



UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR ROBOTIK
UND KOGNITIVE SYSTEME

Learning of Inverse Dynamics for Proprioceptive Force Estimation during Irregular Fine-Scale Robot Motion

*Lernen inverser dynamischer Modelle zur propriozeptiven Kraftschätzung
während unregelmäßiger feinskaliger Roboterbewegungen*

Masterarbeit

verfasst am

Institut für Robotik und Kognitive Systeme

im Rahmen des Studiengangs

Medizinische Ingenieurwissenschaft

der Universität zu Lübeck

vorgelegt von

Tolga-Can Çallar

ausgegeben und betreut von

Univ.-Prof. Dr. Elmar Rückert

mit Unterstützung von

Dipl.-Inf. Sven Böttger

Lübeck, den 21. September 2021

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Tolga-Can Çallar

Zusammenfassung

Durch die andauernde Weiterentwicklung kollaborativer Robotertypen ist der Einsatz von Robotern nicht mehr beschränkt auf rein industrielle Anwendungen und dringt verstärkt in Anwendungsbereiche mit einem inhärent hohen Maß an physischer Mensch-Roboter-Interaktion vor. Dies ist gleichbedeutend mit einem Paradigmenwechsel hinsichtlich der Roboterregelungstechnik in Richtung hochgradiger Kraftregelungssensitivität, um funktionale und sicherheitstechnische Anforderungen an die Regelung von physischen Kontaktkräften zu erfüllen. Eine grundlegende Herausforderung hierbei ist die Schätzung auftretender Kontaktkräfte. Mittels der in modernen Robotern integrierten Gelenkwinkel- und Drehmomentsensorik, die eine größere mechanische Perzeptibilität als externe Kraftsensorik ermöglicht, wurde in dieser Arbeit ein propriozeptiver Ansatz zur Entwicklung eines inversen dynamischen Modells gewählt, um dynamische Störungen während feinskaliger Bewegungen abzuschätzen und auf äußere Kraftwirkungen zurückzuführen. Hierfür wurden verschiedene Modellierungstechniken basierend auf dynamischer Parameteridentifikation, künstlichen neuronalen Netzwerken sowie einer neuartigen hybriden Modellarchitektur mit Gelenkdrehwinkelkodierung implementiert und auf dem KUKA LBR iiwa 14 ausgewertet. Die Evaluationsergebnisse zeigen eine signifikante Verbesserungen der Drehmoment- und Kraftschätzungsgenauigkeit mithilfe einer hybriden Architektur gegenüber konventionellen dynamischen Starrkörpermodellen oder eigenständigen neuronalen Netzwerken.

Abstract

The applicability of robotic automation has transcended the industrial domain through the emergence of collaborative robotics and is increasingly entering the realm of applications with high levels of physical human-robot interactions. This is concomitant with a paradigm shift towards higher force control sensitivity to accomplish functional and safety requirements concerning the regulation of contact forces between robots and humans. A fundamental challenge in this regard is the observability and estimation of interaction forces. Utilizing the availability of joint position and torque sensors in recent collaborative robot models that yield a larger perceptive field for interaction forces than local force sensors, a proprioceptive approach is taken in this thesis to develop inverse dynamic models to estimate dynamic disturbances and determine external interaction forces during fine-scale motion. A series of state-of-the-art techniques are implemented and evaluated on the KUKA LBR iiwa 14, including dynamic parameter identification, neural-network based single-step, and time-series models, and a novel hybrid architecture combining a rigid body dynamics model with downstream neural networks and joint rotational displacement encodings. The results indicate that significant improvements in torque and force estimation accuracy can be obtained by the proposed method when compared with conventional rigid body dynamics models or neural networks alone.

Acknowledgements

Within these short lines, I would like to thank everyone who has given me academic and personal support during the writing this thesis.

I express my sincere gratitude to Prof. Elmar Rückert for his supervision and willingness to give me the freedom to pursue research on this interesting topic, and to Sven Böttger for his valuable advice and guidance, without both of whom this work would not have been possible. For the fruitful discussions and useful feedback, I want to thank Jonas Osburg, Jannis Hagenah, and Daniel Wulff. Last but not least, my wholehearted appreciation goes to my friends and family for their encouragement and company, and especially to my parents for their unconditional love and continuous support.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Contributions of this Thesis	5
1.3	Related Work	8
1.4	Structure of this Thesis	11
2	Experimental Setup	12
2.1	Robotic Evaluation Platform	12
2.2	Control System	17
3	Force Estimation through Dynamic Identification	19
3.1	Dynamics of Serial Robots	19
3.2	Dynamic Modeling as a Statistical Regression Task	21
3.3	Parametric Dynamic Model Identification	24
3.4	Artificial Neural Networks for Dynamic Model Identification	30
4	Hybrid Models for Torque Estimation	41
4.1	Parametric Backbone Model	46
4.2	Downstream Neural Networks	56
4.3	Algorithmic Training Data Generation	59
4.4	Traning Scheme	65
5	Model Performance Evaluation	68
5.1	Architectural Performance Comparison	69
6	Conclusion	76
	Bibliography	80
A	Joint-Wise Torque Predictions	89

1

Introduction

Since the advent of multi-articulated anthropomorphic robots, a conceptual and increasingly practical candidate platform technology has been introduced with the potential to be deployed in substitution for manual labor tasks which were previously deemed to be unsuitable for automation, e.g. due to confined workspaces or complex movement sequences associated with a particular task.

The application of robotics for this purpose has been successful especially for the automation of tasks involving repetitive motion patterns under well-controlled environmental conditions where an accurate dynamic and kinematic modeling of the robot and the environment can be feasibly carried out. Under the additional condition of minimal variation of the operational parameters allowing for high predictability, a robotic system may then be able to well surpass natural human limitations regarding dexterity, precision, endurance, or perception by courtesy of its task-specifically designed structural advantages to effectively perform at super-human performance levels. This is the case for applications without or with at least only minimally constrained direct physical contact, i.e. through inherently compliant or freely movable manipulation objects, and has been demonstrated for various industrial use cases covering e.g. pick and place, palletization, and welding applications where a pure motion control strategy is sufficient.

As its name suggests, motion control is a control scheme reduced to the mere execution of a pre-planned motion path governed by a desired dynamic trajectory and the control objective can be thought of as the mere compensation of arising path or trajectory deviations. This approach appears to be a sufficient control strategy given the preconditions of highly accurate model knowledge of the interaction object and precise robot control, allowing for a control behavior that can be agnostic of the actual physical interaction at task execution. In contact-based applications that require constrained direct interactions, however, inevitable modeling and motion execution inaccuracies often lead to a large discrepancy between the planned motion path and actual motion of the robot relative to the environment, resulting in inadequate control of interaction with an object, i.e. by the inability to apply task- and object-specific contact forces.

Thus, the motion control paradigm has to be augmented with the additional consideration of occurring mechanical forces between the robot, the environment and objects to be interacted with, i.e. force control. This allows for more comprehensive utilization of robotics in applications involving physical interactions, as has been established for industrial tasks,

e.g. with robotic grinding, polishing and deburring (Hägele et al., 2016).

However, either due to the absence of an appropriate object of comparison, an adaptive bias because of our exposure to its inherently ubiquitous presence, or, when assuming a more impressionable perspective, because of its truly and objectively impressive characteristics from which results a great aptitude to interact with the surrounding physical environment – human upper extremity motion is still generally considered a benchmark any technical solution has to measure up against in the realm of complex real-world manipulation and physical interaction tasks with strict interaction force constraints under the influence of perturbations and uncertainty effects. A general setting within which task complexity, force sensitivity requirements, spatial constriction, and uncertainty concurrently culminate is faced when physical interactions between robots and humans occur, i.e. when the operational space is shared between humans and robots. A multitude of elementary challenges consequently deriving from this has rendered the domain comparatively impermeable to the wider deployment of robotic automation solutions. The causes for this mainly revolve around devising robotic motion and force control schemes that allow for a task execution sufficiently fulfilling robustness, efficiency, effectiveness, and safety requirements.

By classifying robotic applications according to their respective degree of conceptual inherence of physical human-robot interaction, one can identify the degree to which operational scenarios exhibit the aforementioned properties and challenges complicating the use of robots for these tasks. A useful evaluative approach in this regard is to assign any considered task a two-dimensional notional value consisting of the level of physical proximity between the human and a robot in relation to the level of autonomy enjoined on the robot during task execution (Haddadin and Croft, 2016). Those tasks that cluster together according to this abstract scoring are hence expected to feature comparable requirements to force control for physical human-robot interaction. From the conjecture that an increase in average interaction proximity and robot autonomy respectively implies a likewise increase in the possibility of direct human-robot contact and uncertainty about any projected course of physical interaction, it can be deduced that close human-robot proximity and robot autonomy call for higher sophistication in robot motion control with incomparably more control strategy objectives. Aside from physical interaction proximity and autonomy, also the task-specific mode of motion execution, i.e. its scale, dynamic properties and characteristic spatial patterns, as well as technical properties of actuation and sensing hardware elements have to be taken into consideration for successful force control, as all of these, in combination with the aforementioned factors, dictate the degree to which interaction forces can be observed and controlled by a robotic system.

1.1 Motivation

While in a classically collaborative setup the main concern in force control is to reliably detect presumably sporadically occurring unsafe physical contacts and quickly respond with a safe reactive motion, the higher continuity of physical interaction in cooperative or similar close contact applications poses a challenge for the successful introduction of many potential human-robot interaction applications. On that note, this subsequent section is intended to

describe a subset of highly contact-based physical human-robot interaction applications requiring critical force control.

One application field that typically involves highly complex and force-sensitive physical interaction is to be found in the healthcare sector. As in the clinical field, virtually by definition, a large proportion of procedures are tactile and carried out in contact with the body of a patient, it becomes immediately apparent that one encounters a multitude of interconnected and volatile dynamic mechanical effects between the respective region of interest of the patient’s anatomy and e.g. a manipulated medical instrument. Because these applications are conventionally accepted to rely on medical personnel to utilize expert knowledge and sensorimotor skill to be executed, the potential of improvement for medical procedures through conceptual change by integration of robotic systems within medical tasks may be overlooked. The transfer of different aspects of the task execution from a human to a robotic system or their employment in augmentation of healthcare professionals may allow for the compensation of detrimental factors of pronounced user-dependency in healthcare tasks, e.g. personal variability of skill and natural human limitations regarding perception, cognition, physical strength, manipulative accuracy, and dexterity (Taylor, 1997). The concurrency of natural human limitations with regards to the tolerable workload and high demand of physical effort due to repetitive and strenuous motions for extended durations, as well as the scarcity of available trained personnel in healthcare further motivate the replication or assistance of manual labor through robots as has been analogously established for industrial purposes. The potential types of interaction with a patient to be performed by a robot in medical applications can be summarized under the terms of *tactile*, *kinesthetic* or *force-sensitive* interactions (Althoefer et al., 2010), where the functional subgroups along with their respective application examples range from explorative motion, e.g. palpational examination, sonography, over manipulative motion, e.g. tool-use during surgery and musculoskeletal manipulation, to responsive motion guidance, as seen e.g. in movement assistance for rehabilitation among others (Cutkosky and Provancher, 2016; Dario et al., 1996; Chang and Kim, 2013).

Despite a vast amount of scientific endeavors along with a steady increase in commercialized robotic applications aiming at force-sensitive human-robot interactions, a major factor impeding the widespread implementation of robotics in healthcare applications is the low tolerance margins concerning force-sensitivity, e.g. with regards to handling forces and torques occurring on the patient’s body during a procedure due to safety considerations as well as the adequate control of desired contact forces acting on the patient (Speich and Rosen, 2004). Therefore a fundamental area of research for the deployment of robotics in medical human interaction applications is the development of robust and highly force-sensitive control schemes to apply task-specific levels of contact forces on the target anatomy while at the same time regulating any occurring contact force limits to maintain safety standards, i.e. in the cases of target force exceedance or unexpected collisions between the robot and patient or environment.

The outlined basic requirements and challenges in potential human-robot interaction applications in the medical and healthcare sector can be illustrated based on the application example of robotic sonography which constitutes the motivational origin of this thesis. The procedural steps of conventional medical ultrasound manually carried out by healthcare professionals on patients are commonly known to incorporate a coarse initial positioning and

orientation of an acoustic transducer at task-specific areas of the patient’s body, i.e. acoustic windows, that are expected to allow for a suitable visualization of internal anatomical structures. This is then generally followed by an exploratory phase characterized by repositioning and reorientation of the transducer with incrementally smaller steps until the specific target structure is satisfactorily visualized. Meanwhile, the contact between the transducer and the patient’s body is constantly maintained as to not interrupt the image acquisition. Additionally, it is important to note that the applied contact force between the transducer and the patient is likewise manually regulated up and downwards by the healthcare professional to visualize deeper located or obstructed internal structures and not inflict discomfort or harm on the patient respectively. From a sensorimotoric perspective, for the physician this process amounts to a complex tactile manipulation of the transducer in contact with the patient involving manual motions on at a greatly variable scale which is directed by multiple sensory input streams and clinical objectives (Nicholls, Sweet, and Hyett, 2014). As such, here again, the diagnostic success is conceivably dependent on personal factors, e.g. the skill, dexterity, cognition, and even mere availability of the performing healthcare professional rendering the integration of robotic agents within this imaging modality highly desirable. Thus, a considerable amount of research focusing on different aspects of robotic automation of medical ultrasound has been carried out (Haxthausen et al., 2021), while a major common subject of investigation is guaranteeing sufficiently sensitive control of occurring contact forces.

The emphasis on and functional expectations from reliable force sensitivity in robotic medical ultrasound stand out rather prominently compared to other robotic force-sensitive applications within and outside of the healthcare sector because of the following reasons among others: The net duration of physical interaction in terms of actual contact and the temporal continuity thereof comprises a major portion of the total task execution time, as opposed to e.g. occupational deployments of robots in collaborative or other arrangements. Furthermore, the patient assumes the role of the actual object of interaction and is often partially or fully immobilized in a lying or seated position instead of possessing full agency and ability to physically react and evade potentially harmful external interactions as when collaboratively working together with a robot, leaving the human in a markedly exposed position. Although the patient’s involvement during the task can thus be thought of as mainly passive, often the patient still has a considerable level of basic agency in form of physiological, psychological, or otherwise medically induced movements, that nonetheless differentiate the patient as an interaction *subject* from normally inanimate interaction objects in industrial applications or patients under general anesthesia in robotic surgery. Another aspect that has to be considered is that the expected typical areas of interaction are not primarily restricted e.g. to the upper extremities as would be the case in hand-guided robotic payload handling or a well-defined anatomic internal target structure in robotic surgery, but can be located at any anatomic region of interest including generally sensitive or injured body parts. Lastly, there is a distinct discrepancy arising from the requirement of the executability of interactive motions a) over a wide kinematic and dynamic range on one hand and b) at a finely controllable scale on the other.

1.2 Contributions of this Thesis

While the challenge revolving around approaching human-like robotic force control is apparent in the context of the automation of critically force-sensitive medical and healthcare applications (Kumar et al., 2000), any progress made in this direction has obvious beneficial implications for similar applications of human-robot interaction such as in high-proximity collaborative robotics (Haddadin and Croft, 2016).

The accomplishment of the overarching goal of force-sensitive robotics can be considered as a classic control problem, that consists of the design of an appropriate regulator \mathbf{R} and an observer \mathbf{O} that together form a feedback-loop to control the system state \mathbf{S} . In the context of robotic force-control, \mathbf{S} represents the dynamic state of the robot and \mathbf{R} corresponds to the actuation system which generates motion control signals \mathbf{u} in form of motor voltages to alter \mathbf{S} , such that the force \mathbf{F} applied to the environment, e.g. during the physical interaction with a human, reaches the control goal of a task-specific target force \mathbf{F}_{targ} under the influence of external influences. The control-loop is closed by the feedback of an estimate \mathbf{F}_{obs} of \mathbf{F} based on an interpretation of the observable features of \mathbf{S} performed by the observer \mathbf{O} . Practically, this corresponds to the acquisition and computational processing of sensory data, which is naturally restricted by the limitations of sensorization, measurement error, noise and other external influences, collectively defined as \mathbf{e}_o (see Figure 1.1). Following from this, it is possible to identify two interrelated but still

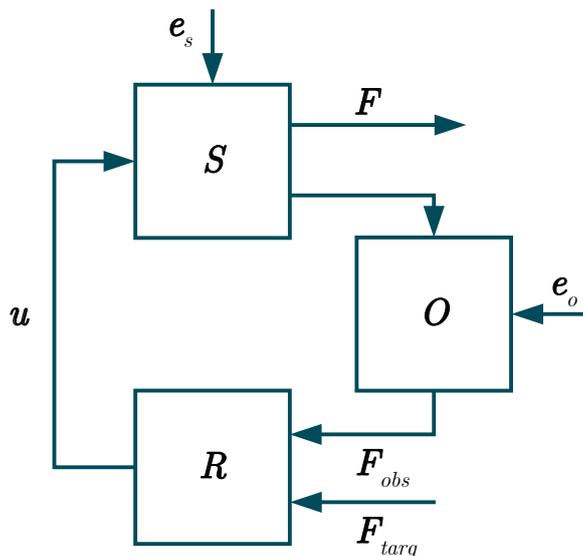


Figure 1.1: A closed loop control system. Adapted from (Glad and Ljung, 2018)

distinct elementary sub-problems within the general formulation that at the same time represent two main concepts of control theory, i.e. the *regulation* and *observation* of a system's state.¹

¹At the same time, this may also be understood as a paraphrase of the duality of controllability and observability (Kalman, 1960)

As the control performance is directly dependent upon the accuracy by which the actual interaction force \mathbf{F} can be approximated with \mathbf{F}_{obs} , the work presented in the following intends to contribute to the sub-problem of observability in robotic force control for physical interactions. Although a multitude of perception and sensing technologies are available that in combination would theoretically allow for a direct measurement of the occurring interaction forces z , in practice this is either often infeasible, due to technical complexity, or prone to measurement errors, as discussed in the next section. Therefore it may appear to be reasonable or even outright necessary to resort to a perception scheme that relies on the readily available sensory capabilities commonly integrated within robotic actuation systems (proprioception), i.e. joint position readings from motor encoders and torque measurements from joint-level strain gauges (Section 1.3 on page 8). Assuming the a priori availability of a model that establishes an accurate mapping between momentary proprioceptive observations of the joint positions, along with their temporal derivations, and observations of the joint torques resulting hereof, i.e. an inverse dynamics model, it is possible to trace any deviations between the joint torques measured and those estimated by the inverse dynamics model back to the entirety of external forces acting upon the robot, i.e. interaction forces in this context.

Practically speaking, the estimation of external forces is primarily deteriorated by modeling inaccuracies as well as by torque measurement errors distorting the discrepancy between the estimated and actual torques, which are the two central subjects of investigation covered by this thesis. Granted that the robust modeling of inverse dynamics minimizing these adverse effects is a well-studied topic in robotics (see Section 1.3 on page 8), the dynamic behavior of robots employed in the class of physical interaction tasks with continuous close human-robot proximity and high force-sensitivity requirements, exhibits several characteristics that specifically impede sufficiently accurate inverse dynamics modeling and force/torque estimation in this domain.

From the perspectives of kinematics and actuation performance, lightweight anthropomorphic robots with integrated joint torque sensors such as those of the KUKA LBR iiwa series, which is certified for human-interaction tasks and constitutes the reference platform evaluated in this thesis, are well-suited for the described applications. In terms of sensing, however, the combination of a predominance of comparatively low-intensity, irregularly directed robot motions in particular, i.e. during heuristic manipulative and explorative robot-human interactions, and the general requirement of high force-sensitivity does exhaust the torque sensing accuracy of joint-level strain gauges. Considering the discussed use cases of close-proximity and long-term human-robot interaction, torque sensing or modeling inaccuracies result in per-joint torque estimation errors otherwise deemed to be tolerable. Through the mechanic coupling between joints that is present in highly nonlinear dynamic systems such as robots, single joint torque estimation errors are prone to propagate and mutually reinforce each other along the kinematic chain. Effectively, this leads to an aggravation of the problem of estimating Cartesian forces acting at contact points on the robot during physical interactions. The origins of the disturbances on joint torque sensing that emerge under the outlined circumstances are identified and conjectured in this work to be e.g. hysteretical torque sensing depending on the joint position history, the transfer of vibrations from the actuator transmission on the strain gauges, complex friction, backlash etc.

To compensate for these effects and, a selection of state-of-the-art methods for inverse

dynamics modeling ranging from the baseline method of inverse dynamic modeling by classic parametric identification to weakly parameterized learning-based techniques are applied and evaluated on data sets of long-term proprioceptive time-series measurements acquired on the KUKA LBR iiwa during the execution of joint-wise independent reference trajectories. The research questions attempted to be answered are revolving around the infamous bias-variance tradeoff with respect to model design:

1. Can the robot dynamics be sufficiently incorporated in a closed-form model expression such as the Newtonian or Lagrangian formulation of rigid body dynamics and possibly yield a desirable global model validity by physically substantiated model assumptions; or, does this introduce an overly strong model bias, preventing the inclusion of marginal and hardly formalizable but nonetheless significant additional dynamic effects?
2. Are data-driven, prior-free modeling approaches, such as artificial neural networks as universal function approximators capable of providing higher model fidelity? Can they generalize to out-of-distribution dynamic states in spite of an inherent susceptibility to model variance given the scarcity of training data due to resource-intensive data acquisition in robotics?
3. Does an hybridization of both approaches provide a benefit over the respective stand-alone methods in terms of a synergetic combination of the rigidity of physical parametric models and the flexibility of neural networks?

These subjects are investigated by the initial application of a parametric identification and subsequent generation of an inverse dynamic model based on Newtonian rigid body dynamics. In a second step, a model based on the multilayer perceptron architecture is evaluated. As dynamics itself is concerned with the temporal change of a system due to acting forces, it is evident that a comprehensive modeling of robot inverse dynamics has to consider the factor of time. Thus it is further investigated how temporal information can be incorporated in inverse dynamics models to improve torque estimation accuracy by additionally evaluating the time-series architectures of Recurrent Neural Networks and Transformer networks. In this context the importance of recurrence for the modeling of temporal dependencies in proprioceptive time-series data is analyzed in comparison with model accuracies obtainable by machine learning methods not relying on sequential information. To this end, a joint positional encoding is developed as an additional feature to be fed into a neural network. The encoding is realized by concatenating every input vector with additional values for the rotational displacements since the most recent changes of the motion direction of the robot joints, aiming at partially substituting sequential information gained through network recurrence.

The last contribution is with respect to the data scarcity problem. Because a near-exhaustive sampling of the vast high-dimensional dynamic and kinematic space within which a kinematically redundant and agile robot can operate is technically infeasible, the amount of recorded data cannot be easily leveraged for statistical learning methods to yield sufficient generalization capabilities regarding cases of exposure to completely novel data. Likewise the limited expressiveness of parametric models results in insufficient. From this follows the issue of possibly unreliable model estimates when utilizing learning-based methods, which theoretically possess superior modeling capability in comparison to parametric methods. Therefore a hybrid model approach is presented where learning based estimators are inte-

grated as downstream error models predicting the error of a backbone pre-estimation by a model obtained through parametric identification of the robot’s rigid body dynamics. In order to mimic real-world heuristic explorative and interactive motions, an algorithm for the random generation of combinations of linear trajectory segments simulating general target approach motions and intermittent randomized parametric low-amplitude sinusoidal trajectories imitating fine-scale interactions has been implemented. In addition to ensuring good resemblance of the motions expected to be encountered under field conditions, the periodicity and joint-wise randomness of the motions is intended to provoke the mentioned torque sensing disturbances over wide range of possible inter-joint dynamic effects.

1.3 Related Work

From a hardware perspective, in order to realize high sensitivity in the force control of close-proximity human-robot interaction applications, the necessity of sensory feedback giving insight on momentarily occurring contact forces and torques is evident. Readily available sensing technologies based on mechanical deformation transducers, e.g. piezo-elements or more frequently strain gauges, constitute a natural choice for this purpose. Regarding the concrete means of the utilization of such force-sensing equipment in a robot, it is useful to assume the following perspective: Creating an analogy between a robotic system and an organism, both can generally be abstracted as sensorimotor systems, for whom, borrowing terminology from the field of biology, two conceptually different approaches may be distinguished with regards to the integration of the sensory elements: *Exteroception* and *proprioception*, the former denoting the perception of sensory input from outside, e.g. vision and cutaneous mechanoreception, and the latter reciprocally describing sensing from within, e.g. through mechanoreceptors embedded in the musculoskeletal system such as muscle spindles, Golgi organs, and articular mechanoreceptors. A complex neurological fusion and processing of these various sensory signals, whose elaboration exceeds the scope of this thesis, then yields a relatively practicable estimate of, and control accuracy over the applied forces during the execution of a force-sensitive interaction task by e.g. a human. Transferring these sensory concepts to robotic perception, one faces a task-specific design optimization problem concerning the most appropriate means of implementing these sensing strategies, i.e. resorting a) to a proprioceptive integration in the form of a force sensor placement inside of the actual kinematic components like the joints or b) exteroceptive sensing by the punctual placement of sensors outside of the robot chassis. Typically the proprioceptive sensing strategy is accomplished by the inference of mechanic torques from measured joint motor voltages, positions, dedicated serial rotational strain gauges between the robotic links and inertial measurement units or, when utilizing exteroception, by mounting of force/torque sensors at sites of assumed contact with the environment, most commonly as wrist sensors at the end-effector between the most distal link and attached tool.

Based on the number and structural arrangement of the mechanical force transducing sub-elements within a sensor, one can generally differentiate between solitary load cells for single-axis force measurements and multi-axis sensors, which allow for the sensing of forces and torques along and around multiple spatial axes, the most common sensor type

being 6-axis force/torque sensors. The application of external load cells or multi-axis force sensors has the major advantage of allowing for direct measurement of contact forces with potentially very high accuracy. This, however, comes at the cost of rather large unobservable physical contact or collision areas by nature of the highly local sensory space of these devices. In working environments affected by uncertainty as in human interaction applications, especially in medicine, this may lead up to the possibility of under-detected contacts and collisions between the robot and a human, i.e. when the contact point is not situated at exact location of an external force sensor. The mounting of multiple external force sensors distributed over the entire housing of the robot may seem to alleviate this problem to an extent, but nonetheless, the spatially discrete sensing scheme of this force measurement approach implies an unavoidable presence of spatial sensory gaps, where a simple increase of the sensor count would scale poorly with the increase in system complexity and robustness (Le, Choi, and Kang, 2013). Moreover, an inherent susceptibility to decalibration and general loss of accuracy through wear and external influences, e.g. temperature and mechanical stress, constitute some undesirable properties in robotic force sensing applications (Zhang et al., 2020). From a structural point of view, the addition of external force sensors may also introduce difficulties because of an alteration of the robot geometry and mass distribution, i.e. leading to a restriction of the collision-free configuration space, particularly when a kinematic design is required to optimize both the dexterous workspace coverage and compactness at the same time. Artificial skins, where multiple interconnected arrays of flexible mechanical transducer sub-elements are embedded in an elastic substrate layer, represent an emerging alternative external sensing technology capable of providing spatially continuous force measurements theoretically encompassing the total robot encasing (Hwang and Hwang, 2013; Calandra et al., 2015; Duan, Taurand, and Soleimani, 2019), albeit currently with a reduced absolute force measuring range and limited capability to yield physically quantifiable measurements of force magnitude.

Although external force sensing with dedicated strain gauge sensors yields high measurement accuracy, a proprioceptive approach to force sensing entails several attractive characteristics that are advantageous in multiple ways. First and foremost the argument of systemic simplicity is to be made. As stated before, proprioception indicates the structural embedding of the sensory elements inside of the components generating and transmitting motion, i.e. the joints and links. For the purposes of motion generation and transmission, robotic systems are equipped with joint position encoders and motor voltage controllers from the outset. These readily provide proprioceptive signals describing the momentary kinematic configurations and occurring joint torques that can either be inferred from motor voltages or, increasingly common since the introduction of lightweight serial robots intended for collaborative use with humans, measured directly by courtesy of dedicated joint torque sensors (Hirzinger et al., 2001). In general, such sensors are designed as rotational strain gauges that mechanically connect the output side of the joint gears with their respective distally following link. In comparison to the estimation of joint torques based on motor voltages, this largely avoids any measurement disturbances originating purely from actuator dynamics, e.g. motor internal friction, backlash, stutter, etc. Proprioception, however, does not necessarily have to be considered as a purely standalone approach, but also as a means to introduce redundancy into the sensorization of a robotic system in case of the presence of external sensory hardware.

Hence, a proprioceptive strategy provides a multitude of benefits over and in addition to other sensing methods for the observation of robot dynamics.

Due to the ubiquity of robotic application scenarios involving some form of physical contact, there exists a vast corpus of research on the topic of robotic force control.

With regards to the focus of this thesis on the force control sub-problem of force observability with proprioceptive sensorization, the theoretic foundations have been laid out in the seminal works on the parametric dynamic identification of robots within the framework of rigid body dynamics. By re-formulating the Newton-Euler equations of motion as an equation system in which the inertial link parameters describing the dynamics of the link bodies appear linearly, the inverse dynamic model can be obtained based on multiple measurements of the input-output relation between the link motion and forces by linear regression (An, Atkeson, and Hollerbach, 1985; Khosla and Kanade, 1985; Atkeson, An, and Hollerbach, 1986). This general principle of dynamic identification is utilized to characterize a dynamic model of the robot in order to relate any deviations from the model estimates of the dynamics state back to external influences, i.e. interaction forces. The majority of force estimation techniques are thereby conceptualized as a disturbance observer, that tracks the momentary dynamic and kinematic state through IMUs and/or joint sensors and performs a comparison with the prediction of a dynamic model for that particular state (Alcocer et al., 2003; Hacksel and Salcudean, 1994). Regarding the underlying dynamic model, several modifications of the original formulation based on the assumption of rigid body dynamics have been introduced, depending on the reliability of the observation of model input quantities, the technical properties of the robotic system, or the model accuracy requirements. An initial extension was the proposal of the identification of joint friction using a simplified Coulomb and Viscous friction model, which augments the notion of robot dynamics being based on the inertial properties of its links to also entail the influence of effects at joint level (Gautier and Khalil, 1988; Stürz, Affolter, and Smith, 2017b; Xu et al., 2020). In order to improve the statistical validity of the parameters to be identified, a variety of experiment design strategies exist for the generation of robotic test trajectories that optimally excite the effects influencing the robot dynamics (Swevers et al., 1997). Along with the advent of collaborative robotics lead by the development of the DLR LWR I to III (Albu-Schäffer et al., 2007), the disturbance observer is formulated as a pure momentum observer, disregarding dynamic effects due to link body acceleration in order to achieve faster model predictions and gain independence from inherently noisy acceleration measurements (De Luca et al., 2006; De Luca, Schroder, and Thummel, 2007). With the aim to achieve higher model fidelity, especially in the context of safety-critical human-robot interactions, dynamic effects caused by joint flexibility are added to the model assumptions by the simultaneous acquisition of the joint position on both the joint input and output side and the subsequent identification of additional joint stiffness and damping parameters (Haddadin, De Luca, and Albu-Schäffer, 2017).

In recognition of the partial insufficiency of parametric model formulations due to model incompleteness, laborious parameter identification procedures, and high levels of required domain knowledge, several efforts have been made in pursuit of applying black-box techniques for dynamic modeling and disturbance observation. These range from Gaussian processes to artificial neural networks (Nguyen-Tuong, Seeger, and Peters, 2009; Hitzler et al., 2019; Jiang, Ishida, and Sunawada, 2006; Liu, Wang, and Wang, 2021; Yilmaz et

al., 2020; Smith and Hashtrudi-Zaad, 2005) that are employed to learn a dynamic model directly from observable data to be used within a disturbance observer scheme. Recent advancements have been made by casting the dynamic modeling problem as a time-series process that can be leveraged by sequence-accepting neural networks, e.g. recurrent neural networks such as the long short-term memory architecture (Rueckert et al., 2017; Wang et al., 2020).

1.4 Structure of this Thesis

The contributions of this thesis with regards to the presented problem of proprioceptive force estimation during irregular low-intensity motion are organized in four parts:

First, the technical specifications of the robotic reference platform investigated in this thesis are analyzed and discussed from a hardware and software perspective in Chapter 2 on the following page. On one hand, this is done with regards to their respective implications on the task at hand. On the other hand, this is in order to gain insight into the physical and technical rationale behind the techniques of force estimation developed in this thesis.

The theoretical foundations of the model-based force estimation techniques used in this thesis are laid out in Chapter 3. Chapter 4 covers the presentation of the proposed novel dynamic modeling techniques.

This is concluded by a comparative performance analysis of the discussed techniques to determine the benefits gained from the developments presented in this work.

2

Experimental Setup

This chapter is intended to provide a description of the technical details of the experimentation system used for the development and evaluation of the techniques for dynamic modeling and force estimation proposed in this thesis. The chapter is structured in two parts, covering the hardware specifications of the utilized robotic platform and the details of the software system employed for the simulation and control of the robot.

2.1 Robotic Evaluation Platform

The entirety of the dynamic model implementations and experiments of this thesis were designed to be applied on the commercially available robot LBR iiwa 14 R820 (KUKA AG, Augsburg, Germany) which is part of the LBR iiwa series of collaborative robots together with the structurally analogous but smaller LBR iiwa 7 R800. The kinematic and mechatronic architecture of these robots is based on the fundamental advancements in lightweight collaborative robot design introduced with the research robot DLR LWR III (German Aerospace Center, Köln, Germany) (Albu-Schäffer et al., 2007). Given the inaccessibility of some details of the actual mechatronic architecture of the LBR iiwa 14 R820 due to manufacturer confidentiality, the structural and dynamic similarity to the DLR LWR III (Haddadin, De Luca, and Albu-Schäffer, 2017) allows the technical analysis of the LBR iiwa 14 R820 based on the technical specifications of the DLR LWR III, which are well-documented in the related research literature, when necessary.

In the context of long-term or rather continuous physical human-robot interaction applications, especially under uncontrolled operational conditions such as in the medical field, a large portion of the robot workspace is occupied by obstacles comprised of both cluttered objects in the environment and the physical body of the interacting human in close proximity. With the exception of the task-specific target area of interaction on the human body, this results in a high number of undesired possible collision points with the robot, greatly reducing the amount of collision-free configurations and configuration paths required for a particular target pose or motion. The kinematic redundancy of the LBR iiwa 14 with 6+1 degrees of freedom alleviates this problem of collision avoidance to some extent (Zhu et al., 2016). In addition to 6 degrees of freedom, i.e. 3 rotational and 3 translational, for the end-effector pose, an auxiliary degree of freedom for the re-configuration of the robot

2 Experimental Setup

without a change in the end-effector pose, i.e. a null-space movement, is provided by the anthropomorphic joint structure of the robot. Following a biomimetic design principle, the kinematics of the LBR iiwa 14 is modeled after the example of the human arm as a serial chain of $n = 7$ revolute joints, where the respective axis of rotation of one joint is perpendicular to its distally following joint.

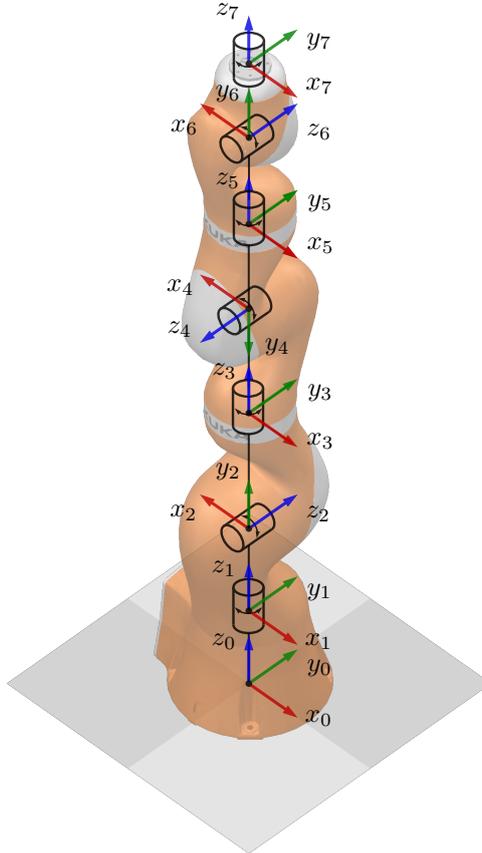


Figure 2.1: Mounting position and reference frame assignment for the joints of the kinematic chain of the robotic evaluation platform KUKA LBR iiwa 14 R820; reference frame 0 designates the world coordinate system located at the robot base. Please note that this frame assignment is deliberately chosen to be in accordance with the structural position of the actual joints and rotation axes between the links of the real KUKA LBR iiwa 14 R820, which is the reason for the otherwise kinematically irrelevant discrepancy between the mechanical link lengths and the Denavit-Hartenberg parameters \mathbf{d}_i (see Table 2.2 on the following page).

Through this configuration, often referred to as a spherical-revolute-spherical, the kinematic degrees of freedom of the human arm, likewise consisting of a spherical shoulder, revolute elbow, and spherical wrist joint, are emulated by approximating each spherical joint with three consecutive perpendicular joints j_1, j_2, j_3 and j_5, j_6, j_7 , and the intermediate elbow joint by a single revolute joint j_4 . In comparison to historically more conventional kinematic designs with 6 degrees of freedom, this redundant constellation with 7 joints has several benefits from the perspective of constrained motion planning, e.g. an increase in

2 Experimental Setup

the dexterous workspace, a relative decrease in end-effector path lengths to reach a target pose and generally improved pose reachability in the presence of obstacles. On a side note, it should be pointed out, however, that the notion of kinematic redundancy is intimately related to the functional requirements imposed on a robot during the execution of a particular task, in the sense that redundancy is only given when the kinematic performance exceeds the minimal dexterity prerequisites for the completion of that task, e.g 6 degrees of freedom for the three-dimensional translation and orientation of the end-effector pose (Chiaverini, 2019). The kinematic structure of the LBR iiwa 14 is depicted in Figure 2.1 on the previous page along with the orientation and position of the respective mechanic joint frames for the real robot in an upright configuration, which at the same time constitutes the mounting orientation in which the robot is operated during the experiments for this thesis. For comparability, the joint reference frames are defined virtually as in Table 2.2, following the formalism of the commonly used modified Denavit-Hartenberg convention (Craig, 2005; Denavit and Hartenberg, 1955).

Table 2.2: Kinematic description of the KUKA LBR iiwa 14 R820 in accordance with the modified Denavit-Hartenberg convention.

Link i	α_i (m)	a_i ($^\circ$)	d_i (m)	θ_i ($^\circ$)
1	0	0	0.360	θ_1
2	0	-90	0	θ_2
3	0	90	0.420	θ_3
4	0	90	0	θ_4
5	0	-90	0.400	θ_5
6	0	-90	0	θ_6
7	0	90	0.126	θ_7

Table 2.3: The range of motion and velocity of the respective joints of the KUKA LBR iiwa 14 R820 (KUKA Robot Group, 2015).

