TUM School of Computation, Information and Technology
Technische Universität München

# Modeling and Identification for Formally Safe Human-Robot Interaction

**Boson Stefan Liu**

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften (Dr.-Ing.)**

genehmigten Dissertation.

**Vorsitz:**
Prof. Dr. Stefan Leutenegger

**Prüfende der Dissertation:**
1. Prof. Dr. Matthias Althoff
2. Prof. Dr. André Platzer

Die Dissertation wurde am 11.12.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 26.04.2024 angenommen.

# Abstract

Safety is a significant challenge when deploying personal care robots, automated vehicles, and collaborative robots in human environments. Furthermore, a safe system shall not overly restrict task efficiency. Due to the increasing complexity of safety functions and their verification, formal methods are desired to create safe-by-design robotic controllers or to mathematically prove safety. Such techniques depend on verification models to describe the behavior of the real system. However, the challenge with formal methods that use verification models is bridging the gap between model and reality. To overcome this gap in verifying safety, the reachset conformance relation has been proposed, which states that the reachable set of the model always includes the behavior of the real system so that an unsafe region unreachable by a model is also unreachable by the real system.

This dissertation proposes a framework for model identification that establishes a reachset conformance relation between the identified model and the real system. This is achieved by considering uncertainties in the verification model, measuring the real system, and adapting the uncertainties to establish reachset conformance. This framework discusses selecting identification objectives and choosing model structures and provides efficient algorithms for identifying linear dynamics with uncertain disturbances.

The effectiveness of the identification framework has been tested on various safety problems in the human-robot interaction domain. By modeling and identifying human-robot interaction as an uncertain system, safety properties are ensured, such as collision avoidance and adhering to force limits in physical interaction. The author demonstrates that using formal methods such as online verification increases the task efficiency of robots while maintaining human safety. This work also uses verification models to ensure that robotic motion controllers do not violate torque constraints. The identification framework can be combined with formal synthesis to simultaneously create safe-by-design controllers and their underlying verification model.

# Zusammenfassung

Sicherheit ist eine große Herausforderung beim Einsatz von Pflegerobotern, automatisierten Fahrzeugen und kollaborierenden Robotern in menschlicher Umgebung. Außerdem darf ein sicheres System die Effizienz der Roboterarbeit nicht zu sehr einschränken. Aufgrund der zunehmenden Komplexität von Sicherheitsfunktionen und deren Verifikation werden formale Methoden gewünscht, um Safe-by-Design-Robotersteuerungen zu erstellen oder die Sicherheit mathematisch zu beweisen. Solche Methoden sind auf Verifikationsmodelle angewiesen, um das Verhalten des realen Systems zu beschreiben. Die Herausforderung bei formalen Methoden, die Verifikationsmodelle verwenden, besteht jedoch darin, die Lücke zwischen Modell und Realität zu schließen. Um diese Lücke bei der Verifizierung der Sicherheit zu schließen, wurde die Reachset-Konformanzrelation vorgeschlagen, die besagt, dass die erreichbare Menge des Modells immer das Verhalten des realen Systems umfasst, so dass eine unsichere Region, die von einem Modell nicht erreicht werden kann, auch vom realen System nicht erreicht werden kann.

In dieser Dissertation wird eine Vorgehensweise für die Modellidentifikation vorgeschlagen, die eine Reachset-Konformanzrelation zwischen dem identifizierten Modell und dem realen System herstellt. Dies wird erreicht, indem Unsicherheiten im Verifikationsmodell berücksichtigt werden, das reale System gemessen wird und die Unsicherheiten so angepasst werden, dass eine Reachset-Konformanz hergestellt wird. Für die Vorgehensweise wird in dieser Arbeit die Auswahl von Identifikationszielen sowie Modellstrukturen erörtert und effiziente Algorithmen zur Identifikation linearer Dynamik mit unsicheren Störungen vorgeschlagen.

Die Wirksamkeit unserer Vorgehensweise für Modellidentifikation wurde an verschiedenen Sicherheitsproblemen im Bereich der Mensch-Roboter-Interaktion getestet. Indem die Mensch-Roboter-Interaktion als unsicheres System modelliert und identifiziert wird, ist es möglich, Sicherheitseigenschaften wie Kollisionsvermeidung und die Einhaltung von Kraftgrenzen in der physischen Interaktion zu gewährleisten. Der Autor zeigt, dass der Einsatz formaler Methoden wie der Online-Verifikation die Arbeitseffizienz von Robotern erhöht und gleichzeitig die Sicherheit des Menschen gewährleistet. Diese Arbeit verwendet auch Verifikationsmodelle, um sicherzustellen, dass Roboterbewegungssteuerungen keine Drehmomentbeschränkungen verletzen. Die Vorgehensweise zur Modellidentifikation kann mit der formalen Synthese kombiniert werden, um gleichzeitig Safe-by-Design-Steuerungen zu entwerfen und das zugrunde liegende Verifikationsmodell zu liefern.

# Acknowledgments

This thesis would not have been possible without the support and contributions of many individuals. First and foremost, I would like to express my deepest gratitude to Prof. Matthias Althoff for his exceptional supervision and guidance. The knowledge I've gained under his mentorship and the personal and professional growth I've experienced as his PhD student have been invaluable.

I am also sincerely thankful to Hendrik Roehm, Andrea Giusti, and Aaron Pereira, whose work has profoundly influenced my research, with many aspects of this thesis building on their achievements. More importantly, working with you during my master's studies sparked my passion for research and set me on this path.

I deeply appreciate the financial support provided by the Deutsche Forschungsgemeinschaft, the EU Horizon 2020 program, and the Central Innovation Program (ZIM) of the German Federal Government. I also want to thank the many students I had the privilege of working with; your contributions have been essential to advancing this research.

My heartfelt thanks go out to my former colleagues Niklas, Stefanie, Markus, Christian, Bastian, Carmella, Felix, Moritz, Jagat, Xiao, Edmond, Egon, and others I may not have mentioned. You all made our time at TUM truly enjoyable, creating a warm and lively environment. I will always cherish the memories of our time together, whether in Garching or during our travels abroad for workshops and conferences.

I am deeply grateful to Manuj, Lok Man, and Hendrik for their careful review of my thesis and their valuable feedback. Your thoughtful suggestions have been instrumental in refining this work.

Finally, I would like to thank my wife, Xing, for her unwavering love, patience, and support throughout this journey. Her belief in me has been a constant source of strength, especially during the most challenging moments. I am profoundly grateful for the countless sacrifices she made, which made it possible for me to complete this thesis.

Munich, July 2024 Stefan Liu

# Contents

# 1 Introduction

Robotics and automation are of increasing importance to a prosperous economy and society by relieving the work of humans. So-called collaborative robots are moving out of their traditional workspaces in industrial applications and operating closer to humans. Examples are robots that assist older people, do household work, act as museum tour guides, or perform surgeries. For these applications, robot manufacturers must ensure that these collaborative robots are safe and reliably fulfilling their tasks. At the same time, humans should never be harmed during an interaction with a robot, no matter in which situation. The future success of robotics will highly depend on how safe the control algorithms will be.

Traditionally, control algorithms are extensively tested before deployment. For example, the Google Waymo project is simulating the equivalent of billions of years of driving to test the safety of their control algorithms [11]. Due to the complexity of our environment and its continuous nature, however, it is unfeasible to simulate every possible scenario and every possible combination of circumstances. In addition, such simulations could even leave out essential details relevant to the real robot's behavior. To counter these issues, formal methods are increasingly explored in robotics [12]. Such techniques aim to ensure safety through mathematical proofs and thus derive control algorithms that are safe by design.

One promising approach for the formal verification of safety is *reachability analysis* [13]: instead of simulating single trajectories of a system, *reachable sets* are computed, which are sets that include all possible trajectories given all uncertain parameters of a system. Subsequently, this work models *unsafe sets*, i.e., these are areas the system should not reach. Safety is formally proven by demonstrating that the reachable set of the robot system never intersects with unsafe sets. This is visualized in Figure 1.1. The advantage of such an approach is that reachability analysis allows modeling engineers to explicitly use uncertain but bounded parameters in their verification models so that non-deterministic and complex system behavior can be abstracted and contained.

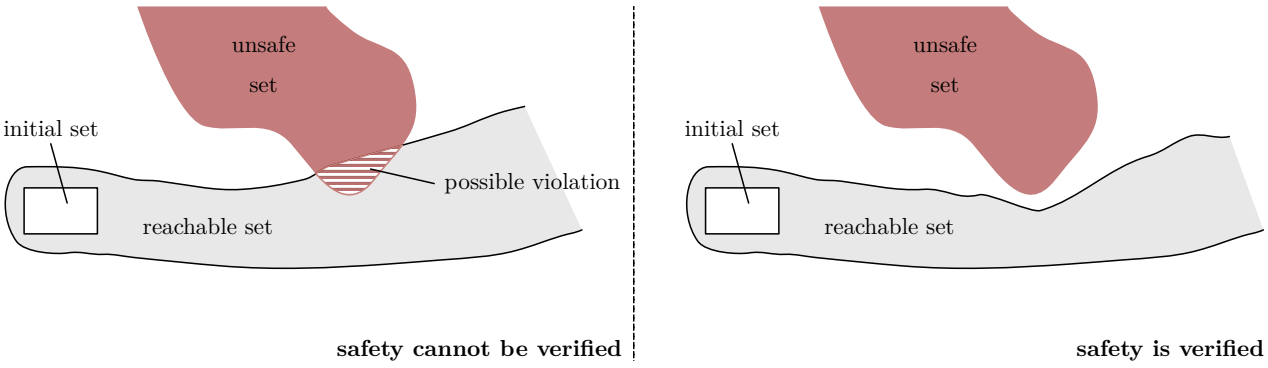


**Figure 1.1:** Safety verification is a reachability problem. If the reachable sets of a system do not intersect with unsafe sets, then the system is verified safe.

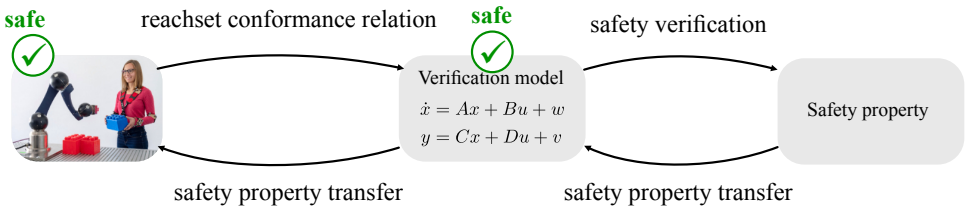


**Figure 1.2:** Reachset-conformant models enable a safety property transfer from the verification model to the real-world system.

Both simulation and formal analysis rely on sufficiently accurate modeling of the robot system interacting with its environment. However, the question arises about what *sufficiently accurate* means. Whether a model is suitable depends on its intended purpose, e.g., a *first-principles model* is derived from physical laws to fundamentally understand the behavior of a system. In contrast, *control models* are developed to help optimizing the performance of a controlled system through simulating the plant system. For *verification models*, it is required that they have a formal relationship to the real system. This is also called *model conformance* [14], i.e., a verification model would be sufficiently accurate if it satisfies the necessary conformance relation. For safety properties, *reachset conformance* is a necessary and sufficient relation between a model and the real system [15]. Figure 1.2 illustrates the relations between the real system, the verification model, and a safety property. Yet, how can such a model be found, and what can be done to ensure that this model satisfies reachset conformance? Hence, this thesis is concerned with the following research question:

> How can a verification model be identified to formally prove the safety of human-robot interaction and ultimately allow robots to safely operate alongside humans?

Throughout my research, I have developed a methodology for solving this question and demonstrated my identification approach to the safety verification of real human-robot interactions. As a result, I published multiple papers with examples of safety properties and applied formal methods to real robotic systems, which in some cases even improved the task efficiency. I have chosen the publication-based thesis format to present my results, and therefore, I have selected the following structure for this thesis:

- Section 1.1 presents a literature overview highlighting the challenges of applying formal methods to human-robot interaction.
- Section 1.2 lists my publications that are included in this thesis, which tackle the main research question.
- Chapter 2 describes the general methodology: it provides basic definitions and introduce the framework for identifying verification models for safety. Simple examples from my papers are included to convey the basic ideas.
- Chapter 3 discusses the conclusions of my research and lists future work.
- The Appendix includes my full papers, where each paper is introduced by a summary describing how they relate to this thesis and listing the contributions of each co-author.

## 1.1 Overview of related literature

This section reviews related literature and identifies the challenges this thesis tackles. Safe human-robot interaction is reviewed in Section 1.1.1, formal methods in robotics are reviewed in Section 1.1.2, and model identification for formal methods are reviewed in Section 1.1.3. An overview of further literature and challenges can be found in my publications in [1–10].

### 1.1.1 Safe human-robot interaction

Multiple standards have been introduced as guidance for safely designing and operating robot systems around humans. The International Organization for Standardization (ISO) norms provide a worldwide common understanding of robot safety. We would like to highlight the following standards [16–19]:

- **ISO 12100** [16] specifies general principles for the safe design of a machine and focuses on risk assessment and risk reduction. Risk reduction should be achieved through (in the order of priority) 1) inherently safe design, 2) safeguards and protective devices, and 3) information for use. The application if such principles lead to the recommendations in ISO 10218 and ISO 13482, but also the works in this thesis can be regarded as a result of applying the first principle. The risk assessment and reduction process for *safety controllers* is specified in more detail in ISO 13849 and IEC 62061 [16].

- **ISO 10218** [17] specifies safety requirements of industrial robots. It also lists four possible modes for *human-robot collaboration* and their requirements, which are 1) safety-rated monitored stop, 2) hand-guiding, 3) speed and separation monitoring, and 4) power and force limiting. Examples for computing the limits for maximum power, force, velocity, and energy are given in ISO/TS 15066 [18], which will be integrated into future versions of ISO 10218.

- **ISO 13482** [19] specifies safety requirements for personal care robots, including mobile servant robots, physical assistant robots, and person-carrier robots. Safety controller requirements for human-robot interaction are concerned with stopping functions, operational spaces, stability, speed and force control, human detection, etc.

The latter two standards also provide implementation examples for fulfilling the safety requirements. Since collaborative applications have been introduced fairly recently, the proposed methods and the safety limits still need to be explored within the research community. Multiple works [20, 21] are pointing out that the abstract models used in ISO/TS 15066 [18] for converting between force, energy, and velocity limits are inaccurate and could lead to dangerous behavior. In our work in [2], we have shown that robots are guaranteed to avoid collisions without defining operational spaces as required in ISO 13482 and that these spaces unnecessarily decrease robot performance.

Naturally, safety standards can only be based on available technologies at the time of their publication, and their content may change in the future, incorporating updated research results. In the meantime, newer methods that improve robot efficiency and cover more applications are being developed. Recent examples of improving efficiency include a dynamic scaling of the operational spaces [22] and a dynamic scaling of robot velocity based on human

behaviors [9, 23]. The work in [24] proposes a new controller that smoothly switches between energy limits for human-robot collaboration depending on human behavior through variable impedance control. Some works consider probabilistic models, e.g., [25] uses Gaussian processes to model the contacts of the robot with the environment to reduce collision forces, and [26] trains convolutional neural networks to compute control policies for navigating in crowded environments. However, the work in [27] points out that probabilistic methods are problematic when considering human uncertainty. The authors argue that a failure probability of under $10^{-8}$ is necessary, but existing learning methods would be ill-equipped to compute accurate confidence bounds at such low probabilities.

Formal methods for guaranteeing safety are not yet considered in current robotic standards. For verifying safety functions, both ISO 10218 and ISO 13482 focus on non-formal techniques, such as testing, code review, and sufficient documentation of the risk assessment procedure. As pointed out earlier, this may not be sufficient for applications such as human-robot interaction, where human behavior varies broadly and is highly uncertain, and the applications are getting increasingly complex. As research on formal methods for safety-critical applications progresses, they are being increasingly advocated for adoption in safety standards [28]. Still, a wide-scale application will ultimately depend on the maturity of these methods. Thus, the following subsection covers recent advances in formal methods in robotics.

### 1.1.2 Formal methods in robotics

First, let me define *formal methods*. The survey of Kress-Gazit et al. in [12] gives the following definition (emphasis added):

> *Formal methods are* ***mathematical tools*** *and techniques used in several engineering domains to* ***reason*** *about systems, their requirements, and their guarantees. Typically, formal methods address two questions:* ***verification*** *(given a set of requirements or specifications and a system model, does the system satisfy the specifications?) and* ***synthesis*** *(given a set of specifications, can one generate a system that is correct by construction, i.e., built in a way that is guaranteed to satisfy the requirements?).*

This thesis addresses both formal synthesis and formal verification. However, requirements, such as those in the previously mentioned safety standards, are usually written in natural language. Requirements must be formulated in a mathematically precise language for use in formal methods. Typically, *temporal logic* can be used for this, e.g., linear temporal logic (LTL), metric temporal logic (MTL), or signal temporal logic (STL) [12, 29]. For example, the authors of [30] cooperated with lawyers to formalize road traffic rules into MTL formulas based on German laws, the Vienna Convention, and previous court decisions. Specifications can be roughly categorized into safety specifications, i.e., how the robot should always behave, and liveness specifications, i.e., task goals that the robot should eventually achieve [12]. For the following discussion, the focus is mainly on safety.

*Formal verification* aims to check whether the controlled system satisfies the required specifications. Multiple techniques have been developed for robotic systems, where the dynamics can be hybrid, i.e., continuous and discrete dynamics. Examples are *theorem proving* [31], *barrier certificates* [32], and *reachability analysis* [13], each of which have a differing view on the task of safety verification:

- Theorem proving aims for logically proving the specifications by also representing the dynamics of hybrid systems as logical formulas, e.g., using differential dynamic logic [31].
- Barrier certificates are functions of the state, which are positive for unsafe states, negative for safe states, and have a negative derivative [32]. Thus, the existence of a barrier certificate proves safety; this notion is similar to Lyapunov functions since the main challenge to verify safety is finding the barrier certificate. Control barrier functions (CBF) are input-dependent barrier certificates, i.e., the CBF is a barrier certificate if the control input is chosen from a constrained set.
- Reachability analysis computes the set of all possible behaviors of a system. Safety is verified by geometrically checking for intersections with unsafe sets, as shown in Figure 1.1. Unsafe sets are part of safety properties, which will be formally defined in Section 2.1.

One example in the literature analyzed with all three verification methods is the collision avoidance problem of mobile robots around moving obstacles [2, 33, 34]. Other examples of using CBFs are guaranteeing the foot placement of legged robots on stepping stones [32], safe adaptive cruise control (ACC) for autonomous driving [32], guaranteeing that a fleet of mobile robots never runs out of energy [32], or guaranteeing torque saturation of robot manipulators [35]. However, an important drawback of classical CBFs is that they do not inherently consider uncertainties. Only recent works such as [36] consider robust variants with measurement errors. In contrast, reachability analysis explicitly considers dynamics where uncertainties can be modeled as bounded sets. The work in [37] predicts possible future occupancies of traffic participants, where adherence to traffic laws can also be factored in the verification model. These predicted occupancies can be used to iteratively verify possible motions of an automated vehicle to avoid collisions, e.g., for safe ACC [38]. The paper in [39] explains the modeling and verification of surgical robots using reachability analysis, where the robot must find a correct region for needle puncturing on a patient without exceeding an interaction force on non-correct areas. In [40], the authors verify that the deviation from the reference trajectory stays below a threshold value for robotic paint spraying tasks. Apart from our work [1–3, 5, 8–10] which includes multiple examples of using reachability analysis in robotics, let me also mention recent works such as in [41], where the authors combine reachability analysis with reinforcement learning to ensure a safe lane-change maneuver of autonomous vehicles and the safe traversing of a quadrotor drone through a tunnel with randomly placed obstacles; and in [42], where the authors verify an airborne collision avoidance system that includes a neural network; in [43], where the authors verify a stop-and-go ACC. Reachability analysis is also suitable for *online verification*, where control inputs are iteratively verified at runtime for short intervals, sufficient for the robot to reach an invariably safe state. Since only the current scenario needs to be considered with a short time horizon, online verification can lead to less conservative results and even increase task efficiency while retaining the same level of safety. Examples of online verification for autonomous driving can be found in [44] and for robot manipulators interacting with humans in [9].

*Formal synthesis* aims to provide inherently correct controllers concerning liveness and safety specifications. The previously mentioned CBFs and online verification can also be considered under this category. Much of the previous work, however, uses an abstraction-based

approach [12, 45], where the continuous dynamics of robots are translated into a (discrete) symbolic model. Possible abstraction techniques are state partitioning [46, 47], time discretization [48], or motions primitives [49, 50]. This way, LTL formulas and abstracted robot dynamics can be translated into transition systems, and the controller can be synthesized through a graph-search algorithm. Recent examples are, e.g., a sampling-based path planning algorithm for mobile robots in [51], which considers LTL specifications, and a mobile robot navigating through a warehouse by concatenating motion primitives [52]. In [48], STL formulas and time-discretized dynamic systems are encoded as a Mixed-Integer-Linear programming for efficient solving. In [49] and [50], motion primitives are generated through optimizing reachable sets, and a maneuver automaton is used to synthesize safe motions. Abstraction-based techniques are only sometimes suitable since abstracted discrete states scale exponentially with the number of states and inputs; most cited examples are only two-dimensional. In robotic control, interval arithmetic techniques for formally proving stability can be employed [8, 53].

Although there have been significant advances recently in formal methods in robotics, critical challenges remain. As [54] points out, formal models have "the problem of the reality gap", meaning that the modeled behavior is "not close enough to the real world to ensure successful transfer of their results." The paper in [55] added to this that also the gap between a complex model and an abstract model which planning algorithms can handle "represents a major challenge that cannot be neglected when addressing real-world robotic systems." The review article on task and motion planning in [56] specifically addresses "planning with uncertainty" as an open research question, as current planning methods often do not consider uncertainties. Whether local control methods can handle uncertainties or even need to be considered on the task level should be validated.

To provide possible answers to these modeling challenges, model conformance and identification for formal methods is reviewed in the following section.

### 1.1.3 Model identification for formal methods

Formal methods are usually applied to a model representing the real system. However, if a property has been verified on a model, how does the same property also apply to the real robot? The answer to this question is that a *model conformance* relation must exist, which means that the behavior of a model is related to the behavior of the real system in a way that allows transference of a verified property. Model conformance relations for cyber-physical systems have been reviewed in [14]. They can be sorted into three main categories: *simulation relation*, *trace conformance*, and *reachset conformance*, which differ from each other in terms of the types of properties that can be transferred if the corresponding conformance relation has been shown. The simulation relation is the strictest among these three but always allows the transference of LTL and MTL properties.

Finding a simulation relation is the goal of most set-based identification literature. The model can take the form of differential inclusions [57, 58], or a coarse-grained abstraction of the state space into a discrete automaton (for example, for mobile robot navigation [12]). A linear system with uncertainties is identified in the study in [57] such that every state measurement falls inside a polytopic reachable set. The work in [58] establishes a simulation relation between measured states and hyper-rectangular reachable sets and identifies piecewise affine models via mixed-integer linear programming.

However, the simulation relation may be overly conservative and restrictive if a system is high-dimensional, while only a small number of its outputs are essential for the verification problem. Therefore, Roehm et al. [14] suggest relaxing the conformance relation to the system's output using trace and reachset conformance. Schürmann et al. [59] reconstruct the disturbance traces given measurements from a real autonomous vehicle to demonstrate trace conformance. The set of uncertain disturbances is then modeled as the bounds of all disturbance traces. The reachset conformance relation is an additional relaxation that only requires that the output traces of a system are contained within the reachable set. To verify safety properties, reachset conformance is sufficient as shown in [14, 15].

Another related class of methods is set-membership identification [60–64]. These works aim to identify *feasible solution sets* for the parameters of a system, such that they are guaranteed to contain the true value. However, these studies assume that these parameters are deterministic and constant over time, while reachability analysis explicitly considers time-varying and non-deterministic parameters. Thus, the identification approach for finding feasible solution sets differs from reachset-conformant model identification and, therefore, cannot be directly used for safety verification.

## 1.2 Publications and outline

To summarize the literature review, current robotic applications suffer from reduced task efficiency when operating around humans, and ensuring safety in human-robot interaction is a complex problem. Formal methods are increasingly demanded in robotics, but proposed methods are only demonstrated in simulation, and a "reality gap", i.e., the lack of a conformance relation, prevents the results from transferring to real applications. To address these challenges by answering my research question on how to identify verification models for proving safety, I have co-written ten publications [1–10]. All of them have been published in peer-reviewed international journals or conferences. The following five publications, where I am the first author, are included in the Appendix of this document as core publications:

[1] **S. B. Liu** and M. Althoff, "Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 370–376.

[2] **S. B. Liu**, H. Roehm, C. Heinzemann, I. Lutkebohle, J. Oehlerking, and M. Althoff, "Provably safe motion of mobile robots in human environments," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1351–1357.

[3] **S. B. Liu** and M. Althoff, "Online Verification of Impact-Force-Limiting Control for Physical Human-Robot Interaction," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 777–783.

[4] **S. B. Liu**, A. Giusti, and M. Althoff, "Velocity Estimation of Robot Manipulators: An Experimental Comparison," *IEEE Open Journal of Control Systems*, vol. 2, pp. 1–11, 2023.

[5] **S. B. Liu**, B. Schürmann, and M. Althoff, "Guarantees for Real Robotic Systems: Unifying Formal Controller Synthesis and Reachset-Conformant Identification," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3776–3790, 2023.

The first listed paper [1] introduces the notion of reachset conformance for robotics, and proposes the first algorithm for uncertainty identification. Our works in [2, 3] demonstrate examples of human-robot interaction, where reachset-conformant models are found and used for safety verification. In both cases, *online verification* is applied. The papers in [4,5] focus on control design: the work in [4] compares velocity estimators for robotics using classical design methods, while [5] shows how controllers can be designed using reachability analysis and reachset-conformant models. The remaining co-authored five publications are also strongly related to the objectives of this dissertation:

[6] M. Wagner, **S. B. Liu**, A. Giusti, and M. Althoff, "Interval-arithmetic-based trajectory scaling and collision detection for robots with uncertain dynamics," in *Proc. of the IEEE International Conference on Robotic Computing (IRC)*, 2018, pp. 41–48.

[7] **S. B. Liu** and M. Althoff, "Optimizing performance in automation through modular robots," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4044–4050.

[8] A. Giusti, **S. B. Liu**, and M. Althoff, "Interval-Arithmetic-Based Robust Control of Fully Actuated Mechanical Systems," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 1, pp. 1–13, 2021.

[9] M. Althoff, A. Giusti, **S. B. Liu**, and A. Pereira, "Effortless creation of safe robots from modules through self-programming and self-verification," *Science Robotics*, vol. 4, no. 31, p. eaaw1924, 2019.

[10] S. Schepp, J. Thumm, **S. B. Liu**, and M. Althoff, "SaRA : A Tool for Safe Human –Robot Coexistence and Collaboration through Reachability Analysis," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1–7.

The work in [6] uses a reachset-conformant inverse dynamics model for trajectory scaling and collision detection, and a similar model is used in [8] for robust control. The works in [7] and [9] concern modular robots and their use for safe human-robot coexistence, while [10] introduces a tool for computing reachable sets of human motion.

# 2 Methodology

Developing adequate verification models is essential to formally ensure safe human-robot interaction. This chapter summarizes and reviews the methodology that I developed through the course of my research. The purpose of this chapter is to give a general context to the publications. The publications in Appendix A provide further details on methods and real application examples. In this chapter, some basic definitions will be provided; I will denote sets using calligraphic letters (e.g., $\mathcal{A}$), matrices using upper case letters (e.g., $A$), vectors using $\vec{\cdot}$, and scalar values using lower case letters (e.g., $a$). In addition, I use $a(.)$ as a notation for the whole trajectory and $a(t)$ as the value $a$ at time $t$.

## 2.1 Modelling for safety verification

Exactly characterizing real systems through mathematical descriptions is commonly accepted to be an impossible feat. Thus, the purpose of models is to reduce the characterization effort by leaving out unnecessary details. How models are developed primarily depends on the given objective, i.e., a first principles model is developed to fundamentally understand the behavior of a system. In contrast, control models are developed to optimize the performance of a controlled system.

This thesis is concerned with developing verification models for analyzing the safety of a system. If it can be proven that an output of a system can never enter unsafe sets, this is referred to as "verifying a safety property". A property is a safety property if it can be defined as a set of time-dependent unsafe sets $\mathcal{B}(t)$, that the reachable set of the system $\mathcal{R}(t)$ should never reach at any time [15, Equation 6]:

$$\forall t : \mathcal{R}(t) \cap \mathcal{B}(t) = \emptyset. \tag{2.1}$$

In human-robot interaction, such unsafe sets are usually associated with unintended collisions [17, 19], or in case of an intended collision, the avoidance of human pain or injury [18].

### 2.1.1 Models and reachability analysis

To prove safety, *reachability analysis* [13] is used to compute the set of all possible behaviors of a human-robot interaction system and show that the system never enters an unsafe set. In the following examples, human-robot interaction is modeled using the state function $f$ and the output function $g$:

$$\dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), \vec{p}(t)), \tag{2.2a}$$
$$\vec{y}(t) = g(\vec{x}(t), \vec{u}(t), \vec{p}(t)), \tag{2.2b}$$
$$\vec{p}(t) \in \mathcal{P}, \tag{2.2c}$$

where $\vec{x}(t)$ is the modeled state, $\vec{u}(t)$ is the measurable system input, $\vec{y}(t)$ is the measurable system output, and $\vec{p}(t)$ are non-measurable time-varying model parameters that belong to a bounded set $\mathcal{P}$ to represent uncertainty in our system. Given an initial state $\vec{x}(0) = \vec{x}_0$, an input trajectory $\vec{u}(.)$, and a possible parameter trajectory $\vec{p}(.)$, the system in (2.2) admits a unique trajectory of the output, denoted in the following by $\vec{y}(., \vec{x}_0, \vec{u}(.), \vec{p}(.))$. The reachable set of the system in (2.2) at time $t \geq 0$ from a set of initial states $\mathcal{X}_0$ is the set of all possible outputs:

$$\mathcal{R}(t) = \{\vec{y}(t, \vec{x}_0, \vec{u}(.), \vec{p}(.)) \mid \vec{x}_0 \in \mathcal{X}_0, \forall \tau \in [0, t] : \vec{p}(\tau) \in \mathcal{P}\}, \tag{2.3}$$

where $u(.)$ is a trajectory from time $[0, t]$. Compared to the definition in [13, Equation 2], the above definition considers an additional output function and distinguishes between measurable inputs $\vec{u}$ and unmeasurable disturbances/model parameters $\vec{p}$.

### 2.1.2 Reachset conformance

As initially described, a model only partially characterizes the real system. In our case, (2.2a) and (2.2b) are making assumptions on the dynamical behavior, while (2.2c) is assuming the uncertainty of the parameter set $\mathcal{P}$. Formally verifying safety for a model does not necessarily mean that safety is verified for the real interaction. To bridge this formal gap, Roehm et al. [29] introduced *reachset conformance* to relate a model with a real system, and thus to allow the safety property (2.1) verified on a model to transfer to the real system. Furthermore, the authors point out in [15] that reachset conformance is a necessary and sufficient relation.

A verification model is *reachset conformant* if its reachable set not only encloses the model behavior but *also* encloses all possible behaviors of the real system. Only if this property holds can we ascertain that the real system is safe because an unsafe state non-reachable by the model is also non-reachable by the real system. The testing of this substantial property is called *reachset conformance checking* and can be executed as follows: Given both the real system and the model, and the same series of inputs $\vec{u}(t)$ belonging to a finite set of input trajectories $\mathcal{U}$, where $t \in [0..t^*]$ and $t^*$ is a finite time horizon, the reachable set $\mathcal{R}(t)$ of the verification model shall always include the measured output $\vec{y}_m(t)$ of the real system:

$$\forall \vec{u}(t) \in \mathcal{U} : \forall t \in [0, t^*] : \vec{y}_m(t) \in \mathcal{R}(t). \tag{2.4}$$

The example shown in Figure 2.1 visualizes reachable sets and reachset conformance. As seen on the right side, the model is not reachset conformant because one of the measured output trajectories is not within the reachable set of the model.

#### Formal reachset conformance vs. reachset conformance checking

Notice that the above definition of reachset conformance checking is not a formal proof of reachset conformance as defined in [14, Section 3.5]; however, it is possible to increase our confidence in reachset conformance through sufficient testing. To safeguard against scenarios where the real system is not behaving conformantly, I propose implementing (2.4) as an online monitor. Such a monitor would detect, whether the assumed verification model is failing and introduce fail-safe actions. One example is the safety filter in [65], which monitors reachset conformance and automatically adapts the verification model and the model-based controller in case of detecting a non-conformant behavior.

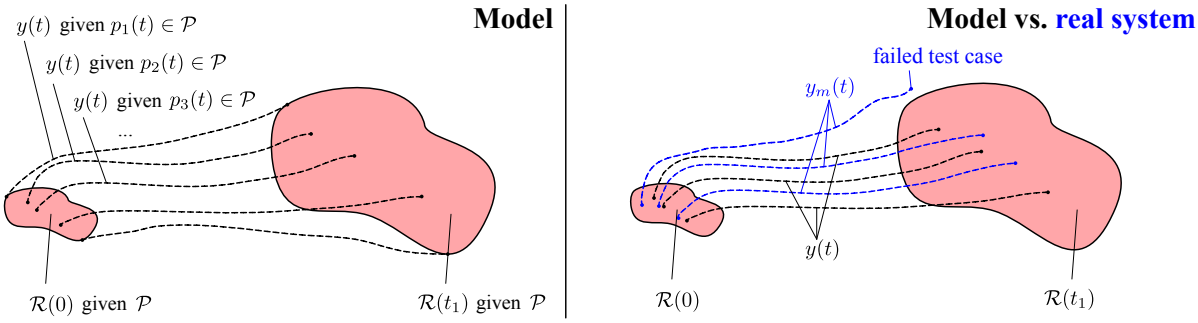


**Figure 2.1:** Left side: the computed reachable set $\mathcal{R}(t)$ encloses all possible behaviors $\vec{y}(t)$ of the model for the same input $\vec{u}(t)$ given a set of possible parameters $\mathcal{P}$. Right side: the computed set $\mathcal{R}(t)$ does not enclose all behaviors $\vec{y}_m(t)$ measured from the real system; thus, the model is not reachset conformant.

### 2.1.3 Reachset conformance of linear systems

In this thesis, linear systems receive special attention due to properties, such as the separation principle [5], that allow us to reduce the problem of reachset conformance checking in (2.4) to a linear inequality. Let us use the notation $a[k]$ to express the value of $a$ at time $k\Delta t$, where $k \in \{0, 1, \dots\}$ and $\Delta t$ is the sampling time. The following difference equations define linear systems:

$$\begin{aligned} \vec{x}[k+1] &= A\vec{x}[k] + B\vec{u}[k] + \vec{w}[k], \\ \vec{y}[k] &= C\vec{x}[k] + D\vec{u}[k] + \vec{v}[k], \end{aligned} \tag{2.5}$$

where $A, B, C, D$ are matrices of proper dimension; and $\vec{w}[k] \in \mathcal{W}$ and $\vec{v}[k] \in \mathcal{V}$ are uncertain parameters representing additive disturbances and sensor noise, respectively. Furthermore, let us use *zonotopes* to represent the parameter sets. A zonotope $\mathcal{Z}$ is defined by a center $\vec{c}$ and a generator matrix $G$ of proper dimension, where $g^{(h)}$ is its $h$-th column:

$$\mathcal{Z} = (\vec{c}, G) := \left\{ \vec{x} = \vec{c} + \sum_{h=1}^{s} \beta_h g^{(h)} \middle| \beta_h \in [-1, 1] \right\}.$$

For our parameter sets, let us define *scaled zonotopes* $\mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$ and $\mathcal{V} = (\vec{c}_V, G'_V \operatorname{diag}(\vec{\alpha}_V))$, where $\vec{\alpha}_W, \vec{\alpha}_V$ are scaling factors for each of the zonotope generators represented in $G'_W$ and $G'_V$. Zonotopes and scaled zonotopes are visualized in Figure 2.2. For the introduced linear system, let me outline the following proposition.

**Proposition 1 (Reachset conformance checking as a linear inequality)** Given a linear system whose dynamics is described by (2.5) and whose uncertain parameter sets are described by $\mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$ and $\mathcal{V} = (\vec{c}_V, G'_V \operatorname{diag}(\vec{\alpha}_V))$, the reachset conformance checking in (2.4) is equal to a linear inequality of the form $N(t)\vec{y}_m(t) \leq H(t)\vec{\xi}$, where $\vec{\xi} = [\vec{c}_W, \vec{\alpha}_W, \vec{c}_V, \vec{\alpha}_V]^T$ are its variables.

*Proof* I am referring to Theorem 1 in [5]. The basic idea is to represent the reachable sets as a set of halfspaces $N(t)\vec{y} \leq \vec{h}(t)$, where $N$ is a matrix of normal vectors, and $\vec{h}$ is a vector of distances. As shown in [5], $\vec{h}(t)$ is linear in $\vec{\xi}$, such that $\vec{h}(t) = H(t)\vec{\xi}$. ■

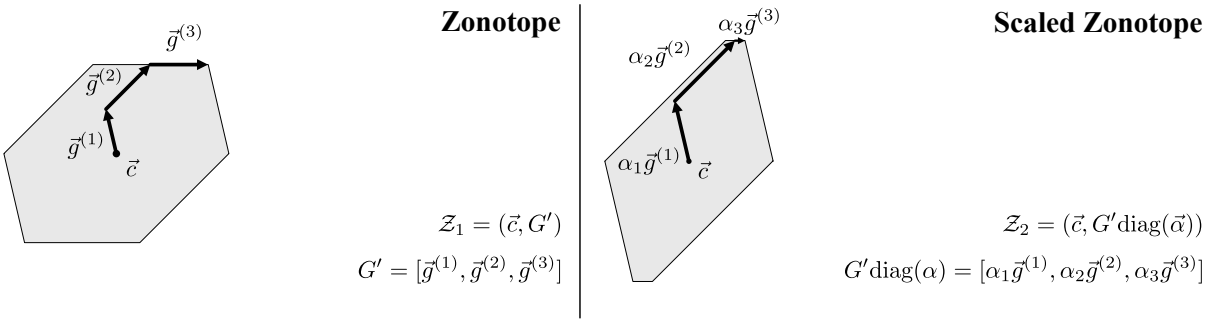


**Figure 2.2:** Left side: the zonotope $\mathcal{Z}_1$ is defined by a center $\vec{c}$ and three generators in $G'$. Right side: the zonotope $\mathcal{Z}_2$ is a scaled version of $\mathcal{Z}_1$, where the generators are the same, but their lengths have been altered by $\alpha_1, \alpha_2, \alpha_3$.

Expressing reachset conformance checking as a linear inequality significantly reduces the computational complexity of identifying reachset-conformant linear systems, as I will describe in Section 2.2.

### 2.1.4 Examples

Table 2.1 lists examples of safety problems published with involvement from the author of this thesis, which have been analyzed using the methodology introduced in this chapter. For a detailed discussion, let us look at an example from [2].

This example is also visualized in Figure 2.3. I consider that humans and robots only move along one dimension for presentation purposes, while the full problem can be found in the publication. The safety problem is to verify online that a proposed series of robot inputs can never lead to a collision between humans and a mobile robot while the robot is moving. Therefore, the unsafe set is defined by

$$\{x_{robot} \mid (\mathrm{occ}(x_{robot}) \cap \mathrm{occ}(x_{human}) \neq \emptyset) \wedge (|\dot{x}_{robot}| > 0)\}$$

where $x_{human}, x_{robot}$ are the human and robot position, respectively, and $\mathrm{occ}(x)$ is the occupancy given a position. Analyzing this problem requires a model for robot motion and a model for human motion. The main difference between these two subsystems is that robot motion can be accurately controlled, i.e., the motion uncertainty is bounded by a control error. In contrast, the control of human motion is unknown to us, which is why I conservatively assume motion uncertainty that is only bounded by the physical capability of humans. For the sake of brevity, I am only illustrating the human motion model and its conformance checking here. I model human physical capability by limiting their acceleration. Therefore, the dynamics are a double integrator, where the state is $\vec{x} = [x_{human}, \dot{x}_{human}]^T$:

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w, \quad w \in \mathcal{W} \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} \vec{x} \end{aligned}$$

where $\mathcal{W} = \{w \in \mathbb{R} \mid -a_{max} < w < a_{max}\}$ is the set of possible accelerations limited by $a_{max}$. Since the safety verification is interested in $x_{human}$ from the human subsystem (and $x_{robot}, \dot{x}_{robot}$ from the robot subsystem), the human motion model must be reachset conformant

**Table 2.1:** Safety verification examples with author's involvement

| | Safety problem | Authors | Publication, Year |
|---|---|---|---|
| [2] | Verify that a moving mobile robot does not collide with walking humans. | **Liu**, Roehm, Heinzemann, Lütkebohle, Oehlerking, Althoff | IEEE/RSJ International Conference on Intelligent Robots and Systems, 2017 |
| [3] | Verify that the interaction force at an unintended collision of a robot with a human hand does not exceed a force limit. | **Liu**, Althoff | IEEE/RSJ International Conference on Intelligent Robots and Systems, 2021 |
| [5] | Synthesize a tracking controller for a robot arm and verify that torque constraints are met. | **Liu**, Schürmann, Althoff | IEEE Transactions on Robotics, 2023 |
| [6] | Ensure torque constraints for motion planning, collision detection. | Wagner, **Liu**, Giusti, Althoff | IEEE International Conference on Robotic Computing, 2018 |
| [8] | Ensure robust stability of a robot tracking controller despite perturbation in the inverse dynamics model. | Giusti, **Liu** Althoff | IEEE Transactions on Control Systems Technology, 2021 |
| [9] | Verify that a moving robot arm does not collide with a human upper body. | Althoff, Giusti **Liu**, Pereira | Science Robotics, 2019 |


**Figure 2.3:** The safety problem presented in [2] is for a mobile robot (blue) to avoid colliding with humans (green). Safety is verified by computing the reachable sets of humans and robots and determining that these sets are not intersecting. Conformance checking for the human motion model was performed using a pedestrian video dataset [66].

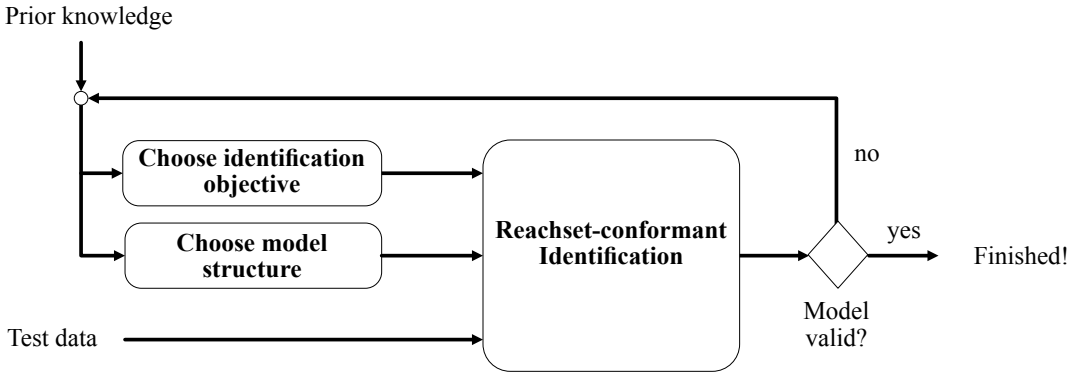


**Figure 2.4:** The identification framework proposed by this thesis follows the general system identification procedure suggested by Ljung in [67, Chapter 1.4].

for the output $y = x_{human}$. As demonstrated in [2], reachset conformance checking was done by recording the position of real human motion and comparing the measured positions with the computed reachable sets from the above model. In theory, I could also choose $y = [x_{human}, \dot{x}_{human}]^T$, but adding another output only increases the number of constraints that the model must fulfill, which generally means larger reachable sets might be required, i.e., adding unnecessary conservativeness (see similar discussion in [5, Remark 2]).

