determined from a QR factorization of the dummy observation matrix. The BIP are identified as those diagonal elements of the (upper triangular) R matrix having values larger than numerical zero (usually $> 10^{-12}$ depending on floating point representation).



(a) none (b) square (c) absolute

Fig. 2: The available Coulomb friction models and their dependence on the joint torque.

B. Joint Dynamics

The Joint Dynamics (JD) are assumed uncoupled among the joints and similar for all joints, thus we describe here the dynamics of a single joint only and omit joint indices.

The JD τ_JD of a robot joint is considered a combination of a Coulomb friction model f_C, a generalized hysteresis model $\mathcal{H}$, and a viscous friction model f_v, specifically

$$\tau_\mathrm{JD} = \mathrm{f}_C(\tau_\mathrm{J})\,\mathcal{H}(\mathbf{z},\dot{q}) + \mathrm{f}_v(\dot{q}), \tag{6}$$

with the joint torque $\tau_\mathrm{J} = \tau_{\mathrm{RBD},j} = \mathbf{Y}_{\mathrm{RBD},j}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\,\boldsymbol{\zeta}_\mathrm{RBD}$ for any joint j with $\mathbf{Y}_{\mathrm{RBD},j}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ the j^th row of $\mathbf{Y}_\mathrm{RBD}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$. For convenience, we let $\boldsymbol{\zeta}_\mathrm{RBD} \triangleq \mathbf{e}_j$ where $\mathbf{e}_j$ is the j-element column vector whose entries are all 1's. This allows to obtain the joint torque (basis) simply by summing the elements of $\mathbf{Y}_{\mathrm{RBD},j}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$.

1) Coulomb Friction: The Coulomb friction is considered an even and strictly positive function of its scalar argument (the joint torque). The following models have been implemented (see Fig. 2): No joint torque dependence (a), squared joint torque dependence (b), and absolute joint torque dependence (c).

2) Generalized Hysteresis: The generalized hysteresis is a real-valued function of the angular velocity and possibly a finite number of internal states. The generalized hysteresis function is at most unitary in magnitude, i.e. $|\mathcal{H}(\mathbf{z},\dot{q})| \leq 1,\ \forall\,\mathbf{z}\in\mathbb{R}^n, \dot{q}\in\mathbb{R}$. The *sgn* function has been implemented, hence using the `square` joint torque dependence yields

$$\mathrm{f}_C(\tau_\mathrm{J})\,\mathcal{H}(\mathbf{z},\dot{q}) = (F_{C,0} + F_{C,1}\,\tau_\mathrm{J}^2)\,\mathrm{sgn}(\dot{q}). \tag{7}$$

3) Viscous Friction: The viscous friction is considered an odd and continuous function of the same sign as its scalar argument (the angular velocity). Let the set $S \subset \mathbb{Z}$ (positive integers), then the polynomial of viscous friction

$$\mathrm{f}_v(\dot{q}) = \sum_{i\in S} F_{v,i}\,b_i, \quad b_i = \begin{cases} |\dot{q}|\,\dot{q}^{i-1}, & \text{if } i \text{ is even} \\ \dot{q}^i, & \text{otherwise} \end{cases} \tag{8}$$

with $F_{v,i}$ the viscous coefficient of friction corresponding to the integer element i of S.

As an example, using the `square` Coulomb friction model and the set $S = \{1,\ 2,\ 4\}$ of viscous friction terms, the JD coefficients

$$\boldsymbol{\zeta}_{\mathrm{JD},j} = [\,F_{C,0}\ \ F_{C,1}\ \ F_{v,1}\ \ F_{v,2}\ \ F_{v,4}\,]^\mathsf{T}, \tag{9}$$

and the corresponding regressor

$$\mathbf{Y}_{\mathrm{JD},j} = [\,\mathrm{sgn}(\dot{q})\ \ \tau_\mathrm{J}^2\,\mathrm{sgn}(\dot{q})\ \ b_1\ \ b_2\ \ b_4\,]^\mathsf{T}, \tag{10}$$

with b_i defined according to (8).