Joint	Position Range	Velocity Range
1	$\pm 170^\circ$	$\pm 85^\circ/\text{s}$
2	$\pm 120^\circ$	$\pm 85^\circ/\text{s}$
3	$\pm 170^\circ$	$\pm 100^\circ/\text{s}$
4	$\pm 120^\circ$	$\pm 75^\circ/\text{s}$
5	$\pm 170^\circ$	$\pm 130^\circ/\text{s}$
6	$\pm 120^\circ$	$\pm 135^\circ/\text{s}$
7	$\pm 175^\circ$	$\pm 135^\circ/\text{s}$

The design goals of anthropomorphism and usability within collaborative settings are also reflected in terms of the dynamics and internal electronics of the robot, and especially the joint design (see Figure 2.4 on the next page), exhibiting several features contributing to the operational functionality, safety, and controllability in physical human-robot interaction applications. As mentioned, detailed specifications of several components of the KUKA LBR iiwa are not publicly accessible. Due to the general structural equivalence with the DLR LWR III, as the architectural ancestor of sorts (Bischoff et al., 2010), the following statements are made in reference to the available data on the DLR LWR III. The motion of

the links is generated by compact low-speed and high-torque brushless DC motors originally developed by the German Aerospace Center under the name *RoboDrive*. The motor torque is transmitted onto the distally following link through a strain wave gear with a large reduction ratio between 100:1 and 160:1, more commonly known by the commercial name *Harmonic Drive*. Both a potentiometer-based joint position sensor as well as a strain gauge torque sensor are attached on the output side of the gearbox through a cross roller bearing, allowing for a measurement of the rotational displacements and torques arising between two successive links that is mechanically largely isolated from friction, vibration and other dynamic perturbations emerging in the motor or transmission. Nevertheless, the attachment of torque sensors onto the eponymously flexible flex-spine of the Harmonic Drive, as well as the inherent flexibility of the strain gauge sensor, which enables the mechanic torque-sensing after all, introduce significant measurement noise (Chawda and Niemeyer, 2017). In addition, hysteretical torque signal deviations depending on the rotational displacement since the last reversal of motion direction have been reported (Allgeier and Evans, 1995) and identified experimentally in this thesis as well.

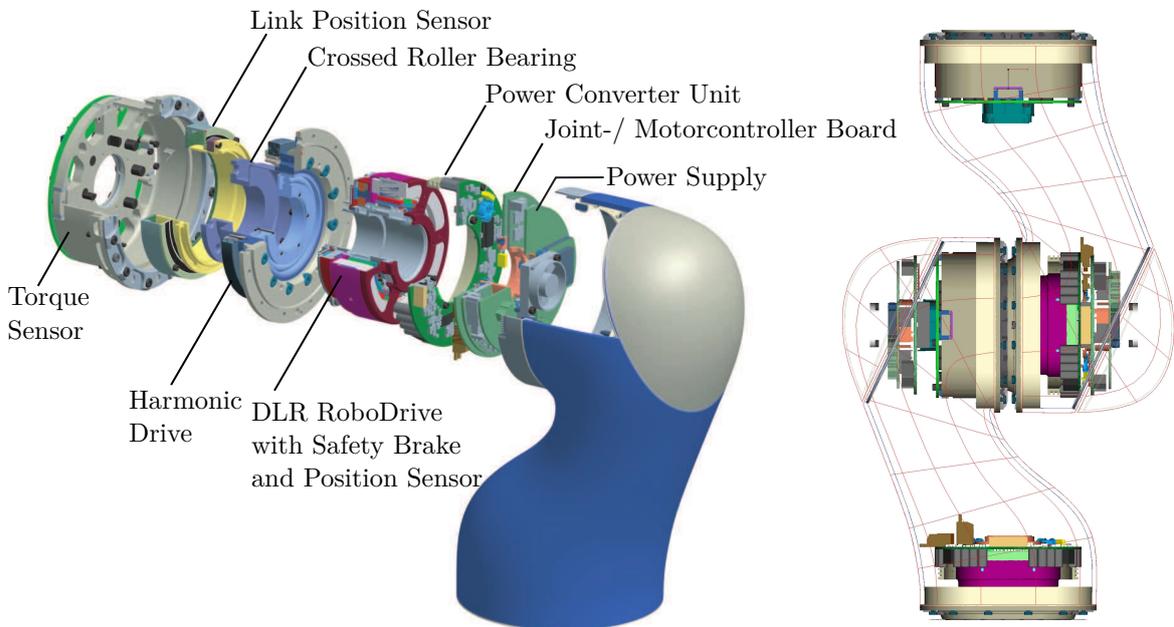


Figure 2.4: The internal structure of a joint of the DLR LWR III depicted in exploded view (Left) and assembly (Right), which the mechatronic structure of the commercial KUKA LBR iiwa investigated in this thesis has been directly derived from, rendering both virtually equivalent from a dynamic perspective. (Modified from (Albu-Schäffer et al., 2007)).

Without the execution of dedicated identification experiments to estimate the hysteresis parameters, a qualitative insight on the approximate geometry of the hysteresis loops associated with joint rotation and measured torques has been deemed to be sufficient. To this end, without the external application of any loads, a slow, joint-wise periodic triangular trajectory with constant velocity, except for the inflection at the triangle vertex, is commanded on the robot that allows for the generation of a joint position loop (see Figure 2.5 on the following page). Based on the visualization of the gathered measurements, one can

2 Experimental Setup

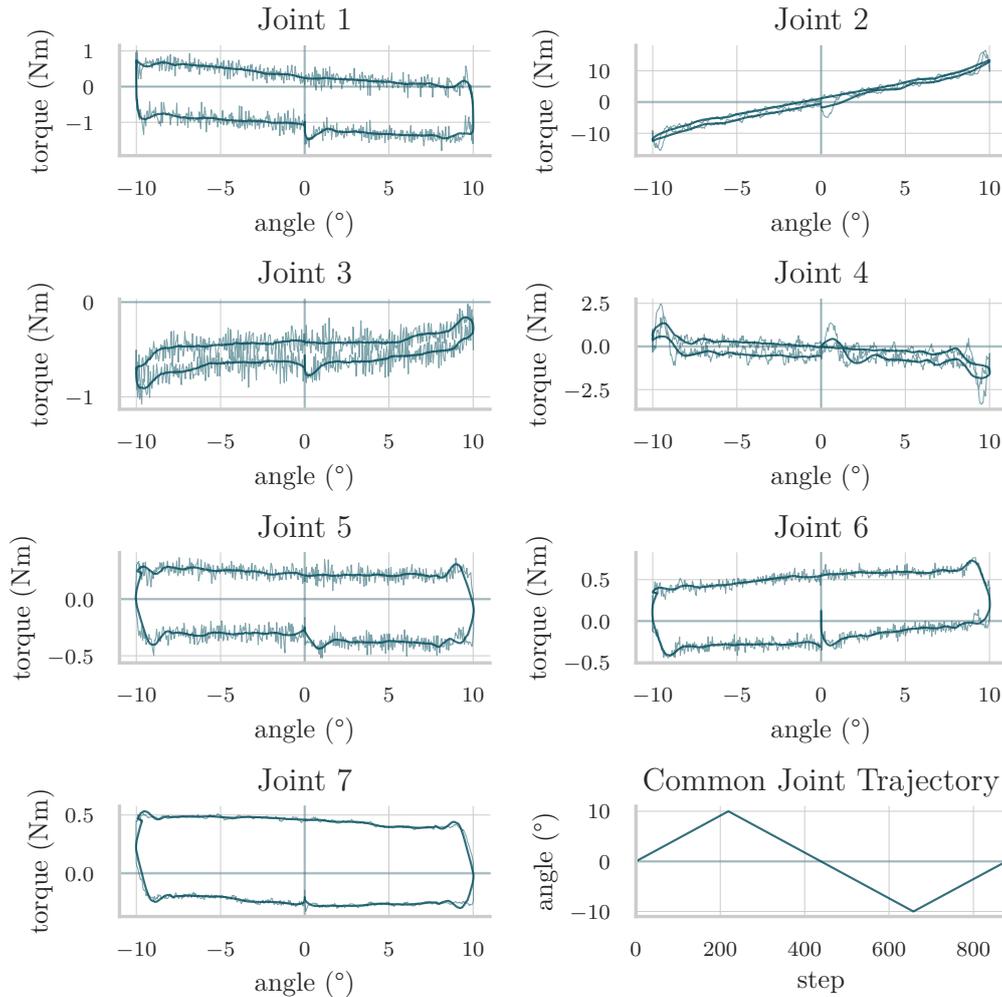


Figure 2.5: Torques measured in relation to angular deflection per joint during a cyclical linear joint-wise motion around the respective zero point with a maximum deflection of $\pm 10^\circ$. Please note the occurrence of significant torque noise in spite of a low relative joint velocity of 10 %, as well as the formation of a pronounced torque hysteresis loop with respect to the history of traversed joint orientations. In order to clarify the hysteretic effect, a smoothed curve based on the noisy measurements is obtained using a Savitzky-Golay filter and laid over the original curves respectively.

immediately recognize a pronounced non-linearity at motion initiation around the respective joints' zero positions. The same applies to the locations of directional reversal at 10° and -10° , where the torque signal shows down- or up-swings respectively. Furthermore, after a rotational displacement of approximately 6° in one motion direction, the hysteresis loops develop into saturation. These effects, however, are not noticeable with the same clarity for every joint given the obtained data, as the upright mounting position leads to a nonuniform influence of gravitational torques on the net measurements, i.e. depending on the cross-product of the gravitational vector with the respective joint's axis of rotation. For instance, the torques of joint 2 are nearly exclusively caused by the gravitational force

perpendicular to its rotational axis due to the masses of the distal links, whereas the torques sensed at joint 1 are almost solely due to its own internal dynamics, especially given the minimal contribution of distal inertial effects as the joint accelerations are constantly close to zero. Aside from this, the data demonstrate the presence of considerable relative torque measurement noise, even at low and constant joint velocities. In summary, this conducted simple experiment demonstrates that despite the availability of dedicated joint torque sensors, which are moreover mechanically insulated to some degree, a significant amount of perturbations and non-linear effects have an impact on the torque measurements.

2.2 Control System

The motion execution and retrieval of proprioceptive data is implemented as a teleoperation control network for this thesis, where joint-level motion commands and data queries sent from the control computer are redirected through proprietary high-level command interpreter application (*RL RobServer*, Institute of Robotics and Cognitive Systems, Lübeck, Germany) running on an external server towards a Java application that is executed on the robot controller (*KUKA Sunrise Cabinet*, KUKA AG, Augsburg, Germany). The communication between these devices is accomplished via Ethernet connection using the TCP/IP protocol at a frequency of approximately 150 Hz depending on minor fluctuations due to the momentary utilization of the local network infrastructure. The interpreted commands are then fed-forward onto the low-level motor controllers located in the robot Section 2.2, where the hardware control between the robot and the KUKA Sunrise Cabinet takes place at a rate of 1000 Hz. As such, the latency introduced by the TCP/IP connection constitutes the bottleneck with regards to the communication frequency, not allowing for a hard real-time control of the robot. Therefore, a soft-real time approach is taken using asynchronous joint motion commands with an immediate joint position retrieval. The joint positions to be commanded are generated in advance, stored in a buffer and then sent sequentially after a previously commanded joint configuration is reached. To further accommodate for the control rate, the distance in configuration space between the respective joint positions of two successive commands is taken into account together with the maximum joint velocity, such that the configuration space distance between joint position commands is adapted to allow for a motion completion during the latency time.

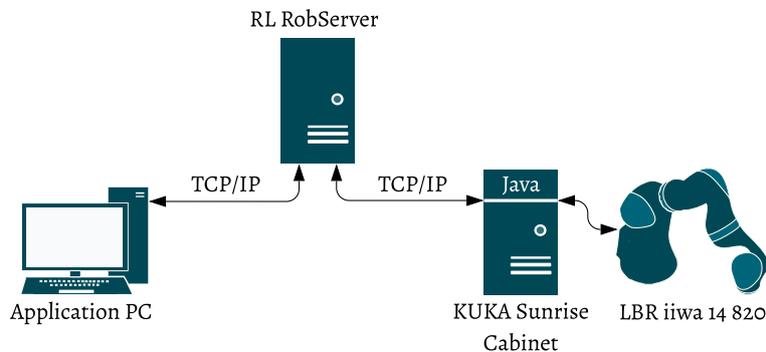


Figure 2.6: Architecture of the robot communication and control network.

2 Experimental Setup

For the generation, simulation, and commanding of the robotic motion, a collection of python applications were developed that integrate within the robotics simulation software *CoppeliaSim* formerly known as V-REP (Coppelia Robotics, Ltd., Zürich, Switzerland; (Rohmer, Singh, and Freese, 2013)) via the python interface toolkit *PyRep* (James, Freese, and Davison, 2019). Thereby, the generated motion commands are checked in simulation for kinematic and dynamic feasibility as well as self-collisions or those with the environment in advance and in real-time during motion execution on the robot as well. To this end, a CAD model of the KUKA LBR iiwa 14 is placed in the simulation environment and surrounded by manually placed virtual cuboid obstacles approximating the position of the obstacles in the proximity of the real robot in the laboratory. Through the simultaneous retrieval of the current joint positions of the real robot, the joint configuration of the simulated counterpart is updated, yielding a synchronized monitoring of the robot.

3

Force Estimation through Dynamic Identification

This section is intended as an outline of the concepts based on which proprioceptive force estimation can be performed independent from external sensory devices using dynamic modeling techniques. Beginning with a description of the formalisms involved, this is then concluded by the presentation of different classes of model-based techniques applicable for the solution of the general force estimation problem in robotics that form the basis on which the methods proposed and evaluated in this thesis were developed.

3.1 Dynamics of Serial Robots

When considering a robot with an open-loop serial kinematic chain composed of n rigid rotational joints with respectively adjacent links, its dynamic model D_{RBD} is commonly formulated by expressing the occurring total joint torques $\boldsymbol{\tau} \in \mathbb{R}^n$ as a function of generalized coordinates of the kinematic state, i.e. joint positions and their respective temporal derivatives $\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}} \in \mathbb{R}^n$

$$\boldsymbol{\tau} = D_{RBD}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \quad (3.1)$$

$$\boldsymbol{\tau} = H(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + g(\boldsymbol{q}) \quad (3.2)$$

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{RBD} , \quad (3.3)$$

where $H(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ denotes the symmetric and positive-definite joint-space inertia matrix describing the robot's inertia for a given joint-space configuration \boldsymbol{q} ; $C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ collectively expresses the Coriolis and centripetal forces depending on the vectors of joint position \boldsymbol{q} and velocity $\dot{\boldsymbol{q}}$; the torques arising due to the influence of gravitation for any configuration \boldsymbol{q} are represented by the term $g(\boldsymbol{q})$ (Featherstone and Orin, 2016). In other words, D_{RBD} establishes the mapping between a the kinematic state $(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$ and the dynamic state $\boldsymbol{\tau}$ based on the inertial properties of the rigid robot links, i.e. the rigid body dynamics.

As discussed in Section 1.3 on page 8, the state of the robot can only be assessed on the basis of observations made by sensory elements, i.e. joint torque or position sensors in the proprioceptive sensing approach. The state variables $\boldsymbol{\tau}$, \boldsymbol{q} , $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are therefore only observable with some uncertainty and residual error, which is why in the case of

applying the dynamic equations on a real system, the used state variables only resemble an imperfect representation of the respective true values. Moreover, there is no certitude over the completeness of the model formulation. Accommodating for these sensory inaccuracies and additional otherwise unmodeled factors that affect the dynamic model, an error term $\epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ is introduced:

$$\boldsymbol{\tau} = H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + \epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star). \quad (3.4)$$

Please note that ϵ , being a collective error function, does not necessarily depend on $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ alone but also on an undetermined set of other variables \star possibly influencing the dynamics or measurement of the model inputs. In order to semantically structure ϵ , i.e. the deviation from pure rigid body dynamics, the torque contributions of additional effects can be denoted separately according to their origin either as a function of internal joint dynamics $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) = \boldsymbol{\tau}_{int}$ or torques $\boldsymbol{\tau}_{ext}$ resulting from external mechanic influences, e.g. due to attached payloads, collisions or, in the context of this thesis, intentional physical human-robot interactions, as in

$$\boldsymbol{\tau} = H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) + \boldsymbol{\tau}_{ext}, \quad (3.5)$$

where $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ itself can be, in analogy to $\epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$, interpreted as a placeholder for an arbitrary number of additional functional terms further compensating for otherwise unmodeled phenomena. In the following, this term is primarily used in collective reference to the joint dynamic model describing torque contributions that emerge from within the joints and exceed the model assumption of joint rigidity. With the inclusion of $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ in D_{RBD} , the dynamic model can be augmented to account for flexible joints and related complex perturbations such as friction, backlash and vibrations originating either on the motor, transmission or link side of the joint (Haddadin, De Luca, and Albu-Schäffer, 2017; Chawda and Niemeyer, 2017; Albu-Schäffer et al., 2007). A commonly applied formalism to phenomenologically represent joint flexibility is to introduce a second set of generalized kinematic state coordinates $\boldsymbol{\theta}_m \in \mathbb{R}^n$ representing the joint rotations that are incongruent with the link rotations $\boldsymbol{\theta}_l = \mathbf{q}$ because of a displacement $\boldsymbol{\delta} = \boldsymbol{\theta}_m - \boldsymbol{\theta}_l$ due to flexibility within the joint. Due to the availability of joint torque and position sensors respectively located at the joint output side in the case of the KUKA LBR iiwa and similar robots, $\boldsymbol{\tau}$ on the left and $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ on the right side of Equation (3.5) can be considered directly observable and mostly exempt from mechanical disturbances and flexibility effects originating from the motor or transmission. For this reason, $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ predominantly accounts for residual flexibility and friction phenomena after the transmission that arise from the the joint torque sensors and their respective mechanic connection to their distally attached links. Under the assumption that there is no inertial coupling between the joint and link bodies, and the dynamic effect of joint friction and flexibility contained within the term $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ can be assumed to be causally restricted to the link side or output side of the transmission

$$i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) = \mathbf{K}(\boldsymbol{\theta}_l - \boldsymbol{\theta}_m) - \mathbf{V}(\dot{\boldsymbol{\theta}}_j - \dot{\boldsymbol{\theta}}_l) + \boldsymbol{\tau}_{f,l} + i_\epsilon(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star), \quad (3.6)$$

where $\mathbf{K} = \text{diag}\{K_j\} \in \mathbb{R}^{n \times n}$ is the joint series elastic stiffness matrix and $\mathbf{V} = \text{diag}\{V_j\} \in \mathbb{R}^{n \times n}$ is the joint damping matrix, each respectively describing the torque contributions occurring due to the torsion between the link and motor rotations; $\boldsymbol{\tau}_{f,l}$ are the torques

due to link side friction (Spong, 1987; Albu-Schäffer, Ott, and Hirzinger, 2007). Following Equation (3.5) on the preceding page, the dynamics model D_{RBD} in Equation (3.1) on page 19 can be modified as

$$\boldsymbol{\tau} = D(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) + \boldsymbol{\tau}_{ext} . \quad (3.7)$$

Provided that for a given robot the values of the functional expressions on the right-hand side of Equation (3.5) on the preceding page are known along with measurements of $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ and $\boldsymbol{\tau}$, the equation can be rearranged to isolate the external torques, such that

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_H + \boldsymbol{\tau}_C + \boldsymbol{\tau}_g + \boldsymbol{\tau}_{int} - \boldsymbol{\tau} \quad (3.8)$$

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{rgb} + \boldsymbol{\tau}_{int} - \boldsymbol{\tau} \quad (3.9)$$

$$\boldsymbol{\tau}_{ext} = \boldsymbol{\tau}_{dyn} - \boldsymbol{\tau} \quad (3.10)$$

which allows inferring an external interaction in terms of the deviation of the measured torques $\boldsymbol{\tau}$ from the torques $\boldsymbol{\tau}_{dyn}$ obtained through the dynamic model for the respective kinematic state of the robot without external influences. The torques $\boldsymbol{\tau}_{ext}$ can be traced back to the causal external interaction, by establishing a relation between the wrench $\mathbf{w}_{ext} = \begin{bmatrix} \mathbf{f}_{ext} \\ \mathbf{n}_{ext} \end{bmatrix} \in \mathbb{R}^6$, composed of a force component $\mathbf{f}_{ext} \in \mathbb{R}^3$ and a moment $\mathbf{n}_{ext} \in \mathbb{R}^3$ acting on the robot at a point of contact due to the external interaction, via the geometric Jacobian $\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_{lin}(\mathbf{q}) \\ \mathbf{J}_{ang}(\mathbf{q}) \end{bmatrix}$ of the manipulator associated with the location of contact on the robot, which, for simplicity, is assumed to be restricted to the robot end-effector in the course of this thesis:

$$\mathbf{w}_{ext} = \mathbf{J}^{-\top}(\mathbf{q}) \boldsymbol{\tau}_{ext} \quad (3.11)$$

$$\mathbf{w}_{ext} = \mathbf{J}^{-\top}(\mathbf{q}) [\boldsymbol{\tau}_{dyn} - \boldsymbol{\tau}] \quad (3.12)$$

$$\mathbf{w}_{ext} = \mathbf{J}^{-\top}(\mathbf{q}) [D(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) - \boldsymbol{\tau}] \quad (3.13)$$

$$\mathbf{w}_{ext} = \mathbf{J}^{-\top}(\mathbf{q}) [H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) - \boldsymbol{\tau}] . \quad (3.14)$$

$\mathbf{J}(\mathbf{q})$ can be constructed based on the geometric description of the robot, where the explicit derivation for redundant kinematic structures as investigated in this thesis follows the formulation in (Schweikard and Ernst, 2015) that mitigates the problem of rang deficiency in the Jacobian matrix arising for redundant kinematics.

3.2 Dynamic Modeling as a Statistical Regression Task

Based on this relationship, the task of estimating external forces, e.g. those due to physical human-robot interactions, is tantamount to the identification of the dynamic model D on the right-hand side of Equation (3.13), which can be used to infer the expected dynamic state given a set of motion measurements, i.e. the inverse dynamics. As a robot constitutes a highly complex dynamic system under the influence of possibly an indefinite number of physical effects, the exhaustive characterization of its true dynamics D may be intractable due to its high dimensionality. Therefore, the aim is to identify the model $\boldsymbol{\tau}_{est} = F(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ that

approximates the real dynamics $\boldsymbol{\tau}_{dyn} = D(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ to infer with maximum accuracy how the dynamic state of the robot would be if there was no external influence, given multiple measurements of $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ made by the available sensory elements, such that

$$\|D(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) - F(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)\| \rightarrow 0 \quad (3.15)$$

$$\|\boldsymbol{\tau}_{dyn} - \boldsymbol{\tau}_{est}\| \rightarrow 0 \quad (3.16)$$

over a maximum range of possible kinematic states $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. This is generally accomplished by constructing F as a finite parametric representation of D . Probing the input-output relation of F for a given input $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$, the parameters of F are adapted, such that the output approximates that of D for the same input:

$$\arg \min_F \|D(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) - F(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)\|_2. \quad (3.17)$$

In order to ensure good approximation for a wide variety of inputs, this step is repeated for multiple data pairs of inputs $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ and targeted outputs $\boldsymbol{\tau}$. This results in a statistical regression task in pursuit of those functions and function parameter values, yielding the best approximation of the target data. Put differently, we are facing a supervised machine learning problem, where the joint torque measurements that occur under a kinematic state are defined as the supervision signal. Accordingly, the performance of any functional model representation F is evaluated with respect to the similarity of its estimation $\boldsymbol{\tau}_{est}$ to the observed dynamic torque $\boldsymbol{\tau}_{dyn}$ given a particular observation of the kinematic state. A commonly utilized metric to statistically assess the model performance over m kinematic states is the *mean squared error* (MSE) or *root mean squared error* (RMSE), respectively:

$$\text{MSE} = \frac{1}{m} \sum_i^m (D(q_i, \dot{q}_i, \ddot{q}_i, \star_i) - F(q, \dot{q}_i, \ddot{q}_i, \star_i))^2 \quad (3.18)$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_i^m (D(q_i, \dot{q}_i, \ddot{q}_i, \star_i) - F(q, \dot{q}_i, \ddot{q}_i, \star_i))^2}. \quad (3.19)$$

With regards to the parametric representation of the true dynamics, there are generally two model paradigms to be followed: parametric and non-parametric modeling, where the exact definition of these terms varies according to the context they are used in. For the purpose of this work, consider the notion that dynamic models exhibit a spectrum of varying levels of parameterization and a model F is denoted as tending towards being parametric, if its approximation of the true dynamics D is conceived by formulating an explicit expression that relates a small set of semantically predefined parameters to each other to infer the dynamic state, given an observation of the kinematic state. The prerequisite for this is the ability to characterize the majority of the dynamic effects as a well-defined interrelation of those parameters, e.g. real physical quantities, that have the highest influence on the dynamics. Contrary to this, in models that can be classified rather non-parametric, no such a priori information, regarding the expected relationship and meaning of those parameters that the model depends on, is applied. Instead, the model is structured to contain a rather large set of semantically undefined and initially unspecifically interrelated parameters, where their values and mutual influence are adapted solely based on the observed input-output

relations. In fact, no strict assumption on the dimensionality of the parameter space is made in advance at all; but it is rather implicitly defined by the statistical data distribution.

These two model paradigms can be thought of as representing two extremes on the spectrum of model parameterization, often termed *white-* and *black-box* models, where the level of parameterization structurally affects the prediction accuracy obtainable by model: A highly parameterized white-box model, i.e. one that assumes D to be the described by a well-defined, analytic relationship of semantically specified parameters, imposes a strong limitation on the possible model output. If the a priori defined model structure sufficiently reflects the true dynamics, the high degree of parameterization averts an undesired variance in the model predictions, leading to robust predictions over a wide range of inputs. At the same time, however, the formulation of such a rigid model, comes with the risk of introducing a prediction bias, by completely disregarding dynamic effects that have not been considered beforehand out of unawareness about their existence or because their complexity prevented them to be formulated mathematically.

Conversely, with a low level of parameterization in black-box models, the increase in model expressiveness introduces the possibility of high variance in the model predictions, as the model's parametric structure is not handcrafted to resemble specific dynamic effects from the outset. A benefit of the low parameterization that accompanies this disadvantage, is the simultaneously diminished susceptibility towards prediction biases caused by the otherwise strong model assumptions.

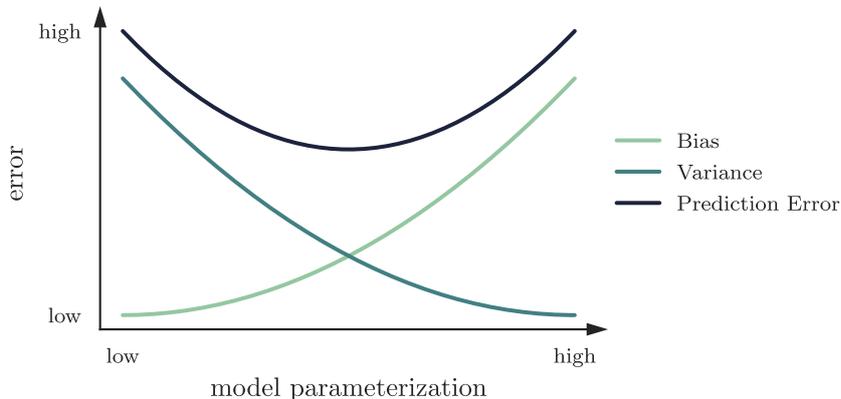


Figure 3.1: Qualitative relation between the level of parameterization of a dynamic model and its prediction characteristics.

From this consideration, it can be conjectured that a balanced level of parameterization is potentially desirable to ensure both good prediction robustness against statistical outliers in the kinematic input state by incorporation of enough a priori knowledge, as well some degree of model flexibility to account for dynamic effects that are difficult to capture. From another perspective, this trade-off between model bias and variance can be regarded as a particularity of the superordinate, and often sought-after generalization ability in statistical learning. Considering the task of modeling robot dynamics, model generalization is of special importance, as the high dimensionality of the kinematic and dynamic space pre-

vents carrying out an exhaustive regression on the complete input space. For this reason, the following sections cover the fundamentals of parametric robot model identification on one hand and the application of various artificial neural network-based machine learning techniques for dynamic identification on the other.

3.3 Parametric Dynamic Model Identification

Disregarding the influence of possible perturbations and other complicating effects subsumed under $i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$ (see Equation (3.5) on page 20), the dynamics of a robot can be described by the collective mechanic properties of its links, i.e. the position, connectivity, movement and distribution of mass of the respective rigid bodies representing those links, which constitutes the eponymous *rigid body dynamics*. This follows immediately from the model defined in Equation (3.3) on page 19, which regards only forces and physical effects pertaining to the motion of bodies with mass, i.e. phenomena mediated by the inertial effects of those bodies.

As such, it appears convenient to reformulate the physical properties implicitly embedded in the terms of the dynamic model of the robot (Equation (3.3) on page 19) to relate the joint torques to an explicitly stated, finite set of parameters. This exploitation of the dependency of a rigid body system’s dynamics on a predefined limited number of parameters greatly simplifies the problem of parameter identification, and consequently the estimation of external forces. A related benefit of this bounded model complexity is the reduction of variability in the model predictions. However, it has to be kept in mind, that such a simplification is inevitably tantamount to a concurrent restriction of the model’s overall representative capacity.

The necessary parameters to describe the dynamic system model of a robot can be divided into kinematic and inertial parameters (Hollerbach, Khalil, and Gautier, 2016). The kinematic parameters, especially in the case of mass-produced robots, are usually readily accessible with sufficient accuracy via technical documentation or construction data provided by the manufacturer, because of which it is often not required to perform dedicated kinematic identification experiments to obtain the geometry of the individual links of the robot. Referring to the kinematic description of the robotic reference system in Section 2.1 on page 12, the further examination of this otherwise important preliminary step is omitted with regards to the scope of this work. The inertial parameters of a given robot, however, are rarely published and only obtainable from CAD data with great uncertainty, which necessitates the experimental identification of those parameters (Khalil and Dombre, 2002), the methodological outline of which is presented in the following.

Extending the notation used so far, each link i is equipped with reference frames and positioned relative to the origin during motion as shown in Figure 3.2 on the following page, where \mathbf{O}_i denotes the reference frames of the links which are connected by the vector \mathbf{s}_i with their respective distally following link $i + 1$. The link frames \mathbf{O}_i are positioned in the origin of their proximal joints relative to the global frame \mathbf{O}_0 by the vector \mathbf{p}_i , where its derivatives also describe the linear motion of link i . \mathbf{C}_i denotes the location of the center of mass of link i and is located by \mathbf{c}_i with respect to \mathbf{O}_i and \mathbf{r}_i in global coordinates respectively. The linear forces acting upon the link i are represented by the gravity vector \mathbf{g} and

an arbitrary force vector \mathbf{f}_f (see Figure 3.2);

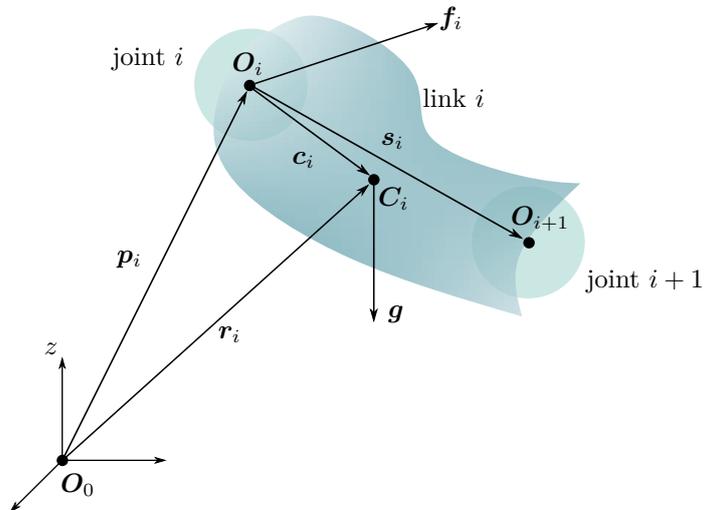


Figure 3.2: Location of the reference frames and vectors of an intermediate link. Adapted from (Atkeson, An, and Hollerbach, 1986)

Based on this assignment of reference frames and vectors, these inertial parameters can be arranged in a parameter matrix $\Phi \in \mathbb{R}^{p \times n}$ composed of n columns of p parameters for each link $i = 1, \dots, n$, where Φ_i denotes the i -th column and is commonly defined to include 10 inertial parameters and 2 friction parameters (Khalil and Dombre, 2002) for link i as in

$$\Phi_i = [m_i \quad m_i c_i^x \quad m_i c_i^y \quad m_i c_i^z \quad I_i^{xx} \quad I_i^{xy} \quad I_i^{xz} \quad I_i^{yy} \quad I_i^{yz} \quad I_i^{zz} \quad F_i^v \quad F_i^c]^\top. \quad (3.20)$$

For the utilization of Φ within linear equations in the following passages, the notation Φ is overloaded to, if necessary, also refer to the vectorized form of Φ following column-major order. m_i denotes the mass of link i , c_i^* is the $*$ -component of the vector pointing at the center of mass of link i with respect to the link's center of mass C_i . $I_i^{*,*}$ represents an element of the symmetric inertia tensor of link i

$$I_i = \begin{bmatrix} I_i^{xx} & I_i^{xy} & I_i^{xz} \\ \cdot & I_i^{yy} & I_i^{yz} \\ \cdot & \cdot & I_i^{zz} \end{bmatrix},$$

again, in relation to C_i (see Figure 3.2). F_i^v and F_i^c respectively designate parameters for the viscous and Coulomb frictions impeding the free motion of link i with respect to link $i - 1$. The influence of actual joint dynamics on the total robot dynamics observable by the aforementioned availability of sensors on the joint output side is obviously minimal. Because of this, the robot dynamics could be solely expressed in terms of link body parameters. However, due to the ease of integration of, strictly speaking, *joint* parameters of friction within a linear model formulation mentioned in the following, these parameters are commonly included.

Supposing the existence of a model that describes the dynamic behavior of a robot with link parameters Φ , the identification of Φ can in general be performed by using two structurally different formulations, beginning with

$$\boldsymbol{\tau} = \mathbf{f}(\mathbf{x}, \Phi), \quad (3.21)$$

where $\mathbf{f} = \{f_1, \dots, f_n\}$ is an arbitrarily complex but explicit expression that evaluates for the joint torques $\boldsymbol{\tau}$ given the input variables $\mathbf{x} \in \mathbb{R}^{n \times *}$ that are directly available by measurement, i.e. at least $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ and the linkage structure, and the sought link parameters Φ . Because elements of Φ themselves may appear non-linearly in the functional terms of \mathbf{f} , this model formulation is accordingly called a nonlinear model. As such, it is evident that to solve the model equation for Φ a likewise non-linear solving technique is required, which in consequence entails the usual challenges of non-linear optimization, revolving around e.g. convergence, optimality and computational complexity. In avoidance of this, the model may also be constructed as a linear equation system

$$\boldsymbol{\tau} = \mathbf{F}(\mathbf{x}) \Phi, \quad (3.22)$$

in which Φ appears linearly with respect to a model matrix $\mathbf{F}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star) \in \mathbb{R}^{n \times p}$ (Atkeson, An, and Hollerbach, 1986), again containing arbitrarily complex expression, that now, however, only depend on the input variables \mathbf{x} . The actual internal structure of \mathbf{X} can theoretically incorporate an arbitrary amount of complex, nonlinear functions, which likewise use an indefinite number of inputs, to model the dynamics of the robot. The derivation of the internal structure of the model matrix is commonly based on the equivalent classical formulations of either Lagrangian (Stürz, Affolter, and Smith, 2017b) or Newtonian mechanics (Atkeson, An, and Hollerbach, 1986). Due to reasons of computational efficiency (Featherstone, 1987) regarding the algorithmic implementation (Section 4.1 on page 46), the Newtonian formulation is chosen in the following. As proposed initially by (Atkeson, An, and Hollerbach, 1986), the internal structure of \mathbf{X} is derived from the Newton-Euler formulation of the rotational dynamics of rigid bodies, which shall be presented next.

During motion, each link i is acted upon by a net wrench \mathbf{w}_i composed of a net force \mathbf{f}_i and torque \mathbf{n}_i about the center of mass \mathbf{C}_i defined by the classical Newton-Euler equations as

$$\mathbf{f}_i = \mathbf{f}_{i,i} + m_i \mathbf{g} = m_i \ddot{\mathbf{r}}_i \quad (3.23)$$

$$\mathbf{n}_i = \mathbf{n}_{i,i} - \mathbf{c}_i \times \mathbf{f}_{i,i} = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i), \quad (3.24)$$

where $\mathbf{f}_{i,i}$ and $\mathbf{n}_{i,i}$ designate a force and torque, respectively, acting at joint i solely due to the motion of link j . Accordingly, let $\mathbf{w}_{i,i}$ be the wrench acting upon link i that is composed of the force and torque vectors occurring solely due to the motion of link i . $\boldsymbol{\omega}_i = [0 \ 0 \ q_i]^\top$ denotes the vector describing the angular velocity q_i by which link i is rotating around the z-axis share by joint i and link i denoted by $\boldsymbol{\omega}_i$.