## 2.2 Identification

As described in [15], reachset conformance is a necessary and sufficient property for a verification model when it comes to safety verification using reachable sets. Usually, the uncertain parameter set $\mathcal{P}$ is unknown, and sometimes the state and output function can also be unknown. Even if all parts of the model are given (e.g., through a first principles modeling), the proposed model may not be reachset conformant or could lead to unnecessarily large reachable sets due to conservative assumptions.

The idea behind *reachset-conformant identification* is to obtain algorithmic support for finding a reachset-conformant model that minimizes conservativeness. Figure 2.4 depicts the proposed identification framework and is inspired by the general system identification procedure suggested by Ljung in [67, Chapter 1.4]. To be able to run algorithms for reachset-conformant identification (Section 2.2.2), three inputs are required:

- an identification objective (Section 2.2.1) shall be chosen, which will serve as the cost function to optimize for and which depends on the final application of the verification model,
- a model structure (Section 2.2.2) shall be chosen that best suits the problem at hand, e.g., parameterized state/output functions. The chosen structure depends on the amount of prior knowledge we intend to apply, and it can range from black-box models (function structure or parameter unknown) to white-box models (functions fully known, parameter set unknown),
- a data set obtained from the real system for identification.

Ideally, the reachset-conformant identification algorithm converges to a solution, and it shall be determined whether that model is valid for the corresponding purposes. For safety verification using reachable sets, it is typically checked whether that model is overly conservative and leads to reachable sets that are too large. In that case, safety verification would fail due to excessive scenarios the model predicts that the real system does not entail. Then, the inputs to the identification algorithm shall be revisited, e.g., adding details to the model structure or changing the objective function. In the following subsections, I will review the steps of this framework.

### 2.2.1 Identification objectives

First and foremost, the identified model should be reachset conformant to the data that is provided, i.e., (2.4) must be fulfilled and thus constrains the solution space. Within the solution space, an identification objective function decides which goal our model optimizes.

Generally, objective functions should help reduce conservativeness regarding the safety goal, i.e., the model should not predict scenarios the real system cannot show. Minimizing a norm of the reachable set of the model output is an obvious choice and was used most in the author's work, as shown in Table 2.2. Depending on the set representation, various norms are available to measure the size of a reachable set (e.g., see [68,69]). When zonotopes are used for linear systems, two norms stand out: the *interval norm* [5, Definition 2] and the *Frobenius norm* [70, Equation 22]. As shown in Appendices A and B in [70] for linear systems, the interval norm is linear in $\vec{\xi}$ and the Frobenius norm is quadratic in $\vec{\xi}$, which are valuable properties for creating efficient identification algorithms, as I will present in the following subsection.

In [5], the reachable set of the tracking error was used in the objective function, which was not the model output but the output of a closed-loop system, i.e., the outcome of a model interacting with a controller system. The advantage of that approach is that controller synthesis and reachset-conformant identification are unified into a single optimization problem with a shared objective function instead of a separate optimization approach leading to suboptimal controllers. This was demonstrated in Section IV-A in [5], which showed that for that specific example, a unified optimization approach is superior to a separate optimization approach, i.e., providing a better-performing controller. As shown in [65, Equation 8], another possible option is to minimize the parameter set $\mathcal{P}$ (in that work called $\mathcal{W}$) directly.

### 2.2.2 Model structures and identification algorithms

Let me formulate the general problem of reachset-conformant identification: the goal is to find an optimal model (2.2) according to some cost function $h$ that satisfies the reachset conformance constraint (2.4):

$$\min_{f,g,\mathcal{P}} \quad h(f,g,\mathcal{P}), \tag{2.6a}$$

$$\text{subject to} \quad \forall \vec{u}(t) \in \mathcal{U} : \forall t \in [0,t^*] : \vec{y}_m(t) \in \mathcal{R}(t,f,g,\mathcal{P}). \tag{2.6b}$$

Potential cost functions were already discussed in Section 2.2.1. Let me discuss the algorithms for identification. These mainly depend on the desired model structure. In this work, I am differentiating between three possible types of model structures:

**Table 2.2:** Identification objectives in the author's work

| | Safety problem | Model | Minimize the norm of reachable set of ... | Which output? |
|---|---|---|---|---|
| [1] | - | Robot forward dynamics | Robot position and velocity | Model |
| [2] | Mobile robot collision avoidance | Human motion on 2D plane | Humans in position space | Model |
| [3] | Limit collision force | Coupled human-robot dynamics | Human-robot collision force | Model |
| [5] | Tracking control with input constraints | Robot forward dynamics | Robot position and velocity tracking error | Closed-loop |
| [6] | Motion planning with torque constraints | Robot inverse dynamics | Robot inverse dynamics perturbation | Model |
| [8] | Robust stability | Robot inverse dynamics | Robot inverse dynamics perturbation | Model |
| [9] | Robot arm collision avoidance | Human arm motion in 3D space | human arms in position space | Model |

- **White-box models** are models, where the state and output functions $f$ and $g$ are fully known and fixed, and the set of variable uncertain parameters $\mathcal{P}$ is known.
- **Grey-box models** are models where the state and output functions $f(p_n)$ and $g(p_n)$ are known but depend on some variable parameters $\vec{p}_n$, which are not part of a set, and the variable *uncertain* parameter set $\mathcal{P}$.
- **Black-box models** are models, where the state and output functions $f$ and $g$ and the uncertain set $\mathcal{P}$ are all unknown.

In the following, identification algorithms for linear systems are presented.

#### 2.2.2.1 White-box algorithms

For linear systems, linear or quadratic programming algorithms [71] can be used for solving white-box identification.

**Proposition 2 (Linear programming for white-box identification of linear systems)** For linear systems as described in (2.5), where

- the matrices $A, B, C, D$ are known,
- the set $\mathcal{P} := \mathcal{W} \times \mathcal{V}$ is unknown, but is described by the scaled zonotopes $\mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$ and $\mathcal{V} = (\vec{c}_V, G'_V \operatorname{diag}(\vec{\alpha}_V))$, where the matrices $G'_W$ and $G'_V$ are known and $\vec{\xi} = [\vec{c}_W, \vec{\alpha}_W, \vec{c}_V, \vec{\alpha}_V]^T$ is unknown,
- the *interval* norm [70, Section 3.1] has been chosen as the cost function,

the reachset-conformant identification problem (2.6) is a linear program.

*Proof* As per Proposition 1, the constraint in (2.6b) is a linear inequality, and as per proof of Theorem 1 in [70], the cost function in (2.6a) is linear in $\vec{\xi}$. ■

**Proposition 3 (Quadratic programming for white-box identification of linear systems)** For linear systems as described in (2.5), where

- the matrices $A, B, C, D$ are known,
- the set $\mathcal{P} := \mathcal{W} \times \mathcal{V}$ is unknown, but is described by the scaled zonotopes $\mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$ and $\mathcal{V} = (\vec{c}_V, G'_V \operatorname{diag}(\vec{\alpha}_V))$, where the matrices $G'_W$ and $G'_V$ are known and $\vec{\xi} = [\vec{c}_W, \vec{\alpha}_W, \vec{c}_V, \vec{\alpha}_V]^T$ is unknown,
- the *Frobenius* norm [70, Section 3.2] has been chosen as the cost function,

the reachset-conformant identification problem (2.6) is a quadratic program.

*Proof* As per Proposition 1, the constraint in (2.6b) is a linear inequality, and as per proof of Theorem 2 in [70], the cost function in (2.6a) is quadratic in $\vec{\xi}$. ■

#### 2.2.2.2 Grey-box algorithms

For linear systems, grey-box models are a generalization of white-box models, where a limited amount of selected parameters $\vec{p}_n$ are unknown, such that the matrices $A(\vec{p}_n), B(\vec{p}_n), C(\vec{p}_n)$, $D(\vec{p}_n), G'_W(\vec{p}_n)$ depend on $\vec{p}_n$. As proposed in [3], I suggest to make use of the white-box algorithms in a nested fashion:

- an outer loop optimization using nonlinear programming algorithms [71] selects candidates for $\vec{p}_n$,
- an inner loop optimization using white-box identification algorithms selects the optimal $\vec{p}$ given the $\vec{p}_n$ candidate. It returns the optimal cost for that candidate to the outer loop.

As an example, the state function of the verification model in [3] for a physical human-robot interaction is defined by

$$\vec{x} = A(\vec{p}_n)\vec{x} + [0,1,0,0]^T u + w, \quad w \in \mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$$

$$A(\vec{p}_n) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(k_r+k_h)}{m_r} & -\frac{(d_h+d_r)}{m_r} & \frac{k_r}{m_r} & \frac{d_h}{m_r} \\ 0 & 0 & 0 & 1 \\ \frac{k_h}{m_r} & \frac{d_h}{m_r} & -\frac{k_h}{m_r} & -\frac{d_h}{m_r} \end{bmatrix}, \vec{p}_n = [m_h, k_h, d_h]^T$$

where $m_r, k_r, d_r$ are the known mass, stiffness, and damping of the impedance-controlled robot and $m_h, k_h, d_h$ are the unknown mass, stiffness, and damping of the human hand. Figure 2.5 illustrates this mass-spring-damper model. In this case, the outer loop optimizes $m_h, k_h, d_h$, while the inner loop optimizes $\vec{c}_W$ and $\vec{\alpha}_W$.

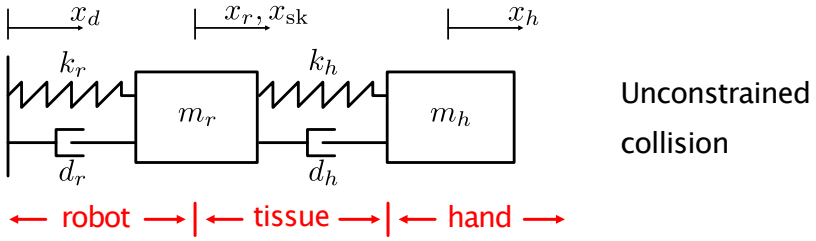


**Figure 2.5:** The safety problem presented in [3] is for the interaction force between a robot end-effector and a human hand to stay below a certain threshold. Safety is verified by computing the reachable sets of this mass-spring-damper system. Conformance checking for this model was performed using a real collision experiments.

#### 2.2.2.3 Black-box algorithms

This thesis does not explore black-box algorithms, but literature examples are reviewed for completeness. Assuming $\vec{y} = \vec{x}$ as the output function, Gruber and Althoff [65] use linear programming to simultaneously find optimal matrices $A, B$ for the state function and an optimal $\mathcal{W}$ such that the model is reachset conformant. Alanwar et al. [72] computes conformant reachable sets assuming $\vec{y} = \vec{x} + \vec{v}$ as the output function by generating sets of matrices $A, B$, where the true matrix is guaranteed to be contained. The authors also present extensions to nonlinear systems.

#### 2.2.2.4 Model structure selection

A chosen model structure may not be the most suitable one. For safety verification, models that lead to large reachable sets are more likely to detect false positive safety violations, so less conservative models are preferred. An inefficient model structure becomes noticeable if the optimal reachset-conformant model still leads to large reachable sets, and verifying safety becomes difficult.

Choosing the model structure also has an impact on both the identification algorithm as well as the reachability analysis technique. If a nonlinear programming technique is necessary to solve the identification in (2.6) (e.g., due to a nonlinear cost function), it may not converge to a global optimum. In addition, some reachability analysis techniques, e.g., for nonlinear systems, experience a *wrapping effect* [13], i.e., the over-approximation error of the reachable sets accumulates over many time steps. In contrast, reachability analysis of linear systems (e.g., [5, Equation 7]) can be exact and computationally efficient compared to nonlinear systems and does not experience the wrapping effect. I recommend starting with a simple model structure for the first iteration of model identification and adding more detail in further iterations when the model is too conservative and therefore the verification fails.

To illustrate this, let me demonstrate a model selection example for the robot forward dynamics in [5], where I evaluate three candidate white-box models with increasing model order and complexity. Further details of this comparison can be found in [73, Section IV-A]. Let us assume the robot has an internal feedback linearization controller [5, Equation 9] that linearizes the input-output dynamics. Given a rigid-body model of a robot [5, Equation 8], a model is obtained where the state is $\vec{x}_1 = [q, \dot{q}]$, $q$ is the position of a robot axis, and $\Delta t$ is the sampling rate. Adding additive process noise $\vec{w}$ and sensor measurement error $v$, **candidate**

**model 1** is a double integrator that is described by:

$$\vec{x}_1[k+1] = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \vec{x}_1[k] + \begin{bmatrix} \Delta t^2/2 \\ \Delta t \end{bmatrix} u[k] + \vec{w}[k],$$
$$\vec{y}[k] = \vec{x}_1[k] + \vec{v}[k].$$

In **candidate model 2**, I take into account that there is a transmission delay of one sampling time step for the input to reach the robot and a transmission delay for the measured output to reach the controller:

$$\vec{x}_2[k+1] = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & \Delta t & \Delta t^2/2 \\ 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}_2[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u[k] + \vec{w}[k],$$
$$\vec{y}[k] = \vec{x}_2[k] + \vec{v}[k],$$

where $\vec{x}_2[k] = [q[k-1], \dot{q}[k-1], q[k], \dot{q}[k], u[k-1]]$. In **candidate model 3**, I take into account, apart from the transmission delay, that only the position $q$ is measured, and the velocity is externally estimated using the following high-gain observer:

$$\begin{bmatrix} \dot{\hat{q}} \\ \ddot{\hat{q}} \end{bmatrix} = \begin{bmatrix} -h_1/\epsilon & 1 \\ -h_2/\epsilon^2 & 0 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \dot{\hat{q}} \end{bmatrix} + \begin{bmatrix} h_1/\epsilon \\ h_2/\epsilon^2 \end{bmatrix} q_m,$$
$$y = \begin{bmatrix} \hat{q} \\ \dot{\hat{q}} \end{bmatrix},$$

where $h_1 = 15, h_2 = 30$, and $\epsilon = 0.01$ are the gains. I use the bilinear transformation discussed in [74] to discretize the observer. The input of the observer is $q_m$, which is the measured robot position and the output of the following robot model:

$$\vec{x}_3[k+1] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & \Delta t & \Delta t^2/2 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}_3[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u[k] + \vec{w}[k],$$
$$\vec{q}_m[k] = \vec{x}_3[k] + v[k].$$

The model order and state dimension of *candidate model 1* is two, the state dimension of *candidate model 2* is five, and the state dimension of *candidate model 3* is six (taking into account both the observer and the robot model). To compare these candidate models, I use the same dataset, where I recorded the one-dimensional input $u$ and the two-dimensional output $\vec{y}_m$ (robot position and velocity) from each axis of a six-degrees-of-freedom robot manipulator. Linear programming (see Proposition 2) is applied to obtain the optimal cost.

The resulting cost for each candidate model and each robot axis are listed in Table 2.3. As seen across all robot axes, the interval norms of both candidate models 2 and 3 are significantly smaller compared to candidate model 1. For our work in [5], where the application is to synthesize a motion controller given the robot position and velocity, I ultimately chose candidate model 3 because it accurately models the interacting dynamics between the high-gain

**Table 2.3:** Comparison of the cost of the candidate models for each robot axis for model structure selection

| **Candidate** | Axis 1 | Axis 2 | Axis 3 | Axis 4 | Axis 5 | Axis 6 |
|---|---|---|---|---|---|---|
| Model 1 | 0.0322 | 0.0422 | 0.0325 | 0.0309 | 0.0505 | 0.0405 |
| Model 2 | 0.0025 | 0.0044 | 0.0022 | 0.0023 | 0.0035 | 0.0050 |
| Model 3 | 0.0022 | 0.0041 | 0.0021 | 0.0023 | 0.0032 | 0.0041 |

**Table 2.4:** Identification algorithms used in the author's work

| | Model | Structure | Dynamics | Algorithm |
|---|---|---|---|---|
| [1] | Robot forward dynamics | white-box | linear | linear programming |
| [2] | Human motion on 2D plane | white-box | linear | manual tuning |
| [3] | Coupled human-robot dynamics | grey-box | linear | nonlinear programming with nested linear program |
| [5] | Robot forward dynamics | white-box | linear | nonlinear programming |
| [6] | Robot inverse dynamics | grey-box | nonlinear | manual tuning |
| [8] | Robot inverse dynamics | grey-box | nonlinear | nonlinear programming |
| [9] | Human arm motion in 3D space | white-box | linear | manual tuning |

observer and the motion controller, and thus provides the smallest reachable set. Although candidate model 1 is the simplest, it ultimately was too conservative for our work in [5]. Given the minimal difference of candidate model 2 and 3, choosing candidate model 2 would also be justified.

### 2.2.3 Examples

Table 2.4 lists the author's work, the model, the model structure type, the dynamics type, and the algorithms used for identification. In [5], reachset-conformant identification is combined with controller synthesis in a single optimization, which is solved using nonlinear programming techniques. In [6,8], the robot inverse dynamics model is a nonlinear output function without states. Please note that for some mechanical systems, the inverse dynamics models are linear to the dynamic parameters [75, Section 7.2.2], which would make it conceivable to use linear programming approaches. Appendix A shows further details on these examples.

## 2.3 Tools

With contributions from the author of this thesis, our research team has created tools that have been made available to the public to foster the adoption of formal methods in robotics and to identify verification models to prove safety.

- The tool SaRA[1] (*Safe Human-Robot Coexistence and Collaboration through Reachability Analysis*) in [10] can be used to compute the reachable occupancy of humans in 2D and 3D space, includes a ROS[2] (*Robot Operating System*) node for visualization, and makes the reachable sets available to other computing nodes. The computations are based on our work in [2] and [9].
- The tool CORA[3] (*COntinuous Reachability Analysis*) initially developed by Althoff [76] is a software for the reachability analysis of verification models with linear, nonlinear, and hybrid dynamics and their formal verification. Version 2023 [70] implements white-box and black-box identification algorithms (see Sec. 2.2.2) and reachset-conformance checking algorithms for linear systems.
- The tool *Unifying Formal Controller Synthesis and Reachset-Conformant Identification*[4] implements a solver for the optimization problem in [5, Equation 12] of the combined controller synthesis and reachset-conformant identification of linear systems. It also includes the dataset from a real robot, for which an optimal forward dynamics model, the optimal motion controller, and the observer are determined.

[1] `https://github.com/Sven-Schepp/SaRA`
[2] `https://www.ros.org`
[3] `https://cora.in.tum.de/`
[4] `https://doi.org/10.24433/CO.1635335.v1`

# 3 Conclusions and Future Work

This chapter discusses the conclusions and future work for this thesis's overall body of research. Further detailed discussions and findings are provided in each publication.

This thesis provides a framework for identifying verification models, such that these models are reachset conformant. This is achieved by measuring the real system and identifying the model and its uncertainties, such that the reachable set of the model covers all real system behaviors. For the first time in the domain of continuous dynamics and human-robot interaction, a systematic approach has been presented for transferring the safety property from a formally verified model to the actual human-robot interaction system. For each of the areas of contribution in the thesis, let me discuss the results, their implications, and future work in more detail:

### Model identification for safety verification

This thesis poses the general problem of identifying verification models for safety as a constrained optimization problem, where the reachset conformance relation is represented as the constraint. Applying this optimization problem to models with linear and nonlinear dynamics is explored. The main steps are to choose an identification objective, a model structure, and a suitable identification algorithm. Based on the identification objective, we can determine whether the model is appropriate and fulfills our needs; if not (e.g., because it is too conservative) we can choose a different identification objective or model structure and repeat this procedure. I propose white-box and grey-box algorithms for linear dynamics, which are efficient to solve. The soundness and effectiveness of this framework have been tested on multiple examples from the human-robot interaction domain in [1–3, 5, 6, 8, 9], where I have found models for the robot forward dynamics, inverse dynamics, human motion prediction, and physical human-robot interaction to solve a range of safety problems.

This thesis provides a starting point for continued exploration in identifying verification models. Nonlinear dynamics remains a significant challenge since many simplifications for linear dynamics do not generalize to nonlinear dynamics. One possible approach is a decomposition into piecewise linear dynamics (such as in [58]) in terms of time or space. Also, hybrid dynamics with switching continuous dynamics and jumping continuous variables remain challenging. The work in [77] provides a starting point in this direction. Black-box identification (or data-driven reachability analysis, as often called in literature) has remaining challenges considering system dynamics where there are fewer outputs than states. Here, it is challenging to determine the model order, i.e., how do I choose an appropriate dimension for state $x$, or what amount of previous data is necessary for ARMAX models [78].

The discipline of test design and test selection also remains a future work. The examples in this thesis mostly use random testing, which, in theory, is probabilistically complete but may not be efficient, especially when gathering large amounts of data from real systems. It may be more efficient to select tests that explore the boundaries and edge cases of the model,

i.e., *falsification-based test selection*, or to find suitable coverage metrics, i.e., *coverage-based test selection*, to filter out tests that do not provide valuable data [14, Section 5]. In addition, defining a *test end criterion* can be helpful to measure the confidence in a model. Finally, I also suggest further exploring online conformance monitoring, such as proposed in [33, Section 9] and in [65], to further safeguard the system against unforeseen non-conformant behavior and to restore formal safety.

### Formal methods: online safety verification

Online safety verification was the main scheme in [2, 3, 9, 10] for proving safety. It uses the identified model to formally prove a safety property at runtime through online verification and fail-safe planning. This method was first proposed in [79] for automated vehicles, and my work extends this to more safety properties in the human-robot interaction domain. I have demonstrated that online verification ensures safety at all times (assuming correct modeling) and leads to better task efficiency than traditional industry approaches. Online verification has two significant advantages that contribute to better task efficiency: 1) it supports continuous and hybrid dynamics with uncertainty for higher-fidelity modeling that is less conservative than the simple models assumed in the safety standards, and 2) it only needs to verify the current control inputs for the current scenario, allowing more aggressive behaviors, as long as safety is ensured. My work in [2] compares online verification with ISO 13482 approaches and demonstrates that a mobile robot in dense human crowds can reach the goal 1.4 to 3.5 times faster while never colliding with humans when the robot is moving. My work in [3] applies a different safety property, allowing collisions during robot motion as long as a force limit is ensured, further increasing potential task efficiency. All the models in these works have been identified using data from real systems. Most of the approaches have been tested on actual robots. I believe that my work has contributed significantly to closing the "reality gap" of formal methods and "unfreezing" the robot operating near humans.

Due to the capability of online verification to handle better models with uncertainty, I see great potential in proving the safety of further tasks in physical human-robot interaction, where the main challenge is to predict highly uncertain human behavior. A different area for exploring the application of reachability analysis is verifying robot tasks involving stiff contact with the environment [80].

### Formal methods: controller synthesis vs. traditional controller tuning

The publication in [5] and my co-authored publication in [8] describe the use of reachset-conformant models in obtaining formally safe controllers. In contrast, my work in [4] describes a rather traditional approach to controller design. My work in [5] minimizes the reachable set of the tracking error of the closed-loop system, where the controller synthesis is performed with few validation iterations on the real system. The work in [8] uses the worst-case inverse dynamics perturbation to prove global uniform ultimate boundedness; the proof is valid for any model perturbation, but a model that minimizes the reachset-conformant perturbation improves the robust performance of that controller. The work in [4] minimizes an integral squared error (ISE) metric, where many controller-observer candidates are tested on a real robot system, and the ISE is measured per candidate. Formal controller synthesis provides safety guarantees, e.g., adhering to input constraints given all identified uncertainties, while

traditional tuning only tests specific scenarios, usually due to limited testing resources. In contrast, reachability analysis can account for all possible scenarios within the bounds of the parameter sets. Formal controller synthesis remains an attractive challenge for ensuring formally correct behavior safe-by-design controllers. In the following Appendix, I will provide the core publications of this thesis, including their summaries.

# Bibliography


[11] N. Webb, D. Smith, C. Ludwick, T. Victor, Q. Hommes, F. Favaro, G. Ivanov, and T. Daniel, "Waymo's safety methodologies and safety readiness determinations," 2020, arXiv:2011.00054.

[12] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for Robots: Guarantees and Feedback for Robot Behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 211–236, 2018.

[13] M. Althoff, G. Frehse, and A. Girard, "Set Propagation Techniques for Reachability Analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 396–395, 2021.

[14] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Model Conformance for Cyber-Physical Systems," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 3, pp. 1–26, 2019.

[15] H. Roehm, A. Rausch, and M. Althoff, "Reachset Conformance and Automatic Model Adaptation for Hybrid Systems," *Mathematics*, vol. 10, no. 19, p. 3567, 2022.

[16] ISO 12100:2010, "Safety of machinery – General principles for design – Risk assessment and risk reduction," Int. Org. for Standardization, Geneva, Switzerland, 2010.

[17] ISO 10218-1:2012-01, "Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots," Int. Org. for Standardization, Geneva, Switzerland, 2012.

[18] ISO/TS 15066:2016, "Robots and robotic devices - collaborative robots," Int. Org. for Standardization, Geneva, Switzerland, 2016.

[19] ISO 13482:2014, "Robots and robotic devices – Safety requirements for personal care robots," Int. Org. for Standardization, 2014.

[20] R. J. Kirschner, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Experimental Analysis of Impact Forces in Constrained Collisions According to ISO / TS 15066," in *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, 2021, pp. 1–5.

[21] P. Svarny, J. Rozlivek, L. Rustler, and M. Hoffmann, "3D Collision-Force-Map for Safe Human-Robot Collaboration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3829–3835.

[22] L. Scalera, R. Vidoni, and A. Giusti, "Optimal scaling of dynamic safety zones for collaborative robotics," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3822–3828.

[23] A. Pupa, M. Arrfou, G. Andreoni, and C. Secchi, "A Safety-Aware Kinodynamic Architecture for Human-Robot Collaboration," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4465–4471, 2021.

[24] K. Artemov, S. Kolyubin, and S. Stramigioli, "Multi-Stage Energy-Aware Motion Control with Exteroception-Defined Dynamic Safety Metric," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 9290–9296.

[25] C.-Y. Kuo, A. Schaarschmidt, Y. Cui, T. Asfour, and T. Matsubara, "Uncertainty-Aware Contact-Safe Model-Based Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3918–3925, 2021.

[26] Z. Xie, P. Xin, and P. Dames, "Towards Safe Navigation Through Crowded Dynamic Environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4934–4940.

[27] R. Cheng, R. M. Murray, and J. W. Burdick, "Limits of Probabilistic Safety Guarantees when Considering Human Uncertainty," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3182–3189.

[28] J. Bowen and V. Stavridou, "Safety-critical systems, formal methods and standards," *Software Engineering Journal*, vol. 8, no. 4, pp. 189–209, 1993.

[29] H. Roehm, J. Oehlerking, T. Heinz, and M. Althoff, "STL Model Checking of Continuous and Hybrid Systems," in *Automated Technology for Verification and Analysis*, 2016, pp. 412–427.

[30] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff, "Formalization of Interstate Traffic Rules in Temporal Logic," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 752–759.

[31] N. Fulton, S. Mitsch, J.-D. Quesel, M. Völp, and A. Platzer, "KeYmaera X: An Axiomatic Tactical Theorem Prover for Hybrid Systems," in *Automated Deduction – CADE-25*, 2015, pp. 527–538.

[32] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[33] S. Mitsch, K. Ghorbal, D. Vogelbacher, and A. Platzer, "Formal verification of obstacle avoidance and navigation of ground robots," *International Journal of Robotics Research*, vol. 36, no. 12, pp. 1312–1340, 2017.

[34] K. Majd, S. Yaghoubi, T. Yamaguchi, B. Hoxha, D. Prokhorov, and G. Fainekos, "Safe Navigation in Human Occupied Environments Using Sampling and Control Barrier Functions," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5794–5800.

[35] M. A. Murtaza, S. Aguilera, V. Azimi, and S. Hutchinson, "Real-Time Safety and Control of Robotic Manipulators with Torque Saturation in Operational Space," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 702–708.

[36] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6286–6291.

[37] M. Koschi and M. Althoff, "Set-Based Prediction of Traffic Participants Considering Occlusions and Traffic Rules," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 249–265, 2021.

[38] M. Althoff, S. Maierhofer, and C. Pek, "Provably-Correct and Comfortable Adaptive Cruise Control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 1, pp. 159–174, 2021.

[39] R. Muradore, D. Bresolin, L. Geretti, P. Fiorini, and T. Villa, "Robotic Surgery," *IEEE Robotics & Automation Magazine*, vol. 18, no. 3, pp. 24–32, 2011.

[40] L. Geretti, R. Muradore, D. Bresolin, P. Fiorini, and T. Villa, "Parametric formal verification: the robotic paint spraying case study," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9248–9253, 2017.

[41] Y. S. Shao, C. Chen, S. Kousik, and R. Vasudevan, "Reachability-Based Trajectory Safeguard (RTS): A Safe and Fast Reinforcement Learning Safety Layer for Continuous Control," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3663–3670, 2021.

[42] J. A. Vincent and M. Schwager, "Reachable Polyhedral Marching (RPM): A Safety Verification Algorithm for Robotic Systems with Deep Neural Network Components," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9029–9035.

[43] F.-C. Chou, M. Gibson, R. Bhadani, A. M. Bayen, and J. Sprinkle, "Reachability Analysis for FollowerStopper: Safety Analysis and Experimental Results," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8607–8613.

[44] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020.

[45] X. Yin and S. Li, "Recent advances on formal methods for safety and security of cyber-physical systems," *Control Theory and Technology*, vol. 18, no. 4, pp. 459–461, 2020.

[46] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[47] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[48] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, 2015, pp. 239–248.

[49] D. Hess, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1474–1481.

[50] B. Schürmann and M. Althoff, "Optimizing Sets of Solutions for Controlling Constrained Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 981–994, 2021.

[51] C. I. Vasile, X. Li, and C. Belta, "Reactive sampling-based path planning with temporal logic specifications," *International Journal of Robotics Research*, vol. 39, no. 8, pp. 1002–1028, 2020.

[52] G. Scher and H. Kress-Gazit, "Warehouse Automation in a Day: From Model to Implementation with Provable Guarantees," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 280–287.

[53] D. Calzolari, A. M. Giordano, and A. Albu-Schaffer, "Error Bounds for PD-Controlled Mechanical Systems Under Bounded Disturbances Using Interval Arithmetic," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1231–1238, 2020.

[54] M. M. Bersani, M. Soldo, C. Menghi, P. Pelliccione, and M. Rossi, "PuRSUE -from specification of robotic environments to synthesis of controllers," *Formal Aspects of Computing*, vol. 32, pp. 187–227, 2020.

[55] F. S. Barbosa, J. Karlsson, P. Tajvar, and J. Tumova, "Formal Methods for Robot Motion Planning with Time and Space Constraints (Extended Abstract)," in *Formal Modeling and Analysis of Timed Systems*, 2021, pp. 1–14.

[56] M. Mansouri, F. Pecora, and P. Schüller, "Combining Task and Motion Planning: Challenges and Guidelines," *Frontiers in Robotics and AI*, vol. 8, p. 637888, 2021.

[57] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-Driven Computation of Minimal Robust Control Invariant Set," in *2018 IEEE Conference on Decision and Control (CDC)*, 2019, pp. 4052–4058.

[58] S. Sadraddini and C. Belta, "Formal Guarantees in Data-Driven Model Identification and Control Synthesis," in *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, 2018, pp. 147–156.

[59] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–8.

[60] A. Vicino and G. Zappa, "Sequential approximation of feasible parameter sets for identification with set membership uncertainty," *IEEE Transactions on Automatic Control*, vol. 41, no. 6, pp. 774–785, 1996.

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

[62] M. Kieffer, E. Walter, and I. Simeonov, "Guaranteed nonlinear parameter estimation for continuous-time dynamical models," *Robust Control Design*, vol. 5, pp. 685–690, 2006.

[63] J. Bravo, T. Alamo, and E. Camacho, "Bounded error identification of systems with time-varying parameters," *IEEE Transactions on Automatic Control*, vol. 51, no. 7, pp. 1144–1150, 2006.

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

[65] F. Gruber and M. Althoff, "Scalable Robust Safety Filter with Unknown Disturbance Set," *IEEE Transactions on Automatic Control*, 2023, doi: 10.1109/TAC.2023.3292329.

[66] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 261–268.

[67] L. Ljung, *System Identification. Theory for the User*, 2nd ed. New Jersey: Prentice Hall, 1999.

[68] V. Gassmann and M. Althoff, "Scalable Zonotope-Ellipsoid Conversions using the Euclidean Zonotope Norm," in *2020 American Control Conference (ACC)*, 2020, pp. 4715–4721.

[69] A. Kulmburg and M. Althoff, "On the co-NP-completeness of the zonotope containment problem," *European Journal of Control*, vol. 62, pp. 84–91, 2021.

[70] M. Althoff, "Checking and Establishing Reachset Conformance in CORA 2023," in *Proceedings of 10th International Workshop on Applied Verification of Continuous and Hybrid Systems (ARCH23)*, vol. 96, 2023, pp. 9–33.

[71] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[72] A. Alanwar, A. Koch, F. Allgower, and K. H. Johansson, "Data-Driven Reachability Analysis From Noisy Data," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 3054–3069, 2023.

[73] S. B. Liu, B. Schürmann, and M. Althoff, "Reachability-based identification, analysis, and control synthesis of robot systems," 2021, arXiv:2103.01626v1.

[74] K. Busawon and H. K. Khalil, "Chapter 9: Digital Implementation," in *High-Gain Observers in Nonlinear Feedback Control*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2017, pp. 279–311.

[75] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, UK: Springer London, 2009.

[76] M. Althoff, "An Introduction to CORA 2015 (Tool Presentation)," in *Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[77] N. Kochdumper, A. Tarraf, M. Rechmal, M. Olbrich, L. Hedrich, and M. Althoff, "Establishing Reachset Conformance for the Formal Analysis of Analog Circuits," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2020, pp. 199–204.

[78] L. Lützow and M. Althoff, "Reachability analysis of armax models," 2023, arXiv:2309.11944.

[79] M. Althoff and J. M. Dolan, "Online Verification of Automated Road Vehicles Using Reachability Analysis," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 903–918, 2014.

[80] C. Tang and M. Althoff, "Formal verification of robotic contact tasks via reachability analysis," 2023, arXiv:2307.13977.

# A Reproduction of Core Publications

## A.1 Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots [1]

**Summary** This paper introduces the concept of reachset conformance for robotic systems to address the need for appropriate modeling techniques in safety-critical applications. Traditional deterministic models will always have a behavior mismatch that can be caused by sensor noise or disturbances. In model-based testing, such mismatches can lead to undetected dangerous behaviors. Instead, we propose models with parameters modeled as uncertain sets, which bound sensor noises and disturbances. These sets must be chosen such that all possible behaviors of the real system are contained within the *reachable sets* of the model; this is also called *reachset conformance*. We enforce reachset conformance by either testing the containment of measured trajectories within the reachable sets of a chosen model or through an identification that optimizes the uncertain parameter sets where the reachset conformance is encoded as a linear inequality. Subsequently, this paper introduces such identification for linear systems, where the unknown uncertainties are modeled as a disturbance of the input and an initial measurement error.

The concept of reachset conformance is demonstrated on the forward dynamics of a six-degrees-of-freedom robot manipulator. The challenge here is that standard forward dynamic models are complex nonlinear differential equations that cannot be explicitly given due to their length. Therefore, we investigate three methods in this work to abstract these equations. The general idea behind this is that we aim for model simplicity (which is suitable for formal analysis) by sacrificing the accuracy of the nominal model. However, as long as the uncertain sets also capture these abstraction errors, reachset conformance is preserved.

A linear model abstraction of the forward dynamics is chosen for the experimental results. Many open-loop tests are carried out on the real robot to gather data for identifying the uncertain sets. Finally, the proposed identification algorithm finds optimal sets that minimize the reachable sets' interval norm while ensuring reachset conformance.

**Author Contributions** M. A. initiated the idea of reachset-conformant models in robotics. **S. L.** developed the methods for abstracting the forward dynamics of robots. **S. L.** developed the method for identifying uncertain sets for linear systems. **S. L.** designed, conducted and evaluated the experiments. **S. L.** wrote the article. M. A. led the research project, provided feedback, and helped improve the manuscript.

**Conference Paper** The accepted version of the conference paper is reprinted in this thesis. The final version of the record is available at `https://doi.org/10.1109/IROS.2018.8593975`.

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

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

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

## I. Introduction

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

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

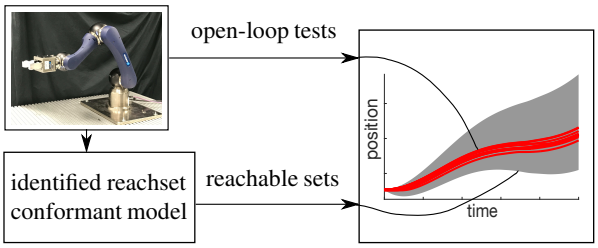


Fig. 1. We identify a reachset conformant model of the Schunk LWA-4P robot, such that the reachable sets enclose all recorded future behaviors of the robot.

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

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

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

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

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

## II. Problem Statement

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

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

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

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

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

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

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

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

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

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

## III. Preliminaries

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

### A. Taylor polynomial arithmetics

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

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

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

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

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

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

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

### B. Interval arithmetics

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

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

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

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

## IV. Forward dynamics modeling

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

### A. Robot model

The standard inverse dynamics model of a robot arm is

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

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

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

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

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

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

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

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

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

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

### B. Abstracting the forward dynamics

We abstract the following forward dynamics:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### C. Identifying the uncertainty sets

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

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

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

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

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

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

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

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

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

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

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

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

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

After introducting

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

TABLE I

SIMULATION ERROR OF MODEL ABSTRACTIONS

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

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

## V. EXPERIMENTAL RESULTS

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

### A. Evaluation of the model abstractions

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

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

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

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

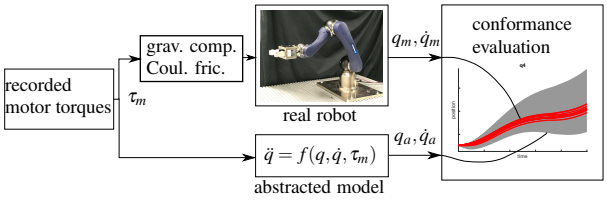


Fig. 2. Open-loop control with gravity and Coulomb friction compensation result in non-deterministic robot behavior (red). The same motor torques influence the abstracted model, which generates the reachable sets (gray).

### B. Conformance testing and optimal uncertainty sets

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

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

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

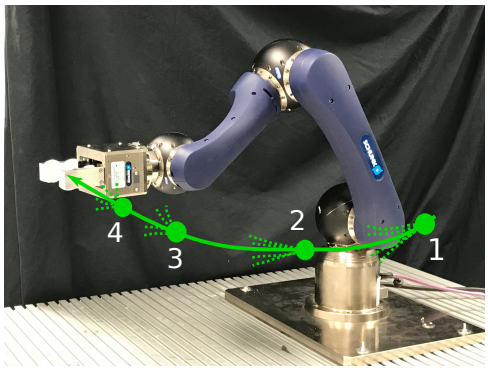


Fig. 3. In each random point-to-point motion the conformance testing starts from four different points indicated by dotted lines, where the controller is switched from closed-loop to open-loop

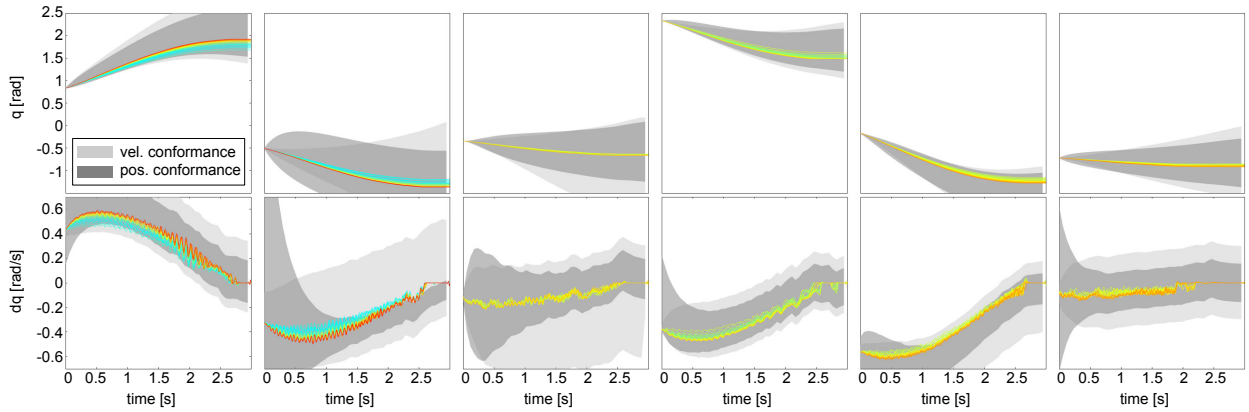


Fig. 4. Reachable set predictions of model 1 for axis 1-6 (left to right) of the Schunk LWA-4P robot. Light gray: both position and velocity are reachset conformant. Dark gray: only the position is reachset conformant. Colored lines are the measured test trajectories. The colors indicate the temperature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

TABLE II
IDENTIFIED OPTIMAL UNCERTAINTY SETS OF MODEL 1 FOR EACH JOINT

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

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

## VI. Conclusions

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

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

## Appendix: Dynamic Parameter Identification

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

## Acknowledgment

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

## References

[1] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Reachset conformance testing of hybrid automata," in *Hybr. Systems: Comp. and Contr.*, 2016, pp. 277–286.
[2] B. Schürmann and M. Althoff, "Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space," in *Proc. 20th IFAC World Congress*, 2017, pp. 12 020–12 027.
[3] A. Pereira and M. Althoff, "Safety control of robots under computed torque control using reachable sets," in *Proc. ICRA*, 2015, pp. 331–338.
[4] M. Milanese and C. Novara, "Set Membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.
[5] M. Kieffer, E. Walter, and I. Simeonov, "Guaranteed nonlinear parameter estimation for continuous-time dynamical models," in *Proc. 14th IFAC World Congress*, vol. 39, no. 1, 2006, pp. 843–848.
[6] N. Ramdani and P. Poignet, "Robust dynamic experimental identification of robots with set membership uncertainty," *IEEE/ASME Trans. Mechatronics*, vol. 10, no. 2, pp. 253–256, 2005.

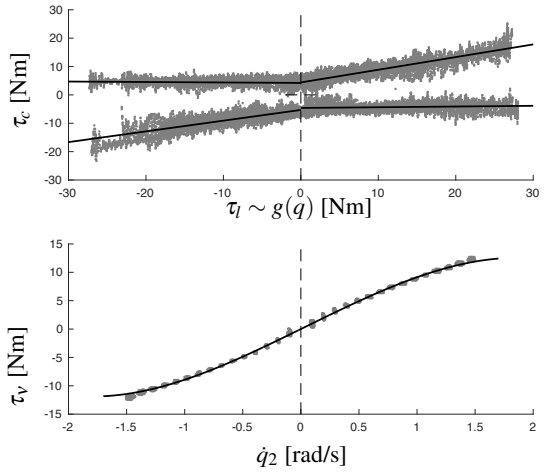


Fig. 5. Fitting results for load-dependent Coulomb friction (up) and viscous friction (down) of joint 2.

