

# Experimental Leg Inverse Dynamics Learning of Multi-Legged Walking Robot

Jiří Kubík<sup>[0000-0002-2219-5764]</sup>, Petr Čížek<sup>[0000-0001-6722-3928]</sup>, Rudolf Szadkowski<sup>[0000-0003-4075-116X]</sup>, and Jan Faigl<sup>[0000-0002-6193-0792]</sup>

Department of Computer Science, Faculty of Electrical Engineering  
Czech Technical University in Prague  
Technická 2, 166 27, Prague 6, Czech Republic  
{kubikji2, petr.cizek, szadkrud, faigl.j}@fel.cvut.cz,  
WWW home page: <https://comrob.fel.cvut.cz/>

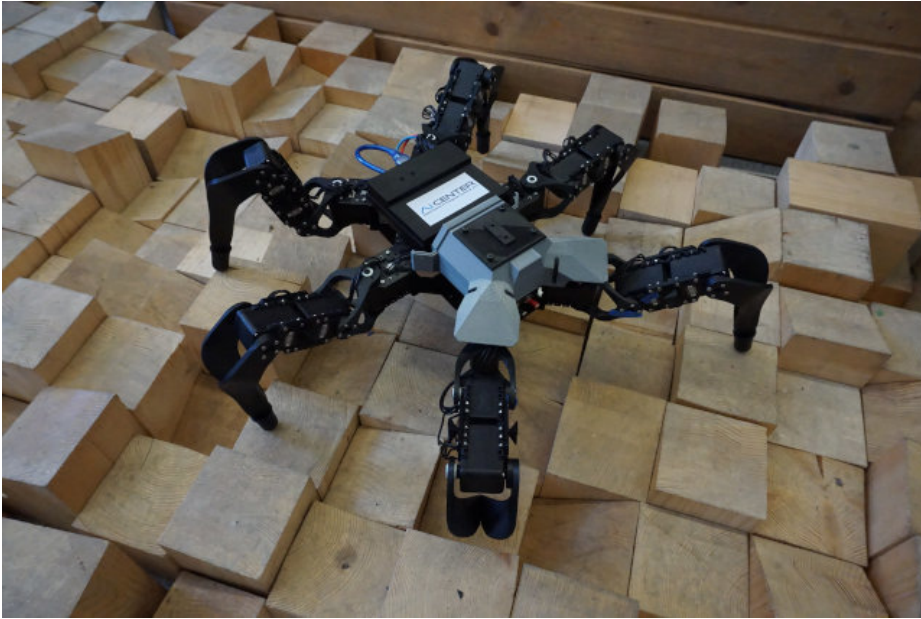
**Abstract.** Rough terrain locomotion is a domain where multi-legged robots benefit from their relatively complex morphology compared to the wheeled or tracked robots. Efficient rough terrain locomotion requires the legged robot sense contacts with the terrain to adapt its behavior and cope with the terrain irregularities. Usage of inverse dynamics to estimate the leg state and detect the leg contacts with the terrain suffers from computational complexity. Furthermore, it requires a precise analytical model identification that does not cope with adverse changes of the leg parameters such as friction changes due to the joint wear, the increased weight of the leg due to the mud deposits, and possible leg morphology change due to damage. In this paper, we report the experimental study on the locomotion performance with machine learning-based inverse dynamics model learning. Experimental examining three different learning models show that a simplified model is sufficient for leg collision detection learning. Moreover, the learned model is faster for calculation and generalizes better than more complex models when the leg parameters change.

## 1 Introduction

Enhanced terrain traversability of multi-legged robots [15] stems from their relatively complex morphology but comes at the cost of complex locomotion control [19]. A critical part of the multi-legged locomotion control is the robot state estimation, including timely and reliable tactile sensing to detect the leg contact with the ground or obstacles [10,4,2]. The leg contact detection and leg-state estimation, i.e., assessing whether the leg is supporting the body or not, are essential in maintaining the attitude of the robot in complex terrains [2,8], and for the accuracy of the legged-odometry [10,3,4,12]. Further, the foot-contact detection is utilized to synchronize oscillations in controllers based on neural oscillators [1,5] or to trigger reflexive behaviors [5,7].

Model-based locomotion control methods [17] use inverse dynamics model in a contact detection [9]. Their applicability in real-world mobile robotic applications may be cumbersome due to the difficulty of accurately determining

the various kinematic and dynamic parameters of such analytical models. It can be especially expected in deployments with increasingly complex scenarios [11,16], where robots might struggle in challenging environments, and their characteristics might change significantly. Automated parameter identification and online adaptation of models are beneficial strategies because they can capture non-stationarities in the mechanical properties of the robot [18]. Such non-stationarities include the adverse changes of the leg parameters, e.g., friction changes due to the joint wear, increased weight of the leg due to the mud deposits, or leg morphology change because of damage.



**Fig. 1.** Hexapod walking robot SCARAB (Slow-Crawling Autonomous Reconnaissance All-terrain Bot) used for the experimental study of machine-learned inverse dynamics.

This work reports on the experimental study of the machine learning (ML) based inverse dynamics model learning to the locomotion performance of a small affordable hexapod walking robot SCARAB shown in Fig 1. The employed ML approaches include linear regression, second-order polynomial regression, and a three-layered neural network, each learned by the real motion data collected using the experimental hexapod walking platform. The linear and polynomial regression can be considered as statistical methods; however, in the experimental evaluation, we prefer lightweight techniques suitable for online model learning instead of methods that require extensive training datasets [14]. The performance of the learned models has been examined with a focus on the following aspects.