Because the exact location of the center of mass \mathbf{C}_i may not be determinable, the acceleration of \mathbf{C}_i with respect to \mathbf{O}_0 denoted by $\ddot{\mathbf{r}}_i$ may be expressed in terms of the acceleration of \mathbf{p}_i pointing at the reference frame of joint i which is known from the kinematic structure:

$$\ddot{\mathbf{r}}_i = \ddot{\mathbf{p}}_i + \dot{\boldsymbol{\omega}} \times \mathbf{c}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{c}_i) \quad (3.25)$$

3 Force Estimation through Dynamic Identification

Updating the formulation of the Newton-Euler equations in Equation (3.23) on the preceding page and Equation (3.24) on the previous page with this expression of $\ddot{\mathbf{r}}_i$ yields

$$\mathbf{f}_i = m_i(\ddot{\mathbf{p}}_i - \mathbf{g}) + \dot{\boldsymbol{\omega}}_i \times m_i \mathbf{c}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times m_i \mathbf{c}_i) \quad (3.26)$$

$$\begin{aligned} \mathbf{n}_i = & (\mathbf{g} - \ddot{\mathbf{p}}_i) \times m_i \mathbf{c}_i + \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + m_i \mathbf{c}_i \times (\dot{\boldsymbol{\omega}}_i \times \mathbf{c}_i) + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) + \\ & m_i \mathbf{c}_i \times (\boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{c}_i)) . \end{aligned} \quad (3.27)$$

As the unknown location of \mathbf{C}_i is still present in the form of \mathbf{c}_i appearing quadratically in the terms $\mathbf{c}_i \times (\dot{\boldsymbol{\omega}}_i \times \mathbf{c}_i)$ and $\mathbf{c}_i \times (\boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{c}_i))$, this factor of uncertainty can be mitigated by substituting the inertia tensor \mathbf{I}_i defined about the center of mass with the inertia tensor ${}^p\mathbf{I}_i$, which is defined about the center of the joint i , using the relation

$${}^p\mathbf{I}_i = \mathbf{I}_i + m_i[(\mathbf{c}_i^\top \mathbf{c}_i)\mathbf{1} - (\mathbf{c}_i \mathbf{c}_i^\top)] , \quad (3.28)$$

with $\mathbf{1}$ being the three-dimensional identity matrix, and substituting into Equation (3.28):

$$\begin{aligned} \mathbf{n}_i = & (\mathbf{g} - \ddot{\mathbf{p}}_i) \times m_i \mathbf{c}_i + \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + m_i[(\mathbf{c}_i^\top \mathbf{c}_i)\mathbf{1} - (\mathbf{c}_i \mathbf{c}_i^\top)]\dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) + \\ & \boldsymbol{\omega}_i \times (m_i[(\mathbf{c}_i^\top \mathbf{c}_i)\mathbf{1} - (\mathbf{c}_i \mathbf{c}_i^\top)]\boldsymbol{\omega}_i) \end{aligned} \quad (3.29)$$

$$\mathbf{n}_i = (\mathbf{g} - \ddot{\mathbf{p}}_i) \times m_i \mathbf{c}_i + {}_r\mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times ({}_r\mathbf{I}_i \boldsymbol{\omega}_i) . \quad (3.30)$$

Building on this formulation of (Atkeson, An, and Hollerbach, 1986), the effect of the additional set of the two frictional parameters \mathbf{F}_i^v and \mathbf{F}_i^c on \mathbf{n}_i (see Definition Equation (3.20) on page 25) can be included, yielding:

$$\mathbf{n}_i = (\mathbf{g} - \ddot{\mathbf{p}}_i) \times m_i \mathbf{c}_i + {}_r\mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times ({}_r\mathbf{I}_i \boldsymbol{\omega}_i) + \mathbf{F}_i^v \dot{\boldsymbol{\omega}}_i + \mathbf{F}_i^c \text{sign}(\dot{\boldsymbol{\omega}}_i) . \quad (3.31)$$

The finally derived forms of the wrench $\mathbf{w}_{i,i}$ composed of \mathbf{f}_{ii} and \mathbf{n}_{ii} that is acting upon link i is now transformed into a matrix formulation, where motion parameters \mathbf{g} , $\ddot{\mathbf{p}}_i$, $\boldsymbol{\omega}_i$ and $\dot{\boldsymbol{\omega}}_i$ appear in the matrix \mathbf{F}_i and inertial parameters m_i , \mathbf{c}_i and ${}_r\mathbf{I}_i$ in the matrix Φ_i respectively:

$$\mathbf{w}_{i,i} = \mathbf{F}_i \Phi_i . \quad (3.32)$$

To do so, the operands in terms containing the matrix operations of the cross- and dot-products have to be adapted to the matrix formulation first, such that the respective operations can be equivalently expressed by matrix multiplication in the order of operands permitted by the linear equation system:

$$\boldsymbol{\omega} \times \mathbf{c} = [\boldsymbol{\omega} \times] \mathbf{c} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix} , \quad (3.33)$$

$$\mathbf{I} \boldsymbol{\omega} = [\cdot \boldsymbol{\omega}] \mathbf{I} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 & 0 & 0 \\ 0 & \omega_x & 0 & \omega_y & \omega_z & 0 \\ 0 & 0 & \omega_x & 0 & \omega_y & \omega_z \end{bmatrix} \begin{bmatrix} I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \end{bmatrix} , \quad (3.34)$$

where $[\mathbf{w}\times]$ and $[\cdot\mathbf{w}]$ respectively express the cross- and dot-product between \mathbf{w} and another matrix as a matrix multiplication. Using these operators, the linear equation in Equation (3.32) on the preceding page can be formulated as

$$\mathbf{w}_{ii} = \begin{bmatrix} \mathbf{f}_{ii} \\ \mathbf{n}_{ii} \end{bmatrix} = \begin{bmatrix} \ddot{\mathbf{p}}_i - \mathbf{g} & [\dot{\boldsymbol{\omega}}_i \times] + [\boldsymbol{\omega}_i \times][\boldsymbol{\omega}_i \times] & \mathbf{0} \\ \mathbf{0} & [(\mathbf{g} - \ddot{\mathbf{p}}_i) \times] & [\dot{\boldsymbol{\omega}}_i] + [\boldsymbol{\omega}_i \times][\dot{\boldsymbol{\omega}}_i] \quad \dot{\boldsymbol{\omega}}_i \quad \text{sign}(\dot{\boldsymbol{\omega}}_i) \end{bmatrix} \times \begin{bmatrix} m_i \\ m_i \mathbf{c}_i \\ I_{xx} \\ I_{xy} \\ I_{xz} \\ I_{yy} \\ I_{yz} \\ I_{zz} \\ F_i^v \\ F_i^c \end{bmatrix}, \quad (3.35)$$

which for the moment only relates the wrench $\mathbf{w}_{i,i}$ acting upon link i as a result of its *own* motion and the influence of gravity to its inertial parameters. However, as the robot is composed of n connected links, the total wrench \mathbf{w}_i arising at link i is the combined effect of $\mathbf{w}_{i,i}$, i.e the wrench due to its own motion \mathbf{F}_i and inertial parameters Φ_i , and the wrenches $\mathbf{w}_{ij}, \dots, \mathbf{w}_{i,n}$, which, in turn, denote the contributions of the wrenches $\mathbf{w}_{i+1,i+1} \dots \mathbf{w}_{n,n}$ transmitted onto link i , occurring solely due to their own respective motion at the links $i+1, \dots, n$:

$$\mathbf{w}_i = \sum_{j=i}^n \mathbf{w}_{ij}. \quad (3.36)$$

The wrench contribution $\mathbf{w}_{i,i+1}$ transmitted from the distal link $i+1$ onto the proximal link i is calculated by pre-multiplication with a screw transformation matrix \mathbf{T}_i as in

$$\mathbf{w}_{i,i+1} = \mathbf{T}_i \mathbf{w}_{i+1,i+1} \quad (3.37)$$

$$\begin{bmatrix} \mathbf{f}_{i,i+1} \\ \mathbf{n}_{i,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_i & \mathbf{0} \\ [\mathbf{s}_i \times] \mathbf{R}_i & \mathbf{R}_i \end{bmatrix} \begin{bmatrix} \mathbf{f}_{i+1,i+1} \\ \mathbf{n}_{i+1,i+1} \end{bmatrix}, \quad (3.38)$$

where \mathbf{R}_i is the rotational part of the homogeneous transformation matrix \mathbf{M}_i that locates the reference frame of link $i+1$ relative to the reference frame of link i . \mathbf{M}_i may be constructed in accordance with the parameters $\mathbf{k}_i = [a_i, \alpha_i, d_i, \theta_i]$ of the Denavit-Hartenberg convention, defining the relative position and orientation link i within the kinematic structure of a given manipulator (Denavit and Hartenberg, 1955) as follows:

$$\mathbf{M}_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & a_i \\ \sin(\theta_i) \cos(\alpha_i) & \cos(\theta_i) \cos(\alpha_i) & -\sin(\alpha_i) & -\sin(\alpha_i) d_i \\ \sin(\theta_i) \sin(\alpha_i) & \cos(\theta_i) \sin(\alpha_i) & \cos(\alpha_i) & \cos(\alpha_i) d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3.39)$$

It should be noted that through the integration of \mathbf{R}_i containing the kinematic parameters, and especially the variable DH-parameter $\theta_i = q_i$, the transmitted wrench becomes a function of the the robot's geometry and joint configuration. The same applies for the motion of

the links, mutually affecting each other, where \mathbf{R}_i can be utilized together with the translational part $\mathbf{s}_i = [a_i \quad -\sin(\alpha)d_i \quad \cos(\alpha)d_i]^\top$ of \mathbf{M}_i (see Figure Figure 3.2 on page 25) to relate the motion of the distally neighboring link $i + 1$ to the one of its predecessor link i :

$$\boldsymbol{\omega}_{i+1} = \mathbf{R}_i^\top \boldsymbol{\omega}_i + \dot{q}_{i+1} [0 \quad 0 \quad 1]^\top \quad (3.40)$$

$$\dot{\boldsymbol{\omega}}_{i+1} = \mathbf{R}_i^\top \dot{\boldsymbol{\omega}}_i + \mathbf{R}_i^\top \boldsymbol{\omega}_i \times \dot{q}_{i+1} [0 \quad 0 \quad 1]^\top + \ddot{q}_{i+1} [0 \quad 0 \quad 1]^\top \quad (3.41)$$

$$\dot{\mathbf{p}}_{i+1} = \mathbf{R}_i^\top [\dot{\boldsymbol{\omega}}_i \times \mathbf{s}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{s}_i) + \dot{\mathbf{p}}_i] . \quad (3.42)$$

Now, in order to transmit the wrenches from a distal link j over multiple intermediate links to a proximal link i , the wrench transmission operation \mathbf{T} decalred above can be applied sequentially to cover all of the link frame transformations occurring between links i and i , as in the following:

$$\mathbf{w}_{i,j} = \mathbf{T}_i \mathbf{T}_{i+1} \dots \mathbf{T}_j \mathbf{w}_{j,j} \quad (3.43)$$

$$\mathbf{w}_{i,j} = \mathbf{T}_i \mathbf{T}_{i+1} \dots \mathbf{T}_j \mathbf{F}_j \boldsymbol{\Phi}_j \quad (3.44)$$

$$\mathbf{w}_{i,j} = \mathbf{U}_{i,j} \boldsymbol{\Phi}_j . \quad (3.45)$$

Due to the sequential transmission of wrenches from distal to proximal links, and from proximal to distal links for the transmission of motion, the Equations (3.40) to (3.42) have to be computed recursively along the forward direction of the kinematic chain, and the Equation (3.43) has to be computed in the backward direction to obtain the values for every link. For this purpose the Recursive Newton Euler Algorithm has been developed (Luh, Walker, and Paul, 1980), essentially unifying the expressions presented so far within an algorithm that lends itself to be implemented with high computational efficiency. The details of implementation are discussed in Section 4.1 on page 46. The total wrench transmitted onto every link (see Equation (3.36) on the previous page) of a serial kinematic chain of arbitrary link count n can be obtained at once by using a linear equation system

$$\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \mathbf{w}_3 \\ \vdots \\ \mathbf{w}_n \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{1,1} & \mathbf{U}_{1,2} & \mathbf{U}_{1,3} & \dots & \mathbf{U}_{1,n} \\ \mathbf{0} & \mathbf{U}_{2,2} & \mathbf{U}_{2,3} & \dots & \mathbf{U}_{2,n} \\ \mathbf{0} & \mathbf{0} & \mathbf{U}_{3,3} & \dots & \mathbf{U}_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{U}_{n,n} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_1 \\ \boldsymbol{\Phi}_2 \\ \boldsymbol{\Phi}_3 \\ \vdots \\ \boldsymbol{\Phi}_n \end{bmatrix} \quad (3.46)$$

$$\mathbf{w} = \mathbf{U} \boldsymbol{\Phi} \quad (3.47)$$

$$\mathbf{w} = \mathbf{U}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi} \quad (3.48)$$

The functional expression $\mathbf{U}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k})$ emphasizes that the elements of \mathbf{U} are fully determined by the kinematic parameters \mathbf{k} and the kinematic state $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. The parameter matrix $\boldsymbol{\Phi}$ appears in vectorized form, where its columns pertaining to the respective links are stacked beginning with the first link's parameters on top. As the entries of \mathbf{w} and \mathbf{U} in this linear system are three-dimensional wrenches of which only the torques around the respective joint axes can be observed with uniaxial joint torque-sensing, the force and torque components of those wrenches are discarded except for the last rows of the torque

components, such that the final parametric model equation

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{RBD} = \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}, \quad (3.49)$$

with $\boldsymbol{\tau}_i = [0 \ 0 \ 0 \ 0 \ 0 \ 1] \cdot \mathbf{w}_i$ and $\mathbf{K}_{i,j} = [0 \ 0 \ 0 \ 0 \ 0 \ 1] \cdot \mathbf{U}_{i,j}$, is obtained, bearing close resemblance to the linear model equation initially defined in Equation (3.22) on page 26. The model matrix \mathbf{F} is represented by the kinematic observation matrix \mathbf{K} , which defines the kinematic state according to the Newton-Euler equations for one observation of the kinematic parameters. As the model assumptions encoded in the model matrix \mathbf{K} are all made within the framework of the equations of motion for rigid bodies, the model predictions made by this formulation are denoted as $\boldsymbol{\tau}_{RBD} = \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}$ when it is necessary to distinguish them from predictions made on the basis of different model assumptions.

For the purpose of dynamic identification of a robot, this linear equation can now be solved for $\boldsymbol{\Phi}$, given n measured joint torques to be inserted for $\boldsymbol{\tau}$ on the left-hand side of the equation. On the right-hand side, we find that \mathbf{K} is known as well by being essentially a function of the robot's kinematic parameters \mathbf{k} and the temporally differentiated joint position measurements $\mathbf{q}, \dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$. However, because the linear system is overdetermined as shown by the dimensions of the system's variables $\boldsymbol{\tau} \in \mathbb{R}^{n \times 1}$, $\mathbf{K} \in \mathbb{R}^{n \times 12n}$ and $\boldsymbol{\Phi} \in \mathbb{R}^{12n \times 1}$ and the measurements of the model inputs are further corrupted by noise, it is necessary to solve Equation (3.22) on page 26 for multiple input-output pairs of $\boldsymbol{\tau}$ and $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}$ to ensure statistical validity over a wide range of inputs for the model parameters identified by this linear regression.

Therefore, after the acquisition of N data pairs composed of dynamic state measurements of $\boldsymbol{\tau}_{meas}$ and kinematic state measurements ($\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k}$), a model equation is constructed for each observation, which are then stacked, such that the stacked linear equation system

$$\bar{\boldsymbol{\tau}}_{meas} = \bar{\mathbf{K}}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k}) \bar{\boldsymbol{\Phi}} \quad (3.50)$$

$$\begin{bmatrix} \boldsymbol{\tau}_{meas}^1 \\ \boldsymbol{\tau}_{meas}^2 \\ \vdots \\ \boldsymbol{\tau}_{meas}^N \end{bmatrix} = \begin{bmatrix} \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^1 \\ \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^2 \\ \vdots \\ \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^N \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi} \\ \vdots \\ \boldsymbol{\Phi} \end{bmatrix}, \quad (3.51)$$

where $\bar{\boldsymbol{\tau}}_{meas} \in \mathbb{R}^{N \cdot n \times 1}$, $\bar{\mathbf{K}}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k}) \in \mathbb{R}^{N \cdot n \times 12n}$ and $\bar{\boldsymbol{\Phi}} \in \mathbb{R}^{N \cdot 12n \times 1}$ is obtained.

3.4 Artificial Neural Networks for Dynamic Model Identification

Although models based on neural networks, strictly speaking, still constitute parametric models in the narrow sense, they can be regarded to effectively behave in a fashion similar to proper non-parametric models, due to the, usually high parameter count and absence of specifically defined, rigid parameter relationships (Lee et al., 2018). In fact, the property of approximating non-linearity can be seen as a result of the indefinitely extendable parameter count of an artificial neural network. This low degree of parameterization, i.e. dependence

on predefined semantic model structure, allows for a model identification purely based on dynamic measurement data without imparting of any prior model knowledge in terms of an analytical decomposition of the dynamics into interrelated physical effects and quantities as is the case with parametric modeling. While this naturally entails the risk of failing to extract an adequate dynamic model representation from data alone, at the same time, the opportunity to capture phenomena outside of the preset borders of dynamic model formulations arises. In view of the complexity of robot dynamics itself on one hand and of the multitude of perturbing effects on the sensory observation of it, it appears desirable to investigate more flexible model identification techniques, such as artificial neural networks. Given the vastness of the field of neural network-based machine learning techniques, the following is intended as a concise overview of the architectural and functional fundamentals with a focus on the application of dynamic identification and modeling. In awareness of the multitude of developed variations and different flavors regarding the design and use of the techniques to be discussed, primarily, the architectural and functional design choices investigated in this thesis are considered to be succinct.

Leaving the fundamental task defined in Equation (3.21) on page 26 unchanged, the approximation of the true robot dynamics does not necessarily require an underlying model formulation that is derived from physical insight. Instead of incorporating hand-crafted functional, possibly nonlinear inter-parametric relationships (see Newton-Euler formulae in Equations (3.23) and (3.24) on page 26) in hope of representing the majority of dynamic effects, an arbitrarily complex parametric relationship can be distilled from observations of the dynamic behavior by courtesy of the functional properties of certain types of artificial neural networks.

Multilayer Perceptrons

Multilayer perceptrons (MLP's) arguably constitute the archetypal neural network models to be utilized for function approximation purposes. As such they can also be applied on the same fundamental task defined in Equation (3.22) on page 26 to define a mapping $\boldsymbol{\tau} = f(\boldsymbol{x}, \boldsymbol{\theta})$ between observations of the joint torques $\boldsymbol{\tau}$ and other available sensory inputs \boldsymbol{x} within a supervised regression framework, wherein the joint torque measurements represent the supervision signal, i.e. *label* data. This mapping is again established via a collection of parameters, here denoted as $\boldsymbol{\theta}$ to avoid ambiguity with regards to the semantically defined parameters from before.

In accordance with associated standard theory, the mapping is composed of multiple interconnected chains of parallel functional units or *neurons* arranged in a network, where the number and connectivity of these units is dictated by a layered corresponding directed acyclic graph that is $m^{(l)}$ units wide and $l + 2$ unit layers deep.² The first layer, also known as input layer, accepts the inputs of the dynamic model consisting of an vector $\boldsymbol{x} \in \mathbb{R}^{n \cdot p \times 1}$ containing an arbitrary set of p observable quantities (see Section 3.1 on page 19), i.e. features, for every joint $j = 1, \dots, n$, most commonly in form of a concatenation of the

²For the sake of completeness, it should be pointed out that the overall topology and functional behavior of artificial neural networks are inspired by neuroscientific observations of biological nervous systems (McCulloch and Pitts, 1943), which is also why the associated terminology is pervaded by biological metaphors.

3 Force Estimation through Dynamic Identification

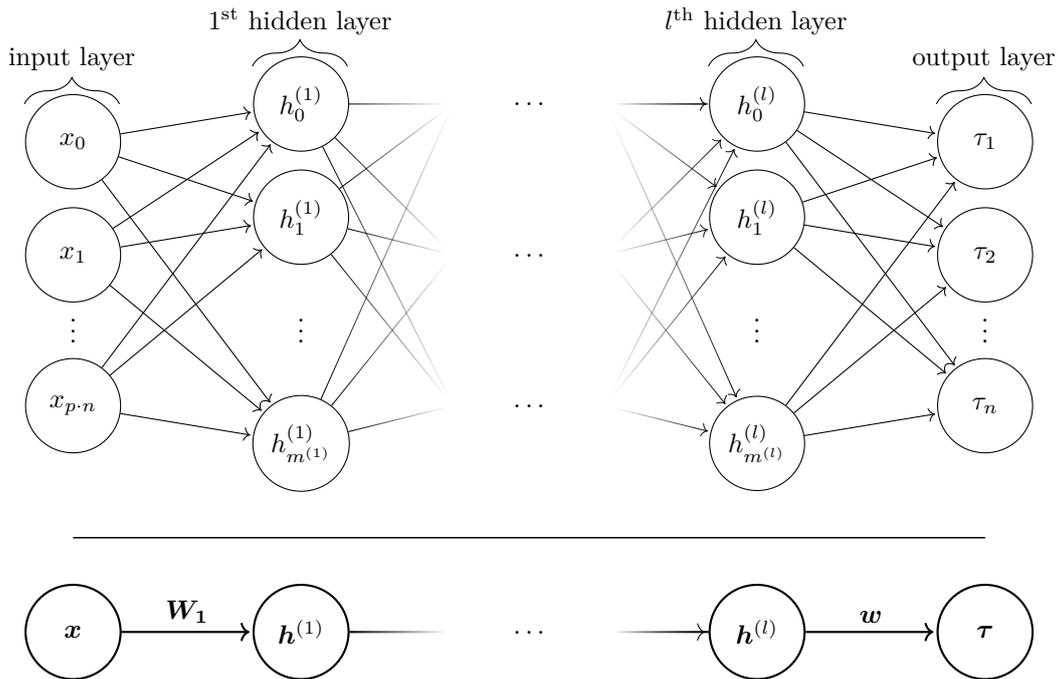


Figure 3.3: Network graph of a $(l + 1)$ -layer perceptron with $p \cdot n$ input and n output units. The l^{th} hidden layer contains $m^{(l)}$ hidden units.

usual joint-wise kinematic parameters plus any other observable quantity $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$, such that $\mathbf{x} = \text{vec}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star)$. The input is fed forward through the network over multiple paths in l hidden layers towards the n units in the output layer corresponding to the target vector $\boldsymbol{\tau} \in \mathbb{R}^n$. Along this path consisting of a sequence of connected units, the inputs are subjected to a chain of elementary operations occurring at the respective units as illustrated by Figure 3.3.

The layer-wise operations up until the output layer are collectively defined as

$$\mathbf{h}^{(1)} = f^{(1)}(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{b}^{(1)}) = a(\mathbf{W}^{(1)\top} \mathbf{x} + \mathbf{b}^{(1)}) \quad (3.52)$$

$$\mathbf{h}^{(2)} = f^{(2)}(\mathbf{h}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}) = a(\mathbf{W}^{(2)\top} \mathbf{h}^{(1)} + \mathbf{b}^{(2)}) \quad (3.53)$$

\vdots

$$\mathbf{h}^{(l)} = f^{(l)}(\mathbf{h}^{(l-1)}, \mathbf{W}^{(l)}, \mathbf{b}^{(l)}) = a(\mathbf{W}^{(l)\top} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}), \quad (3.54)$$

where $\mathbf{h}^{(1)}$ denotes the vector of the state of the units in the first hidden layer as the result of the application of the function $f^{(1)}(\mathbf{x}, \mathbf{W}^{(1)}, \mathbf{b}^{(1)})$ on the input vector \mathbf{x} ; $\mathbf{h}^{(l)}$ denotes the units' values in hidden layer l due to the application of $f^{(l)}(\mathbf{h}^{(l-1)}, \mathbf{W}^{(l)}, \mathbf{b}^{(l)})$ on the previous hidden layer $\mathbf{h}^{(l-1)}$. The parameters of those respective layerwise transformations $f^{(l)}$ are the weight matrix $\mathbf{W}^{(l)}$, which defines the scaling factors to be applied on the output values of the previous layer's units when they are fed forward to a particular unit of the following layer, and the bias vector $\mathbf{b}^{(l)}$ denoting an additional offset value to be added to the unit. As these two parameters define affine transformations on the respective layer inputs that are incapable of approximating non-linear relations present in the data, an *activation function* a

is subsequently applied on every hidden unit to introduce a non-linearity into the network. Otherwise, a sequence of linear transformations could be expressed with a single affine transformation in one layer anyway, defying the need for multiple layers. Here, the *rectified linear unit* activation function

$$a(z) = \max(0, z) \quad (3.55)$$

is a common choice (Nair and Hinton, 2010) because of its piecewise linearity still preserves some desirable properties of linear models with regards to their ease of optimization and good generalization capabilities (Goodfellow, Bengio, and Courville, 2016). Within the

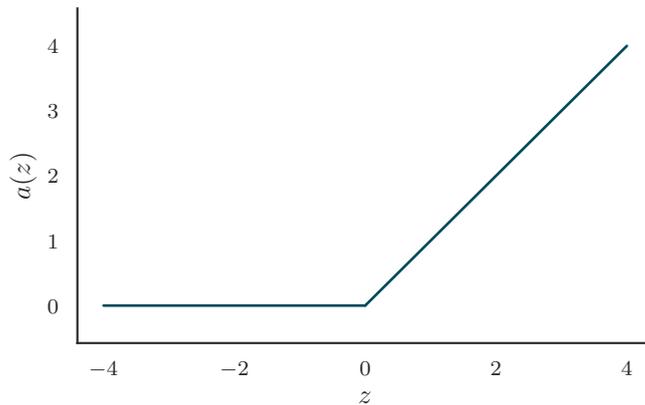


Figure 3.4: Plot of the rectified linear unit activation function (ReLU).

context of regression, after sequentially performing these operations for every layer, the values of the last hidden layer are then again fed into a purely linear function to avoid non-linear distortions in the output:

$$\boldsymbol{\tau} = f^{(l+1)}(\mathbf{h}^{(l)}, \mathbf{W}^{(l+1)}, \mathbf{b}^{(l+1)}) = \mathbf{W}^{(l+1)\top} \mathbf{h}^{(l)} + \mathbf{b}^{(l+1)} \quad (3.56)$$

The totality of the network, and thus the dynamic model function, can be expressed as

$$\boldsymbol{\tau} = \mathbf{W}^{(l+1)\top} \left[f^{(l)} \circ f^{(l-1)} \circ \dots \circ f^{(1)} \right] + \mathbf{b}^{l+1} \quad (3.57)$$

$$\begin{aligned} &= \mathbf{W}^{(l+1)\top} \left[a(\mathbf{W}^{(l)\top} \mathbf{h}^{(l)} + \mathbf{b}^l) \circ a(\mathbf{W}^{(l-1)\top} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l-1)}) \circ \dots \right. \\ &\quad \left. \circ a(\mathbf{W}^{(1)\top} \mathbf{x} + \mathbf{b}^{(1)}) \right] + \mathbf{b}^{(l+1)} \end{aligned} \quad (3.58)$$

$$= f(\mathbf{x}, \mathbf{W}, \mathbf{b}) = f((\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \star), \mathbf{W}, \mathbf{b}), \quad (3.59)$$

where the torque predictions of the model for a given observation now only depend on the weights \mathbf{W} and biases \mathbf{b} of the respective neurons within a defined network topology of depth l and layerwise width $m^{(l)}$, which constitute the model parameters to be identified, or *learned*, when applying the technical terminology common in the related research field. By courtesy of the nonlinearity introduced through the activation function, it is then eventually

possible to define a topology of sufficient width or depth, depending on the complexity of the real dynamics, which is in theory capable of fully approximating that true underlying dynamics. This is a reduced formulation of the fundamental statement of the *theorem* (Cybenko, 1989; Hornik, Stinchcombe, and White, 1989). As a sufficient topology cannot be determined exactly in advance, one is left to rely on existing experience, experimentation, and other heuristics.

Please note, however, that the universal approximation capability is guaranteed only in terms of the model structure, i.e. in the sense that MLP's provide the necessary framework to represent any function with the condition that its corresponding model parameters are known. In combination with the generic or non-task-specific structure of an MLP, in contrast to the physical parametric formulation from before, the number of model parameters usually is drastically increased to capture the observed dynamic effects. Therefore, the regression task for the identification of the model parameters is complicated by an increase in the required amount of training data and an aggravation of the optimization of the model parameters. Essentially, this results in a trade-off between the enhancement of a network's representative potential through an increase in model size on hand and the difficulty of parameter tuning, i.e. model *training*, on the other.

Regardless of the specific network topology, the goal of the training process is to adjust the network parameters $\boldsymbol{\theta} = (\mathbf{W}, \mathbf{b})$ such that a network $f(\mathbf{x}, \mathbf{W}, \mathbf{b})$ is obtained that outputs torque predictions $\boldsymbol{\tau}_{est}$ that minimize the *loss function* $L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas})$, i.e. the scalar prediction error, with respect to the measured target torques $\boldsymbol{\tau}_{meas}$ resulting from the true dynamics, which is commonly defined as the mean squared error over i samples

$$L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas}) = \text{MSE} = \frac{1}{m} \sum_i^m \|(\boldsymbol{\tau}_{est} - \boldsymbol{\tau}_{meas})_i\|^2. \quad (3.60)$$

In order to optimize the parameters to yield minimum loss, the prediction loss of network function $f(\mathbf{x}, \mathbf{W}, \mathbf{b})$ has to be evaluated with respect to its optimization parameters $\boldsymbol{\theta}$. This process is conceptually divided into the two steps of *forward* and *back-propagation*. Following a random initialization of the network parameters, the inputs \mathbf{x}_i are fed into the network and propagated forward through the layers, producing the output torques for which the loss $L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas})$ is computed. The influence of the parameters of the network on the prediction accuracy is then retrieved by the back-propagation of the loss through the network to compute the gradient of the loss with respect to the network parameters $\nabla_{\boldsymbol{\theta}} J$, where the cost $J = L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas})$ is introduced as a shorthand for the loss. As J is the result of a chain of elementary operations defined by both the computational graph (see Figure 3.3 on page 32) and the Equations (3.57) to (3.59) on the preceding page, the initially unknown total gradient $\nabla_{\boldsymbol{\theta}} J$ likewise can be represented as a composition of separate gradients, as stated by the chain rule of calculus. A minimal example can be construed by letting $\mathbf{y} = g(\mathbf{x})$ be a function that maps from \mathbb{R}^m to \mathbb{R}^n and $z = f(\mathbf{y})$ be a function that maps from \mathbb{R}^n to \mathbb{R} . According to these relations, z can also be represented as a composition $z = f(\mathbf{y}) \circ g(\mathbf{x})$, where the gradient $\frac{\partial z}{\partial x_i}$ can be written as a product of

two gradients as in

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j} \frac{\partial y_j}{\partial x_i} \quad (3.61)$$

$$\nabla_{\mathbf{x}} z = \left(\frac{\partial \mathbf{y}}{\partial \mathbf{x}} \right)^\top \nabla_{\mathbf{y}} z. \quad (3.62)$$

This basic rule can then be applied to recursively express $\nabla_{\boldsymbol{\theta}} J$ as a multiplicative chain of the gradients of the operations occurring in the network from the output layer towards the input layer using the back-propagation algorithm shown in Algorithm 3.5. Thus, the computed gradient is utilized in conjunction with an optimization algorithm belonging to the family of iterative gradient descent algorithms, which eponymously leverage the multidimensional direction along which the parameters have to be changed to reduce the loss, as the extremely high dimensionality of the network renders a direct optimization intractable.

Algorithm 3.5: Computation of the gradient $\nabla_{\boldsymbol{\theta}} J$ for a MLP with l hidden layers. After forward propagation of the input, $\nabla_{\boldsymbol{\theta}} J$ is determined by recursively extending a gradient expression with the gradients of the layer-wise operations by courtesy of the chain rule while back-propagating towards the input layer (Rumelhart, Hinton, and Williams, 1986). Adapted from (Goodfellow, Bengio, and Courville, 2016).

algorithm back-propagation

inputs: \mathbf{x} , $\boldsymbol{\tau}_{meas}$, \mathbf{W} , \mathbf{b} , l

outputs: J , $\nabla_{\boldsymbol{\theta}} J$

// forward propagation

$\mathbf{h}^{(0)} = \mathbf{x}$

for $k = 1, \dots, l$ **do**

$\mathbf{a}^{(k)} = \mathbf{W}^{(k)} \mathbf{h}^{(k-1)} + \mathbf{b}^{(k)}$

$\mathbf{h}^{(k)} = f(\mathbf{a}^{(k)})$

end for

$\boldsymbol{\tau}_{est} = \mathbf{h}^{(l)}$

$J = L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas})$

// back-propagation

compute gradient on *output*: $\mathbf{g} \leftarrow \nabla_{\boldsymbol{\tau}_{est}} J = \nabla_{\boldsymbol{\tau}_{est}} L(\boldsymbol{\tau}_{est}, \boldsymbol{\tau}_{meas})$

for $k = l, l-1, \dots, 1$ **do**

compute gradient on activations: $\mathbf{g} \leftarrow \nabla_{\mathbf{a}^{(k)}} J = \mathbf{g} \odot f'(\mathbf{a}^{(k)})$

compute gradient on biases: $\nabla_{\mathbf{b}^{(k)}} J = \mathbf{g}$

compute gradient on weights: $\nabla_{\mathbf{W}^{(k)}} J = \mathbf{g} \mathbf{h}^{(k-1)\top}$

compute gradient on total layer: $\mathbf{g} \leftarrow \nabla_{\mathbf{h}^{(k-1)}} J = \mathbf{W}^{(k)\top} \mathbf{g}$

end for

$\nabla_{\boldsymbol{\theta}} J \leftarrow \mathbf{g}$

return J , $\nabla_{\boldsymbol{\theta}} J$

Algorithm 3.6: Stochastic gradient descent for one training iteration k . Adapted from (Goodfellow, Bengio, and Courville, 2016).

```

algorithm stochastic gradient descent
inputs:  $\epsilon_k, \boldsymbol{\theta}$ 
output:  $\boldsymbol{\theta}$ 

while stopping criterion not met do
    get minibatch with  $m$  input-label pairs:  $\{(\mathbf{x}^1, \boldsymbol{\tau}_{meas}^1), \dots, (\mathbf{x}^m, \boldsymbol{\tau}_{meas}^m)\}$ 
    back-propagation over minibatch:  $\mathbf{g} \leftarrow \frac{1}{m} \nabla_{\boldsymbol{\theta}} \sum_i^m L(f(\mathbf{x}^{(i)}, \boldsymbol{\theta}), \boldsymbol{\tau}_{meas}^{(i)})$ 
    update parameters:  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \mathbf{g}$ 
end while
return  $\boldsymbol{\theta}$ 

```

As a consequence of the inherent model flexibility and a high number of parameters, there is a need to optimize the network parameters, i.e. train the network on a large data set including as many samples as to sufficiently resemble the data encountered in practice and ensure good generalization of the model. One may argue, that it would be beneficial to then to sequentially compute the loss gradient and adjust the network parameters accordingly for every training sample. The sheer size of the training set in almost all cases, however, prevents the repeated computation of the gradient for every sample due to the infeasibly high computational time requirements. Therefore the algorithmic foundation for the adjustment of the network parameters is almost exclusively derived from the basic algorithm of *stochastic gradient descent* (see Algorithm 3.6), which makes use of an estimate of the loss gradient based on a randomly sampled subset of the training set. The network parameters are then updated according to this estimate, where a preset learning rate ϵ dictates step size with which the parameters are adjusted descent direction. In order to realistically assess the benefit gained from the parameter updates, and especially its generalization capability, the available data are split into a training set on the data of which the network parameters are optimized, and a validation set which is solely used to obtain an unbiased evaluation of the network's prediction accuracy on unseen data.

Recurrent Neural Networks

As the defining property of any dynamic system like a robot is the time dependence of its state, there is a natural incentive to incorporate temporal information within the general framework of artificial neural networks for dynamic modeling. The network architecture of conventional feedforward MLP's presented so far, however, is designed to learn on data containing single, temporally disjunct relations between an input and target output. Although in the case of robot dynamic modeling, some temporal effects are already embedded in the common input quantities that consist of temporal derivatives, the opportunity to encode information on dynamic effects that emerge as the result of a longer history of system and motion states, which may e.g. be grouped under the term of *hysteresis*, is still neglected.

In order to leverage the temporal information for dynamic modeling, the computational graph of a feedforward network such as the MLP (see Figure 3.3 on page 32) can be modified to form a cyclic connection between its output and input, transforming it into a type of

recurrent neural network (RNN), whereof many architectural variants exist (Elman, 1990; Lang, Waibel, and Hinton, 1990; Hochreiter and Schmidhuber, 1997). The underlying idea is to abstract a dynamic system as a time series of system states of length T , for which a model can be approximated by a recurrent network function accepting a new input $\mathbf{x}^{(t)}$ at every time step t :

$$\mathbf{h}^{(t)} = f(\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}, \boldsymbol{\theta}), \quad (3.63)$$

where in case of a simple recurrent networks (SRN), the hidden states \mathbf{h} functions may be defined as

$$\mathbf{h}^{(t)} = \tanh(\mathbf{W}\mathbf{h}^{(t-1)} + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{b}) \quad (3.64)$$

$$\mathbf{o}^{(t)} = \mathbf{c} + \mathbf{V}\mathbf{h}^{(t)}. \quad (3.65)$$

Here, $\mathbf{o}^{(t)}$ denotes the output of the SRN at time step t that in our case constitutes the torque prediction $\boldsymbol{\tau}_{est}$ for that particular time step based on which the prediction loss with regards to the label $\boldsymbol{\tau}_{meas}$ will be determined. The recurrence extends the conventional neural network function in Equation (3.59) on page 33 to also refer back to the inputs and states of the hidden units of a previous forward pass. As such, the network function of an RNN is able to encode a complete input *history* into its hidden units and essentially build a *memory* of past inputs based on which model predictions can be made. By means of this added recurrence, the universal approximation theorem for feedforward networks can similarly be expanded to imply the capability to approximate any time-sequence mapping with a sufficient number of hidden units (Hammer, 2000). The training of an RNN is only marginally different from that of a feedforward network, as RNN's can be considered equivalent to deep feedforward networks when unfolded in time, with the only distinction being the cyclic repetition of the network parameters. Similarly to feedforward networks, the gradients are computed by back-propagation through the time-unfolded network, which commonly is referred to as *back-propagation through time* (BPTT) (Williams and Zipser, 1995)). However, depending on whether an RNN is employed to map a sequence to a single time-step output or another sequence, a single loss gradient or multiple loss gradients for each prediction within a defined time window must be back-propagated to determine the total gradient (*truncated back-propagation through time*). Regardless of the specific design of the inner workings of a RNN, the recurrent execution of a network function inevitably restricts the choice of nonlinear activations to *squashing* functions, such as the sigmoid logistic function $\sigma(x) = \frac{1}{1+e^{-x}} \rightarrow (0, 1)$ or the hyperbolic tangent $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \rightarrow (-1, 1)$, to regularize the returned intermediary hidden states (*exploding gradients problem*). This has a negative impact on the training process which manifests itself in the computation of the loss gradient: When unfolding the network over multiple time steps, due to the repeated execution of the network function during the successive forward propagation of an input sequence, the gradient of the loss becomes the product of a long chain of back-propagated gradients (*back-propagation through time*, that are ≤ 1). Therefore, with increasing sequence length, i.e. distance to the first layers, the gradients tend to exponentially decrease. Among other reasons, this issue, commonly known as the *vanishing gradient problem* (see Figure 3.8 on page 39), leads to a significant stagnation of the parameter optimization via stochastic gradient descent for long sequences as well as difficulties with the learning of long-range

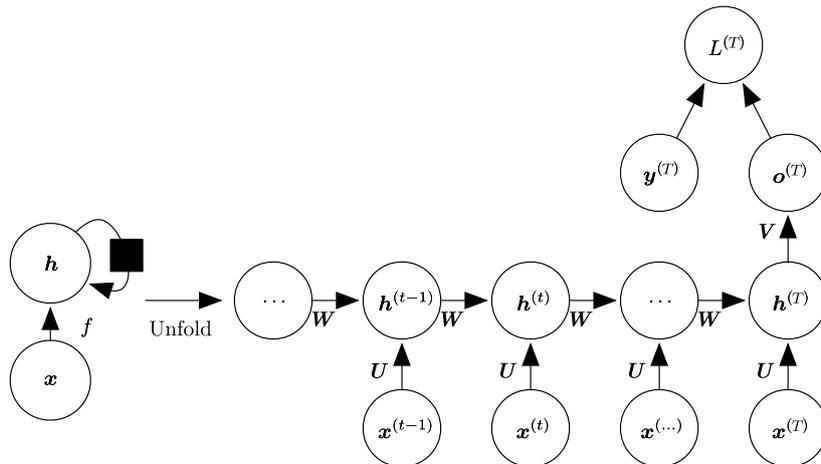


Figure 3.7: A time-unfolded computational graph for an RNN that maps a sequence of inputs onto single output. For every time step t an input $\mathbf{x}^{(t)}$ weighted by \mathbf{U} is fed into the hidden unit $\mathbf{h}^{(t)}$ together with the output of the previous hidden unit $\mathbf{h}^{(t-1)}$ weighted by \mathbf{W} . At the end of the input sequence, the last hidden state undergoes an affine transformation parameterized by \mathbf{V} into the final output vector \mathbf{o} based on which the loss \mathbf{L} with respect to a label vector for that time step is calculated. This type of configuration is a natural choice for the purpose of dynamic modeling, as all information of contained within the sequence can be condensed in one prediction, which is in contrast to sequence-to-sequence mappings, where earlier hidden states experience a reduced history of inputs. Adapted from (Goodfellow, Bengio, and Courville, 2016)

dependencies (Hochreiter et al., 2001), which is especially detrimental for applications such as the dynamic modeling of robots.