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

| **DH parameters** | | | | | | |
|---|---|---|---|---|---|---|
| Par. | Link 1 | Link 2 | Link 3 | Link 4 | Link 5 | Link 6 |
| a | 0 | 0.35 | 0 | 0 | 0 | 0 |
| $\alpha$ | $-\frac{\pi}{2}$ | $\pi$ | $-\frac{\pi}{2}$ | $\frac{\pi}{2}$ | $-\frac{\pi}{2}$ | 0 |
| $\theta$ | $q_1$ | $q_2 - \frac{\pi}{2}$ | $q_3 - \frac{\pi}{2}$ | $q_4$ | $q_5$ | $q_6$ |
| d | 0 | 0 | 0 | 0.3012 | 0 | 0.1548 |
| **Friction parameters** | | | | | | |
| Par. | Joint 1 | Joint 2 | Joint 3 | Joint 4 | Joint 5 | Joint 6 |
| $a_1$ | 6.08 | 4.46 | 6.27 | 6.08 | 1.71 | 2.33 |
| $a_2$ | 0 | 0.38 | 0.25 | 0 | 0.58 | 0 |
| $a_3$ | −6.08 | −5.46 | −6.50 | −6.08 | −1.95 | −2.33 |
| $a_4$ | 0 | 0.32 | 0.25 | 0 | 0.46 | 0 |
| $a_5$ | −6.08 | −4.64 | −5.95 | −6.08 | −2.10 | −2.33 |
| $a_6$ | 0 | 0.041 | −0.029 | 0 | 0.21 | 0 |
| $a_7$ | 6.08 | 4.08 | 6.38 | 6.08 | 1.67 | 2.33 |
| $a_8$ | 0 | −0.0019 | 0.0025 | 0 | −0.081 | 0 |
| $v_1$ | 11.52 | 10.08 | 10.68 | 8.41 | 3.61 | 3.53 |
| $v_2$ | 0.088 | 0.10 | 0.93 | −0.0098 | 0.024 | 0.11 |
| $v_3$ | −0.79 | −1.02 | −1.56 | 0.099 | −0.24 | −0.11 |
| **Gravity model** | | | | | | |
| | | | | | Parameter | Value |
| | | | | | $m_2 + m_3 + m_4 + m_5 + m_6 + 2.8571c_{x2}m_2$ | 6.36 |
| | | | | | $0.3012(m_4 + m_5 + m_6) + c_{y4}m_4 + c_{z3}m_3$ | 1.04 |
| | | | | | $c_{z4}m_4 - c_{y5}m_5$ | −0.0661 |
| | | | | | $0.1548m_6 + c_{z5} * m_5 + c_{z6}m_6$ | 0.189 |
| **Inertial parameters** | | | | | | |
| | | | | | Parameter | Value |
| | | | | | $I_{1yy} + I_{2yy} - I_{2zz} + I_{3zz} + k_{r2}^2 I_{m1} - k_{r2}^2 I_{m2}$ | −0.0162 |
| | | | | | $m_2 - 8.163(k_{r2}^2 I_{m2} - I_{2zz}) + m_3 + m_4 + m_5 + m_6$ | −7.45 |
| | | | | | $0.301(m_4 + m_5 + m_6) + c_{y4}m_4 + c_{z3}m_3$ | 0.94 |
| | | | | | $I_{3xx} - I_{3zz} + I_{4zz} + 0.0907(m_4 + m_5 + m_6) + 0.602c_{y4}m_4$ | 0.171 |
| | | | | | $I_{3yy} + I_{4zz} + 0.0907(m_4 + m_5 + m_6) + 0.602c_{y4}m_4$ | 0.371 |
| | | | | | $I_{5xx} + I_{6yy} - I_{5zz} + 0.024m_6 + 0.31c_{z6}m_6$ | 0.0555 |
| | | | | | $I_{5yy} + I_{6yy} + 0.024m_6 + 0.31c_{z6}m_6$ | 0.126 |
| | | | | | $2.857(I_{2zz} + k_{r2}^2 I_{m2}) + c_{x2}m_2$ | 5.02 |
| | | | Parameter | Value | Par. | Val. |
| | | | $0.155m_6 + c_{z5}m_5 + c_{z6}m_6$ | 0.174 | $k_{r3}^2 I_{m3}$ | 1.78 |
| | | | $c_{z2}m_2 - c_{y3}m_3 - 2.857I_{2xz}$ | −0.183 | $I_{4yz}$ | 0.0726 |
| | | | $I_{2xx} - I_{2yy} + I_{2zz} + k_{r2}^2 I_{m2}$ | 1.64 | $k_{r4}^2 I_{m4}$ | 1.63 |
| | | | $c_{z4}m_4 - c_{y5}m_5$ | −0.0388 | $I_{5yz}$ | 0.0352 |
| | | | $I_{4xx} - I_{4zz} + I_{5zz}$ | 0.135 | $k_{r5}^2 I_{m5}$ | 0.541 |
| | | | $I_{4yy} + I_{5zz}$ | 0.0269 | $I_{6zz}$ | 0.00569 |
| | | | $I_{6xx} - I_{6yy}$ | −0.0352 | $k_{r6}^2 I_{m6}$ | 0.637 |
| | | | $I_{3yz}$ | 0.106 | | |


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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## A.2 Provably Safe Motion of Mobile Robots in Human Environments [2]

**Summary** The safety problem of this paper is to ensure that a mobile robot can never collide with humans when the robot is moving. At the same time, current approaches in the industry (e.g., ISO 13482 [19]) suffer from conservative human motion models, which frequently lead to a *freezing robot problem*. To increase the efficiency of robot motion while maintaining human safety, we propose to apply *online verification*. This algorithm continuously analyses the robot motion planning, generates fail-safe trajectories, and verifies the safety of the human-robot interaction for the robot input, including the fail-safe trajectory. If verification fails, i.e., the robot input leads to an unsafe collision, the previously verified fail-safe trajectory is selected to transition the robot into a safe state.

We choose verification models for humans and mobile robots on a 2D plane and compute their reachable set in position space. We assume uncertainties such as their possible maximum acceleration, velocity, and sensor measurement errors for the human motion model and tune them manually to minimize the reachable set. Reachset conformance of the human model is shown through a publicly available video dataset, where the trajectories of the humans on pedestrian walkways are checked against computed reachable sets.

The algorithm is tested in a pedestrian simulator, where the robot is commanded to move either with, across, or against a flow of pedestrians. Using online verification, we have determined that the robot's goal position can be reached between 1.4 and 3.5 times faster than the ISO 13482 methods, while safety is never violated. Therefore, we have shown the effectiveness of formal methods in mobile robotics and their potential improvements in task efficiency that such approaches can bring to the application through improved modeling and verification.

**Author Contributions** **S. L.** developed the main algorithm (Alg. 1) and the models required for verification. **S. L.** and H. R. performed the reachset conformance checking for pedestrians. **S. L.**, C. H., and I. L. designed, conducted, and evaluated the experiments. **S. L.**, H. R., C. H., I. L., and J. O. wrote the article. J. O. and M. A. led the research project. M. A. drafted the initial idea, provided feedback, and helped improve the manuscript.

**Conference Paper** The accepted version of the conference paper is reprinted in this thesis. The final version of the record is available at `https://doi.org/10.1109/IROS.2017.8202313`.

# Provably Safe Motion of Mobile Robots in Human Environments

Stefan B. Liu*†, Hendrik Roehm†, Christian Heinzemann†, Ingo Lütkebohle†,
Jens Oehlerking†, and Matthias Althoff*
*Department of Informatics, Technische Universität München, Germany, Email: [stefan.liu | althoff]@tum.de
†Robert Bosch GmbH, Corporate Research, Renningen, Germany, Email:[firstname.lastname]@de.bosch.com

***Abstract*— Mobile robots operating in a shared environment with pedestrians are required to move provably safe to avoid harming pedestrians. Current approaches like safety fields use conservative obstacle models for guaranteeing safety, which leads to degraded performance in populated environments. In this paper, we introduce an online verification approach that uses information about the current pedestrian velocities to compute possible occupancies based on a kinematic model of pedestrian motion. We demonstrate that our method reduces the need for stopping while retaining safety guarantees, and thus goals are reached between 1.4 and 3.5 times faster than the standard ROS navigation stack in the tested scenarios.**

## I. Introduction

Mobile service robots often need to operate freely and flexibly in environments occupied by pedestrians. Because collisions could cause serious harm, particularly in settings with heavier robots, safety mechanisms always have to be considered. Despite the large body of work in path planning and obstacle avoidance [1], [2], in practice, most production robots still rely on hardware safety devices such as certified laser scanners. The main reason is that demonstrating the safety of software to the satisfaction of a safety body is difficult, and difficulty scales with the complexity of the algorithm.

In environments with none or only a few humans, a common way to reduce the problem is through a simple model of human motion: Either assume people will always stop (ISO 3691-4 [3]), or assume they always move at full speed (ISO 13855 [4] and ISO 13482 [5]). The latter is usually applied and results in a circular safety area as illustrated in the left part of Fig. 1.

Unfortunately, a circular field seriously restricts robot motion, including in areas which common sense would indicate as usable, such as beside or following a walking pedestrian. In more populated environments, it leads to frequent stopping of the robot and is therefore almost unusable.

This paper, in contrast, proposes a safeguard that predicts all possible motions based on a kinematically-accurate model of human motion, as well as the humans' current position and velocity. It guarantees the same level of safety but allows much more efficient motion.

Specifically, we compute so-called *reachable sets* that include all possible future occupancies of pedestrians and the robot based on their kinematic models. Based on these sets, we consider a velocity command as verified safe if the robot can stop before entering the reachable set of any pedestrian, i.e., no collision can occur before the robot stops (passive safety [6]). From the example reachable sets shown on the right side of Fig. 1 it is immediately obvious that they leave much more maneuvering space compared to the static approach on the left.

While our current kinematic model considers walking pedestrians only, our approach is extensible to multiple models, e.g., to other dynamic behaviors (running, wheelchairs, etc.) as well as to structured environments. For instance, reachability analysis has also already been used in other safety-related applications such as autonomous driving [7] and robot manipulators [8].

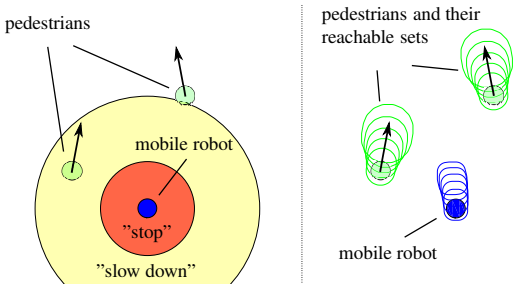


Fig. 1: Standard safety field (left) compared to our method.

We evaluate the performance of our approach regarding safety and efficiency in a Gazebo-based simulation environment which combines the standard ROS navigation stack[1] with a safeguard based on either i) the one proposed in ISO 13482, ii) an obstacle model of the *braking ICS* approach [9], or iii) our method. We use three different scenarios and two pedestrian densities. Here, our approach improves the average velocity by a factor of 1.4 – 3.5. In absolute numbers, the average velocities for an easy situation are 1 m/s (our approach) compared to 0.64 m/s (ISO 13482), or, more striking for a difficult situation, 0.92 m/s vs 0.27 m/s.

In the following, we first outline the main ideas of reachability analysis, passive safety, and reachable set conformance in Sect. II. In Sect. III, we describe the modeling and online verification approach. Sect. IV shows the evaluation results. Works related to our approach are described in Sect. V.

## II. Preliminaries / Definition of Terms

In this section we briefly introduce the main methods and terms relevant to our approach.

[1] `http://www.ros.org/navigation`

### A. Reachability analysis of continuous systems

We model a system as a differential equation $\dot{x}(t) = f(x(t),u(t))$, where $t$, $x$ and $u$ are time, state, and input respectively. The initial state $x(0)$ can be chosen arbitrarily within the initial set $\mathscr{X}^0$. The time-dependent input trajectory $u(t)$ is allowed to vary, but is assumed to stay in a time-dependent input trajectory set $\mathscr{U}(t)$. These two sources of non-determinism are used to model the fact that we do not exactly know the current positions and velocities of objects and their behavior in the future, see Sect. III-A.

Reachability analysis is the computation of reachable sets of states over time of such a model. For instance, the reachable sets for a robot and two pedestrians for different points in time are illustrated in Fig. 1. Reachable sets are formally defined as follows:

**Definition 1** (Reachable Set (Reachset), see [7])**.** Given an initial set $\mathscr{X}^0$ and a time-dependent input trajectory set $\mathscr{U}(t)$, the reachable set $\mathscr{R}(t)$ at time $t$ of a system of the form $\dot{x}(t) = f(x(t),u(t))$ is the set of all reachable states at time $t$:

$$\mathscr{R}(t) = \Big\{ x(t) = \int_0^t f(x(\tau),u(\tau))d\tau + x(0) \Big| \; x(0) \in \mathscr{X}^0, \forall \tau \in [0,t] : u(\tau) \in \mathscr{U}(\tau) \Big\}. \quad (1)$$

We are using CORA for efficient computation of reachable sets for high dimensional and nonlinear problems. Although CORA itself has yet to be proved formally in a theorem prover such as in [10], the method proposed in this work can be proven similarly.

For safety analysis it is very important to account for system uncertainties. Since we do this by using the non-determinism of the model, $\mathscr{R}(t)$ is a set containing *all* possible future states of the system at a time $t$. We verify safety by checking that there is never an intersection between reachable sets $\mathscr{R}(t)$ of our system and sets of unsafe states (e.g. position of surrounding objects, unsafe velocities, etc.).

### B. Passive Safety & Safe Motion Trajectory

Passive safety means that no collisions with surrounding pedestrians are allowed to happen when the robot moves [11]. This is equivalent to ensuring a complete stop before a potential collision. We therefore choose an input trajectory that brings the robot to a stop. The trajectory is considered safe if the following property holds:

**Definition 2** (Safe Motion Trajectory)**.** An input trajectory $u(t)$ of a robot system that brings the robot to a safe stop at $t_{stop}$ is safe according to passive safety, if

$$\forall t \in [0; t_{stop}] : \mathscr{R}_{ped}(t) \cap \mathscr{R}_{rob}(t) = \emptyset, \quad (2)$$

where $\mathscr{R}_{ped}(t)$ are pedestrian reachable sets and $\mathscr{R}_{rob}(t)$ are robot reachable sets with input trajectory $u(t)$.

### C. Model Conformance

Validating that our pedestrian model conforms to real pedestrian behavior is very important for the overall verification technique. For our model-based results to hold in reality, our model of the pedestrian has to conform to real behavior. We therefore check our pedestrian model against real measured data before using it for verification.

Recently, we showed that for verifying the absence of collisions, *reachset conformance testing* is a suitable approach to check the conformance of models to real behavior [12].

**Definition 3** (Reachset Conformance)**.** Measured state data $p_1,\ldots,p_n$ of a pedestrian with timestamps $t_1,\ldots,t_n$ is reachset conformant to the pedestrian model, if the following holds:

$$\forall i : p_i \in \mathscr{R}_{ped}(t_i). \quad (3)$$

We evaluate reachset conformance in Sect. IV-A based on the pedestrian model introduced in the next section.

## III. Modelling and Verification

In this section, we provide the models for the pedestrians and the robot that we employ for the reachable set computations and give an algorithm for the online computation of safe motion trajectories based on reachable sets.

### A. Pedestrian Modeling

We model a single pedestrian as a point on a two-dimensional plane. The shape of the pedestrian is then taken into account after the reachable set computation by enlarging the reachable sets accordingly. We assume that we can measure the pedestrian's position and velocity with some known uncertainty. Also, we assume that the pedestrian performs a forward walking motion while possibly changing directions and that the pedestrian has a maximum speed and acceleration. We represent these constraints as two separate differential equation models: one constraining the acceleration and one constraining the velocity. Reachable states of the pedestrian are then states which are reachable under both models.

It would be possible to merge these two models into one that includes state constraints. This can be realized in CORA by a hybrid model, for which reachable sets are difficult to obtain. However, it has been shown in [13] that for reachability analysis it is possible to define multiple abstracting models, such that their reachable set intersection overapproximates the reachable sets of the hybrid model.

Therefore, we define the following two models. The acceleration-constrained model

$$\begin{gathered} \dot{p}_x = v_x, \qquad \dot{p}_y = v_y, \qquad \dot{v}_x = a_x, \qquad \dot{v}_y = a_y \\ \mathscr{U}_{ped}^{(a)} = \left\{ (a_x,a_y) \in \mathbb{R}\times\mathbb{R} | a_x^2 + a_y^2 \leq a_{max}^2 \right\} \end{gathered} \quad (4)$$

has the two-dimensional position $p$ and velocity $v$ as its state variables. The input trajectory is a time-invariant set representing all possible two-dimensional accelerations and bounded by $a_{max}$. The velocity-constrained model

$$\begin{gathered} \dot{p}_x = v_x, \qquad \dot{p}_y = v_y \\ \mathscr{U}_{ped}^{(v)} = \left\{ (v_x,v_y) \in \mathbb{R}\times\mathbb{R} | v_x^2 + v_y^2 \leq v_{max}^2 \right\} \end{gathered} \quad (5)$$

has only the two-dimensional position $p$ as its state variables, while the velocity $v$ is instead an input bounded by $v_{max}$.

The initial position $[p_x(0), p_y(0)]$ and initial velocity $[v_x(0), v_y(0)]$ are assumed to lie in the sets $\mathscr{X}_{ped}^{(a),0}$ and $\mathscr{X}_{ped}^{(v),0}$, respectively. Since both models are used to predict possible pedestrian behavior, their initial states can be interpreted as the currently measured position and velocity of the pedestrian, plus some assumed measurement uncertainty.

The reachable sets of a single pedestrian are obtained by computing the reachable sets $\mathscr{R}_{ped}^{(a)}(t)$ and $\mathscr{R}_{ped}^{(v)}(t)$ (Fig. 2) of both models and then taking their intersection $\mathscr{R}_{ped}(t) = \mathscr{R}_{ped}^{(a)}(t) \cap \mathscr{R}_{ped}^{(v)}(t)$. Lastly, we enlarge all $\mathscr{R}_{ped}(t)$ by a circle in the $(p_x, p_y)$-dimensions to account for the shape of the human; any other shape can also be used.

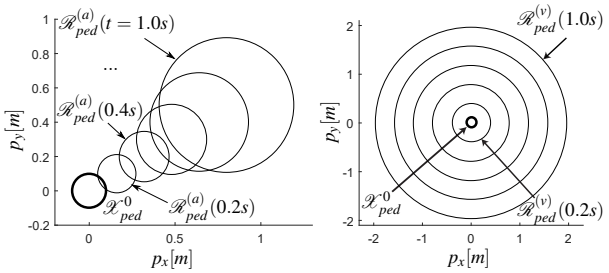


Fig. 2: Reachable sets according to the acceleration-constrained (left) and velocity-constrained (right) model.

### B. Robot Modeling

For modeling the mobile robot we use a kinematic model of a differential-drive robot

$$\dot{p}_x = v_{tra}\cos(\phi), \qquad \dot{p}_y = v_{tra}\sin(\phi), \qquad \dot{\phi} = v_{rot}.$$

The initial state $[p_x(0), p_y(0), \phi(0)]^T$ represents the current pose of the robot and is bounded by an initial set $\mathscr{X}_{rob}^{0}$ that accounts for the inaccuracy of the robot's localization algorithm. The input of the system is the vector $[v_{tra}, v_{rot}]^T$, consisting of the translational and rotational velocities of the differential drive. For verification we consider that the input is not allowed to change at a larger rate than the maximum acceleration of the robot. In the same fashion as for the reachable sets of the pedestrians, we add the shape of the robot to the $(p_x, p_y)$-dimensions of all $\mathscr{R}_{rob}(t)$.

### C. Online Motion Trajectory Verification

In our approach, we verify passive safety (Sect. II-B) of motion commands for every step $k$, where the sampling time is $\Delta t$. To this end, we employ reachability analysis to predict whether the robot can still come to a collision-free stop after applying the input $u_{plan}^{(k)} = [v_{tra}^{(k)}, v_{rot}^{(k)}]^T$ to the robot motors for a time step. For that we define a candidate input trajectory $u^*(t)$: It begins with the planned motion command $u^*(t) = u_{plan}$ for $t \in [0, \Delta t]$ and continues with $u^*(t) = u_{brk}(t)$ for $t \geq \Delta t$. The braking trajectory $u_{brk}(t)$ brings the robot to a stop $u(t_{stop}) = [0,0]^T$ and the slope is the maximum deceleration of the robot system (see Fig. 3). Note that $t = 0$ always corresponds to the current time step in this analysis, while signal values and reachable sets at $t > 0$ correspond to (predictions of) future behavior.

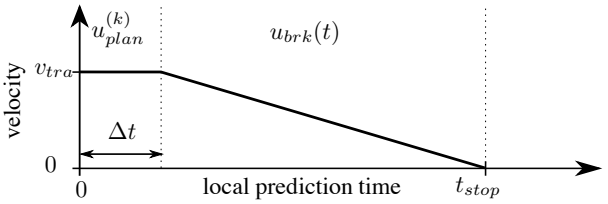


Fig. 3: Candidate input trajectory $u^*(t)$ that we use to compute $\mathscr{R}_{rob}(t)$, here shown for $v_{tra}$. $v_{rot}$ is analogous.

We then compute pedestrian and robot reachable sets and verify $u^*(t)$ by checking whether $u^*(t)$ satisfies the property in Def. 2. If $u^*(t)$ is verified, we store $u_{safe}^{(k)}(t) := u^*(t)$ as a safe input trajectory and apply $u_{safe}^{(k)}(t)$ to the motors for the next time step $\Delta t$.

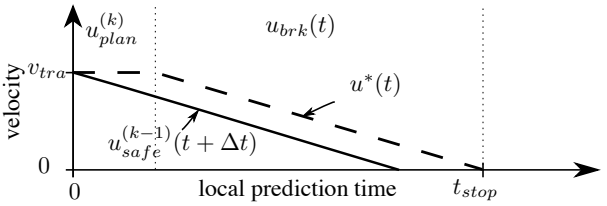


Fig. 4: Continued use of previously verified safe trajectory $u_{safe}^{(k-1)}(t)$, shifted by $\Delta t$, if $u^*(t)$ is not safe

If $u^*(t)$ is not successfully verified, we simply execute the previously verified safe input trajectory $u_{safe}^{(k)}(t) := u_{safe}^{(k-1)}(t + \Delta t)$ (see Fig. 4), which is shifted by $\Delta t$. Thus, we ensure that only inputs verified as safe are applied to the motors. The robot does not necessarily approach zero velocity if during the braking maneuver a new motion command is verified as safe and applied to the motors. This approach is implemented as a safety control module (Alg. 1) between the planning module and the robot motor control (see Fig. 5). We initially set $u_{safe}^{(k-1)}(t) := [0,0]^T$.

Depending on the hardware, robot control often encounters a system delay. We can easily account for that by extending $u^*(t)$ with all the delayed motion commands in $[-t_{delay}; 0]$.

## IV. Evaluation

In the following, we present the results of evaluating our pedestrian model (Sect. IV-A), and we present results on the performance of our online verification approach (Sect. IV-B).

### A. Model Evaluation

We check whether the pedestrian model of Sect. III-A overapproximates the real behavior of a walking-only human by performing a reachset conformance test (Def. 3) using ground truth pedestrian trajectories from a labeled video source of a street scene in Zurich, Switzerland [14]. We check if the trajectories lie inside the computed pedestrian reachable sets. This test was performed for a time horizon

**Algorithm 1** Online safety control (step $k$, sampl. time $\Delta t$)

**Input:** $u_{safe}^{(k-1)}(t), u_{plan}^{(k)}, \hat{\mathcal{X}}_{rob}^{0}, \hat{\mathcal{X}}_{ped,1..n}^{0}$ for $n$ pedestrians
**Output:** $u_{safe}^{(k)}(t)$

1: Set $u^*(t) := \begin{cases} u_{plan}^{(k)} & 0 \le t < \Delta t \\ u_{brk}(t) & \Delta t \le t \le t_{stop} \\ 0 & t_{stop} < t \end{cases}$ (see Fig. 3)
2: Compute $\mathcal{R}_{rob}(t)$ with $\hat{\mathcal{X}}_{rob}^{0}$, $u^*(t)$ for $t \in [0; t_{stop}]$
3: Compute $\mathcal{R}_{ped,i}(t)$ with $\hat{\mathcal{X}}_{ped,i}^{0}$ for $t \in [0; t_{stop}]$ for all $i = 1..n$ pedestrians
4: **if** $\forall i : \mathcal{R}_{ped,i}(t) \cap \mathcal{R}_{rob}(t) = \emptyset$ for all $t \in [0; t_{stop}]$ **then**
5: Set $u_{safe}^{(k)}(t) := u^*(t)$ for all $t \ge 0$
6: **else**
7: Set $u_{safe}^{(k)}(t) := u_{safe}^{(k-1)}(t + \Delta t)$ for all $t \ge 0$
8: **end if**

TABLE I: Pedestrian model and conformance test results

| Pedestrian Model | | Conformance Test | |
|---|---|---|---|
| Time horizon | 1.6 s | Pedestrians | 420 |
| $a_{max}$ | 0.6 m/s$^2$ | Gener. test cases | 20084 |
| $v_{max}$ | 2.0 m/s | Passed tests | 19843 |
| Ped. diameter | 0.54 m | Rate | 98.80 % |
| $\hat{\mathcal{X}}_{ped}^{0}$: $(p_x, p_y)$-uncertain. | $\pm$ 0.1 m | | |
| $\hat{\mathcal{X}}_{ped}^{0}$: $(v_x, v_y)$-uncertain. | $\pm$ 0.1 m/s | | |

of 1.6 s, which is larger than the largest $t_{stop}$ of the robot in our evaluation (Sect. IV-B).

For the pedestrian model, we parameterize $v_{max} = 2.0$ m/s as suggested by [4], because it is the transition speed between walking and running. To set $a_{max}$, we apply numerical differentiation and filtering on the velocity data of the video source and then set $a_{max} = 0.6\,\text{m/s}^2$ as an overapproximative value. The parameters of our model are shown in Tab. I.

The results (Tab. I) show good reachset conformance results. However, there are some unsuccessful tests. A closer look at these failed cases reveals that the unsuccessful tests are caused by special pedestrian behavior lying outside of our initial assumptions such as changing directions too fast (12 cases), and velocities faster than $v_{max}$ (229 cases).

This conformance test shows that our pedestrian model is reachset conformant to walking-only pedestrians which do not change their direction of movement very abrupt. This pedestrian model can therefore be used for our verification approach if we are able to constrain human behavior to walking-only and slow-direction-changing, which is possible in a closed environment setting, as in production plants. However, our model is not reachset conformant to all pedestrian behaviors and there are two possible solutions. First, we could increase the bounds $a_{max}$ and $v_{max}$ leading to a richer set of behaviors and thus, to bigger reachable sets. Second, one may consider hybrid models switching to more conservative models, as suggested in [13], once the special cases above are detected. In addition, runtime monitors as proposed by ModelPlex [15] could be used to continuously validate the correctness of the used model at runtime.

### B. System Evaluation

We evaluate the performance of our online verification in a ROS simulation for different scenarios where the robot has to navigate in the presence of pedestrians.

*Considered Approaches:* We compare three approaches with different obstacle models. First, we use our approach introduced in the previous sections. Second, we consider an ISO13482-compliant safety field [5] with 360° warning and protective fields. The size of the safety field is fixed and dimensioned based on the maximum speed of the robot and the assumption that a pedestrian may approach the robot at full speed at any point in time. In contrast, the size of the reachable sets in our approach is dynamic and depends on the current velocity of the robot and pedestrians. The third approach is based on the obstacle model used in braking ICS [9] and by Mitsch et al. [11]. This obstacle model assumes that obstacles may always move at full speed in any direction if we do not know their future behavior and requires that the robot is able to come to a rest before the obstacle may hit it. We refer to this approach as braking ICS in the following. In contrast, our approach computes reachable sets based on current velocity and direction of movement.

*Experiment Setup:* We execute our evaluation based on ROS Indigo. The physics simulation is carried out in Gazebo 7[2] and the robot uses a standard move base based on the Dynamic Window Approach (DWA, [16]). We use the default parameters from the Indigo release for the move base, except that we set the maximum velocity and acceleration for the differential drive robot to $v_{tra} = 1.5$ m/s, $v_{rot} = 2.0$ rad/s, $a_{tra} = 1.5\,\text{m/s}^2$, and $a_{rot} = 1.0\,\text{rad/s}^2$. Initially, the robot is stationary. The robot model is based on the Robotino models for Gazebo by RWTH Aachen[3], where we use its laser scanner for navigation and use the standard Planar Move Plugin to steer the robot.

The setup of our ROS system is shown in Fig. 5. The pedestrian simulation (upper left box) computes the pedestrian positions and velocities that are then sent to Gazebo (upper right box) and to the online safety control. The robot simulation in Gazebo (left box) simulates the robot actuators and provides laser scan data for localization in our map. The localization is performed in amcl, and information on robot pose and static obstacles contained in the laser scans are used by the ROS move base for computing a path to the goal position (lower right boxes). In addition, the move base contains the DWA implementation that generates the velocity commands for the robot. These velocity commands are then fed into the online safety control node (middle right box) and only forwarded to the robot actuators in the Gazebo Robot Simulation if they are safe. In our experiments, the whole setup of the ROS system remains unchanged except that we change the safety checker inside the online safety control.

For improving the efficiency of our approach and to enable real-time performance, we created a library of reachable sets for the pedestrians at design time that we store in a look-

[2] http://gazebosim.org/
[3] https://git.fawkesrobotics.org/gazebo-models.git

up table (14 MB). Since the pedestrian model equations are independent from the initial position, we only need to sample based on the initial velocity, which we do in steps of 0.1 m/s.

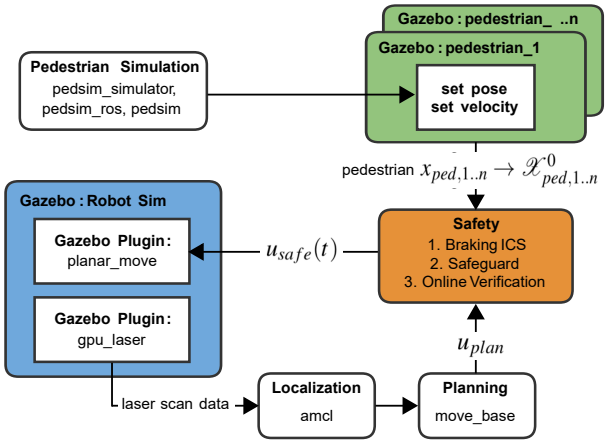


Fig. 5: Setup of the simulation environment in ROS.

The evaluations are carried out on a map that is illustrated in Fig. 6. The map is 24 m times 30 m from wall to wall and pedestrians walk continuously counter-clockwise along the green area. For obtaining realistic pedestrian motion, we simulate pedestrian motion in a dedicated Pedestrian Simulator[4] (PedSim) that is based on social forces. The simulated pedestrian positions are then transferred to Gazebo, while the robot position is also considered in PedSim such that the pedestrians react to the robot.

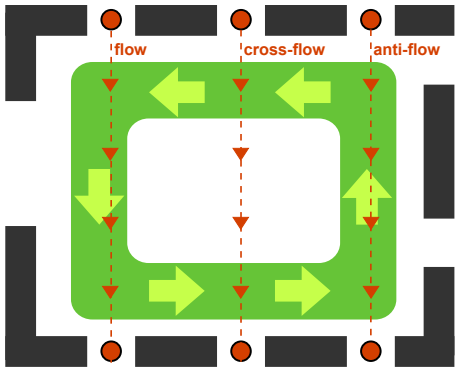


Fig. 6: Illustration of the map used for the experiments.

In our experiments, the robot will always move from top to bottom through the green area with different starting positions. Depending on the starting position, we create three scenarios for encountering pedestrians: flow (move in same direction as pedestrians), cross-flow (pedestrians coming from left or right), and anti-flow (pedestrians approaching from front). In addition, we consider two pedestrian densities: light population and dense population. For light population, we place 25 pedestrians uniformly at random

[4] https://github.com/srl-freiburg/pedsim_ros

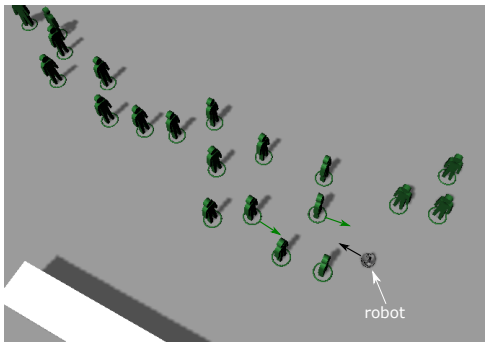


Fig. 7: Gazebo screenshot for a dense antiflow scenario.

in the green area; for dense population, we distribute 60 pedestrians. We create 10 different placements of pedestrians for each density with a minimum distance of 0.5 m between any two pedestrians. Fig. 7 shows a Gazebo screenshot of a typical situation in anti-flow scenarios with dense population.

*Assumptions:* The current pose and velocity of each pedestrian, which we require as inputs to our online verification, would need to be provided by a people-tracking approach on a real robot. In simulation, we instead take this information directly from PedSim. This therefore represents the best case, where we can track all pedestrians perfectly and exactly know their current position and velocity. In order to account for the imprecision of current perception and tracking approaches, we add an uncertainty of 0.1 m to the pedestrian positions and 0.1 m/s to the pedestrian velocities to make our simulation more realistic. Finally, we have also included the actual control delay of 100 ms that the real hardware exhibits.

*Experimental Execution*: For each scenario (flow, cross-flow, anti-flow), and for both light and dense populations, we generate 10 different pedestrian placements. All three approaches are executed on all of the situations. Based on the collected data, we compute (1) whether the goal has been reached, (2) how long it took to reach the goal, (3) the distance traveled by the robot, (4) the average velocity, and (5) the number of unsafe collisions. An unsafe collision is one in which the robot's velocity is greater than 0.

*Results:* The results are summarized in Table II for lightly populated situations and in Table III for the densely populated ones. Throughout our simulation runs, no unsafe collisions occurred for any of the approaches, so we omitted the corresponding column in the result tables. All values are arithmetic means over all runs.

The results clearly show that our method performs best in all cases by a large margin. Even in the simplest situation, motion with a lightly populated flow, our method is 1.4 times faster, and in the dense situation this even increases to a factor of 3.5. The example in Fig. 8 illustrates how the robot is able to follow a group of pedestrians in a flow scenario with light population when applying our online verification approach. It is also notable that both safety field and Braking ICS exhibit very bad performance in the anti-flow situation. This is despite the fact that our pedestrian

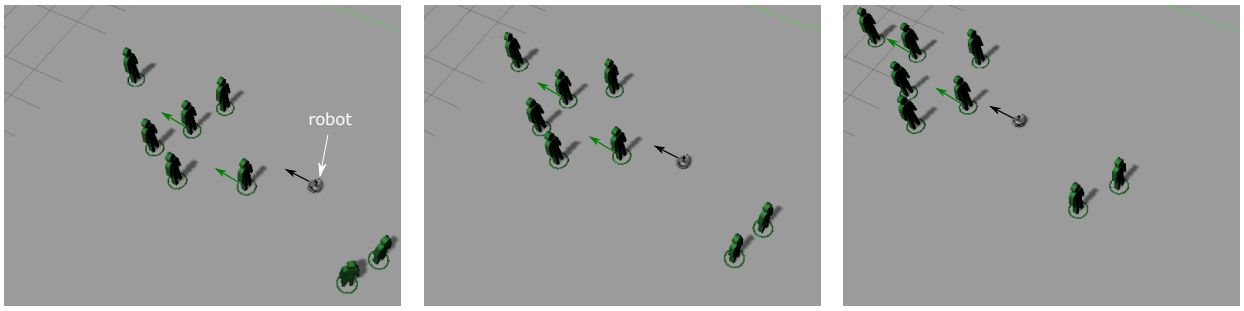


Fig. 8: Gazebo screenshots from the same position with a time step of approx. 1 s for a light flow scenario where the robot uses the online verification approach.

TABLE II: Results from ROS Simulation (Lightly Populated Scenarios)

| | Flow | | | | Cross-flow | | | | Anti-flow | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Approach** | @Goal | Time(s) | Len(m) | Vel(m/s) | @Goal | Time(s) | Len(m) | Vel(m/s) | @Goal | Time(s) | Len(m) | Vel(m/s) |
| Braking ICS | 10 | 34.3 | 22.9 | 0.73 | 10 | 40.0 | 23 | 0.63 | 10 | 116.1 | 23.0 | 0.21 |
| Safety Field | 10 | 37.9 | 22.9 | 0.64 | 10 | 35.7 | 23 | 0.68 | 10 | 74.8 | 22.9 | 0.32 |
| Onl. Verif. | 10 | 22.4 | 22.9 | 1.04 | 10 | 26.3 | 23.2 | 0.91 | 10 | 52.3 | 23.0 | 0.45 |

TABLE III: Results from ROS Simulation (Densely Populated Scenarios)

| | Flow | | | | Cross-flow | | | | Anti-flow | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Approach** | @Goal | Time(s) | Len(m) | Vel(m/s) | @Goal | Time(s) | Len(m) | Vel(m/s) | @Goal | Time(s) | Len(m) | Vel(m/s) |
| Braking ICS | 10 | 108.0 | 22.9 | 0.25 | 10 | 114.2 | 23.1 | 0.21 | 10 | 519.5 | 23.2 | 0.05 |
| Safety Field | 10 | 96.0 | 22.9 | 0.27 | 10 | 76.9 | 23 | 0.31 | 10 | 251.5 | 23.0 | 0.10 |
| Onl. Verif. | 10 | 26.0 | 23 | 0.92 | 10 | 37.8 | 23.1 | 0.65 | 10 | 159.2 | 23.2 | 0.15 |

simulator is cooperative, i.e. humans attempt to actively avoid the robot, and that the robot also uses a normal obstacle avoidance algorithm (albeit one that makes a static obstacle assumption). To this end, the effect of the obstacle avoidance seems to be minimal because the path lengths traveled by the robot are nearly the same for the three approaches in all considered situations. That means the robot takes almost the same path in all situations and only adjusts its speed instead of going round the populated areas, which we attribute to the static obstacle assumption.

For dense population, our online verification method provides significant improvements in average velocity for the flow and cross-flow scenarios. For the anti-flow scenario, the online verification still enables an average velocity that is a factor 2 (safety field) or 3 (braking ICS) higher compared to the other approaches, but an absolute average velocity of 0.15 m/s still leaves significant room for improvement.

Last, but not least, it might be surprising that the Braking ICS approach often performs worse than the ISO 13482 safety field. We surmise that this is because the safety field includes a warning field, which just reduces speed but still enables to robot to move. In contrast, the Braking ICS approach always stops the robot when it detects a potential collision, which essentially corresponds to a having safety field without a warning field.

## V. Related Work

We discuss related work that aims at establishing provably safe motion of mobile robots with respect to a mathematical model considering moving obstacles, particularly humans.

*Full Obstacle Knowledge:* The first approaches in this regard are inevitable collision states (ICS, [17]), non-linear velocity obstacles [18], and the FD* path planner [19]. All of these approaches, however, make the assumption of exactly knowing the future behavior of all obstacles during the planning horizon, which is unrealistic for pedestrians.

*Conservative Obstacle Model:* Braking ICS [9] use a conservative obstacle model where an obstacle may always move with maximum speed in any direction. This obstacle model corresponds to the relevant safety norms [5], [4]. Similar to our approach, braking ICS append to each verified trajectory a braking trajectory for proving passive safety [6]. In contrast to our approach, they compute and check several possible braking trajectories for the robot. Mitsch et al. [11] and Zhang et al. [20] use theorem proving for showing that the DWA enables passively safe motion for differential drive robots using the same obstacle model as braking ICS. Likewise, Dabadie et al. [21] assume that obstacles behave in the worst possible way while proving collision-free motion based on a reach-avoid problem formulation. Aniculaesi et al. [22] construct an observer monitor that considers a fixed braking distance and obstacle velocity for constructing a safety circle around the robot. Similar to a safeguard, the monitor triggers a safe braking maneuver if an obstacle enters the safety circle. As indicated by our evaluation results, such models with a fixed maximum velocity are conservative and lead to decreased performance in populated environments.

*Motion Primitives:* The approaches by Hess et al. [23] and Majumdar et al. [24] build a library of motion primitives from which they construct motion plans. These motion plans

include occupancies of the robot along the trajectory and may be verified against occupancies of obstacles. Such approaches could be a useful extension to our approach if the online computation of robot reachsets is too resource demanding.

*Occluded Obstacles:* The approaches by Alami et al. [25] and Chung et al. [26] consider (partially) occluded obstacles, e.g., if humans appear from a crossing corridor. Obstacles are assumed to appear with maximum speed at any time. For these cases, our model cannot be applied and the use of such conservative models is necessary.

*Probabilistic Approaches:* Probabilistic approaches like probabilistic ICS [27] and probabilistic collision states [28], [29] accept a small probability of collision, which is not acceptable in settings with heavier robots such as intralogistics.

## VI. Conclusions

We present a safety approach for mobile robots that guarantees passive safety regarding walking pedestrians by an online verification using reachability analysis. Based on models of pedestrians and the mobile robot, we compute their reachable future occupancy at every timestep to determine whether a braking trajectory leads to a safe stop. In our evaluation, we demonstrate the validity of the pedestrian model using reachset conformance testing and discuss possible improvements. The evaluation of the online verification in a ROS simulation shows that our approach enables significantly improved navigation performance through crowds compared to standard safety approaches.

Future work focuses on real world applicability of our approach. Today's people-tracking algorithms still lack accuracy and reliability especially in velocity estimation, which needs to be the major focus. The effect of sensor occlusion on our approach is also of interest. Furthermore, additional modelling effort is needed, e.g., to extend our pedestrian model or to consider other dynamical objects in the environment.

## Acknowledgment

The authors gratefully acknowledge financial support by the European Commission project UnCoVerCPS under grant number 643921.

## References

[1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726 – 1743, 2013.

[2] J. Minguez, F. Lamiraux, and J.-P. Laumond, "Motion planning and obstacle avoidance," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2016, pp. 1177–1202.

[3] *Industrial trucks - Safety requirements and verification - Part 4: Driverless industrial trucks and their systems (ISO/DIS 3691-4:2006)*, ISO Std., 2011.

[4] *Safety of machinery – Positioning of safeguards with respect to the approach speeds of parts of the human body (ISO 13855:2010)*, ISO Std., 2010.

[5] *Robots and robotic devices – Safety requirements for personal care robots (ISO 13482:2014)*, ISO Std., 2014.

[6] K. Maček, D. Vasquez, T. Fraichard, and R. Siegwart, "Towards safe vehicle navigation in dynamic urban scenarios," *Automatika*, vol. 50, no. 3-4, pp. 184–194, 2009.

[7] M. Althoff and J. M. Dolan, "Online verification of automated road vehicles using reachability analysis," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 903–918, 2014.

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

[9] S. Bouraine, T. Fraichard, and H. Salhi, "Relaxing the inevitable collision state concept to address provably safe mobile robot navigation with limited field-of-views in unknown dynamic environments," in *Proc. of IROS 2011*, pp. 2985–2991.