1. Comparison of ML-based inverse dynamics models with the baseline analytical model [6].
2. Model performance w.r.t. the size of the training dataset.
3. Computational complexity of model learning and prediction.
4. Robustness of the learned model to a non-stationary environment.
5. Performance of the collision detection integrated with the ML-based model compared to the baseline approach [6].

The main challenge of the addressed problem is to learn the leg inverse dynamics model to predict the future state of the leg using the current motion command and the current leg state. The prediction is used to close the feedback loop in the leg-contact detection process via the leg state monitoring [9]. ML approaches are well applicable to the addressed problem, as the leg state is influenced by numerous factors, including the previous trajectory of the leg. The performed experiments indicate that the learned dynamic model provides a similar performance of inverse dynamics regression and collision detection as the baseline dynamic model [6], but it is computationally less demanding. The results show a more reliable prediction of the learned model than the baseline model when the leg parameters change, which supports the idea of ML-based adaptive online incrementally learned locomotion controller.

The remainder of the paper is organized as follows. Section 2 details the studied problem and briefly presents the baseline model [6] used in the evaluation. The examined ML regressors are briefly described in Section 3. Results on the experimental deployment are reported in Section 4. Finally, Section 5 is dedicated to concluding remarks.

## 2 Problem Statement

addressed problem is to learn the robot leg inverse dynamics model to predict the collision-free motion of the leg. In this section, a background of the robot leg inverse dynamics model [6] is provided that is utilized in the experimental verification of the studied ML-based models. Besides, the baseline locomotion controller is briefly described in Section 2.2.

### 2.1 Leg Inverse Dynamics Model

The inverse dynamics can be modeled analytically using Euler-Lagrange formulation [17] for the vector of the generalized  $n$ -dimensional coordinates  $\mathbf{q} = \{\theta_1, \theta_2, \dots, \theta_n\}$ , corresponding to the leg joint angles

$$\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}, \quad (1)$$

where  $\mathbf{D}(\mathbf{q})$  is the *inertia matrix* of the chain of the rigid bodies,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$  is a tensor representing the *centrifugal* and *Coriolis* effects induced on the joints,  $\mathbf{G}(\mathbf{q})$  is the vector of moments generated at the joints by the *gravitational acceleration*, and  $\boldsymbol{\tau}$  is the vector of actuation torques at the respective joints. All

the terms  $D(\mathbf{q})$ ,  $C(\mathbf{q}, \dot{\mathbf{q}})$ , and  $G(\mathbf{q})$  depend on a set of parameters that has to be identified. The most influencing parameters, w.r.t. the precision of the inverse dynamic model, are the leg links inertia matrices and the estimated frictions in the leg joints. Due to the complexity of the calculation and measurement of the inertia matrices, simplified models such as point mass and rigid-rod models are used for the model calculation, which introduces error into the prediction of the inverse dynamics. Moreover, the inertia matrices are most influenced by the non-stationarities that may occur during the robot deployment.

In our particular case of SCARAB, the servo motors provide only the position feedback. Furthermore, the torque nor the electric current is measured, which can be utilized for joint torque estimation. Therefore, an additional step in the inverse dynamics modeling is necessary. The real behavior of the actuator composed of the motor and reduction gear is modeled together with the underlying servo motor controller. The dynamic model is given by

$$J\ddot{q}^M + B\dot{q}^M + F(q^M) + R\tau = KV, \quad (2)$$

where  $q^M$  is the rotor position angle before the reduction,  $J$  is the rotor inertia,  $B$  is the rotor damping,  $F$  is the sum of the static, dynamic, and viscous frictions that depend on the current rotor speed,  $R$  is the gearbox ratio,  $\tau$  is the servo motor torque,  $K$  is the back electromotive force, and  $V$  is the motor voltage. The appropriate values of  $J$ ,  $B$ ,  $F$ ,  $R$ , and  $K$  have to be experimentally identified using the real servo motor and the values specified in the manufacturer datasheet.

The servo motor controller is modeled as the P-type position controller, which sets the voltage as  $V = k_P \cdot err$ , where  $k_P$  is the controller gain, and  $err$  is the difference between the set position and the current position of the actuator. The controller operates with 1 kHz frequency. The complete model of the leg inverse dynamics in the joint angles can be derived by substituting (2) into (1).

The major issue of the analytical inverse dynamic model is the numerous joint-related and link-related parameters that have to be identified before using the inverse dynamics model. The identification process and parametrization of the baseline model are detailed in Section 4.

## 2.2 Hexapod Robot Locomotion Controller

The inverse dynamics model is utilized in the position tracking controller [6] that executes the leg trajectory step-by-step. At each step, the controller reads the current joint angles and compares them to the predicted values provided by the inverse dynamic model. The actuator is iteratively commanded with a new desired position  $\theta_{des}$ , and the tracking continues until the difference between the real measured position  $\theta_{real}$  and the position estimated by the model  $\theta_{est}$  is above the threshold  $\epsilon_{thld}$  that indicate a tactile event is recognized. This simple principle allows for terrain negotiation and rough terrain locomotion even with affordable multi-legged platforms with the position feedback only. However, the performance of the locomotion controller tightly depends on the precision of the inverse dynamics model and identification of its parameters. Therefore, we

aim to employ ML-based techniques for estimating the leg inverse dynamics model to avoid the cumbersome identification of the parameters needed in the analytical model. The ML-based methods considered in our experimental study are described in the following section.