IV. Calibration

The measurements (usually actuator currents or joint torques) $\boldsymbol{\Gamma} \equiv \lambda\boldsymbol{\tau}$ for some (scalar) constant λ and angular positions $\mathbf{q}$ are sampled at times $t(k) = k\,T_S$, $k = 1,\ldots,\bar{M}$, and T_S the sampling period.

The measured angular positions $\mathbf{q}$ are filtered by a 4th order Butterworth filter. The filtering is performed in both the forward and reverse directions to eliminate lag of the filtered trajectories $\hat{\mathbf{q}}$. To keep the useful signal of the robot dynamics in the filter bandwidth, the cut-off frequency of the filter is chosen to be 5 times the frequency $f_\mathrm{dyn} = 10$ Hz of the robot dynamics, i.e. 50 Hz – see [12] for a discussion on this filtering operation. A central finite difference procedure (compact stencil) is used to obtain the angular velocities $\dot{\hat{\mathbf{q}}}$ and angular accelerations $\ddot{\hat{\mathbf{q}}}$ from $\hat{\mathbf{q}}$.

The measurement vector $\bar{\mathbf{y}} = \begin{bmatrix}\bar{\mathbf{y}}_1^\mathsf{T} & \cdots & \bar{\mathbf{y}}_N^\mathsf{T}\end{bmatrix}^\mathsf{T}$, with $\bar{\mathbf{y}}_j = \begin{bmatrix}\Gamma_j(1) & \cdots & \Gamma_j(\bar{M})\end{bmatrix}^\mathsf{T}$, and the observation matrix $\bar{\mathbf{W}} = \begin{bmatrix}\bar{\mathbf{W}}_1^\mathsf{T} & \cdots & \bar{\mathbf{W}}_N^\mathsf{T}\end{bmatrix}^\mathsf{T}$ with $\bar{\mathbf{W}}_j = \begin{bmatrix}\mathbf{Y}_j(\hat{\mathbf{q}}(1),\dot{\hat{\mathbf{q}}}(1),\ddot{\hat{\mathbf{q}}}(1))^\mathsf{T} & \cdots & \mathbf{Y}_j(\hat{\mathbf{q}}(\bar{M}),\dot{\hat{\mathbf{q}}}(\bar{M}),\ddot{\hat{\mathbf{q}}}(\bar{M}))^\mathsf{T}\end{bmatrix}^\mathsf{T}$ the j^th row of the regressor evaluated in the filtered trajectory.

The sampling frequency f_S is assumed much higher than the frequencies of interest in the dynamics, so to limit the required computational resources and to avoid that $\bar{\mathbf{W}}$ and $\bar{\mathbf{y}}$ are statistically correlated with the noise $\boldsymbol{\rho}$, the dynamics is subject to parallel filtering and decimation/downsampling to obtain $\mathbf{W} = \begin{bmatrix}\mathbf{W}_1^\mathsf{T} & \cdots & \mathbf{W}_N^\mathsf{T}\end{bmatrix}^\mathsf{T}$ and $\mathbf{y} = \begin{bmatrix}\mathbf{y}_1^\mathsf{T} & \cdots & \mathbf{y}_N^\mathsf{T}\end{bmatrix}^\mathsf{T}$ with $\mathbf{W}_j$ and $\mathbf{y}_j$ the parallel filtered and downsampled $\bar{\mathbf{W}}_j$ and $\bar{\mathbf{y}}_j$, respectively. The parallel filtering is performed using a 4th order Butterworth filter with a cut-off frequency of $2\,f_\mathrm{dyn} = 20$ Hz (again eliminating lag by filtering in both directions) and the downsampling factor $F_\mathrm{DS} = 0.8/(4\,f_\mathrm{dyn}\,T_S)$, as discussed in [12].

