Practical Approach to Characterize Realistic Motor Dynamics for Robotic Simulation Independent of the Use Case

Annika Schmidt[1,2], Thomas Gumpert[2], Stefan Schreiber[1,2], Alin Albu-Schäffer[1,2]

***Abstract*— Incorporating realistic actuator dynamics in robotic simulations is key to a successful simulation-to-reality transfer. But real actuation chains are often complex and impossible to model with analytical methods alone. Although it is feasible to reverse-engineer the actuator dynamics from hardware measurements, this requires the completed robotic system to be already available. To enable the inclusion of realistic actuator dynamics in robot models also during the design phase or for initial controller tuning, this work presents an alternative hands-on approach for actuator characterization. Based on actuator measurements taken independently of the overall system integration, a model expression for the actuator is derived. This can be added to the simulation of any robotic system. To showcase this concept, we present the workflow for a robotic leg with a Series Elastic Actuation chain. We create a simulation of the leg incorporating the derived actuator model and show its validity through comparison with analogous hardware. The observed motor and link dynamics of both cases show close correspondence without increasing the needed computation times with respect to a simulation without actuation. Thus, the proposed method offers a promising approach to include realistic actuator dynamics during the design and development process of robotic applications.**

I. INTRODUCTION

Legged robotic systems increasingly offer a valid alternative to wheeled rovers for exploration purposes. Especially in rough and uneven terrain, adaptable foot placement allows for more flexibility of legged robots in comparison to their wheeled counterparts [1]. Therefore, various legged robots are being developed with one of the most intensely researched type being quadrupeds, such as MIT's Mini Cheetah [2], EPFL's Cheetah Cub [3], ANYmal [4], Boston Dynamics' Spot robot [5] as well as Bert [6], [7] (see Fig. 1) developed by the German Aerospace Center (DLR). Due to size and weight restrictions, the mounting space for actuators is limited in these robots, such that the reachable forces of the used electric drives usually need to be increased with gear mechanisms. Additionally, many of the mentioned quadrupeds incorporate springs [2], [3], [4], [6], which allows to store and reuse energy during walking and make the systems more robust against impacts. Although these actuator improvements are useful, they simultaneously increase the complexity of the actuation chains leading to hard to predict dynamics. For controller development in lab scenarios, simplified actuator models often suffice [8]. However, especially during preparation of the robots for applications in real field scenarios, simulating a robot with realistic actuators and energy needs is crucial for a successful simulation-to-reality transfer. Deriving accurate models for the mentioned complex actuation chains solely by analytical means is almost impossible. To include all effects due to friction, gear transmissions, temperature and non-linearities, additional measurements of the motor need to be taken, i.e. speed step responses, to estimate parameters [9] or derive motor transfer functions [10]. Nevertheless, these approximations often lack accuracy, especially when both constant high speeds and high torques are required in the applications. Further, the estimations often do not hold when taken out of the test setup and applied to complete systems.

Fig. 1. The compliant quadruped Bert developed by the German Aerospace Center in a realistic field scenario.

To tackle this problem, Hwangbo et al. [11] recently provided a new approach with the ANYmal quadruped. Instead of analytically deriving the motor dynamics, a deep learning approach was used to train an *'actuator net'* in simulation based on recorded hardware data. In this way, motor dynamics and losses could be incorporated in the simulation and realistic controllers were developed that translated well from simulation to hardware. Although this approach is computationally efficient and can be recalculated for different high-level tasks or actuators, it requires that the complete robot, i.e., ANYmal, is already built and controllable. However, especially in the hardware development phase, the simulation of realistic actuation dynamics would be beneficial as it allows to optimize mechanics for the intended actuation and helps to design realistic controllers.

To enable the inclusion of realistic actuator dynamics in the development phase of a robot, this paper proposes a hands-on approach to characterize a desired actuation chain independent of the system it should be used in. Additionally,

This research has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (Grant Agreement No. 835284)

[1] Department of Informatics, Technical University Munich, Garching Germany an.schmidt@tum.de

[2]Institute of Robotics and Mechtronics, German Aerospace Center, Wessling, Germany

 DOI: 10.1109/AIM52237.2022.9863397

it presents the workflow to use the proposed method to create a successful simulation-to-reality transfer validated through hardware experiments. The proposed actuation characterization is based solely on independently taken measurements recorded in a dedicated test setup. Here, not only speed step responses are taken, but constant speeds and torques are applied leading to more precise and for applications relevant measurements. After recording, the actuator measurements are fitted to derive a model that captures the actuator dynamics, which can be incorporated in the simulation of any robot. It is here applied to a single robotic leg of the quadruped Bert (see Fig. 1) serving as simplified use case with a Series Elastic Actuation (SEA) chain. A simulation model of the robotic leg with the derived actuator dynamics is created and the method's validity is shown through comparison with an analogous hardware setup.

In the future, we hope to push towards a new actuator specification standard, with a comprehensive validation of the proposed method through a catalog of use cases and actuator types. By providing the model obtained through the suggested actuator characterization and the relevant inertial dimensions as part of an actuator's data sheet, mechatronic and robotic systems design could be vastly streamlined with accurate simulations, and component selection before the hardware prototype.