This problem inherent to RNN's is alleviated by gated RNN's such as the *long short-term memory* (LSTM) architecture (Hochreiter and Schmidhuber, 1997), which nowadays constitutes the most prominently utilized RNN type. Inside of an LSTM block, multiple gates, that are controlled by learnable parameters, dictate the passage and transformation of the incoming inputs and recurrently connected outputs. This allows an adaptive regulation of the information flow during a input sequence in order to preserve long-term contextual information. A canonical configuration of an LSTM cell (Sak, Senior, and Beaufays, 2014) depicted in Figure 3.9 on page 40, commonly accepts an input vector $\mathbf{x}^{(t)} \in \mathbb{R}^{n \times 1}$ and recurrently fed-back hidden state vector $\mathbf{h}^{(t-1)} \in \mathbb{R}^{m \times 1}$, of which four copies each are pairwise directed towards four non-linearly activated gates respectively, i.e. an input-gate \mathbf{i} , forget-gate \mathbf{f} (Gers, Schmidhuber, and Cummins, 1999), cell gate $\tilde{\mathbf{c}}$ and output gate \mathbf{o} . These output gate determines the output hidden state $\mathbf{h}^{(t)}$ based on an recurrently fed-back, auxiliary cell state $\mathbf{c}^{(t-1)}$ that is computed in parallel by the forget, input and cell

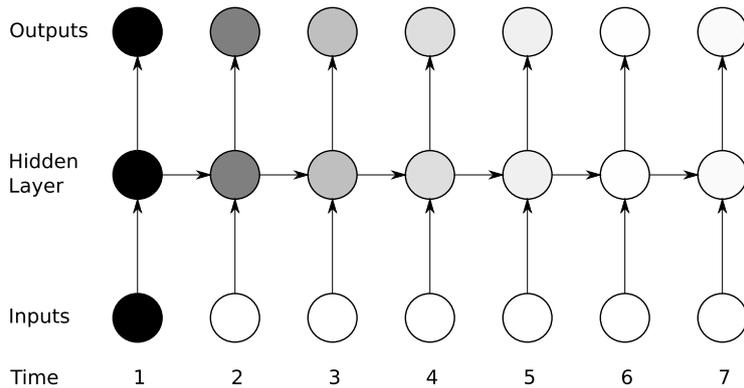


Figure 3.8: The vanishing gradient problem. The shading indicates the decrease of the gradient towards the end of an input sequence. From (Graves, 2012)

states according to following formal definition:

$$\mathbf{i}^{(t)} = \sigma \left(\mathbf{W}_{xi} \mathbf{x}^{(t)} + \mathbf{W}_{hi} \mathbf{h}^{(t-1)} + \mathbf{b}_i \right) \quad (3.66)$$

$$\mathbf{f}^{(t)} = \sigma \left(\mathbf{W}_{xf} \mathbf{x}^{(t)} + \mathbf{W}_{hf} \mathbf{h}^{(t-1)} + \mathbf{b}_f \right) \quad (3.67)$$

$$\tilde{\mathbf{c}}^{(t)} = \tanh \left(\mathbf{W}_{xc} \mathbf{x}^{(t)} + \mathbf{W}_{hc} \mathbf{h}^{(t-1)} + \mathbf{b}_c \right) \quad (3.68)$$

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} \quad (3.69)$$

$$\mathbf{o}^{(t)} = \sigma \left(\mathbf{W}_{xo} \mathbf{x}^{(t)} + \mathbf{W}_{ho} \mathbf{h}^{(t-1)} + \mathbf{b}_o \right) \quad (3.70)$$

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \sigma \left(\mathbf{c}^{(t)} \right) . \quad (3.71)$$

The diminished susceptibility of LSTM's against gradient instabilities stems from the cell state's general independence from the direct influence of adaptable parameters. Because this cell state is fairly consistent over time and is intimately involved in the computation of the subsequent hidden state, i.e. output, the information contained in past time steps is preserved and can exert significant influence on the current predictions. The negative implication following from this, i.e. that possibly irrelevant historic information is conserved is mitigated by the learnable forget gate (Gers, Schmidhuber, and Cummins, 1999), which can eponymously forget, i.e. reset, the cell state if necessary to remove the perhaps disadvantageous influence of past inputs.

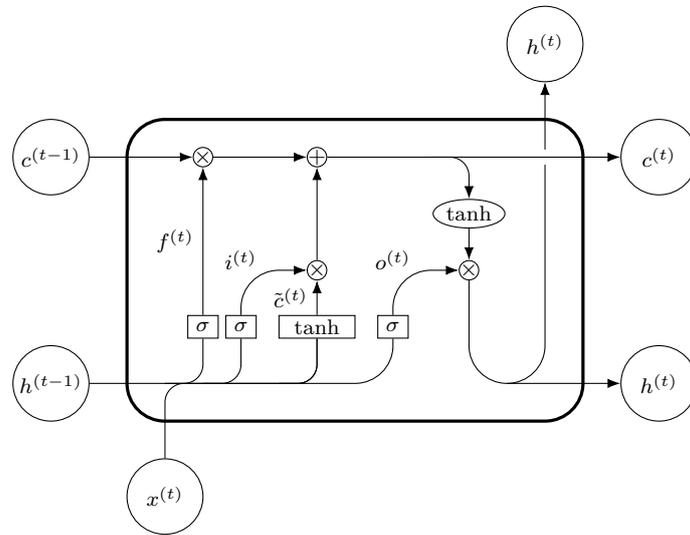


Figure 3.9: The internal structure of a common configuration of an LSTM cell. In contrast to an SRN, the input $x^{(t)}$ is not solely concatenated with the previous hidden state $h^{(t-1)}$ and fed through a non-linearity. The processing of $x^{(t)}$ is further orchestrated by a separate recurrently connected, linearly activated cell state $c^{(t)}$ and three multiplicative non-linear gates.

4

Hybrid Models for Torque Estimation

Generally speaking, techniques for dynamic modeling may be classified according to their respective level of semantic parameterization and incorporation of a priori model knowledge (see Section 3.2 on page 21) as either *white-* or *black-box* models. The benefits and drawbacks of the respective modeling approaches alluded to in Section 3.2 on page 21, become apparent in view of the dynamic complexity of robots. On one hand, the complexity manifests itself phenomenologically in a high number of interacting effects and influences pertaining to the dynamics of link bodies and joints, and on the other, in the high dimensionality of the dynamic system reflected in the multiplicity of mechatronic components and operational parameters.

Utilizing a white-box approach by tracing the observable dynamic phenomena back to a succinct, analytical expression describing the interplay of physically meaningful quantities, e.g. the Newton-Euler equations of motion as described in Section 3.3 on page 24, has the desirable advantage of providing a model regression framework that is exclusively rooted in fundamentally verifiable, compact physical laws. Accordingly, the model predictions are explainable, guaranteed to generalize beyond the regression data set, and sample-efficient. The main disadvantage, however, originates from the limited expressiveness of the used model formulation. Although the mutually equivalent formulations of classical mechanics provide the necessary theoretic basis to macroscopically describe any mechanical system, the conception of a model expression that augments the Newtonian or Lagrangian equations of motion for rigid bodies to accommodate for every additional significant dynamic effect, e.g. due to joint flexibility, dissipative forces or mechanic hysteresis, is, depending on the intended model accuracy, a non-trivial or even impossible task. Inevitably, this leads to prediction errors due to model bias, as unmodeled effects exceeding any prior model assumptions confound the model output.

The black-box approach, e.g. via universal function approximators such as artificial neural networks, naturally mitigates the problems related to expressiveness and the design of sufficiently representative model formulations. As discussed in Section 3.4 on page 30, this is achieved by a high amount of semantically weakly defined model parameters that are interrelated through a generic network structure, which in theory allows for the replication of the true dynamics with arbitrary fidelity, given sufficient training data and topological complexity. The great advantage of potentially unbound model expressiveness is traded with a lack of model explainability, lower sample efficiency, and a limited generalization

capability resulting in the risk of prediction errors due to model variance.

Hence, we face the recurring challenge in model characterization tasks of whether to choose a white-or black-box model approach, i.e. the fundamental choice to be made revolves around the question of whether one can tractably incorporate the entirety of dynamic and perturbing effects into a closed model formulation. While the white-box approach of the parametric model identification scheme (see Section 3.3 on page 24) can certainly be extended to identify a multitude of parameters describing additional effects, e.g. through various non-linear friction and hysteresis model identification techniques (Ruderman and Bertram, 2012; Kircanski and Goldenberg, 1997), this is often concomitant with the laborious design and execution of a series of statistical parameter excitation experiments along with the associated regression process for each actuator, as well as the constant possibility of neglected model components (see Section 3.3 on page 24). An alternative approach may be to exploit the universal function approximation capabilities of neural network-based techniques, and shift the problem towards the design of suitable network architecture and the acquisition of representative training data (see Section 3.4 on page 30).

In an effort to balance the individual advantages and disadvantages of the paradigms of white- and black-box modeling, a gray-box approach was taken for the model-based torque prediction technique developed as part of this thesis. With the aim to adhere to customary argumentative order, the developed methodology is presented in the following before the evaluation, albeit it is worth mentioning that the general idea and conceptional phase of the development of this method was intimately related with the preliminary experimentation with and evaluation of the classic parametric and neural network-based techniques described in Sections 3.3 and 3.4 on page 24 and on page 30. Consequently, the observation of the strengths and weaknesses of the respective approaches inspired the very design choices made with respect to the taken approach.

The rationale behind the utilization of a hybrid model approach can be illustrated intuitively by examination of the expected properties of the motion to be executed by the robot. As mentioned in the introductory remarks, the thematic focus of this thesis lies on the model-based estimation of external forces during continuous contact human-robot interaction tasks, especially in the medical field for the robotic replication of physically interactive motions that are conventionally carried out by a healthcare professional on the patient anatomy. The defining characteristics of robotic motion that are intended for these purposes are a comparatively low dynamic range and intensity as well as an inherent irregularity of motion direction. Primarily, this is due to safety reasons and functional requirements with regard to robotic motion in such tasks. During continuous contact of the robot with the human body, the occurring interaction forces have to be precisely regulated to compensate for human counter motion and maintain desired interaction force levels while reaching a task-specific goal. This often results in a motion sequence composed of multiple intermittent fine-scale positional changes and frequent alternations of the motion direction, i.e. when performing heuristic, explorative motions, reacting to external physical interactions, or manipulating anatomic target regions. With regards to the applicability of different dynamic modeling techniques, these motion properties have several implications.

During the execution of motion sequences that incorporate comparatively higher dynamic intensity, i.e. greater joint velocities and accelerations, and consistency of move-

ment direction over longer kinematic distances³ the induction of measurement signals in the torque sensors can be assumed to be dominated by proper dynamic effects pertaining to rigid body dynamics caused by the acceleration of the link inertias resulting in gravitational and motion-dependent torques. Due to the structural location of the sensory elements on the joint output side, the influence of motor friction and flexibility is greatly reduced. A relative minority can be attributed to joint velocity-dependent vibrational disturbances originating from the strain wave gearing transmission (Chawda and Niemeyer, 2017), joint flexibilities, and sensor noise.

In the case of low velocities, accelerations, and the operation in the vicinity of a quasi-static state, however, the rigid body dynamics simplifies to primarily gravitational and frictional influences. If the motion is further composed of small positional but frequent directional changes, there is a relative increase in torque signal contribution of complex phenomena, which are conjectured to originate from compound mechanic effects that normally contribute linearly to the dynamics when they are saturated. For instance, the torque contributions due to joint friction exhibit a pronounced nonlinearity around joint velocities close to zero (Kircanski and Goldenberg, 1997) and at the reversal of motion direction (Swevers et al., 2000), which cannot only be approximated by parametric joint friction models such as the viscous Coulomb-friction (Khalil and Dombre, 2002) or Stribeck friction model (Olsson et al., 1998) with limited accuracy.

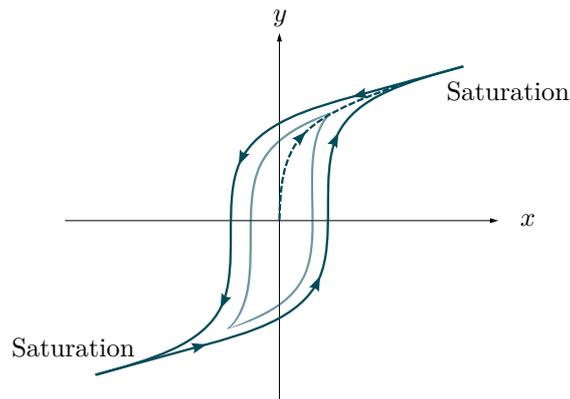


Figure 4.1: A generic hysteresis loop that can be associated with the dependency of an arbitrary state variable y upon the histories of both the past input values \mathbf{x}_{hist} and state values \mathbf{y}_{hist} .

Similarly, effects pertaining to mechanic hysteresis (see Chapter 4) primarily caused joint flexibility, e.g. mechanical loss of motion due to transmission backlash (Ruderman and Bertram, 2012) or torque sensor hysteresis due to rotational strain gauge elasticity (Allgeier and Evans, 1995), further exert nonlinear influences complicating the dynamic model in case of frequent changes of joint rotation direction. A descriptive example for the occurrence of hysteresis in robot dynamics is the change of measurable joint torques $\boldsymbol{\tau}$ in relation to

³Motions occurring during rapid pick-and-place task, for example, could be considered to fall under the category of highly dynamic motions with frequent directional changes. However, the kinematic distance, within which the directional changes take place, is quite large in comparison to fine-scale motions that are the main focus of this thesis.

the history of joint positions \mathbf{q} as has been observed during the experimentation with the KUKA iiwa 14 for this thesis (see Section 2.1 on page 12). For a sequence history of S joint position inputs $\mathbf{q}_j^{hist} = \{q_j^1, \dots, q_j^S\}$ for joint j where the respective changes between successive joint positions $\Delta q_j^s = q_j^s - q_j^{s-1}$ have fairly consistent sign and the net change over the total sequence $\Delta q_j^S = q_j^S - q_j^1$ is sufficient large, the hysteretic behavior of the measured joint torque τ_j may saturate and transition into an approximately linear dependence of τ_j on the respective current value of q_j subsequent to \mathbf{q}_j^{hist} . In the inverse case however, i.e. when $\text{sign}(\Delta q_j^s)$ fluctuates due to changes in rotational direction, hence Δq_j^S as the net total rotational displacement with respect to the joint position at sequence start remains small. Thereby the joint torque transitions along a multitude of minor loop paths depending on the specific sequence history and properties of the actual underlying hysteretic behavior, which can be approximate e.g. by a so called *Preisach-Model* (Ruderman and Bertram, 2012).

Mechanically, the hysteresis of the measured joint torques with respect to the joint position history appears to be consistent with effects of plastic deformation or backlash, where following causal mechanisms are proposed: Assuming that some mechanical components within a joint are subject to plastic deformations due to a torsional stress transmitted through friction between the parts rotating in one direction and reach a maximum deformation due to strain hardening (Ruderman and Bertram, 2012), then a part of the torque generated for the counter rotation is spent on the back-torsion of those components. A similarly hypothesis can be made with regards to backlash. A small bidirectional rotational play between the movable components is held constant due to internal friction until a torque is generated that overcomes this internal friction to set the joint into rotation, leading to the inertia-mediated, eponymous backlash that reduces the rotational play in the counter-direction of the respective rotation, which is equivalent to the hardening effect in particular or hysteretic saturation in general. If afterwards the rotation is reversed, a certain amount of generated torque is again lost during the backlash until the joint flexibility locks-out anew. Such *lost* torques are observable in the case of the KUKA iiwa as well, i.e. in the form of absent changes in torque measurement within an angular window of the magnitude of approximately 1° after joint rotation reversals. If we now suppose the presence of a combined effect of multiple hysteretic effects on the dynamics, then the reason for the enhanced influence of such nonlinear effects at the quasi-static state is that the hysteresis loops of the respective effects are not quickly traversed until loop saturation is reached.

In summary, it is hypothesized that, as a multitude of mechanic effects are not sufficiently excited during the execution of low velocity and directionally inconsistent motions as to reach hysteretic saturation, an adversarial amplification of nonlinear and hysteretic torque contributions is induced, which aggravates the option of identification of the underlying parameters due to the difficulties in the associated analytical formulations and the possibility to establish experimental conditions to sufficiently excite those parameters. Whereas under "normal" operational conditions allowing for a more consistent attainment of saturation, e.g. linear friction at higher joint velocities and deformational hardening at rotational consistency, the respective effects may be approximated by piecewise linear functions (Tjahjowidodo, Al-Bender, and Van Brussel, 2013) or even outright omitted. By virtue of joint torque and position sensors that are moreover attached at the transmission output side, the joint torque and position measurements are almost exempt from the effects of friction and inherent flexibility arising in the actuation itself. This has led to the dy-

dynamic model simplification of joint rigidity being commonly considered well acceptable for dynamic identification (Jubien, Gautier, and Janot, 2014; Gaz, Flacco, and De Luca, 2014) and external torque estimation purposes (Haddadin, De Luca, and Albu-Schäffer, 2017). Nonetheless, the insufficiency of a simplified rigid body model is still very noticeable from measurements and performance evaluations. This is evident from the fact that obvious prediction error patterns, e.g. in the form of characteristic joint torque over- and underestimations around motion turning points (Rueckert et al., 2017), still persist. At the same time, one can observe causally rather obscure prediction inaccuracies, which may in part result from stochastic outliers and other missing model assumptions as well (see Chapter 5 on page 68).

Under these aspects, the dynamics to be modeled are assumed to arise from a composite type of motion, that consists of interweaved phases with low and high kinematic intensity and directional regularity that provoke generalized mechanic hysteresis and quasi-static nonlinearities to a higher or lower degree respectively. Accordingly, this implies that such a composite motion may in part sufficiently comply with basic model assumptions of linear joint friction, joint rigidity and independence from past system states integrable into the common rigid body dynamics formulation⁴ (see Equation (3.31) on page 27). The complexity of the other motion components exceeding the simplifications of rigid body dynamics can in theory be accommodated for by multiple extensions of the parametric identification scheme, provided that the knowledge about the existence and nature of sufficiently many of those additional factors is available and can be cast into an analytical formulation that lends itself to the parametric regression from observable data. From a practical point of view, this approach may be considered rather detrimental, as extensive domain expertise is required and the risk of unintentionally neglecting minor but nonetheless significant dynamic effects cannot be precluded with certainty. Therefore it seems advantageous to utilize a weakly parameterized, data-driven approach, e.g. via artificial neural networks by leveraging the universal approximation capability with the hope of possibly extracting a near-complete, albeit hardly model directly from observable dynamic input-output relations.

In this sense, the proposed dynamic modeling technique is equally conceived as a hybrid composed of the white-box approach of parametric identification based on rigid body dynamics and a black-box component employing artificial neural networks. Generally speaking, the proposed model approximation technique is devised as a hybrid dynamic model architecture D_{hyb} that utilizes a backbone model $\mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \Phi = \boldsymbol{\tau}_{RBD}$ based on rigid body dynamics (see Section 3.3 and Equation (3.49) on page 24 and on page 30) and a downstream neural network $f_{NN}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \boldsymbol{\tau}_{NN}$ trained to predict a compensatory torque that is added onto the backbone prediction $\boldsymbol{\tau}_{RBD}$ such that the error with respect to the measured torques is minimized. The basic structure of the associated model function

⁴Strictly speaking, the kinematic state parameters obtainable from the temporal differentiation of the joint positions, i.e. commonly the joint velocities and accelerations, naturally incorporate information with regards to the immediately recent kinematic state history.

is thus defined as

$$\boldsymbol{\tau}_{HYB} = \boldsymbol{D}_{HYB} \tag{4.1}$$

$$\boldsymbol{\tau}_{HYB} = \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi} + f_{NN}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \tag{4.2}$$

$$\boldsymbol{\tau}_{HYB} = \boldsymbol{\tau}_{RBD} + \boldsymbol{\tau}_{NN} , \tag{4.3}$$

with the optimization goal for the parameters of f_{NN} being the minimization of

$$L(\boldsymbol{\tau}_{HYB}, \boldsymbol{\tau}_{meas}) = \frac{1}{m} \sum_i^m \|(\boldsymbol{\tau}_{HYB} - \boldsymbol{\tau}_{meas})_i\|^2 . \tag{4.4}$$

The rationale behind this gray-box strategy follows from the hypothesis that the physics-based framework provided by rigid body dynamics is capable of robustly predicting a large majority of dynamic effects, as the occurrence of joint torques is predominantly governed by the motion and inertial properties of the robot links, with the exception of the highly non-linear and hysteretical phenomena that are assumed to be exaggerated when approaching a quasi-static kinematic state and the joint motion direction oscillates within small angular ranges. Considering that the adequate analytical characterization and identification of the entirety of the contributing effects under these operational conditions is non-trivial, neural network architectures, due to their expressive power,, constitute a valid option for the compensation of the shortcomings of rigid body dynamics, which is as are a manifestation of the inherently high model bias. Conversely, this bias caused by the strong model assumptions of rigid body dynamics exerts a beneficial effect in combination with neural networks, as their inherent variance is thus balanced by the neural network predictions being guided by the rigid body model predictions, leading to a mutual optimization of the bias-variance trade-off. These presumable synergies between two model components are intended to be exploited especially in view of the high dimensionality of both the joint dynamic and kinematic state-space comprised of $(\boldsymbol{\tau}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbb{R}^{7 \times 4}$ for a 7-joint robot, where achieving global reliability similar to rigid body dynamics models over the total input space using neural networks without physical prior knowledge would at least presuppose an exceptionally large and rich data set as well as numerous optimizations of the network topology and training parameters. Lastly, this approach guarantees a certain degree of model explainability, as the final model predictions are based on a physically consistent foundation. The actual parameter identification for the rigid body dynamics model as well as the architecture, feature space design, and training scheme of the neural network component f_{NN} to be utilized are the subject of the following sections of this chapter.

4.1 Parametric Backbone Model

The development of the backbone model

$$\boldsymbol{\tau}_{RBD} = \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi} \tag{4.5}$$

is directly derived from the basic theory of parametric identification within the framework of rigid body dynamics described in Section 3.3 on page 24. Due to the inevitable presence

of linear dependencies between the terms of the Newton-Euler equations embedded in the columns of the stacked observation matrix $\overline{\mathbf{K}}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})$ obtained from multiple measurements (see Equation (3.51) on page 30), this matrix is rank deficient, i.e. $\text{rank}(\overline{\mathbf{K}}) < 84$ for $\Phi \in \mathbb{R}^{p \times n}$, containing 12 parameters for each of the 7 link bodies in the case of the KUKA iiwa platform. From a kinematic perspective, this is a consequence of the restriction of the kinematic degree of freedom of the motion of proximal links (Atkeson, An, and Hollerbach, 1986; Gautier and Khalil, 1988), which prevents the appear. Hence, the least squares solution of Equation (3.51) on page 30, i.e.

$$\begin{bmatrix} \tau_{meas}^1 \\ \tau_{meas}^2 \\ \vdots \\ \tau_{meas}^N \end{bmatrix} = \begin{bmatrix} \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^1 \\ \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^2 \\ \vdots \\ \mathbf{K}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas}, \mathbf{k})^N \end{bmatrix} \begin{bmatrix} \Phi \\ \Phi \\ \vdots \\ \Phi \end{bmatrix}, \quad (4.6)$$

by simple pseudo-inversion

$$\Phi^{LS} = \left(\overline{\mathbf{K}}^\top \overline{\mathbf{K}} \right)^{-1} \overline{\mathbf{K}}^\top \tau_{meas}. \quad (4.7)$$

is not viable, which is why the rank-deficiency of $\overline{\mathbf{K}}$ has to be mitigated, e.g. through singular value or QR decomposition. Using the QR decomposition of $\overline{\mathbf{K}}$ at the same time allows for the simultaneous elimination of the linear dependent columns in the model matrix yielding $\overline{\mathbf{K}}_b$ and thus the isolation of the parameters corresponding with the remaining columns that are actually identifiable (Khalil and Gautier, 1991). Although a virtually equivalent, geometrically justified determination of the identifiable parameters is possible (Khalil and Dombre, 2002), the purely numerical parameter reduction is preferred for the purposes of this thesis, due to its independence from kinematic considerations. These identifiable parameters, often referred to as base parameters Φ_b , constitute a subset of the full parameter vector Φ that contains the entirety of the link parameters. Thus, Φ_{LS} and $\overline{\mathbf{K}}$ in Equation (4.7) are substituted with their reduced counterparts Φ_b and \mathbf{K}_b , yielding

$$\Phi_b^{LS} = \left(\overline{\mathbf{K}}_b^\top \overline{\mathbf{K}}_b \right)^{-1} \overline{\mathbf{K}}_b^\top \tau_{meas}, \quad (4.8)$$

which can be readily solved to obtain the parameters Φ_b^{LS} numerically determined to be identifiable for the measured data.

Optimal Dynamic Excitation

In order to maximize the number of identifiable parameters and thus the fidelity of the model and robustness of the identification process against measurement noise, the kinematic states based on which $\overline{\mathbf{K}}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas})$ is constructed, i.e. the reference motion trajectory $(\mathbf{q}(t), \dot{\mathbf{q}}(t), \ddot{\mathbf{q}}(t))$ which the robot will be commanded to track during a statistical data acquisition experiment, can be optimized. A common optimization criterion for this purpose is the minimization of the condition number of the observation matrix $\text{cond}(\overline{\mathbf{K}})$ (Gautier and Khalil, 1992). From a phenomenological perspective, such an optimization has the effect of maximally exciting the influence of the dynamic effects included in the model on the robot, which is reflected in the appearance of the maximum number of independent

columns in $\overline{\mathbf{K}}(\mathbf{q}_{meas}, \dot{\mathbf{q}}_{meas}, \ddot{\mathbf{q}}_{meas})$. For this purpose, each joint i is assigned an excitation trajectory $\mathbf{q}_i(t)$ that is defined as a parameterized finite Fourier series composed of L sums of harmonic sine and cosine functions (Swevers et al., 1997)

$$q_i(t) = \sum_{l=1}^L \frac{a_{i,l}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{i,l}}{\omega_f l} \cos(\omega_f l t) \quad (4.9)$$

with

$$\dot{q}_i(t) = \sum_{l=1}^L a_{i,l} \cos(\omega_f l t) + b_{i,l} \sin(\omega_f l t) \quad (4.10)$$

$$\ddot{q}_i(t) = \sum_{l=1}^L -a_{i,l} \omega_f l \sin(\omega_f l t) + b_{i,l} \omega_f l \cos(\omega_f l t) , \quad (4.11)$$

where $L = 5$ and the angular frequency $\omega_f = 0.1\pi$ which yields a duration of 20 seconds for one period of the sinusoidal trajectory. The variables $a_{i,l}$ and $b_{i,l}$ parameterizing the respective joint trajectories constitute the variables of the constrained nonlinear optimization problem (Stürz, Affolter, and Smith, 2017a) for the determination of the total trajectory $\mathbf{q}(t) = \{q_1(t), \dots, q_{n=7}(t)\}$ based on which the observation matrix $\overline{\mathbf{K}}_b$ with the lowest condition number can be constructed, i.e.

$$\min_{a_{i,l}, b_{i,l}} \text{cond}(\overline{\mathbf{K}}_b) \quad (4.12)$$

$$\text{s.t.} \quad q_i(t) = \sum_{l=1}^L \frac{a_{i,l}}{\omega_f l} \sin(\omega_f l t) - \frac{b_{i,l}}{\omega_f l} \cos(\omega_f l t) , \quad (4.13)$$

$$\sum_{l=1}^L \frac{a_{i,l}}{l}, \sum_{l=1}^5 \frac{a_{i,l}}{l}, \sum_{l=1}^5 \frac{a_{i,l}}{l} = 0 , \quad (4.14)$$

$$\sum_{l=1}^L \frac{1}{l} \|a_{i,l} + b_{i,l}\| \leq \omega_f q_{i,\max} , \quad (4.15)$$

$$\sum_{l=1}^L \frac{1}{l} \|a_{i,l} + b_{i,l}\| \leq \dot{q}_{i,\max} , \quad (4.16)$$

$$[a_{i,l} \ b_{i,l}]^\top \leq \min \left(\frac{2\pi l f}{L} q_{i,\max}, \dot{q}_{i,\max} \right) [1 \ 1]^\top , \quad (4.17)$$

$$[a_{i,l} \ b_{i,l}]^\top \leq -\max \left(\frac{2\pi l f}{L} q_{i,\min}, \dot{q}_{i,\min} \right) [1 \ 1]^\top , \quad (4.18)$$

$$\forall i = 1, \dots, 7, \forall l = 1, \dots, 5 , \quad (4.19)$$

where the respective motion limits $q_{i,\min}, q_{i,\max}, \dot{q}_{i,\min}, \dot{q}_{i,\max}$ are those specified in Table 2.3 on page 14. Please note, that the upper and lower bounds on q_i have been decreased and increased respectively by 2° . Because of link inertia and the joint flexibility, the commanding of joint positions at the limit of the range of motion limit of the respective joint at times results in a slight overshoot of the link position sensed by the link-side position sensors, causing the controller to trigger a safety mode that locks all joint breaks in order to prevent the forceful exceedance of the mechanical joint limits. Equations (4.14) to (4.16) on the

previous page respectively denote constraints that ensure the trajectory to be initialized with zero joint position, velocity, and acceleration for every joint to prevent self-collisions during execution and provide a smooth commencing of the trajectory as to allow for a close tracking by not provoking overshooting control responses due to abrupt start accelerations. Equations (4.17) and (4.18) on the preceding page impose general feasibility upon the trajectory with regards to the specifications of Table 2.3 on page 14. The optimization problem is implemented in MATLAB and solved by the interior point optimization algorithm *fmincon*, which is run from multiple random initial points in the 70-dimensional search space using the *MultiStart* function (MATLAB, 2020). After optimization, the obtained trajectory is executed in a simulation using the robotics simulation software CoppeliaSim (Rohmer, Singh, and Freese, 2013) and checked for feasibility with regards to the occurrence of self-collisions or collisions with the environment. To this end, a geometric collision detection is performed at each simulation timestep using the internal *Coppelia Geometric Routines* function collection, where the collidable entities are defined as a CAD-model of the KUKA LBR iiwa 14 and a manually designed arrangement of primitive shapes that approximate the geometry of the obstacles surrounding the robot in the real experimental setup. After a few reiterations of the optimization process, an optimal trajectory yielding $\text{cond}(\overline{\mathbf{K}}_b) = 68$ was obtained (see Figure 4.2).

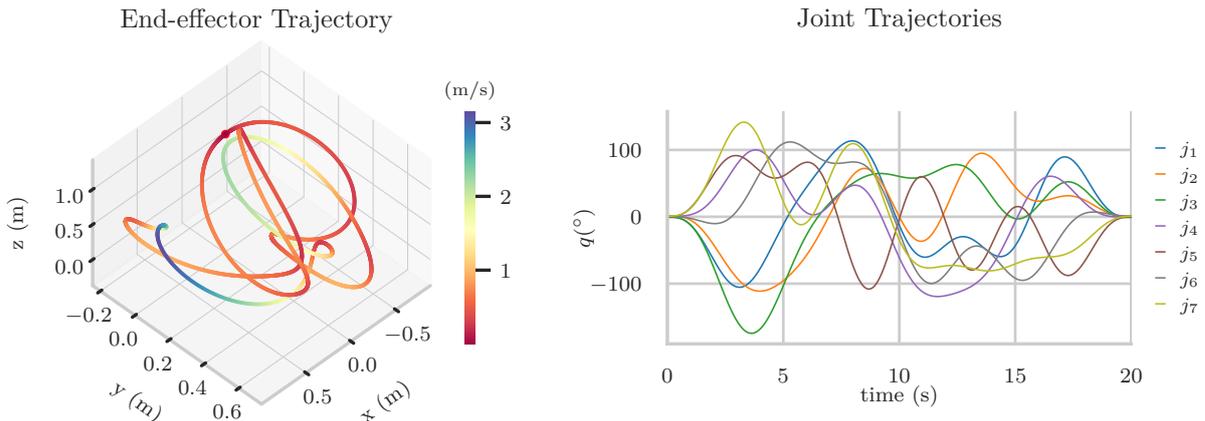


Figure 4.2: (Left) Cartesian path traversed by the end-effector during the execution of the optimized excitation trajectory motion, beginning and ending at the joint configuration $\mathbf{q} = (0, 0, 0, 0, 0, 0, 0)$, which is equivalent to the Cartesian end-effector position (0 m, 0 m, 1.306 m). The coloring of the line denotes the linear velocity of the end-effector at the respective location on the motion path. (Right) The excitation trajectory depicted in the configuration space, where each curve shows the temporal angular position of the respective joint.

As described in the introductory remarks of Chapter 4 on page 41, the main focus of this thesis with respect to dynamic modeling is the identification of a model dedicated to motions with lower dynamic range and frequent directional reversals due to the increase in complex nonlinear effects under these operational conditions, which at the same time are characteristic for continuous contact human-robot interactions applications. In order to incorporate the associated dynamic phenomena, especially regarding low-velocity fric-

tion, within the extended rigid body dynamics formulation which also entails a viscous and Coulomb friction model Equation (3.35) on page 28, the optimized trajectory is executed repeatedly with decreasing trajectory velocity per repetition. This is accomplished by the repeated concatenation of the original excitation trajectory (see Figure 4.2 on the previous page) with itself, where the trajectory point of every concatenated version of the original trajectory are interpolated, i.e. resampled with the MATLAB function *resample* with a gradually increasing sampling rate. After interpolation of the trajectory points according to the desired sampling rate, *resample* applies a low-pass anti-aliasing filter on the resampled trajectory. Due to the periodicity of the trajectory, the concatenation does not introduce any discontinuities between the respective end- and start-points of the concatenated segments. For the construction of the stacked observation matrix to be used for

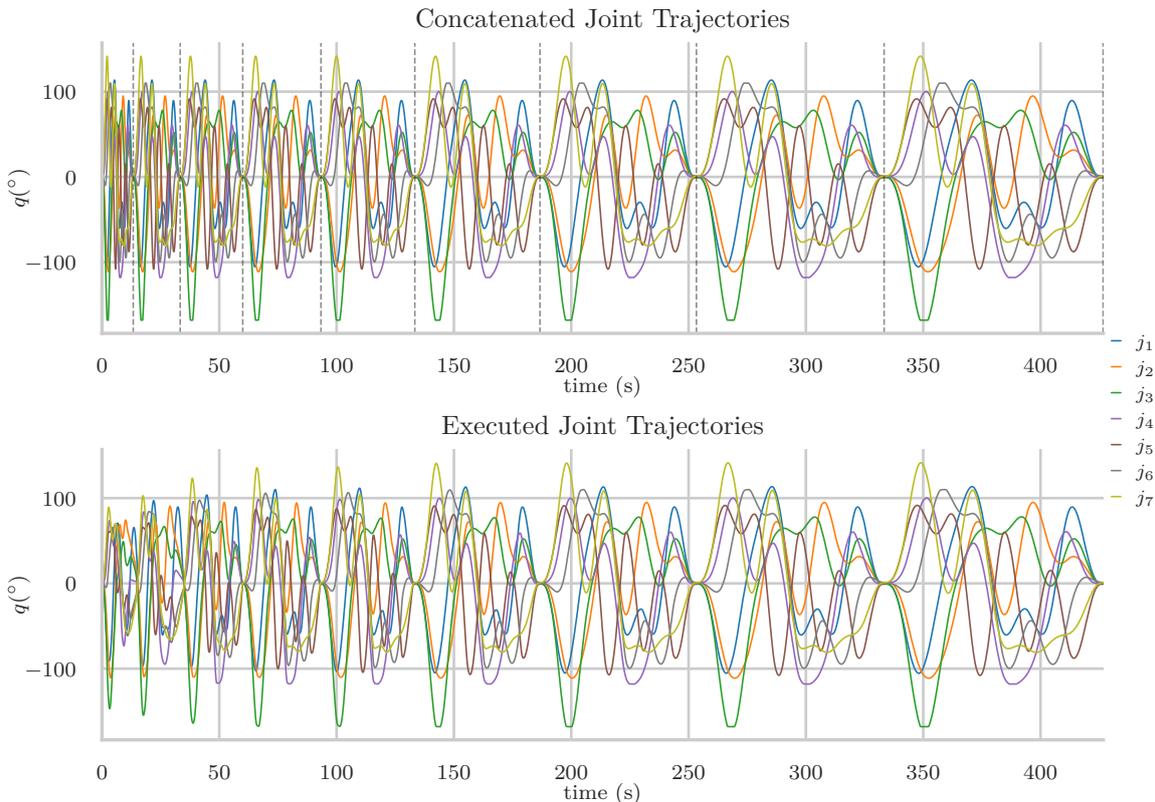


Figure 4.3: (*Top*) The excitation trajectory depicted in Figure 4.2 on the previous page is concatenated 9 times with itself, where each concatenated sub-sequence represents an resampled version of the original trajectory with sampling rates of 100 Hz, 150 Hz, 200 Hz, 250 Hz, 300 Hz, 400 Hz, 500 Hz, 600 Hz and 700 Hz respectively. The concatenated segments are demarked by vertical dashed lines at their respective temporal endpoints. (*Bottom*) The same trajectory tracked by the robot during the dynamic excitation experiment. For higher accelerations and frequencies of directional reversal in the earlier segments of the concatenated trajectory, the actually executed trajectory shows more pronounced deviations from the reference. This is attributable in part to the limited attainable control frequency and unidentified processes within the low level controller, likely altering the commanded trajectory to ensure kinematic feasibility of motion at high joint velocities.

the parameter regression, the concatenated excitation trajectory is executed on the robot by successively commanding the trajectory points in joint space with an average control frequency of 150 Hz depending on some minor fluctuations of the network latency (see Figure 4.3 on the preceding page). The proprioceptive data consisting of $(\mathbf{q}_{meas}(t), \boldsymbol{\tau}_{meas}(t))$ is queried from the controller simultaneously with every motion command and recorded with the corresponding timestamp, yielding a data set containing $N = 64009$ samples over a total trajectory duration of $T = 426$ seconds. Although the joint position sensors yield relatively precise measurements, they nonetheless are slightly perturbed by noise. Because of this, the temporal derivatives of the joint position that are necessary for the construction of the observation matrix cannot be obtained by simple numerical differentiation, as this would drastically amplify the noise level for every subsequent derivative that would render the equations of motions based on this data unusable for the parameter regression. Equations (4.10) and (4.11) on page 48 can be employed to determine the joint velocities and accelerations analytically from the joint positions, provided that the reference trajectory $\mathbf{q}(t)$ is tracked closely by the robot, which eliminates the complications associated with the numerical differentiation of the joint position measurements to obtain the full kinematic state for the construction of the observation matrix (Swevers et al., 1997), i.e. the strong amplification of the joint position measurement noise in its derivatives and the resulting necessity of signal filtering. However, the close trajectory tracking is oftentimes not guaranteed, like in the case of the experimental setup of this thesis, e.g due to low-level robot control routines that alter the commanded trajectory (Schreiber, Stemmer, and Bischoff, 2010) and variable network communication latency in teleoperation control (Salcudean et al., 1999). Therefore, the trajectory is repeated 10 times and the obtained data $(\mathbf{q}_{meas}(t), \boldsymbol{\tau}_{meas}(t))$ are averaged over all of the excitation experiments according to

$$\tilde{\mathbf{q}}_{meas}(t) = \frac{1}{10} \sum_{i=1}^{10} \mathbf{q}_{meas}(t), \quad (4.20)$$

$$\tilde{\boldsymbol{\tau}}_{meas}(t) = \frac{1}{10} \sum_{i=1}^{10} \boldsymbol{\tau}_{meas}(t), \quad (4.21)$$

for every time step t of the trajectory. Afterward, the averaged data $(\tilde{\mathbf{q}}_{meas}, \tilde{\boldsymbol{\tau}}_{meas})$ are subjected to Gaussian-weighted moving average filter with a window size of 10 time steps. The temporal derivatives of the joint positions are then determined by numerical central differentiation and then immediately filtered in forward and reverse direction by a first order low-pass Butterworth filter with a cutoff frequency of 1.