For $M > n_p$ the following over-determined linear system is obtained

$$\mathbf{y} = \mathbf{W}\,\hat{\boldsymbol{\zeta}} + \boldsymbol{\rho} \tag{11}$$

with the noise $\boldsymbol{\rho} = \begin{bmatrix}\boldsymbol{\rho}_1^\mathsf{T} & \cdots & \boldsymbol{\rho}_N^\mathsf{T}\end{bmatrix}^\mathsf{T}$ assumed to be additive, have zero mean, be serially uncorrelated, and be heteroskedastic, thus the covariance matrix

$$\boldsymbol{\Omega} = \mathbb{E}(\boldsymbol{\rho}\,\boldsymbol{\rho}^\mathsf{T}) = \mathrm{diag}\big(\sigma_{\rho,1}^2\,\mathbf{E}_M \quad \cdots \quad \sigma_{\rho,N}^2\,\mathbf{E}_M\big)\,, \tag{12}$$

with $\sigma_{\rho,j}$ the standard deviation of $\boldsymbol{\rho}_j$, $\mathbb{E}$ the expectation operator, and $\mathbf{E}_M$ the $M\times M$ identity matrix. The WLS solution is obtained as

$$\hat{\boldsymbol{\zeta}}_\mathrm{WLS} = \big(\mathbf{W}^\mathsf{T}\,\boldsymbol{\Omega}^{-1}\,\mathbf{W}\big)^{-1}\,\mathbf{W}^\mathsf{T}\,\boldsymbol{\Omega}^{-1}\,\mathbf{y}. \tag{13}$$

Usually, such weighting operations normalize the error terms such that with $\tilde{\rho} = \boldsymbol{\Omega}^{-1/2}\,\boldsymbol{\rho}$, one has $\boldsymbol{\Sigma}_{\tilde{\rho}\rho} = \mathbb{E}(\tilde{\rho}\,\boldsymbol{\rho}^\mathsf{T}) = \boldsymbol{\Omega}^{-1/2}\,\mathbb{E}(\boldsymbol{\rho}\,\boldsymbol{\rho}^\mathsf{T})\,\boldsymbol{\Omega}^{-1/2} = \mathbf{E}_M$. An unbiased estimation of $\sigma_{\rho,j}$ is obtained from the Ordinary Least Squares solution of subsystem j by computing the variance of system j as

$$\sigma_{\rho,j}^2 = \frac{||\mathbf{y}_j - \mathbf{W}_j\,\hat{\boldsymbol{\zeta}}||^2}{M - n_p}, \tag{14}$$

with M the number of samples (after the downsampling operation), n_p the number of dynamics coefficients, and $\mathbf{W}_j$

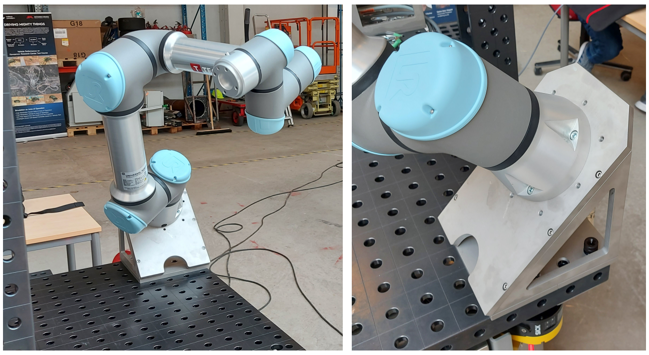


Fig. 3: The Universal Robots UR5e robot manipulator mounted at a 45° slope relative to gravity.

j	α_{j-1}	a_{j-1}	d_j
1	0	0	0.1625
2	$\pi/2$	0	0
3	0	-0.425	0
4	0	-0.3922	0.1333
5	$\pi/2$	0	0.0997
6	-$\pi/2$	0	0.0996

```
alpha,a,d
pi/2,0,0.1625
0,-0.425,0
0,-0.3922,0
pi/2,0,0.1333
-pi/2,0,0.0997
0,0,0.0996
```

TABLE I: MDH parameters of the UR5e robot (left) and equivalent representation in `mdh.csv` (right).

the observation matrix related to joint j, i.e. the j^{th} row of the regressor evaluated in the M observations [13], [14].

V. Case Studies

For the experimental verification of AURT, we use the Universal Robots UR5e robot manipulator mounted on a 45° slope as shown in Fig. 3 with MDH parameters listed in Table I. AURT includes the code to reproduce this example.

We present two case studies: A full robot dynamics model calibration and a reduced model calibration for demonstrating the prediction capabilities using the different Coulomb friction models illustrated in Fig. 2. For both case studies, the actuator currents and angular positions are sampled at 500 Hz using our `URInterface` software [15].