II. Theoretical Background

In order to better understand the need for the proposed hands-on approach to approximate motor dynamics experimentally, a short overview on common analytical methods to model electric actuators is given. For this, we focus on Permanent Magnet brushed DC Motors, which are in general one of the most commonly used drive types [12]. Thus, the functioning and modeling of such DC motors has been well investigated [13], [12]. The output torque τ on the shaft of an electric motor can be described with

$$\tau = \tau_{em} - \tau_{diss} - J\frac{d\omega}{dt} \ , \tag{1}$$

where τ_{em} is the magnetically generated torque, which is reduced by dissipative losses τ_{diss} and the motor's inertia. Dissipation τ_{diss} is due to friction and drag effects, while the inertia losses depend on the rotational speed ω and the rotor's rotational inertia J [12], [14]. In a static case, the magnetic torque τ_{em} is proportional to a torque constant k_T and the armature current I. Substituting terms to yield a relationship between the rotational speed of the motor n and the output torque τ leads to

$$n = n_0 - R\frac{\tau}{2\pi {k_T}^2} \ , \tag{2}$$

where n_0 is the ideal no-load speed, and R is the sum of resistances arising due to armature windings and conduction in the motor. For details on the derivation of this relationship refer to [12]. As n_0, R and k_T are all constant in the static case, (2) shows that the relationship between the rotational speed of a motor and its output torque is linear for unchanged *Pulse-width modulation* (PWM). The speed decreases with increasing torque until eventually a *stall torque* is reached. However, this relationship is idealized and does not consider effects such as temperature increase when the motor warms up. Furthermore, in application scenarios, the motor is often coupled to gears, e.g. planetary gears or harmonic drives, to enable higher torques for a given workspace constraint. The resulting couplings, friction and inertia effects add unknown components that are hard to predict. If elastic elements such as springs are added, additional vibration effects need to be considered [14]. Deriving a mathematical model for a complete actuation chain with analytical methods exclusively is impossible and often requires additional hardware tests to approximate friction terms [9]. Although simplified motor models might suffice for initial tests [8], for the development of effective controllers applicable to real robotic hardware, including accurate actuator dynamics in the robot model is crucial. For this purpose, the actuator behavior can be reverse-engineered when the robot is readily available as proposed by Hwangbo et al. [11]. However, we offer an alternative solution in which the drive dynamics are derived independently of the robotic hardware. This would allow them to be included in actuator specifications to help select a suitable actuator already during the system design phase, prior to hardware construction.

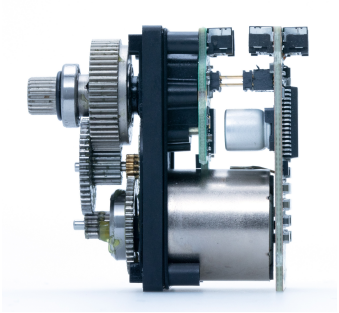

Fig. 2. Custom modified servo drive unit with the commercial stage spur gearbox, the brushed DC motor and customized electronics.

III. Actuator Characterization

To determine the dynamics of an actuator for robotic simulations without deriving models analytically or record data of a complete robot, we derive a model solely based on measurements of the actuator itself. The method is exemplary presented for an electric drive that is detailed in this section. To obtain the actuator measurements, a test setup is built and a measurement procedure is defined. From the recorded data, a model is derived that can be added to the simulation of any robot once the inertia of the actuator is estimated.

A. Servo Drive

The actuator used in this work was a *servo drive unit* designed for robotic applications. It consisted of two parts: 1) an electric motor with its power stage and 2) a gearbox (see Fig. 2). It was a self-modified version of the commercially

available Savöx SV-1270TG, which incorporates a coreless brushed motor and a four-stage spur gear gearbox made of titanium to transmit the power to the output shaft. Since the commercial version did not incorporate a telemetry data link, which is necessary for complex robotic control approaches, the internal electronics were removed. Instead, two self-designed PCBs were added. The PCBs incorporated an integrated power stage, an 8-bit microcontroller with 16 MHz clock rate and a contactless on-axis magnetic position sensor based on the Hall effect. The brushed motor was powered by a mounted H-Bridge with 20 kHz PWM frequency. The communication interface to the control host was implemented via isochronous USB 2.0.

B. Actuator Measurements

To characterize the dynamics of the servo unit, a test setup was built, in which systematic measurements of the actuator were recorded, as detailed in the following.

1) Setup: In the measurement setup (see Fig. 3), the servo unit was the *device under test* (DUT). It was coupled to an inline torque transducer (Magtrol TMHS 306), which was in turn connected to a hysteresis brake (Magtrol Compressed-air-cooled Hysteresis Brake AHB-6). The transducer had a nominal range of 5 N m, while the brake had a rated torque of 6 N m. To ensure proper alignment of all components, the setup was mounted on optical rails (OWIS SYS 90) guaranteeing the offset in axis alignment to be $<$0.1 mm and the angle mismatch to be $<$0.5 degrees. The parts were connected with high precision and zero backlash through torsionally stiff metal bellow couplings (R+W bellows coupling BK5-30) with a rated torque up to 30 N m. The setup was controlled via an EtherCAT field bus from Beckhoff Automation. As data acquisition module an EL3102 Ethernet terminal with an analogue voltage range of $\pm$ 10 V and 16-bit resolution was used. On the output side, the pulswidth current terminal EL2535 was added to control the hysteresis brake with up to 3.5 A excitation current. The torque controller for the brake was implemented in Simulink. The control loop ran with a frequency of 1 kHz interfaced by DLR's own middleware suite Links-and-Nodes [15].