### 3 Learning-based Inverse Dynamics Models

The main motivation behind using the ML-based model of the leg inverse dynamics is to overcome the cumbersome identification of the analytical model. For the considered SCARAB, 18 sets of joint and 18 sets of link parameters have to be found. Additional parameter changes are introduced by the non-stationary electrical and mechanical characteristics of the servo motor that change due to the heating up, gearbox wear-out, and variations in the link shape and mass caused by imperfect 3D printing and also environmental effects. A robust robotic system should overcome parameter variations, but the analytical inverse dynamic model lacks such ability as it requires an online parameter identification step of the adaptive control [17]. Therefore we prospect ML techniques to learn the model.

In the experimental evaluation, we focus on lightweight ML techniques that do not require extensive training datasets like deep-learning-based techniques [14]. We consider three ML approaches: (i) Ordinary Least Squares regression (OLS) further referred to as the *linear regressor*; (ii) Ordinary Least Squares regression with second-order polynomial features denoted the *polynomial regressor*, and (iii) three-layer feedforward neural network with Rectified Linear Unit (ReLU) activation function further referred to as *ReLU regressor*. The used learning input is formed from the  $n$  most recent triplets of the discrete position measurements accompanied by the triplets of the desired positions set to the servo motors further considered with the known baud rate. The regressors are trained to predict the leg dynamics for  $m$  steps to the future. The second-order differential equations (2) are used for the robot leg dynamics. The value of the dynamic variables can be estimated from at least three recent position samples, but we use  $n = 4$  the most recent measurements. The expected leg position is predicted two-steps-ahead  $m = 2$ , as possible delay can occur in the data flow pipeline, and predictions into a more distant future are losing accuracy.

The regressors have been implemented in Python with the Scikit-learn library [13] for the linear and polynomial regressors, whereas the ReLU regressor uses Chainer framework [20] with 100 neurons in the hidden layer and Leaky ReLU activation function. The ReLU regressor hidden layer size has been selected randomly as the hyper-parameter search would require extensive testing. The main aim of this work is to experimentally validate the concept of inverse dynamics learning for the small legged robot. The performance of regressors compared to the baseline analytical model [6] is reported in the following section.

## 4 Experimental Evaluation

The performance of the three regressors of the leg inverse dynamics has been validated in the experimental deployment scenarios with the hexapod walking platform SCARAB shown in Fig. 1. SCARAB is an affordable six-legged robot with 18 controllable degrees of freedom, actuated by 18 Dynamixel AX-12A servo motors. Three servo motors per each leg are named according to the entomology nomenclature (from the body to foot-tip): coxa, femur, and tibia. Each Dynamixel AX-12A actuator enables position control with the internal P-type controller and provides reading its current position at the limited rate of 1000 Hz. All the experiments have been performed using the laptop computer with the dual-core Intel Core i5-3320M CPU @ 2.60 GHz, 16 GB RAM without GPU acceleration, running Ubuntu 18.04 Bionic Beaver operating system with the ROS melodic.

**Table 1.** Mechanical properties of SCARAB

Product	Variable	Measurement	Unit	Description
Coxa	$a_c$	52	mm	Coxa link length
CoM 1	$a_{cc}$	25	mm	Coxa link center of mass position
Mass 1	$m_c$	20	gm	Coxa link mass
Femur	$a_f$	66	mm	Femur link length
CoM 2	$a_{cf}$	20	mm	Femur link center of mass position
Mass 2	$m_f$	115	gm	Femur link mass
Tibia	$a_t$	132	mm	Tibia link length
CoM 3	$a_{ct}$	50	mm	Tibia link center of mass position
Mass 3	$m_t$	62	gm	Tibia link mass

The baseline analytical model [6] is parameterized using mechanical properties as in Table 1 utilized to calculate  $\mathbf{D}$ ,  $\mathbf{C}$ , and  $\mathbf{G}$  of (1). The rigid rod simplified model has been used to calculate the inertia matrices. The dynamic model defined by (2) has been parameterized by values from experimental identification based on measured two reference positions for the actuator moving forth and back without load for different control voltage. The identified minimum voltage is  $v_{\min} = 0.5$  V that defines the maximal static friction as  $F \simeq (k/R_a)v_{\min}$ , where  $k = 3.07 \cdot 10^{-3} \text{N m A}^{-1}$  is the back EMF constant, and  $R_a = 6.5 \Omega$  is the motor resistance, which can be found together with the gearbox ratio  $R = 1/254$  in the actuator datasheet. The values of the parameters have been estimated using the minimum square root method with Euler’s method employed in the solution of (2). The identified parameters of the Dynamixel AX-12A are listed in Table 2.