A. Full Robot Dynamics Calibration

The trajectory used for calibration and validation purposes is shown in Fig. 4. The 45° rotation around $\mathbf{x}_0$ yields the gravity components

$$\mathbf{g} = \mathbf{R}_x(45°)[0 \;\; 0 \;\; -9.81]^\mathsf{T} = [0 \;\; 6.937 \;\; -6.937]^\mathsf{T} \quad (15)$$

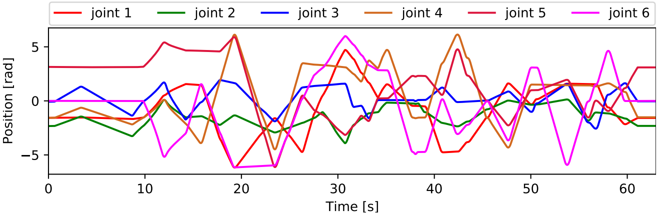


Fig. 4: Trajectory used in the calibration and validation of the Universal Robots UR5e robot dynamics model.

Joint	1	2	3	4	5	6
MSE	0.0581	0.1372	0.1365	0.0125	0.0055	0.0199
NMSE	0.0333	0.0455	0.0981	0.0487	0.0161	0.0414

TABLE II: Model prediction errors for each joint.

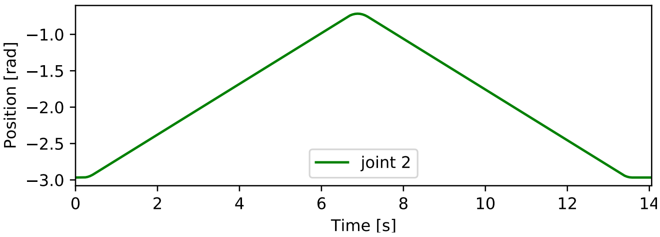


Fig. 5: Trajectory for joint 2 used in the evaluation of the different Coulomb friction models.

Thus, the rigid-body dynamics are generated by `compile-rbd` having a total of 38 BIP. The robot dynamics are generated by the `compile-rd` command with `friction-torque-model` as `square` and `friction-viscous-powers` as `1 2 3`. The Mean Squared Error (MSE) and Normalized MSE (NMSE) of each joint are listed in Table II, the normalization factor being the average signal amplitude. The prediction capability on the validation data demonstrates that AURT is able to produce a high fidelity robot dynamics model. The prediction errors are generally low compared to the magnitude of the signals as indicated by the NMSE.

B. Coulomb Friction Models

To demonstrate the performance of the different choices of Coulomb friction models, the trajectory of Fig. 5 is performed. A simplified (1-link) model of the robot manipulator is assumed. Three robot dynamics models are developed differentiated by the choice of `friction-torque-model` being either `none`, `square`, or `absolute` as illustrated in Fig. 2. The measured and model predicted actuator currents are shown in Fig. 6 and MSEs for each model is listed in Table III. The results show little difference in model performance for models `square` and `absolute` with a slightly smaller MSE for the `absolute` model. Model `none`, which does not take into account the joint torque dependence of the Coulomb friction model, yield a MSE more than twice as large compared to the `square` and `absolute` models. The `square` model is expected to perform better compared to the `absolute` model for joint torques larger than those present in this experiment. In this experiment the allowable domain of joint torque was not explored.

Model type	`square`	`absolute`	`none`
MSE	0.006657	0.006474	0.015314
NMSE	0.001906	0.001832	0.004672

TABLE III: Model prediction errors for joint 2 using the different Coulomb friction models of Fig. 2.