2) Procedure: Three variables of interest were selected to be measured in the test setup to derive the model capturing the servo unit's dynamics for simulation: 1) the PWM signal V_{DUT}, 2) the torque load M_{DUT} and 3) the motor temperature T_{DUT}. For each variable, a data grid of measurement points was generated, in which the tested variable was varied while the other two were set constant. The PWM was varied from -0.9 to 0.9 (bridge supply voltage: 7.5 V) in steps of 0.05. The torque was applied in a range from 0 to 1200 N mm with 50 N mm step size and the temperature was tested in a range from 40 to 70 °C (step size: 15 °C). The order within the three data grids were pseudo-randomized to avoid systematical errors due to monotonously increasing data. All randomized grids were linked to one test protocol, where each data point appeared three times. To start a measurement, a data point from the protocol was loaded. First, the required temperature T_{DUT} for this point was cross checked with the current temperature of the motor and adjusted if needed. Once the desired temperature was reached, the system variables were zeroed to account for possible offsets. Next, the defined voltage V_{DUT} of the data point was set, letting the motor run in idle mode. The break side was then commanded to apply the required torque load M_{DUT} to the motor. When the torque was reached, all parameters were recorded for 1 s. If a required torque could not be reached with the set PWM, the measurement was automatically canceled and marked as invalid. After each completed recording, the next data point was loaded from the protocol until all points were measured.

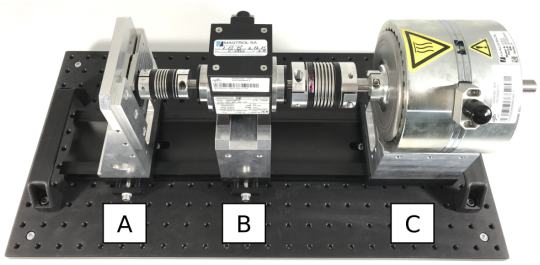

Fig. 3. Mechanical test setup for the servo unit as DUT mounted in the aluminum frame on the left side of the (A) mounting rail. It is connected by the (B) inline torque tansducer in the middle and (C) to the hysteresis brake on the right.

C. Measurement Analysis

Following the measurement of the servo unit, the recorded data was analyzed to derive an expression that captured the actuator dynamics. For this purpose, the measurement points were organized and the ones that were marked invalid due to stalling torque or malfunction were disregarded. Next, all recorded data points with the same PWM signal were inspected individually. As explained in Section II, according to (2) for a constant PWM a linear relationship was expected between rotational speed of the motor and output torque. This held true as can be seen in Fig. 4A. Outliers that differed more than twice the standard deviation were removed from each PWM data set leading to a total number of 4419 data points for the analysis. Due to the known linearity between the data points for identical PWM, it was decided to use a linear regression to interpolate the data. Consequently, the set PWM and the measured rotational speed n in rpm were used as input for the regression model to predict the output torque of the actuator τ leading to

$$\tau = 1804.38\ PWM - 16.96\ n - 139.78\ . \qquad (3)$$

Despite the theoretically linear relationship, the linear regression prediction did not lead to an ideal match for all actually measured output torques (see Fig. 4B). Specifically, for high PWM and low speeds, differences could be seen. However, in the range of PWM that was actually applied in the robotic example use case (below 800 N mm), the regression showed overall correspondence to the data points. Obviously, a better

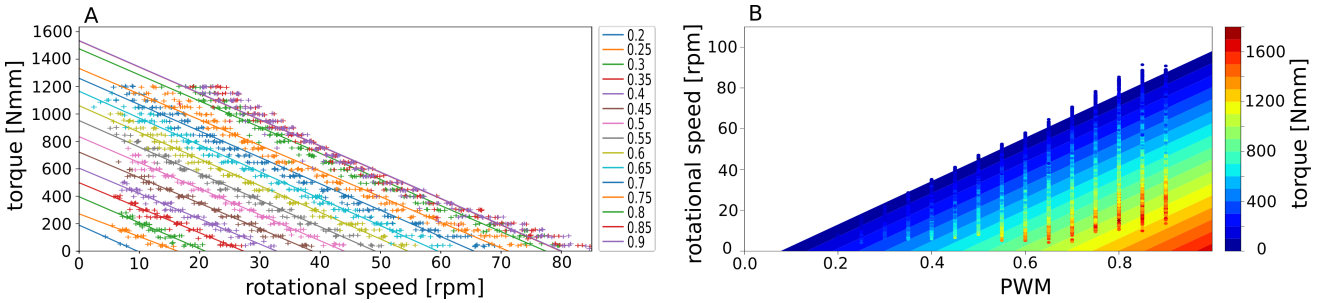

Fig. 4. (A) Plotted data points measured in the motor test setup and grouped by color for the corresponding PWM applied (see legend). For each set a linear relationship between motor torque and rotational speed can be fitted. (B) Linear regression plot to predict the torque by relating the PWM measured during the motor test to the rotational speed. The colored points in the foreground indicate the actually measured torques in the test setup.

fit could be determined with more complex fitting methods or learning algorithms, but this work's focus was to show the general applicability of the proposed concept. Thus, the approximation by linear regression was deemed sufficient, but shall be improved in future work.