[10] F. Immler, "Verified reachability analysis of continuous systems," in *TACAS 2015*, ser. Lecture Notes in Computer Science. Springer, 2015, vol. 9035, pp. 37–51.

[11] S. Mitsch, K. Ghorbal, and A. Platzer, "On provably safe obstacle avoidance for autonomous robotic ground vehicles," in *Proc. of Robotics: Science and Systems*, 2013.

[12] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Reachset conformance testing of hybrid automata," in *Proc. of the HSCC*, 2016, pp. 277–286.

[13] M. Althoff, D. Hess, and F. Gambert, "Road occupancy prediction of traffic participants," in *16th International IEEE Conf. on Intelligent Transportation Systems*, 2013, pp. 99–105.

[14] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: modeling social behavior for multi-target tracking," in *Proc. of ICCV*, 2009, pp. 261–268.

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

[16] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[17] L. Martinez-Gomez and T. Fraichard, "Collision avoidance in dynamic environments: An ics-based solution and its comparative evaluation," in *Proc. of ICRA 2009*, pp. 100–105.

[18] F. Large, C. Laugier, and Z. Shiller, "Navigation among moving obstacles using the nlvo: Principles and applications to intelligent vehicles," *Auton Robots*, vol. 19, no. 2, pp. 159–171, 2005.

[19] M. Seder and I. Petrović, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Proc. of ICRA 2007*, pp. 1986–1991.

[20] M. Zhang and X. Zhang, "Formally verifying navigation safety for ground robots," in *2016 IEEE Intern. Conf. on Mechatronics and Automation*, 2016, pp. 1000–1005.

[21] C. Dabadie, S. Kaynama, and C. J. Tomlin, "A practical reachability-based collision avoidance algorithm for sampled-data systems: Application to ground robots," in *Proc. of IROS 2014*, pp. 4161–4168.

[22] A. Aniculaesei, D. Arnsberger, F. Howar, and A. Rausch, "Towards the verification of safety-critical autonomous systems in unknown environments," in *Proc. of the 1st International Workshop on Verification and Validation of Cyber-Physical Systems*, 2016.

[23] D. Hess, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *Proc. of IROS 2014*, 2014, pp. 1474–1481.

[24] A. Majumdar and R. Tedrake, "Robust online motion planning with regions of finite time invariance," in *Algorithmic Foundations of Robotics X*, ser. Springer Tracts in Advanced Robotics. Springer, 2013, vol. 86, pp. 543–558.

[25] R. Alami, K. M. Krishna, and T. Siméon, "Provably safe motions strategies for mobile robots in dynamic domains," in *Autonomous Navigation in Dynamic Environments*, ser. Springer Tracts in Advanced Robotics. Springer, 2007, vol. 35, pp. 85–106.

[26] W. Chung, S. Kim, M. Choi, J. Choi, H. Kim, C.-b. Moon, and J.-B. Song, "Safe navigation of a mobile robot considering visibility of environment," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3941–3950, 2009.

[27] A. Bautin, L. Martinez-Gomez, and T. Fraichard, "Inevitable collision states: A probabilistic perspective," in *Proc. of ICRA 2010*, pp. 4022–4027.

[28] D. Althoff, M. Althoff, D. Wollherr, and M. Buss, "Probabilistic collision state checker for crowded environments," in *Proc. of ICRA 2010*, pp. 1492–1498.

[29] D. Althoff, J. J. Kuffner, D. Wollherr, and M. Buss, "Safety assessment of robot trajectories for navigation in uncertain and dynamic environments," *Auton Robots*, vol. 32, no. 3, pp. 285–302, 2012.

## A.3 Online Verification of Impact-Force-Limiting Control for Physical Human-Robot Interaction [3]

**Summary** The safety problem of this paper is to verify that in case of an unintended collision between humans and robots, the impact force never exceeds a limit, e.g., a pain limit as defined in ISO/TS-15066 [18]. In contrast to our previous work on human-robot interaction [2, 9], where we verified that the robot never collides with humans when it is moving, this safety objective allows robot motion during a collision, as long as the impact force threshold is not violated. Such safety objectives would further increase human-robot interaction efficiency and relieve the freezing-robot problem. To realize that, we use the same online verification framework for motion planning introduced in [2, 9], but with a changed safety objective and interaction modeling.

For the safety verification, we use a model for predicting human arm and robot arm motion (same as [9], and switch to a human-robot physical interaction model if a collision is predicted. The physical interaction model accounts for the stiffness and damping of the impedance-controlled robot and the human tissue, as well as the masses of the robot and human. In addition, we distinguish between constrained collisions (the human hand is clamped between the robot and another object) and unconstrained collisions (the human hand can move freely). Reachset conformance of the model is shown through a series of real constrained and unconstrained collision experiments, and the verification model has been found using the grey-box identification approach, such that the measured collision force is always contained within the predicted reachable set of the collision force.

The algorithm is tested on a real six-degrees-of-freedom robot, where the end-effector interacts with a human hand. We demonstrate that the online verification approach for motion planning always reduces the robot's speed to prevent unsafe collisions. This method is effective with or without sensors for human arm detection, although arm detection helps the robot to increase its speed when the human is not near the robot. For the first time, we use formal methods to verify controllers for physical human-robot interaction.

**Author Contributions** **S. L.** developed the verification algorithm (Alg. 1) and the models required for verification. **S. L.** developed the grey-box identification approach. **S. L.** designed, conducted and evaluated the experiments. **S. L.** wrote the article. M. A. led the research project, provided feedback, and helped improve the manuscript.

**Conference Paper** The accepted version of the conference paper is reprinted in this thesis. The final version of the record is available at `https://doi.org/10.1109/IROS51168.2021.9636610`.

# Online Verification of Impact-Force-Limiting Control for Physical Human-Robot Interaction

Stefan B. Liu, and Matthias Althoff

***Abstract*— Humans must remain unharmed during their interaction with robots. We present a new method guaranteeing impact force limits when humans and robots share a workspace. Formal guarantees are realized using an online verification method, which plans and verifies fail-safe maneuvers through predicting reachable impact forces by considering all future possible scenarios. We model collisions as a coupled human-robot dynamical system with uncertainties and identify reachset-conforming models based on real-world collision experiments. The effectiveness of our approach for human-robot co-existence is demonstrated for the human hand interacting with the end effector of a six-axis robot manipulator with force sensing. By integrating a human pose detection system, the efficiency of robot movements increases.**

## I. Introduction

Humans and robots are sharing their workspaces, collaborating, and interacting with each other. Trending application areas include collaborative manufacturing, assistive robotics for rehabilitation and elderly care, and robotic surgery. When designing robot controllers, safety is one of the top priorities; humans should never be harmed or injured. To mitigate pain, ISO/TS 15066 [1] defined interaction-force thresholds for each body part, which should not be exceeded. However, guaranteeing safety through limiting these forces is a challenging task:

- The human body is capable of performing a range of movements, making it difficult to predict exact collision scenarios.
- Interaction forces depend on the mechanical properties of robots and humans. These are subject to uncertainty, e.g., stiffnesses changes according to muscle activity.
- Varying tasks and diverse environments create many possibilities for collision, thus, offline assessments become infeasible. Therefore, an online approach should be preferred, considering only the current situation.
- To guarantee safety properties despite uncertainties, formal methods should be used.

We propose to tackle these challenges through an online verification approach based on human pose detection, fail-safe planning, and reachability analysis. The fail-safe planner decides whether an upcoming motion command can be executed by verifying the safety of possible fail-safe maneuvers. The online proofs are based on reachability analysis, which checks, whether all possible interaction forces are within specified limits. Reachable sets are computed using a model of the coupled human-robot interaction dynamics. Uncertainties in the system, such as human velocity, collision time, varying stiffnesses, control performance are all modeled as sets, and the interaction dynamics are identified in a way that preserves reachset conformance with real behaviors.

Most of the previous approaches only assess safety without proving thresholds. Shivakumar et al. [2] propose that impact forces with environmental objects can be predicted using a spring-damper model or an energy-based model. Yamada et al. [3] describe how to design the thickness of a viscoelastic coat for robots to avoid exceeding pain limits during collisions. Ikuta et al. [4] introduce a danger index relative to the maximum tolerable collision force at the end effector, which depends on factors such as the robot's mass and velocity and joint- and and coating elasticities. Heinzmann and Zelinski [5] propose an online safety controller that derives admissible control torques from the maximum collision forces of a rigid robot, coupled with scaling of the robot velocity. Models used in [3]–[5], however, assume that human is a rigid obstacle, which reduces uncertainty but contributes to a conservative force estimation. Post-collision force-limiting strategies in Navarro et al. [6] and Li et al. [7] focus on reactive behavior for overshoots during the interaction, however, it cannot guarantee impact-force limits. Some non-mentioned works use the model provided in ISO/TS 15066 [1] to guarantee force limits. However, Kirschner et al. [8] reported that the model is inaccurate and unsuitable for estimating collision forces. In contrast to these non-formal studies, we consider impedance models with reachset conforming uncertainties to provide formal guarantees.

In addition, alternative metrics for reducing impact injury have been proposed, involving velocity [9], [10] or energy and power [11]–[14], which are easier to evaluate, since only the robot model is required. Haddadin et al. [9] realized that injury occurrence is directly related to the impact velocity beyond a certain robot mass. A database has been implemented by Mansfeld et al. [10], which can be used for online and offline injury assessments based on collision speeds and robot modeling. Meguenani et al. [11] indirectly limit impact force by limiting the kinetic and potential energy of the robot. Raiola et al. [12] scale the stiffness and damping of impedance controllers to guarantee energy and power limits. The port-Hamiltonian formulation of coupled human-robot dynamics in [13], [14] allows one to directly control energy in physical interaction to preserve passivity. The difficulty with speed, energy, and power metrics is that suitable limits are unavailable, or are based on the non-formal derivations from [1]. In contrast, we verify established force-based pain

Authors are with Cyber-Physical Systems Group, Department of Informatics, Technical University of Munich, 85748 Garching, Germany [stefan.liu,althoff]@tum.de

limits [1] for humans.

Our study is the first one that uses formal methods to verify controllers for physical human-robot interaction. In addition, we provide an identification method for models and uncertainties based on real-world experiments. Also, other methods for formal verification, such as differential dynamic logic theorem-proving [15] and inevitable collision states [16] consider uncertainties in dynamical systems.

This study is structured as follows: we define the safety properties to be verified in Sec. II. Modeling and identification of the coupled human-robot dynamics are discussed in Sec. III. The impact-force-limiting controller is presented in Sec. IV. The experimental evaluation in Sec. V demonstrates the effectiveness of our approach on a real interaction scenario, followed by the conclusions in Sec. VI.

## II. Safety objectives

This section poses the safety problem that is encountered inbetween humans and robots. We denote sets in calligraphic letters (e.g., $\mathcal{A}$), matrices with upper case letters (e.g., $A$), vectors by $\vec{\cdot}$, and scalar values by lower case letters (e.g., $a$). Considering a system with state vector $\vec{z}$, input vector $\vec{u}$, and parameters $\vec{p}$, of which the dynamical equation is $\dot{\vec{z}} = \vec{f}(\vec{z}, \vec{u}, \vec{p})$. We make use of reachable sets, which are defined as follows:

**Definition 1** (Reachable Set)**.** Given the initial set $\mathcal{Z}_0$, the uncertain input set $\mathcal{U}$, and the non-deterministic parameter set $\mathcal{P}$, the reachable set of $\dot{\vec{z}} = \vec{f}(\vec{z}, \vec{u}, \vec{p})$ at time $t$ is

$$\mathcal{R}(t) = \Big\{ \int_0^t \vec{f}(\vec{z}(\tau), \vec{u}(\tau), \vec{p}) d\tau + \vec{z}(0) \Big| \vec{z}(0) \in \mathcal{Z}_0, \\ \forall \tau \in [0, t] : \vec{u}(\tau) \in \mathcal{U}, \vec{p}(\tau) \in \mathcal{P} \Big\}.$$

To compute $\mathcal{R}(t)$ (also denoted as `reach(`$\mathcal{Z}_0$,$\mathcal{U}$,$\mathcal{P}$`)`), we use an optimized version of the software CORA [17].

We regard systems consisting of humans sharing a workspace with a robot manipulator. From the goal that a robot should not actively cause harm to the human, we derive three safety objectives:

1) A non-moving robot cannot actively cause harm to a human. Consider a robot manipulator with $n$ degree of freedoms, where $\vec{q}, \dot{\vec{q}} \in \mathbb{R}^n$ are the joint position and velocity of the robot, and $\vec{x} = [\vec{q}, \dot{\vec{q}}]^T$ is its state. Let us define the predicate $\mathrm{standstill}(t)$ indicating whether the system is safe:
$$\mathrm{standstill}(t) \iff \vec{x}(t) \in \mathcal{ISS} := \mathbb{R}^n \times \vec{0},$$
where $\vec{0}$ is a vector of $n$ zeros. We refer to the set on the right hand side as an *invariably safe set*, implying that our system is safe for an infinite time horizon when it is reached.
2) We consider that a robot cannot cause harm to the human, if they are not physically interacting, i.e., the occupied space of the human does not overlap with the occupied space of the robot. We denote $\mathcal{M}(t)$ and $\mathcal{H}(t)$ as occupancy sets of the robot and human, respectively:
$$\mathrm{noInteraction}(t) \iff \mathcal{M}(t) \cap \mathcal{H}(t) = \emptyset.$$
To predict occupancy sets of humans, a tracking system is required. For additional information, we refer to our previous work in [18], [19].
3) We consider that harm is caused to the human if force thresholds are violated during an impact. For ISO/TS 15066, two limits are defined: a *transient force* limit $f_{\mathrm{tra,lim}}$, which is the peak at the beginning of a collision, and the quasi-static force $f_{\mathrm{qs,lim}}$ limit, which is the converged stationary force acting on a clamped human. We introduce the reachable set of the absolute force $\mathcal{F}_{\mathrm{coll}}(\tau) \subseteq \mathbb{R}, t < \tau < t + t_e$, where $t_e$ is a prediction horizon. Our system is safe if
$$\mathrm{safeForce}(t) \iff \sup\left(\mathcal{F}_{\mathrm{coll}}(\tau)\right) \leq f_{\mathrm{tra,lim}} \\ \wedge \lim_{\tau \to t_e} \sup(\mathcal{F}_{\mathrm{coll}}(\tau)) \leq f_{\mathrm{qs,lim}},$$
where $\sup$ is the supremum, and $t_e$ needs to be large to converge to the quasi-static force.

We consider a system to be verified as safe, if any of the above three conditions hold at all times:

$$\forall t : \mathrm{standstill}(t) \vee \mathrm{noInteraction}(t) \\ \vee \mathrm{safeForce}(t) \iff \text{safe}. \quad (1)$$

The remaining part of this paper focuses on the prediction of reachable forces to evaluate the predicate $\mathrm{safeForce}(t)$. For the other predicates, we refer to [18], [19].

## III. Interaction Modeling

To represent physical interaction, we state the dynamical models with uncertainties in Sec. III-A, and present its reachset-conforming model identification in Sec. III-B.

### A. Physical interaction modeling

The goal of the model is to predict the set of reachable forces $\mathcal{F}_{\mathrm{coll}}$, given the planned robot trajectory, and the human and robot collision velocities. We make the following assumptions:

- We model the case of a hand interacting with the robot end-effector, which is controlled by a Cartesian impedance controller.
- The collision is a blunt impact with any part of the end effector from any direction, for which the force limits apply [1]. We do not consider robots with sharp edges; for their safety analysis, pressure limits apply [1].
- The collision is unintended, i.e., the human does not push against the robot, and remains passive after impact.

In addition, we use a scalar model to represent the dynamics in all possible (three-dim.) spatial directions. A projection operator over-approximatively transforms three-dimensional inputs of our models into a one-dimensional interval:

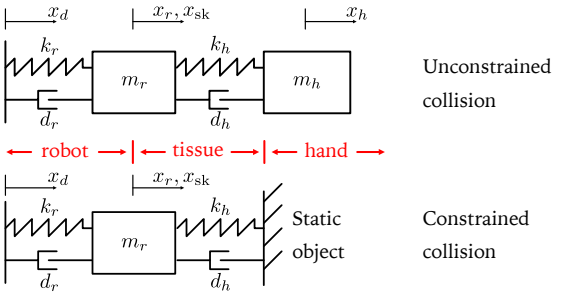


Fig. 1. Physical interaction is modelled as mass-spring-damper systems.

**Definition 2** (Scalar projection of a set)**.** The scalar projection of a three-dimensional set $\mathcal{S}$ is defined as an interval

$$\texttt{proj}(\mathcal{S}) := [-\|\mathcal{S}\|_2, \|\mathcal{S}\|_2],$$

where the 2-norm of a set is $\|\mathcal{S}\|_2 := \sup\{\|s\|_2 | s \in \mathcal{S}\}$.

Our modeling approach takes the following steps: 1) derive a human model, 2) derive a robot model, and 3) couple the dynamics and introduce uncertainty into the model. We distinguish between two types of collisions [20]:

1) *Unconstrained collision*: the human hand can move away after a collision, i.e., it is not clamped.
2) *Constrained collision*: the human hand is clamped between the robot end-effector and another static object.

We choose mass-spring-damper systems to model both interactions (Fig. 1). For the unconstrained collision, the human hand is modeled by a moving mass $m_h$, where $x_{\text{sk}}$ are the hand and skin position, respectively, and the impact force is $f_{\text{coll},1}$. The skin has a tissue stiffness $k_h$, and a damping $d_h$:

$$m_h \ddot{x}_h = \underbrace{k_h(x_{\text{sk}} - x_h) + d_h(\dot{x}_{\text{sk}} - \dot{x}_h)}_{f_{\text{coll},1}}, \tag{2}$$

For the constrained collision, the hand position is assumed to be fixed, thus cannot move ($x_h, \dot{x}_h, \ddot{x}_h = 0$). Therefore, we define the dynamical equation as

$$f_{\text{coll},2} = k_h x_{\text{sk}} + d_h \dot{x}_{\text{sk}}. \tag{3}$$

To model the robot, we consider the rigid-body dynamics

$$M(\vec{q})\ddot{\vec{q}} + C(\vec{q}, \dot{\vec{q}})\dot{\vec{q}} + \vec{g}(\vec{q}) = \vec{\tau} + J(\vec{q})^T \vec{f}_{\text{ext}}, \tag{4}$$

where $\vec{q}$ is the joint position, $M(\vec{q})$ the mass matrix, $C(\vec{q}, \dot{\vec{q}})$ the Coriolis and centripetal matrix, $\vec{g}(\vec{q})$ the gravity torques, $J(\vec{q})$ the Jacobian, $\vec{f}_{\text{ext}}$ the measured external force at the end effector, and $\vec{\tau}$ the input torque. To track the desired trajectory $\vec{x}_d(t)$, we use a Cartesian impedance controller [21]—a prominent method for controlling human-robot interaction [22]—given by

$$\begin{aligned}
\vec{\tau} =& \vec{g}(\vec{q}) + J(\vec{q})^T(\Lambda(\vec{q})\ddot{\vec{x}}_d + \mu(\vec{q}, \dot{\vec{q}})\dot{\vec{x}}_r) - \\
& J(\vec{q})^T \Lambda(\vec{q}) \Lambda_r^{-1}(K_r(\vec{x} - \vec{x}_d) + D_r(\dot{\vec{x}} - \dot{\vec{x}}_d)) + \\
& J(\vec{q})^T(\Lambda(\vec{q})\Lambda_r^{-1} - I)\vec{f}_{\text{ext}}, \\
\Lambda(\vec{q}) =& J(\vec{q})^{-T} M(\vec{q}) J(\vec{q})^{-1}, \\
\mu(\vec{q}, \dot{\vec{q}}) =& J(\vec{q})^{-T} \left( C(\vec{q}, \dot{\vec{q}}) - M(\vec{q}) J(\vec{q})^{-1} \dot{J}(\vec{q}) \right) J(\vec{q})^{-1},
\end{aligned}$$

where $\vec{x}_r$ is the end effector position, $\Lambda_r$ is the desired mass matrix, $K_r$ is the desired stiffness matrix, and $D_r$ is the desired damping matrix. Thus, the end effector behaves like a mass-spring-damper system, which can be seen in the closed-loop robot dynamics [21]:

$$\Lambda_r(\ddot{\vec{x}}_r - \ddot{\vec{x}}_d) + D_r(\dot{\vec{x}}_r - \dot{\vec{x}}_d) + K_r(\vec{x}_r - \vec{x}_d) = \vec{f}_{\text{ext}}. \tag{5}$$

We consider only the translational part of the closed-loop dynamics since our interest is in translational forces, i.e., $\vec{x}_r, \vec{x}_d, \vec{f}_{\text{ext}} \in \mathbb{R}^3$ and $\Lambda_r, D_r, K_r \in \mathbb{R}^{3\times 3}$. When choosing $\Lambda_r = m_r I, D_r = d_r I$ and $K_r = k_r I$, where $m_r, d_r, k_r$ are scalars and $I$ is a three-dimensional identity matrix, then the following equation

$$m_r(\ddot{x}_r - \ddot{x}_d) + d_r(\dot{x}_r - \dot{x}_d) + k_r(x_r - x_d) = f_{\text{ext}} \tag{6}$$

is the orthogonal projection of (5) onto any spatial direction.

We derive the coupled dynamics of the unconstrained collision by coupling the forces $f_{\text{coll},1,2} = -f_{\text{ext}}$, connecting the end effector to the skin of the human hand $x_r = x_{\text{sk}}$, and inserting (2) into (6). Given the vector $\vec{z}_1 = [x_r, \dot{x}_r, x_h, \dot{x}_h]^T$, the state-space representation of the dynamics is:

$$\dot{\vec{z}}_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(k_r+k_h)}{m_r} & -\frac{(d_h+d_r)}{m_r} & \frac{k_r}{m_r} & \frac{d_h}{m_r} \\ 0 & 0 & 0 & 1 \\ \frac{k_h}{m_r} & \frac{d_h}{m_r} & -\frac{k_h}{m_r} & -\frac{d_h}{m_r} \end{bmatrix} \vec{z}_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u + \vec{w}_1, \tag{7}$$

$$f_{\text{coll},1} = \begin{bmatrix} k_h & d_h & -k_h & -d_h \end{bmatrix} \vec{z}_1 + v_1, \tag{8}$$

and $u$ is an orthogonal projection of

$$\vec{u} = \ddot{\vec{x}}_d + \Lambda_r^{-1} D_r \dot{\vec{x}}_d + \Lambda_r^{-1} K_r \vec{x}_d \tag{9}$$

The coupled dynamics for the constrained collision are derived by inserting (3) into (6). Given state $\vec{z}_2 = [x_r, \dot{x}_r]^T$:

$$\dot{\vec{z}}_2 = \begin{bmatrix} 0 & 1 \\ -\frac{(k_r+k_h)}{m_r} & -\frac{(d_h+d_r)}{m_r} \end{bmatrix} \vec{z}_2 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \vec{w}_2 \tag{10}$$

$$f_{\text{coll},2} = \begin{bmatrix} k_h & d_h \end{bmatrix} \vec{z}_2 + v_2. \tag{11}$$

The state-space dynamics have been augmented by additive disturbances $\vec{w}_1 \in \mathcal{W}_1, \vec{w}_2 \in \mathcal{W}_2$ and $v_1 \in \mathcal{V}_1, v_2 \in \mathcal{V}_2$, which shall represent the model uncertainties. We assume that the system starts in a relaxed state $x_r(0) = x_h(0) = 0$, and without loss of generality, we set $x_r(0)$ as the origin of the variables $x_r, x_h$, and $x_d$. We can now apply Def. 1 to compute the reachable forces $f_{\text{coll},1} \in \mathcal{F}_{\text{coll},1}$ and $f_{\text{coll},2} \in \mathcal{F}_{\text{coll},2}$.

The Cartesian impedance controller is a convenient choice, since the resulting coupled dynamics are linear in the Cartesian spatial dimensions. Reachable sets of linear systems can be efficiently computed [23]. Generally, choosing other robot controllers is also possible, and the coupled dynamics can be derived similarly. Then, the systems are generally non-linear. The generalization into three-dimensional models is straightforward; the mass, spring, and damping parameters for both robots and humans are replaced by three-dimensional

matrices. The proj() operator is not needed anymore, reducing over-approximativity. The three-dimensional model is general, however, the number of parameters increases, which makes the model identification difficult. The number of states increases from 4 to 12 for the unconstrained collision model, which leads to a slower reachability analysis. A typical algorithm with zonotopic set-representation has complexity $\mathcal{O}(n^3)$ [24], where $n$ is the number of states.

### B. Reachset-conforming model identification

For our chosen interaction models in (7)–(11), only the parameters $m_r, d_r$, and $k_r$ of the Cartesian impedance controller are known. The parameters $m_h, d_h$, and $k_h$, as well as the uncertainties $\mathcal{P}_1 = \{\mathcal{W}_1, \mathcal{V}_1\}, \mathcal{P}_2 = \{\mathcal{W}_2, \mathcal{V}_2\}$, are unknown.

The parameters are selected in a way that allows the reachable sets $\mathcal{F}_{\text{coll}}$ to include the behavior of the real system. We also refer to this property as *reachset conformance* [25]. We propose to ensure this property by means of testing the real system: from real collision experiments, we collect the inputs for our models, which are the initial states $\vec{z}_1(0)$, $\vec{z}_1(0)$, and $u$. We then make a forward prediction using a set of parameters and check if measured forces $f_m(t)$ are contained in $\mathcal{F}_{\text{coll}}(t)$ for all times. We wish to keep the reachable sets as small as possible.

Given $m$ test cases, we formulate the identification as a constrained optimization problem minimizing the norm of the reachable sets, where $\mathcal{P}$ are the unknown parameters:

$$\min_{\mathcal{P}} \quad \sum_{1\le i\le m} \int_0^{t^*} \|\mathcal{F}^{(i)}_{\text{coll},\mathcal{P}}(t)\| dt, \tag{12a}$$

$$\text{subject to} \quad \forall i \, \forall t : f_m(t) \subseteq \mathcal{F}^{(i)}_{\text{coll},\mathcal{P}}(t). \tag{12b}$$

Because (7)–(11) are linear systems, the above optimization can be solved in a nested fashion, according to [25]: an inner loop computes the cost of optimal disturbances $\mathcal{W}$ and $\mathcal{V}$ using linear programming, given $m_h, d_h$, and $k_h$; an outer loop uses nonlinear programming to find $m_h, d_h$, and $k_h$ with the smallest cost computed using the inner loop.

For safety analysis, reachset-conformant force predictions are sufficient. Requiring other variables (e.g., position trajectories) to be reachset-conformant would pose unnecessary constraints on the identification, which leads to more conservative models.

## IV. Online Verification

This section describes our novel online verification procedure for our novel impact-force-limiting control, which always ensures the safety objective in (1). We first illustrate the fail-safe planning framework in Sec. IV-A, and then present our algorithm for evaluating $\text{safeForce}(t_k)$ in Sec. IV-B.

### A. Fail-safe planning

The main idea of fail-safe planning [18], [26] is that during normal operation, the controller aims to generate and verify *fail-safe maneuvers*, as shown in Fig. 2. The next section

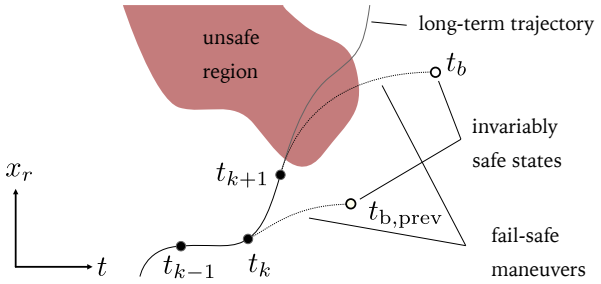


Fig. 2. *Fail-safe planning*: The robot is moving on the trajectory on $[t_{k-1}, t_k]$. The verification of the fail-safe maneuver on $[t_{k+1}, t_b]$ fails, because it passes an unsafe region. Therefore, the robot will execute the previously stored verified maneuver for $[t_k, t_{b,prev}]$.

of an intended *long-term trajectory* in the time interval $[t_k, t_{k+1}]$ can only be executed, if a consecutive and verified fail-safe maneuver to the invariably safe set $\mathcal{ISS}$ exists, while satisfying (1) upon reaching the $\mathcal{ISS}$. If a verified fail-safe maneuver cannot be found, then the previously verified one, starting at $t_k$, will be immediately executed, as can be seen in Fig. 2. If the robot has already been in a fail-safe maneuver during $[t_{k-1}, t_k]$, it is attempted to verify and then execute a *recovery maneuver* for $[t_k, t_{k+1}]$ to bring the system back to the long-term trajectory. Similar to [18], we limit ourselves to path-consistent fail-safe and recovery maneuvers [27] in this work to focus on the novel aspect of limiting forces.

Let us denote the time of reaching the $\mathcal{ISS}$ as $t_b$. A fail-safe maneuver for $[t_{k+1}, t_b]$ is verified by first checking the predicate noInteraction, by computing the reachable occupancies of the robot and the tracked human for the interval $[t_k, t_b]$. If the occupancies indicate a possible collision at $t_k \le t_c < t_b$, we also check the predicate safeForce for the interval $[t_c, t_b]$. If humans are not tracked, then we disregard noInteraction and directly evaluate safeForce by setting $t_c = t_k$. If an actual collision occurs, as measured by force sensors, then the robot brakes, until the force acting on the robot has vanished.

In practice, due to the interplay of intended trajectories, fail-safe trajectories, and recovery trajectories, the speed of the robot will always be as high as safely possible. Thus, it is not necessary to offline design a safe long-term trajectory.

### B. Verifying compliance to impact-force limits

We present Alg. 1, which verifies at each time instant $t_k$ that a fail-safe maneuver adheres to both transient and quasi-static force limits. Given is the maneuver $\vec{x}_d(t)$ for $t \in [t_k, t_b]$, which brings the robot to an $\mathcal{ISS}$. We first compute the reachable occupancies of the human $\mathcal{H}([t_k, t_b])$ and of the robot $\mathcal{M}([t_k, t_b])$, using the approach in [19], to detect possible future collisions, which trigger subsequent force evaluations. In case of a potential collision, we compute the set of reachable forces $\mathcal{F}_{\text{coll},1}$ for possible unconstrained collisions and $\mathcal{F}_{\text{coll},2}$ for possible constrained collision dynamics, as presented Sec. III-A. However, additional uncertainties have to be considered here:

**Algorithm 1** Verification of $\mathrm{safeForce}(t_k)$

**Input:** $\vec{x}_d(t), t \in [t_k, t_b]$
**Output:** `isSafe`

1: $\mathcal{M}(t), \mathcal{H}(t) \leftarrow$ (see [19])
2: find $t_c$, s.t. $\mathcal{M}([t_c, t_b]) \cap \mathcal{H}([t_c, t_b]) \neq \emptyset$
3: $\vec{x}_r(t) \leftarrow \vec{x}_d(t) + \mathcal{E}_r$ {assume tracking error bound}
4: $\dot{\vec{x}}_r(t) \leftarrow \dot{\vec{x}}_d(t) + \mathring{\mathcal{E}}_r$
5: $\mathcal{U} \leftarrow \mathtt{proj}(\vec{u}([t_c, t_b]))$ {uncertain input set}
6: $\mathring{\mathcal{X}}_r \leftarrow \mathtt{proj}(\dot{\vec{x}}_r([t_c, t_b]))$ {robot collision velocities}
7: $\mathring{\mathcal{X}}_h \leftarrow [-v_{\max}, v_{\max}]$ {maximum hand velocities}
8: $\mathcal{Z}_{0,1} \leftarrow 0 \times \mathring{\mathcal{X}}_r \times 0 \times \mathring{\mathcal{X}}_h$ {initial set for $z_1$}
9: $\mathcal{Z}_{0,2} \leftarrow 0 \times \mathring{\mathcal{X}}_r$ {initial set for $z_2$}
10: $\mathcal{F}_{\mathrm{coll},1}(\tau) \leftarrow \mathtt{reach}_1(\mathcal{Z}_{0,1}, \mathcal{U}, \mathcal{P}_1)$
11: $\mathcal{F}_{\mathrm{coll},2}(\tau) \leftarrow \mathtt{reach}_2(\mathcal{Z}_{0,2}, \mathcal{U}, \mathcal{P}_2)$
12: $f_{\mathrm{tra}} = \sup(\mathcal{F}_{\mathrm{coll},1}(\tau) \cup \mathcal{F}_{\mathrm{coll},2}(\tau)) \quad \text{for} \quad 0 \leq \tau \leq t_e$
13: $f_{\mathrm{qs}} = \lim_{\tau \to t_e} \sup(\mathcal{F}_{\mathrm{coll},2}(\tau))$
14: **if** $f_{\mathrm{tra}} \leq f_{\mathrm{tra,lim}} \wedge f_{\mathrm{qs}} \leq f_{\mathrm{qs,lim}}$ **then**
15: `isSafe` $\leftarrow$ `true`
16: **else**
17: `isSafe` $\leftarrow$ `false`
18: **end if**

- Due to the acceleration capabilities of the human hand, we cannot predict its future velocity. Thus, we assume that it is bounded by an interval $\mathring{\mathcal{X}}_h := [-v_{\max}, v_{\max}]$. E.g., $v_{\max} = 2$ (m/s) complies with ISO 13855 [28].
- We assume that the robot position $\vec{x}_r$ and velocity $\dot{\vec{x}}_r$ are bounded by the errors $\mathcal{E}_r, \mathring{\mathcal{E}}_r \in \mathbb{R}^3$ around the desired trajectory before a collision.
- The collision time can be at any $t \in [t_c, t_b]$. Therefore, the collision speed of the robot is uncertain, but can be bounded by the union of all possible robot velocities $\mathring{\mathcal{X}}_r = \mathtt{proj}(\dot{\vec{x}}_r([t_c, t_b]))$. Similarly, we bound the input by a $\mathcal{U} = \mathtt{proj}(\vec{u}([t_c, t_b]))$.

The initial sets are defined as $\mathcal{Z}_{0,1} := 0 \times \mathring{\mathcal{X}}_r \times 0 \times \mathring{\mathcal{X}}_h$ and $\mathcal{Z}_{0,2} := 0 \times \mathring{\mathcal{X}}_r$, accounting for the above uncertainties. The operations in line 10–18 of Alg. 1 compute the reachable sets and evaluate the predicate $\mathrm{safeForce}$ from Sec. II.

## V. Experimental results

This section experimentally evaluates our verified impact-force-limiting control for the interaction of a robot end-effector with the right hand of a human. The robot used is a Schunk LWA-4P lightweight robot using the Cartesian impedance controller from Sec. III-A, where the desired impedances are chosen as $\Lambda_r = 5I, D_r = 50I$, and $K_r = 150I$. To measure the impact force at the end effector, we designed a custom 3D-printed blunt impactor, which contains a 6-axis force-torque sensor. To measure the hand position and velocities, we use a Vicon Vero motion capture system. We show the experimental identification of the physical interaction model in Sec. V-A. We show the effectiveness of our controller in a human-robot co-existence scenario by comparing the robot performance with and without human tracking in Sec. V-B.

TABLE I
Identified parameters of unconstrained (up) and constrained (down) collision models

| Dim. | $\mathcal{W}_1$ | $\mathcal{V}_1$ | Param. | Value |
|---|---|---|---|---|
| 1 | 0.196 | $[-79.27, 85.33]$ | $m_h$ | 0.29 |
| 2 | 19.20 | - | $d_h$ | 55.05 |
| 3 | 0 | - | $k_h$ | 5434 |
| 4 | 0 | - | | |
| **Dim.** | $\mathcal{W}_2$ | $\mathcal{V}_2$ | **Param.** | **Value** |
| 1 | $-0.498$ | $[-69.67, 38.37]$ | $d_h$ | 719.3 |
| 2 | 5.459 | - | $k_h$ | 29900 |

### A. Results for model identification

We use our approach in Sec. III-B to identify reachset conforming model parameters. For that, two series of tests are conducted with the impedance-controlled robot, one for the unconstrained collision model, and the other for the constrained collision model. In the first experiment, multiple collisions of a hand with the end effector are initiated from random directions, and with random parts of the hand. In the second experiment, the robot collides with a resting hand on a table at different velocities. The hand is moved around in-between experiments, such that different parts of the hand are clamped. Due to safety reasons, we only did a reduced amount of tests, and these experiments were only conducted by the first author of this paper. Forty-three collisions have been evaluated for identifying the unconstrained collision model, whereas 41 collisions for the constrained collision model. The identified parameters are shown in Tab. I.

The results for a few randomly selected test cases are plotted in Fig. 3. For the unconstrained collision model, we only test until $t_e = 0.06$ seconds, since the impact transient has finished for all test cases at that time. For the constrained collision model, we test until $t_e = 0.5s$, because we are interested in the quasi-static force, to which our system converges. Regarding the values in Tab. I, we observe that the identified stiffnesses are smaller than in other works (e.g., [1]), and the damping values are high. The reason is that the identification algorithm decided that it is more effective (i.e., smaller reachable sets) to let the nominal parameters $m_h, k_h$, and $d_h$ model the low-frequency dynamics, whereas high-frequency dynamics resulting from high stiffness and low damping are lumped inside the sets $\mathcal{V}_{1,2}$.

### B. Results for the impact-force-limiting control

We demonstrate the effectiveness of the impact-force-limiting control using our online verification approach described in Sec. IV. We consider two scenarios. In the first scenario, we assume that human hand tracking is available to the controller, i.e., the collision time $t_c \geq t_k$. In the second scenario, tracking is not available, i.e., we set $t_c = t_k$. The robot moves between the joint angles $q_1 = [\frac{\pi}{2}, \frac{\pi}{6}, -\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, 0]^T$ and $q_2 = [-\frac{\pi}{4}, \frac{\pi}{6}, -\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2}, 0]^T$, for three times. At the first time, the human does not intervene. At the second time, the human intervenes without a collision, and at the third time with a collision. We set the transient force limit to 220 N, and quasi-static force limit

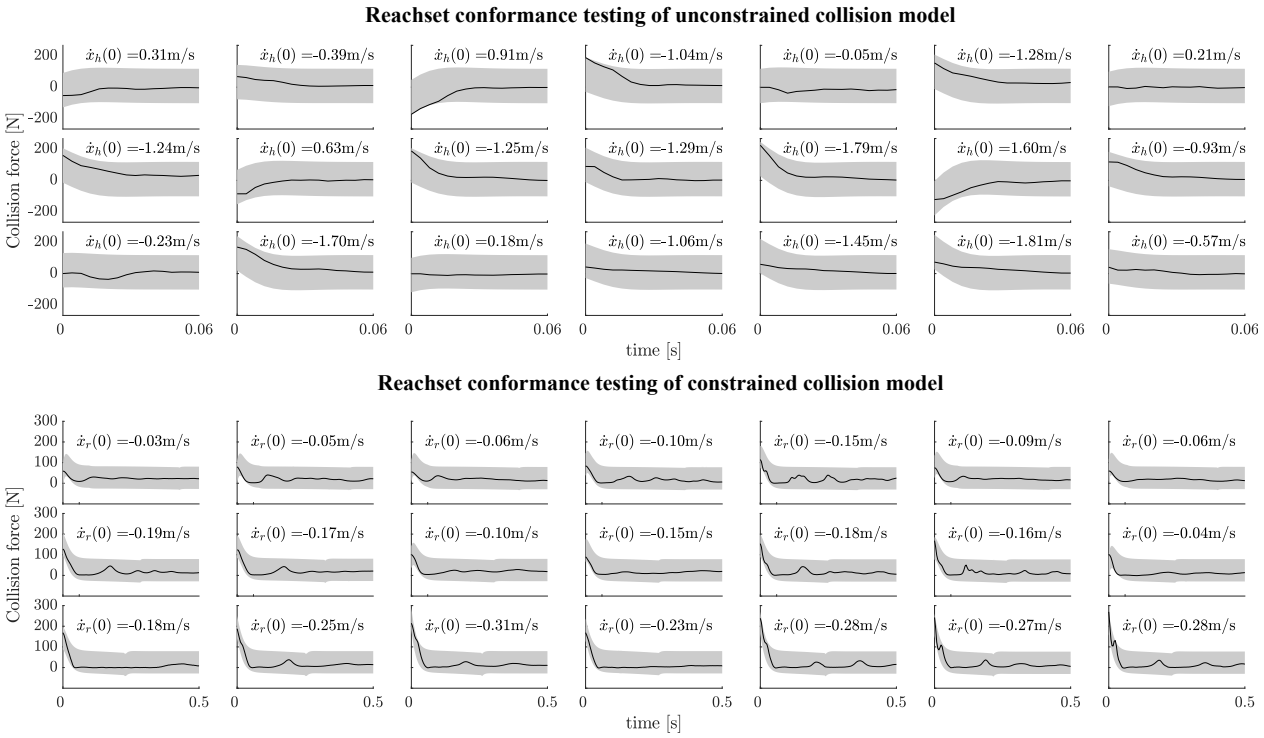


Fig. 3. *Reachset conformance testing of physical interaction models.* The model identified in Tab. I is reachset conformant. The reachable sets (gray) of of the models always over-approximate the force profiles (black) of real collision experiments. For each test case, the collision velocity is shown.

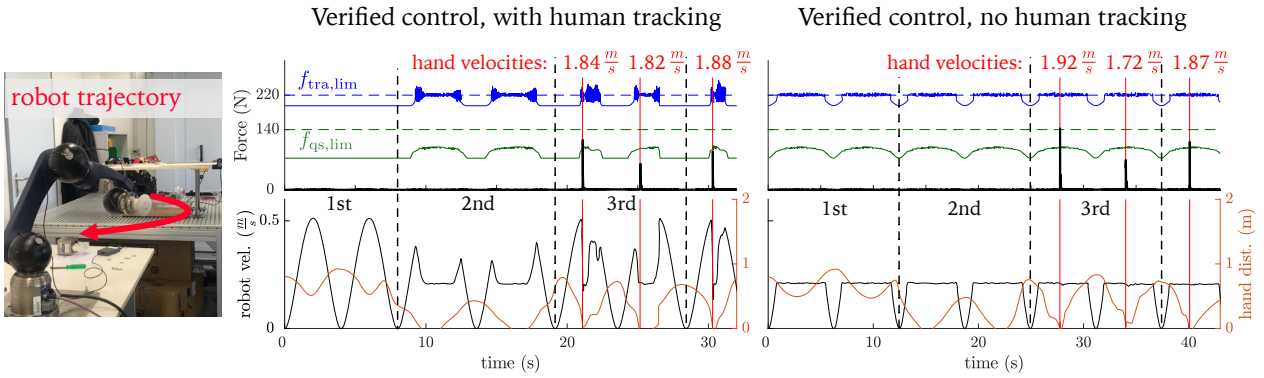


Fig. 4. *Verified impact-force-limiting control.* Upper graph: Transient force estimations are shown in green, quasi-static force estimations are shown in blue, force measurements are shown in **bold**, and collisions are shown in red. Lower graph: robot velocity is shown in black, and the relative distance of the hand to the robot is shown in orange All numbers are absolute values.

to 140 N. The resulting behavior of the robot is shown in Fig. 4. A video recording of the experiments is provided in supplementary materials.

For the scenario with tracking, we observe that in the first run, the robot can run at full speed, although the human-robot distance is closer than one meter. In the second run, we place our hand directly on the path of the robot. The robot automatically slows down because the reachable force predictions hit the transient force limit. However, as soon as the robot approaches a certain distance to the hand, it speeds up again. In the third run, we initiate collisions with the robot. The collision force never exceeds the estimated maximum transient.