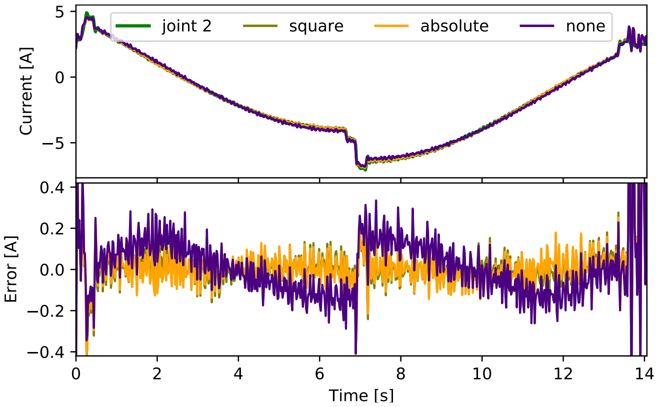


Fig. 6: Actuator current measured and prediction by the Coulomb friction models embedded in AURT performing the trajectory of Fig. 5. The MSEs and NMSEs are listed in Table III.

VI. Related Work

In the state of the art, the tools used for manipulator modeling can be organized as (see classification in Table IV):

Code Generators – which generate efficient code that represent the robot model.

Libraries – designed to be included within the code of a complete solution rather than just being invoked. Usually allow direct calculation of positions, accelerations and forces for specific moments and operating conditions, and in some cases, it is also possible to modify and calibrate the robot characteristics specified during the execution time.

All the surveyed tools have different features as summarized in Table IV. Most tools, particularly code generators, focus on efficiency, sacrificing ability to use the models for calibration, or are closed source. Compared to the state-of-the-art, AURT provides support for calibration with advanced joint dynamics models i.e., with a generalized polynomial and joint torque dependent Coulomb friction.

1) Algorithms: Each tool use their corresponding linear algebra package to perform the transformations required by the kinematic calculation or the dynamic algorithms. AURT, like most flexible or open tools, use the Recursive Newton-Euler Algorithm (RNEA) as the algorithm to calculate the inverse dynamics of the robot. OpenSYMORO is the only tool that also allows the use of a Lagrange method.

VII. Conclusion & Future Work

We have introduced AURT, a tool for dynamics modeling and calibration of robot manipulators. The symbolic dynamics model is developed using the Recursive Newton-Euler Algorithm based on the provided modified DH parameters. Various different joint dynamics models are available and the tool – due to its separation of rigid-body dynamics and joint dynamics – allows for fast calibration of a robot dynamics model using different joint dynamics models.

AURT was used to describe the dynamics of the Universal Robots UR5e robot. A motion was performed by the UR5e robot while data was sampled and recorded. The data was used to calibrate the dynamics coefficients of the robot dynamics model and the calibrated model was validated on another set of data. The validation results showed normalized mean squared errors in the range from 0.0161 for joint 5 to 0.0981 for joint 3. The tool and code to reproduce the example are available online. Another motion was performed to demonstrate how the joint torque dependent Coulomb friction models embedded in AURT outperforms the standard approach in the state of the art.

Our future developments will consist of:

- Extending the available types of joint dynamics models to include hysteresis effects by implementing dynamic friction models such as the LuGre model [24] or the Generalized Maxwell-Slip (GMS) model [25], the latter having been recently extended to describe joint torque dependent Coulomb friction [26].
- Allow for the user to optionally further reduce the set of BIP to a set of Essential Inertial Parameters (EIP).
- Provide support for co-simulation [27] by exporting Functional Mockup Units [28].
- In the short term, AURT is being integrated into a toolchain that enables automated configuration of manufacturing cells [29].

References

[1] N. Kousi, C. Gkournelos, S. Aivaliotis, C. Giannoulis, G. Michalos, and S. Makris, “Digital twin for adaptation of robots’ behavior in flexible robotic assembly lines,” vol. 28, pp. 121–126.

[2] H. Feng, C. Gomes, C. Thule, K. Lausdahl, A. Iosifidis, and P. G. Larsen, “Introduction to Digital Twin Engineering,” in *The Annual Modeling and Simulation Conference*, p. to appear.

[3] E. Madsen, O. S. Rosenlund, D. Brandt, and X. Zhang, “Comprehensive modeling and identification of nonlinear joint dynamics for collaborative industrial robot manipulators,” vol. 101, p. 104462.

[4] M. Gautier and W. Khalil, “Identification of the minimum inertial parameters of robots,” in *Proceedings, 1989 International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, pp. 1529–1534.

[5] M. Gautier, “Numerical calculation of the base inertial parameters of robots,” vol. 8, no. 4, pp. 485–506.