D. Inertia Approximation

To achieve realistic actuator dynamics in a robotic simulation, not only the above torque relation was needed, but also an approximations of the actuator's inertia, as seen in (1). These inertial properties were practically deduced: The four stages of the servo unit were dissembled to be weighed and measured. Using the approximation of a disk, the inertia of each stage was obtained. Knowing the gear ratio between each stage determined by the teeth count and diameter, the individual stage inertias could be summed up to find the overall inertia of the rotating mass. The total gearbox ratio resulted to 287.74:1 and the rotating mass was 14.3 g. The overall inertia of the complete servo unit was $0.009\,579\,\mathrm{kg\,m^2}$. Detailed measurements are stated in Tab. I.

IV. Robotic Use Case

To showcase the workflow of implementing a derived actuator model in a robotic simulation and prove its validity, the use case of a single robotic leg was chosen. It incorporated two SEAs driven by the above presented servo unit. This was deemed a realistic application example as elastic elements are progressively added in the joints of robotic legs [2], [3], [6], [4] while requiring high forces within a small workspace. This usually leads to complex actuation chains that are impossible to derive analytically, but are needed for a successful simulation-to-reality transfer. Especially when the hardware is not yet built or controllable, our proposed method to derive and add actuator dynamics will be beneficial for such systems. Thus, this section presents the mechanics of the chosen robotic leg and explains the control and incorporation of the actuator in the simulation. The simulation results of the leg are then compared to data that was recorded with an analogous hardware setup of the leg.

TABLE I

Measured values of all components of the dissembled servo unit to identify the overall inertia of the rotating mass.

	teeth count	diameter [mm]	gear ratio	inertia [g mm^2]
rotor				114.69
stage 1	12 59	3.46 15.31	4.92	0.31 12.86
stage 2	11 48	4.16 15.00	4.36	0.35 41.11
connector				20.02
stage 3	12 48	4.46 14.75	4.00	1.07 8.94
stage 4	17 57	5.86 17.35	3.35	6.42 473.74
shaft				7.42
total			287.74	9578657.94

A. Mechanical Model

As test platform to incorporate the derived actuator dynamics, a single articulated robotic leg of the quadruped Bert (see Fig. 1) was selected. Since in a complete quadruped the trunk constrains the legs, the body of the robotic leg was here assumed to be a floating base. It could freely translate, but was fixed in all rotational axes. The leg had two rotational joints (see Fig. 5) denoted as *hip* and *knee*. Each joint was driven by the servo unit detailed above. Between each servo unit and the link, a torsional spring with a stiffness of $k = 2\,\mathrm{N\,m\,rad^{-1}}$ was added, forming a complete SEA chain. While the hip SEA was directly connected to the upper link, the SEA for the knee was fixed in the base and linked to the lower link via belt drives. The joint angles were denoted q_1 and q_2 for the hip and knee, respectively, and directions were defined as shown in Fig. 5. The initial position of the joints were set to be $q_0 = [-0.61, 0.61]^T$ rad. The corresponding actuator angles were denoted as θ_1 and θ_2. As hardware, the complete system had a mass of 0.562 kg. However, a boom had to be attached to the leg's base to realize the floating base assumption. To account for the added inertia of the boom in simulation, the simulated mass was increased by 4.5 %, which was the relative inertia of the boom in comparison to the hopper. The total mass in simulation thus amounted to 0.587 kg with 0.059 kg and 0.038 kg being the masses of the upper and the lower link, respectively. For details on the mechanical model please refer to [6].

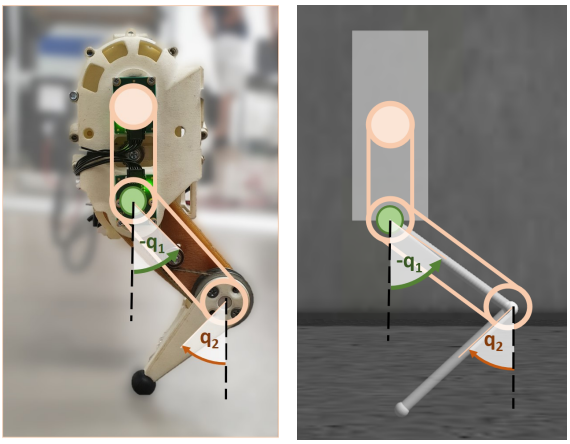

Fig. 5. Hardware (left) and simulation model (right) of the robotic hopper. It consists of a floating base with two links connected by compliant joints. The hip (q_1) and knee joint (q_2) are individually driven by Series Elastic Actuators without affecting each other's motion. The beige circle indicates the knee motor location, the green one the hip motor. Angles are defined counter clockwise relative to vertically extended links.