For the scenario without human tracking, the results are similar, however, the robot remains at a slow speed for all times, and thus needs more time to complete its task. Human tracking benefits the efficiency of the robot.

## VI. Conclusions and Future Work

This study presents the first work on guaranteeing impact-force limits during possible unintentional collisions between the human hand and robot end-effectors, despite uncertain-

ties. We believe that this concept can be extended to the entire human body and robot arm using similar coupled interaction models. The primary innovation is the prediction of the impact forces using reachability analysis, combined with a fail-safe motion planning. Our model identification method ensures that the interaction model is reachset conformant with the real interaction. In an experiment, we demonstrated that the impact force limit criterion allows robot motion, even if humans work closely to the robot. In addition, we have shown the advantages of tracking humans, which allows the robot to move faster when humans are distant to the robot. Multiple extensions to this work are possible:

- To extend this approach to the entire human body, the identification experiments need to be repeated for every body part. Additionally, high-volume testing and experiments on more diverse human tissues are needed to make sure that edge cases of the model are covered.
- We have not regarded the fact, that the robot closed-loop dynamics can be uncertain. In this study, such uncertainties were lumped inside the sets $\mathcal{W}_{1,2}$ and $\mathcal{V}_{1,2}$. Thus, our approach is only applicable to the controller used in the identification experiments. To verify variable impedance controllers, the uncertain dynamics of the robot and the human should be separately identified, and the coupling between these should be created online to analyze interaction forces.
- Online verification can also be combined with any other safety metric, i.e., by exchanging predicate $\mathrm{safeForce}(t)$ with power, energy, or safe velocity limits.
- An interesting extension is the verification of continuous physical interaction, where the dynamics of the human arm are also usually modeled as impedances.

## ACKNOWLEDGMENT

The authors gratefully acknowledge partial financial support by the Central Innovation Programme of the German Federal Government under grants ZF4086004LP7, ZF4086012DB9, and the European Commission project CONCERT under grant number 101016007.

## REFERENCES

[1] ISO/TS 15066:2016, "Robots and robotic devices - collaborative robots," Int. Org. for Standardization, Geneva, Switzerland, 2016.

[2] K. N. Shivakumar, W. Elber, and W. Illg, "Prediction of impact force and duration due to low-velocity impact on circular composite laminates," *J. of Applied Mechanics*, vol. 52, no. 3, pp. 674–680, 1985.

[3] Y. Yamada, Y. Hirasawa, S. Huang, Y. Umetani, and K. Suita, "Human-robot contact in the safeguarding space," *IEEE/ASME Trans. on Mechatronics*, vol. 2, no. 4, pp. 230–236, 1997.

[4] K. Ikuta, H. Ishii, and M. Nokata, "Safety evaluation method of design and control for human-care robots," *Int. J. of Robotics Research*, vol. 22, no. 5, pp. 281–297, 2003.

[5] J. Heinzmann and A. Zelinsky, "Quantitative safety guarantees for physical human-robot interaction," *Int. J. of Robotics Research*, vol. 22, no. 7-8, pp. 479–504, 2003.

[6] B. Navarro, A. Cherubini, A. Fonte, R. Passama, G. Poisson, and P. Fraisse, "An ISO10218-compliant adaptive damping controller for safe physical human-robot interaction," in *Proc. of ICRA*, 2016, pp. 3043–3048.

[7] Z. J. Li, H. B. Wu, J. M. Yang, M. H. Wang, and J. H. Ye, "A position and torque switching control method for robot collision safety," *Int. J. of Automation and Computing*, vol. 15, no. 2, pp. 156–168, 2018.

[8] R. J. Kirschner, N. Mansfeld, S. Abdolshah, and S. Haddadin, "Experimental Analysis of Impact Forces in Constrained Collisions According to ISO / TS 15066," in *Proc. of ISR*, 2021.

[9] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "Requirements for safe robots: measurements, analysis and new insights," *Int. J. of Robotics Research*, vol. 28, no. 11-12, pp. 1507–1527, 2009.

[10] N. Mansfeld, M. Hamad, M. Becker, A. G. Marin, and S. Haddadin, "Safety map: A unified representation for biomechanics impact data and robot instantaneous dynamic properties," *IEEE Rob. Autom. Letters*, vol. 3, no. 3, pp. 1880–1887, 2018.

[11] A. Meguenani, V. Padois, J. Da Silva, A. Hoarau, and P. Bidaud, "Energy-based control for safe human-robot physical interaction," in *Int. Symp. on Exp. Robotics*. Cham: Springer, 2017, pp. 809–818.

[12] G. Raiola, C. A. Cardenas, T. S. Tadele, T. De Vries, and S. Stramigioli, "Development of a safety and energy aware impedance controller for collaborative robots," *IEEE Rob. Autom. Letters*, vol. 3, no. 2, pp. 1237–1244, 2018.

[13] M. Geravand, E. Shahriari, A. De Luca, and A. Peer, "Port-based modeling of human-robot collaboration towards safety-enhancing energy shaping control," in *Proc. of ICRA*, 2016, pp. 3075–3082.

[14] M. Angerer, S. Music, and S. Hirche, "Port-hamiltonian based control for human-robot team interaction," in *Proc. of ICRA*, 2017, pp. 2292–2299.

[15] S. Mitsch, K. Ghorbal, D. Vogelbacher, and A. Platzer, "Formal verification of obstacle avoidance and navigation of ground robots," *Int. J. of Robotics Research*, vol. 36, no. 12, pp. 1312–1340, 2017.

[16] S. Petti and T. Fraichard, "Safe motion planning in dynamic environments," in *Proc. of IROS*, 2005, pp. 2210–2215.

[17] M. Althoff, "An introduction to CORA 2015," in *Proc. of Workshop on Applied Verification for Cont. and Hybr. Systems*, 2015, pp. 120–151.

[18] D. Beckert, A. Pereira, and M. Althoff, "Online verification of multiple safety criteria for a robot trajectory," in *Proc. of IEEE Conf. on Decision and Control*, 2017, pp. 6454–6461.

[19] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira, "Effortless creation of safe robots from modules through self-programming and self-verification," *Science Robotics*, vol. 4, no. 31, 2019, eaaw1924.

[20] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger, "The role of the robot mass and velocity in physical human-robot interaction - Part I: Non-constrained blunt impacts," in *Proc. of ICRA*, 2008, pp. 1331–1338.

[21] C. Ott, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, ser. Springer Tracts in Advanced Robotics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 49.

[22] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the humanrobot collaboration," *Autonomous Robots*, vol. 42, no. 5, pp. 957–975, 2018.

[23] M. Althoff, G. Frehse, and A. Girard, "Set Propagation Techniques for Reachability Analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, 2021.

[24] A. Girard, C. Le Guernic, and O. Maler, "Efficient Computation of Reachable Sets of Linear Time-Invariant Systems with Inputs," in *Lecture Notes in Comp. Sci.*, 2006, vol. 3927 LNCS, pp. 257–271.

[25] S. B. Liu, B. Schrmann, and M. Althoff, "Reachability-based identification, analysis, and control synthesis of robot systems," 2021, arXiv:2103.01626.

[26] M. Althoff, S. Maierhofer, and C. Pek, "Provably-Correct and Comfortable Adaptive Cruise Control," *IEEE Trans. on Intelligent Vehicles*, vol. 6, no. 1, pp. 159–174, 2021.

[27] T. Kröger and F. Wahl, "Online trajectory generation: basic concepts for instantaneous reactions to unforeseen events," *IEEE Trans. on Robotics*, vol. 26, no. 1, pp. 94–111, 2010.

[28] ISO 13855:2010, "Safety of machinery positioning of safeguards with respect to the approach speeds of parts of the human body," Int. Org. for Standardization, Geneva, Switzerland, 2010.

## A.4 Velocity Estimation of Robot Manipulators: An Experimental Comparison [4]

**Summary** Precise controllers and safety algorithms depend upon accurately estimating the robot's velocity because most of today's robots are manufactured without including direct velocity sensing. Instead, they are mostly equipped with encoders for position measurements, which we need to derive the robot velocity. Many estimators exist in the literature to accomplish this, but a proper comparison between these methods has yet to be included. This work aims to close this gap by comprehensively comparing estimators on a real robot, focusing on the practicality of the velocity estimation methods. In addition, this work is a precursor to our subsequent work on formal controller synthesis in [5], for which we chose a velocity estimation method from this work. This paper is also a contrast to [5] in terms of tuning: the work in [5] uses formal synthesis to obtain optimal observers. In contrast, this work uses a classical tuning approach by testing parameter candidates directly on the robot and optimizing a classical tuning metric (integral squared error) using a genetic algorithm.

This paper compares velocity estimation methods using various trajectories: finite difference, moving average filter, derivative filter, Kalman filter, linear high-gain observer, nonlinear high-gain observer, and sliding-mode observer. To evaluate the methods, we look at the optimal estimation error, the closed-loop tracking error, convergence behavior, sensor fault tolerance, implementation, and tuning effort.

The experimental results show that the linear high-gain observer consistently displays the best accuracy when the gains are properly tuned. The nonlinear high-gain observer was challenging to adjust and could not reach a good performance despite our efforts to identify the robot model as accurately as possible. The sliding-mode observer and the Kalman filter showed good robustness to sensor errors, but the sliding-mode observer tends to lose convergence at high accelerations. From our experiments, we cannot conclude that model-based observers perform better than mode-free methods, despite including the robot model, which requires high identification efforts. When optimally tuned, the velocity estimation methods, except for the nonlinear high-gain observer, do not impact the tracking error.

**Author Contributions** **S. L.** performed the literature research and selected the estimation methods for the comparison. **S. L.** developed the automatic tuning method. **S. L.** and A. G. designed, conducted, and evaluated the experiments. **S. L.** and A. G. wrote the article. M. A. led the research project, provided feedback, and helped improve the manuscript.

**Journal article** The final version of the conference paper is reprinted in this thesis. The final version of the record is also available at `https://doi.org/10.1109/OJCSYS.2022.3222753`.

IEEE CSS

Received 29 June 2022; revised 27 September 2022; accepted 28 October 2022. Date of publication 16 November 2022; date of current version 7 December 2022. Recommended by Senior Editor Sonia Martinez.

*Digital Object Identifier 10.1109/OJCSYS.2022.3222753*

# Velocity Estimation of Robot Manipulators: An Experimental Comparison

**STEFAN B. LIU [1] (Member, IEEE), ANDREA GIUSTI [2] (Member, IEEE), AND MATTHIAS ALTHOFF [1] (Member, IEEE)**

[1]Department of Informatics, Technical University of Munich, 85748 Garching, Germany
[2]Fraunhofer Italia Research, 39100 Bolzano, Italy

CORRESPONDING AUTHOR: S. B. LIU (e-mail: stefan.liu@tum.de)

This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant 101016007 (Project CONCERT).

**ABSTRACT** Accurate velocity information is often essential to the control of robot manipulators, especially for precise tracking of fast trajectories. However, joint velocities are rarely directly measured and instead estimated to save costs. While many approaches have been proposed for the velocity estimation of robot joints, no comprehensive experimental evaluation exists, making it difficult to choose the appropriate method. This paper compares multiple estimation methods running on a six degrees-of-freedom manipulator. We evaluate: 1) the estimation error using a ground-truth signal, 2) the closed-loop tracking error, 3) convergence behavior, 4) sensor fault tolerance, 5) implementation and tuning effort. To ensure a fair comparison, we optimally tune the estimators using a genetic algorithm. All estimation methods have a similar estimation error and similar closed-loop tracking performance, except for the nonlinear high-gain observer, which is not accurate enough. Sliding-mode observers can provide a precise velocity estimation despite sensor faults.

**INDEX TERMS** Genetic algorithms, manipulators, robots, tuning, velocity estimation.

## I. INTRODUCTION

Accurate joint velocity signals of robot manipulators are needed for many fundamental control purposes, e.g., trajectory tracking, collision detection, and force control [1]. Sensors for measuring joint positions, e.g., encoders, have become inexpensive, reliable, and have a high resolution. The same cannot be said for velocity measurements. Direct measurements, e.g., through magnetic tachometers are affected by discontinuities of the magnetic field, ripple torques, and other high-frequency noise [2], while encoders are much more robust. Compactness and economic reasons often lead to not integrating joint velocity sensors at all.

Starting with the works of Nicosia and Tomei [3] in the 1990 s, velocity estimation for robots has been discussed widely in the literature, and many different methods have been proposed since then. From a practitioner's point of view, however, it is still hard to select a proper estimation method, because 1) it is hard to infer differences between estimation methods from previous papers, 2) many techniques have only been evaluated in simulation, and 3) the evaluations have been carried out on different robots.

Our paper addresses this issue by systematically comparing popular velocity estimation concepts and evaluating them using criteria which are important to practitioners, such as tuning and robustness to faults. Together with this paper, we also publish a MATLAB tool package, that includes an implementation of all discussed methods ready-to-use.

Previous studies that involve comparing velocity estimation methods can be found in [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. Simulative comparisons of derivative filters for discrete position measurements showed that no approach works best for all velocity profiles [4], [5], [6]. In the simulative comparison in [7], an extended Kalman filter, a nonlinear high-gain observer, and a linear observer have been compared considering their position estimation error and tracking error on a two degrees-of-freedom (DOF) robot. The authors in [8], [9] experimentally compare the tracking error of different tracking controllers using linear high-gain observers. The study in [10] experimentally analyzes the tracking error of a 2-DOF planar robot using five different observers. However, each observer uses a different tracking controller and only

trapezoidal trajectories were tested. An experimental comparison between a nonlinear high-gain observer, a Kalman filter, and a lead-lag based filter has been conducted on a parallel kinematics robot in [11], where the authors use a dual-mode controller and a proportional-derivative (PD) controller. The comparisons in [10] and [11] have two main drawbacks: 1) the gains of the proposed methods were chosen without apparent justification, although the performance of velocity estimation mainly depends on such gains, and 2) these comparisons did not evaluate the performance in terms of velocity estimation error. Our paper addresses these issues by including a gain tuning approach that allows a comparison of optimally tuned estimators, while the velocity estimation error is measured through an actual ground-truth signal. Further works that compare model-free and model-based observers, used in conjunction with different controller structures, can be found in [12], [13], [14].

In contrast to previous works, our paper presents for the first time an experimental comparison of a wide variety of estimation methods, including multiple filters and multiple observers. We use the same 6-DOF robot manipulator, the same tracking controller, and the same test trajectory for all estimators.

This paper is organized as follows: in Section I, we state the problem at hand and survey the literature for popular existing velocity estimators. In Section II, we provide an in-depth review of selected estimators, which we consider to be among the most suitable for practical applications. The automatic parameter tuning is explained in Section III. The evaluated estimators are compared experimentally in Section IV and we conclude the paper in Section V.

### A. PROBLEM STATEMENT

Let us consider the rigid dynamics of robot manipulators with $n$ revolute joints written in state-space form as

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= f(x_1, x_2, u) = M^{-1}(x_1)(u - n(x_1, x_2)), \end{aligned} \tag{1}$$

where $x_1 \in \mathbb{R}^n$ is the vector of joint positions, $x_2 \in \mathbb{R}^n$ is the vector of joint velocities, $u \in \mathbb{R}^n$ is the vector of motor torques, $M(x_1) \in \mathbb{R}^{n\times n}$ is the inertia matrix, and $n(x_1, x_2) \in \mathbb{R}^n$ is the vector-valued function including Coriolis and centripetal forces, gravity, and friction. The joint positions are measured at a finite resolution using rotational encoders.

The robot tracks a desired trajectory of positions, velocities, and accelerations $x_1^d(t), x_2^d(t), x_3^d(t) \in \mathbb{R}^n$ via an inverse dynamics controller [15, Sec. 8.5.2]

$$u = M(\hat{x}_1)v + n(\hat{x}_1, \hat{x}_2), \tag{2}$$

$$v = x_3^d + K_p(\hat{x}_1 - x_1^d) + K_d(\hat{x}_2 - x_2^d), \tag{3}$$

where $\hat{x}_1, \hat{x}_2$ are the vectors of estimated joint positions and velocities. Some of the discussed methods do not estimate $\hat{x}_1$; but since $x_1$ is measured directly, $\hat{x}_1$ can be replaced by $x_1$ in (2) and (3), when applicable.

**TABLE 1. Velocity estimation methods identified in this survey (references with * are evaluated in our comparison).**

| | Validated in simulation | Validated in experiments |
|---|---|---|
| model-free | [5]*, [12]–[14], [28]–[30], [32]–[34], [37]* | [8]*, [26], [27], [31] |
| model-based | [2], [12]–[14], [18], [19], [20]*, [22], [24], [25]* | [17], [21], [23] |

The objective of this paper is to compare different methods to obtain $\hat{x}_2$ and to tune the parameters of all estimators, such that the error $x_2 - \hat{x}_2$ between the estimated and the ground-truth velocity is minimized. To obtain a ground-truth signal, any method can be used that returns a significantly more accurate velocity than the evaluated estimations, e.g., using external encoders with a higher resolution and sampling rate. In our paper, we simulate external measurement by artificially decreasing the sensor resolution and sampling rate of the internal sensors for closed-loop control, while the ground truth is obtained using the actual sensor resolution at a higher sampling rate.

The estimation methods are subject to disturbances in our robot system. Amongst others, there can be

- *quantization errors* due to the finite resolution of the encoders;
- *high-frequency noises* due to manufacturing errors of the encoders [16];
- *modeling errors* due to an inaccurate parametrization of $f(x_1, x_2, u)$;
- *sensor faults* due to communication errors.

In the subsequent literature survey we group the approaches we identified for velocity estimation into *model-based* approaches, that require the computation of the nonlinear dynamical model in (1), and *model-free* approaches, which do not need this model. Model-free methods can be implemented decentrally at each individual joint, if the methods do not have dependencies between joints. Model-based methods, however, must be implemented in a centralized manner. The considered approaches are collected in Table 1, which also sorts them according to the fact that they are validated in the literature using simulations or experiments.

### B. MODEL-BASED METHODS

We first survey model-based schemes. The popular and pioneering model-based method of Nicosia and Tomei in [3] presents an asymptotically stable observer whose region of attraction can be enlarged via the observer gain. In contrast to previous work, the authors design the model-based observer in conjunction with a controller; many subsequent works followed this idea. The authors in [17] propose a model-based observer which provides semi-global exponentially stable error dynamics of the velocity tracking error, considering a dedicated controller structure. Effectiveness of this approach is shown by experiments on a 2-DOF manipulator. Another model-based extension of the approach from Nicosia and

Tomei can be found in [18], in which the authors show semi-global exponential stability of their proposed combined observer and controller. The authors in [19] and [20] discuss nonlinear high-gain observers for the velocity estimation problem, including how to avoid the peaking phenomenon in the transient behavior, i.e., the initial estimation error may exhibit an impulse that could destabilize the controller. An adaptive approach providing locally asymptotically stable estimation error dynamics has been proposed in [21] in which the authors show that their proposed approach is superior to simple numerical differentiation; the authors perform experiments on a 6-DOF PUMA-560 robot. The work in [22] introduces a combined observer/controller structure providing global exponential convergence of the estimation error. However, that paper only shows practical effectiveness by means of simulations. More recently, the author in [23] showed theoretically that a proposed Luenberger-like observer with a simple proportional-derivative control with gravity compensation achieves uniformly ultimately bounded stability, which is confirmed by experiments on a 2-DOF robot. Model-based approaches using sliding mode observers have been proposed for robots in [2], [24], [25], whose effectiveness was only demonstrated by simulations.

### C. MODEL-FREE METHODS

In this subsection, we survey model-free approaches. In the works of Nicosia et al. [8], a simple high-gain observer is introduced, which supports distributed implementations; this approach also provides uniformly ultimate boundedness of the velocity estimate and is presented and tested using both simulations and experiments on a 6-DOF robot. The work in [26] introduces a model-free observer providing uniformly ultimate boundedness of the velocity estimation error. This scheme accounts for model uncertainties in its design and its effectiveness have been verified by means of experiments with a 2-DOF robot. Both [8] and [26] consider inverse dynamics control and proportional-derivative control for tracking, provide a closed-loop stability analysis for both cases, and suggest parameters to ease gain tuning of the observer. Subsequently, the authors in [27] also proposed a model-free observer that provides uniformly ultimate boundedness of both tracking and observer errors when used in conjunction with their proposed robust controller. The authors in [28], [29] introduce a model-free observer which provides asymptotic stability of the velocity estimation error dynamics. To achieve this, [28] uses passivity arguments, while more general Lyapunov arguments are used in [29], where also external disturbance is taken into account and the performance is shown using simulations on a 2-DOF robot. Similarly, the works in [30], [31] present model-free observers that provide asymptotic stability demonstrated in simulation [30] and experiments [31]. A model-free sliding-mode observer has been proposed in [32]; its practical effectiveness has been presented by simulations. As an extension, some estimators incorporate neural networks [33], [34].

Furthermore, there exist popular estimators without explicit closed-loop stability proofs. Kalman filters [35] are such an example, which assume white noise to approximate the robot dynamics. Also, the derivative filtering methods, such as the ones in [5], are not yet proven to be stable in closed-loop. However, the author of [36] introduces a possible theoretical framework to foster the use of derivative filtering in place of state observers for a stable output-feedback control of robots.

From the available literature, we select several estimation methods, of which we conduct an in-depth review, which can be divided into four model-free methods from the works in [5], [8], [37], and two model-based methods from the works in [20], [25]. The selected methods have an asterisk in Table 1. These have been mainly selected for their popularity, ease of implementation, ease of tuning, and their robustness with respect to the chosen controller.

## II. REVIEW OF SELECTED ESTIMATORS

In this section, we discuss the estimators that we experimentally compare, namely moving average filtering [5], derivative filtering [5], Kalman filtering [37], linear high-gain observer [8], nonlinear high-gain observer [19], and sliding-mode observer [25]. We review their respective properties as studied in the literature. Furthermore, we discuss the implementation aspects.

### A. FINITE DIFFERENCE AND MOVING AVERAGE FILTERING

This basic technique numerically approximates the derivative by dividing the difference between successively obtained position measurements by a time window $p\Delta t$, where $\Delta t$ is the sampling time of the controller and $p$ is an integer that determines the size of the window for which we take the average. We denote a position measurement as $x_{1,k-1} = x_1((k-1)\Delta t)$. The estimated velocity is given by

$$\hat{x}_{2,k} = \frac{x_{1,k} - x_{1,k-p}}{p\Delta t}. \tag{4}$$

With large $p$, the averaging effect attenuates quantization noise in the measurements, but introduces a delay in the estimated velocity, while small $p$ values amplify the noise [37]. For our comparison, we use $p = 1$, which we also call the finite difference (*FinDiff*) method, and an optimally chosen $p > 1$, which we call the moving average (*MovAv*) method.

### B. DERIVATIVE FILTERING

Here, we describe a class of methods that compute the derivative through filtering the position signal. Various predictive strategies have been proposed in the literature based on a polynomial fitting of previous measurements, such as Taylor series expansion (TSE), and backward difference expansion (BDE), which are characterized by the number of samples $n_{\mathrm{TSE}}$ and $n_{\mathrm{BDE}}$. To counter the problem of overfitting and the resulting noise amplification, the least-squares fit (LSF) has been proposed in [5], that uses regression to find a polynomial of the order $p_{\mathrm{LSF}}$ with the smallest error among $n_{\mathrm{LSF}}$ measurements, where $p_{\mathrm{LSF}} < n_{\mathrm{LSF}}$. In-depth comparisons of these methods

and a description of their implementation can be found in [4], [5], [38]. The findings of Brown et al. [5] are that TSE and BDE are good for transient responses and LSF filters are more suited for constant velocities. For velocity profiles that vary a lot, such as for robot manipulators, no single filtering method is best [38]. In our comparison, we will first evaluate the TSE, BDE, and LSF filters amongst each other, and choose the best one for the overall comparison with other methods.

### C. KALMAN FILTER

The Kalman filter is a linear observer, that has been used in many engineering fields, such as for state and parameter estimation, data merging, or signal processing [39]. Bélanger proposes such an observer for rotary encoders [37], and instead of using the dynamical model in (1), assumes a triple integrator model for each individual axis $i$, consisting of the state $z_i = [x_{1,i}, x_{2,i}, x_{3,i}] \in \mathbb{R}^3$ (position, velocity, and acceleration of each axis), the output $y_i$, and the Gaussian white noises $v_i$ (sensor noise) and $w_i$ (process noise) [37, Eq. 14]:

$$\begin{aligned}
\dot{z}_i &= Az_i + \Gamma w_i \\
y_i &= Cz_i + v_i, \\
A &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \Gamma = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \\
C &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

According to [37], $v_i$ can be chosen to be zero-mean with variance $\frac{\Delta q_{m,i}^2}{3}$, where $\Delta q_{m,i}$ is the quantization error of each axis. The noise $w_i$ is also assumed to be zero-mean and has a variance $Q_i$, which has to be tuned for each axis. In the same work, a second-order system is additionally proposed for velocity estimation which, however, does not perform as well as the third-order one. The analysis in [37] showed an improvement compared to the finite difference method, especially at low speeds up to one tenth of an encoder increment per time step.

To further improve the acceleration estimation [7] or to provide estimations for flexible robots [40], one could estimate the state of the full model of the robot considering the nonlinear dependencies between joints. For those systems, extended Kalman filters are required due to the nonlinearity of the system. Since both cases are not relevant in our application, we deliberately exclude this method in our comparison.

### D. LINEAR HIGH-GAIN OBSERVER

High-gain observers are theoretically well understood (see, e.g., the works of Khalil [20]) and have been experimentally examined, e.g., in [7], [8], [9], [10], [11]. In this work, we discuss both the linear and the nonlinear versions. The linear observer (*linHG*) uses a scalar gain $\epsilon_l$ and two matrix gains $H_1, H_2 \in \mathbb{R}^{n\times n}$ [8, 9]:

$$\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 + \frac{1}{\epsilon_l} H_1 (x_1 - \hat{x}_1), \\
\dot{\hat{x}}_2 &= \frac{1}{\epsilon_l^2} H_2 (x_1 - \hat{x}_1).
\end{aligned}$$

This observer is asymptotically stable if the eigenvalues of $\begin{bmatrix} -H_1 & I \\ -H_2 & 0 \end{bmatrix}$ have negative real parts [20]. It has been shown in [8], that there exists an $\epsilon_l^*$ so that the closed-loop dynamics is asymptotically stable for $\epsilon_l \in [0, \epsilon_l^*]$, for any uniformly asymptotically stable controller. In other words, high-gain observers can be flexibly combined with any tracking controller, while overall stability is guaranteed.

In practice, however, the observer gains are limited, i.e., $\epsilon_l$ is lower-bounded by measurement noise and the sampling time of the controller [20]. Therefore, a trade-off between the noise suppression and estimation accuracy has to be found. To partially overcome this compromise, one can filter measurements and implement time-varying gains, as discussed in [20]; this extension is excluded in our comparison since we limit ourselves to easily implementable approaches. Also, a peaking phenomenon occurs (not examined by [8]). For these cases, an input saturation is sufficient for stability [20].

### E. NONLINEAR HIGH-GAIN OBSERVER

Considering the full robot model (1) as an additional source of information, there exists a potential for better results using nonlinear observers. The nonlinear high-gain observer (*nnlHG*) is such a model-based approach, that is similar to its above-discussed linear version. For robots, this observer has first been introduced by Lee and Khalil in [19]. The observer uses the scalar gain $\epsilon_n$ and two matrix gains $L_1, L_2 \in \mathbb{R}^{n\times n}$ [20, 9.4]:

$$\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 + \frac{1}{\epsilon_n} L_1 (x_1 - \hat{x}_1), \\
\dot{\hat{x}}_2 &= f(x_1, \hat{x}_2, u) + \frac{1}{\epsilon_n^2} L_2 (x_1 - \hat{x_1}).
\end{aligned}$$

Similar to the linear version, asymptotic stability is given if the real part of the eigenvalues of $\begin{bmatrix} -L_1 & I \\ -L_2 & 0 \end{bmatrix}$ are negative and a nonlinear separation principle can be established for the stability of the closed-loop system [41], meaning that also this observer can be flexibly combined with any stable tracking controller. A simulation study in [20] has shown that, indeed, a better velocity estimation compared to the linear version can be achieved, if the model is precise. However, this advantage becomes less and less significant, when the gains $\epsilon_l$ and $\epsilon_n$ decrease [20].

### F. SLIDING MODE OBSERVER

Robustness, finite-time convergence, and the ability to handle discontinuous systems are major reasons for the application of sliding mode observers (*SliMod*) [42]. Real implementations of sliding mode observers, however, suffer from chattering

while sliding along the switching surfaces. This effect can be alleviated by second or [42] or third-order [43] sliding-mode observers. Third-order versions experience a slower convergence than second-order observers, as shown by Fraguela Cuesta et al. in [43]. The version we consider in our comparison is the third-order version proposed in [25], which adds a linear term to improve convergence. It consists of the gain vectors $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^{n\times 1}$, the linear gain matrices $K_1, K_2 \in \mathbb{R}^{n\times n}$, the signum function $sgn(\cdot)$, the element-wise absolute values $|\cdot|$, and element-wise powers, such that [25, Eq. 28]:

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 + \alpha_3|x_1-\hat{x}_1|^{2/3}sgn(x_1-\hat{x}_1) + K_1(x_1-\hat{x}_1),\\ \dot{\hat{x}}_2 &= \hat{f}(x_1,\hat{x}_2,u) + \alpha_2|\dot{\hat{x}}_1-\hat{x}_2|^{1/2}sgn(\dot{\hat{x}}_1-\hat{x}_2),\\ &\quad + K_2(x_1-\hat{x}_1) + \hat{z}_{eq},\\ \dot{\hat{z}}_{eq} &= \alpha_1 sgn(\dot{\hat{x}}_1-\hat{x}_2),\end{aligned}$$

where $\hat{z}_{eq}$ is the observed input disturbance, which could also be used to improve the tracking performance, as described in [25]. For continuous-time systems, this observer has finite-time convergence, and can thus be trivially combined with stable tracking controllers, since the observer only has to reach the exact velocity before the controlled system would leave the stability bounds [42].

### G. IMPLEMENTATION

Except for derivative filtering, the above estimation methods and their properties have been developed and presented in the literature assuming continuous-time control. Real implementations, however, are usually in discrete time, which is why we briefly review their implementation here.

Discrete-time versions of Kalman filters can be obtained by transforming the system model to discrete time and solving the discrete Riccati equation. As an example, the MATLAB functions `c2d`, `dlqe`, and `destim` provide the respective functionality. Discrete-time implementation of both linear and nonlinear high-gain observers are reviewed in [20, Ch. 9], and boundedness of the estimation error has been shown. For the linear version, the bilinear transformation performs best, as shown in [20], and can also be formulated as an FIR filter [44]. For both nonlinear high-gain and sliding-mode observers, the forward difference transformation can be used, for which boundedness of the estimation error has been shown in [20] and [42].

## III. GAIN TUNING USING A GENETIC ALGORITHM

Control gain tuning is one of the main concerns in industrial applications [45]. For a fair comparison, one has to find the optimal gains for each estimator—some of them feature up to 90 gains, when every matrix element is considered (see Table 2). Manually tuning the gains is a time-consuming task for some estimation methods. Instead, we propose to use an automatic approach to find the optimal gains, which can be applied to all estimators.

**TABLE 2. Number of gains of velocity estimators for 6-DOF robots.**

| Method | Gains | Total # | Reduced # |
|---|---|---|---|
| Moving Average | $n$ | 6 | 6 |
| BDE, TSE | $n_{\mathrm{BDE,TSE}}$ | 6 | 6 |
| LSF | $n_{\mathrm{LSF}}, p_{\mathrm{LSF}}$ | 12 | 12 |
| Kalman filter | $Q$ | 6 | 6 |
| Linear high-gain | $\epsilon_l, H_1, H_2$ | 73 | 12 |
| Nonlinear high-gain | $\epsilon_n, L_1, L_2$ | 73 | 12 |
| Sliding Mode | $\alpha_1, \alpha_2, \alpha_3, K_1, K_2$ | 90 | 8 |

Possible automatic tuning techniques for PID controllers are reviewed in [46]; however, these are not applicable to multi-input multi-output systems. Instead, genetic algorithms (GA) have shown promising results for the gain tuning of nonlinear controllers, as demonstrated by simulations in [47] for flight controllers and in [48], [49], [50] for robot controllers. Genetic algorithms are bio-inspired techniques, for which existing tools can be used, e.g., the Global Optimization Toolbox in MATLAB.

To accelerate the tuning process, we reduce the number of gains. For high-gain observers, the original authors propose to choose $H_1, H_2, L_1, L_2$ a priori and subsequently decrease $\epsilon_l, \epsilon_n$ as far as possible. As can be seen in the equations of Section II-D and Section II-E, the $\epsilon$ gains are, however, redundant, since one can equally choose large values for the matrices. This is why we arbitrarily set $\epsilon_l = \epsilon_n = 0.03$ a priori and tune $H_1, H_2, L_1, L_2$ instead. Additionally, we only consider to tune their diagonals to reduce the number of gains. For the sliding-mode observer, we choose $\alpha_1 = 1.1f^+$, $\alpha_2 = 1.5(f^+)^{1/2}$, and $\alpha_3 = 1.9(f^+)^{1/3}$, as proposed in [42], [43], where $f^+ \in \mathbb{R}^6$ represents the upper bound of the model perturbation. Furthermore, we replace $K_1$ and $K_2$ by the scalars $k_1$ and $k_2$, respectively. We found these choices to be suitable as a compromise between optimality and decreased tuning effort.

The cost function we use in this paper for tuning, as well as for evaluating the performance of the estimators is the integral squared error (ISE) of the velocity estimation

$$\mathrm{ISE}(t) = \int_0^t (x_2(\tau) - \hat{x}_2(\tau))^2 d\tau. \tag{5}$$

This cost function depends on a measurement of the ground truth of $x_2$, which must be more accurate than the estimation from the reviewed methods. In our case, we run the estimation methods online at a lower sampling rate with a lower encoder resolution, while the ground truth position is measured by the same encoders at a higher sampling rate and resolution. The ground truth velocity is then obtained offline by computing the finite difference and downsampling it with an anti-aliasing filter [51] to match the sampling rate of the online estimation methods. If measuring at a higher sampling rate and resolution is not feasible, we propose to use offline zero-phase filtering [51] to obtain a ground truth, so that we can minimize the phase delay of the estimated velocity.

The hyperparameters of the genetic algorithm are chosen to be almost the same as in [49] (see Table 3), except for the number of generations. Although our tuning is carried out

**TABLE 3. Hyperparameters of GA for gain tuning.**

| Parameter | Value |
|---|---|
| Population size | 60 |
| Number of elites | 6 |
| Selection | Tournament |
| Crossover | Uniform crossover at 87.5% |
| Mutation | Random mutation at 0.38% |
| Maximum generation | 20 |

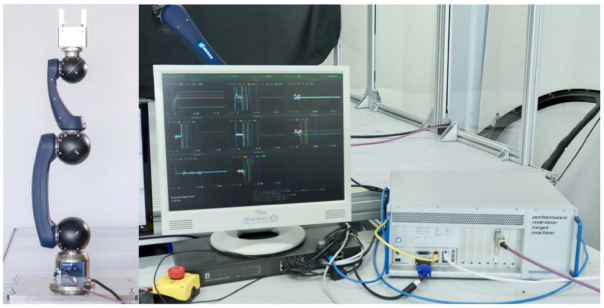

**FIGURE 1. The testbed consists of a 6-DOF robot manipulator and a controller running on Simulink Real-Time.**

on a real robot instead of simulations, we determined that 20 generations are sufficient for the gains to converge to an optimal value.

## IV. EXPERIMENTAL COMPARISON

In this section, we experimentally compare the velocity estimation methods reviewed in Section II. Our testbed consists of a 6-DOF Schunk LWA-4P robot, whose model has been identified in [52]. The controller and estimators are implemented in Simulink Real-Time on a target machine with an i7-3770 K 3.5 GHz processor (see Fig. 1). For the computed-torque controller, we choose $K_p = 100$ and $K_d = 13$. To measure the ground-truth velocity, we run the position encoder at 1 millidegree per increment at a sampling rate of 250 Hz. The actual velocity estimation is done at a resolution of 10 millidegrees per increment and at a sampling rate of 125 Hz.

We structure the experimental comparison as follows: in Section IV-A, we compare the tuning process using our proposed genetic algorithm. In Section IV-B, we show the main performance results, including the estimation error, the tracking error, and the convergence behavior of each estimator. Afterwards in Section IV-C, we compare the performance, when sensor faults are introduced. In Section IV-D, we compare the performance when using different encoder resolutions or sampling rates. The experimental comparison is concluded with a discussion of the results in Section IV-E. For simplicity in some of the plots, we only show the behavior of one axis because the behavior of the other axes are similar. The implemented velocity estimation methods, the experimental results, trajectories, and the robot model are provided as supplementary data[1] to this paper.

[1] [Online]. Available: https://dx.doi.org/10.21227/tse3-h285

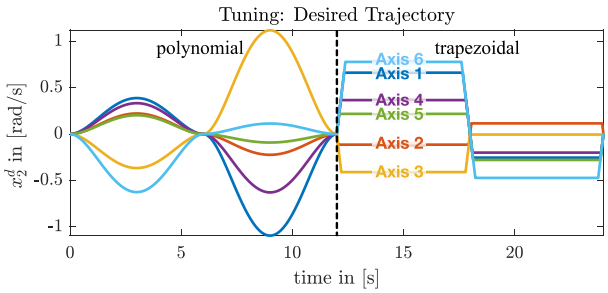


**FIGURE 2. Desired trajectory for gain tuning.**

**TABLE 4. Optimal gains of velocity estimation methods (10 millidegrees, 125 Hz).**

| Method | Par. | Optimal values (diag. elements of matrices) | | | | | |
|---|---|---|---|---|---|---|---|
| MovAv | $n$ | 2 | | | | | |
| LSF | $n$ | 4 | | | | | |
| | $p$ | 1 | | | | | |
| Kalman | $Q$ | 18.55 | 17.93 | 0.010 | 14.62 | 0.037 | 11.48 |
| linHG | $\epsilon_l$ | 0.03 | | | | | |
| | $H_1$ | 47.01 | 25.98 | 59.80 | 54.96 | 35.83 | 34.25 |
| | $H_2$ | 233.1 | 97.74 | 243.9 | 192.3 | 93.03 | 206.9 |
| nonHG | $\epsilon_n$ | 0.03 | | | | | |
| | $L_1$ | 1.090 | 0.818 | 1.155 | 1.172 | 1.196 | 1.191 |
| | $L_2$ | 1.812 | 0.999 | 2.205 | 2.428 | 2.006 | 2.552 |
| SliMod | $f^+$ | 17.19 | 12.20 | 30.38 | 32.24 | 28.38 | 34.77 |
| | $k_1$ | 84.46 | | | | | |
| | $k_2$ | 41.03 | | | | | |

### A. TUNING BEHAVIOR

We apply our automatic tuning procedure described in Section III to the four observers: Kalman filter, linear high-gain observer, nonlinear high-gain observer, and the sliding-mode observer. The remaining estimation methods only involve integer gains, which is why a grid search for each robot axis was sufficient. For tuning, we execute the trajectory displayed in Fig. 2 for each genome. With 20 generations, each with a population of 60 genomes, this translates to roughly 12 hours of tuning per estimation method, including the computation time.

In Fig. 3, we show how fast our genetic algorithm converges. The model-free observers (Kalman filter and linear high-gain observer) converge fast, while the model-based observers converge more slowly. According to our intuition, this may be because the gains of the model-based observers are more dependent on each other, where varying one gain affects the estimation performance of multiple joints. For the model-free observers, the gains are decoupled for each joint, which makes the search easier. The resulting optimal gains are shown in Table 4.

### B. ESTIMATION PERFORMANCE

We compare the performance of the velocity estimation methods using a more varied trajectory than the tuning trajectory. As shown in Fig. 4, it consists of a sine wave, a point-to-point trajectory in joint space using 5th-order polynomials,

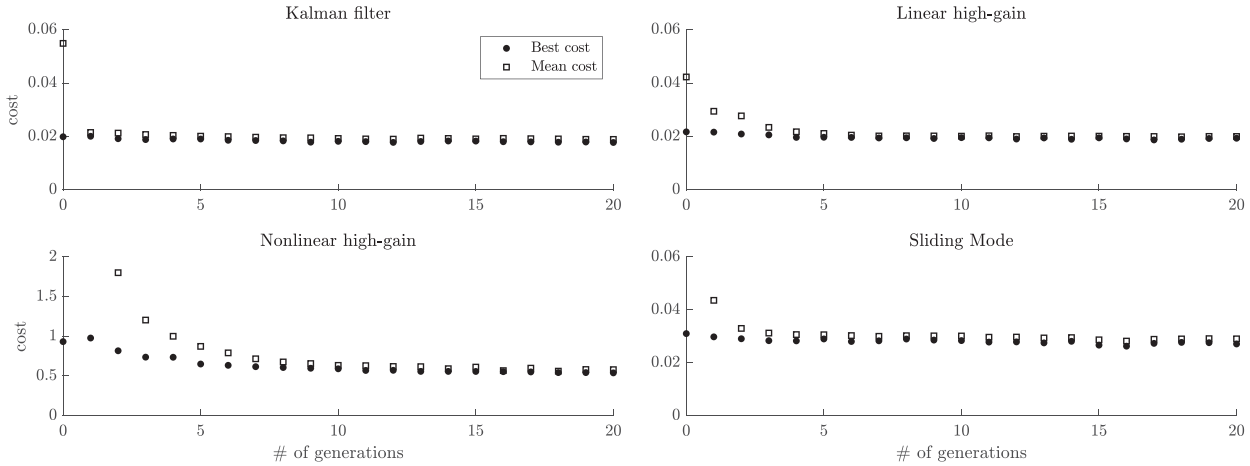


**FIGURE 3.** Gain tuning using our proposed genetic algorithm. After each generation the mean cost (5) slowly approaches the best achieved cost per generation.

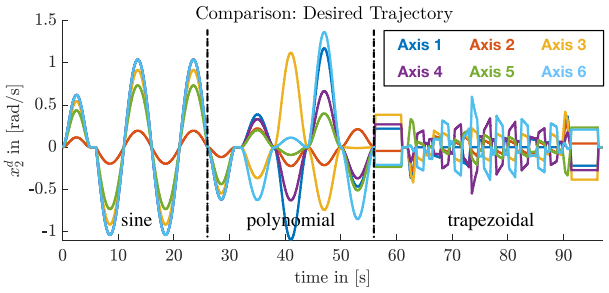


**FIGURE 4.** *Estimation performance:* desired trajectory. The velocity signal $\dot{q}_d$ consists of a sine part, a polynomial part, and a trapezoidal part.

**TABLE 5.** Integral squared error of the derivative filters (10 millidegrees, 125 Hz).

| Filter | ISE | Filter | ISE |
|---|---|---|---|
| MovAv | 0.035 | LSF3/8 | 0.434 |
| LSF1/4 | 0.046 | BDE2 | 0.092 |
| LSF1/8 | 0.114 | BDE3 | 0.184 |
| LSF2/8 | 1.197 | TSE3 | 0.128 |

and a trapezoidal point-to-point trajectory in both joint and task space with inverse kinematics included. The experiments are performed in closed-loop control, meaning that the estimated velocities are directly applied to the computed-torque controller.