[6] E. Madsen, D. Tola, and C. Gomes, “Aarhus University Robotics Toolbox (AURT),” 8 2021. [Online]. Available: https://github.com/INTO-CPS-Association/aurt

[7] J. Denavit and R. S. Hartenberg, “A Kinematic Notation of Lower-Pair Mechanisms Based on Matrices,” *Trans. ASME E, Journal of Applied Mechanics*, vol. 22, pp. 215–221, Jun. 1995.

[8] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, “On-Line Computational Scheme for Mechanical Manipulators,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 2, pp. 69–76, jun 1980.

[9] M. Gautier, “Numerical calculation of the base inertial parameters of robots,” in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, 1990.

[10] A. Janot, P. O. Vandanjon, and M. Gautier, “Identification of 6 DOF Rigid Industrial Robots with the Instrumental Variable Method,” *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 1659–1664, jul 2012.

[11] J. Klodmann, D. Lakatos, C. Ott, and A. Albu-Schäffer, “A Closed-Form Approach to Determine the Base Inertial Parameters of Complex Structured Robotic Systems,” *IFAC-PapersOnLine*, vol. 48, no. 1, pp. 316–321, 2015.

[12] M. Gautier, A. Janot, and P.-O. Vandanjon, “A New Closed-Loop Output Error Method for Parameter Identification of Robot Dynamics,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 428–444, Mar. 2013.

[13] M. Gautier and P. Poignet, “Extended Kalman Filtering and Weighted Least Squares Dynamic Identification of Robot,” *IFAC Proceedings Volumes*, vol. 33, no. 15, pp. 935–940, jun 2000.

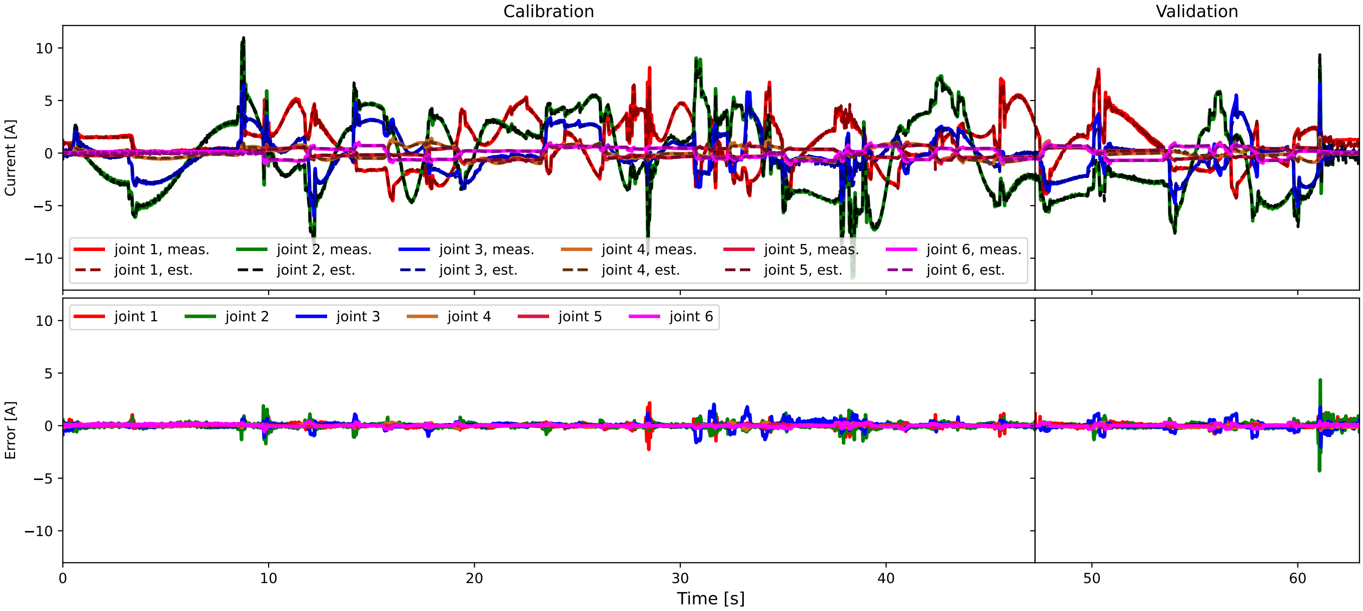


Fig. 7: Calibration and validation results for the Universal Robots UR5e robot manipulator.