B. Controller

To compare the resulting dynamics of the hardware and the simulation with added actuator model, the robotic leg was commanded to hop forward with a high-level bang-bang controller. The functioning and implementation of this controller are detailed in [16], [17], [18] and here only set constants are mentioned: Whenever a defined torque threshold ($\varepsilon = 0.3$) was crossed, a 1D-control signal $\theta_z = 0.23$ was triggered. This signal was then transformed to position commands $\theta_{1,2}$ for each servo by a weighting vector ($w = [-0.5, 1]^T$) resulting in a hopping motion. The commanded position signal was converted to a PWM signal by means of a low-level PD-controller, which generated the output torque on the motor shaft to deflect the attached springs and move the links to position $q_{1,2}$. The employed control architecture was identical for the simulation and the hardware setup. As the inherent viscous friction damping in the joint was found to be sufficient, the D-portion of the PD-controller was here set to zero. For the simulation, the proportional gain was set to $K_P = 75$. This value had been tuned for a previously unused motor like the one measured in the test setup. However, the motor in the hardware robot had already been used for extended time, which lead to wear, and reduced friction in the motor gears. Thus, the K_P-value had to be lowered to 55 in the experiment setup. With exception of K_P, all control variables were set identical in the simulation and hardware setup. In the hardware, the high-level control was implemented in Simulink, while for the simulation it was run from a python script. The communication between the controller and the hardware and simulation, respectively, was again implemented via Links-and-Nodes, allowing realtime communication on a bandwidth of 1 kHz without delays.

C. Actuator Simulation

For the simulation of the robotic leg Gazebo 10 was used. To physically represent the servo units, a cylindrical body was added for each leg joint. As in the real hardware, the servo for the knee joint was positioned in the leg's base and connected with a (frictionless) belt drive (gearbox joint), while the hip servo unit was directly positioned in the hip joint. Both servo links were added with the rotational mass of 0.015 kg and the inertia value of $0.010\,\mathrm{kg\,m^2}$ as determined in Section III-D. To account for the gearbox friction, a damping term was added between the robot base and each servo unit link. This damping coefficient was estimated to be $0.35\,\mathrm{N\,m\,s\,rad^{-1}}$ based on prior experience and simulation considerations of a gearbox presented in [19]. The friction between the hopper model and the ground was defined as $\mu = 1.0$. According to the hardware setup, a stiffness of $k = 2\,\mathrm{N\,m\,rad^{-1}}$ was added to the revolute joints connecting each motor link with the upper and lower leg link, respectively. A conventionally used value of 1 % of the spring stiffness was additionally added as damping term ($c = 0.0219\,\mathrm{N\,m\,s\,rad^{-1}}$) [20]. To add the derived model from the preceding measurements and analysis of the servo unit detailed in Sec. III, the torque-relation (3) was implemented in a C++ Gazebo plugin using otherwise the identical control scheme of the hardware setup. In this, the required PWM was calculated with the PD-controller based on the commanded position of the bang-bang control and the measured current position returned by the simulation. Based on the PWM and the measured motor speed, the torque for each motor joint was predicted based on (3) in each time step.

To emphasize the importance of the actuator in simulation, the robot was also modeled without actuator dynamics. In this case, the springs connected each leg link directly to the base. The damping term of 1 % of the stiffness remained. The motor position θ was applied to the joint by relaying the torque resulting from desired position and spring stiffness.

D. Validation Experiment

For this initial concept validation, the performance of the robotic leg was solely compared in the simple scenario of hopping forward in a plane using the control architecture and parameters described in Section IV-B. For both, the simulations and hardware setup, a time period of 10 s jumping was recorded and analyzed. The focus for the comparison lied on the progression of the motor and link positions and velocities as well as the jumping height and frequency. The simulations were run with the Gazebo default time step of 0.001 s using the default ode-solver. The simulations were additionally run with maximum possible update frequencies to further investigate the capabilities of the simulation model. Moreover, the simulation was repeated with four identical leg models as a form of a simplified quadruped to scope the validity of the results for more complex models. All simulations were carried out on personal computers with quad-core CPUs of 1.8 or 2.4 GHz. The sensor measurements of the robot in hardware as well as in the simulations were updated with a frequency of 1 kHz. For details on the hopper implementation and hardware setup please refer to [18].

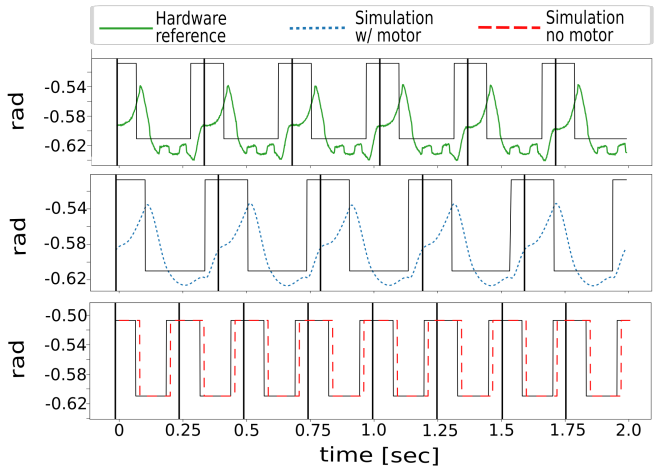

Fig. 6. Comparison of the motor hip position of hardware measurements set as reference (green), the simulation including motor dynamics (blue) and the the simulation without a motor model (red) each displayed for a time period of 2 s. The vertical black lines indicate the beginning of a new period of the jumping motion initialized at the point of maximum flexion of the hopping leg in the stance phase.