At first, we choose the best derivative filter out of the TSE, BDE, and LSF filters. Table 5 shows the ISE metric for the test trajectory for all considered filters. These are the same ones that have been analyzed in the previous comparisons in [4] and [5]. For this experiment (and all subsequent ones) we have determined that the LSF1/4-filter (i.e., $n_{\mathrm{BDE}} = 2$) has the smallest velocity estimation error. Mathematically, the LSF1/3-filter (i.e., $p_{\mathrm{BDE}} = 1$ and $n_{\mathrm{LSF}} = 3$) equals the moving average filter for $n = 2$, and LSF1/2, BDE1, and TSE1 equal the finite difference method, which is why we exclude them in this comparison.

Next, we compare the LSF1/4-filter with all other optimally tuned estimation methods. To reflect the fact that the estimation performance can vary over time when used in closed-loop, we run the test trajectory four times for each estimator. In Fig. 5(b) we show the mean cumulative ISE over the course of the test trajectory, as well as their maxima and minima (shaded areas).

Except for the nonlinear high-gain observer, all other methods have a very similar estimation error. By small margins we can see that the Kalman filter and the linear high-gain observer perform slightly better than the rest. Although the finite difference method performs well in terms of ISE, we can also see in Fig. 5(a) that it is a noisy estimation due to the quantization error of the position encoder. On the one hand, the moving average filter improves the smoothness, but on the other hand, the error is larger due to the increased delay. Except for nnlHG, the other estimation signals are less noisy than the finDiff, while having a smaller delay than MovAv, which results in smaller estimation errors. The sliding-mode observer behaves interestingly: for the smooth sine and polynomial trajectories, it is an accurate estimation method. However, in the trapezoidal section, the error increases faster than for other methods, especially at the sections with sudden high acceleration.

In terms of the performance of the tracking control, the estimation methods do not differ significantly. As the overlapping shaded areas in Fig. 6 show for the ISE of the tracking error, the variation between multiple tests is far larger than the influence of the estimation method. Only nnlHG has a worse tracking error, resulting from its poor velocity estimation performance.

To explain the different behaviors of the methods, we analyse how fast they converge by analyzing their step responses. To do that, with reference to the result in Fig. 7, we execute

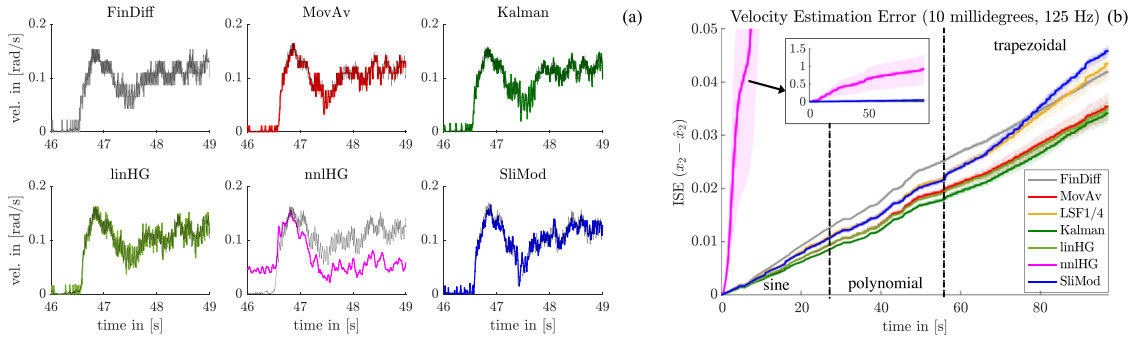


**FIGURE 5.** ***Estimation performance:*** **velocity error. The left side (a) shows an excerpt of the estimated velocity $\hat{x}_2$ (colored) versus the ground truth $x_2$ (black) for axis 6. The right side (b) shows the cumulative mean integral squared error (ISE) for each method (four experiments each).**

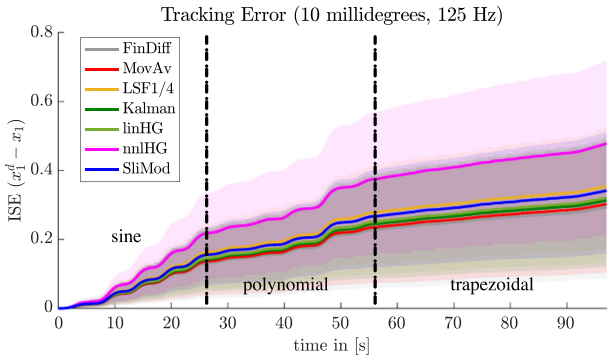


**FIGURE 6.** ***Estimation performance:*** **tracking error. Cumulative ISE of the tracking performance $x_1^d - x_1$.**

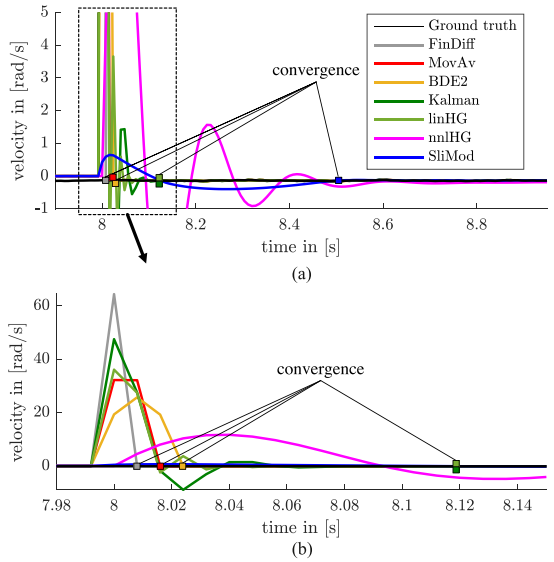


**FIGURE 7.** ***Estimation performance:*** **convergence of each method. Both plots show at different time scales the step response and the time of convergence (squares) of each estimator compared to the ground truth $x_2$ for axis 2 of the robot manipulator.**

**TABLE 6. Estimation error (ISE) with faulty sensors.**

| | normal | retuned normal | faulty | retuned faulty |
|---|---|---|---|---|
| **FinDiff** | 0.041 | 0.041 | 2.737 | 2.737 |
| **MovAv** | 0.033 | 0.097 | 0.763 | 0.160 |
| **LSF** | 0.041 | 0.088 | 0.526 | 0.207 |
| **Kalman** | 0.032 | 0.036 | 1.162 | 0.177 |
| **linHG** | 0.033 | 0.060 | 1.831 | 0.145 |
| **nonHG** | 0.508 | 2.011 | 1.312 | 2.017 |
| **SliMod** | 0.047 | 0.049 | 0.067 | 0.074 |

a trapezoidal trajectory and activate the estimators simultaneously when the reference velocity is constant (t = 8 seconds) to observe their response. We can see that the nonlinear high-gain observer never really converges, since it is not fast enough. The sliding-mode observer has the smallest overshoot, but requires much longer than the rest of the estimators to converge to the actual velocity. This is why in cases such as high accelerations in trapezoidal trajectories, the sliding-mode observer deviates, and the slow re-convergence accumulates to a large estimation error, although otherwise it is an accurate observer. The other methods converge significantly faster, which explains their good closed-loop performance.

### C. FAULT TOLERANCE

We analyze how the estimation methods react to errors in the position measurement. To do that, we randomly simulate a loss of communication for 10% of the measurements of axis 6 of our robot. The resulting velocity estimation can be seen in Fig. 8(a). The estimated velocities experience severe chattering, except for the sliding mode observer, which responds more robustly by remaining smoother.

However, the robustness of the estimators can be improved. To demonstrate that, we repeat the tuning process, in which the sensor stays faulty. As Fig. 8(b) shows, the estimation improves. As Table 6 shows, the retuned estimators significantly sacrifice accuracy during normal operation, except for the Kalman filter and the sliding-mode observer, which is why we conclude that these two are the most fault tolerant methods regarding our error model.

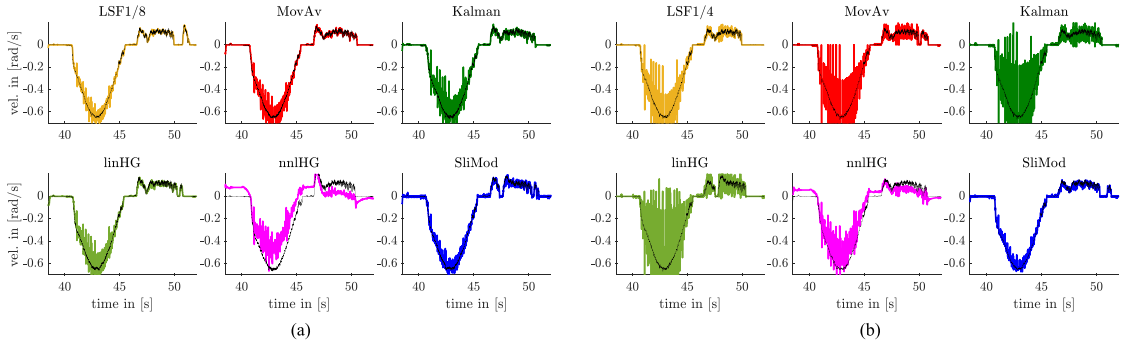


**FIGURE 8.** ***Sensor fault tolerance:*** **the left side (a) shows an excerpt of the estimated velocity of axis 6 given sensor errors, using the parameters from Table 4. The right side (b) shows the estimated velocity with new gains, that mitigate the noise.**

**TABLE 7. Optimal gains of velocity estimation methods (2 millidegrees, 125 Hz).**

| Method | Par. | Optimal values (diag. elements of matrices) | | | | | |
|---|---|---|---|---|---|---|---|
| MovAv | $n$ | 2 | | | | | |
| LSF | $n$ | 4 | | | | | |
| | $p$ | 1 | | | | | |
| Kalman | $Q$ | 9.650 | 5.479 | 15.34 | 9.382 | 7.265 | 1.453 |
| linHG | $\epsilon_l$ | 0.03 | | | | | |
| | $H_1$ | 36.81 | 30.77 | 45.18 | 27.40 | 55.17 | 34.04 |
| | $H_2$ | 284.6 | 501.7 | 475.1 | 330.8 | 219.4 | 289.2 |
| nonHG | $\epsilon_n$ | 0.03 | | | | | |
| | $L_1$ | 1.011 | 0.768 | 1.153 | 1.180 | 1.193 | 1.057 |
| | $L_2$ | 1.689 | 0.997 | 2.175 | 2.216 | 2.399 | 1.835 |
| SliMod | $f^+$ | 19.04 | 25.38 | 19.93 | 29.86 | 29.27 | 48.02 |
| | $k_1$ | 119.5 | | | | | |
| | $k_2$ | 41.13 | | | | | |

**TABLE 8. Optimal gains of velocity estimation methods (1 millidegree, 250 Hz).**

| Method | Par. | Optimal values (diag. elements of matrices) | | | | | |
|---|---|---|---|---|---|---|---|
| MovAv | $n$ | 2 | | | | | |
| LSF | $n$ | 4 | | | | | |
| | $p$ | 1 | | | | | |
| Kalman | $Q$ | 1.07 | 16.71 | 26.07 | 7.87 | 0.88 | 1.28 |
| linHG | $\epsilon_l$ | 0.01 | | | | | |
| | $H_1$ | 15.63 | 19.54 | 17.09 | 9.43 | 15.75 | 19.02 |
| | $H_2$ | 32.11 | 50.34 | 34.72 | 25.92 | 19.05 | 38.35 |
| nonHG | $\epsilon_n$ | 0.01 | | | | | |
| | $L_1$ | 1.637 | 2.433 | 2.079 | 1.671 | 2.144 | 1.753 |
| | $L_2$ | 3.061 | 3.359 | 3.463 | 3.343 | 3.046 | 3.404 |
| SliMod | $f^+$ | 26.84 | 21.19 | 37.39 | 26.75 | 32.29 | 44.83 |
| | $k_1$ | 37.47 | | | | | |
| | $k_2$ | 88.84 | | | | | |

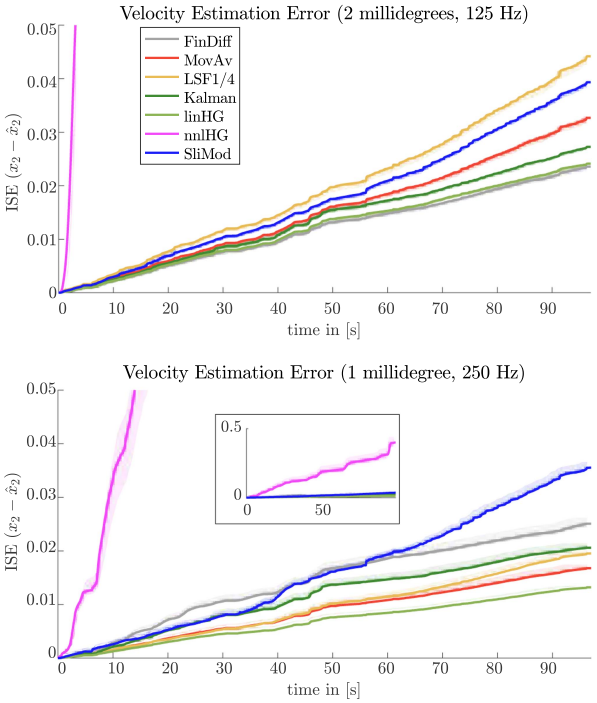


**FIGURE 9.** ***Higher sensor resolution and sampling rate:*** **comparison of the velocity estimation methods.**

### D. HIGHER SENSOR RESOLUTION AND SAMPLING RATE

At last, we compare the estimation methods when operating the robot at a higher sensor resolution and higher sampling rate. We repeat the tuning process for two configurations: 1) 2 millidegrees per increment and sampling at 125 Hz, and 2) 1 millidegrees per increment and sampling at 250 Hz. For the latter configuration we compute the ground truth using the `filtfilt` zero-phase filter from MATLAB with a cut-off frequency at 28 Hz.

The resulting optimal gains are shown in Tables 7 and 8, and the cumulative ISEs are shown in Fig. 9. The estimation errors decrease due to the improved measurement and the higher gains for the observer methods. What is immediately noticeable from the ISE graphs is that the relative difference between the filtering methods (FinDiff, MovAv, LSF1/4) depends much on the sampling rate and the encoder resolution. In the first case, FinDiff and LSF1/4 perform better than MovAv; in the second case, MovAv is the best of the filters. In contrast, the relative difference between the observer methods (Kalman, linHG, nnlHG, SliMod) stays similar in all

**TABLE 9. Qualitative comparison of each estimation method.**

| | MovAv | Deriv. filters | Kalman | linHG | nnlHG | SliMod |
|---|---|---|---|---|---|---|
| Implementation | ● | ● | ◐ | ◐ | ○ | ○ |
| Tuning | ● | ● | ◐ | ◐ | ◐ | ◐ |
| Estimation accuracy | ◐ | ◐ | ◐ | ● | ○ | ◐ |
| Fault tolerance | ◐ | ◐ | ● | ◐ | ◐ | ● |
| Tracking error | ● | ● | ● | ● | ◐ | ● |

● = best, ○ = worst, and ◐ = in between

considered cases. The linear high-gain observer is consistently amongst the best in terms of accuracy in all experiments.

### E. DISCUSSION

We have thoroughly investigated the performance of velocity estimation methods, considering multiple aspects that are important for applying them to real robots. The moving average and the derivative filters are easy to implement and to tune, but their accuracy varies between different sensor resolutions and sampling rates. In our experiments, the derivative filters (LSF, BDE, TSE) did not perform very well and can lead to large errors, as can be seen by comparing Table 5 to the ISE. The linear high-gain observer consistently has the best accuracy when the gains are properly tuned. The nonlinear high-gain observer has not proven to be a suitable option in our experiments, although we tried our best to identify the robot model as accurately as possible. The sliding-mode observer, which is also model-based, performs well and has the added benefit of being robust against erroneous sensor measurements, but often experienced a loss of accuracy at high accelerations. From our comparison we cannot conclude that model-based observers, which are harder to implement due to the dynamical model of the robot, perform better than model-free estimation methods. Finally, our experiments have shown that most of the chosen estimators do not noticeably influence the tracking error, when they are optimally tuned. However, an inaccurate velocity estimation, such as the nonlinear high-gain observer in Fig. 6, will significantly degrade the tracking performance of the controller. This emphasizes the practical relevance of having accurate estimates and the importance of tuning to reach the best performance. In Table 9, we qualitatively summarize our discussed observations.

## V. CONCLUSION

This work experimentally compares multiple velocity estimation methods for robot manipulators, namely the finite difference algorithm, moving average filtering, derivative filtering, Kalman filtering, linear high-gain observer, nonlinear high-gain observer and the sliding mode observer. Additionally, we propose an automatic tuning procedure based on a genetic algorithm. The linear high-gain observer is consistently amongst the best in terms of accuracy, independent of the sampling rate and sensor resolution, while the sliding-mode observer is robust against sensor faults. Simple-to-implement schemes, such as the moving average filter, can perform well enough, when optimally tuned, without affecting the tracking error of the closed-loop robot system. The nonlinear high-gain observer was not suitable for our robot. Overall, when the other estimators are tuned using our genetic algorithm, their optimal performance is similar.

## REFERENCES

[1] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2016.
[2] C. C. De Wit, N. Fixot, and K. J. Aström, "Trajectory tracking in robot manipulators via nonlinear estimated state feedback," *IEEE Trans. Robot. Automat.*, vol. 8, no. 1, pp. 138–144, Feb. 1992.
[3] S. Nicosia and P. Tomei, "Robot control by using only joint position measurements," *IEEE Trans. Autom. Control*, vol. 35, no. 9, pp. 1058–1061, Sep. 1990.
[4] L. Bascetta, G. Magnani, and P. Rocco, "Velocity estimation: Assessing the performance of non-model-based techniques," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 2, pp. 424–433, Mar. 2009.
[5] R. H. Brown, S. C. Schneider, and M. G. Mulligan, "Analysis of algorithms for velocity estimation from discrete position versus time data," *IEEE Trans. Ind. Electron.*, vol. 39, no. 1, pp. 11–19, Feb. 1992.
[6] P. S. Carpenter, R. H. Brown, J. A. Heinen, and S. C. Schneider, "On algorithms for velocity estimation using discrete position encoders," in *Proc. IEEE Conf. Ind. Electron.*, 1995, vol. 2, pp. 844–849.
[7] J. M. Daly and H. M. Schwartz, "Experimental results for output feedback adaptive robot control," *Robotica*, vol. 24, no. 6, pp. 727–738, 2006.
[8] S. Nicosia, A. Tornambè, and P. Valigi, "State estimation in robotic manipulators: Some experimental results," *J. Intell. Robot. Syst.*, vol. 7, no. 3, pp. 321–351, 1993.
[9] S. Islam, P. X. Liu, and A. Saddik, "Experimental comparison of model-based and model-free output feedback control system for robot manipulators," in *Proc. Int. Conf. Auton. Intell. Syst.*, 2011, pp. 177–188.
[10] B. Bona and M. Indri, "Analysis and implementation of observers for robotic manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1998, vol. 4, pp. 3006–3011.
[11] G. S. Natal, A. Chemori, and F. Pierrot, "Nonlinear control of parallel manipulators for very high accelerations without velocity measurement: Stability analysis and experiments on par2 parallel manipulator," *Robotica*, vol. 34, no. 1, pp. 43–70, 2016.
[12] S. Islam and P. X. Liu, "Output feedback sliding mode control for robot manipulators," *Robotica*, vol. 28, no. 7, pp. 975–987, 2010.
[13] S. Islam and P. X. Liu, "PD output feedback control design for industrial robotic manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 1, pp. 187–197, Feb. 2011.
[14] S. Islam and P. X. Liu, "Robust adaptive fuzzy output feedback control system for robot manipulators," *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 2, pp. 288–296, Apr. 2011.
[15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control* (Advanced Textbooks in Control and Signal Processing Series). London, U.K.: Springer, 2009.
[16] R. C. Kavanagh and J. M. D. Murphy, "The effects of quantization noise and sensor nonideality on digital differentiator-based rate measurement," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 6, pp. 1457–1463, Dec. 1998.
[17] S. Y. Lim, D. M. Dawson, and K. Anderson, "Re-examining the Nicosia-Tomei robot observer-controller from a backstepping perspective," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 3, pp. 304–310, May 1996.
[18] C.-C. Yih, "Extended Nicosia–Tomei velocity observer-based robot-tracking control," *IET Control Theory Appl.*, vol. 6, pp. 51–61, 2012.
[19] K. W. Lee and H. K. Khalil, "Adaptive output feedback control of robot manipulators using high-gain observer," *Int. J. Control*, vol. 67, no. 6, pp. 869–886, 1997.
[20] H. K. Khalil, *High-Gain Observers in Nonlinear Feedback Control*. Philadelphia, PA, USA: SIAM, 2017.

[21] M. Erlic and W.-S. Lu, "A reduced-order adaptive velocity observer for manipulator control," *IEEE Trans. Robot. Automat.*, vol. 11, no. 2, pp. 293–303, Apr. 1995.
[22] S. Malagari and B. Driessen, "Globally exponential controller/observer for tracking in robots without velocity measurement," *Asian J. Control*, vol. 14, no. 2, pp. 309–319, 2012.
[23] P. Ordaz, "Nonlinear robust output stabilization for mechanical systems based on Luenberger-like controller/observer," *ASME J. Dyn. Syst., Meas. Control*, vol. 139, no. 8, pp. 1–6, 2017.
[24] C. Canudas de Wit and J. J. E. Slotine, "Sliding observers for robot manipulators," *Automatica*, vol. 27, no. 5, pp. 859–864, 1991.
[25] M. Van, H.-J. Kang, Y. S. Suh, and K. S. Shin, "Output feedback tracking control of uncertain robot manipulators via higher-order sliding-mode observer and fuzzy compensator," *J. Mech. Sci. Technol.*, vol. 27, no. 8, pp. 2487–2496, 2013.
[26] Z. Qu, D. M. Dawson, J. F. Dorsey, and J. D. Duffie, "Robust estimation and control of robotic manipulators," *Robotica*, vol. 13, no. 3, pp. 223–231, 1995.
[27] M. Arteaga and R. Kelly, "Robot control without velocity measurements: New theory and experimental results," *IEEE Trans. Robot. Automat.*, vol. 20, no. 2, pp. 297–308, Apr. 2004.
[28] F. Bouakrif, D. Boukhetala, and F. Boudjema, "Passivity-based controller observer for robot manipulators," *Int. J. Robot. Automat.*, vol. 25, no. 1, pp. 1–8, 2010.
[29] F. Bouakrif, "Trajectory tracking control using velocity observer and disturbances observer for uncertain robot manipulators without tachometers," *Meccanica*, vol. 52, no. 4-5, pp. 861–875, 2017.
[30] J. A. Heredia and W. Yu, "High-gain observer-based PD control for robot manipulator," in *Proc. IEEE Amer. Control Conf.*, 2000, pp. 2518–2522.
[31] P. Pagilla and M. Tomizuka, "An adaptive output feedback controller for robot arms: Stability and experiments," *Automatica*, vol. 37, no. 7, pp. 983–995, 2001.
[32] X. Liang, X. Huang, M. Wang, and X. Zeng, "Adaptive task-space tracking control of robots without task-space- and joint-space-velocity measurements," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 733–742, Aug. 2010.
[33] Y. H. Kim and F. L. Lewis, "Neural network output feedback control of robot manipulators," *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 301–309, Apr. 1999.
[34] T. Sun, H. Pei, Y. Pan, H. Zhou, and C. Zhang, "Neural network-based sliding mode adaptive control for robot manipulators," *Neurocomputing*, vol. 74, no. 14-15, pp. 2377–2384, 2011.
[35] P. R. Bélanger, P. Dobrovolny, A. Helmy, and X. Zhang, "Estimation of angular velocity and acceleration from shaft-encoder measurements," *Int. J. Robot. Res.*, vol. 17, no. 11, pp. 1225–1233, 1998.
[36] A. Loria, "Observers are unnecessary for output-feedback control of Lagrangian systems," *IEEE Trans. Autom. Control*, vol. 61, no. 4, pp. 905–920, Apr. 2016.
[37] P. R. Bélanger, "Estimation of angular velocity and acceleration from shaft encoder measurements," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1992, pp. 585–592.
[38] S. M. Phillips and M. S. Branicky, "Velocity estimation using quantized measurements," in *Proc. IEEE Conf. Decis. Control*, 2003, pp. 4847–4852.
[39] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska, and S. Katsura, "Industrial Applications of the Kalman Filter: A Review," *IEEE Trans. Ind. Electron.*, vol. 60, no. 12, pp. 5458–5471, Dec. 2013.
[40] C. A. Lightcap and S. A. Banks, "An extended Kalman filter for real-time estimation and control of a rigid-link flexible-joint manipulator," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 1, pp. 91–103, Jan. 2010.
[41] A. N. Atassi and H. K. Khalil, "A separation principle for the stabilization of a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 44, no. 9, pp. 1672–1687, Sep. 1999.
[42] J. Davila, L. Fridman, and A. Levant, "Second-order sliding-mode observer for mechanical systems," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1785–1789, Nov. 2005.
[43] L. Fraguela Cuesta, L. Fridman, and V. V. Alexandrov, "Position stabilization of a stewart platform: High-order sliding mode observers based approach," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 5971–5976.
[44] A. M. Dabroom and H. K. Khalil, "Discrete-time implementation of high-gain observers for numerical differentiation," *Int. J. Control*, vol. 72, no. 17, pp. 1523–1537, 1999.
[45] Z. Gao, "Scaling and bandwidth-parameterization based controller tuning," in *Proc. Amer. Control Conf.*, 2003, pp. 4989–4996.
[46] K. J. Åström, T. Hägglund, C. C. Hang, and W. K. Ho, "Automatic tuning and adaptation for PID controllers - A survey," *Control Eng. Pract.*, vol. 1, no. 4, pp. 699–714, 1993.
[47] K. Krishnakumar and D. E. Goldberg, "Control system optimization using genetic algorithms," *J. Guid., Control, Dyn.*, vol. 15, no. 3, pp. 735–740, 1992.
[48] S. Ge, T. Lee, and G. Zhu, "Genetic algorithm tuning of Lyapunov-based controllers: An application to a single-link flexible robot system," *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, pp. 567–574, Oct. 1996.
[49] F. Nagata, K. Kuribayashi, K. Kiguchi, and K. Watanabe, "Simulation of fine gain tuning using genetic algorithms for model-based robotic servo controllers," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Automat.*, 2007, pp. 196–201.
[50] W. Ainhauser, J. Gerstmayr, and A. Giusti, "Multi-objective trajectory tracking optimization for robots with elastic joints," in *Advances in Service and Industrial Robotics*. Cham, Switzerland: Springer, 2021, pp. 250–258.
[51] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 1999.
[52] S. B. Liu and M. Althoff, "Reachset conformance of forward dynamic models for the formal analysis of robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 370–376.

**STEFAN B. LIU** (Member, IEEE) received the B.S. degree in mechatronics, and the M.S. degree in robotics from the Technical University of Munich (TUM), Munich, Germany, in 2015, and 2017, respectively. He is currently working toward the Ph.D. degree with the Cyber-Physical Systems Group, TUM Department of Informatics. His research interests include formal methods in robotics, physical human-robot interaction, modeling and identification, and modular robots.

**ANDREA GIUSTI** (Member, IEEE) received the bachelor's degree in telecommunications engineering and the master's degree (summa cum laude) in mechatronic engineering from the University of Trento, Trento, Italy, in 2010 and 2013, respectively, and the Ph.D. degree in robotics from the Technical University of Munich, Munich, Germany, in 2018. He is currently a Researcher and Head of robotics and intelligent systems engineering with Fraunhofer Italia Research, Bolzano, Italy. His research interests include modeling and control of robotic and mechatronic systems, modular and reconfigurable robots, and human-robot collaboration.

**MATTHIAS ALTHOFF** (Member, IEEE) received the Diploma Engineering degree in mechanical engineering, and the Ph.D. degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 2005 and 2010, respectively. He is currently an Associate Professor in computer science with the Technical University of Munich. From 2010 to 2012, he was a Postdoctoral Researcher with Carnegie Mellon University, Pittsburgh, PA, USA, and from 2012 to 2013, an Assistant Professor with the Ilmenau University of Technology, Ilmenau, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, robotics, automated vehicles, and power systems.

## A.5 Guarantees for Real Robotic Systems: Unifying Formal Controller Synthesis and Reachset-Conformant Identification [5]

**Summary** The safety problem of this paper is to find an optimal motion controller for a robot manipulator such that we formally ensure the torque constraints of the motors. Previous work demonstrated formal controller synthesis on simulated models, but this work transfers the results to a real robotic system by applying controller synthesis on an identified reachset-conformant model. This work proposes combining synthesis and identification into a single optimization problem, and the model is determined to optimally meet the controller and safety objectives. This work demonstrates this idea for the class of feedback-linearizing motion controllers.

The underlying verification model of the robot is a forward dynamics model that assumes a feedback-linearizing control. The result is that the dynamics of each robot axis are decoupled from each other and are linear. The main theorem of this work formulates reachset conformance checking as a linear inequality so that it can be easily integrated into the optimization problem for controller synthesis. Finally, motion controllers are synthesized by computing and minimizing the reachable set of the tracking error and the reachable set of the robot input such that the robot input meets the torque constraints.

The experimental results are obtained from a real six-degrees-of-freedom robot manipulator and show that our approach successfully predicts the actual tracking error and that the combined synthesis and identification approach works better than a separate approach. A comparison with LQG optimization demonstrates that our approach is more straightforward to tune. Finally, the effectiveness of this approach is evaluated using motion controllers with and without disturbance compensation to show that our approach can consider the effect of disturbance compensation and successfully predict the possible reduction of the tracking error. Most importantly, we have established a path to applying formal controller synthesis on real systems where the model is partially unknown.

**Author Contributions** **S. L.** developed the reachset-conformant identification method. **S. L.** and B. S. developed the combined controller synthesis and reachset-conformant identification method. **S. L.** developed the theorems and the iterative synthesis approach **S. L.** designed, conducted and evaluated the experiments. **S. L.** and B. S. wrote the article. M. A. led the research project, provided feedback, and helped improve the manuscript.

**Journal article** The accepted version of the journal paper is reprinted in this thesis. The final version of the record is available at `https://doi.org/10.1109/TRO.2023.3277268`.

TRANSACTIONS ON ROBOTICS, VOL. X, NO. X, AUGUST XXXX 

# Guarantees for Real Robotic Systems: Unifying Formal Controller Synthesis and Reachset-Conformant Identification

Stefan B. Liu, Bastian Schürmann, and Matthias Althoff

***Abstract*—Robots are used increasingly often in safety-critical scenarios, such as robotic surgery or human-robot interaction. To ensure stringent performance criteria, formal controller synthesis is a promising direction to guarantee that robots behave as desired. However, formally ensured properties only transfer to the real robot when the model is appropriate. We address this problem by combining the identification of a reachset-conformant model with controller synthesis. Since the reachset-conformant model contains all the measured behaviors of the real robot, the safety properties of the model transfer to the real robot. The transferability is demonstrated by experiments on a real robot, for which we synthesize tracking controllers.**

***Index Terms*—formal methods, model identification, reachability analysis, reachset conformance, controller synthesis, robots.**

## I. Introduction

Guaranteeing and optimizing control performance has been a challenge for the robust control of robots for a long time (e.g., see the surveys in [1], [2]). One of the reasons is that models of robots and their controllers do not consider certain effects: 1) rigid-body models of robots do not consider flexible joints and links; 2) some model parameters are falsely assumed to be constant, e.g., some friction parameters in robots depend on load and temperature, which are not accounted for in standard models; and 3) control limitations, such as finite motor capabilities, finite sampling time, measurement errors, delays, noise within circuit boards, etc., are typically not modeled. Due to these and other reasons, an identified model can never exhibit exactly the same behavior as the real system.

We propose a novel formal synthesis framework that uses *reachability analysis* [3] to optimize the controller and provide formal guarantees for robotic systems. Reachability analysis allows us to formally bound all possible behaviors, making it possible to decide whether a given specification is always met.

Our main challenge is how to correctly identify models such that the guarantees obtained for these transfer to the corresponding real robot. We will make use of the *reachset conformance* relation [4], which means that the reachable sets of the model must contain all possible behaviors of the real robot. Broadly speaking: if a property can be guaranteed for a conservative model, then we can guarantee the same property for the real system (a formal explanation will be provided in Sec. II). In this paper, we combine reachset-conformant identification with controller synthesis in a single optimization problem that simultaneously finds the optimal model and controller. Obviously, if one is interested in only identifying a reachset-conformant model or only finding a controller for a given model of a robot, our approach is also applicable.

This paper focuses on the synthesis of tracking controllers for feedback-linearized robots, but is applicable to all linear systems. The software, as well as the scripts to replicate our experimental results, can be obtained from Code Ocean[1].

All authors are with the Department of Informatics, Technical University of Munich, Garching, 85748, Germany. Email: [stefan.liu; bastian.schuermann; althoff]@tum.de.

Manuscript received April XX, XXXX; revised August XX, XXXX. This work was supported by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreement 101016007 (Project CONCERT).

### A. Literature overview

We divide our review of relevant works into three parts: robust control, formal synthesis, and model identification.

*1) Robust control:* Previous robustness analyses of feedback-linearizing robot controllers, many of which are surveyed in [1] and [2], assume that system uncertainties originate from model errors, which can be considered additive nonlinear disturbances in the feedback-linearized model. For instance, the nonlinear disturbance representation helps to prove general uniform ultimate boundedness (UUB) for a computed torque controller in [5]. In [6, Section 8.5.3], a robust controller is proposed, where UUB is shown by bounding the mass matrix and other nonlinear terms of the robot dynamics. The approach in [7] presents a control scheme for robots that achieves a desired tracking error with a pre-specified convergence rate. Generally, in previous works, UUB is only shown through Lyapunov's theorem, which can be very tedious. In contrast, we quantitatively model the additive disturbances as an uncertain set and show UUB directly by computing the reachable tracking error of a robot using standard algorithms for reachability analysis [3]. These algorithms also make it possible to incorporate sampling times, measurement errors, and delays—all of which influence the final tracking error.

$\mathcal{H}_\infty$-synthesis (e.g., in [8], [9]) is a method that optimally designs robot controllers that minimize an $\mathcal{H}_\infty$-norm, which captures disturbance effects expressed in the frequency domain. However, $H_\infty$-synthesis does not provide any guarantees with respect to input constraints. Similarly, the linear quadratic regulator (LQR) is an optimization-based approach, which has robustness properties [10] but fails to consider constraints (more details in Sec. IV).

[1]https://doi.org/10.24433/CO.1635335.v1

A well-known type of controller ensuring the satisfaction of state and input constraints despite the presence of disturbances is tube-based model predictive control (MPC). There, an optimization algorithm iteratively optimizes a reference trajectory over a moving horizon while a feedback controller keeps the system in a tube around the reference trajectory. For linear systems, the computation of the reference trajectory and the control invariant set of the tube can be decoupled due to the superposition principle [11]–[14], while for nonlinear systems, this becomes more complex. Still, a number of approaches exist for nonlinear systems, e.g., [15]–[18]. Other ways to ensure the satisfaction of constraints are to embed an invariance controller [19], [20] or use control barrier functions [21], [22]. In contrast to tube-based MPC, our approach meets the specification for the real robot and not just its model. In addition, our approach does not require finding a Lyapunov function, as required for most tube-based MPC approaches.

*2) Formal synthesis:* Formal controller synthesis is a research area with many recent results in robotics; we refer to [23] for an overview. The idea is to compute a controller which formally guarantees the satisfaction of complex specifications. Many of the control approaches mentioned in [23] focus on high-level planning with little focus on uncertainty, while our method formally synthesizes low-level controllers, where uncertainty plays a larger role.

Many formally correct controllers are realized as abstraction-based controllers [24]–[34], which satisfy rich specifications such as temporal logic expressions. By discretizing the state and input space, they obtain a finite state abstraction of the system so that they can use techniques from automata theory to synthesize controllers. The necessity to discretize the state space leads to an exponential computational complexity with respect to the number of continuous state variables, which restricts the application to lower-dimensional systems. Some works try to avoid this problem by not abstracting the whole state space, e.g., see [35]–[37]. In contrast to these papers, we avoid discretizing the state space and directly compute the reachable set of the dynamic system, which scales polynomially with the number of state variables [3].

Instead of abstracting the whole state space, other approaches compute safe motion primitives for mobile robots, i.e., short trajectory pieces with a corresponding controller that keeps the system in predefined sets. By computing many motion primitives and storing them in a maneuver automaton, they can be used with a discrete online planner, which only needs to find a suitable concatenation of motion primitives [38], [39]. There are different methods to compute these motion primitives, e.g., using LQR trees [40], [41], or by combining optimization with reachability analysis [42]–[44]. For robotic systems, such as manipulators, precomputing motion primitives would be infeasible since the number of required motion primitives scales exponentially with the number of states and inputs. Instead, our goal is to provide guarantees for the tracking error independently from the desired motion.

Other techniques, such as interval arithmetics [45] or Hamilton-Jacobi reachability [46], can also be used to compute and ensure the tracking error bounds of dynamical systems given known disturbances. In the next few paragraphs, we will review techniques that help us if disturbances are unknown.

*3) Identification of model uncertainties:* Uncertainties can be generally categorized as stochastic and set-based uncertainties formulated in the frequency or time domain [47]. A discussion of uncertainties in the frequency domain for robust control can be found in [48]. Stochastic aspects of model uncertainty are treated in large detail in [49]. For instance, in [50], the stochastic uncertainty of robot kinematics is identified through Monte Carlo sampling. Since we focus on providing guarantees, we will discuss set-based uncertainties in the time domain.

Formal synthesis requires models that enclose the behavior of real systems. This is also called the *model conformance* relation and has been treated in-depth in [4]. Most literature on set-based identification is based on finding a *simulation relation* since it allows a transfer of, e.g., temporal logic properties for the entire state space. The model can be a coarse-grained abstraction of the state space into a discrete automaton (e.g., for the navigation of mobile robots [23]) or differential inclusions [51], [52]. The paper in [51] identifies a linear system with non-determinism such that all state measurements are within a polytopic reachable set. The paper in [52] identifies piece-wise affine models using mixed-integer linear programming, also establishing a simulation relation between measured states with hyperrectangular reachable sets. In contrast to these works, we use zonotopes, which have a special structure that allows us to reduce the identification to a linear problem.

However, if a system is high-dimensional, but only a few outputs are relevant for synthesis, then the simulation relation can be too restrictive and conservative. Thus, *trace* and *reachset conformance* have been proposed to relax the formal relation only to the output of a system [4]. In [53], the authors apply trace conformance by reconstructing disturbance traces for a real autonomous vehicle. The set of non-deterministic disturbances is then taken as the outer bounds of all disturbance traces. *Reachset conformance*, on the other hand, is a further relaxation that only requires that the output traces of a system must be within the reachable set of the model. The main advantage is that we can handle sensor noise and arbitrary disturbances, which is not possible for trace conformance since this would create infinitely many possible behaviors, resulting in a more flexible model-order reduction [54] or even applying black-box identification methods [55]. For transferring safety properties, reachset conformance is sufficient [4].

Our previous work on the reachset conformance of robot manipulators, on which this paper is based, can be found in [56], [57]. Our work in [56] aims to identify the uncertain sets of a forward dynamical model, while here, we identify a feedback-linearized robot model. In [57], a reachset-conformant inverse dynamical robot model is identified. In these works, we have not combined reachset-conformant identification with controller synthesis.

The identification of conformant parameter sets can also be viewed as a synthesis problem. The authors in [58], [59] are able to incorporate additional model knowledge as temporal

logic constraints to improve identification results.

The main criterion for the identification of parameter sets is usually the size of their range. However, small uncertainties do not necessarily lead to good robust control, and large model errors do not necessarily lead to bad control performance, as [60] has pointed out. Therein lies the motivation for *identification for control*, in which the model uncertainties are determined in a way that is optimal for the control goal [47]. Our framework builds upon these ideas to formulate controller synthesis and model identification as a unified optimization problem, where they share a common cost function.

Notably, set-membership identification [61]–[65] has certain similarities to our approach because it is also a set-based method. There, the goal is to identify the true parameter of a system by reducing the feasible solution set as much as possible. This is different from reachset-conformant identification, where the goal is to model the parameter set large enough to ensure reachset conformance. Parameters obtained from set-membership identification are generally not reachset conformant and cannot be used for our robust control framework.

### B. Structure of this paper

This paper is structured as follows: in Sec. II, we provide preliminaries on zonotopes and on the reachability analysis of uncertain linear systems. Our combined controller synthesis and reachset-conformant identification framework is presented in Sec. III. We address the application of these methods to the tracking control problem of robots in Sec. IV and conclude this paper in Sec. V.

## II. Preliminaries and Problem Statement

We first introduce preliminaries on set operations and subsequently describe the control problem.

### A. Preliminaries on set operations

We denote sets using calligraphic letters (e.g., $\mathcal{A}$), matrices using upper case letters (e.g., $A$), vectors using $\vec{\cdot}$, and scalar values using lower case letters (e.g., $a$). To represent sets, we mainly use zonotopes.

**Definition 1** (Zonotope)**.** *A zonotope $\mathcal{Z}$ is defined by a center $\vec{c}$ and a generator matrix $G$ of proper dimension, where $\vec{g}^{(h)}$ is its $h$-th column:*

$$\mathcal{Z} = (\vec{c}, G) := \left\{ \vec{x} = \vec{c} + \sum_{h=1}^{s} \beta_h \vec{g}^{(h)} \middle| \beta_h \in [-1, 1] \right\}.$$

A $\theta$-dimensional zonotope $\mathcal{Z}$ with $s$ generators can also be described by an intersection of $2\binom{s}{\theta-1}$ half-spaces.

**Proposition 1** (H-representation of a zonotope [66])**.** *The half-space representation of a zonotope is $\{\vec{y} \,|N\vec{y} \leq \vec{d}\}$,*

$$N = \begin{bmatrix} N^+ \\ -N^+ \end{bmatrix}, \quad \vec{d} = \begin{bmatrix} \vec{d}^+ \\ \vec{d}^- \end{bmatrix},$$

*where each row of $N$ and $\vec{d}$ contains the normal vectors and distances of a half-space, respectively. The direction of each normal vector is computed from a reduced generator matrix $G^{\langle\gamma,\dots,\eta\rangle}$, where $\gamma, \dots, \eta$ are the $s - \theta + 1$ indices of the generators that have been removed from $G$. The $j$-th row of $N^+$, where $j \in 1..\binom{s}{\theta-1}$, is*

$$\vec{n}_j^+ = \mathrm{nX}(G^{\langle\gamma,\dots,\eta\rangle}) / \| \mathrm{nX}(G^{\langle\gamma,\dots,\eta\rangle}) \|_2, \tag{1}$$

$$\mathrm{nX}(H) := [\dots, (-1)^{i+1} \det(H^{[i]}), \dots]^T, \tag{2}$$

*where $H^{[i]}$ means that the $i$-th row of $H$ is removed, and the $j$-th row of $\vec{d}^+$ and $\vec{d}^-$ are*

$$d_j^+ = \vec{n}_j^{+T} \vec{c} + \Delta d_j, \qquad d_j^- = -\vec{n}_j^{+T} \vec{c} + \Delta d_j, \tag{3}$$

$$\Delta d_j = \sum_{h=1}^{s} |\vec{n}_j^{+T} g^{(h)}|. \tag{4}$$

Many operations on zonotopes can be exactly and efficiently computed [3]. Let us define the Minkowski sum of sets as $\mathcal{A} \oplus \mathcal{B} = \{\vec{a} + \vec{b} \mid \vec{a} \in \mathcal{A}, \vec{b} \in \mathcal{B}\}$. For zonotopes, the following propositions hold:

**Proposition 2** (Minkowski sum of zonotopes [67])**.** *Zonotopes are closed under Minkowski sum:*

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = (\vec{c}_1, G_1) \oplus (\vec{c}_2, G_2) = (\vec{c}_1 + \vec{c}_2, [G_1, G_2]).$$

**Proposition 3** (Linear transformation of zonotopes [67])**.** *Zonotopes are closed under linear transformation:*

$$A\mathcal{Z} = (A\vec{c}, AG).$$

To reason about the size of a zonotope, we introduce a norm that is defined based on the edge lengths of its interval hull. Alternative norms can be found in [68].