The experimental examination of the regressors is based on the off-line processed datasets collected using SCARAB. A single leg data has been utilized as all the legs share the same morphology apart from minor differences in the servo

**Table 2.** Dynamic model parameters of the Dynamixel AX-12A

Parameter	Value	Unit
$J$	$1.032 \cdot 10^{-7}$	kg m <sup>2</sup>
$B$	$3.121 \cdot 10^{-6}$	N m s
$F$	$2.369 \cdot 10^{-4}$	N m
$R$	$3.937 \cdot 10^{-3}$	-
$K$	$3.912 \cdot 10^{-3}$	N m A <sup>-1</sup>

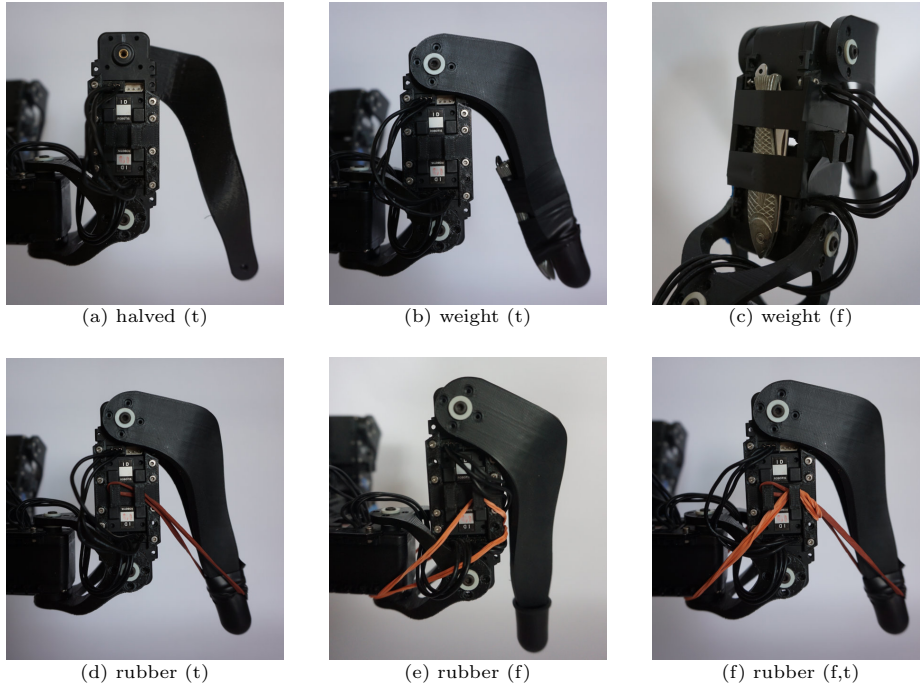
motor orientation and offset angles. Nine datasets have been collected, capturing different leg movements with various induced non-stationarities that alter the leg parameters. The individual datasets listed in Table 3 and the made leg modifications are depicted in Fig. 2.

**Table 3.** List of collected datasets

#	Dataset name	Dataset size [ $n$ ]	Induced modifications
1	Vanilla	55 333	No modifications
2	halved(t)	27 859	Weight of tibia link reduced by 12 g (see Fig. 2a)
3	weight(t)	27 033	Weight of tibia link increased by 31 g (see Fig. 2b)
4	weight(f)	27 603	Weight of femur link increased by 31 g (see Fig. 2c)
5	loosen(t)	26 994	Tibia link freely moving regardless of tibia servo position
6	rubber(t)	27 277	Tibia joint load increased with rubber band (see Fig. 2d)
7	rubber(f)	27 273	Femur joint load increased with rubber band (see Fig. 2e)
8	rubber(f,t)	27 493	Merge of rubber(t) and rubber(f) setups (see Fig. 2f)
9	collision	100	Different obstacles placed in the pathway of the leg to induce leg collisions

The datasets 1 and 2–8 have been collected using 2000 and 1000 randomly chosen target points within the leg’s operational space, respectively, and interpolating the path between the targets with the maximum allowed step size of 0.4 mm, which is transferred into the joint coordinates using inverse kinematics. The path in joint coordinates is then executed in the open-loop by commanding the leg servo motors with the desired joint angles. For all the datasets, the desired joint angle  $\theta_{\text{des}}$  and the real (measured) joint angle  $\theta_{\text{real}}$  were collected from the daisy-chained leg servo motors at the highest possible sampling rate of 100 Hz. The Ordinary Least Squares method is used for training *linear regressor* and *polynomial regressor*, whereas the *ReLU regressor* has been trained using backpropagation.

The performance of the learned regressors is studied in five benchmarks focused on: (1) model precision, (2) model generalization to the cases with the



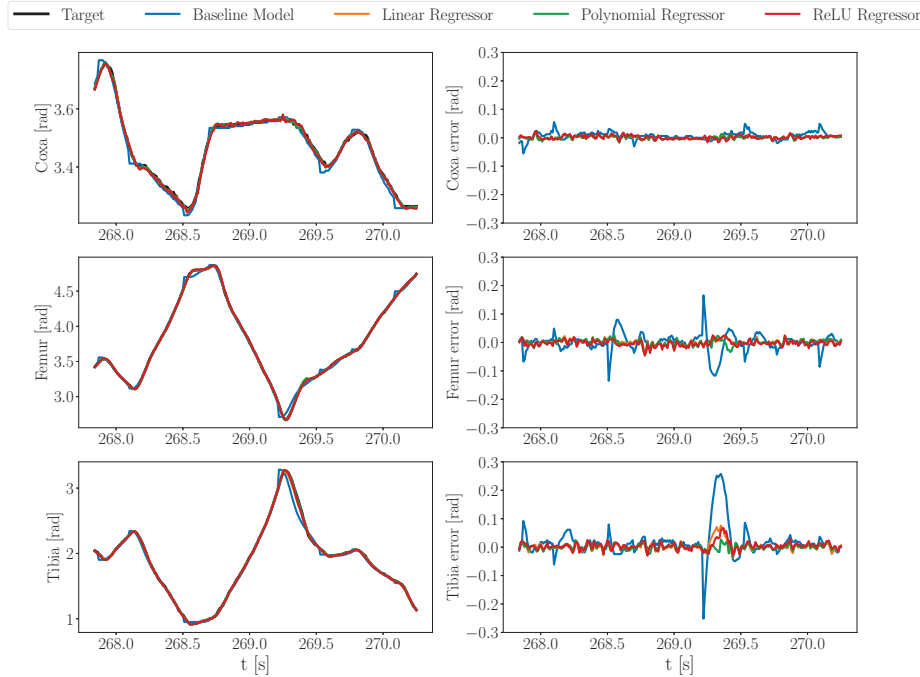
**Fig. 2.** Leg modifications to simulate non-stationarities and alter leg parameters.