Base Parameter Estimation

The resultant averaged and filtered data are used to construct $\widetilde{\mathbf{K}}_b(\tilde{\mathbf{q}}_{meas}, \tilde{\dot{\mathbf{q}}}_{meas}, \tilde{\ddot{\mathbf{q}}}_{meas}, \mathbf{k})$ and set up the linear regression equation

$$\boldsymbol{\Phi}_b^{LS} = \left(\widetilde{\mathbf{K}}_b^\top \widetilde{\mathbf{K}}_b \right)^{-1} \widetilde{\mathbf{K}}_b^\top \tilde{\boldsymbol{\tau}}_{meas}, \quad (4.22)$$

where the rank revealing QR-decomposition of $\widetilde{\mathbf{K}}$ resulted in $\text{rank}(\widetilde{\mathbf{K}}_b) = 57$ independent columns of the observation matrix, or in other words, identifiable base parameters, which is

in accordance with the previously reported number of base parameters for the KUKA LBR iiwa 14 (Stürz, Affolter, and Smith, 2017b; Xu et al., 2020). The obtained estimates of the base parameters Φ_b that now define the backbone dynamic model

$$\tau_{RBD} = \mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \Phi_b^{LS} \quad (4.23)$$

are given in Table 4.6 on page 54. Using Φ_b^{LS} for the inference of the averaged and filtered versions of the joint torques recorded during the execution of the excitation trajectory depicted in Figure 4.3 on page 50 based on the corresponding averaged and filtered kinematic data, a average mean squared error of $\frac{1}{N} \sum_{n=1}^N \|(\tau_{RBD}(n) - \tau_{meas}(n))^2\| = 0.2543$ Nm over $N = 64009$ could be achieved; for the joint-wise mean squared errors, see Table 4.4.

Table 4.4: The statistical accuracy of the joint torque estimates for the filtered and averaged joint torque measurements acquired during the execution of the concatenated excitation trajectory (see Figure 4.3 on page 50). The average mean squared error and standard deviation of the predictions is given separately for every joint as well as an total prediction accuracy average over all joint torques.

Joint	MSE (Nm)	σ (Nm)
1	0.4926	0.7019
2	0.8015	0.8953
3	0.1168	0.3417
4	0.2068	0.4548
5	0.0721	0.2686
6	0.0656	0.2542
7	0.0255	0.1596
average	0.2543	0.4394

The statistical reliability of the parameters identified through the described process and the acquired data, as well as the torque prediction accuracy following from this, can be evaluated by means of their relative standard deviations $\sigma_{\Phi_{b,p}^{LS}}$ of the respective p -th base parameters $\Phi_{b,p}$, as proposed by (Khalil and Dombre, 2002). The computation of $\sigma_{\Phi_{b,p}^{LS}}$ is defined as

$$\sigma_{\Phi_{b,p}^{LS}} = 100 \left(\frac{\sigma_{\Phi_{b,p}^{LS}}}{|\Phi_{b,p}^{LS}|} \right), \quad (4.24)$$

$$\sigma_{\Phi_{b,p}^{LS}} = \sqrt{C_{\Phi_{b,p}^{LS}}(p, p)}, \quad (4.25)$$

$$C_{\Phi_{b,p}^{LS}} = \sigma_{var}^2 \left(\widetilde{\mathbf{K}}_b^\top \widetilde{\mathbf{K}}_b \right)^{-1}, \quad (4.26)$$

$$\sigma_{var}^2 = \frac{\left\| \widetilde{\tau}_{meas} - \widetilde{\mathbf{K}}_b \Phi_b^{LS} \right\|^2}{Nn - p_b}, \quad (4.27)$$

where $N = 64009$, $n = 7$, $p_b =$ are the numbers of joints, sampled time-steps and base parameters respectively. It is worth pointing out that the obtained base parameters along with their respective relative standard deviation (see Table 4.6 on page 54) result

4 Hybrid Models for Torque Estimation

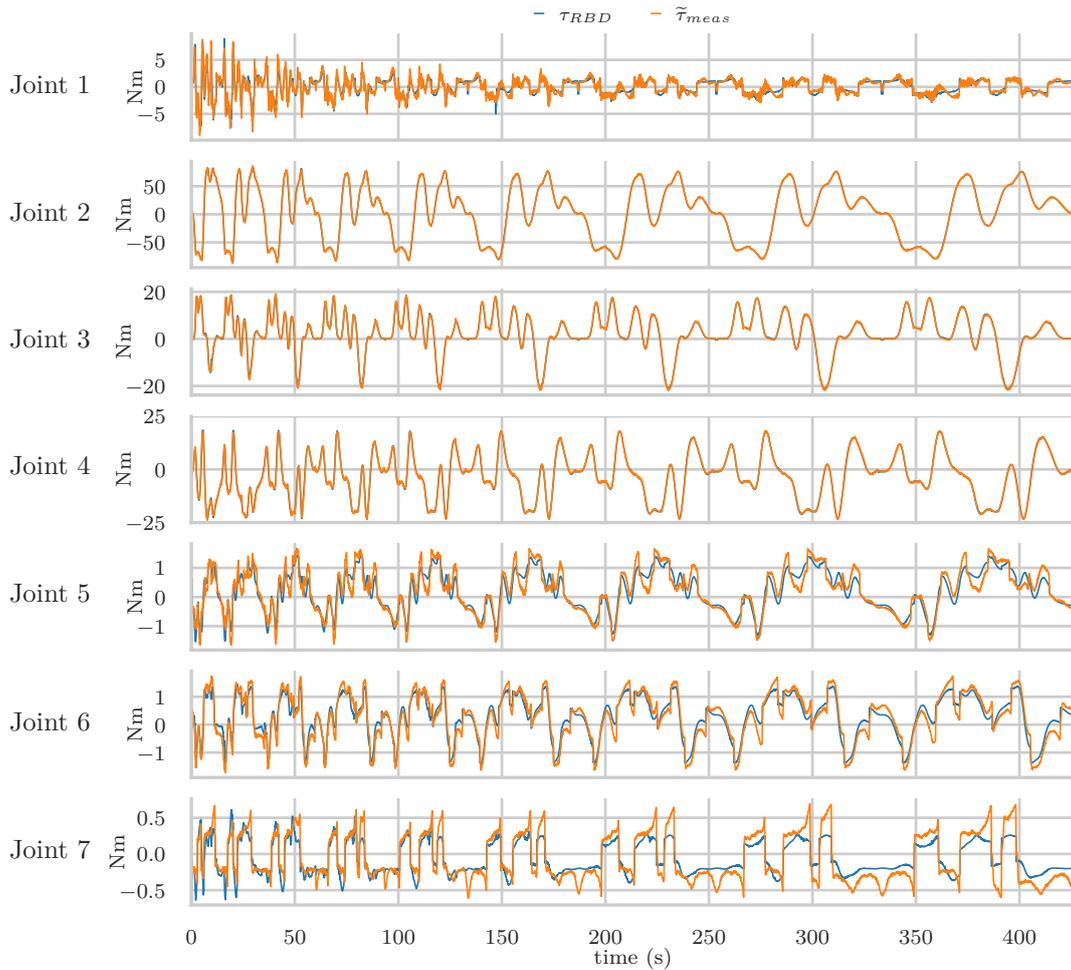


Figure 4.5: Comparison between the filtered and averaged joint torques $\tilde{\tau}_{meas}$ recorded during the execution of the concatenated excitation trajectory, and the model estimates τ_{RBD} made based on the identified parameters. The absolute error between

from an unconstrained, ordinary least squares solution. As such, they are not strictly physically consistent in the sense that the quantities expressed by the respective parameters are not necessarily plausible with respect to the real-world link properties, e.g due to the appearance of negative link masses or centers of mass outside of the actual link geometry. Physical consistency can certainly be imposed upon the parameters, albeit one has to forfeit the simplicity of the solution by inversion in favor of a constrained optimization problem (Stürz, Affolter, and Smith, 2017b; Xu et al., 2020; Ledezma Díaz and Haddadin, 2018) to enforce physical consistency, which yet at the same constrains the fitting of the observed data by the model parameters. The knowledge of the true physical parameters, however, is only of secondary importance for the purposes of this thesis, while the primary concern is the estimation of the occurring joint torques, where the accuracy is generally unaffected by the use of physically inconsistent, virtual link parameters (Díaz Ledezma and Haddadin, 2017). Therefore, the trade-off between parameter plausibility on one hand and the rapid attainability of an effectively equivalent model on the other is decided in favor of the latter.

Table 4.6: The base parameters Φ_b^{LS} identified via the least-squares solution of Equation (4.8) on page 47 for measurements obtained during the tracking of the excitation trajectory depicted in Figure 4.3 on page 50. The parameters identified for each link are grouped by the dashed lines; the number of identified parameters per link increases distally along the kinematic chain approximately in accordance with the distally increasing degree of motion of the respective links.

$\Phi_{b,p}$	$\Phi_{b,p}^{LS}$	$\sigma_{\Phi_{b,p}^{LS}}$	$\Phi_{b,p}$	$\Phi_{b,p}^{LS}$	$\sigma_{\Phi_{b,p}^{LS}}$	$\Phi_{b,p}$	$\Phi_{b,p}^{LS}$	$\sigma_{\Phi_{b,p}^{LS}}$
I_1^{zz}	0.029	137.41	m_4	-0.021	5.30	$m_6 c_6^z$	0.074	1.31
F_1^v	-0.925	1.15	$m_4 c_4^y$	0.004	36.73	I_6^{xy}	-0.074	10.62
F_1^c	0.002	18.48	I_4^{xx}	0.416	12.73	I_6^{xz}	-0.126	6.10
$m_2 c_2^x$	0.066	3.33	I_4^{xy}	0.057	17.51	I_6^{yy}	-0.080	30.36
I_2^{xy}	0.006	342.12	I_4^{xz}	-0.116	-11.02	I_6^{yz}	0.090	8.37
I_2^{xz}	-0.012	213.76	I_4^{yy}	0.326	14.59	I_6^{zz}	0.007	288.00
I_2^{yz}	-0.371	8.77	I_4^{yz}	-0.042	31.33	F_6^v	-0.271	2.27
I_2^{zz}	-0.154	40.08	I_4^{zz}	0.295	7.85	F_6^c	0.005	7.66
F_2^v	-0.920	1.11	F_4^v	-0.377	2.04	$m_7 c_7^x$	-0.012	5.94
F_2^c	-0.019	2.89	F_4^c	0.007	6.09	$m_7 c_7^y$	0.002	39.78
$m_3 c_3^x$	-0.004	40.05	$m_5 c_5^x$	0.006	17.61	$m_7 c_7^z$	-0.115	0.69
$m_3 c_3^z$	-5.823	0.02	$m_5 c_5^z$	-2.308	0.05	I_7^{xx}	0.023	110.26
I_3^{xy}	-0.163	10.78	I_5^{xy}	0.016	59.70	I_7^{xy}	-0.019	28.90
I_3^{xz}	0.214	7.01	I_5^{xz}	-0.011	98.25	I_7^{xz}	-0.061	10.85
I_3^{yy}	0.731	7.21	I_5^{yy}	0.113	10.44	I_7^{yy}	0.031	76.61
I_3^{yz}	-0.424	4.42	I_5^{yz}	0.051	49.96	I_7^{yz}	-0.008	68.70
I_3^{zz}	-0.187	22.95	I_5^{zz}	-0.116	6.52	I_7^{zz}	0.072	16.18
F_3^v	-0.280	2.90	F_5^v	0.00	20.44	F_7^v	-0.172	3.62
F_3^c	-0.002	17.54	$m_6 c_6^x$	0.004	22.40	F_7^c	-0.002	13.25

The preliminary validation of the model accuracy attainable through the virtual base parameters is given in Table 4.6 and Figure 4.5 on the preceding page and on the current page, where the torque prediction performance is evaluated on the same post-processed proprioceptive data based on which the parameters were identified. These results show that the rigid body dynamics framework is indeed capable of describing the majority of the occurring dynamic effects, as there is a minor absolute deviation between the torque predictions and measurements during the entire trajectory. Naturally, these results on their own do not necessarily imply a global validity of the obtained model, for which a more extensive performance evaluation was carried out in unison with the training and validation data sets used for the neural-network based models in the following sections. Regardless, it is immediately visible from the temporal course of the torque predictions in relation to the torque measurements, that there is a relative rise in the estimation error $\|(\tau_{RBD}(t) - \tau_{meas}(t))^2\|$ for time steps t at which a local torque peak was sensed, i.e. due to a reversal of motion direction or acceleration (compare with Figure 4.3 on page 50). The extent of these errors is less apparent in relation to the high torque ranges sensed at joints 2 to 4 caused by them bearing the torques occurring due to the summed weights of the distal links. This does not apply for the remaining joints, including joint 1, which obviously, although being the most proximal link, does not experience any gravitational torque load

from the distal link due to the collinearity of its rotational axis with the gravity vector in the upright mounting position (see Figure 2.1 on page 13). Instead, here one can recognize significant relative prediction errors, that moreover exhibit an increase at the slower joint velocities towards the end of the trajectory. Overall, these findings are consistent with the hypothesis of the inherent insufficiency of rigid-body modeling at low velocities and directional irregularity of motion.

Integration within Hybrid Model

The integration of the backbone model $\tau_{RBD}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}, \Phi_b^{LS}) = \mathbf{K}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \Phi_b^{LS}$ within the proposed hybrid model defined in Equation (4.3) on page 46 is accomplished via the upstream pre-connection of an algorithmic version of the model components of rigid body dynamics outlined in Section 3.3 on page 24 along with the associated model assumptions. Specifically, the *Recursive Newton-Euler Algorithm* (RNEA, see Algorithm 4.7) (Luh, Walker, and Paul, 1980) derived from the Newton-Euler equations of motion has been widely established as the computationally most efficient implementation of rigid body dynamics known (Featherstone, 1987).

As already described, the underlying algorithmic principle is a two-part procedure beginning with the propagation of the motion of a proximal link onto its distally following link along the kinematic chain. In this sense, the motion of the most proximal link i is only affected by the rotation of its own proximal joint; the motion of the succeeding link $i + 1$ is then the combined result of both the rotation of its own proximal joint and the motion of the preceding link i ; the motion link $i + 2$ accordingly results from the rotation of its own joint and the combined motion of both preceding link i and $i + 1$ and so on. The second stage of RNEA computes the torques acting on the links along their rotational axes coinciding with their respective proximal joints, where the torques are propagated in the opposite direction from the most distal link towards the proximal ones, i.e. the torque τ_i occurring at the last link $i = n$ is solely the result of its own motion and inertial properties if we are to disregard any external influences or dynamic effects exceeding purely inertial considerations; τ_{i-1} arises due to the motion of the rigid body of link $i - 1$ as well as link i and so on.

Algorithm 4.7: The Recursive Newton-Euler Algorithm for a robot with n revolute joints, where the joint frames are oriented such as to ensure the coincidence between the respective z-axis $\mathbf{z} = [0 \ 0 \ 1]^\top$ and the axis of rotation. Based on the original efficient formulation of the algorithm by (Luh, Walker, and Paul, 1980) and modifications from (Featherstone and Orin, 2016). Please refer to Chapter 3 and Section 3.3 for the definition of the algorithm's inputs and outputs as well as the derivations of the remaining operations.

algorithm RNEA

inputs: $n, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{R}, \mathbf{s}$, // $n, \mathbf{R}, \mathbf{s}$ are derived from the kinematic model \mathbf{k}

$\mathbf{m}, \mathbf{c}, \mathbf{I}, \mathbf{g}$ // m, c, I are the inertial parameters, \mathbf{g} is the gravitational vector

output: $\boldsymbol{\tau}$

$\boldsymbol{\omega}_0 = \mathbf{0}$

$\dot{\boldsymbol{\omega}}_0 = \mathbf{0}$

$\ddot{\mathbf{p}}_0 = -\mathbf{g}$

```

for i = 1 to n do
 $\boldsymbol{\omega}_i = \mathbf{R}_{i-1}^\top \boldsymbol{\omega}_{i-1} + \dot{q}_i \mathbf{z}$ 
 $\dot{\boldsymbol{\omega}}_i = \mathbf{R}_{i-1}^\top \dot{\boldsymbol{\omega}}_{i-1} + (\mathbf{R}_{i-1}^\top \boldsymbol{\omega}_{i-1}) \times \dot{q}_i \mathbf{z} + \ddot{q}_i \mathbf{z}$ 
//  $\dot{\mathbf{p}}_i = 0$  in next line for inertial identification without linear force sensing
 $\dot{\mathbf{p}}_i = \mathbf{R}_{i-1}^\top [\dot{\boldsymbol{\omega}}_{i-1} \times \mathbf{s}_{i-1} + \boldsymbol{\omega}_{i-1} \times (\boldsymbol{\omega}_{i-1} \times \mathbf{s}_{i-1}) + \dot{\mathbf{p}}_{i-1}]$ 
 $\mathbf{f}_i = m_i (\dot{\mathbf{p}}_i + \dot{\boldsymbol{\omega}}_i \times \mathbf{c}_i + \boldsymbol{\omega}_i \times \dot{\boldsymbol{\omega}}_i \times \mathbf{c}_i)$ 
 $\mathbf{n}_i = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i + \mathbf{c}_i \times \mathbf{f}_i$ 
end for
for i = n-1 to 1 do
 $\mathbf{f}_i = \mathbf{f}_i + \mathbf{R}_i^\top \mathbf{f}_{i+1}$ 
 $\mathbf{n}_i = \mathbf{n}_i + \mathbf{R}_i^\top \mathbf{n}_{i+1} + \mathbf{s}_i \times \mathbf{R}_i^\top \mathbf{f}_i$ 
 $\tau_i = \mathbf{z}^\top \mathbf{n}_i$ 
end for
return  $\boldsymbol{\tau}$ 

```

4.2 Downstream Neural Networks

The parametric backbone model $\boldsymbol{\tau}_{RBD} = \mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS}$ from Equation (4.23) on page 52 is complemented with a sequential black box model approximator that is implemented as a neural network due to their expressiveness (Hornik, Stinchcombe, and White, 1989) and scalability to large data-sets (Rueckert et al., 2017). Inserting Equation (4.23) in Equation (4.3) on page 46, the hybrid model equation is restated for the identified inertial parameters $\boldsymbol{\Phi}_b^{LS}$ as

$$\boldsymbol{\tau}_{HYB} = \mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS} + f_{NN}((\mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS}), \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}), \quad (4.28)$$

$$\boldsymbol{\tau}_{HYB} = \boldsymbol{\tau}_{RBD} + \boldsymbol{\tau}_{NN}. \quad (4.29)$$

For the actual network architecture, several alternatives are proposed, taking inspiration from a plethora of related previous research in the field of purely neural-network-based or hybrid dynamic modeling of robots (Hitzler et al., 2019; Jiang, Ishida, and Sunawada, 2006; Wang et al., 2020; Liu, Wang, and Wang, 2021; Yilmaz et al., 2020; Smith and Hashtrudi-Zaad, 2005). Rather surprisingly, only a fraction of the existing work on this topic incorporates the influence of temporal effects in the respectively propose robot dynamic models that span a longer time frame than is already included in the trivial temporal motion derivatives. The existing contributions in pursuit of learning robot dynamics as a long(er) time-series model are predominantly based on a recurrent neural network architecture (see Section 3.4 on page 36), which has been reported to yield a good model accuracy, surpassing that of other state-of-the-art techniques learning single time-step dynamic relations (Rueckert et al., 2017; Hirose and Tajima, 2017; Hanafusa and Ishikawa, 2019). Thereby, the fundamental questions, for which an answer is attempted to be given, are 1) whether the existing approaches yield satisfactory modeling performance for the irregular, small-scale motions defined in the introduction of Chapter 4 on page 41, where the algorithmic generation of an appropriate data set for this purpose is subject of the following Section 4.3, and 2) whether the accuracy can be improved by the proposed hybrid model approach. The investigated architectures are presented in the following sections.

Rotational Displacement Encoding

The encoding of long-term temporal dependencies in the observations of the robot dynamics can be accomplished by utilization of specialized sequence modeling network architectures either through the exploitation of recurrence, e.g. RNN’s, LSTM’s, or on the basis of an attention-mechanism, e.g. Transformer networks (Vaswani et al., 2017). Still, these techniques have a practical upper limit for the temporal distance over which they can effectively learn. Conversely, more generic feed-forward network topologies such as MLP’s or CNN’s have no inherent architectural feature that makes them particularly applicable for sequence learning. The Transformer architecture, developed for natural language processing, compensates for the lack of recurrence by making use of a sequentially varying, sinusoidal positional encoding that is summed with the input features to impose a more pronounced temporal order on the input sequence.

Here, a similar approach is proposed, that is adapted to the particularities of the robotics domain. With regards to the tentative causal effect for time-dependency effects in robot dynamics, i.e. some manifestation of mechanical hysteresis, the temporal relations pertain to the history of joint positions which defines the location on a hysteresis loop Chapter 4 on page 43, where a sufficiently long rotational displacement in one direction lets the hysteresis transition into linear saturation and frequent directional reversals confine the associated dynamic effects to manifest with pronounced non-linearity. To abstractly express the location on such a hysteresis loop in a form that is easily applicable as a numeric input feature, the rotational displacement since the last directional reversal $r \in [-10^\circ, 10^\circ]$ for each joint is concatenated as a vector $\mathbf{r} \in \mathbb{R}^n$ with the remaining input features, yielding the full feature space $\text{vec}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r}) \in \mathbb{R}^{35}$ such that Equation (4.28) on the preceding page is changed to

$$\boldsymbol{\tau}_{HYB} = \mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS} + f_{NN} \left((\mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS}), \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r} \right). \quad (4.30)$$

The rotational displacements since directional reversal $\|\mathbf{r}^\circ\| > 10^\circ$ are replaced by the displacement threshold of $\pm 10^\circ$, representing the area where any effect of mechanical hysteresis is assumed to have reached saturation (see Figure 2.5 on page 16). In practice, the feature \mathbf{r} can simply be calculated by buffering the retrieved joint positions and triggering the joint-wise measurement of the rotation since the last instance where the sign of the positional change switched for the respective joints.

This auxiliary feature is hypothesized to guide the extraction of long-term dependencies within the data for sequence modeling networks as well as augment the performance of non-recurrent architectures for robot dynamics modeling. The evaluation of the influence of this additional feature in the context of the obtained data set (see Section 4.3 on page 59) is presented in Chapter 5 on page 68.

Multilayer Perceptron

Derived from the network topologies proposed in (Hitzler et al., 2019; Yilmaz et al., 2020), a multilayer perceptron (MLP) based approach is utilized as baseline neural-network architecture. Using on the standard theory provided in Section 3.4 on page 31, one MLP is constructed for each of the $n = 7$ joints, where each MLP_j for the j -th joint consists of

- an input layer with 35 neurons accepting the same vectorized input $\mathbf{x} = \text{vec}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r}) \in \mathbb{R}^{35}$ consisting of all the observable data for each joint,
- one hidden layer with 100 neurons with ReLu activation,
- an output layer with 1 neuron corresponding to the torque prediction τ_{HYB}^j for the j -th joint.

Each MLP_j performing a single-joint prediction $f_{\text{MLP}_j}(\mathbf{x}) = \tau_{NN}^j$, where $\tau_{NN}^j \in \mathbb{R}^1$, is arranged in a shared network topology where the respective outputs are concatenated, such that the overall network function $f_{\text{MLP}}(\mathbf{x}) = \boldsymbol{\tau}_{NN}$, where $\boldsymbol{\tau}_{NN} \in \mathbb{R}^{n=7}$, is obtained, where the network parameters are learned jointly. For comparison purposes, this architecture is abbreviated as MLP-7.

LSTM

Considering the discussed influence of effects with a pronounced time-dependency on the dynamic system, that cannot possibly be encoded by trivial temporal derivatives of positional measurements, e.g. the entirety of hysteretic phenomena ranging from the nonlinearity of unsaturated frictional effects, backlash and hypothetical minimal internal elastic deformations (see introductory remarks of Chapter 4 on page 41), the dynamic modeling problem is extended from regarding single-time-step relations between kinematic and dynamic states to the mapping of a time-series of kinematic states to a single dynamic state by using a long short-term memory network (LSTM) (Rueckert et al., 2017; Wang et al., 2020). The LSTM (see Section 3.4 on page 36) is utilized to establish a sequence-to-single-step mapping

$$f_{\text{LSTM}}(\mathbf{x}^{(t, \dots, T)}) = \boldsymbol{\tau}_{NN}^{(T)} \quad (4.31)$$

with T being the length of the sequence that contains observations $\mathbf{x}^{(t)} = \text{vec}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r}^{(t)})^{(t)}$ for every time-step t up until the prediction step Tt corresponding to the end of the sequence. Two time-window sizes $T = 100$ and $T = 500$ are compared, in order to investigate whether possible accuracy improvements can be gained from longer sequence information, or put differently, whether the dynamic model is significantly influenced by effects that exhibit a very long temporal dependency. Using the fundamental functions governing an LSTM cell given by Equations (3.66) to (3.71) on page 39, the LSTM network is implemented as two consecutive LSTM cells where

- the first LSTM cell has 35 input neurons accepting $\mathbf{x}^{(t)} = \text{vec}(\boldsymbol{\tau}_{RBD}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r}^{(t)})^{(t)}$, and 35 hidden neurons $\mathbf{h}^{(t)}$,
- the second LSTM cell has likewise 35 input neurons, which are connected to the hidden neurons of the first cell, and 7 hidden neurons, corresponding to the individual joint torques,
- the prediction is equal to the final states of the hidden neurons $\mathbf{h}^{(T)}$ of the second LSTM cell at the end of the sequence.

This network is abbreviated as LSTM-2. Based on the same topology as LSTM-2, second recurrent architecture is evaluated, where a fully connected layer with linear activation is followed after the second cell, that is typically added to improve the overall regression performance by compensating for the squashing sigmoid output function (see Equation (3.71) on page 39). This modification is named LSTM-2-FCL.

Transformer

Lastly, a Transformer network is investigated as a non-recurrent topology that has recently achieved good results on a number of sequence-to-sequence modeling and time-series prediction tasks from various domains (Giuliani et al., 2021). Originally proposed in (Vaswani et al., 2017), the Transformer constitutes an encoder-decoder architecture that does not rely on recurrence. The input is concatenated by a sinusoidal encoding and fed through a series of encoding layers each containing a multi-head scaled dot-product attention, normalization, and fully connected sub-layer. The encoded features are then decoded by a similar layered structure where they are subject to a joint attention layer accepting both the encoding as well as the previous predictions to generate the output for the next step. The *vanilla* implementation of the architecture is adopted and configured to accept the input sequence and output dimensions defined for the LSTM, yielding the network function

$$f_{\text{TF}}(\mathbf{x}^{(t, \dots, T)}) = \boldsymbol{\tau}_{NN}^{(T)}, \quad (4.32)$$

to be inserted in the hybrid model function when using the Transformer as the neural network component.

Input/Output Normalization

Considering the largely varying numerical scales of the different features and benefits with respect to network generalization and convergence (Sola and Sevilla, 1997), the entirety of the investigated architectures is equipped with a pre-and post-processing component to normalize the input and output features respectively, that is implemented as an integral part of the networks. Using the training data (see Section 4.4 on page 65) obtained through the process described in Section 4.3, the mean and standard deviation over all samples is calculated for the occurring joint positions, velocities, accelerations, and torques for each joint respectively. Let \bar{x} be the mean of one feature x for one joint and $\sigma(x)$ the corresponding standard deviation, then the input features are normalized as in

$$x_j^{\text{norm}} = \frac{x_j - \bar{x}_j}{\sigma(x_j)}, \quad (4.33)$$

and propagated through the network. The inverse operation is performed at the output where the prediction torques of one joint are subject to

$$\tau_j^{\text{out}} = \sigma(\tau_j) \tau_j^{\text{norm}} + \bar{\tau}_j, \quad (4.34)$$

where the de-normalization parameters for the output torques are the same used for normalization of the auxiliary input torques originating from the rigid body dynamics model. The visualization of the numerical ranges and distribution of the observable joint-space features is given in Figure 4.11 on page 66.

4.3 Algorithmic Training Data Generation

Most data sets obtained for robot dynamics research either focus on complex small-scale motion tasks within a fairly constrained effective workspace, simple large-scale motion (Vijayakumar and Schaal, 2000), larger-scale repetitive motions (Polydoros and Nalpantidis,

2016). To the extent of the research carried out during this work, there is no proprioceptive data set available for multi-articulated, possibly redundant, serial robots, that covers a wide kinematic range while specifically capturing the joint torques occurring during irregular, low-intensity, and fine-scale motion. In order to provide such a data set for the purposes of developing and evaluating dynamic modeling and torque estimation techniques for this specialized motion type, an algorithmic method for the randomized, parametric generation of a continuous motion path has been developed. The intention is to primarily provoke dynamic effects encountered during fine motion at low velocities and frequent irregular directional reversals within narrow positional windows on the one hand, e.g. dynamic and sensory hysteresis and low-velocity non-linear friction. On the other hand, the aim is to capture these effects over a wide kinematic range covering large amounts of the effective workspace, i.e. in as many different combinations of joint positions and motion directions as possible to faithfully represent general motions that would be executed during practical tasks.

For this purpose, the developed random fine motion generation algorithm (RFMA, see Algorithm 4.10 on page 64) is inspired by the manual motion during the manipulation of an ultrasound probe for sonography: After a phase of a medium velocity, long-distance translation of the probe onto an arbitrarily defined anatomic target region, the motion speed and range is significantly reduced and the manner of manipulation transitions to be primarily explorative and dominated by frequent re-orientations of the probe. Likewise, RFMA makes use of a two-phased partitioning of the robotic motion that is composed by an initial *reaching*-phase where a target configuration is approached, and a subsequent *exploring*-phase where the robot performs fine-scale irregular reconfigurations based on the prior target configuration (see Figure 4.8 on the next page). This is realized by the pre-calculation of geometric motion path defined as path of length P of joint configurations $\mathbf{C} = \{\mathbf{q}_1, \dots, \mathbf{q}_P\}$ where $\mathbf{q}_p \in \mathbb{R}^{n=7}$ designates the p -th waypoint in the configuration path \mathbf{C} , in contrast to a Cartesian motion description based on end-effector poses. By doing so, the necessity of any inverse kinematics calculations during motion execution or is avoided, and the verification of the kinematic feasibility of a specific pose is reduced to a simple check for (self-)collisions in simulation. Considering the robot motion from the perspective of its configuration space further mitigates the problem of possible configuration ambiguities with respect to a certain end-effector pose when dealing with kinematically redundant robots. The implementation of the reaching- and exploring-phases is based on a randomly generated list of kinematically feasible and collision-free *scaffold*-configurations $\mathbf{S} = \{\mathbf{q}_1^{\text{scaf}}, \dots, \mathbf{q}_L^{\text{scaf}}\}$ of length L , where the feasibility check is performed for every $\mathbf{q}_i^{\text{scaf}}$ in simulation (see Section 2.2 on page 17). A scaffold configuration $\mathbf{q}_l^{\text{scaf}}$ is utilized as the goal configuration to be reached in the l -th reaching-phase and represents a randomly encounterable configuration to reach a specific target. The geometric configuration path for the l -th reaching-phase is simply defined by the one configuration waypoint $\mathbf{q}_l^{\text{scaf}}$ and practically realized as a joint-wise linear motion by the controller. At the same time, $\mathbf{q}_l^{\text{scaf}}$ acts as the starting configuration for the l -th exploration-phase, which is implemented as a time-unfolded joint-wise random sinusoidal path ${}_j\mathbf{e}_l \in \mathbb{R}^T$ that ranges $T \in \mathbb{N}$ joint

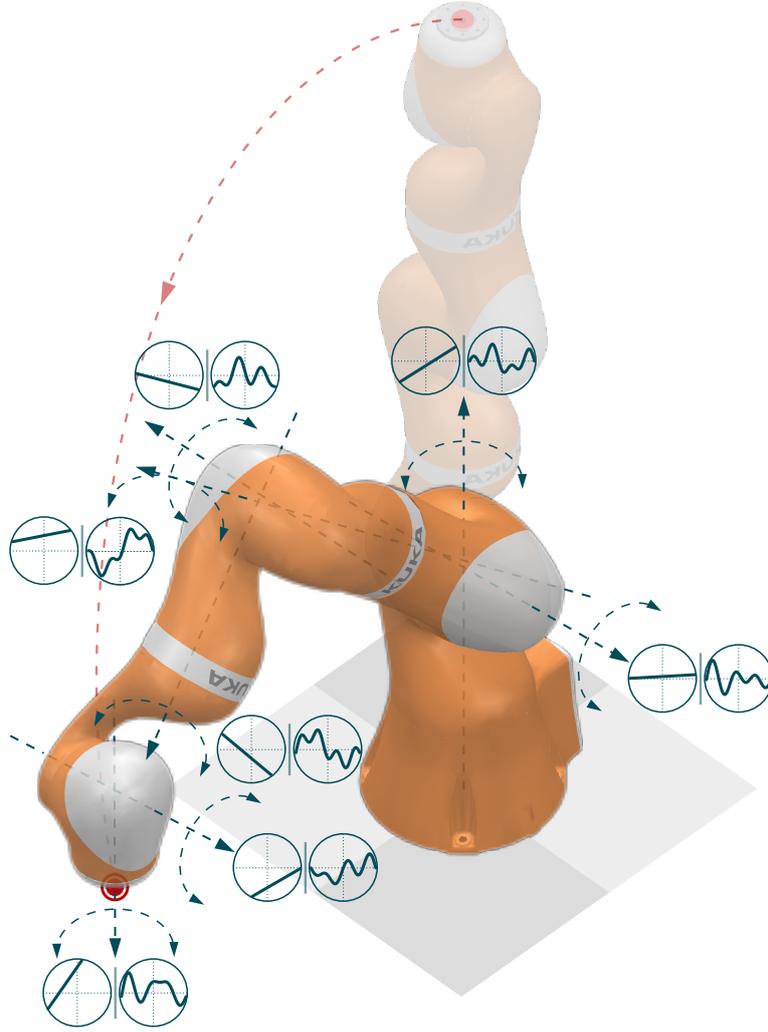


Figure 4.8: An illustration of the path geometry generated by RFMA (see Algorithm 4.10 on page 64). Following an initial path point, visualized here by the opaque schematic of the robot in the upright position, a random goal configuration is generated, shown by the articulated robot, which is reached via joint-wise linear motions, depicted by the linear segments in the spheres. This motion is ensured to result in an arbitrary but collision-free end-effector path, depicted by the dashed red line, via simulation. To provoke hysteresis in joint torque measurements and sample rich data regarding dynamic behavior during fine motion, combinations of low-amplitude sinusoidal are then superimposed upon the reached goal configuration as indicated by the sinusoidal joint motion profiles.

orientation waypoints and is parameterized by a discrete Fourier series of length 3

$${}_j e_l(t) = \sum_{k=1}^3 A_k^{\text{rand}} \sin(2\pi f_k^{\text{rand}} t + \varphi_k^{\text{rand}}), \quad (4.35)$$

$${}_j e_l = \{{}_j e_l(1), \dots, {}_j e_l(T^{\text{rand}})\}, \quad (4.36)$$

where ${}_j e_l(t)$ corresponds to the position of joint j at waypoint t of the joint-wise exploration-

phase path $j\mathbf{e}_l$. $A_n^{\text{rand}} \in [0.5, 3]$, $f_n^{\text{rand}} \in [-4, 4]$ and $\varphi_n^{\text{rand}} \in [0, 2\pi]$ are drawn uniformly and randomly from the respective intervals for each joint j and the k -th summand of the Fourier series. The path length, i.e. waypoint number, for every l -th sinusoidal exploration phase $\mathbf{e}_l = ({}_1\mathbf{e}_l, \dots, {}_{n=7}\mathbf{e}_l) \in \mathbb{R}^{(n=7) \times T^{\text{rand}}}$ following the l -th reach-phase is randomly determined by $T^{\text{rand}} \in [2000, 2500]$. This results in a geometric configuration path composed of the configuration-space waypoints

$$\mathbf{C} = \left\{ \underbrace{\mathbf{q}_1^{\text{scaf}}}_{7 \times 1}, \underbrace{\mathbf{e}_1}_{7 \times T^{\text{rand}}}, \underbrace{\mathbf{q}_2^{\text{scaf}}}_{7 \times 1}, \underbrace{\mathbf{e}_2}_{7 \times T^{\text{rand}}}, \dots, \underbrace{\mathbf{q}_L^{\text{scaf}}}_{7 \times 1}, \underbrace{\mathbf{e}_L}_{7 \times T^{\text{rand}}} \right\}, \quad (4.37)$$

where every pair $(\mathbf{q}_l^{\text{scaf}}, \mathbf{e}_l)$ designates a sinusoidal exploration \mathbf{e}_l around the l -th scaffold configuration $\mathbf{q}_l^{\text{scaf}}$. The joint-wise independently random sinusoidal motion is deliberately chosen to excite a wide range of dynamic state combinations that also provoke hysteretic behavior with oscillating motion; the parameter intervals are defined experimentally to resemble realistic short-range explorative motions and emphasize hysteretic nonlinearity by preventing the torque measurement hysteresis from saturating Figure 2.5 on page 16.