E. Results

The bang-bang controller could excite a regular hopping motion in both simulations and the hardware setup using identical control parameter values (see video attachment). However, the hopping frequency in each case differed slightly. The hardware hopper set as reference jumped with 2.9 Hz. The simulation with added actuation dynamics reached a similar jumping frequency of 2.5 Hz, while the model without actuation differed clearly with 4 Hz (see Fig. 6). Accordingly, the jumping height differed as well with the hardware robot reaching 10 cm, while in the simulation with actuation the robot jumped 12 cm high. Corresponding to the higher frequency, the jump height in the simulation without actuator was much lower with 4 cm.

To better compare the progression of the position and velocity curves for the motor and link side in the different cases, the data was normalized by the period length in each setup. The start of each period was defined at the moment of maximum flexion of the robotic leg, i.e., when reaching the lowest point during the stance phase. The overlayed normalized plots are exemplary shown for the hip joint in Fig. 7. The progression of the motor position curve simulated with the derived actuator expression from (3) overall matched the behavior of the real hardware (Fig. 7A). The plateau that was reached in the hardware due to a torque overload during the flexion could also be found with the simulated actuation. Likewise, the velocity profile obtained with this simulation captured the characteristics of the hardware. However, the curve of the real motor position showed some additional step-like deflections during the flight phase with corresponding peaks in the velocity. These deflections and peaks were absent with the simulated motor model. As expected, the recorded motor positions in the idealized simulation differed greatly as the commanded step function was directly applied

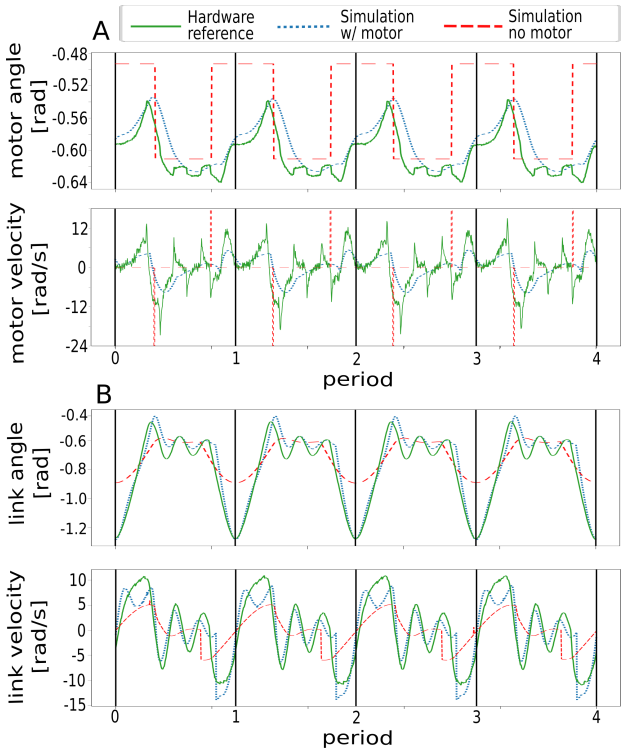

Fig. 7. Overlayed plots of data recorded for the hip joint of the robotic hopper with the hardware setup (green), the motor simulation (blue) and the simulation without the motor (red). The x-axis is normalized by the period in each scenario. The black vertical lines indicate the beginning of each period measured from the point of maximum flexion during stance phase. (A) The progression of the hip motor position and velocity between motor simulation and hardware match well. (B) The motor simulation is also capable to capture the position and velocity profile on the link side.

as joint torque. This led to almost instantaneous position changes and in turn to unrealistic high velocity spikes.

The progression of the position and velocity on the link side also corresponded well between the motor simulation and the hardware reference (see Fig. 7B). In both cases, the same maximum positions during the stance phase were reached and similar free swing motions show during the flight phase. The simulation without the motor could also capture the rough link dynamics, but with identical control parameter it differed clearly from the other two curves. The control signal was triggered earlier leading to a smaller flexion of the leg joints and in turn explains the higher jumping frequency. It can also be observed that the leg did not swing realistically in the flight phase despite applying the same spring stiffness.

Independent whether the motor dynamics were added to the robot model or not, the simulation was capable to run at realtime with an update frequency at 1 kHz using the Gazebo default parameters. The maximum possible update frequency was confirmed to be more than 3 kHz with and without the added motor model. When simulating four legs (eight joints), each with added motor dynamics, as a simplified form of a quadruped, the simulation was still capable to run at almost realtime (0.9).