Tool	Type	Access	Reference	Symbolic Model	Joint Dynamics Model	Calibration Support	Latest Release
SD-Fast	Code generator	Closed	[16]	✗	User defined	✓	?
Robotran	Code generator	Flexible	[17]	✓	✗	✗	2021
RobCoGen	Code generator	Open	[18]	✗	✗	✗	2020
RigidBodyDynamics.jl	Library	Open	[19]	✓	Coulomb, viscous	✗	2021
RBDL	Library	Open	[20]	✓	Coulomb	✗	2018
Pinocchio	Library	Open	[21]	✓	Coulomb, viscous	✗	2021
OpenSYMORO	Library	Open	[22]	✗	Coulomb, viscous	✗	2014
ROSdyn	Library	Open	[23]	✓	Coulomb, viscous (1st and 2nd degree)	✓	2021
AURT	Code gen. & Lib.	Open		✓	Torque dependent Coulomb, viscous (n^{th} degree)	✓	2021

TABLE IV: Summary of related works.

[14] ——, "Extended Kalman filtering and weighted least squares dynamic identification of robot," *Control Engineering Practice*, vol. 9, no. 12, pp. 1361–1372, dec 2001.

[15] "URInterface," https://gitlab.au.dk/clagms/urinterface, accessed: 2021-08-05.

[16] SD/FAST. [Online]. Available: https://support.ptc.com/support/sdfast/index.html

[17] N. Docquier, A. Poncelet, and P. Fisette, "ROBOTRAN: A powerful symbolic gnerator of multibody models," vol. 4, no. 1, pp. 199–219.

[18] M. Frigerio, "RobCoGen: A code generator for efficient kinematics and dynamics of articulated robots, based on Domain Specific Languages."

[19] T. Koolen and R. Deits, "Julia for robotics: Simulation and real-time control in a high-level programming language," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 604–611.

[20] RBDL: An efficient rigid-body dynamics library using recursive algorithms — SpringerLink. [Online]. Available: https://link.springer.com/article/10.1007/s10514-016-9574-0

[21] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, jan 2019.

[22] W. Khalil, A. Vijayalingam, B. Khomutenko, I. Mukhanov, P. Lemoine, and G. Ecorchard, "OpenSYMORO: An open-source software package for symbolic modelling of robots," in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, jul 2014.

[23] ROSdyn. ROSIN. [Online]. Available: https://www.rosin-project.eu/ftp/rosdyn

[24] C. C. de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, mar 1995.

[25] V. Lampaert, F. Al-Bender, and J. Swevers, "A generalized Maxwell-slip friction model appropriate for control purposes," in *2003 IEEE International Workshop on Workload Characterization (IEEE Cat. No.03EX775)*. IEEE, 2003.

[26] E. Madsen, O. S. Rosenlund, D. Brandt, and X. Zhang, "Adaptive feedforward control of a collaborative industrial robot manipulator using a novel extension of the Generalized Maxwell-Slip friction model," *Mechanism and Machine Theory*, vol. 155, p. 104109, jan 2021.

[27] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: A Survey," vol. 51, no. 3, pp. 49:1–49:33.

[28] A. Junghanns, T. Blochwitz, C. Bertsch, T. Sommer, K. Wernersson, A. Pillekeit, I. Zacharias, M. Blesken, P. Mai, K. Schuch, C. Schulze, C. Gomes, and M. Najafi, "The Functional Mock-up Interface 3.0 - New Features Enabling New Applications," in *Proceedings of the 14th International Modelica Conference*. Linköping University Electronic Press, Linköpings Universitet, p. to be published. [Online]. Available: https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=169&Article_No=16

[29] D. Tola, C. Gomes, C. Schultz, C. Schlette, C. Hansen, and L. Esterle, "RoboCIM: Towards a Domain Model for Industrial Robot System Configurators," in *5th International Joint Conference on Rules and Reasoning*, p. to be published.