induced non-stationarities, (3) size of the training set, (4) computational requirements, and (5) the final deployment in the leg contact detection scenario. In each benchmark, the trained regressors are requested to process the collected time-series testing data per individual sample. The testing error is calculated as the difference  $\theta_{\text{err}} = |\theta_{\text{est}} - \theta_{\text{real}}|$  between the one-step look ahead regressor prediction and the corresponding real measured error. The cumulative mean absolute error (MAE) is then used to report the results.

**Table 4.** Cumulative mean absolute prediction error

Method	Cumulative Mean Absolute Prediction Error [rad]
Baseline dynamic model	0.0170
Linear regressor	0.0069
Polynomial regressor	0.0068
ReLU regressor	0.0075



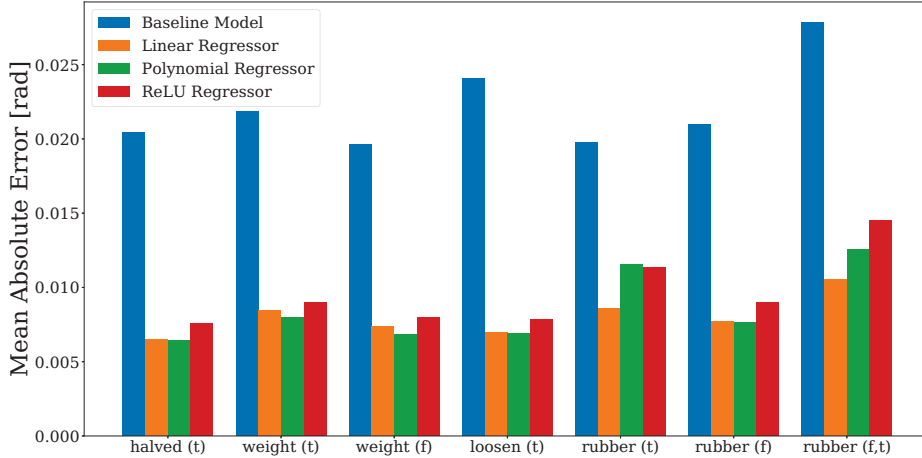


**Fig. 3.** Example of the estimated leg trajectory in joint angles (left column) and the corresponding prediction accuracy calculated as  $\theta_{\text{err}} = |\theta_{\text{est}} - \theta_{\text{real}}|$  (right column) for considered regressors. In the presented example, the leg follows a random trajectory. The MAE of  $\theta_{\text{err}}$  is used to report the results.

*Model precision* has been studied on the regressors learned on the vanilla dataset and compared to the base analytical model. The vanilla dataset has been divided into training and test data with a 0.5:0.5 ratio. The cumulative mean absolute errors are depicted in Table 4, and an example of the estimate positions and the prediction error is shown in Fig. 3. The results indicate that considered ML approaches cope better with the leg position estimation than the baseline model [6].

*Generalization* ability has been examined using regressors learned using the vanilla dataset that has been then utilized for prediction using the datasets 2 to 8 collected on a modified leg mimicking parameter changes. For each scenario, the cumulative mean absolute error over all three servo motors has been computed to examine how regressors generalize leg dynamics and handle changes in its parameters. The results presented in Fig. 4 indicate that the ML-based approaches perform better compared to the baseline model.

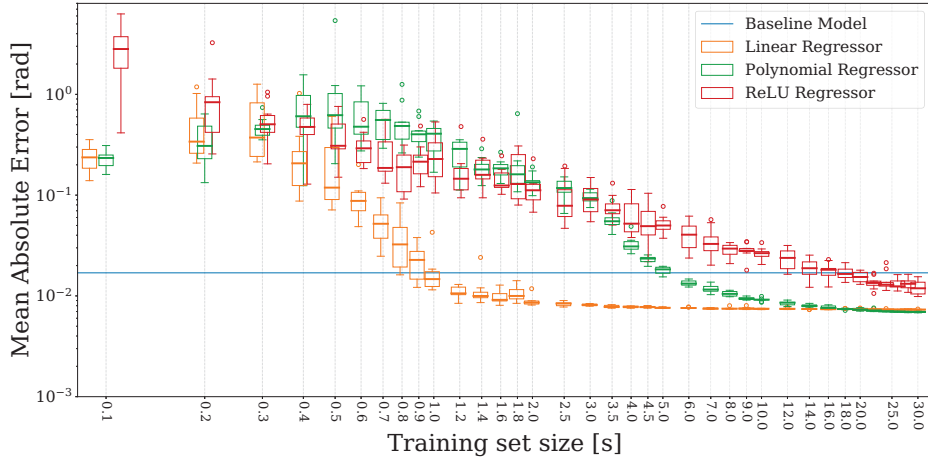
*Size of the training set* influences the quality of the prediction. Besides, a new dataset can be collected when the model becomes inaccurate during the deploy-