V. Discussion

This paper presented a hands-on approach to determine the dynamics of complex actuation chains for robotic simulations without the need to derive expressions analytically or record data of a complete robot. Additionally, the workflow was presented to add a derived actuator model to a robotic simulation, i.e., of a robotic leg, and showed the validity of the approach through the comparison of the simulation with corresponding hardware.

A. Simulation-to-Reality Transfer

The validation experiment with the robotic leg showed clearly that the proposed concept to characterize the actuator dynamics independent of a complete robotic system led to a simulation model that captured all main characteristics of the corresponding hardware (see Fig. 6 and Fig. 7). Additionally, an identical control architecture and parameter values could be used for both cases. Solely the step-like deflections seen in the hardware during the flight phase with corresponding peaks in the velocity profile were absent in the simulation. It is likely that these features in the hardware were caused by minor backlash in the motor and consequently cannot appear in the simulation. Nevertheless, the overall motor behavior and even more importantly the link dynamics of the system could be captured. As minimal backlash will almost always be present in real systems and often increases over time due to wear, it is hard to account for it in simulation. However, the results suggest that this might also not be necessary. More importantly, the simulation showed the obvious overload of the hip joint, which can be valuable information especially in the design phase of a robot.
Although, for jumping frequency and height slight differences could be seen between the simulation with the derived actuation model and the hardware, it is likely that these are related to the ideal contact properties and high friction coefficient in the simulation rather than the actuator dynamics. Despite adding a rubber foot to the real robot to increase ground contact friction, natural occurring circumstances such as dust or room temperature influenced the actually occurring contact friction in the hardware setup. Thus, the higher and more consistent ground friction in the simulation led to slightly higher jumps and in turn to the extended jumping period. The much shorter jumping frequency in the simulation without the actuation model was likely due to a mismatch with the set control parameter values, which were tuned for the system with actuation. However, the control parameters were deliberately not tuned for each case since identical parameter values are important for an easy simulation-to-reality transfer.

B. Advantages and Limitations

The proposed concept for the experimental characterization of the actuation dynamics shows promising results that might make it advantageous over conventional analytical methods. However, the limitations of the presented research must also be considered.

In contrast to the method applied by Hwangbo et al. [11], where the completely built robot had to be recorded in application settings to obtain the actuator net, our approach solely requires the actuation hardware to be available. This makes our approach particularly useful for the design phase of new robots as the actuator dynamics can be individually assessed and then be added to any robotic simulation. Investigating the combination of the desired actuator model in the complete system in simulation prior to built can help to detect limitations of the intended combination such as the torque overload in the hip joint of the robotic leg example (see Fig. 7). This can help to reevaluate choices and develop overall better systems. Another advantage was that due to the realistic model, the control architecture and parameter values could be implemented identically in hardware and simulation, which is particularly important for a successful simulation-to-reality transfer without the need to re-tune developed controllers. Moreover, the obtained actuator expression did not require more computation time in comparison to the simulation without added actuation dynamics and was even capable to run significantly above the 1 kHz update frequency required for most controller designs. This makes the proposed approach also a valid option to develop controllers based on machine learning where at least realtime capabilities of the simulation are required.

Although, the initial validation on the presented use case shows the applicability of the proposed concept, further validation should be pursued to justify it as valid alternative to more established methods. This could include the comparison with a naive analytical model of the used servo unit as well as the validation against more complex input commands instead of the applied bang-bang control. In a next step, the method should also be tested on different actuators and use cases. In addition, only linear regression was used in this work to fit the motor measurements, which did not result in an ideal match (see Fig. 4B). The choice for this approximation was based on the known linear relationship between motor speed and torque shown in (2), but more sophisticated methods will lead to a better fit of the motor data and likely improve the simulation further. Nevertheless, the applied fitting method captured well the overall actuation dynamics and was thus deemed sufficient to showcase the workflow of the concept. Moreover, a better method to identify the damping term between the robot base and the motor link to account for the gear friction is needed as this term is essential for a realistic simulation. In the presented case, the term was mainly estimated based on experience since accurate calculations of gear friction is a very complex task on itself [9]. It remains to be investigated what influence the exact value of this term has in the simulation-to-reality transfer, as it is likely that an experience-based approximation will suffice. Another obvious drawback of the proposed concept is that experimental measurements of

the actuator are required. This, and especially the inertia approximation, is evidently a tedious and time-consuming task that is only possible if the desired actuator is available as hardware. However, for common analytical approaches the measurement effort is not less, especially when using a complex actuator with integrated gearbox unit as presented here [9], [13]. With our approach only inertia and mass measurements of the gear stages and rotor were needed. Otherwise, the complete actuation chain, from electrical DC-Link to output shaft, could be regarded as a black-box in the test setup. With further validations for more use cases and different motor types as well as investigations of better data fitting methods, such actuation measurements could contribute to form a new standard for actuator specifications as part of the manufacturer's data sheet. This would allow engineers to easily add realistic actuator dynamics to robotic simulations during the design phase to help them select the most suitable actuator for a desired application.