**Proposition 4** (Interval hull of zonotopes [67])**.** *The interval hull $\mathcal{I}(\mathcal{Z}) = [\vec{i}^-, \vec{i}^+]$, where $\vec{i}^-$ is the left bound and $\vec{i}^+$ is the right bound, is the smallest interval enclosing a set $\mathcal{Z} = (\vec{c}, [\dots, \vec{g}^{(h)}, \dots])$, where*

$$\vec{i}^- = \vec{c} - \vec{\delta}, \qquad \vec{i}^+ = \vec{c} + \vec{\delta}, \qquad \vec{\delta} = \sum_{h=1}^{p} |\vec{g}^{(h)}|.$$

**Definition 2** (Norm of zonotopes)**.** *We define the norm of a zonotope as the sum of each element of $\vec{\delta}$, which represents the size of the interval hull:*

$$\|\mathcal{Z}\| := \sum_{i=1}^{\theta} |\delta_i|.$$

Next, we introduce reachable sets for linear systems. Since robots are commonly measured and controlled by computers, we restrict our discussion to discrete time. We use the notation $a[k]$ to express the value of $a$ at time $k\Delta t$, where $k \in \{0, 1, \dots\}$ and $\Delta t$ is the sampling time. Discrete-time linear systems are defined by the following difference and output equations:

$$\begin{aligned} \vec{x}[k+1] &= A\vec{x}[k] + B\vec{u}[k] + \vec{w}[k], \\ \vec{y}[k] &= C\vec{x}[k] + D\vec{u}[k] + \vec{v}[k], \end{aligned} \tag{5}$$

where $A, B, C, D$ are matrices of proper dimension, $\vec{x}[k]$ is the state, $\vec{y}[k]$ is the output, $\vec{u}[k] \in \mathcal{U}$ is the control input constrained by $\mathcal{U}$, and $\vec{w}[k] \in \mathcal{W} = (\vec{c}_W, G'_W \operatorname{diag}(\vec{\alpha}_W))$

and $\vec{v}[k] \in \mathcal{V} = (\vec{c}_V, G'_V \operatorname{diag}(\vec{\alpha}_V))$ are the disturbances sensor noise, respectively, bounded by appropriate zonotopes to capture the errors of the nominal model. The operator $\operatorname{diag}(.)$ returns a matrix where the elements of the input vector are on the diagonal. Subsequently, vectors $\vec{\alpha}_W$ and $\vec{\alpha}_V$ are variables that scale the length of each generator of $\mathcal{W}$ and $\mathcal{V}$, respectively.

Reachable sets are defined as the set of all possible outputs of a system, given a set of initial states and the set of all possible inputs. The reachable set of (5) after one time step is computed through a set-based evaluation of the difference and output equations in (5):

$$\mathcal{R}[k+1] = C\left(A\mathcal{X}[k] \oplus B\vec{u}[k] \oplus \mathcal{W}\right) \oplus D\vec{u}[k+1] \oplus \mathcal{V}, \tag{6}$$

where $\mathcal{X}[k]$ is the current set of states. Given an initial set $\mathcal{X}[0]$, the reachable set after $k$ time steps can be computed by recursively applying (6):

$$\mathcal{R}[k+1] = C\left(A^{k+1}\mathcal{X}[0] \oplus \sum_{i=0}^{k} A^i B\vec{u}[i] \oplus \bigoplus_{i=0}^{k} A^i\mathcal{W}\right) \oplus D\vec{u}[k+1] \oplus \mathcal{V}. \tag{7}$$

When using zonotopes, the above computation is exact since (7) only involves Minkowski sums and linear transformations.

### B. Plant model and reachset conformance

In this subsection, we discuss the model of our use case. Because our method applies to linear systems and the robot dynamics are nonlinear in general, we implement an internal *feedback linearization* in the robot. Let us derive the plant model by regarding the following rigid-body dynamics of a robot [2, Sec. 2.2]:

$$M(\vec{q})\ddot{\vec{q}} + \vec{\psi}(\vec{q}, \dot{\vec{q}}) = \vec{\tau}, \tag{8}$$

where $\vec{q}$ is the vector of joint positions, $\vec{\tau}$ is the vector of joint torques, $M$ is the mass matrix, and $\vec{\psi}$ contains the Coriolis, centripetal, gravity, and friction forces. The feedback linearization technique [6] applies a control torque

$$\vec{\tau} = M(\vec{q})\vec{u}_r + \vec{\psi}(\vec{q}, \dot{\vec{q}}) \tag{9}$$

to (8); for the rigid-body dynamics, this results in linear dynamical systems that are decoupled for each joint $i$:

$$\ddot{q}_i = u_{r,i}, \tag{10}$$

where $u_{r,i}$ is the plant input for the feedback-linearized robot with rigid-body dynamics. In the discretized state-space model for one robot joint, we additionally consider that both the input and the output are delayed by one sampling instant. Let us denote the linear dynamics by the subscript $r$ (for *robot*):

$$\begin{aligned} \vec{x}_r[k+1] &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}_r[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_r[k], \\ y_r(t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \vec{x}_r[k], \end{aligned} \tag{11}$$

where $\vec{x}_r[k] = [q[k-1], q[k], \dot{q}[k], u_r[k-1]]^T$ is the state, and $y_r$ is the measured joint position.

The dynamics of a real robot, however, will never be exactly as in (11) because 1) the rigid-body assumption has limitations, e.g., there are flexible parts in the system, 2) the inertial parameters used in the feedback linearization in (9) are usually not accurate, and 3) measurement errors affect the feedback linearization. To capture the uncertainties of the robot, we add the following uncertainties: a time-varying additive disturbance $\vec{w}_{p,i}[k] \in \mathcal{W}_{p,i} \subseteq \mathbb{R}^3$, and a measurement error $v_{p,i}[k] \in \mathcal{V}_{p,i} \subseteq \mathbb{R}$, and an additional constant disturbance state $d$, where $\dot{d} = 0$. The full model is denoted the subscript $p$ (for *plant*) and is fully described in Appendix A.

The next definitions specify the data we require to test for reachset conformance.

**Definition 3** (Test case)**.** *Let $k^* \in \mathbb{N}_0$. A test case is a tuple $(y_p[0], ..., y_p[k^*], u_p[0], ..., u_p[k^*], \vec{x}_p[0])$ of output measurements $y_p[k]$, control inputs $u_p[k]$, and the initial state $\vec{x}_p[0]$.*

To account for disturbances at any point in time, we should generate *sequential* test cases (defined subsequently) to have as many initial states as possible and to maximize the number of test cases from one recording.

**Definition 4** (Sequential test cases)**.** *From one recording, we generate multiple test cases, where the state of each time step can be the start of a new test case. Sequential test cases are denoted by a superscripted index. The following relation holds for sequential test cases:*

$$\begin{aligned} \vec{y}_p^{(m+1)}[k] &= \vec{y}_p^{(m)}[k+1], \\ u_p^{(m+1)}[k] &= u_p^{(m)}[k+1]. \end{aligned}$$

Finally, we establish reachset conformance [4, Sec. 3.5] by testing the real system.

**Definition 5** (Reachset conformance testing)**.** *Given are a plant model and $M$ test cases of a real system. The model is reachset conformant for the sampling instants $k \in \{0, ..., k^*\}$ if, for each test case, the measurement of the real system is enclosed in the corresponding reachable set of the model:*

$$\forall m \, \forall k : \vec{y}_p^{(m)}[k] \in \mathcal{R}_p^{(m)}[k],$$

*where $\vec{y}_p^{(m)}[k]$ is the measured output and $\mathcal{R}_p^{(m)}[k]$ is computed using (7) considering $\vec{x}^{(m)}[0]$ and $u^{(m)}[k]$.*

We call finding of unknown parameters of the plant model, such that Def. 5 is fulfilled, *reachset-conformant identification*.

### C. Problem statement

Now, let us discuss the problem at hand. Our goal is to synthesize an optimal closed-loop system given a linear plant model, while the disturbance sets have unknown parameters to be identified. The control goal is for the output of the closed-loop system $\vec{y}_{\text{cl}} := [\hat{q}, \dot{\hat{q}}]^T$ to track a reference output $\vec{y}_{\text{ref}} := [q_d, \dot{q}_d]$ containing the desired position and velocity. The observed variables $\hat{q}, \dot{\hat{q}}$ have been chosen for $\vec{y}_{\text{cl}}$ since the

robot velocity is usually not measurable, so an observer [69] is recommended.

For the closed-loop system, we select a parameterizable linear feedback controller and a parameterizable linear observer such that the closed-loop system is also linear, and its reachable set can be computed using (7). Furthermore, we can include input feedforward signals $u_{\mathrm{ff}}$ that are added to the plant input, e.g., a desired acceleration $u_{\mathrm{ff}} := \ddot{q}_d$. In Sec. IV, we demonstrate two different closed-loop systems with unknown parameters.

Next, we specify the optimization problem for the combined controller synthesis and reachset-conformant identification. Subsequently, we define the two main reachable sets considered in our controller synthesis:

**Definition 6** (Reachable tracking error)**.** *$\mathcal{R}_e$ is a reachable set that encloses all tracking errors of the closed-loop system, such that*

$$\vec{y}_{\mathrm{cl}} \in \vec{y}_{\mathrm{ref}} \oplus \mathcal{R}_e.$$

**Definition 7** (Reachable input)**.** *$\mathcal{R}_u$ is the reachable set of all plant inputs $u_p$ in the closed-loop dynamics. A controller is considered safe if the reachable input is within the allowed set $\mathcal{U}_p$, such that $\mathcal{R}_u \subseteq \mathcal{U}_p$.*

The computation of $\mathcal{R}_e$ and $\mathcal{R}_u$ are explained in Sec. III-A. As a cost function, we choose the norm of the reachable tracking error $\mathcal{R}_e$. The variables are the unknown controller and observer parameters, as well as $\vec{c}_{W_p}, \vec{c}_{V_p}, \vec{\alpha}_{W_p}, \vec{\alpha}_{V_p}$ from the zonotopic disturbances of the plant model. These variables are aggregated into a parameter vector $\vec{p} \in \mathcal{P}$, where $\mathcal{P}$ is a user-defined search space. The optimization problem has two constraints:

- the plant model shall be reachset conformant (Def. 5),
- the plant input is constrained so that we never exceed the allowed motor torques of the robot,

and the optimization problem is formulated as:

$$\min_{\vec{p}\in\mathcal{P}} \quad \|\mathcal{R}_e(\vec{p})\|, \tag{12a}$$

$$\text{subject to} \quad \forall m \, \forall k \in [0, k^*] : \vec{y}_p^{(m)}[k] \in \mathcal{R}_p^{(m)}(\vec{p})[k], \tag{12b}$$

$$\mathcal{R}_u(\vec{p}) \subseteq \mathcal{U}_p, \tag{12c}$$

where all computed reachable sets depend on $\vec{p}$. The optimization problem is defined for each robot axis $i \in \{1..n\}$, but the set of allowed inputs $\mathcal{U}_{p,i}$ for each axis are derived from the allowed joint torque and depend on the axis configuration. Given the feedback linearization in (9), the allowed set of inputs $\mathcal{U}_p = \mathcal{U}_{p,1} \times ... \times \mathcal{U}_{p,n}$ must satisfy the torque limits:

$$\mathcal{T} \supseteq M(\mathcal{Q})\mathcal{U}_p \oplus \vec{\psi}(\mathcal{Q}, d\mathcal{Q}), \tag{13}$$

where $\mathcal{T}$ is the set of allowed torques, and $\mathcal{Q}, d\mathcal{Q}$ are the sets of allowed positions and velocities of the robot. Since (13) is nonlinear, we recommend Taylor models [70], [71] as a set representation to prove the above statement because the precision of Taylor models in approximating nonlinear functions can be set arbitrarily high.

The main advantage of this combined approach is that all parameters are synthesized for the same goal, while an approach with separate goals would lead to sub-optimal models. Notice, however, that a standalone reachset-conformant identification can be derived from the above problem by leaving out (12c) and switching to any other cost function, e.g., a prediction error as demonstrated in [56], [57], [72]. Also, by removing (12b), we arrive at the standalone controller synthesis problem proposed in [44].

## III. Combined controller synthesis and reachset-conformant identification

This section describes how to solve (12). In Sec. III-A, we first explain the computation of the reachable tracking error $\mathcal{R}_e$ and the reachable input $\mathcal{R}_u$. In Sec. III-B, we derive a linear formulation of reachset conformance (12b), which reduces the complexity of the constraint evaluation to a linear inequality check. In Sec. III-C, we discuss the need to solve (12) iteratively and cover the computational aspects in Sec. III-D.

### A. Computing the reachable tracking error and input

Often in robotics, the desired position and velocity may not be known in advance, e.g., when using online trajectory generation. Therefore, our aim is to solve (12) independently from the reference. Nevertheless, we shall restrict the desired acceleration by a set $u_{\mathrm{ff}} \in \mathcal{U}_{\mathrm{ff}}$ to disallow unbounded feedforward inputs. To later extract both $\mathcal{R}_e$ and $\mathcal{R}_u$ as a projection [73, Sec. 2.1] of the reachable set of the closed-loop system, we augment its output by the plant input $u_p$; the new output is denoted by a tilde: $\tilde{\vec{y}}_{\mathrm{cl}} := [\vec{y}_{\mathrm{cl}}, u_p]^T$.

Similar to [44], we use the superposition principle for linear systems to divide the reachable set of the closed-loop system into two parts: a set $\tilde{\mathcal{R}}_{\mathrm{cl,e}}$ that is only dependent on the disturbances $\mathcal{W}_p$ and $\mathcal{V}_p$, and a vector $\tilde{\vec{y}}_{\mathrm{cl,ref}}$ that is only dependent on the reference $\vec{y}_{\mathrm{ref}}$ and the feedforward $u_{\mathrm{ff}}$, such that the final reachable set is $\tilde{\vec{y}}_{\mathrm{cl,ref}}[k] \oplus \tilde{\mathcal{R}}_{\mathrm{cl,e}}[k]$.

The set $\tilde{\mathcal{R}}_{\mathrm{cl,e}}$ is computed using (7) by setting $\vec{y}_{\mathrm{ref}} = 0$ and $u_{\mathrm{ff}} = 0$. If the system is stable, then $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k]$ will converge to an invariant set $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$ [74], i.e., $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty{+}1] \subseteq \tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$. In practice, this convergence might not happen due to numerical issues; therefore, we implement [74, Alg. 2], which computes $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$ from an arbitrarily small and an arbitrarily large $\mathcal{X}(0)$ until they converge to a final set with a tolerance criterion that is chosen to be arbitrarily small. Thus, the computed $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$ is a positive invariant set [74] of both the tracking error and the plant input. An example of $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k]$ converging to $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$ is shown in Fig. 1.

If $\tilde{\vec{y}}_{\mathrm{cl,ref}}[k] = [\vec{y}_{\mathrm{ref}}[k], u_{\mathrm{ff}}]^T$, then the sets for the reachable tracking error and the reachable input are given by the following projections; since the reachable input $\mathcal{R}_u$ should also contain $u_{\mathrm{ff}}$, we add the bounded set $\mathcal{U}_{\mathrm{ff}}$:

$$\mathcal{R}_e := \begin{bmatrix} I_i & 0_{i\times j} \end{bmatrix} \tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty],$$
$$\mathcal{R}_u := \begin{bmatrix} 0_{j\times i} & I_j \end{bmatrix} \tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty] \oplus \mathcal{U}_{\mathrm{ff}},$$

where $I_{i\times i}$ is an identity matrix with dimension $i$, $0_{i\times j}$ is a matrix of zeros with $i$ rows and $j$ columns, $i$ is the dimension of $\vec{y}_{\mathrm{cl}}$, and $j$ is the dimension of $u_p$. However, we note that $\vec{y}_{\mathrm{ref}}[k]$ is not always equal to $\vec{y}_{\mathrm{cl,ref}}[k]$; extensions considering the remaining error can be found in Appendix B.

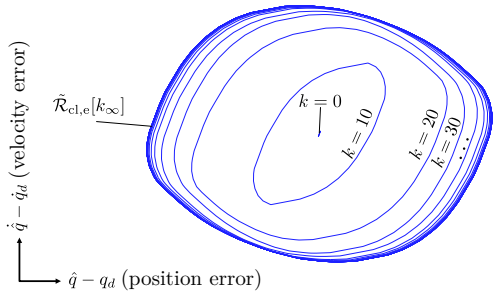


Fig. 1. Computation of the reachable tracking error, which is a projection of the converged set $\tilde{\mathcal{R}}_{\mathrm{cl,e}}[k_\infty]$.

### B. Reachset conformance as a set of linear inequalities

If constraint (12b) was naively implemented, the reachable set $\mathcal{R}_p^{(m)}$ would have to be computed for each test case. However, we will show that (12b) can be reduced to a set of linear inequalities depending on $\vec{\xi} = [\vec{c}_{W_p}, \vec{c}_{V_p}, \vec{\alpha}_{W_p}, \vec{\alpha}_{V_p}]^T$. For the remainder of this subsection, all variables refer to the plant model, and the subscript $p$ is omitted for ease of notation.

The first simplification is to combine all test cases as described subsequently to check reachset conformance by a single reachability analysis. Let us define $\vec{y}_*^{(m)}[k]$ as the nominal solution of (7) for the plant without the disturbance sets $\mathcal{W}$ and $\mathcal{V}$, and consider $\vec{x}^{(m)}[0]$ as the initial state. To make (12b) independent of each test case, we subtract the nominal solution from both $\vec{y}^{(m)}[k]$ and the reachable set $\mathcal{R}^{(m)}[k]$:

$$\vec{y}_a^{(m)}[k] := \vec{y}^{(m)}[k] - \vec{y}_*^{(m)}[k], \tag{14}$$

$$\mathcal{R}_a[k] := \mathcal{R}^{(m)}[k] - \vec{y}_*^{(m)}[k] \overset{(7)}{=} \bigoplus_{i=0}^{k-1} CA^i\mathcal{W} \oplus \mathcal{V}, \tag{15}$$

where $\vec{y}_a^{(m)}[k]$ is the deviation of the real behavior from the nominal one, and $\mathcal{R}_a[k]$ is now independent of the input and the initial state. Therefore, for linear systems, the following statement is equal to (12b):

$$\forall k \in \{0..k^*\} : \bigcup_m \{\vec{y}_a^{(m)}[k]\} \subseteq \mathcal{R}_a[k], \tag{16}$$

where the left side is the union of all trajectories deviating from the nominal behavior. Since $\mathcal{W}$ and $\mathcal{V}$ are zonotopes, we can apply propositions 2 and 3 to derive that the center and generator matrix of $\mathcal{R}_a[k] = (\vec{c}_k, G_k)$ are

$$\vec{c}_k := \begin{bmatrix} \sum_{i=0}^{k-1} E_i & 1 \end{bmatrix} \begin{bmatrix} \vec{c}_W \\ \vec{c}_V \end{bmatrix}, \quad E_i = CA^i, \tag{17}$$

$$G_k := \begin{bmatrix} E_0 G_W & \dots & E_{k-1} G_W & G_V \end{bmatrix}. \tag{18}$$

The second simplification is to formulate (16) as a set of linear inequalities by using the H-representation of $\mathcal{R}_a$ (see Proposition 1): if all $\vec{y}_a^{(m)}[k]$ satisfy all the half-space inequalities of $\mathcal{R}_a[k]$ for all $k$, then (16) follows, and the model is reachset conformant. As we will show in the following theorem, the half-space inequalities for $\mathcal{R}_a$ are not only linear in $\vec{y}_a^{(m)}$, but they are also linear in $\vec{\xi}$. The directions of the half-space normal vectors do not depend on $\vec{\xi}$, but our optimization in (12) is rather varying the distance of each half-space from the measured outputs. The number of test cases can be arbitrarily large since we will use the measurement with the largest deviation from the nominal behavior.

**Theorem 1.** *The constraint* (12b) *for the reachset conformance of linear systems is linear in* $\vec{\xi} = [\vec{c}_W, \vec{c}_V, \vec{\alpha}_W, \vec{\alpha}_V]^T$*:*

$$\forall k \in [0, k^*] : \max_m (N_k y_a^{(m)}[k]) \leq D_k \vec{\xi}, \tag{19}$$

*where* $N_k = [N_k^+, -N_k^+]^T$*, and* $D_k = [D_k^+, D_k^-]$*. The* $j$*-th row of* $N_k^+$*, where* $j \in 1..\binom{s}{\theta-1}$*, is a normal vector of the H-representation of* $\mathcal{R}_a[k]$ *and independent from* $\xi$*:*

$$\vec{n}_{j,k}^+ = \mathrm{nX}(G_k'^{\langle\gamma,\dots,\eta\rangle})^T / \| \mathrm{nX}(G_k'^{\langle\gamma,\dots,\eta\rangle}) \|_2,$$

*and the* $j$*-th row of* $D_k^+$ *and* $D_k^-$ *are defined as*

$$\vec{d}_{j,k}^+ = \Big[ \textstyle\sum_{i=0}^{k-1} \vec{n}_{j,k}^+ E_i, \quad \vec{n}_{j,k}^+, \quad \sum_{i=0}^{k-1} |\vec{n}_{j,k}^+ E_i G_W'|, \quad |\vec{n}_{j,k}^+ G_V'| \Big],$$

$$\vec{d}_{j,k}^- = \Big[ -\textstyle\sum_{i=0}^{k-1} \vec{n}_{j,k}^+ E_i, \quad -\vec{n}_{j,k}^+, \quad \sum_{i=0}^{k-1} |\vec{n}_{j,k}^+ E_i G_W'|, \quad |\vec{n}_{j,k}^+ G_V'| \Big].$$

*Proof.* We demonstrate that the normal vectors of the H-representation of any zonotope $(\vec{c}, G' \operatorname{diag}(\vec{\alpha}))$ are independent from $\vec{\alpha}$. The numerator of $\vec{n}_j^+$ is (see (1)):

$$\begin{aligned} &\mathrm{nX}(G \operatorname{diag}(\vec{\alpha})) = \\ &\quad = [\dots, (-1)^{i+1} \det(G^{[i]} \operatorname{diag}(\vec{\alpha})), \dots]^T \\ &\quad = \det(\operatorname{diag}(\vec{\alpha}))[\dots, (-1)^{i+1} \det(G^{[i]}), \dots]^T \\ &\quad = \det(\operatorname{diag}(\vec{\alpha})) \cdot \mathrm{nX}(G) = \rho\, \mathrm{nX}(G), \end{aligned}$$

and since all elements of $\vec{\alpha}$ are positive, we infer $\rho > 0$, and the denominator of $\vec{n}_j^+$ is

$$\left\| \rho\, \mathrm{nX}(G) \right\|_2 = \rho \| \mathrm{nX}(G) \|_2,$$

such that $\vec{\alpha}$ cancels out from the definition of $\vec{n}_j^+$ in (1). Next, we show that $\vec{d}_{j,k}^+$ and $\vec{d}_{j,k}^-$ can be derived from applying the definition of $G_k$ in (18) to (4) in Proposition 1, considering $\vec{1}$ as a vector of ones:

$$\begin{aligned} \Delta d_{j,k} &= \begin{bmatrix} |\vec{n}_{j,k}^+ E_0 G_W| & \dots & |\vec{n}_{j,k}^+ E_{k-1} G_W| & |\vec{n}_{j,k}^+ G_V| \end{bmatrix} \vec{1} \\ &= \begin{bmatrix} |\vec{n}_{j,k}^+ E_0 G_W'| & \dots & |\vec{n}_{j,k}^+ E_{k-1} G_W'| & |\vec{n}_{j,k}^+ G_V'| \end{bmatrix} \begin{bmatrix} \alpha_W \\ \vdots \\ \alpha_W \\ \alpha_V \end{bmatrix} \\ &= \begin{bmatrix} \sum_{i=0}^{k-1} |\vec{n}_{j,k}^+ E_i G_W'| & |\vec{n}_{j,k}^+ G_V'| \end{bmatrix} \begin{bmatrix} \alpha_W \\ \alpha_V \end{bmatrix}. \end{aligned}$$

The first two elements of $\vec{d}_{j,k}^+$ and $\vec{d}_{j,k}^-$ directly follow from (3), which are linear in the zonotope center $\vec{c}$, so that $\vec{d}_j^+$ and $\vec{d}_j^-$ in Proposition 1 are linear in $\vec{\xi}$ when $G_k$ is applied. □

One problem which we could encounter is that the number of constraints is $2\binom{p}{n-1}$ and exponentially increases with $k$ since $p$ (the number of generators of $G_k$ in (18)) grows for

each time step. The following corollary can be used for a conservative approximation of the linear inequality, which reduces the number of constraints, yet guarantees reachset conformance for $k^* \to \infty$. This is achieved by making the estimated states of the plant $\vec{x}^{(m)}$ conformant to the reachable set of the plant states. As the following proof will show, this reduces (12b) to a simple inclusion check for $\mathcal{W}$ and $\mathcal{V}$.

**Corollary 1.** *Let us consider sequential test cases. Then, a linear system is reachset conformant for $k^* \to \infty$, if*

$$\bigcup_{m\in\{1..M\}} \{\vec{x}_a^{(m)}[1]\} \subseteq \mathcal{W}, \tag{20}$$

$$\bigcup_{m\in\{1..M\}} \{\vec{y}^{(m)}[0] - C\vec{x}^{(m)}[0] - D\vec{u}^{(m)}[0]\} \subseteq \mathcal{V}, \tag{21}$$

*where*

$$\vec{x}_a^{(m)}[k] := \vec{x}^{(m)}[k] - \vec{x}_*^{(m)}[k] \tag{22}$$

*is the deviation from the nominal state: $\vec{x}_*^{(m)}[0] = \vec{x}^{(m)}[0]$ and $\vec{x}_*^{(m)}[1] = A\vec{x}^{(m)}[0] + Bu^{(m)}[0]$.*

*Proof.* We first rewrite the original problem before we perform the actual proof. Let us define a set $\mathcal{R}_{x,a}$ as the reachable set of states of $\vec{x}_a$, such that

$$\mathcal{R}_{x,a}[k+1] = A\mathcal{R}_{x,a}[k] \oplus \mathcal{W}, \quad \mathcal{R}_{x,a}[0] = \vec{x}_a[0] = \vec{0}. \tag{23}$$

The reachable output in (15) can thus be rewritten as

$$\mathcal{R}_a[k] = C\mathcal{R}_{x,a}[k] \oplus \mathcal{V},$$

and the definition of reachset conformance in (16) can be rewritten as

$$\forall k \in \mathbb{N}_0 : \bigcup_m \{C\vec{x}_a^{(m)}[k] + v[k]\} \subseteq C\mathcal{R}_{x,a}[k] \oplus \mathcal{V}. \tag{24}$$

We can derive reachset conformance by proving that the summands of (24) are conformant. Since (21) is given,

$$\bigcup_m v^{(m)}[k] \subseteq \mathcal{V} \tag{25}$$

for any $k$. Next, we show $\forall k : C\vec{x}_a^{(m)}[k] \in C\mathcal{R}_{x,a}[k]$. Here, we prove

$$\vec{x}_a^{(m)}[k] \in \mathcal{R}_{x,a}[k], \quad k \to \infty \tag{26}$$

by induction, when (20) is given. Using $u^{(m+k)}[0] = u^{(m)}[k]$ from Def. 4, we derive

$$\begin{aligned} &\vec{x}_*^{(m+k)}[1] - \vec{x}_*^{(m)}[k+1] \\ &= A\vec{x}^{(m+k)}[0] + Bu^{(m+k)}[0] - A\vec{x}_*^{(m)}[k] - Bu^{(m)}[k] \\ &= A(\vec{x}^{(m)}[k] - \vec{x}_*^{(m)}[k]) \\ &= A\vec{x}_a^{(m)}[k]. \end{aligned} \tag{27}$$

Since (20) is given, the base case for $k = 1$ holds:

$$\forall m : \vec{x}_a^{(m)}[1] \in A\mathcal{R}_{x,a}[0] \oplus \mathcal{W} = \mathcal{X}_a[1],$$

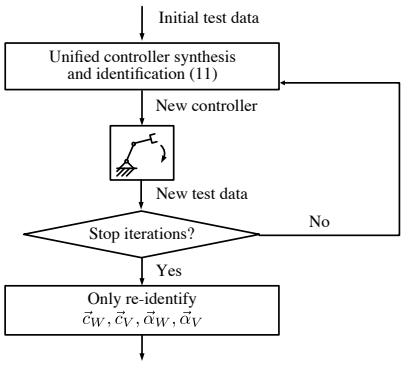


Fig. 2. Iterative procedure for simultaneous reachability-based identification and control synthesis.

because $\mathcal{X}_a[0] = \vec{0}$. Now we apply the induction step $k+1$:

$$\begin{aligned} \vec{x}_a^{(m)}[k+1] &\overset{(22)}{=} \vec{x}^{(m+k)}[1] - \vec{x}_*^{(m)}[k+1] \\ &= \vec{x}_a^{(m+k)}[1] + \vec{x}_*^{(m+k)}[1] - \vec{x}_*^{(m)}[k+1] \\ &\overset{(27)}{=} A\vec{x}_a^{(m)}[k] + \vec{x}_a^{(m+k)}[1] \overset{(20)}{\in} A\vec{x}_a^{(m)}[k] \oplus \mathcal{W} \\ &\overset{\text{induction hypothesis}}{\subseteq} A\mathcal{R}_{x,a}[k] \oplus \mathcal{W} \overset{(23)}{=} \mathcal{R}_{x,a}[k+1]. \end{aligned}$$

□

**Remark 1.** *The H-representation of $\mathcal{W}$ and $\mathcal{V}$ can also be used to formulate the constraints in the corollary as linear inequalities. The proof is similar to Theorem 1.*

**Remark 2.** *Using Corollary 1 to identify the disturbances is generally more conservative than using Theorem 1: if the column rank of $C$ is not full, then checking $\vec{x}_a \in \mathcal{R}_{x,a}$ is more strict than checking $\vec{y}_a \in \mathcal{R}_a = C\mathcal{R}_{x,a} \oplus \mathcal{V}$ because $C\mathcal{R}_{x,a}$ is a projected set. Another explanation is that the conformance of states constitutes a simulation relation [4, Sec. 3.3], which entails the conformance of outputs.*

**Remark 3.** *In practice, a threshold exists where any larger $k^*$ does not affect the results of Theorem 1 anymore. This threshold can be found by testing the synthesis with increasing $k^*$. For Corollary 1, this step is not required.*

### C. Iterative synthesis

As has been demonstrated in [60], the error of the nominal plant model can change depending on the chosen controller parameters, e.g., our nominal model does not consider flexible elements, which could lead to vibrations when controller parameters are ill-chosen. Since we use $\mathcal{W}_p$ and $\mathcal{V}_p$ to enclose the model errors, these sets, therefore, could also change depending on the controller parameters.

When we solve (12), a new set of controller parameters are proposed. We, therefore, need an iterative approach (see Fig. 2) that adjusts the sets $\mathcal{W}_p$ and $\mathcal{V}_p$ based on re-testing the real robot, which in turn influences the controller synthesis again. Similar to previous concepts in identification for control [47], we propose the following iterations:

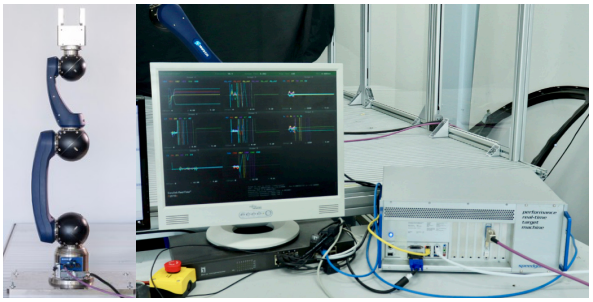

Fig. 3. The testbed consists of a Schunk LWA-4P 6-DOF robot manipulator and a controller running on Simulink Real-Time.

1) Given an initial set of test data obtained using an initial controller, we compute the linear constraint using Theorem 1 and solve (12a)–(12c) to synthesize an optimal controller.
2) Using the new controller, we repeat the tests on the real robot and obtain a new set of test data.
3) Repeat step 1 with the new data to synthesize a new controller. If no further iteration is desired, then we perform a re-identification of the disturbances only, i.e., solving (12a)–(12c) without changing the controller, to obtain the final result.

### D. Computational aspects

The optimization problem in (12) is posed as a nonlinear program with a nonlinear cost function (12a). If a solution exists, we are able to check reachset conformance (12b) and satisfy input constraints (12c). However, we cannot guarantee convergence to a globally optimal solution; we can only expect to obtain a local optimum. Nevertheless, practical tuning rules can be helpful in improving convergence, e.g., consider a static feedback controller [6, eq. 8.58] that we will consider in Sec. IV: $u_p = \ddot{q}_d + k_p(q_d - q) + k_d(\dot{q}_d - \dot{q})$. By replacing the parameters $k_p = \omega^2$ and $k_d = 2\zeta\omega$ with the natural frequency $\omega$ and damping ratio $\zeta$, the convergence improved. Such tuning rules were initially developed for manual tuning to converge faster to an optimal solution and can obviously also serve as hints to improve convergence for our automatic approach.

We cannot provide concrete complexity bounds for nonlinear programming since no bounds exist for them. Nevertheless, let us give an idea of the complexity of the different evaluations. The cost (12a) and the constraint function (12c) mainly involve computing reachable sets and some algebraic operations on the resulting zonotopes, which together have a complexity of $\mathcal{O}(n^3)$ [75], where $n$ is the number of states. The conformance constraints in (12b) can be efficiently evaluated since they are linear inequalities. Checking the constraint $\mathcal{R}_u \subseteq \mathcal{U}_p$ in (12c) requires only checking if a zonotope is inside a polytope, which can also be efficiently computed [75, Lemma 2].

## IV. Experiments on a 6-axis robot manipulator

In this section, we show the results of applying our combined controller synthesis and reachset-conformant identification to a real 6-axis robot manipulator (see Fig. 3). In the first experiment in Sec. IV-A, we work out the benefits of using the combined approach in comparison to separate identification and synthesis. In the second experiment in Sec. IV-B, we compare our method against the linear-quadratic-Gaussian control (LQG). In the third experiment in Sec. IV-C, we demonstrate how our method makes it possible to compare the guarantees of different controllers.

TABLE I
Deriving $\mathcal{U}_p$ from specified robot limits using (13)

| Axis | $\vec{\tau}_{\max}$ | $\vec{q}_{\max}$ | $\dot{\vec{q}}_{\max}$ | $\mathcal{U}_p$ satisfying (13) |
|---|---|---|---|---|
| 1 | 160 Nm | 140° | 0.7 rad/s | $[-26, 26]$ rad/$s^2$ |
| 2 | 160 Nm | 45° | 0.7 rad/s | $[-26, 26]$ rad/$s^2$ |
| 3 | 160 Nm | 100° | 0.7 rad/s | $[-26, 26]$ rad/$s^2$ |
| 4 | 160 Nm | 140° | 0.7 rad/s | $[-26, 26]$ rad/$s^2$ |
| 5 | 40 Nm | 80° | 0.7 rad/s | $[-100, 100]$ rad/$s^2$ |
| 6 | 40 Nm | 140° | 0.7 rad/s | $[-100, 100]$ rad/$s^2$ |

The data for testing reachset conformance were obtained from the real robot running closed-loop trapezoidal and polynomial trajectories[2] with random target positions, velocities, and accelerations up to $\ddot{q}_d \in \mathcal{U}_{\text{ref}} = [-2, 2]$ rad/$s^2$. The total duration of the dataset is 33 minutes and 20 seconds. Each sampling instant is considered a starting point of a new test case, resulting in 497,880 test cases for each robot joint. Other test selection methods (e.g., [4], [76]) can be used to find test cases that explore edge scenarios more effectively; however, a basic approach—such as random testing—may already be sufficient. An inherent problem with testing will always be that there are cases that are not covered by the tested trajectories. In addition, changes to the robot dynamics can happen that are also not covered by the test cases. We propose to implement (12b) as an online conformance monitor that detects non-conformant measurements, transitions the system to a safe stop, and repeats identification for this new test case. If the resulting new disturbance violates the input constraint in (12c), the controller synthesis needs to be repeated.

The time horizon for reachset conformance has been selected to be $k^* = 125$. At a sampling time $\Delta t = 0.004$ s, this amounts to 0.5 seconds. Because $y_p$ is one-dimensional, this amounts to 252 conformance constraints (two half-spaces per time step, including $k = 0$). To check whether a selected $\mathcal{U}_p$ satisfies the allowed set of joint torques $\mathcal{T} := [-\vec{\tau}_{\max}, \vec{\tau}_{\max}]$, we set $\mathcal{Q} := [-\vec{q}_{\max}, \vec{q}_{\max}]$ and $d\mathcal{Q} := [-\dot{\vec{q}}_{\max}, \dot{\vec{q}}_{\max}]$ and evaluate (13) using tenth-order Taylor models [71]. The values can be seen in Table I. To avoid the wrapping effect, which accumulates approximation errors, we split $\mathcal{Q}$ into four intervals and evaluate (13) for each interval combination.

### A. Combined vs. separate identification and synthesis

In the first experiment, we compare our combined approach against a separate approach, where a reachset-conformant model is identified before the controller synthesis. The

[2]A video showing the initial tests, and the code for reproducing all experiments are provided within the supplementary materials.

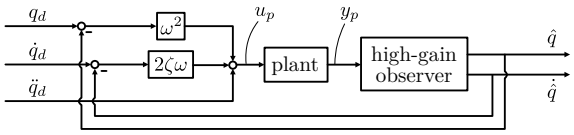


Fig. 4. The control loop considered in Sec. IV-A tracks the reference $\vec{y}_{\text{ref}} = [q_d, \dot{q}_d]$ and has an input feedforward $u_{\text{ff}} = \ddot{q}_d$. A high-gain observer [77] is used to observe the position $q$ and velocity $\dot{q}$, which are used in the feedback control.

TABLE II
OPTIMALLY SYNTHESIZED STATIC FEEDBACK $(\omega, \zeta)$ AND IDENTIFIED MODEL UNCERTAINTIES $(\alpha_{W,1}, \alpha_{W,2}, \alpha_{V,1})$

| Axis | $\|\mathcal{R}_e\|$ | $\omega$ | $\zeta$ | $\alpha_{W_p,1}$ | $\alpha_{W_p,2}$ | $\alpha_{V_p,1}$ |
|---|---|---|---|---|---|---|
| 1 | 0.62 | 24.40 | 0.96 | 0.0348 | 3.59 | $3.49 \cdot 10^{-5}$ |
| 2 | 0.70 | 24.39 | 0.96 | 0.0442 | 3.82 | $3.49 \cdot 10^{-5}$ |
| 3 | 0.58 | 23.06 | 0.91 | 0.0381 | 2.93 | $3.49 \cdot 10^{-5}$ |
| 4 | 0.58 | 23.81 | 0.96 | 0.0363 | 3.00 | $3.49 \cdot 10^{-5}$ |
| 5 | 0.91 | 24.71 | 1.00 | 0.0142 | 7.72 | $3.49 \cdot 10^{-5}$ |
| 6 | 2.39 | 25.23 | 1.00 | 0.0000 | 23.28 | $2.80 \cdot 10^{-5}$ |

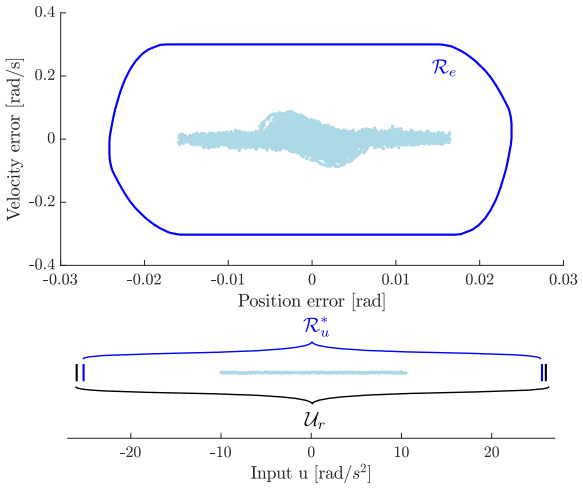


Fig. 5. After solving (12), both the computed reachable tracking error $\mathcal{R}_e$ and the computed reachable input $\mathcal{R}_u$ enclose their measured counterparts, while $\mathcal{R}_u$ satisfies the input constraint.

controller-observer structure for this experiment is depicted in Fig. 4 and is chosen as follows: a high-gain observer [77] uses the plant output $y_p$ to estimate $\hat{q}$ and $\dot{\hat{q}}$:

$$\begin{bmatrix} \dot{\hat{q}} \\ \ddot{\hat{q}} \end{bmatrix} = \begin{bmatrix} -h_1/\epsilon & 1 \\ -h_2/\epsilon^2 & 0 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \dot{\hat{q}} \end{bmatrix} + \begin{bmatrix} h_1/\epsilon \\ h_2/\epsilon^2 \end{bmatrix} y_p, \tag{28}$$

where $h_1 = 15, h_2 = 30$, and $\epsilon := 0.01$ are the gains. To discretize the observer, we use the bilinear transformation discussed in [78]. As the controller, we consider a static feedback one [6, eq. 8.58]:

$$u_p = \ddot{q}_d + \omega^2(q_d - \hat{q}) + 2\zeta\omega(\dot{q}_d - \dot{\hat{q}}), \tag{29}$$

where $\omega$ and $\zeta$ are the parameters to be optimized. For the combined approach, we set $\vec{p} = [\omega, \zeta, \alpha_{W_p,1}, \alpha_{W_p,2}, \alpha_{V_p}]^T$ and solve (12) for two iterations. The final result can be seen in Tab. II, and we plot $\mathcal{R}_e$ and $\mathcal{R}_u$ for the first robot axis in Fig. 5. Our combined approach returned feasible solutions for all six axes. The uncertainties for axes 5 and 6 are larger than others, mainly due to the inaccuracy of the feedback linearization for these axes. As Fig. 5 shows, our reachable sets correctly predict the real tracking errors and the real inputs.

For the separate approach, we first identify a reachset-conformant model by solving an optimization problem, where $\|\mathcal{W}_p\| + \|\mathcal{V}_p\|$ is set as the cost function and (12b) is set as the constraint function, and $\alpha_{W_p,1}, \alpha_{W_p,2}, \alpha_{V_p}$ are the parameters. For the subsequent controller synthesis, we set (12a) as the cost, (12c) as the constraint, and $\omega, \zeta$ as the parameters. The plots in Fig. 6 show that the separate approach leads to a significantly larger reachable set $\mathcal{R}_e$, although the identified values $\alpha_{W_p,1} = 0.75$ and $\alpha_{W_p,2} = 0$ for axis 1 lead to a smaller value of $\|\mathcal{W}_p\|$ than the values identified in the combined approach $\alpha_{W_p,1} = 0.0348$ and $\alpha_{W_p,2} = 3.59$ for axis 1. This is because the combined approach optimally balances the disturbance parameters to ultimately converge to the smallest reachable tracking error.