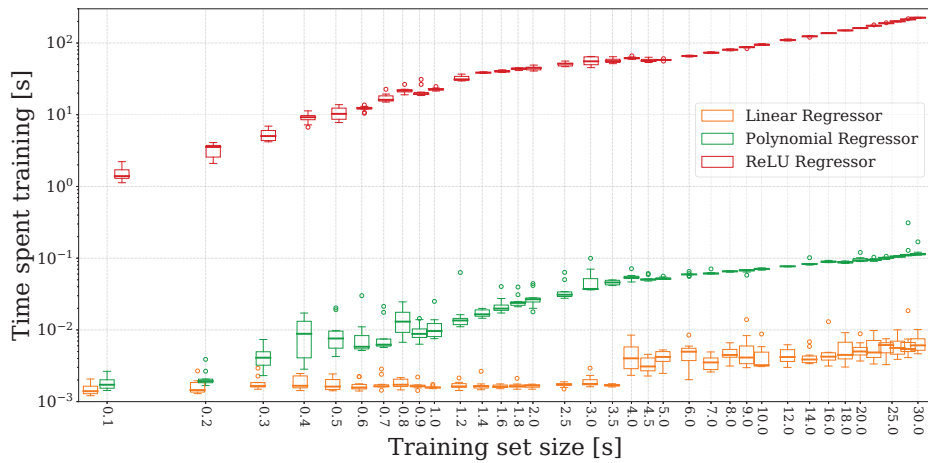
**Fig. 4.** Mean absolute prediction error of the regressors learned using the vanilla dataset in scenarios with differently modified leg morphology.

ment, and the regressor can be retrained in an online learning fashion. Learning from a relatively small batch of data is desirable to enable relearning from data collected in the field. We examine the mean prediction accuracy based on the size of the training set. Since the servo motor joint angle is periodically read at the rate  $\Delta t = 10$  ms, it is possible to directly compute how long it takes to collect a dataset with a particular number of samples. Hence, the size of the vanilla dataset has been utilized to create a sequence of logarithmically increasing time intervals of training data corresponding to the period 0.1 s to 30 s. For each such time interval of  $m$  samples, a random starting point has been selected within the range  $[0, n - m]$ , where the  $n$  is the number of samples in the dataset. Following  $m$  samples have been selected from the vanilla dataset to learn the regressors initialized at random. Ten independent trials have been performed to examine the cumulative mean absolute prediction error. The five-number summary shows the minimum value, lower quartile, median value, upper quartile, and maximum value. The cumulative error per trial is computed using prediction error for all three servo motors of the leg. The influence of prediction error on the training set size is shown in Fig. 5.

The reported results suggest that the size of the training set required to surpass the baseline model by the learned regressor significantly depends on the particular regressor as both the error and its variance decrease with the size of the training set. The ReLU regressor seems to be unsuitable for online learning because a competitive performance with the baseline model is achieved with the considerably large training dataset, which is likely caused by the size of the hidden layer. On the other hand, for the linear and polynomial regressors, it takes only a few seconds of the collected data to surpass the laboriously crafted baseline dynamic model [6].



**Fig. 5.** Cumulative mean absolute prediction error for training set of different size. The shown five-number summary is computed from ten independent trials. Note both axes are in the logarithmic scale.



**Fig. 6.** Required computational time to train the examined regressors based on the size of the training set. The real computational time is shown as the five-number summary.

*Computational requirements* are essential when the method is deployed onboard of the walking robots as computational-demanding methods increase power consumption and decrease the operational time. The required computational time for learning the changed-parameter leg dynamics increases the training time and slows down the average robot speed in online learning. The time spent in prediction might increase the gait control period and thus also decrease the robot speed. The training depends on the size of the training set. Therefore, the required computational time for regressors training has been examined using the

vanilla dataset with logarithmically increasing time intervals of the training data starting at 0.1 s to 30 s. The plot of the five-number summary of the required computational time is depicted in Fig. 6. The mean required computational time for prediction using the baseline model and learned regressors is listed in Table 5.

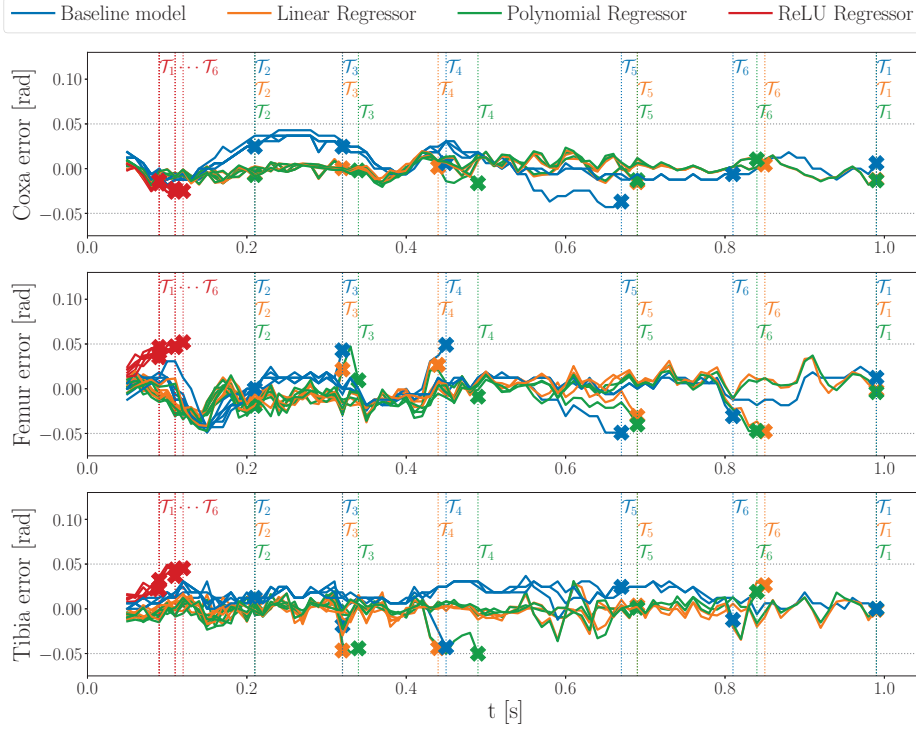
**Table 5.** Mean required computational time for position prediction