VI. Conclusion

This research has proposed a practical approach to characterize actuator dynamics for robot simulations without requiring analytical derivation or data recordings of the complete robot. Our method shows promising results, as validated for an example use case of a jumping robot leg with a complex SEA drive chain. The simulation of the leg using the derived actuator model was able to represent all the main features of the actuator and link dynamics seen in the corresponding hardware. In addition, we have shown that identical control architecture and parameter values can be used for simulation and hardware, facilitating easy control development and successful simulation-to-reality transfer. While an actuator net trained with a complete robot could lead to marginally more accurate simulations, our approach offers the possibility to study the performance of robot hardware in the design phase prior to hardware implementation. As the characterization measurements can be taken independently of the complete system, our vision is to promote this to become a part of the standard actuator specifications. This would vastly streamline and improve the actuator selection in the design flow of a robot or mechatronic system, as a design could be analyzed for its performance and limitations prior to hardware development.

ACKNOWLEDGMENT

The authors would like to thank Daniel Seidel, Robert Burger and Florian Schmidt for the support on the hardware and software as well as Dr. Aaron Pereira, Dr. Neal Y. Lii and Dr. Philipp Stratmann for the constructive criticism on the written form.

References

[1] C. D. Bellicoso, M. Bjelonic, L. Wellhausen, K. Holtmann, F. Günther, M. Tranzatto, P. Fankhauser, and M. Hutter, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[2] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2019, pp. 6295–6301.

[3] P. Eckert, A. E. Schmerbauch, T. Horvat, K. Söhnel, M. S. Fischer, H. Witte, and A. J. Ijspeert, "Towards rich motion skills with the lightweight quadruped robot serval-a design, control and experimental study," in *Int. Conf. on Simulation of Adaptive Behavior*. Springer, 2018, pp. 41–55.

[4] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch *et al.*, "Anymal-toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[5] (2021, Februrary) Spotmini autonomous navigation. [Online]. Available: https://www.youtube.com/watch?v=wlkCQXHEgjA

[6] D. Lakatos, K. Ploeger, F. Loeffl, D. Seidel, F. Schmidt, T. Gumpert, F. John, T. Bertram, and A. Albu-Schäffer, "Dynamic locomotion gaits of a compliantly actuated quadruped with slip-like articulated legs embodied in the mechanical design," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3908–3915, 2018.

[7] D. Seidel, M. Hermann, T. Gumpert, F. C. Loeffl, and A. Albu-Schäffer, "Using elastically actuated legged robots in rough terrain: Experiments with dlr quadruped bert," in *2020 IEEE Aerospace Conference*. IEEE, 2020, pp. 1–8.

[8] T. Buschmann, S. Lohmeier, H. Ulbrich, and F. Pfeiffer, "Dynamics simulation for a biped robot: Modeling and experimental verification," in *2006 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2006, pp. 2673–2678.

[9] S. Wolf and M. Iskandar, "Extending a dynamic friction model with nonlinear viscous and thermal dependency for a motor and harmonic drive gear," in *2018 IEEE Int. Conf. on Robotics and Automation*. IEEE, 2018, pp. 783–790.

[10] B. Arifin, A. A. Nugroho, B. Suprapto, S. A. D. Prasetyowati, and Z. Nawawi, "Review of method for system identification on motors," in *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*. IEEE, 2021, pp. 257–262.

[11] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[12] H.-D. Stölting, E. Kallenbach, and W. Amrhein, *Handbook of fractional-horsepower drives*. Springer, 2008.

[13] B. Bilgin, J. Liang, M. V. Terzic, J. Dong, R. Rodriguez, E. Trickett, and A. Emadi, "Modeling and analysis of electric motors: state-of-the-art review," *IEEE Transactions on Transportation Electrification*, vol. 5, no. 3, pp. 602–617, 2019.

[14] Q. Zhang and G. Liu, "Precise control of elastic joint robot using an interconnection and damping assignment passivity-based approach," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 6, pp. 2728–2736, 2016.

[15] C. Ott, M. A. Roa, F. Schmidt, W. Friedl, J. Englsberger, R. Burger, A. Werner, A. Dietrich, D. Leidner, B. Henze *et al.*, "Mechanisms and design of dlr humanoid robots."

[16] D. Lakatos, M. Görner, F. Petit, A. Dietrich, and A. Albu-Schäffer, "A modally adaptive control for multi-contact cyclic motions in compliantly actuated robotic systems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 5388–5395.

[17] P. Stratmann, D. Lakatos, M. C. Özparpucu, and A. Albu-Schäffer, "Legged elastic multibody systems: adjusting limit cycles to close-to-optimal energy efficiency," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 436–443, 2016.

[18] A. Schmidt, B. Feldotto, T. Gumpert, D. Seidel, A. Albu-Schäffer, and P. Stratmann, "Adapting highly-dynamic compliant movements to changing environments: A benchmark comparison of reflex-vs. cpg-based control strategies," *Frontiers in Neurorobotics*, vol. 15, 2021.

[19] W. Bartelmus, "Transformation of gear inter-teeth forces into acceleration and velocity," *Int. Journal of Rotating Machinery*, vol. 5, no. 3, pp. 203–218, 1999.

[20] D. Lakatos, D. Seidel, W. Friedl, and A. Albu-Schäffer, "Targeted jumping of compliantly actuated hoppers based on discrete planning and switching control," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2015, pp. 5802–5808.