### B. Our method vs. LQG control

The linear-quadratic-Gaussian control (LQG) [79] is an optimization-based design approach, where the full state is estimated via a Kalman filter and state-feedback is generated, such that a cost function with weighting factors $Q$ and $r$ is minimized:

$$J = \sum_{k=0}^{\infty} (\vec{e}[k]^T Q \vec{e}[k] + u_p[k] r u_p[k]), \tag{30}$$

where $\vec{e} = \vec{x}_r - \vec{x}_{\text{ref}}$ is the state tracking error, and $\vec{x}_{\text{ref}}[k] = [q_d[k-1], q_d[k], \dot{q}_d[k], \ddot{q}_d[k-1]]^T$ is the state reference. The Kalman filter assumes uncertainties in the model using zero-mean Gaussian noises with covariance matrices $S_W$ for the process and $S_V$ for the measurement, respectively. Here, we set $S_W = (1/3 \cdot G_{W_p} G_{W_p}^T)^2$ and $S_V = (1/3 \cdot G_{V_p} G_{V_p}^T)^2$, which assumes that the zero-centered sets $\mathcal{W}_p$ and $\mathcal{V}_p$ represent three times the standard deviation. We apply the `lqg` function from MATLAB and use our model from (11) for the design.

LQG relies on the user to set the weights in $Q$ and $r$. This is a difficult task, especially when there are input constraints to consider because, normally, the only way to determine whether a controller is feasible and desirable is to test it on the real system. Our paper realizes a different solution: using the reachset-conformant model from Tab. II, we can evaluate whether a possible weight combination may lead to an infeasible controller. To demonstrate this, we set $Q = \text{diag}(1000, 1000, 0.01, r)$ and compute the reachable sets by varying $r$. For axis 1, we display the results in Tab. III and the sets $\mathcal{R}_e$ are also visualized in Fig. 7, including the reachable set obtained from Sec. IV-A using combined synthesis.

The results show that if a high $r$ is set, then a weak controller is obtained, resulting in a large tracking error, but we receive the smallest $\mathcal{R}_u$ interval. The more $r$ is decreased, the more the tracking error improves. However, at $r$ near zero,

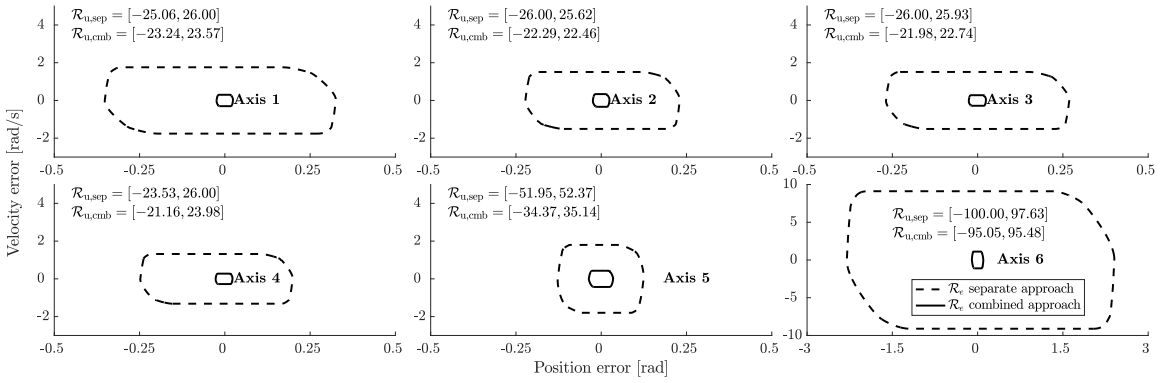


Fig. 6. Comparing $\mathcal{R}_e$ and $\mathcal{R}_u$ for the separate approach against the combined approach for identification and synthesis. A separate identification may lead to suboptimal $\mathcal{W}_p$ and $\mathcal{V}_p$, such that the closed-loop reachable sets become unnecessarily large. In the separate approach, the controller synthesis converged to smaller gains, e.g., $\omega = 5.9$ for axis 1 to satisfy the input constraint; compared to the combined approach, where for axis 1, $\omega = 24.4$.

TABLE III
COMPARISON: LQG OPTIMIZATION WITH $Q = \mathrm{diag}(1000, 1000, 0.01)$ VS. OUR CONTROLLER SYNTHESIS METHOD FOR ROBOT AXIS 1

| $R$ | $\|\mathcal{R}_e\|$ | $\mathcal{R}_u$ |
|---|---|---|
| $10^{-7}$ | 0.831 | $[-75.37, 75.70]$ |
| **our method** | **0.625** | $[\mathbf{-23.24, 23.57}]$ |
| $10^{-4}$ | 0.762 | $[-24.40, 24.72]$ |
| 0.01 | 1.091 | $[-17.12, 17.45]$ |

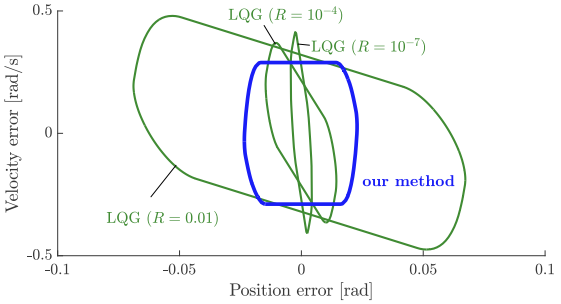


Fig. 7. Comparison of the reachable tracking error $\mathcal{R}_e$ for controllers obtained with LQG optimization (green) and our method from Sec. IV-A (blue).

the input constraints are violated. Instead, our optimization-based approach not only satisfies the input constraint but can use any controller and observer, while LQG is restricted to a state feedback design. As we described earlier, LQG requires test iterations to validate different combinations of possible $Q$ and $r$ and their resulting closed-loop performance, while our method requires test iterations only to make sure that the model remains conformant. As Sec. IV-A showed, two iterations can be sufficient here.

### C. Comparing static feedback vs. disturbance-compensated feedback

In the third experiment, we will demonstrate that our method is generalizable to other controllers besides the one

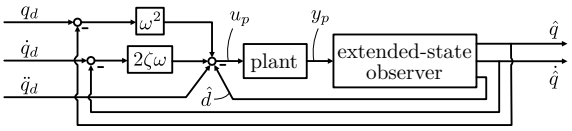


Fig. 8. The control loop considered in Sec. IV-C tracks the reference $\vec{y}_{\mathrm{ref}} = [q_d, \dot{q}_d]$ and has an input feedforward $u_{\mathrm{ff}} = \ddot{q}_d$. An extended state observer [80] is used to observe the position $q$, velocity $\dot{q}$, and the disturbance $d$, which are used in the feedback control.

specified in the previous two experiments. In the following, we synthesize an observer-based feedback control law with disturbance compensation

$$u_p = \ddot{q}_d + \omega^2(\dot{q}_d - \dot{\hat{q}}) + 2\zeta\omega(q_d - \hat{q}) - \hat{d}, \tag{31}$$

where $\hat{q}, \dot{\hat{q}}$, and $\hat{d}$ are estimated by an extended-state observer (ESO) [80]:

$$\begin{bmatrix} \dot{\hat{q}} \\ \ddot{\hat{q}} \\ \dot{\hat{d}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{q} \\ \dot{\hat{q}} \\ \hat{d} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u_p + \begin{bmatrix} l_1/\epsilon \\ l_2/\epsilon^2 \\ l_3/\epsilon^3 \end{bmatrix} (q - \hat{q}). \tag{32}$$

We compare this new controller against the one from the previous experiments. For the sake of brevity, we set $\omega = 20$ and $\zeta = 1$ and only synthesize $h_1, h_2$ for the high-gain observer and $l_1, l_2, l_3$ for the extended-state observer, as well as the model uncertainties $\alpha_{W_p,1}, \alpha_{W_p,2}, \alpha_{V_p,1}$, respectively. We perform two iterations for each method: the results for the respective optimal parameters are shown in Tab. IV and the reachable set, as well as the measured tracking errors from the real robot, are shown in Fig. 9.

As the plotted reachable tracking errors show, the extended-state observers help to significantly improve the position error of the real robot, while the velocity error stays similar to the high-gain observer. As can be seen in the plots, the guarantees for the tracking error reflect a similar behavior. Axis 5 and 6 of our robot perform badly mainly due to insufficient feedback linearization. Nevertheless, the identified model remains conformant, and the reachable tracking error

is correctly predicted. What is also noticeable is that the identified uncertain parameters $\alpha$ differ depending on the controller, e.g., for axis 6, $\alpha_{W_p,1}$ is larger for HG, while $\alpha_{W_p,2}$ is larger for ESO. One reason is that our controller synthesis chooses the optimal value that minimizes $\|\mathcal{R}_e\|$. Another reason is that the disturbance also depends on the controller since different controllers can suppress disturbances differently, e.g., the $\alpha_{W_p,2}$ are larger when using ESO, but the feedback law in (31) is able to compensate for it, resulting in a smaller positional tracking error.

We summarize the experimental results of our combined controller synthesis and reachset-conformant identification. We demonstrated in Sec. IV-A that a combined approach is necessary to avoid conservative results. In Sec. IV-B, we showed that LQG methods require careful balancing of the tracking error and the input effort, while our approach automatically satisfies the input constraints. In Sec. IV-C, we showed that our approach could be used for any controller structure as long as the closed-loop dynamics are linear. The experiment has also shown that although the observers do not consider the full dynamics of the plant, it is still possible to derive guarantees, and the soundness of our approach is not affected. Rather, we have shown for our robot that an observer with a better model may lead to a better performance of the closed-loop system.

## V. Conclusion

In this paper, we have shown that our method can be used to optimally design a controller and to derive guarantees for the input constraint and tracking error. In contrast to previous work, these guarantees are also applicable to real robotic systems. Using our method, we can now formally analyze any linear robotic controller for their safety.

The formal relation between the robot model and the real system is established by identifying reachset-conformant model parameters. The controller synthesis and identification are unified into a single optimization, which means that the model and controller are both optimized for the smallest reachable tracking error. Our experiments have shown that the computed reachable sets always successfully enclose all behaviors of the real robot system, however large the disturbance in the system is. Our approach does not require tuning of hyper-parameters, in contrast to LQR. We have shown the effectiveness of our novel approach to synthesizing different feedback laws.

Our method can be applied to any robot in practice that uses feedback linearization, linear observers, and feedback controllers. In the future, we would like to extend this approach to nonlinear plant models and controllers.

## Appendix A
## Full robot model including disturbance

To model the disturbance of the system, we make the following assumptions: 1) the velocity is disturbed by an interval $[-\alpha_{W_p,1}, \alpha_{W_p,1}]$, 2) the acceleration is disturbed by an interval $[-\alpha_{W_p,2}, \alpha_{W_p,2}]$, 3) the measurement is disturbed by an interval $[-\alpha_{V_p}, \alpha_{V_p}]$, and 4) we consider an additional disturbance state, such that $\ddot{q} = u_p + d$ and $\dot{d} = 0$. The full model of the plant for each robot joint, including the uncertainties, is described by the following linear system:

$$\begin{aligned}\vec{x}_p[k+1] &= \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & \Delta t^2/2 & \Delta t^2/2 \\ 0 & 0 & 1 & \Delta t & \Delta t \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}_p[k] + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u_p[k] \\ &\quad + \vec{w}_p[k], \\ y_p[k] &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}_p[k] + v_p[k],\end{aligned}$$

where $\vec{x}_p[k] = [q[k-1], q[k], \dot{q}[k], d[k], u_p[k-1]]^T$, $\vec{w}_p[k] \in \mathcal{W}_p$, and $\vec{v}_p[k] \in \mathcal{V}_p$:

$$\mathcal{W}_p = \left( \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ \Delta t & \Delta t^2/2 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_{W_p,1} \\ \alpha_{W_p,2} \end{bmatrix} \right), \mathcal{V}_p = \left(0, \alpha_{V_p}\right),$$

where $\alpha_{W_p,1}$, $\alpha_{W_p,2}$, and $\alpha_{V_p}$ are the scaling factors of the zonotopes $\mathcal{W}_p$ and $\mathcal{V}_p$. The generator matrix of $\mathcal{W}_p$ is a discretization similar to the plant linear dynamics.

## Appendix B
## Analysis of the reference error

The vector $\tilde{\vec{y}}_{\mathrm{cl,ref}}$ is computed using (7) considering $\vec{y}_{\mathrm{ref}}$ and $u_{\mathrm{ff}}$ and considering $\vec{w}_p = \vec{0}, v_p = 0$. The result is a trajectory that tracks the reference with a *reference error*, which we define as $\vec{y}_{\mathrm{e,ref}}$ and $u_{\mathrm{e,ff}}$ such that

$$\tilde{\vec{y}}_{\mathrm{cl,ref}} = \begin{bmatrix} \vec{y}_{\mathrm{ref}} + \vec{y}_{\mathrm{e,ref}} \\ u_{\mathrm{ff}} + u_{\mathrm{e,ff}} \end{bmatrix}. \tag{33}$$

In cases where $u_{\mathrm{ff}}$ is the output of the inverted plant model [81] given $y_{\mathrm{ref}}$ as an input, there will be no reference error. A simple example is a double-integrator model $\ddot{q} = u_p$, where the output is $\vec{y}_p = [q, \dot{q}]^T$. Applying $u_p = \ddot{q}_d$ would exactly produce the reference $\vec{y}_p = [q_d, \dot{q}_d]$ without any error. In any other case, the tracking error increases by $\vec{y}_{\mathrm{e,ref}}$ and thus requires an additional input $u_{\mathrm{e,ff}} = -K\vec{y}_{\mathrm{e,ref}}$, where $K = [\omega^2, 2\zeta\omega]$, to compensate for the additional tracking error. In some cases, the additional input could lead to a violation of the input constraint: $u_{\mathrm{e,ff}} \oplus \mathcal{R}_u \not\subset \mathcal{U}_p$. In the following paragraphs, we present three different ways to deal with the reference error to arrive at an actual reachable tracking error $\mathcal{R}_e^*$ and reachable input $\mathcal{R}_u^*$:

*1) Tracking $\vec{y}_{\mathrm{cl,ref}}$ instead of $\vec{y}_{\mathrm{ref}}$:* Let us rewrite the control law in (29), considering $\vec{y}_{\mathrm{ref}} = [q_d, \dot{q}_d]^T$, $\vec{y}_{\mathrm{cl}} = [\hat{q}, \dot{\hat{q}}]^T$, $K = [\omega^2, 2\zeta\omega]$, and $\vec{y}_e \in \mathcal{R}_e$ such that

$$\begin{aligned} u_p &= u_{\mathrm{ff}} + K(\vec{y}_{\mathrm{ref}} - \vec{y}_{\mathrm{cl}}) \\ &= u_{\mathrm{ff}} + K(\vec{y}_{\mathrm{ref}} - (\vec{y}_{\mathrm{ref}} + \vec{y}_{\mathrm{e,ref}} + \vec{y}_e)) \\ &= u_{\mathrm{ff}} - K(\vec{y}_{\mathrm{e,ref}} + \vec{y}_e), = u_{\mathrm{ff}} + u_{\mathrm{e,ff}} - K\vec{y}_e. \end{aligned}$$

We slightly modify the static-feedback control law to track $\vec{y}_{\mathrm{cl,ref}}$ instead of $\vec{y}_{\mathrm{ref}}$ such that

$$\begin{aligned} u_p^* &= u_{\mathrm{ff}} + K(\vec{y}_{\mathrm{cl,ref}} - \vec{y}_{\mathrm{cl}}) \\ &= u_{\mathrm{ff}} + K(\vec{y}_{\mathrm{ref}} + \vec{y}_{\mathrm{e,ref}} - (\vec{y}_{\mathrm{ref}} + \vec{y}_{\mathrm{e,ref}} + \vec{y}_e)) \\ &= u_{\mathrm{ff}} - K\vec{y}_e. \end{aligned}$$

TABLE IV
COMPARING SYNTHESIZED HIGH-GAIN OBSERVERS WITH SYNTHESIZED EXTENDED-STATE OBSERVERS

| | High-Gain Observer | | | | | | Extended-State Observer | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Axis | $\|\mathcal{R}_e\|$ | $h_1$ | $h_2$ | $\alpha_{W_p,1}$ | $\alpha_{W_p,2}$ | $\alpha_{V_p,1}$ | $\|\mathcal{R}_e\|$ | $l_1$ | $l_2$ | $l_3$ | $\alpha_{W_p,1}$ | $\alpha_{W_p,2}$ | $\alpha_{V_p,1}$ |
| 1 | 0.373 | 135.0 | 416.1 | 0.0417 | 0.976 | $8.73 \cdot 10^{-6}$ | 0.535 | 57.1 | 103.3 | 18.1 | 0.0234 | 3.587 | $3.49 \cdot 10^{-5}$ |
| 2 | 0.325 | 135.1 | 416.4 | 0.0397 | 0.652 | $8.73 \cdot 10^{-6}$ | 0.662 | 93.5 | 182.0 | 37.9 | 0.0362 | 4.110 | $3.49 \cdot 10^{-5}$ |
| 3 | 0.409 | 135.0 | 416.1 | 0.0539 | 0.679 | $8.73 \cdot 10^{-6}$ | 0.534 | 75.6 | 113.7 | 16.5 | 0.0358 | 2.951 | $3.49 \cdot 10^{-5}$ |
| 4 | 0.428 | 135.0 | 416.1 | 0.0540 | 0.881 | $8.73 \cdot 10^{-6}$ | 0.647 | 80.5 | 153.4 | 30.1 | 0.0283 | 4.415 | $3.49 \cdot 10^{-5}$ |
| 5 | 0.542 | 103.6 | 517.8 | 0.0344 | 2.780 | $8.73 \cdot 10^{-6}$ | 1.573 | 37.6 | 490.0 | 210.6 | 0.0055 | 17.936 | $8.73 \cdot 10^{-6}$ |
| 6 | 1.297 | 105.5 | 602.4 | 0.0309 | 9.379 | $8.73 \cdot 10^{-6}$ | 4.582 | 23.1 | 638.9 | 338.7 | 0 | 55.742 | $8.73 \cdot 10^{-6}$ |

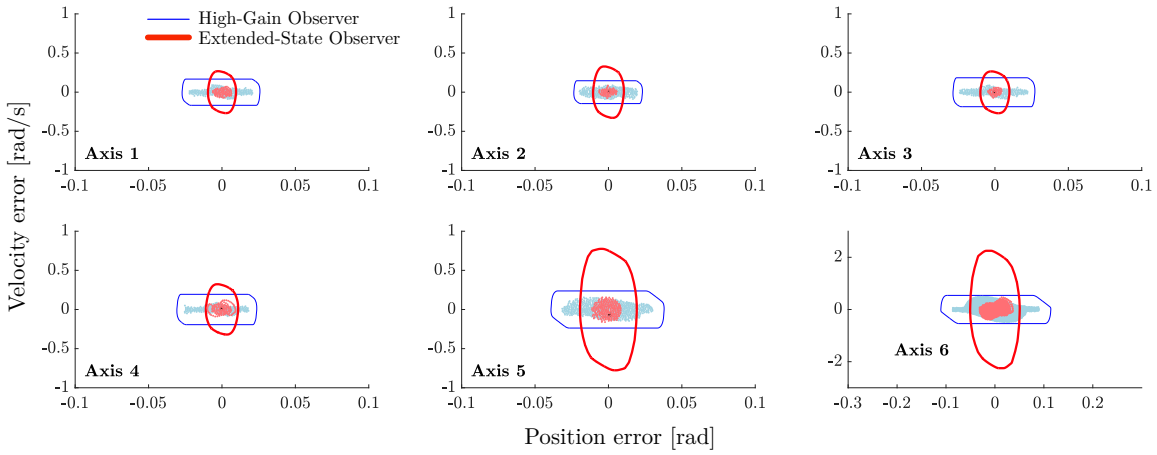


Fig. 9. Comparison of the guaranteed tracking error $\mathcal{R}_e$ for controllers using a high-gain observer (blue) and an extended-state observer (red).

By definition, $u_{\mathrm{e,ff}}$ vanishes using the new control law, and the input constraint cannot be violated anymore. Since no uncertainty is involved in obtaining $\vec{y}_{\mathrm{cl,ref}}$ and $\vec{y}_{\mathrm{e,ref}}$, they can be exactly precomputed before executing a trajectory. The actual reachable sets are then defined as

$$\mathcal{R}_e^* := \vec{y}_{\mathrm{e,ref}} \oplus \mathcal{R}_e,$$
$$\mathcal{R}_u^* := \mathcal{R}_u.$$

The advantage is that the input constraint is guaranteed independently of the desired trajectory. The disadvantage, however, is that we deviate from the original control law, and that $\mathcal{R}_e$ is relative to $\vec{y}_{\mathrm{cl,ref}}$ instead of $\vec{y}_{\mathrm{ref}}$.

*2) Precomputing the reference error:* As no uncertainty is involved, $\vec{y}_{\mathrm{e,ref}}$ and $u_{\mathrm{e,ff}}$ can be precomputed before executing a trajectory. We define the actual reference-dependent sets as

$$\mathcal{R}_e^* := \vec{y}_{\mathrm{e,ref}} \oplus \mathcal{R}_e,$$
$$\mathcal{R}_u^* := u_{\mathrm{e,ff}} \oplus \mathcal{R}_u.$$

The disadvantage of this approach is, however, that the input constraint cannot be guaranteed at all times; $\mathcal{R}_u^* \subseteq \mathcal{U}_p$ must be checked before every execution of a trajectory on the robot. We only recommend this approach if the controller is designed for a single reference trajectory.

*3) Solve* (12) *for a predefined set of references:* In this approach, we predefine a large set of reference trajectories before solving (12), e.g., we can use the same trajectories from the test cases used to identify the disturbances. We then compute two sets $\mathcal{Y}_{\mathrm{e,ref}}$ and $\mathcal{U}_{\mathrm{e,ff}}$, that enclose all $\vec{y}_{\mathrm{e,ref}}$ and $u_{\mathrm{e,ff}}$ for all references. The actual reachable sets are then defined as

$$\mathcal{R}_e^* := \mathcal{Y}_{\mathrm{e,ref}} \oplus \mathcal{R}_e,$$
$$\mathcal{R}_u^* := \mathcal{U}_{\mathrm{e,ff}} \oplus \mathcal{R}_u,$$

and replace $\mathcal{R}_e$ and $\mathcal{R}_u$ when solving (12). The advantage is that the input constraint is guaranteed for all considered references, and also all non-considered references where $u_{\mathrm{ff}} \in \mathcal{U}_{\mathrm{e,ff}}$, while the effort for solving (12) is only slightly increased. We used this method in our experiments in Sec. IV. We recommend this approach if the controller is designed for unknown references, but when the method for reference generation stays similar, e.g., always $u_{\mathrm{ff}}[k] = \ddot{q}_d[k]$, or $u_{\mathrm{ff}}[k] = \ddot{q}_d[k+2]$ to consider delays in the plant. However, during pre-computation, sufficiently many reference trajectories are necessary so that the largest possible $\mathcal{U}_{\mathrm{e,ff}}$ can be found.

## REFERENCES

[1] C. Abdallah, D. M. Dawson, P. Dorato, and M. Jamshidi, "Survey of robust control for rigid robots," *IEEE Control Systems Magazine*, vol. 11, no. 2, pp. 24–30, 1991.

[2] H. G. Sage, M. F. De Mathelin, and E. Ostertag, "Robust control of robot manipulators: A survey," *Int. Journal of Control*, vol. 72, no. 16, pp. 1498–1522, 1999.
[3] M. Althoff, G. Frehse, and A. Girard, "Set propagation techniques for reachability analysis," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 369–395, 2021.
[4] H. Roehm, J. Oehlerking, M. Woehrle, and M. Althoff, "Model Conformance for Cyber-Physical Systems," *ACM Trans. on Cyber-Physical Systems*, vol. 3, no. 3, pp. 1–26, 2019.
[5] Z. Qu, J. F. Dorsey, X. Zhang, and D. M. Dawson, "Robust control of robots by the computed torque law," *Systems and Control Letters*, vol. 16, no. 1, pp. 25–32, 1991.
[6] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. London, UK: Springer London, 2009.
[7] S. Zenieh and M. Corless, "Simple Robust r-$\alpha$ Tracking Controllers for Uncertain Fully-Actuated Mechanical Systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 119, no. 4, pp. 821–825, 1997.
[8] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear H-infinity optimality to robot controllers," *IEEE Trans. on Robotics*, vol. 31, no. 3, pp. 682–698, 2015.
[9] M. Makarov, M. Grossard, P. Rodríguez-Ayerbe, and D. Dumur, "Modeling and Preview H-infinity Control Design for Motion Control of Elastic-Joint Robots with Uncertainties," *IEEE Trans. on Industrial Electronics*, vol. 63, no. 10, pp. 6429–6438, 2016.
[10] Feng Lin and R. Brandt, "An optimal control approach to robust control of robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 69–77, 1998.
[11] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.
[12] W. Langson, I. Chryssochoos, S. Raković, and D. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.
[13] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, "Parameterized tube model predictive control," *IEEE Trans. on Automatic Control*, vol. 57, no. 11, pp. 2746–2761, 2012.
[14] S. V. Raković, B. Kouvaritakis, R. Findeisen, and M. Cannon, "Homothetic tube model predictive control," *Automatica*, vol. 48, no. 8, pp. 1631–1638, 2012.
[15] M. Rubagotti, D. M. Raimondo, A. Ferrara, and L. Magni, "Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 56, no. 3, pp. 556–570, 2011.
[16] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer, "Robust model predictive control for nonlinear discrete-time systems," *Int. Journal of Robust and Nonlinear Control*, vol. 13, no. 3-4, pp. 229–246, 2003.
[17] D. Q. Mayne, E. C. Kerrigan, E. J. van Wyk, and P. Falugi, "Tube-based robust nonlinear model predictive control," *Int. Journal of Robust and Nonlinear Control*, vol. 21, no. 11, pp. 1341–1353, 2011.
[18] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2017, pp. 5883–5890.
[19] J. Wolff and M. Buss, "Invariance control design for constrained nonlinear systems," *IFAC Proceedings Volumes*, vol. 38, no. 1, pp. 37–42, 2005, 16th IFAC World Congress.
[20] M. Kimmel and S. Hirche, "Invariance control with chattering reduction," in *Proc. IEEE Conf. on Decision and Control*, 2014, pp. 68–74.
[21] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
[22] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.
[23] H. Kress-Gazit, M. Lahijanian, and V. Raman, "Synthesis for Robots: Guarantees and Feedback for Robot Behavior," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 211–236, 2018.
[24] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Trans. on Automatic Control*, vol. 53, no. 1, pp. 287–297, 2008.
[25] M. Zamani, G. Pola, M. Mazo Jr., and P. Tabuada, "Symbolic models for nonlinear control systems without stability assumptions," *IEEE Trans. on Automatic Control*, vol. 57, no. 7, pp. 1804–1809, 2012.
[26] J. A. DeCastro and H. Kress-Gazit, "Synthesis of nonlinear continuous controllers for verifiably correct high-level, reactive behaviors," *The Int. Journal of Robotics Research*, vol. 34, no. 3, pp. 378–394, 2015.
[27] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.
[28] A. Girard, "Controller synthesis for safety and reachability via approximate bisimulation," *Automatica*, vol. 48, no. 5, pp. 947–953, 2012.
[29] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Trans. on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.
[30] J. Liu, N. Ozay, U. Topcu, and R. M. Murray, "Synthesis of reactive switching protocols from temporal logic specifications," *IEEE Trans. on Automatic Control*, vol. 58, no. 7, pp. 1771–1785, 2013.
[31] J. Liu and N. Ozay, "Finite abstractions with robustness margins for temporal logic-based control synthesis," *Nonlinear Analysis: Hybrid Systems*, vol. 22, pp. 1–15, 2016.
[32] G. Pola, A. Girard, and P. Tabuada, "Symbolic models for nonlinear control systems using approximate bisimulation," in *Proc. IEEE Conf. on Decision and Control*, 2007, pp. 4656–4661.
[33] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*, 2015, pp. 239–248.
[34] M. Rungger, M. Mazo Jr., and P. Tabuada, "Specification-guided controller synthesis for linear systems and safe linear-time temporal logic," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*, 2013, pp. 333–342.
[35] M. Zamani, A. Abate, and A. Girard, "Symbolic models for stochastic switched systems: A discretization and a discretization-free approach," *Automatica*, vol. 55, pp. 183–196, 2015.
[36] E. M. Wolff and R. M. Murray, "Optimal Control of Nonlinear Systems with Temporal Logic Specifications," in *Robotics Research: 16th Int. Symposium ISRR*. Cham: Springer Int. Publishing, 2016, pp. 21–37.
[37] J. A. DeCastro and H. Kress-Gazit, "Nonlinear Controller Synthesis and Automatic Workspace Partitioning for Reactive High-Level Behaviors," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*, 2016, pp. 225–234.
[38] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe LTL specifications," in *Proc. Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 1525–1532.
[39] R. G. Sanfelice and E. Frazzoli, "A hybrid control framework for robust maneuver-based motion planning," in *Proc. American Control Conference*, 2008, pp. 2254–2259.
[40] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *The Int. Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.
[41] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The Int. Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.
[42] B. Schürmann and M. Althoff, "Convex interpolation control with formal guarantees for disturbed and constrained nonlinear systems," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*, 2017, pp. 121–130.
[43] ——, "Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space," in *Proc. 20th IFAC World Congress*, 2017, pp. 12 020–12 027.
[44] ——, "Optimal control of sets of solutions to formally guarantee constraints of disturbed linear systems," in *Proc. American Control Conference*, 2017, pp. 2522–2529.
[45] D. Calzolari, A. M. Giordano, and A. Albu-Schaffer, "Error Bounds for PD-Controlled Mechanical Systems under Bounded Disturbances Using Interval Arithmetic," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1231–1238, 2020.
[46] M. Chen, S. L. Herbert, H. Hu, Y. Pu, J. F. Fisac, S. Bansal, S. Han, and C. J. Tomlin, "FaSTrack:A Modular Framework for Real-Time Motion Planning and Guaranteed Safe Tracking," *IEEE Trans. on Automatic Control*, vol. 66, no. 12, pp. 5861–5876, 2021.
[47] P. M. Van Den Hof and R. J. Schrama, "Identification and control - Closed-loop issues," *Automatica*, vol. 31, no. 12, pp. 1751–1770, 1995.
[48] S. G. Douma and P. M. Van Den Hof, "Relations between uncertainty structures in identification for robust control," *Automatica*, vol. 41, no. 3, pp. 439–457, 2005.
[49] L. Ljung, *System Identification. Theory for the User*, 2nd ed. New Jersey: Prentice Hall, 1999.
[50] J. Santolaria and M. Ginés, "Uncertainty estimation in robot kinematic calibration," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 2, pp. 370–384, 2013.

[51] Y. Chen, H. Peng, J. Grizzle, and N. Ozay, "Data-Driven Computation of Minimal Robust Control Invariant Set," in *Proc. IEEE Conf. on Decision and Control*, 2019, pp. 4052–4058.
[52] S. Sadraddini and C. Belta, "Formal Guarantees in Data-Driven Model Identification and Control Synthesis," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*, 2018, pp. 147–156.
[53] B. Schürmann, D. Heß, J. Eilbrecht, O. Stursberg, F. Köster, and M. Althoff, "Ensuring drivability of planned motions using formal methods," in *2017 IEEE Int. Conf. on Intelligent Transportation Systems*, 2017, pp. 1–8.
[54] M. Althoff and J. M. Dolan, "Reachability computation of low-order models for the safety verification of high-order road vehicle models," in *American Control Conference*, 2012, pp. 3559–3566.
[55] Z. Wang and R. M. Jungers, "Scenario-Based Set Invariance Verification for Black-Box Nonlinear Systems," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 193–198, 2021.
[56] S. B. Liu and M. Althoff, "Reachset Conformance of Forward Dynamic Models for the Formal Analysis of Robots," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2018, pp. 370–376.
[57] A. Giusti, S. B. Liu, and M. Althoff, "Interval-arithmetic-based robust control of fully actuated mechanical systems," *IEEE Trans. on Control Systems Technology*, vol. 30, no. 4, pp. 1525–1537, 2022.
[58] T. Dang, T. Dreossi, E. Fanchon, O. Maler, C. Piazza, and A. Rocca, "Set-Based Analysis for Biological Modeling," in *Automated Reasoning for Systems Biology and Medicine*. Springer Int. Publ., 2019, pp. 157–189.
[59] G. Batt, C. Belta, and R. Weiss, "Model Checking Genetic Regulatory Networks with Parameter Uncertainty," in *Hybrid Systems: Computation and Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4416 LNCS, pp. 61–75.
[60] R. E. Skelton, "Model error concepts in control design," *Int. Journal of Control*, vol. 49, no. 5, pp. 1725–1753, 1989.
[61] A. Vicino and G. Zappa, "Sequential approximation of feasible parameter sets for identification with set membership uncertainty," *IEEE Trans. on Automatic Control*, vol. 41, no. 6, pp. 774–785, 1996.
[62] M. Milanese and C. Novara, "Set Membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.
[63] M. Kieffer, E. Walter, and I. Simeonov, "Guaranteed nonlinear parameter estimation for continuous-time dynamical models," *Robust Control Design*, vol. 5, pp. 685–690, 2006.
[64] J. Bravo, T. Alamo, and E. Camacho, "Bounded error identification of systems with time-varying parameters," *IEEE Trans. on Automatic Control*, vol. 51, no. 7, pp. 1144–1150, 2006.
[65] N. Ramdani and P. Poignet, "Robust dynamic experimental identification of robots with set membership uncertainty," *IEEE/ASME Trans. on Mechatronics*, vol. 10, no. 2, pp. 253–256, 2005.
[66] M. Althoff, O. Stursberg, and M. Buss, "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.
[67] A. Girard, "Reachability of Uncertain Linear Systems Using Zonotopes," in *Proc. ACM Int. Conf. on Hybrid Systems: Computation and Control*. Springer, 2005, pp. 291–305.
[68] V. Gassmann and M. Althoff, "Scalable Zonotope-Ellipsoid Conversions using the Euclidean Zonotope Norm," in *Proc. American Control Conference*, 2020, pp. 4715–4721.
[69] S. B. Liu, A. Giusti, and M. Althoff, "Velocity estimation of robot manipulators: An experimental comparison," *IEEE Open Journal of Control Systems*, pp. 1–12, 2022.
[70] K. Makino and M. Berz, "Taylor models and other validated functional inclusion methods," *Int. Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 379–456, 2003.
[71] M. Althoff, D. Grebenyuk, and N. Kochdumper, "Implementation of Taylor models in CORA 2018," in *ARCH18. 5th Int. Workshop on Applied Verification of Continuous and Hybrid Systems*, ser. EPiC Series in Computing, vol. 54. EasyChair, 2018, pp. 145–173.
[72] S. B. Liu and M. Althoff, "Online Verification of Impact-Force-Limiting Control for Physical Human-Robot Interaction," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2021, pp. 777–783.
[73] M. Althoff, "An Introduction to CORA 2015 (Tool Presentation)," in *Proc. Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.
[74] F. Gruber and M. Althoff, "Computing Safe Sets of Linear Sampled-Data Systems," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 385–390, 2020.
[75] B. Schürmann and M. Althoff, "Optimizing Sets of Solutions for Controlling Constrained Nonlinear Systems," *IEEE Trans. on Automatic Control*, vol. 66, no. 3, pp. 981–994, 2021.
[76] J. Deshmukh, M. Horvat, X. Jin, R. Majumdar, and V. S. Prabhu, "Testing Cyber-Physical Systems through Bayesian Optimization," vol. 16, no. 5s, pp. 1–18, 2017.
[77] S. Nicosia, A. Tornambè, and P. Valigi, "State estimation in robotic manipulators: Some experimental results," *Journal of Intelligent & Robotic Systems*, vol. 7, no. 3, pp. 321–351, 1993.
[78] K. Busawon and H. K. Khalil, "Chapter 9: Digital Implementation," in *High-Gain Observers in Nonlinear Feedback Control*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2017, pp. 279–311.
[79] J. Doyle and G. Stein, "Multivariable feedback design: Concepts for a classical/modern synthesis," *IEEE Trans. on Automatic Control*, vol. 26, no. 1, pp. 4–16, 1981.
[80] W. H. Chen, J. Yang, L. Guo, and S. Li, "Disturbance-Observer-Based Control and Related Methods - An Overview," *IEEE Trans. on Industrial Electronics*, vol. 63, no. 2, pp. 1083–1095, 2016.
[81] P. J. Moylan, "Stable Inversion of Linear Systems," *IEEE Trans. on Automatic Control*, vol. 22, no. 1, pp. 74–78, 1977.

**Stefan B. Liu** received a B.S. degree in mechatronics, and an M.S. degree in robotics from the Technical University of Munich (TUM), Germany, in 2015 and 2017, respectively. He is currently pursuing a Ph.D. degree at the Cyber-Physical Systems Group of the TUM Department of Informatics. His research interest includes formal methods in robotics, physical human-robot interaction, modeling and identification, and modular robots.

**Bastian Schürmann** received a Bachelor of Science in Electrical and Computer Engineering from Technische Universität Kaiserslautern, Germany, in 2012; a Master of Science in Electrical Engineering from the University of California, Los Angeles, USA, in 2014; a Master of Science in Engineering Cybernetics from Universität Stuttgart, Germany, in 2015; and a Ph.D. in Informatics from Technische Universität München in 2022. In 2018, he was a visiting student researcher at the California Institute of Technology. His research focuses on combining control theory, reachability analysis, and optimization.

**Matthias Althoff** is an Associate Professor in computer science at the Technical University of Munich, Germany. He received his Diploma Engineering Degree in Mechanical Engineering in 2005 and his Ph.D. in Electrical Engineering in 2010, both from the Technical University of Munich, Germany. From 2010 to 2012, he was a postdoctoral researcher at Carnegie Mellon University, Pittsburgh, USA, and from 2012 to 2013, he was an assistant professor at the Ilmenau University of Technology, Germany. His research interests include formal verification of continuous and hybrid systems, reachability analysis, planning algorithms, nonlinear control, robotics, automated vehicles, and power systems.

# B Licenses

This chapter contains all explicit licenses for the publications reprinted in Appendix A, as required by the TUM Graduate School.

# License for Appendix A.1



RightsLink

Sign in/Register

IEEE Requesting permission to reuse content from an IEEE publication

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

**Conference Proceedings:**
2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

**Author:** Stefan B. Liu

**Publisher:** IEEE

**Date:** October 2018



### Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

# License for Appendix A.2



RightsLink

Sign in/Register

IEEE Requesting permission to reuse content from an IEEE publication

## Provably safe motion of mobile robots in human environments

**Conference Proceedings:**
2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

**Author:** Stefan B. Liu

**Publisher:** IEEE

**Date:** September 2017



### Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

# License for Appendix A.3



RightsLink

Sign in/Register

IEEE Requesting permission to reuse content from an IEEE publication

## Online Verification of Impact-Force-Limiting Control for Physical Human-Robot Interaction

**Conference Proceedings:**
2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

**Author:** Stefan B. Liu

**Publisher:** IEEE

**Date:** 27 September 2021



### Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

# License for Appendix A.4





## CC BY 4.0

# ATTRIBUTION 4.0 INTERNATIONAL

## Deed

**Canonical URL :** https://creativecommons.org/licenses/by/4.0/

**See the legal code**

### You are free to:

**Share** — copy and redistribute the material in any medium or format for any purpose, even commercially.

**Adapt** — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

### Under the following terms:

**Attribution** — You must give appropriate credit , provide a link to the license, and indicate if changes were made . You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

**No additional restrictions** — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

## Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation .

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

## Notice

This deed highlights only some of the key features and terms of the actual license. It is not a license and has no legal value. You should carefully review all of the terms and conditions of the actual license before using the licensed material.

Creative Commons is not a law firm and does not provide legal services. Distributing, displaying, or linking to this deed or the license that it summarizes does not create a lawyer-client or any other relationship.

Creative Commons is the nonprofit behind the open licenses and other legal tools that allow creators to share their work. Our legal tools are free to use.

- Learn more about our work
- **Learn more about CC Licensing**
- Support our work
- Use the license for your own material.
- Licenses List

- Public Domain List

## Footnotes

**appropriate credit** — If supplied, you must provide the name of the creator and attribution parties, a copyright notice, a license notice, a disclaimer notice, and a link to the material. CC licenses prior to Version 4.0 also require you to provide the title of the material if supplied, and may have other slight differences.

- More info

**indicate if changes were made** — In 4.0, you must indicate if you modified the material and retain an indication of previous modifications. In 3.0 and earlier license versions, the indication of changes is only required if you create a derivative.

- Marking guide
- More info

**technological measures** — The license prohibits application of effective technological measures, defined with reference to Article 11 of the WIPO Copyright Treaty.

- More info

**exception or limitation** — The rights of users under exceptions and limitations, such as fair use and fair dealing, are not affected by the CC licenses.

- More info

**publicity, privacy, or moral rights** — You may need to get additional permissions before using the material as you intend.

- More info

# License for Appendix A.5



RightsLink

Sign in/Register

IEEE Requesting permission to reuse content from an IEEE publication

## Guarantees for Real Robotic Systems: Unifying Formal Controller Synthesis and Reachset-Conformant Identification

**Author:** Stefan B. Liu
**Publication:** IEEE Transactions on Robotics
**Publisher:** IEEE
**Date:** October 2023



### Thesis / Dissertation Reuse

**The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:**

*Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:*

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

*Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:*

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

# C Theses of Supervised Students

Lastly, we acknowledge the students [81–90], who have completed their Bachelor's Thesis or Master's Thesis at the Technical University of Munich under the supervision of the author of this dissertation and have thereby contributed to this research:

[81] M. Riedel, "Smart Modules for Modular and Reconfigurable Robots," Master's Thesis, Technical University of Munich, 2019.

[82] G. Michels, "Conformance Testing of a Robot Manipulator with Safety Guarantees," Master's Thesis, Technical University of Munich, 2019.

[83] B. Gaida, "A Robotics Software Framework for R&D," Bachelor's Thesis, Technical University of Munich, 2019.

[84] P. Maroldt, "Verified Safe Human-Robot Interaction with Impedance Control Based on Force Limits," Master's Thesis, Technical University of Munich, 2019.

[85] S. Schepp, "Visualization of Reachable Spaces for Interactions with Modular Robots," Bachelor's Thesis, Technical University of Munich, 2019.

[86] D. Beckert, "Verified safe collision system for human-robot collaboration," Master's Thesis, Technical University of Munich, 2019.

[87] F. Guan, "Robust people detection and online verification of mobile robots," Master's Thesis, Technical University of Munich, 2020.

[88] M. Perschl, "Reachability Analysis and Conformance Checking for Robot Manipulators using Parallel Hybrid Automatons," Bachelor's Thesis, Technical University of Munich, 2021.

[89] C. Pan, "Identification and Learning-Based Test Case Generation for the Safe Model of Autonomous Vehicles," Master's Thesis, Technical University of Munich, 2021.

[90] P. Schmutz, "Online Motion Verification of Industrial Mobile Robots Using Reachable Sets," Master's Thesis, Technical University of Munich, 2021.