Method	Prediction time [ $\mu$ s]
Baseline dynamic model [6]	6.7
Linear regressor	<b>0.5</b>
Polynomial regressor	13.8
ReLU regressor	9.5

The results indicate that real computational requirements are insignificant even for relatively large input data in the linear and polynomial regressors. The ReLU regressor is about several orders of magnitude more demanding because of the underlying backpropagation.

*Contact detection* represents a practical use case of the position prediction that enables the legged robot to negotiate the terrain. In this setup, the leg follows a circular trajectory with the diameter 10 cm, regularly sampled to 100 data points, with period 1 s. The trajectory has been performed in six trials. During the first trial, denoted  $\mathcal{T}_1$ , the leg followed the trajectory freely without any collision. The collected data has been then used for the detection of leg contact with an obstacle. The contact is detected whenever the prediction error  $\theta_{\text{err}} = |\theta_{\text{est}} - \theta_{\text{real}}|$  is above the predefined threshold value  $e_{\text{thld}} = 0.052$  rad. An obstacle has been placed into the leg trajectory for all other trials causing the leg to collide at different trajectory parts. For the trials  $\mathcal{T}_2$ ,  $\mathcal{T}_3$ , and  $\mathcal{T}_4$ , only the foot-tip has been in contact with the obstacle. For  $\mathcal{T}_5$  and  $\mathcal{T}_6$ , the collision occurred with the femur link. The course of the position error  $\theta_{\text{err}}$  shown up to the collision detection is visualized in Fig. 7.

The presented results suggest that the linear and polynomial regressors provide similar performance to the baseline dynamics model. In the descending part of the trajectory, these regressors predict the collisions using a few samples of the baseline model. The linear regressor reports the collision sooner than the baseline model. During the ascending phase of the circular movement, the errors of the regressors' prediction exceed the threshold a few samples late than the baseline. On the other hand, the ReLU regressor failed in all scenarios, which is most likely because the vanilla dataset size is not large enough to train the ReLU model with its 100 neurons in the hidden layer properly, albeit the main prediction error is lower than the baseline model, as shown in Table 4 and Fig 5. As the comparison in Table 4 is based on the mean absolute error, it may cover up erroneous behavior that will only become apparent in the collision detection experiment.



**Fig. 7.** Plots of the prediction accuracy calculated as  $\theta_{\text{err}} = |\theta_{\text{est}} - \theta_{\text{real}}|$  of each leg's joints for the particular trials shown up to the collision detection using the threshold value  $e_{\text{thld}} = 0.052$  rad. The first trial  $\mathcal{T}_1$  is an obstacle-free trajectory. An obstacle has been placed at a different part of the trajectory in the five other trials  $\mathcal{T}_2, \dots, \mathcal{T}_6$ . The annotated vertical lines represent the contact of the corresponding regressor and trial with the respective color-coding.

## 5 Conclusion

Three learning-based approaches for inverse dynamics model learning of hexapod walking robot have been examined and compared with the baseline analytical dynamic model. Based on the reported results from five evaluation scenarios, the performance of the learned models is competitive to the baseline model, which requires laborious identification of the proper values of the model parameters. The learned models achieved higher precision than the baseline approach, and all the regressors demonstrate generalization to changes in the leg properties. The linear and polynomial regressors further show satisfactory performance for the practical deployment in the collision detection scenario. As our future work, we plan to deploy the regressors for online learning in real-life environments.

## Acknowledgment

The presented work has been supported under the OP VVV funded project CZ.02.1.01/0.0/0.0/16\_019/0000765 “Research Center for Informatics” and the Czech Science Foundation (GAČR) under research Project 18-18858S.