Due to the high dimensionality of robot dynamics, statistical dynamic modeling techniques in general and neural network-based approaches, in particular, rely heavily on the availability of a rich data set that reflects the data distributions encountered in practice, ideally even with some redundancy; if we assume the part of the Newton Euler equations Equation (3.31) on page 27 describing the torques acting on rigid bodies to minimally define the dynamic effects that are conventionally observable with joint sensors, we are confronted with a state-space that is composed of at least $28 \times n$ dimensions, i.e. one torque and one joint position along with its two time derivatives for n joints. Because it is intractable to exhaustively sample such a big state space, some restrictions and discretizations that balance the sampling efficiency and representative quality have to be introduced: From an application-oriented point of view, the usual Cartesian workspace that is transited by the end-effector of the robot within a specific setup can often be estimated with quite certainty. Under real-world conditions, the fixed position of environmental obstacles, pose-dependent reduced dexterity and typically well-defined spatial target areas (Çallar, 2018; Böttger et al., 2019) allow for the demarcation of an expected effective workspace W_{eff} , which is often smaller than the theoretical total reachable workspace $W_{\text{tot}} \supset W_{\text{eff}}$. Therefore one can accredit some plausibility to the reduction of the sample space to a practically relevant workspace, without sacrificing all too much sampling density on practically insignificant regions of the total workspace. In order to impose minimal prior workspace restriction such that $W_{\text{eff}} \ll W_{\text{tot}}$, the effective workspace W_{eff} is marked out as the kinematically reachable quarter-sphere in front of the robot, circumscribed by the positive x- and z-axis of the base coordinate system (see Figure 2.1 on page 13). As this still includes spatial points that are only reachable in fairly flexed or extended configurations that are undesirable in practice due to the nearness to kinematic singularities and low local dexterity (Kuhlemann et al., 2016), the workspace $W_{\text{samp}} \subset W_{\text{eff}}$ to be actually sampled is defined as the collection of configurations where the distance between the base and end-effector is less than 0.8 m. Given the range of motion of each joint (see Table 2.3 on page 14), the respective joint spaces are uniformly sampled to obtain a candidate configuration. This candidate configuration is then used together with the known forward kinematics of the robot to check in simulation whether the end-effector position for this configuration lies within the sample

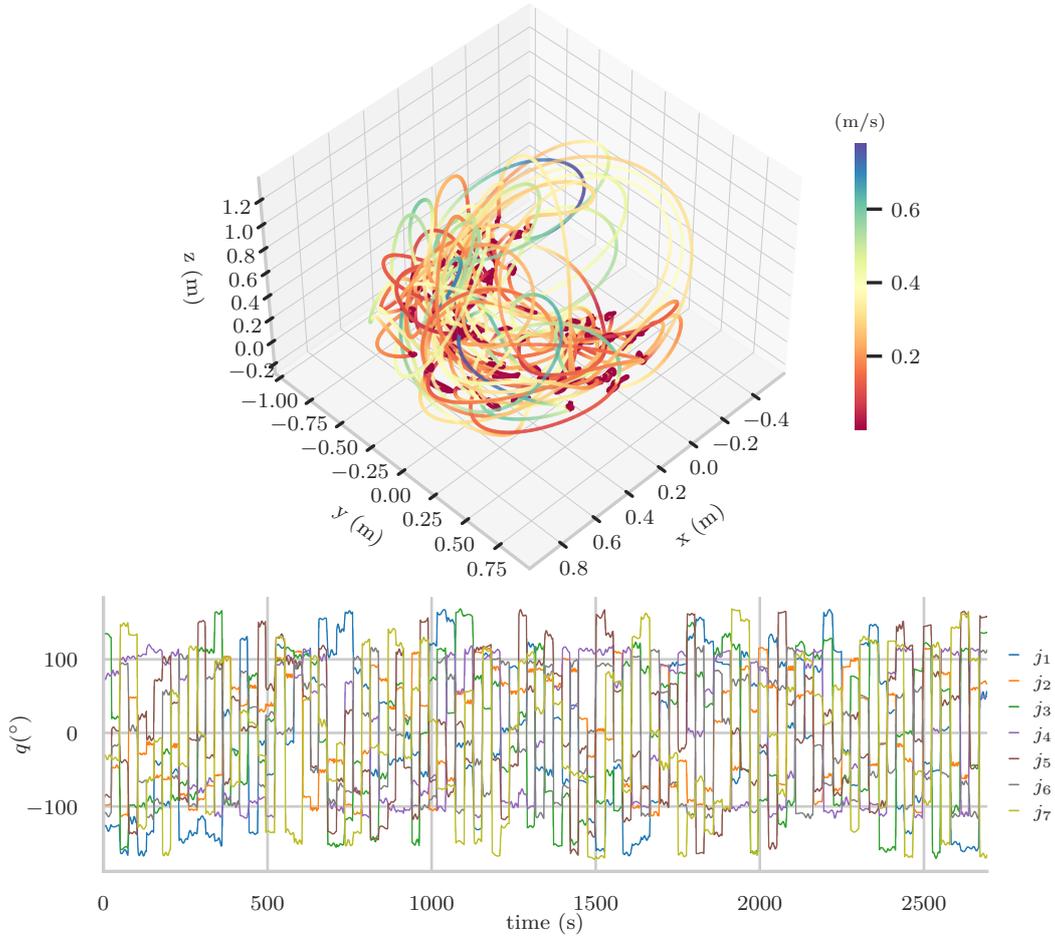


Figure 4.9: (*Top*) Spatial distribution of points traversed by the end-effector during the execution of a test motion based on a path generated by RMSA. The coloring of the points denotes the linear velocity of the end-effector at the respective location on the motion path. Note the poses depicted in dark-blue where a low-velocity and -amplitude fine motion is executed. (*Bottom*) The depiction of the trajectory of the test motion in configuration space. Please note the sinusoidal joint trajectories intermitted by steep, linear segments representing the point-to-point motion until the respective scaffold configuration is reached, around which the randomized oscillations are executed.

workspace and between the robot and itself or the environment occurs. This step is repeated until a desired number of feasible configurations – 150 per total path was chosen here – is reached, which is then utilized as the set of scaffold configurations. Due to the fact that dynamic data acquisition experiments in robotics are particularly time-intensive and lead to mechanical wear and tear, an effort for path length optimization has been made. After the target number of scaffold configurations is obtained, the distance in joint configuration space between two randomly determined successive scaffold configurations is possibly very large, leading to a joint-wise linear motion towards distant scaffold configurations which adds up to a drastic increase in path execution duration. Aside from resource constraints, this is may also be considered undesirable from the perspective of variation in the data

set, as during these linear configuration space interpolations with constant speed between two distant scaffold configurations results in frequent re-samplings of similar kinematic and dynamic states. Therefore, RFMA performs a prior spatial sorting of the scaffold configurations using a nearest-neighbor-search based on a kd-tree representation of the configuration space, where the scaffold configurations are sequentially ordered, reducing the total distance traveled during each reaching-phase.

As described, the final geometric path is lastly constructed by inserting an exploration sequence after each scaffold configuration. With an average of 2250 waypoints for each exploration-phase, this results in approximately 337500 waypoints for the total path and an execution duration of roughly 1 hour, assuming that 100 waypoints are passed through per second.

To allow for a variable speed of motion execution, RFMA, essentially being a purely geometric path planner, returns this path solely as a relative sequence of configurations as waypoints, which is then be translated by an external motion controller into a dynamic trajectory. The motion execution for the experiments carried out for this work was realized by a TCP/IP transmission of the waypoint data to the proprietary controller of the robot manufacturer (see Section 2.2 on page 17 and (Safeea and Neto, 2017)), where the waypoint transitions are executed as linear configuration space interpolations with soft real-time constraints by sequential execution of the buffered list of waypoints where the next waypoint is commanded as soon as the previous one is reached. The path execution dynamics are regulated in two ways. First, at the beginning of each reaching-phase, the proprietary control values of the joint-wise relative velocities and accelerations are randomly set between 60% and 90% with respect to the individual maximum joint velocities given in Table 2.3 on page 14. After completion of the reaching-phase, the relative velocities are randomly set between 10% and 30%.

Algorithm 4.10: The Random Fine Motion Generation Algorithm.

algorithm RFMA

require: goal_number_scaffold_configs,

number_joints, // $n \times 1$

joint_intervals, // $n \times 2$, rotation ranges for each joint

robot_collision_model,

environment_collision_model,

Cartesian_sample_range, // Cartesian space within which scaffold configs must lie

sinusoidal_parameters_range, // $n \times 2$, intervals for amplitude A , frequency f and
phase φ

sinusoidal_steps, // $t \times 1$

output: config_path_points // $P \times n$

config_path_points = []

scaffold_configs = []

while length(scaffold_configs) < goal_number_scaffold_configs **do**

for j = 1 **to** number_joints **do**

 random_config[j] = get_random_sample(joint_intervals[j], min),

```

                                                                    joint_intervals[j, max])
    forward_kinematics(random_pose)
    end_effector_location = get_end_effector_location(random_pose)
    collision = check_collisions(robot_geometry, environment_geometry)
end for
if end_effector_location <= Cartesian_sample_range and collision == false
    do
        scaffold_configs[i] = random_config
    end if
end while
sorted_scaffold_configs = empty(size(scaffold_configs))
sorted_scaffold_configs[1] = scaffold_configs[1]
for l = 1 to length(sorted_scaffold_configs) do
    query_config = sorted_scaffold_configs[l]
    remove query_config from scaffold_configs
    neighbor_config = kd_nearest_neighbor(query_config, scaffold_configs)
    remove neighbor_config from scaffold_configs
    sorted_scaffold_configs[l+1] = neighbor_config
    config_path_points[end+1] = neighbor_config
    for j = 1 to n do
        // sinusoidal =  $\sum_{k=1}^3 A_k^{\text{rand}} \sin(2\pi f_k^{\text{rand}} t + \varphi_k^{\text{rand}})$ 
        sinusoidals[i] = generate_random_sinusoidal_sum(
            sinusoidal_parameters_range, sinusoidal_steps)
        config_path_points[end+1..length(sinusoidal_steps)]
    end for
end for

return config_path_points

```

Through this motion generation strategy, a thorough coverage of the Cartesian as well configuration workspace is achievable (see Table 2.3 on page 14). At the same time, the coverage of a wide range of dynamic state combinations in the acquired data is ensured, which includes multiple permutations of simultaneously occurring joint positions, velocities, accelerations, motion directions as well as torques, yielding a comprehensive yet efficient sampling of the effective dynamic space Figure 4.11 on the following page.

4.4 Training Scheme

The neural networks are implemented in *PyTorch* (Paszke et al., 2019) and trained on a data set that is acquired by the execution of the trajectory shown in Figure 4.9 on page 63 that is commanded on the robot via the control scheme explained in Section 2.2 on page 17. The training is implemented to support CUDA API (Nickolls et al., 2008) and run in a cloud computing instance on a Tesla T4 GPU with 25 GB RAM, resulting in an average training duration of approximately 12 hours per network, depending on the parameter count of the trained architecture respectively. The obtained data set contains 804205 temporally sequential joint positions and torques. The position measurements are

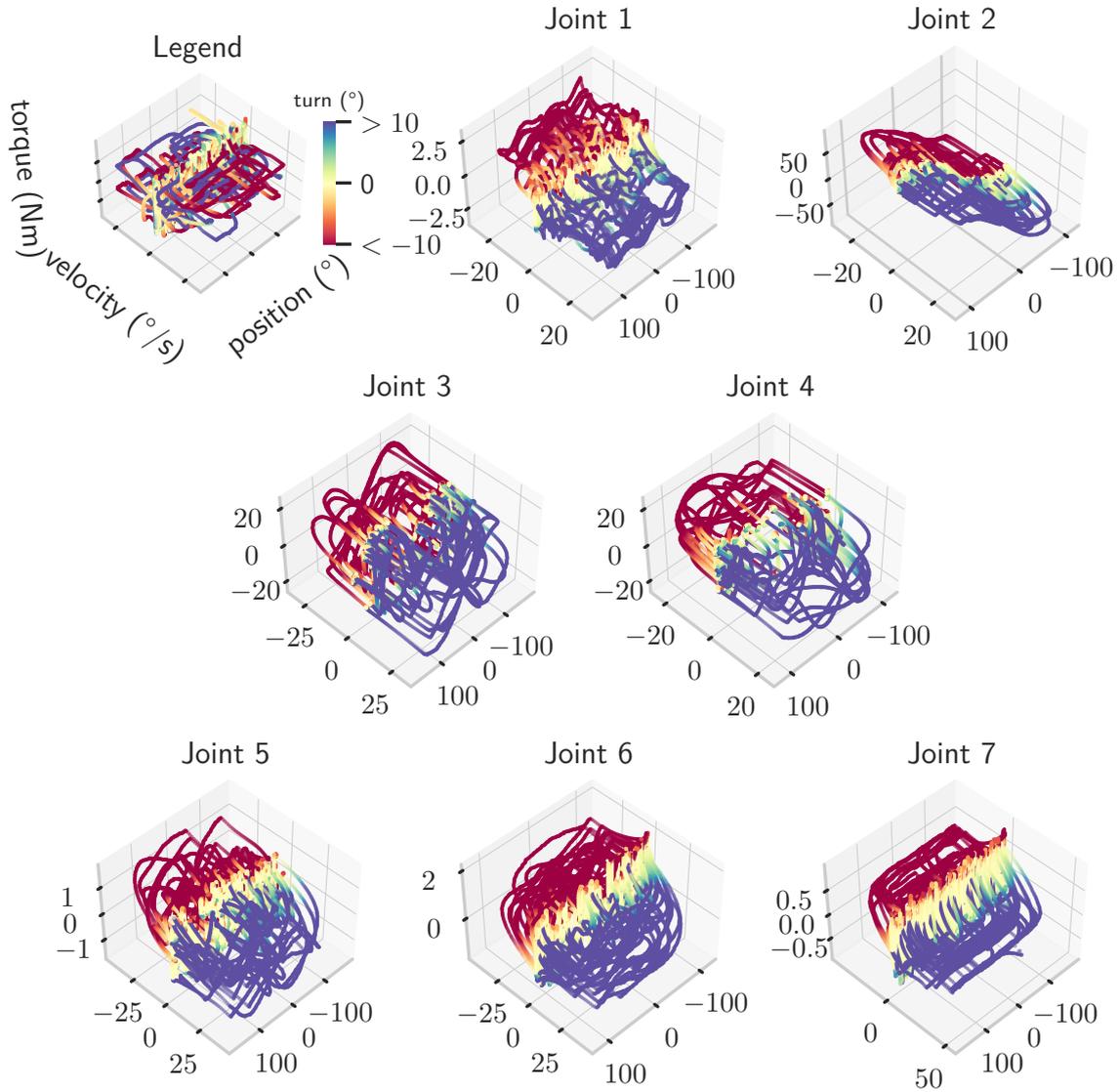


Figure 4.11: Coverage of the joint-wise dynamic ranges captured by the training data acquired during the execution of the test motion. The data are visualized in four dimensions as shown by the legend in the upper left corner: The z-axes denote the torques recorded for the respective joints, where the most distal joints have the smallest torque ranges and joints 2 to 4 the largest due to the position and orientation of the joints within the kinematic chain. The joint velocities are shown on the y-axes and are nearly uniformly distributed within the same ranges for each joint. The sampled joint positions are depicted on the x-axes where the distribution is likewise uniform, with the exception of joint 2, due to the restriction of the effective workspace on the positive x-axis of the world coordinate frame. The color-coding indicates the rotational displacement since the last directional reversal of that joint at the respective sampling time (see Section 4.2 on page 57).

numerically differentiated to obtain the respective velocities and accelerations, which are then, together with the noisy torque measurements, subjected to the same post-processing

pipeline described in Section 4.1 on page 47.

Due to the sampling rate of approximately 150 Hz in combination with the slow and steady joint velocities enforced by RFMA, there is a comparatively small temporal change in the measurements. This diminished temporal variation is deemed to be detrimental for the sequence modeling network architectures of the LSTM and Transformer, given the sampling rate and the history time windows of $T = 100$ and $T = 500$ used for the prediction of one set of momentary joint torques as defined before. As a simple increase of the time window can possibly exceed the feasible sequence for LSTM’s and exacerbate the vanishing gradient problem still affecting this architecture (Li et al., 2018), the data set is reduced by excluding every second sample, yielding 402102 samples in total. Employing a common train-test-split, the neural networks are trained on a random 80% of the data and tested on the remaining 20 %. With respect to the evaluated architectures that, instead of single inputs, accept an input sequence, the data set is partitioned into sub-sequences of length T , such that the shuffling affects only the order of the sub-sequences but not that of the samples within one such sub-sequence.

The optimization of the parameters of the respective networks is all performed using the adaptive learning rate optimizer AdamW (Loshchilov and Hutter, 2019) proposed to improve the generalization capability obtainable by the popular Adam optimizer (Kingma and Ba, 2017). The initial learning rate is determined empirically based on a series of preliminary trials and recommendations in the literature (Hitzler et al., 2019) and set as $\epsilon = 0.001$ for every network. During the course of training spanning 100 epochs, the adaptation of the learning rate via the optimizer is complemented by the learning rate decay scheduler *ReduceLROnPlateau* implemented in *PyTorch* that reduces the learning rate in case the training loss does not improve for two consecutive epochs, with the intent to guide the optimization out of local minima.

The training is performed based on the loss metric of the mean squared error

$$L_{\text{MSE}}(\boldsymbol{\tau}_{HYB}, \tilde{\boldsymbol{\tau}}_{meas}) = \frac{1}{m} \sum_i^m \|(\boldsymbol{\tau}_{HYB} - \tilde{\boldsymbol{\tau}}_{meas})_i\|^2, \quad (4.38)$$

between $\boldsymbol{\tau}_{HYB} = \boldsymbol{\tau}_{RBD} + \boldsymbol{\tau}_{NN}$, i.e. the sum of the downstream network predictions and upstream rigid body dynamics model estimates, and the processed torque measurements $\tilde{\boldsymbol{\tau}}_{meas}$. For the evaluation of the performance of the various neural network architectures proposed for the hybrid model (see Section 4.2 on pages 57–59), after training with the full feature space defined in Equation (4.28) on page 56, the network architectures are retrained as a) standalone networks without the input feature inclusion and output addition of the upstream estimates $\boldsymbol{\tau}_{RBD}$, b) without the inclusion of the rotational displacement encoding feature \boldsymbol{r} , and c) in case of the time-series networks LSTM and Transformer with different input sequence lengths $T = \{100, 500\}$. As the main benefit of the Transformer architecture is supposedly most apparent for long sequences exceeding the hypothetical capability of LSTM’s and the prediction delay due to recurrence is rather insignificant for shorter sequences, the Transformer is, however, only trained once on sequence lengths of $T = 500$ and with the full feature space.

5

Model Performance Evaluation

In pursuit of finding the optimal model architecture and configuration, the following evaluation intends to provide a answers for these questions:

- Which architectural base is best suited for the purpose of proprioceptive torque estimation and how does joint-torque estimation accuracy relate to contact force estimation?
- How does a parametric model based on rigid body dynamics perform at fine-scale, irregular motion?
- What is the benefit of the hybrid model architecture?
- Can temporal information be successfully leveraged to improve model performance? What is the importance of network recurrence in this regard?

The evaluation of the respective model performances is carried out in two ways. Initially, the different models are assessed with regards to their joint torque prediction accuracy based on the mean squared error obtained for the data set used for the training and testing of the hybrid models. This is accompanied by a discussion of the training error convergence behavior and generalization capability on the test data set. As the goal of robotic force control is the regulation of contact forces emerging between the robot and its environment or the interacting human respectively, one evaluation criterion is the model performance with respect to the accuracy with which the external interaction forces can be estimated, constituting the second evaluation approach. Experimentally, this is investigated by replicating external interaction forces with the attachment of known masses at the end-effector during the execution of a validation trajectory that is generated by the same proposed algorithm RFMA described in Section 4.3 on page 59 and comprises 42260 samples after exclusion of every second sample to improve temporal variation for the training of the sequence models as explained before. The validation trajectory is then repeatedly executed without as well as with machined masses weighing $m_{500} = 500$ g and $m_{1000} = 1000$ g that are attached at the end-effector respectively to emulate the presence of constant external downward force vectors $[0 \ 0 \ -g \ m_0 = 0]^\top$, $[0 \ 0 \ -g \ m_{500} = 4.905\text{N}]^\top$ and $[0 \ 0 \ -g \ m_{1000} = 9.81\text{N}]^\top$ respectively, where $g \approx 9.81\text{m/s}^2$. The evaluation in this regard is performed by computation of the linear external force via the geometric Jacobian of the manipulator as defined in Equation (3.11) on page 21 which is compared against the actual linear force acting upon the end-effector due to the attached mass.

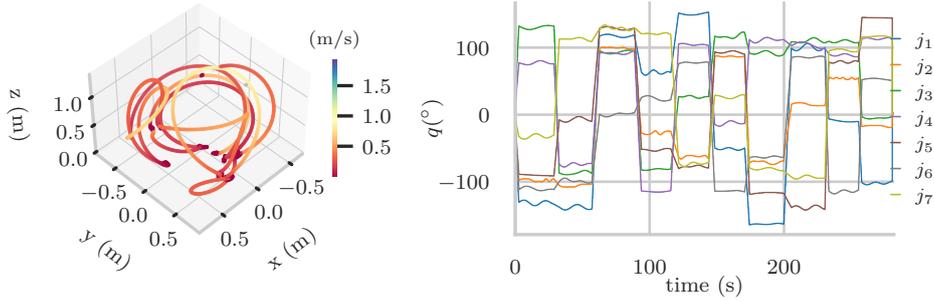


Figure 5.1: *Left*) Cartesian path traversed by the end-effector during the repeated execution of the validation trajectory motion with the respectively successive attachment of the test masses, beginning and ending at the joint configuration $\mathbf{q} = (0, 0, 0, 0, 0, 0, 0)$, which is equivalent to the Cartesian end-effector position (0 m, 0 m, 1.306 m). The coloring of the line denotes the linear velocity of the end-effector at the respective location on the motion path. *Right*) The excitation trajectory depicted in the configuration space, where each curve shows the temporal angular position of the respective joint. Note the presence of spatially restricted motions with slow motion as generated by RFMA,

5.1 Architectural Performance Comparison

The evaluation experiments are performed for every hybrid model combination

$$\boldsymbol{\tau}_{HYB} = \mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS} + f_{NN}(\mathbf{K}_b(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{k}) \boldsymbol{\Phi}_b^{LS}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{r}) \quad (5.1)$$

$$\text{with} \quad (5.2)$$

$$f_{NN} = \{(), f_{MLP-7}, f_{LSTM-2}, f_{LSTM-2-FCL}, f_{LSTM-2}, f_{TF}\}, \quad (5.3)$$

in order to elucidate the benefits gained from the architectural variations with regards to the estimation accuracy of the joint torques and external linear forces, where $f_{NN} = ()$ denotes the absence of a neural network component making the hybrid model equivalent to the backbone rigid body dynamics model abbreviated as RBD in the following. Furthermore, the analysis is carried out through an inexhaustive grid search by successively excluding parts of the input feature space and the auxiliary RBD predictions (see the end of Section 4.4 on page 65). The trained hybrid model configurations are listed along with the statistical estimation results in Tables 5.3 and 5.4 on page 74 and on page 75; the training loss curves are given for the base architectures in Section 5.1 on page 71.

Joint Torque Estimation

On the basis of the residual joint-wise training and test error results (see Table 5.3) obtained through the training of the individual hybrid model architectures and configurations, it is evident that those models that are trained with the full feature space, i.e. with the rotational displacement encoding feature \mathbf{r} (see Section 4.2 on page 57), have consistently achieved lower residual losses on both the training and test data sets than their counterparts trained without \mathbf{r} . On its own, this finding allows for a tentative confirmation of the benefit of this auxiliary input parameter. The positive effect of the hybrid model structure is even more

pronounced. With the exception of the LSTM-2-FCL trained on sequences of length 500, where the gain is insignificant, every case of omission of the rigid body dynamics support resulted in a significant reduction of the achievable MSE in comparison to the hybrid model counterparts. With respect to the MLP and LSTM-2 100 architectures, the relative mean prediction accuracy difference over all joints between the standalone neural networks and hybrid versions amounts to a magnitude of more than $1 \cdot 10^4\%$. This suggests the plethora of multiple conjectures, that are common in machine learning. Either, these architecture configurations are inherently unable to adapt to the type of motion data presented during training; the training data amount is insufficient, or the culprit lies in the hyperparameter configuration. As the performance can demonstrably be improved through the hybrid model structure and the rotational displacement encoding, and there is no significant overfitting noticeable as per the small train-test-error difference, the architectural properties are deemed to be causal in connection with the characteristics of the motion represented in the data set (see Section 4.3 on page 59). In view of related grey-box neural network-based error model learning research carried out on a robotic system that is dynamically similar to the one discussed here (Hitzler et al., 2019), the problem of underfitting of highly dynamic robotic motion data seems to be possibly a common phenomenon, necessitating to resort to the inclusion of some kind of model prior such as the identified rigid body dynamics model. The shortcomings of the standalone MLP’s can be further analyzed with respect to the joint-wise differences in the residual MSE, which show the highest errors for the joints with the largest torque range due to their structurally load-bearing location, especially in a robot that is mounted upright (see Figure 4.11 on page 66). This is most likely caused by the resulting high variability in the dynamic ranges of those joints, which cannot be encoded without prior information. The tentative reason for reportedly favorable results obtained by MLP’s for robotic force estimation (Kuhleemann, 2019) may be an increased sampling density of the robotic workspace but also a restriction of the modeling task to quasi-static states, where the input space is reduced to the number of joint positions of the robot. In such quasi-static cases, the influence of complex dynamic nonlinearities arising from intra-joint dynamics, especially during slow irregular motion as discussed in the previous chapters, is essentially removed, rendering the dynamic influences to be mostly due to static gravitation. Because the dynamic range that is targeted in this work and reflected in the training data is possibly at the interface between the quasi-static and dynamic states, a variety of mechanic phenomena are provoked that are not easily interpretable by a neural network without additional information that goes beyond the conventional temporal joint position derivatives. Conversely, the cause for the LSTM-2-FCL architectures to yield favorable MSE’s even without support from a model prior within a hybrid architecture or an additional rotational displacement encoding – although these two elements indeed lead to some improvements – may be the ability to leverage long-term temporal information over long distances due to the combination of recurrence by LSTM cells and regression by the added fully connected layer. Interestingly, the LSTM-2 architecture was not able to qualitatively achieve the same performance without integration within the hybrid model structure, indicating synergetic effects through the combination of LSTM’s with final fully connected layers. Accordingly, the hybrid LSTM model trained on sequences of 500 time-steps with rotational displacement encoding achieved the lowest training and test MSE.

Accepting the speculative benefits gained from time-series neural networks by the ex-

5 Model Performance Evaluation

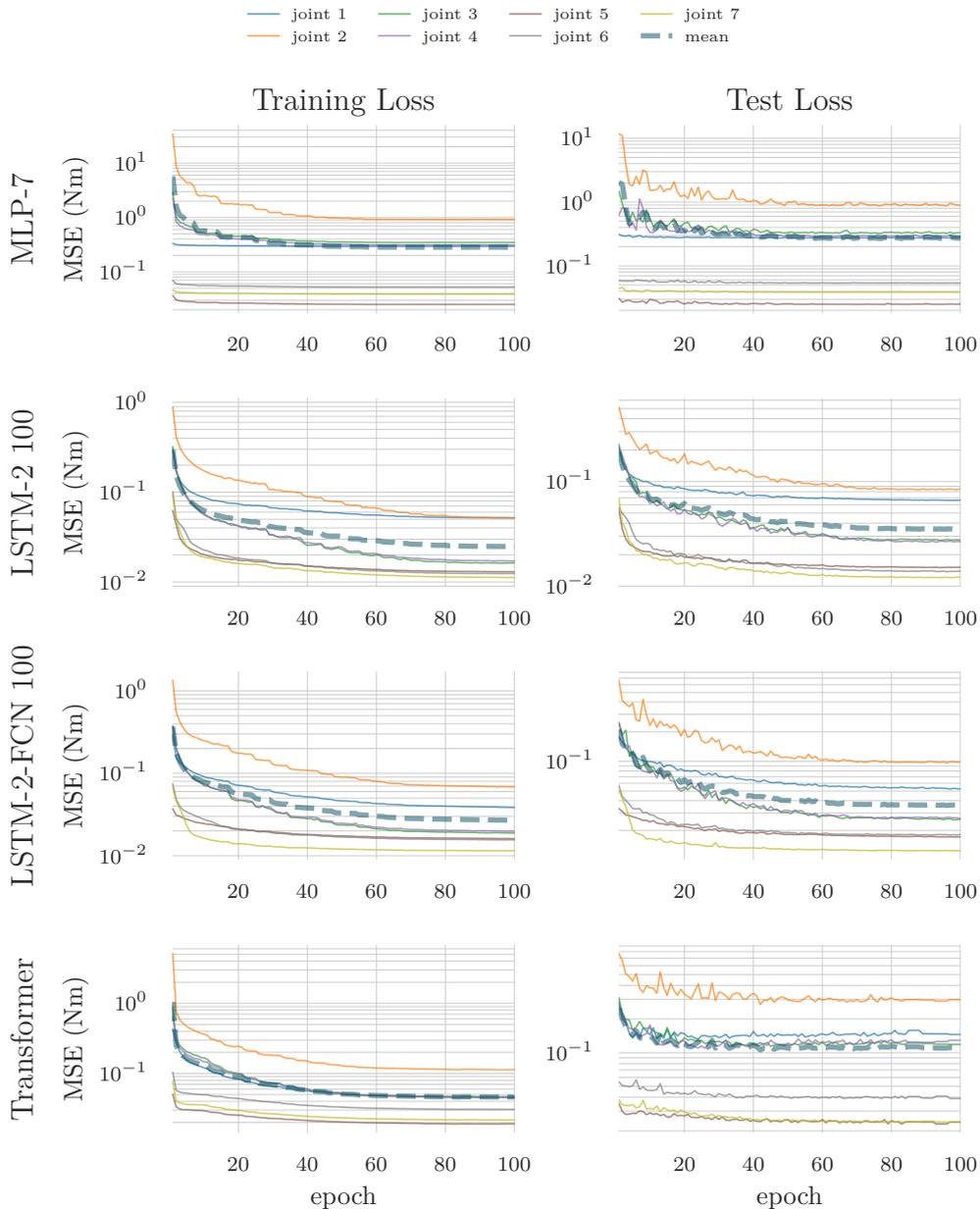


Figure 5.2: Development of the MSE between the inferences of the best performing configurations of the respective model architectures and post-processed measured joint torques from the training and test data sets over a course of 100 epochs. The training and test losses are depicted for every single joint respectively as well as a mean over all individual per joint estimates. The loss curves of the remaining models are not included either due to qualitative similarity to those shown or, immediate convergence after few epochs on high losses in the case of the bad-performing standalone neural networks without support from the rigid body dynamics estimates.

ample of LSTM’s, this begs the question of whether recurrence is ultimately necessary for the successful encoding of temporal information in robot dynamics modeling. The usual

indication for using Transformer networks over RNN’s is the practical sequence length limit up until which the latter can feasibly learn distant sequential relations, especially in the context of high-dimensional feature spaces, e.g. found in natural language processing. For this reason, the Transformer network is only evaluated on sequences of length $T = 500$, aiming to reveal possible model accuracy gains through long sequence information. Under the experimental and configurational specifications of this thesis, no such improvement could be derived from this architecture. In fact, the entirety of the hybrid recurrent models with rotational displacement encoding has lower MSE’s, while those without encoding are rather similar accuracy-wise, which leads to the possible conclusion that recurrence in combination with the proposed encoding may be equal or even superior compared to the Transformer.

The estimation errors obtained by the backbone RBD model on the same training data are significantly worse than those of the hybrid models that are, rather ironically, based on the RBD model and achieve their superior accuracy mainly through the prior information included within that physical parametric model. This is also to show, that the generalization capability of the RBD model is limited especially with respect to the dynamic properties of the training motion, leading to a stark discrepancy between its MSE on the neural network training data set and that on the data of the dynamic excitation experiment (see Section 4.1 on page 47). One can interpret this as an effect of unavoidable overfitting of the rigid body inertial parameters obtained on the excitation data on which they were determined by regression. Another possible explanation, however, is that the kinematic-dynamic state relations contained in the training data reflect proportionally higher influences of dynamic effects outside the framework of rigid body dynamics. Intuitively, this can be comprehended by the fact that the terms of the formula describing the rigid body dynamics (see Equation (3.5) on page 20) depending on the link velocities and accelerations have minimal influence during slow and generally steady motion, leaving only the gravitational term. As the dynamics are still not in a completely static state, the multitude of causally obscure and possibly hysteretical effects provoked by slow velocities and directional reversals, e.g. complex friction, actuator/transmission dynamics, etc., gain influence. This provides tentative evidence for the significant decrease in torque estimation performance of models based on a closed-form parametric dynamic description for applications within which such motions predominantly occur, although this can, in theory, be mitigated by more sophisticated model formulations and parameter identification schemes.

The temporal course of the joint torque predictions τ_{PRED} given by the individual model architectures in comparison with the filtered joint torque measurements $\tilde{\tau}_{meas}$ is given in the Appendix. From these visualizations, the qualitative superiority of the hybrid recurrent architectures with rotational encoding is particularly apparent from the near perfect congruence of the predictions with the measurements across all joints, especially in the case of LSTM-2-FCL 500.

Cartesian Force Estimation

The statistical accuracy with which the Cartesian forces due to the attached masses were estimated is given in Table 5.4 on page 75. The RMSE’s of the predicted force are derived from the observed deviation in the joint torques with respect to the model estimates using the relation between the end-effector contact Jacobian and the joint-wise torque deviations

as defined in Equation (3.11) on page 21. Considering the joint-wise MSE's obtained by the different model architectures and configurations on the neural network training data set, the results for the Cartesian force estimates indicate, that the joint-wise accuracy is comparatively weakly correlated with the actual accuracy with which the Cartesian forces can be inferred. For the validation trajectory without any attached weights, the discrepancy between the Cartesian force estimates of the different architectural configurations is not as pronounced as the one regarding the joint torque estimation, although a qualitative trend is noticeable, where the standalone neural network, especially the MLP's, still achieved the highest RMSE with the exception of the standalone LSTM-2-FLC 500 architecture, and the hybrid recurrent models achieved the lowest RMSE lead by the hybrid LSTM-2-FCL 500 with rotational encoding. The same applies to the standard deviations. Interestingly the spatial distribution of the RMSE force error vector of the MLP's have components largely varying in magnitude, which is not seen with the other architectures. With an increase in the mass of the attached weights, there is an increase in the z-component of the force error vectors that is equal to the weight force of the respective mass. Likewise, the standard deviation of the prediction error assumes approximately the force vector due to the attached masses. The architectures that are relatively less affected by this are the MLP's, which show continuously high RMSE's, and the RBD model, where. although the difference in RSME between the weights follows the same rule, the standard deviation of the error vectors is higher for no attached weights.

Table 5.3: Average residual MSE between the joint torque predictions τ_{HYB} and the post-processed torque measurements τ_{HYB} of the same time step in **Nm** after 100 epochs of training on the data set recorded with no attached weights (see Equation (4.38) on page 67). For reference, the MSE for the backbone model predictions τ_{RBD} (see Section 4.1 on page 46) are also included under the name RBD. The results are given as averages for each joint. The names of the network configurations follow the naming convention introduced in Sections 3.4 and 4.2 on page 31, on page 58 and on page 59. The additional number behind the time-series models designates the sequence lengths on which the respective models were trained

Architecture	MSE Training Error (Nm)							MSE Test Error (Nm)								
	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Average	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6	Joint 7	Average
MLP-7	0.2938	0.9160	0.3455	0.3115	0.0251	0.0523	0.0387	0.2833	0.2747	0.8845	0.3309	0.2916	0.0252	0.0054	0.0385	0.2644
MLP-7 no r	0.3937	201.8147	17.8439	76.9489	0.3541	0.2857	0.0049	65.7351	0.3937	206.3667	17.9016	75.5315	0.3590	0.2899	0.0479	66.0006
MLP-7 no r , τ_{RBD}	1.0049	202.5057	179.1369	77.0004	0.4084	0.5081	0.1949	65.8228	0.9836	207.1595	178.9357	75.4605	0.4061	0.5070	0.196	66.2355
Transformer 500	0.4293	0.1134	0.0459	0.0455	0.0190	0.0306	0.0216	0.0461	0.1432	0.3179	0.1222	0.1284	0.0236	0.0390	0.0244	0.1142
LSTM-2 100	0.0518	0.0517	0.0163	0.0171	0.0129	0.0124	0.0111	0.0248	0.0663	0.0849	0.0274	0.0266	0.0151	0.0138	0.0121	0.0352
LSTM-2 100 no r	0.0593	0.0952	0.0207	0.0208	0.0214	0.0252	0.0234	0.0380	0.0730	0.1297	0.0321	0.0338	0.0239	0.0291	0.0262	0.0497
LSTM-2 100 no r , τ_{RBD}	0.6039	82.8548	12.9735	12.1421	0.2604	0.4004	0.2052	15.6344	0.6095	80.8355	12.3105	11.5995	0.2737	0.4068	0.2058	15.1774
LSTM-2-FCL 100	0.0385	0.0683	0.0188	0.0198	0.0161	0.0155	0.0115	0.0270	0.0526	0.0991	0.0258	0.0265	0.0171	0.0181	0.0124	0.0360
LSTM-2-FCL 100 no r	0.0770	0.2207	0.0586	0.0567	0.0270	0.0495	0.0413	0.0759	0.1224	0.2955	0.1092	0.0777	0.0315	0.0595	0.0487	0.1064
LSTM-2-FCL 500	0.0292	0.0507	0.0196	0.0199	0.0159	0.0133	0.0119	0.0230	0.0400	0.0628	0.0253	0.0271	0.0168	0.0141	0.0124	0.0284
LSTM-2-FCL 500 no r	0.0317	0.1177	0.0237	0.0249	0.0284	0.0265	0.0300	0.0759	0.0410	0.1380	0.0309	0.0319	0.0293	0.02731	0.0319	0.1064
LSTM-2-FCL 500 no r , τ_{RBD}	0.0713	0.1468	0.0648	0.0698	0.0690	0.2011	0.1816	0.1150	0.0754	0.1842	0.0798	0.0954	0.0740	0.2200	0.1825	0.1302
RBD	1.9499	25.2967	0.8434	2.7134	0.3373	0.0963	0.0442	4.4687	2.0882	23.1839	0.0991	2.1749	0.4097	0.0985	0.0471	4.0145

Table 5.4: Estimation of the external linear forces due to the attachment of the test weights $m = \{0\text{g}, 500\text{g}, 1000\text{g}\}$ obtained by the different model architectures and input space variations. The results are given for each architectural variation as a root mean squared error along with the standard deviation of the prediction error which are both listed as force vectors in \mathbf{Nm} , designating the errors and standard deviations with respect to three spatial dimensions $[x \ y \ z]$. The names of the network configurations follow the naming convention introduced in Sections 3.4 and 4.2 on page 31, on page 58 and on page 59. The additional number behind the time-series models designates the sequence lengths on which the respective models were trained

Mass (g)	MLP-7	MLP-7 no r	MLP-7 no r, τ_{RBD}
0	[5.54 6.09 5.377]	[11.77 6.00 5.59]	[5.54 11.59 16.25]
	[7.77 8.45 7.29]	[7.74 8.43 7.62]	[16.74 16.79 22.60]
500	[7.56 4.27 7.38]	[10.17 4.13 7.48]	[7.42 8.30 19.33]
	[8.06 6.62 9.08]	[7.95 6.36 8.89]	[16.51 10.20 14.07]
1000	[8.28 4.72 10.91]	[8.13 4.60 11.11]	[9.36 7.78 22.31]
	[8.54 7.25 12.61]	[8.41 7.02 12.63]	[15.32 9.72 16.66]
	Transformer 500	LSTM-2 100	LSTM-2 100 no r
0	[0.24 0.25 0.33]	[0.09 0.09 0.14]	[0.11 0.12 0.16]
	[0.36 0.35 0.43]	[0.13 0.14 0.21]	[0.16 0.16 0.23]
500	[0.95 0.73 4.52]	[0.96 0.64 4.49]	[0.97 6.57 4.54]
	[1.11 1.08 5.00]	[0.98 0.95 5.01]	[1.02 1.01 5.00]
1000	[1.74 1.25 9.12]	[1.74 1.07 9.07]	[1.77 1.12 9.11]
	[1.95 1.83 9.94]	[1.76 1.62 9.96]	[1.77 1.72 9.96]
	LSTM-2 100 no r, τ_{RBD}	LSTM-2-FCL 100	LSTM-2-FCL 100 no r
0	[1.98 1.92 2.44]	[0.22 0.28 0.28]	[0.18 0.18 0.25]
	[2.82 2.84 4.41]	[0.32 0.32 0.41]	[0.26 0.26 0.36]
500	[3.54 2.03 5.00]	[1.01 0.79 4.52]	[0.95 0.69 4.50]
	[4.54 3.15 6.82]	[1.11 1.23 5.03]	[1.08 1.03 5.00]
1000	[4.20 2.45 9.23]	[1.82 1.20 9.06]	[1.75 1.17 9.05]
	[5.06 3.84 11.02]	[1.89 1.86 9.96]	[1.81 1.17 9.94]
	LSTM-2-FCL 500	LSTM-2-FCL 500 no r	LSTM-2-FCL 500 no r, τ_{RBD}
0	[0.08 0.09 0.12]	[0.11 0.12 0.15]	[0.15 0.16 0.21]
	[0.12 0.12 0.17]	[0.15 0.16 0.21]	[0.21 0.23 0.29]
500	[0.96 0.68 4.49]	[0.91 0.68 5.53]	[1.26 0.94 4.79]
	[1.07 0.98 5.00]	[1.04 1.02 5.01]	[1.68 1.36 5.28]
1000	[1.74 1.25 9.12]	[1.67 1.28 9.13]	[1.86 1.36 9.38]
	[1.90 1.81 9.96]	[1.80 1.86 9.96]	[2.21 1.97 10.13]
	RBD		
0	[1.53 1.74 1.93]		
	[5.11 5.85 6.30]		
500	[0.90 0.83 4.77]		
	[1.11 1.19 5.03]		
1000	[1.65 1.24 9.26]		
	[1.80 1.85 9.91]		

6

Conclusion

Throughout this work, the robotic force control problem in view of newly emerging physical human-robot interaction applications has been investigated with regards to the sub-problem of the observability of external contact forces in order to contribute to the applicability of robotic systems in highly interactive and force-sensitive usage scenarios, such as in health-care or collaborative robotics. Instead of utilizing exteroceptive sensorization in the form of force/torque sensors that are locally attached e.g. at the robot's end-effector for the direct measurement of contact forces, a proprioceptive approach was taken by employing the joint position and torque sensing capabilities available with recent collaborative robots such as the KUKA LBR iiwa 14, which was investigated in this thesis (see Chapter 2 on page 12).