## References

1. Aoi, S., Manoonpong, P., Ambe, Y., Matsuno, F., Wörgötter, F.: Adaptive control strategies for interlimb coordination in legged robots: a review. *Frontiers in Neurorobotics* **11**, 39 (2017). DOI: [10.3389/fnbot.2017.00039](https://doi.org/10.3389/fnbot.2017.00039)
2. Bledt, G., Wensing, P.M., Ingersoll, S., Kim, S.: Contact model fusion for event-based locomotion in unstructured terrains. In: *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4399–4406 (2018). DOI: [10.1109/ICRA.2018.8460904](https://doi.org/10.1109/ICRA.2018.8460904)
3. Bloesch, M., Gehring, C., Fankhauser, P., Hutter, M., Hoepflinger, M.A., Siegwart, R.: State estimation for legged robots on unstable and slippery terrain. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 6058–6064 (2013). DOI: [10.1109/IROS.2013.6697236](https://doi.org/10.1109/IROS.2013.6697236)
4. Camurri, M., Fallon, M., Bazeille, S., Radulescu, A., Barasuol, V., G. Caldwell, D., Semini, C.: Probabilistic contact estimation and impact detection for state estimation of quadruped robots. *Robotics and Automation Letters* **2**(2), 1023–1030 (2017). DOI: [10.1109/LRA.2017.2652491](https://doi.org/10.1109/LRA.2017.2652491)
5. Chung, H.Y., Hou, C.C., Hsu, S.Y.: A cpg-inspired controller for a hexapod robot with adaptive walking. In: *Automatic Control Conference (CACCS)*. pp. 117–121. *IEEE* (2014). DOI: [10.1109/CACCS.2014.7097173](https://doi.org/10.1109/CACCS.2014.7097173)
6. Faigl, J., Čížek, P.: Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only. *Robotics and Autonomous Systems* **116**, 136–147 (2019). DOI: [10.1016/j.robot.2019.03.008](https://doi.org/10.1016/j.robot.2019.03.008)
7. Focchi, M., Barasuol, V., Havoutis, I., Buchli, J., Semini, C., Caldwell, D.G.: Local reflex generation for obstacle negotiation in quadrupedal locomotion. *Nature-Inspired Mobile Robotics* pp. 443–450 (2013). DOI: [10.1142/9789814525534\\_0056](https://doi.org/10.1142/9789814525534_0056)
8. Focchi, M., Del Prete, A., Havoutis, I., Featherstone, R., Caldwell, D.G., Semini, C.: High-slope terrain locomotion for torque-controlled quadruped robots. *Autonomous Robots* **41**(1), 259–272 (2017). DOI: [10.1007/s10514-016-9573-1](https://doi.org/10.1007/s10514-016-9573-1)
9. Haddadin, S., De Luca, A., Albu-Schaffer, A.: Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics* **33**(6), 1292–1312 (2017). DOI: [10.1109/TR0.2017.2723903](https://doi.org/10.1109/TR0.2017.2723903)
10. Hwangbo, J., Bellicoso, C.D., Fankhauser, P., Hutter, M.: Probabilistic foot contact estimation by fusing information from dynamics and differential/forward kinematics. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3872–3878 (2016). DOI: [10.1109/IROS.2016.7759570](https://doi.org/10.1109/IROS.2016.7759570)
11. Kolvenbach, H., Wisth, D., Buchanan, R., Valsecchi, G., Grandia, R., Fallon, M., Hutter, M.: Towards autonomous inspection of concrete deterioration in sewers with legged robots. *Journal of Field Robotics* (2020). DOI: [10.1002/rob.21964](https://doi.org/10.1002/rob.21964)
12. Lubbe, E., Withey, D., Uren, K.R.: State estimation for a hexapod robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 6286–6291 (2015). DOI: [10.1109/IROS.2015.7354274](https://doi.org/10.1109/IROS.2015.7354274)

13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830 (2011). DOI: [10.5555/1953048.2078195](https://doi.org/10.5555/1953048.2078195)
14. Polydoros, A.S., Nalpantidis, L., Krüger, V.: Real-time deep learning of robotic manipulator inverse dynamics. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 3442–3448 (2015). DOI: [10.1109/IROS.2015.7353857](https://doi.org/10.1109/IROS.2015.7353857)
15. Raibert, M., Blankespoor, K., Nelson, G., Playter, R.: BigDog, the Rough-Terrain Quadruped Robot. *IFAC Proceedings Volumes* **41**(2), 10822–10825 (Jan 2008). DOI: [10.3182/20080706-5-KR-1001.01833](https://doi.org/10.3182/20080706-5-KR-1001.01833)
16. Rouček, T., Pecka, M., Čížek, P., Bayer, J., Šalanský, V., Heřt, D., Petrлік, M., Bába, T., Spurný, V., Pomerleau, F., Kubelka, V., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., Krajník, T.: Darpa subterranean challenge: Multi-robotic exploration of underground environments. In: *2019 Modelling and Simulation for Autonomous Systems (MESAS)*. pp. 274–290 (2020). DOI: [10.1007/978-3-030-43890-6\\_22](https://doi.org/10.1007/978-3-030-43890-6_22)
17. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control*. Springer Science & Business Media (2010). DOI: [10.1007/978-1-84628-642-1](https://doi.org/10.1007/978-1-84628-642-1)
18. Sigaud, O., Salaun, C., Padois, V.: On-line regression algorithms for learning mechanical models of robots: A survey. *Robotics and Autonomous Systems* **59**(12), 1115 – 1129 (2011). DOI: [10.1016/j.robot.2011.07.006](https://doi.org/10.1016/j.robot.2011.07.006)
19. Tedeschi, F., Carbone, G.: Design Issues for Hexapod Walking Robots. *Robotics* **3**(2), 181–206 (Jun 2014). DOI: [10.3390/robotics3020181](https://doi.org/10.3390/robotics3020181)
20. Tokui, S., Okuta, R., Akiba, T., Niitani, Y., Ogawa, T., Saito, S., Suzuki, S., Uenishi, K., Vogel, B., Yamazaki Vincent, H.: Chainer: A deep learning framework for accelerating the research cycle. In: *25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p. 2002–2011 (2019). DOI: [10.1145/3292500.3330756](https://doi.org/10.1145/3292500.3330756)