This has the benefits of reduced system complexity compared to additional external sensors and a larger perceptive field with respect to external mechanical contacts that might occur anywhere on the robot housing and can remain undetected by local sensors. To infer external physical influences acting on the robot due to intentional interactions or collisions, the force observation task is cast as a dynamic modeling problem to determine the expected dynamic state reflected in the joint torque measurements for a particular kinematic state, i.e. pose and motion. The primary focus of this thesis was to develop a dynamic model that targets a specific dynamic range, which was motivated as follows: Conventionally, in collaborative scenarios, the robotic motion path is composed of rather long and directionally consistent changes in configuration space of medium velocity, which prevents the induction mechanical dynamic phenomena that predominantly occur during at slow velocities and frequent directional reversals within small configuration space distances due to complex effects, e.g. joint flexibility and nonlinear friction which exhibit pronounced hysteretical behavior that is otherwise not provoked. These perturbing causes have lead to the insufficiency of proprioceptive external force estimation for highly force-sensitive applications, as they affect pertain to joint-level sensors.

This has prevented this sensing scheme from being employed as a primary force estimation strategy for functionally and safety-wise force-critical applications such as in the medical field. The presence of joint torque measurement hysteresis was experimentally demonstrated in Figure 2.5 on page 16, while the problems this is causing for the parametric model-based external torque estimation have been presented in Chapter 5 on page 68. As many proprietary force estimation methods implemented by robot manufacturers are representations of the class of disturbance observers using parametric inverse dynamic models

(Haddadin, De Luca, and Albu-Schäffer, 2017), this poses an important problem for their applicability in scenarios with high estimation accuracy requirements and adversarial dynamic conditions with respect to the modeling capabilities of parametric inverse dynamics approximations.

Because of the inaccessibility of the actual parametric model used by the robot controller due to manufacturer confidentiality and general uncertainty about the accuracy and status of calibration on the actual unit used during experimentation, a classical identification of the inertial parameters was conducted to be used within the Newton-Euler formulation of rigid body dynamics as a parametric model specific to the given robot (Atkeson, An, and Hollerbach, 1986); although this is not equivalent to the actual approach used by the robot manufacturer, which implements a momentum-observer in order to also avoid the model’s dependence on joint accelerations due to difficulties in the robust retrieval of this quantity by real-time numerical differentiation. As this was not a direct concern for the basic investigations conducted in this thesis with regards to the theoretical upper boundaries reachable for dynamic model accuracy, and the post-processing of noise-amplified acceleration measurements was feasible, and due to the hypothesis that the consideration of accelerations would yield a more comprehensive model representation, this first approach was taken (see Section 3.3 on page 24). The observation of the shortcomings of this technique during experimentation motivated the search of alternative dynamic modeling methods, for which neural networks were chosen due to their inherent universal approximation property as well as their capability to infer a range of dynamic effects simply from data, possibly exceeding that which can be included in any closed model-formulation in spite of extensive domain knowledge.

In order to provide a suitable database that captures the targeted dynamic and motoric range of irregular fine-scale movements occurring in a large variety of configurations, the algorithm RFMA was developed during the course of this thesis to randomly generate a sequential geometric motion path in configuration space that was inspired by the manipulation of medical ultrasound transducers, representing a possible motion to be replicated by a robotic agent for the automated execution of this medical procedure Section 4.3 on page 59. The generated random paths are composed of slow joint-wise linear motions corresponding to the human arm motion during the initial translation of the ultrasound probe onto a specific anatomic target site. This movement phase is followed by a randomized superposition of low amplitude and rather high frequency parametric sinusoidal joint motion profiles to emulate explorative motions to find a particular sonographic visualization plane. With this motion component, it was at the same time intended to provoke the nonlinearities and hysteretical behavior in the actuation and torque sensors.

Using proprioceptive data obtained during the execution of the motion path yielded by RFMA, the training of a parallel multilayer perceptron (Yilmaz et al., 2020) was conducted to infer the joint torque levels from the proprioceptive motion observations. The initial failure of this approach, yielding results drastically worse than those obtained by the parametric model, which is speculated to be caused by the high dimensionality of the feature space in spite of comparatively minimal variations of many features such as the relatively constant velocities and nearly absent accelerations during steady motions produced by RFMA. A further reason for the incapability to learn from the data, except for the ever-present possibility of an insufficiently large and rich data set, is hypothesized to lie

in the temporal dependency of the hysteretical effects, which could not be captured by the presentation of momentary kinematic-dynamic relations. This challenge was approached by considering the robot dynamics as essentially being a time-series process where some causally undefined effects are responsible for the inadequacy of parametric modeling for the specific motion type exhibiting long-term temporal dependencies. Accordingly, an LSTM architecture was investigated (Rueckert et al., 2017), which yielded unsatisfactory results on the data set as well.

Given the difficulties in obtaining a serviceable model for the targeted motion class and the contrasting shortcomings of the parametric and neural network-based modeling, where the former is inherently incapable of representing dynamics outside of its model formulation and the latter is incapable of extracting sufficient model information from the data, the idea to combine both model architectures in a hybrid structure was pursued Equation (4.3) on page 46 by implementing the investigated neural networks as a downstream mode component following the initial estimates made by the parametric rigid body dynamics model 1 Section 4.2 on page 56. The hybridization of both model approaches led to a significant improvement in modeling accuracy on the RFMA data set over the previously investigated standalone techniques, led by the hybrid LSTM architecture. To further optimize the performance, a novel positional encoding based on the rotational displacement of a respective joint since the last reversal of its motion direction was developed to be concatenated with the input features of the kinematic observations, providing a simple quantity that can numerically express the transition phase on the hysteresis loops associated time-dependent mechanical effects. Although this auxiliary feature on its own did not improve the performances, the experimental data indicate an additional benefit gained from the encoding within the respective hybrid models.

With respect to the implementation of the hybrid model architectures for real-time applications, a solution for the robust online estimation of the joint velocities, accelerations, and torques has to be worked on. While the velocities and accelerations can be determined comparatively easily with conventional methods of low-pass filtered differentiation, or even be entirely omitted if the experimental data suggests an insignificant influence of the already minimal magnitude of velocity and acceleration during slow movements, the joint torque measurements are deteriorated by characteristic torque ripples that vary depending on the joint velocity. Although this can be dealt with via offline post-processing and case-individual filter tuning, during real-time operation, this appears to be a problem. Using a similar approach to the one proposed in (Chawda and Niemeyer, 2017), this may be tackled by as a variable notch filter realized as an experimentally determined lookup table defining the filter parameters in relation to the momentary joint velocities that can be estimated more robustly by filtered temporal differentiation of the joint positions.

Considering the translation of the joint torque estimation accuracy on the inference of external Cartesian contact forces, a series of validation experiments were conducted by attaching known masses at the robot end-effector and commanding a test motion likewise generated by RFMA. The fundamental finding is that the inference of the external Cartesian forces due to the weight force of the attached mass during motion was possible with an RMSE and that is approximately equal to the weight force of the respectively attached mass, meaning that the absence of any attached weight could be predicted by most of the model architectures with good accuracy and, e.g. the weight force of approximately

9.81 N for the attached mass of 1000 g was predicted with both an RMSE and standard deviation of the estimation error of circa 9 N. To determine the reason for this dependency of the estimation error on the applied external force, the conduction of further validation experiments is indicated to reproduce the model error behavior. A tentative causality that is proposed for this rather interesting result is that the dynamic effect imposed upon the robot by the weight is not significant enough to be captured by the joint torque sensors during the slow and irregular motion, where measurement noise and the influence of gravitational torques, as well as the nonlinear friction and sensor hysteresis, predominate. Combined with momentary configuration and the orientation-dependent perceptive limitation of the uniaxial joint torque sensors, this may lead to an alteration of the dynamic state that is simply not observable during motion. This assumption is further backed by the qualitatively similar RMSE obtained by the RBD model, which is exempt from any presumable effects of data maladaptation that can be suspected for the neural network components. A general task for the development of this work is the conduction of an extensive, systematic, and individualized optimization of the network hyperparameters.

In order to either confirm or dismiss this conjecture, the future research work to be carried out on this topic has to include an augmentation of the training data set to incorporate a higher sampling density regarding both the configuration space as well as the quasi-static range of the respective dynamic spaces of the joints. This extended data set may then be used to investigate whether further training can improve the force estimation, or the proposed models in their current training state may even be capable of accurately predicting joint torques and Cartesian external forces under quasi-static conditions. Another aspect to be considered is the further exploration of proprioceptively observable features that may guide the model inference capabilities. A possible approach in this regard, which has been contemplated upon during the work on this subject, is the doubling of the joint positions using the motor position as well as the link side position data to achieve some degree of informational redundancy with respect to the deflection of the flexible joints under the influence of external forces in addition to the joint torque readings.

From a practical point of view, however, the estimation of Cartesian forces should not necessarily be regarded as the only functional benefit to be gained from the general method of model-based proprioceptive dynamic disturbance observation. The hybrid model architecture with the obtained joint-wise torque estimation accuracy may be exceptionally well-suited for the identification of non-Cartesian external influence such as interaction torques or as a redundant mechanical disturbance system together with locally highly accurate exteroceptive sensorization at places of probable contact.

Bibliography

Albu-Schäffer, A., Haddadin, S., Ott, C., Stemmer, A., Wimböck, T., and Hirzinger, G. (2007). The DLR Lightweight Robot – Design and Control Concepts for Robots in Human Environments. *INDUSTRIAL ROBOT-AN INTERNATIONAL JOURNAL* *34*, 376–385. URL: <https://elib.dlr.de/51178/>.

Albu-Schäffer, A., Ott, C., and Hirzinger, G. (2007). A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research* *26*, 23–39. DOI: 10.1177/0278364907073776.

Alcocer, A., Robertsson, A., Valera, A., and Johansson, R. (2003). Force estimation and control in robot manipulators. *IFAC Proceedings Volumes 36. 7th IFAC Symposium on Robot Control (SYROCO 2003)*, Wroclaw, Poland, 1-3 September, 2003, 55–60. ISSN: 1474-6670. DOI: [https://doi.org/10.1016/S1474-6670\(17\)33369-4](https://doi.org/10.1016/S1474-6670(17)33369-4).

Allgeier, T and Evans, W.T. (1995). Mechanical Hysteresis in Force Transducers Manufactured from Precipitation-Hardened Stainless Steel. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* *209*, 125–132. DOI: 10.1243/PIME\PROC_1995_209_132_02.

Althoefer, K., Liu, H., Puangmali, P., Zbyszewski, D., Noonan, D., and Seneviratne, L.D. (2010). Force sensing in medical robotics. *Mechatronics in Action*, Springer, pp. 157–172.

An, C.H., Atkeson, C.G., and Hollerbach, J.M. (1985). Estimation of inertial parameters of rigid body links of manipulators. In *1985 24th IEEE Conference on Decision and Control*, pp. 990–995. DOI: 10.1109/CDC.1985.268648.

Atkeson, C.G., An, C.H., and Hollerbach, J.M. (1986). Estimation of Inertial Parameters of Manipulator Loads and Links. *The International Journal of Robotics Research* *5*, 101–119. DOI: 10.1177/027836498600500306.

Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schaeffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G., et al. (2010). The KUKA-DLR Lightweight Robot arm - a new reference platform for robotics research and manufacturing. In *ISR 2010 (41st International Symposium on Robotics)* and *ROBOTIK 2010 (6th German Conference on Robotics)*, pp. 1–8.

Böttger, S., Çallar, T.-C., Schweikard, A., and Rückert, E. (2019). Medical robotics simulation framework for application-specific optimal kinematics: Current Directions in Biomedical Engineering *5*, 145–148. URL: <https://doi.org/10.1515/cdbme-2019-0037>.

Bibliography

- Calandra, R., Ivaldi, S., Deisenroth, M.P., Rueckert, E., and Peters, J. (2015). Learning inverse dynamics models with contacts. In 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 3186–3191. DOI: 10.1109/ICRA.2015.7139638.
- Çallar, T.-C. (2018). Design of a Simulation Framework for the Exploration of Kinematic Structures for Robotic Ultrasound Imaging. In. URL: <https://cps.unileoben.ac.at/2018/11/11/tolga-can-callar-design-of-a-simulation-framework-for-the-exploration-of-kinematic-structures-for-robotic-ultrasound-imaging/>.
- Chang, W.H. and Kim, Y.-H. (2013). Robot-assisted therapy in stroke rehabilitation. *Journal of stroke* 15, 174.
- Chawda, V. and Niemeyer, G. (2017). Toward torque control of a KUKA LBR IIWA for physical human-robot interaction. In 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6387–6392. DOI: 10.1109/IROS.2017.8206543.
- Chiaverini, S. (2019). Redundant Robots. *Encyclopedia of Systems and Control*, (Baillieul, J. and Samad, T., eds.). London: Springer London, pp. 1–10. ISBN: 978-1-4471-5102-9. DOI: 10.1007/978-1-4471-5102-9_173-2. URL: https://doi.org/10.1007/978-1-4471-5102-9_173-2.
- Craig, J. (2005). *Introduction to Robotics: Mechanics and Control*. Pearson/Prentice Hall.
- Cutkosky, M.R. and Provancher, W. (2016). Force and tactile sensing. *Springer Handbook of Robotics*, Springer, pp. 717–736.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* 2, 303–314.
- Dario, P., Guglielmelli, E., Allotta, B., and Carrozza, M.C. (1996). Robotics for medical applications. *IEEE Robotics & Automation Magazine* 3, 44–56.
- De Luca, A., Albu-Schaffer, A., Haddadin, S., and Hirzinger, G. (2006). Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. In 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1623–1630. DOI: 10.1109/IROS.2006.282053.
- De Luca, A., Schroder, D., and Thummel, M. (2007). An Acceleration-based State Observer for Robot Manipulators with Elastic Joints. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 3817–3823. DOI: 10.1109/ROBOT.2007.364064.
- Denavit, J. and Hartenberg, R.S. (1955). A kinematic notation for lower-pair mechanisms based on matrices.
- Duan, X., Taurand, S., and Soleimani, M. (2019). Artificial skin through super-sensing method and electrical impedance data from conductive fabric with aid of deep learning. *Scientific reports* 9, 1–11.

Bibliography

- Díaz Ledezma, F. and Haddadin, S. (2017). First-order-principles-based constructive network topologies: An application to robot inverse dynamics. In 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), pp. 438–445. DOI: 10.1109/HUMANOIDS.2017.8246910.
- Elman, J.L. (1990). Finding structure in time. *Cognitive science* *14*, 179–211.
- Featherstone, R. (1987). Inverse dynamics—the recursive Newton-Euler method. *Robot Dynamics Algorithms*, Springer, pp. 65–77.
- Featherstone, R. and Orin, D.E. (2016). Dynamics. *Springer Handbook of Robotics*, Springer, pp. 37–66.
- Gautier, M. and Khalil, W. (1988). On the identification of the inertial parameters of robots. In Proceedings of the 27th IEEE Conference on Decision and Control, 2264–2269 vol.3. DOI: 10.1109/CDC.1988.194738.
- Gautier, M. and Khalil, W. (1992). Exciting trajectories for the identification of base inertial parameters of robots. *The International journal of robotics research* *11*, 362–375.
- Gaz, C., Flacco, F., and De Luca, A. (2014). Identifying the dynamic model used by the KUKA LWR: A reverse engineering approach. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1386–1392. DOI: 10.1109/ICRA.2014.6907033.
- Gers, F., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with LSTM. In 1999 Ninth International Conference on Artificial Neural Networks ICANN 99. (Conf. Publ. No. 470), vol. 2, 850–855 vol.2. DOI: 10.1049/cp:19991218.
- Giuliani, F., Hasan, I., Cristani, M., and Galasso, F. (2021). Transformer Networks for Trajectory Forecasting. In 2020 25th International Conference on Pattern Recognition (ICPR), pp. 10335–10342. DOI: 10.1109/ICPR48806.2021.9412190.
- Glad, T. and Ljung, L. (2018). *Control theory*. CRC press, p. 8.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Graves, A. (2012). Supervised Sequence Labelling with Recurrent Neural Networks. *Studies in computational intelligence*. Berlin: Springer. DOI: 10.1007/978-3-642-24797-2. URL: <https://cds.cern.ch/record/1503877>.
- Hacksel, P. and Salcudean, S. (1994). Estimation of environment forces and rigid-body velocities using observers. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 931–936 vol.2. DOI: 10.1109/ROBOT.1994.351233.
- Haddadin, S. and Croft, E. (2016). Physical human–robot interaction. *Springer handbook of robotics*, Springer, pp. 1835–1874.

Bibliography

- Haddadin, S., De Luca, A., and Albu-Schäffer, A. (2017). Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics* *33*, 1292–1312. DOI: 10.1109/TR0.2017.2723903.
- Hägele, M., Nilsson, K., Pires, J.N., and Bischoff, R. (2016). *Industrial robotics*. Springer handbook of robotics, Springer, pp. 1385–1422.
- Hammer, B. (2000). On the approximation capability of recurrent neural networks. *Neurocomputing* *31*, 107–123. ISSN: 0925-2312. DOI: [https://doi.org/10.1016/S0925-2312\(99\)00174-5](https://doi.org/10.1016/S0925-2312(99)00174-5).
- Hanafusa, M. and Ishikawa, J. (2019). External Force Estimation for Nonlinear Systems Using Recurrent Neural Network. In 2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp. 1055–1061. DOI: 10.1109/AIM.2019.8868900.
- Haxthausen, F. von, Böttger, S., Wulff, D., Hagenah, J., García-Vázquez, V., and Ipsen, S. (2021). Medical robotics for ultrasound imaging: current systems and future trends. *Current Robotics Reports*, 1–17.
- Hirose, N. and Tajima, R. (2017). Modeling of rolling friction by recurrent neural network using LSTM. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 6471–6478. DOI: 10.1109/ICRA.2017.7989764.
- Hirzinger, G., Albu-Schaffer, A., Hahnle, M., Schaefer, I., and Sporer, N. (2001). On a new generation of torque controlled light-weight robots. In Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164), vol. 4. IEEE, pp. 3356–3363.
- Hitzler, K., Meier, F., Schaal, S., and Asfour, T. (2019). Learning and Adaptation of Inverse Dynamics Models: A Comparison. In 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 491–498. DOI: 10.1109/Humanoids43949.2019.9035048.
- Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A Field Guide to Dynamical Recurrent Neural Networks*, (Kremer, S.C. and Kolen, J.F., eds.). IEEE Press.
- Hochreiter, S. and Schmidhuber, J. (Nov. 1997). Long Short-Term Memory. *Neural Comput.* *9*, 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.
- Hollerbach, J., Khalil, W., and Gautier, M. (2016). *Model identification*. Springer handbook of robotics, Springer, pp. 113–138.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks* *2*, 359–366.
- Hwang, S. and Hwang, H. (2013). Development of a tactile sensing system using piezoelectric robot skin materials. *Smart materials and structures* *22*, 055004.

Bibliography

- James, S., Freese, M., and Davison, A.J. (2019). PyRep: Bringing V-REP to Deep Robot Learning. arXiv preprint arXiv:1906.11176.
- Jiang, Z.-H., Ishida, T., and Sunawada, M. (2006). Neural Network Aided Dynamic Parameter Identification of Robot Manipulators. In 2006 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, pp. 3298–3303. DOI: 10.1109/ICSMC.2006.384627.
- Jubien, A., Gautier, M., and Janot, A. (2014). Dynamic identification of the Kuka LightWeight robot: Comparison between actual and confidential Kuka’s parameters. In 2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 483–488. DOI: 10.1109/AIM.2014.6878124.
- Kalman, R.E. (1960). On the general theory of control systems. In Proceedings First International Conference on Automatic Control, Moscow, USSR, pp. 481–492.
- Khalil, W and Gautier, M (1991). Calculation of the identifiable parameters for robots calibration. IFAC Proceedings Volumes 24, 687–691.
- Khalil, W. and Dombre, E. (2002). Modeling identification and control of robots. CRC Press.
- Khosla, P.K. and Kanade, T. (1985). Parameter identification of robot dynamics. In 1985 24th IEEE Conference on Decision and Control, pp. 1754–1760. DOI: 10.1109/CDC.1985.268838.
- Kingma, D.P. and Ba, J. (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG].
- Kircanski, N.M. and Goldenberg, A.A. (1997). An Experimental Study of Nonlinear Stiffness, Hysteresis, and Friction Effects in Robot Joints with Harmonic Drives and Torque Sensors. The International Journal of Robotics Research 16, 214–239. DOI: 10.1177/027836499701600207. URL: <https://doi.org/10.1177/027836499701600207>.
- Kuhlemann, I., Jauer, P., Ernst, F., and Schweikard, A. (2016). Robots with seven degrees of freedom: Is the additional DoF worth it? In 2016 2nd International Conference on Control, Automation and Robotics (ICCAR), pp. 80–84. DOI: 10.1109/ICCAR.2016.7486703.
- Kuhlemann, I. (2019). Methods for Quasi-Static Tasks with Redundant Manipulators. PhD thesis. Universität zu Lübeck. URL: <https://www.rob.uni-luebeck.de/fileadmin/downloads/Dissertationen/Kuhlemann,%20Ivo.pdf>.
- KUKA Robot Group Augsburg, G. (2015). *LBR iiwa 7 R800, LBR iiwa 14 R820 Specification*. Ver. 5.
- Kumar, R., Berkelman, P., Gupta, P., Barnes, A., Jensen, P., Whitcomb, L., and Taylor, R. (2000). Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation. In Proceedings 2000 ICRA. Millennium Conference.

Bibliography

- IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, 610–617 vol.1. DOI: 10.1109/ROBOT.2000.844120.
- Lang, K.J., Waibel, A.H., and Hinton, G.E. (Jan. 1990). A Time-Delay Neural Network Architecture for Isolated Word Recognition. *Neural Netw.* *3*, 23–43. ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90044-L.
- Le, D.P., Choi, J., and Kang, S. (2013). External force estimation using joint torque sensors and its application to impedance control of a robot manipulator. In 2013 13th international conference on control, automation and systems (ICCAS 2013), IEEE, pp. 1794–1798.
- Ledezma Díaz, F. and Haddadin, S. (2018). FOP networks for learning humanoid body schema and dynamics. In 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), IEEE, pp. 1–9.
- Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep Neural Networks as Gaussian Processes. In. arXiv: 1711.00165 [stat.ML].
- Li, S., Li, W., Cook, C., Zhu, C., and Gao, Y. (2018). *Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN*. arXiv: 1803.04831 [cs.CV].
- Liu, S., Wang, L., and Wang, X.V. (2021). Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation. *Robotics and Computer-Integrated Manufacturing* *71*, 102168. ISSN: 0736-5845. DOI: <https://doi.org/10.1016/j.rcim.2021.102168>. URL: <https://www.sciencedirect.com/science/article/pii/S0736584521000521>.
- Loshchilov, I. and Hutter, F. (2019). *Decoupled Weight Decay Regularization*. arXiv: 1711.05101 [cs.LG].
- Luh, J.Y., Walker, M.W., and Paul, R.P. (1980). On-line computational scheme for mechanical manipulators.
- MATLAB (2020). version 9.9.0.1538559 (R2020b) Update 3. Natick, Massachusetts: The MathWorks Inc.
- McCulloch, W.S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics* *5*, 115–133.
- Nair, V. and Hinton, G.E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10*. Haifa, Israel: Omnipress, pp. 807–814. ISBN: 9781605589077.
- Nguyen-Tuong, D., Seeger, M., and Peters, J. (2009). Model Learning with Local Gaussian Process Regression. *Advanced Robotics* *23*, 2015–2034.
- Nicholls, D., Sweet, L., and Hyett, J. (2014). Psychomotor skills in medical ultrasound imaging: an analysis of the core skill set. *Journal of Ultrasound in Medicine* *33*, 1349–1352.

Bibliography

- Nickolls, J., Buck, I., Garland, M., and Skadron, K. (Mar. 2008). Scalable Parallel Programming with CUDA: Is CUDA the Parallel Programming Model That Application Developers Have Been Waiting For? *Queue* 6, 40–53. ISSN: 1542-7730. DOI: 10.1145/1365490.1365500.
- Olsson, H., Åström, K., Wit, C.C. de, Gäfvert, M., and Lischinsky, P. (1998). Friction Models and Friction Compensation. *European Journal of Control* 4, 176–195. ISSN: 0947-3580. DOI: [https://doi.org/10.1016/S0947-3580\(98\)70113-X](https://doi.org/10.1016/S0947-3580(98)70113-X).
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. arXiv: 1912.01703 [cs.LG].
- Polydoros, A.S. and Nalpantidis, L. (2016). A reservoir computing approach for learning forward dynamics of industrial manipulators. In 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 612–618. DOI: 10.1109/IROS.2016.7759116.
- Rohmer, E., Singh, S.P.N., and Freese, M. (2013). CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework. In Proc. of The International Conference on Intelligent Robots and Systems (IROS),
- Ruderman, M. and Bertram, T. (Sept. 2012). Modeling and Observation of Hysteresis Lost Motion in Elastic Robot Joints. In Vol. 45, pp. 13–18. DOI: 10.3182/20120905-3-HR-2030.00061.
- Rueckert, E., Nakatenus, M., Tosatto, S., and Peters, J. (Nov. 15, 2017). Learning Inverse Dynamics Models in $O(n)$ time with LSTM networks. In Proceedings of the International Conference on Humanoid Robots (HUMANOIDS), URL: <https://cps.unileoben.ac.at/wp/Humanoids2017Rueckert.pdf,ArticleFile>. published.
- Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *nature* 323, 533–536.
- Safeea, M. and Neto, P. (2017). KUKA Sunrise Toolbox: Interfacing Collaborative Robots with MATLAB. CoRR *abs/1709.01438*. arXiv: 1709.01438. URL: <http://arxiv.org/abs/1709.01438>.
- Sak, H., Senior, A.W., and Beaufays, F. (2014). Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. CoRR *abs/1402.1128*. arXiv: 1402.1128. URL: <http://arxiv.org/abs/1402.1128>.
- Salcudean, S., Zhu, W.H., Abolmaesumi, P., Bachmann, S., and Lawrence, P.D. (1999). A Robot System for Medical Ultrasound. In 9th International Symposium of Robotics Research (ISRR'99), Snowbird, pp. 152–159.
- Schreiber, G., Stemmer, A., and Bischoff, R. (2010). The fast research interface for the kuka lightweight robot. In IEEE Workshop on Innovative Robot Control Architectures for

Demanding (Research) Applications How to Modify and Enhance Commercial Controllers (ICRA 2010), Citeseer, pp. 15–21.

Schweikard, A. and Ernst, F. (2015). *Medical robotics*. Springer.

Smith, A. and Hashtrudi-Zaad, K. (2005). Application of neural networks in inverse dynamics based contact force estimation. In *Proceedings of 2005 IEEE Conference on Control Applications*, 2005. CCA 2005. Pp. 1021–1026. DOI: 10.1109/CCA.2005.1507264.

Sola, J. and Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science* 44, 1464–1468. DOI: 10.1109/23.589532.

Speich, J.E. and Rosen, J. (2004). *Medical robotics*. *Encyclopedia of biomaterials and biomedical engineering* 983, 993.

Spong, M.W. (Dec. 1987). Modeling and Control of Elastic Joint Robots. *Journal of Dynamic Systems, Measurement, and Control* 109, 310–318. ISSN: 0022-0434. DOI: 10.1115/1.3143860. eprint: https://asmedigitalcollection.asme.org/dynamicsystems/article-pdf/109/4/310/5604812/310_1.pdf. URL: <https://doi.org/10.1115/1.3143860>.

Stürz, Y.R., Affolter, L., and Smith, R.S. (2017a). Parameter Identification of the KUKA LBR iiwa Robot Including Constraints on Physical Feasibility. *IFAC-PapersOnLine* 50, 6863–6868.

Stürz, Y.R., Affolter, L.M., and Smith, R.S. (2017b). Parameter identification of the KUKA LBR iiwa robot including constraints on physical feasibility. *IFAC-PapersOnLine* 50, 6863–6868.

Swevers, J., Al-Bender, F., Ganseman, C., and Prajogo, T. (2000). An integrated friction model structure with improved presliding behavior for accurate friction compensation. *IEEE Trans. Autom. Control*. 45, 675–686.

Swevers, J., Ganseman, C., Tukel, D., Schutter, J. de, and Van Brussel, H. (1997). Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation* 13, 730–740. DOI: 10.1109/70.631234.

Taylor, R.H. (1997). Robots as surgical assistants: where we are, wither we are tending, and how to get there. In *Conference on Artificial Intelligence in Medicine in Europe*, Springer, pp. 1–11.

Tjahjowidodo, T., Al-Bender, F., and Van Brussel, H. (2013). Theoretical modelling and experimental identification of nonlinear torsional behaviour in harmonic drives. *Mechatronics* 23, 497–504. ISSN: 0957-4158. DOI: <https://doi.org/10.1016/j.mechatronics.2013.04.002>.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., and Polosukhin, I. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Bibliography

Vijayakumar, S. and Schaal, S. (2000). Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional spaces. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), vol. 1. clmc. Stanford, CA, pp. 288–293. URL: <http://www-clmc.usc.edu/publications/V/vijayakumar-ICML2000.pdf>.

Wang, S., Shao, X., Yang, L., and Liu, N. (2020). Deep Learning Aided Dynamic Parameter Identification of 6-DOF Robot Manipulators. *IEEE Access* 8, 138102–138116. DOI: 10.1109/ACCESS.2020.3012196.

Williams, R.J. and Zipser, D. (1995). Gradient-Based Learning Algorithms for Recurrent Networks and Their Computational Complexity. *Backpropagation: Theory, Architectures, and Applications*, USA: L. Erlbaum Associates Inc., pp. 433–486. ISBN: 0805812598.

Xu, T., Fan, J., Chen, Y., Ng, X., Ang, M.H., Fang, Q., Zhu, Y., and Zhao, J. (2020). Dynamic Identification of the KUKA LBR iiwa Robot With Retrieval of Physical Parameters Using Global Optimization. *IEEE Access* 8, 108018–108031. DOI: 10.1109/ACCESS.2020.3000997.

Yilmaz, N., Wu, J.Y., Kazanzides, P., and Tumerdem, U. (May 2020). Neural Network based Inverse Dynamics Identification and External Force Estimation on the da Vinci Research Kit. In. DOI: 10.1109/ICRA40945.2020.9197445.

Zhang, H., Qin, W., Gao, Y., Li, Q., Chen, Z., and Zhao, J. (2020). Disturbance Elimination for the Modular Joint Torque Sensor of a Collaborative Robot. *Mathematical Problems in Engineering* 2020.

Zhu, L., Mao, H., Luo, X., and Xiao, J. (2016). Determining null-space motion to satisfy both task constraints and obstacle avoidance. In 2016 IEEE International Symposium on Assembly and Manufacturing (ISAM), pp. 112–119. DOI: 10.1109/ISAM.2016.7750723.

are

A

Joint-Wise Torque Predictions

The joint-wise torque predictions over a sub-sequence of the validation trajectory without attached weights are listed in the following for all of the implemented and evaluated architectures. Please refer to Chapters 4 and 5 for the naming conventions of the respective models.

A Joint-Wise Torque Predictions

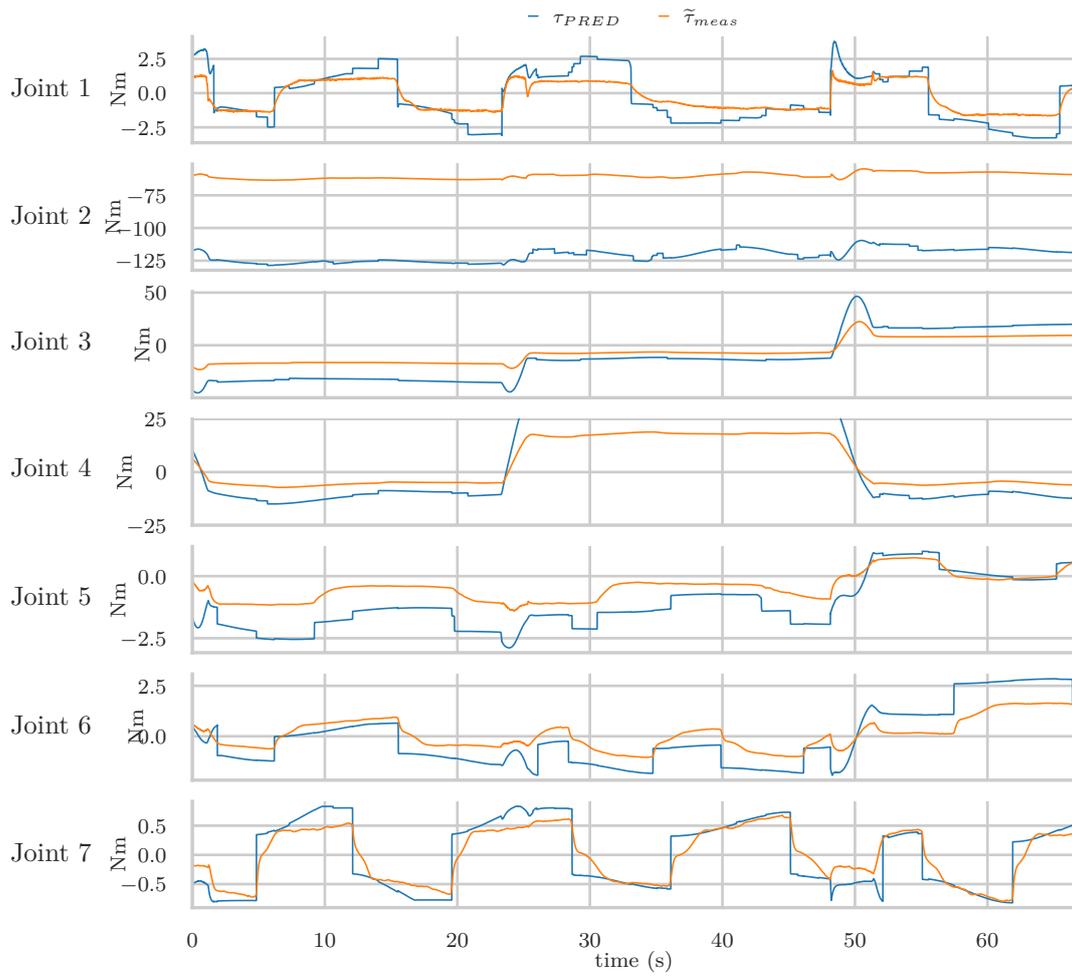


Figure A.1: MLP-7

A Joint-Wise Torque Predictions

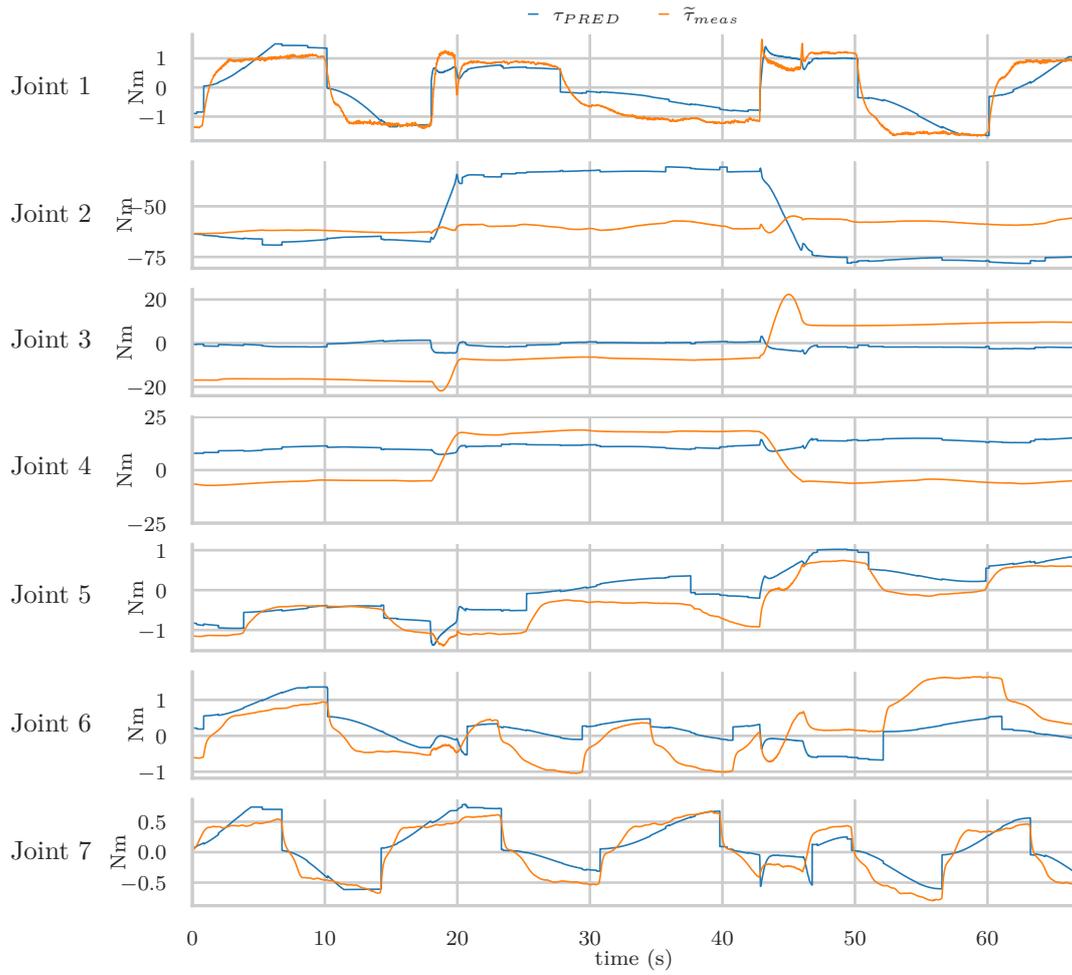


Figure A.2: MLP-7 no r

A Joint-Wise Torque Predictions

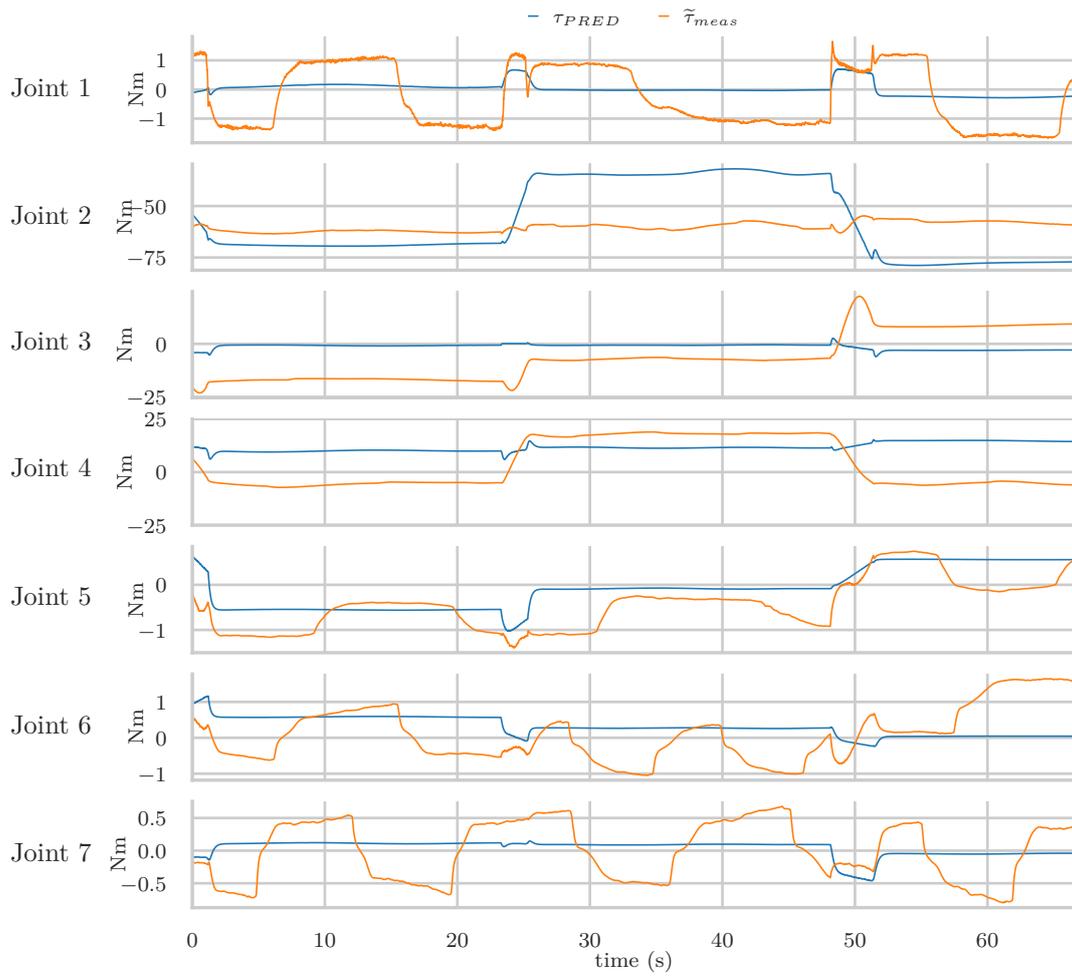


Figure A.3: MLP-7 no r, τ_{RBD}

A Joint-Wise Torque Predictions

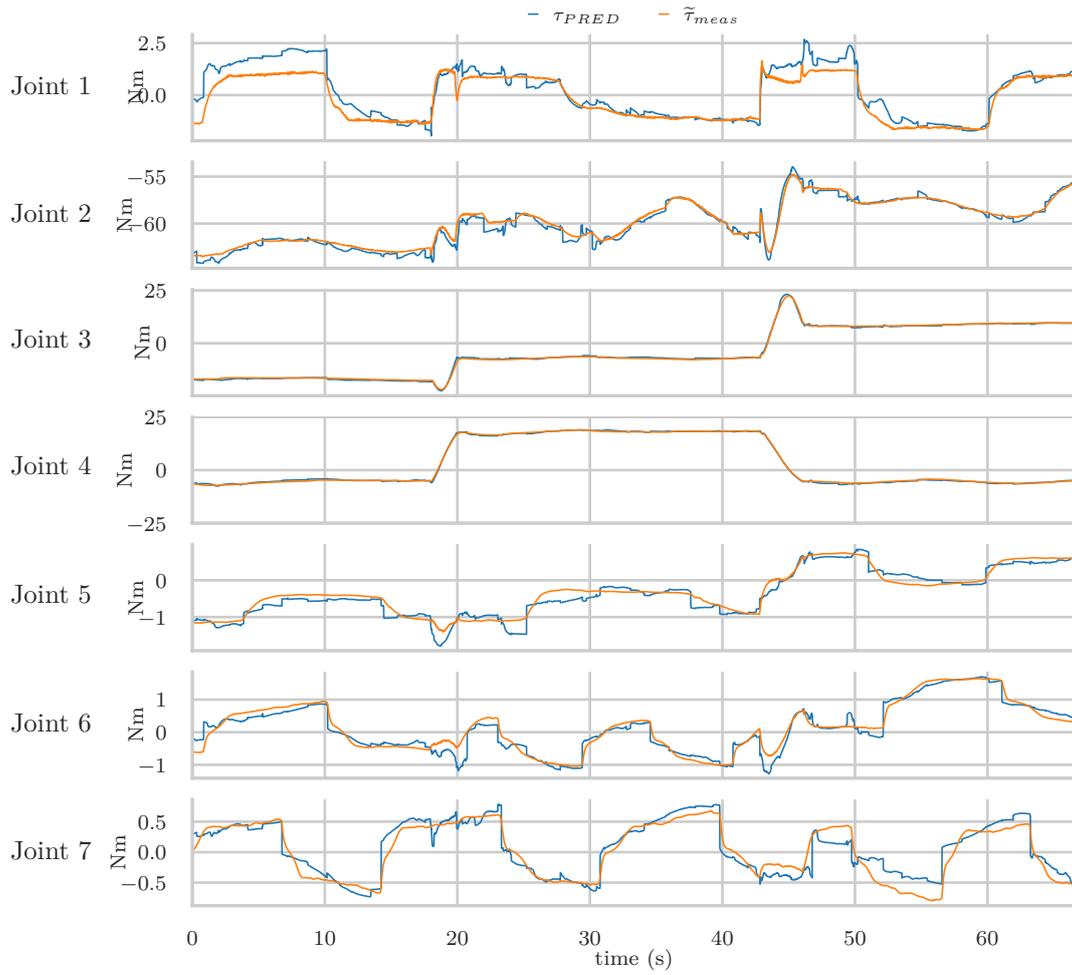


Figure A.4: Transformer 500

A Joint-Wise Torque Predictions

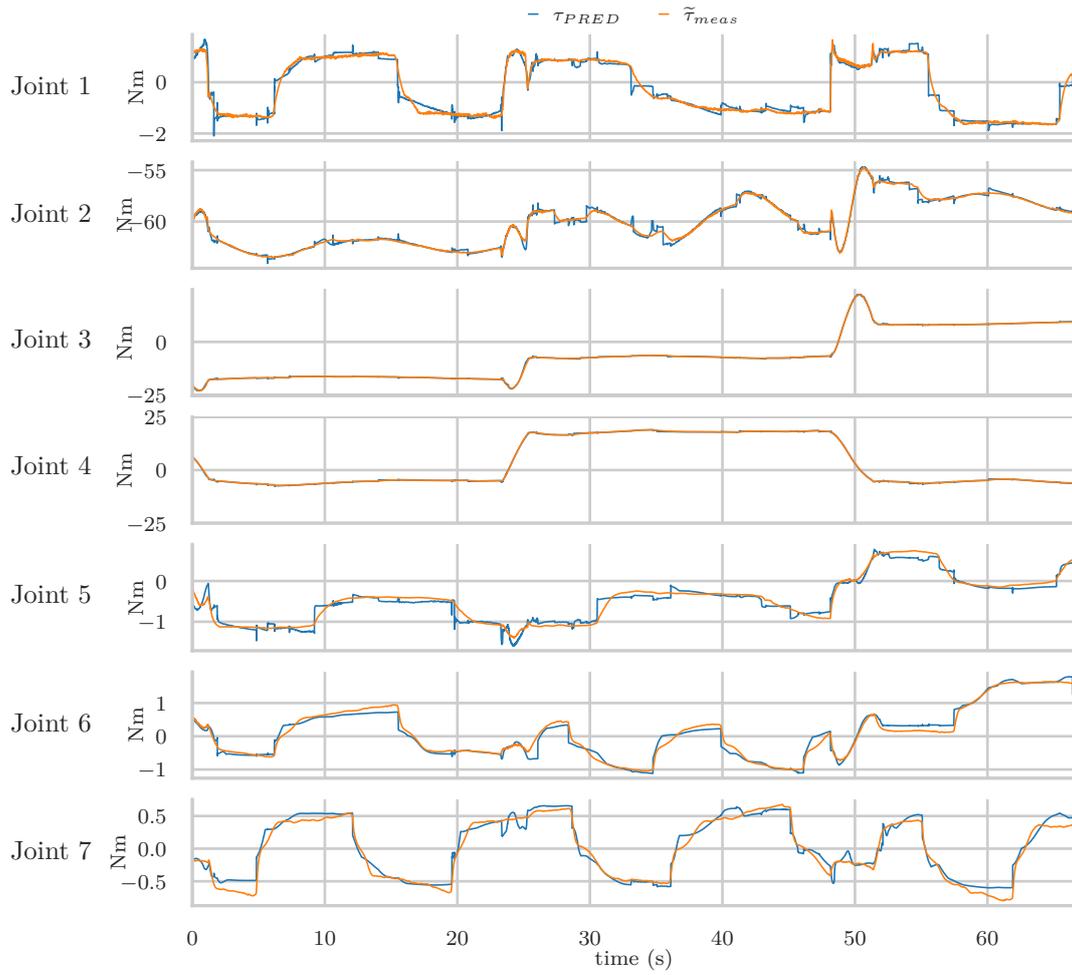


Figure A.5: LSTM-2 100

A Joint-Wise Torque Predictions

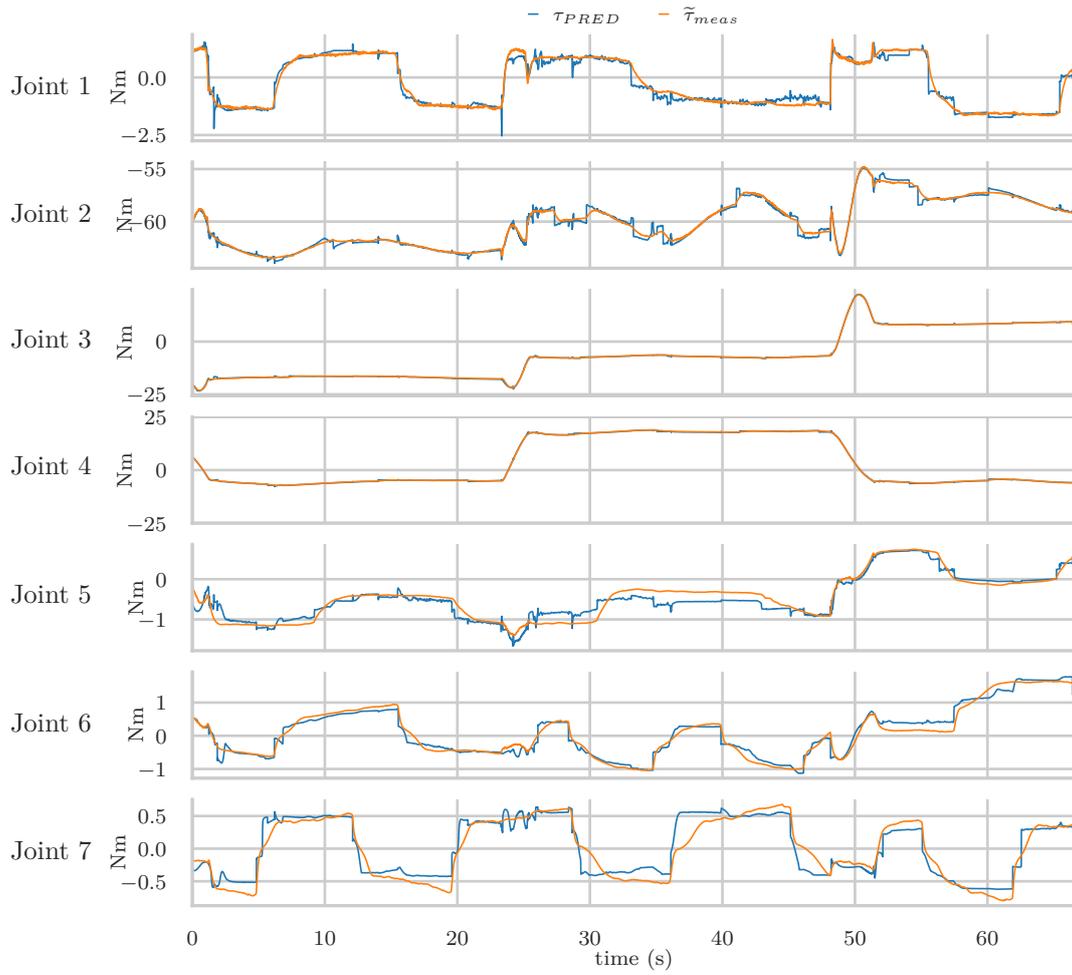


Figure A.6: LSTM-2 100 no r

A Joint-Wise Torque Predictions

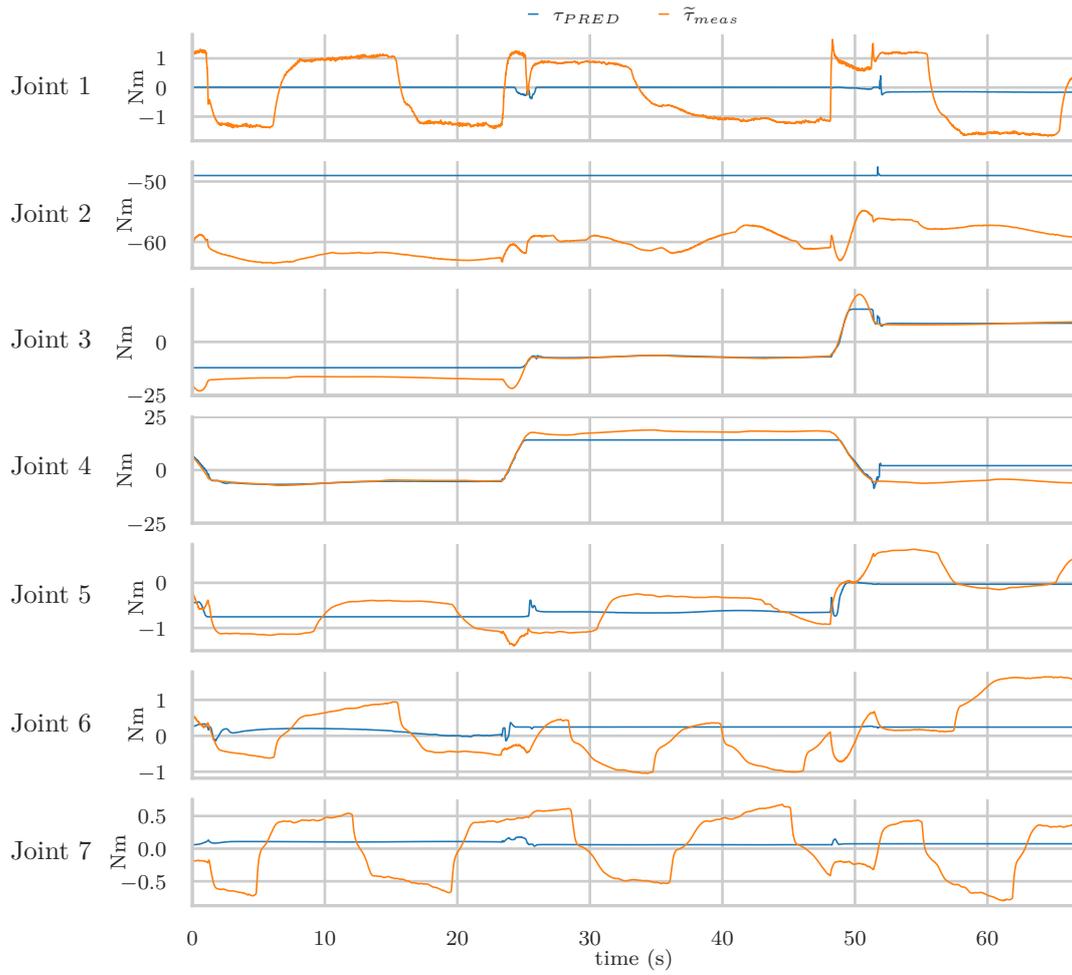


Figure A.7: LSTM-2 100 no r, τ_{RBD}

A Joint-Wise Torque Predictions

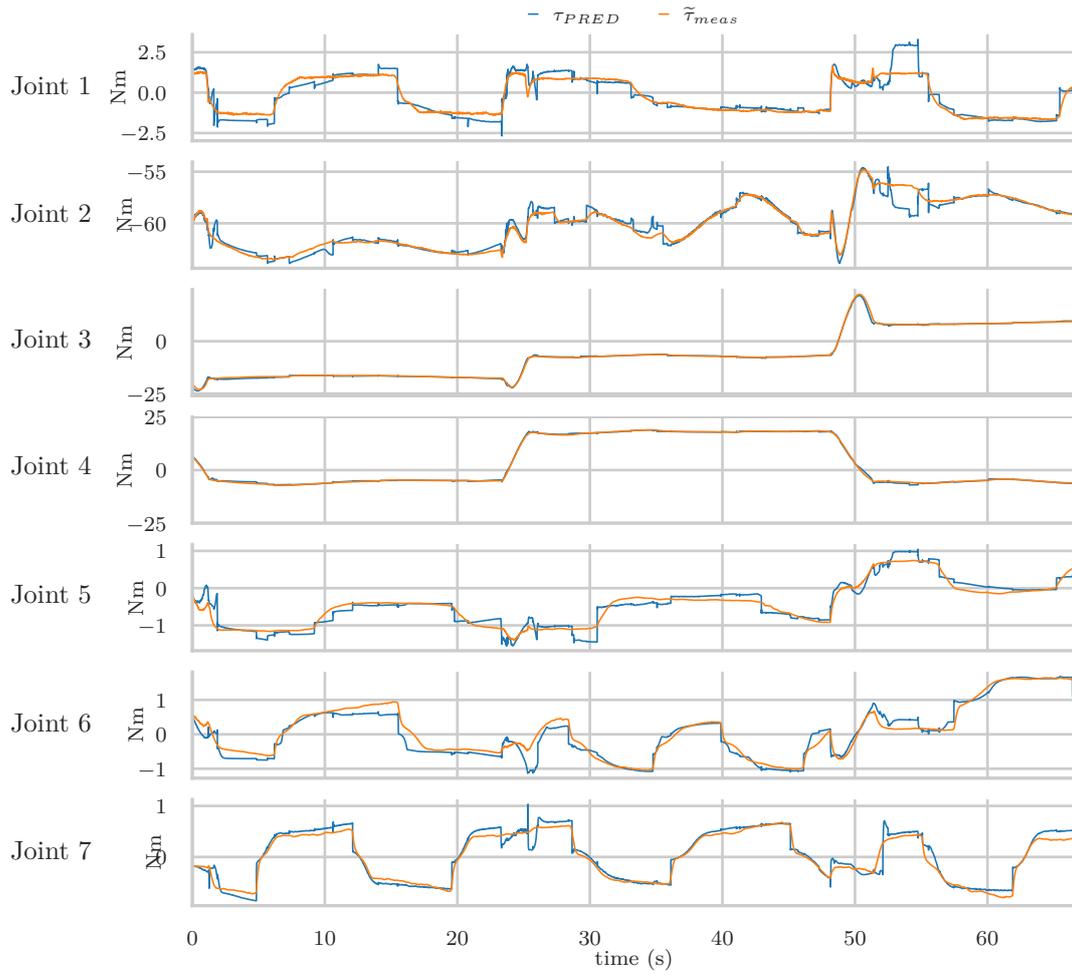


Figure A.8: LSTM-2-FCL 100

A Joint-Wise Torque Predictions

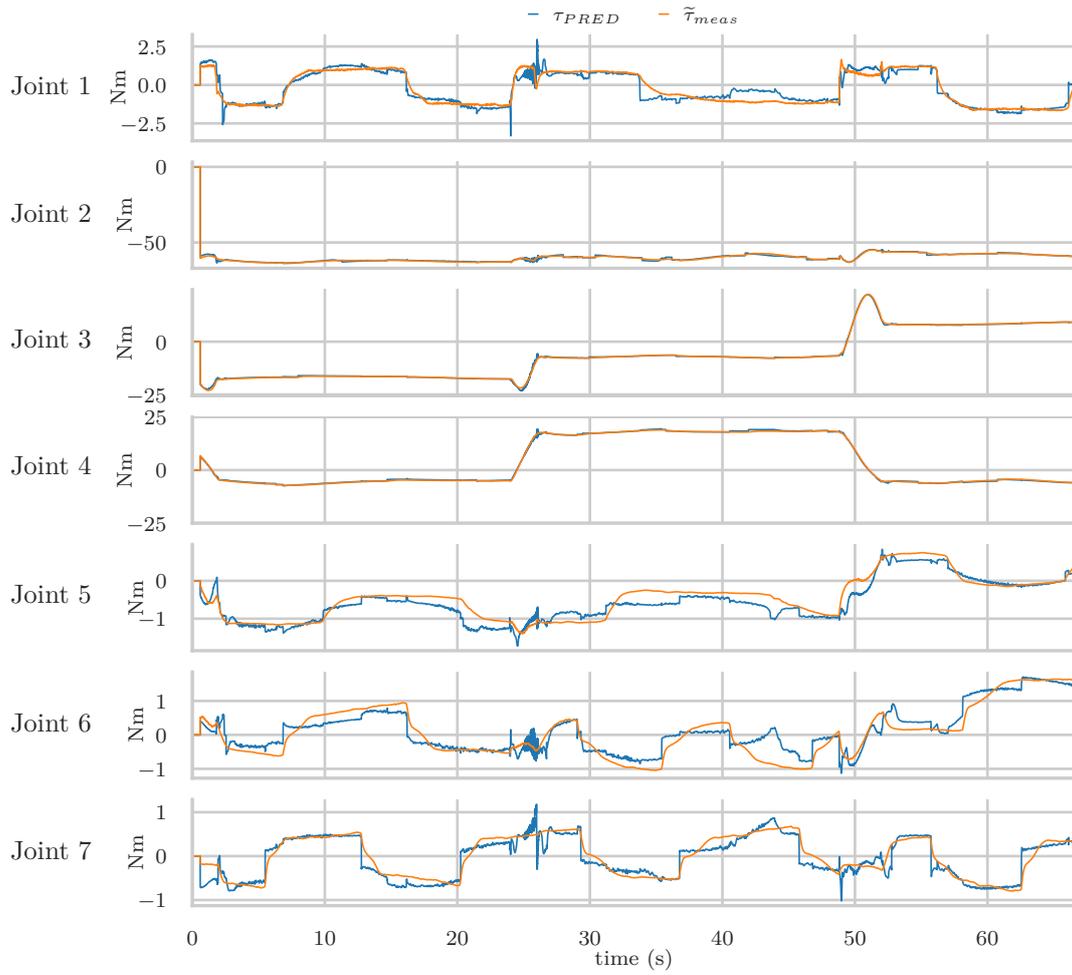


Figure A.9: LSTM-2-FCL 100 no r

A Joint-Wise Torque Predictions

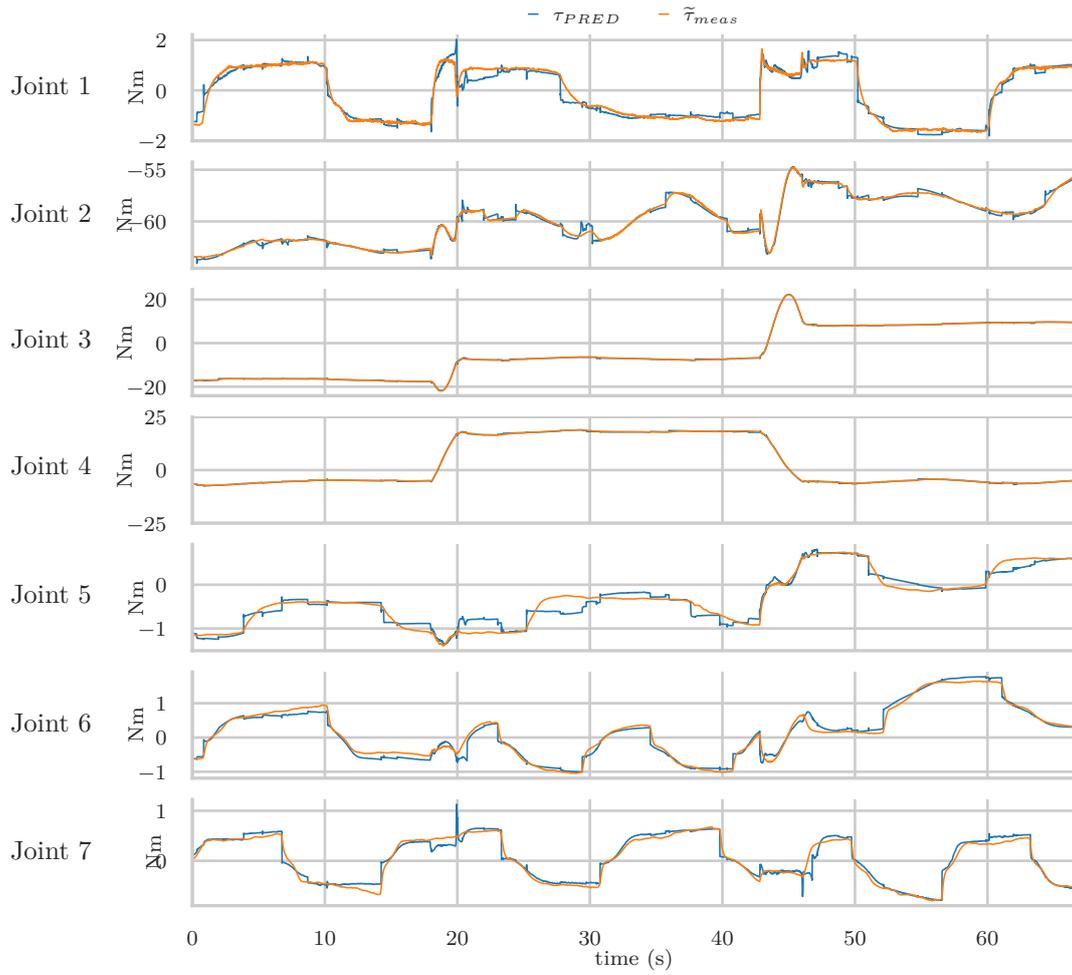


Figure A.10: LSTM-2-FCL 500

A Joint-Wise Torque Predictions

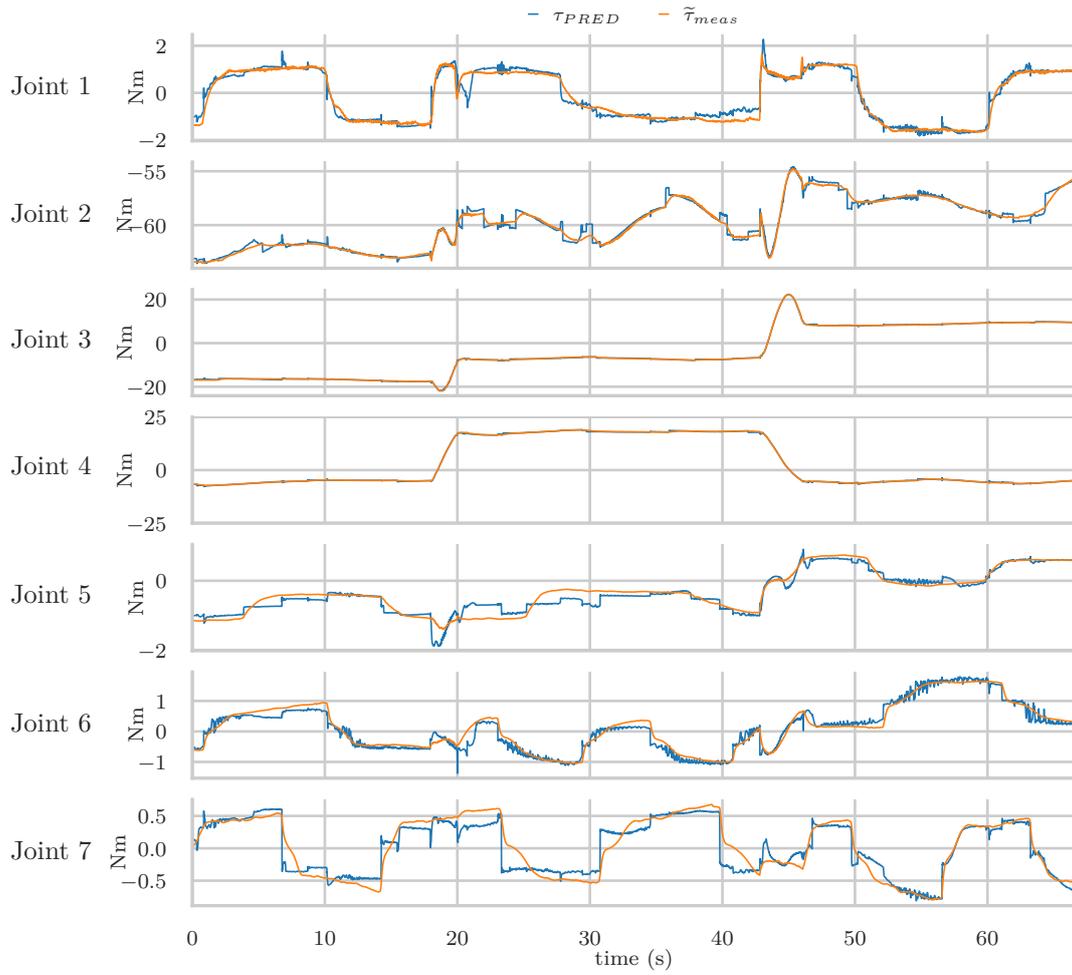


Figure A.11: LSTM-2-FCL 500 no r

A Joint-Wise Torque Predictions

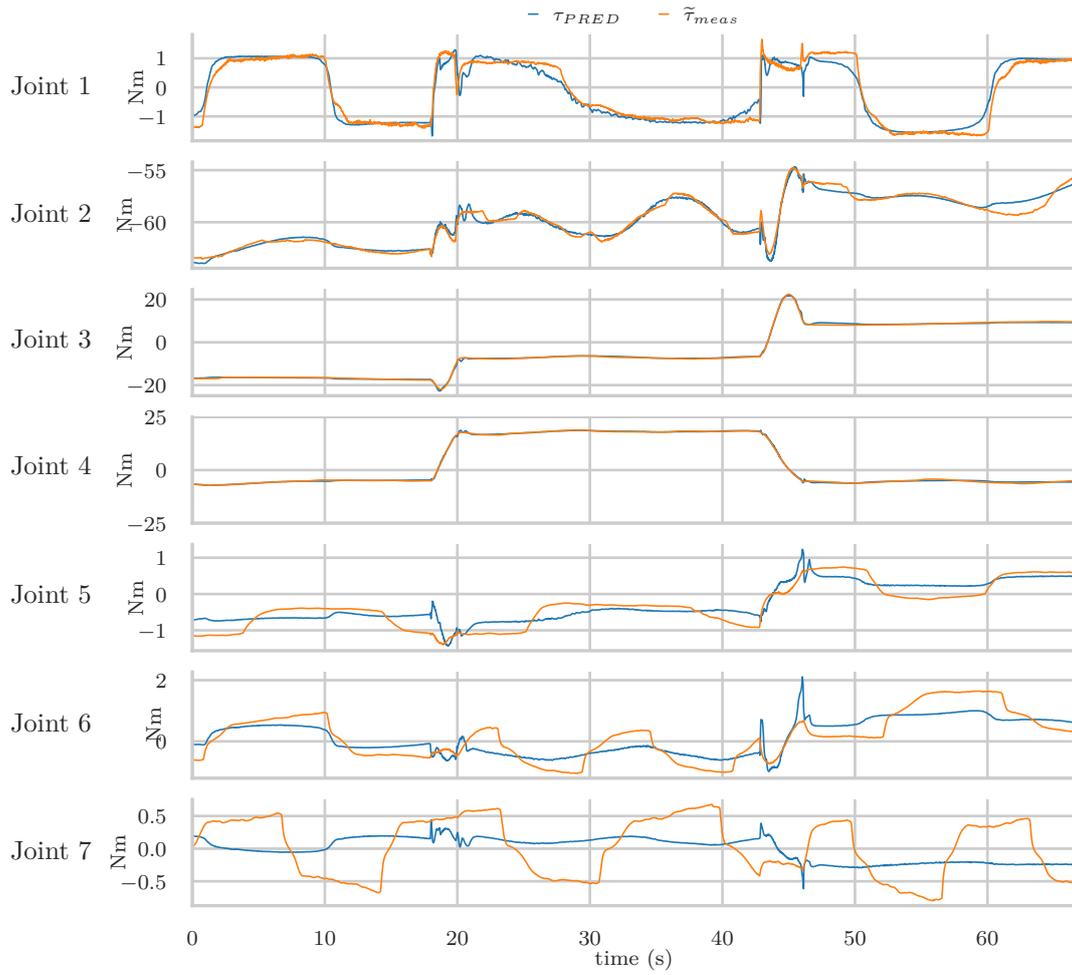


Figure A.12: LSTM-2-FCL 500 no r, τ_{RBD}

A Joint-Wise Torque Predictions

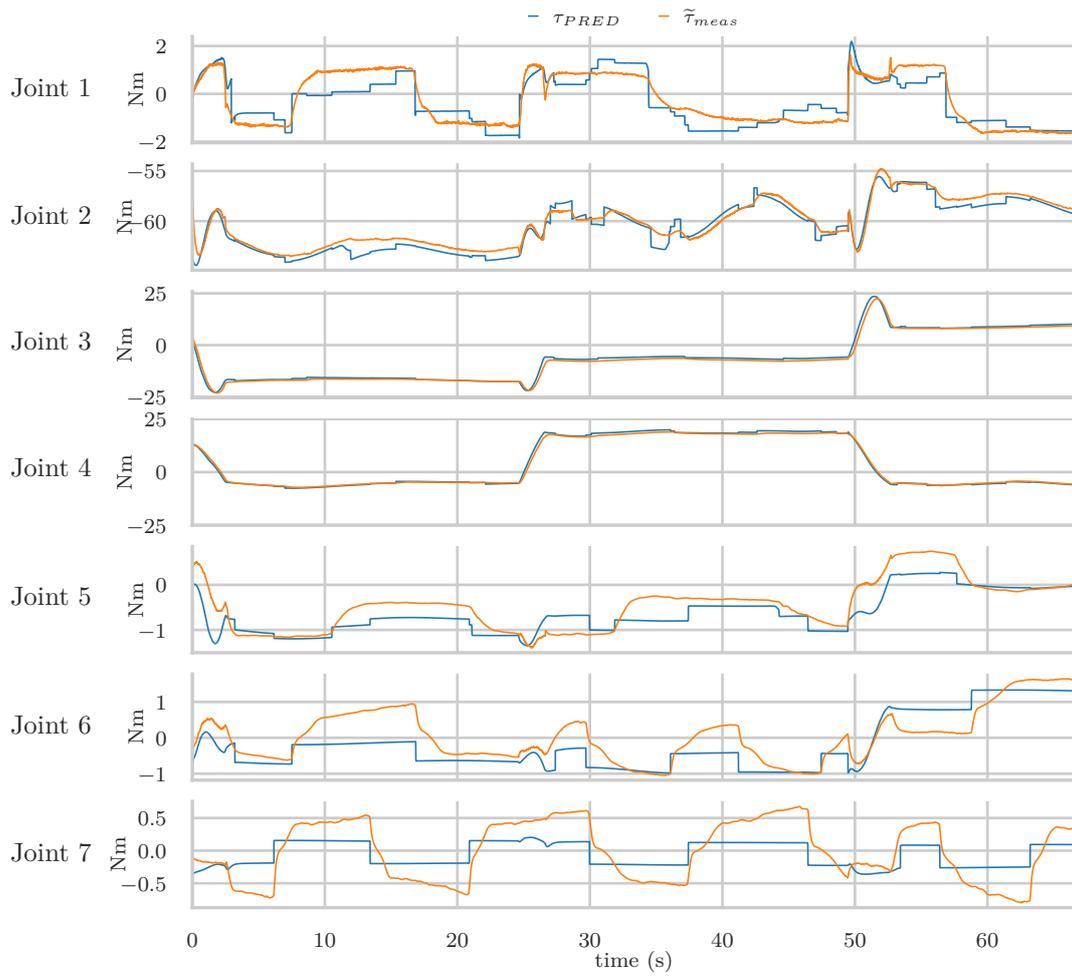


Figure A.13: RBD