TU WIEN Informatics

Independent Wheel Offset Steering

Analysis of a Wheeled Robot Locomotion System and an Odometry based Motion Model

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Eugen Kaltenegger, BSc.
Matrikelnummer 01125205

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Ass. Dipl.-Ing. Dr.techn. Markus Bader

Wien, 9. August 2023

Eugen Kaltenegger | Markus Bader

TU WIEN Informatics

Independent Wheel Offset Steering

Analysis of a Wheeled Robot Locomotion System and an Odometry based Motion Model

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Eugen Kaltenegger, BSc.
Registration Number 01125205

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Ass. Dipl.-Ing. Dr.techn. Markus Bader

Vienna, 9th August, 2023

Eugen Kaltenegger

Markus Bader

Erklärung zur Verfassung der Arbeit

Eugen Kaltenegger, BSc.

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 9. August 2023

Eugen Kaltenegger

Danksagung

Ich möchte diese Abschlussarbeit meiner Frau, Sarah, ihrer unendlichen Geduld und unserer bedingungslosen Liebe widmen.

Acknowledgements

I want to dedicate this thesis to my wife, Sarah, her infinite patience and our unconditional love.

Kurzfassung

Mobile Roboter, insbesondere solche mit einem Differential Drive (DD), werden vielseitig in Lagerhallen und Produktionsstätten eingesetzt. Sie kommen auch für den Transport in Umgebungen, in denen sich Menschen bewegen zur Anwendung.

In dieser Arbeit wird das neuartige Roboterantriebssystem Independent Wheel Offset Steering (IWOS) erstmals umfangreich vorgestellt. Dieses Bewegungssystem besteht aus einer linken wie auch einer rechten angetriebenen Lenkrolle, bei der sowohl die Gabel als auch das Rad gesteuert werden. Eine zusätzliche nicht gesteuerte Lenkrolle (Bockrad) sorgt für Stabilität.

Das innovative IWOS-System, inspiriert von der DD-Technologie, bietet erweiterten Bewegungsumfang, der eine Verbesserung der Human-Robot Interaction (HRI) sowie der Kollisionserkennung und -bewältigung ermöglicht. Der Bewegungsumfang der Gabel ist zwar begrenzt, bietet aber dennoch Vorteile für dieses Roboter-Antriebssystem.
Bei einer geringen Komplexität von Hardware und Software kann das IWOS-System eine andere Blickrichtung als Fahrtrichtung einnehmen. Mit dieser Fähigkeit kann das Blickverhalten von Menschen nachgeahmt werden, wodurch der Pfad eines mobilen Roboters mit seiner Blickrichtung angedeutet werden kann.

Diese Arbeit konzentriert sich auf die Erforschung der Grundlagen des IWOS-Systems um damit den Weg für weitere Studien zu ebnen. Im Zuge dieser Erörterung werden die Vorteile in Bezug auf die Lenkfähigkeit des IWOS gegenüber dem DD beleuchtet. Darüber hinaus wird ein Ansatz für einen Odometer sowie ein an dieses System angepasstes Odometry Motion Model (OMM) präsentiert. Dieses OMM zielt darauf ab, unvermeidliches Abdriften sowie Fehler der Odometrie mithilfe stochastischer Methodik auszugleichen.

Um die Machbarkeit und Möglichkeiten des IWOS zu demonstrieren, wurde als Teil gegenständlicher Arbeit ein Prototyp dieses Systems, der erste seiner Art, gebaut. Eine qualitative Analyse der mit diesem Prototyp durchgeführten Experimente liefert vielversprechende Ergebnisse für das IWOS-System sowie das OMM, das für dieses Antriebssystem entwickelte wurde, wobei das OMM Fehler des Odometers ausgleichen kann.

Abstract

Mobile robots, especially those with Differential Drive (DD), are widely used in warehousing and production, as well as in transportation in environments shared with humans.

In this thesis, a novel robot locomotion system, the Independent Wheel Offset Steering (IWOS), debuts. This locomotion system consists of an actuated left and right caster-like wheel with both the fork and the wheel steered independently. An additional freely-moving caster wheel ensures stability.

The innovative IWOS, inspired by the DD, offers greater maneuverability, resulting in enhancements to Human-Robot Interaction (HRI), collision detection and collision handling. The fork's range of motion is limited to a certain degree, but still provides benefits to this robot locomotion system in terms of maneuverability.
While keeping complexity of hardware and software to a minimum, the IWOS allows for facing a direction different to the driving direction. With this capability, the gaze behavior of humans can be imitated, thereby indicating the trajectory of a mobile robot with its gaze direction.

This work focuses on exploring the fundamentals of the IWOS, paving the way for further studies. In the course of this discussion, the advantages of the IWOS in terms of maneuverability over the DD are examined. Additionally, an approach to an odometer and an Odometry Motion Model (OMM) tailored to this locomotion system are introduced. This OMM aims to compensate inevitable drift and error in odometer data with stochastic methodology.

In order to provide a proof of concept, a prototype of the IWOS, the first of its kind, was built in the course of this thesis. A qualitative analysis of experiments conducted with this prototype provides promising results for the IWOS and the OMM designed for this locomotion system, leading to an enhancement in odometer data by the OMM.

Contents

CHAPTER 1

Introduction

The fields of automation and robotics are experiencing rapid advancements, leading to the integration of autonomous robots into various aspects of our daily lives. As robots and humans increasingly share spaces, it becomes essential to ensure seamless interaction between them. Specifically, when it comes to mobile robots navigating through shared environments, it is beneficial to avoid interruptions and crucial to avoid collisions with humans. In order to achieve this, it is advantageous for humans to intuitively understand the path of the robot [HS13].

Addressing the challenge of indicating a robot's path, Markus Bader and George Todoran Horatio invented and patented the novel robot locomotion system IWOS at the Vienna University of Technology [THM18]. For this thesis, a functional prototype of the IWOS locomotion system has been developed and constructed. The prototype, which serves as a tangible representation of the IWOS concept, is depicted in Figure 1.1.

The IWOS draws influence from the well-established and widely-used DD, which has two separately-steerable wheels. This wheel configuration allows the DD to drive straight lines, execute smooth curves and even achieve precise spot rotations. The DD finds application in various domains such as warehousing, production and transportation in crowded environments.

The IWOS aims to enhance the capabilities of the DD. For this reason, the IWOS extends the DD by incorporating two additional joints that connect the robot's wheels to its base. These joints resemble hinge joints with vertical axes, and an offset between these vertical axes and the wheel's axes gives the IWOS its distinctive name. This configuration is similar to caster wheels, but with the wheels and the forks actuated, enabling the wheels to rotate in an arc-like motion, significantly enhancing the robot's maneuverability. The motion of these additional joints can be limited, but still provide the full benefit of the robot locomotion system. On the contrary, such limitations contribute to the overall stability and reliability of the system.

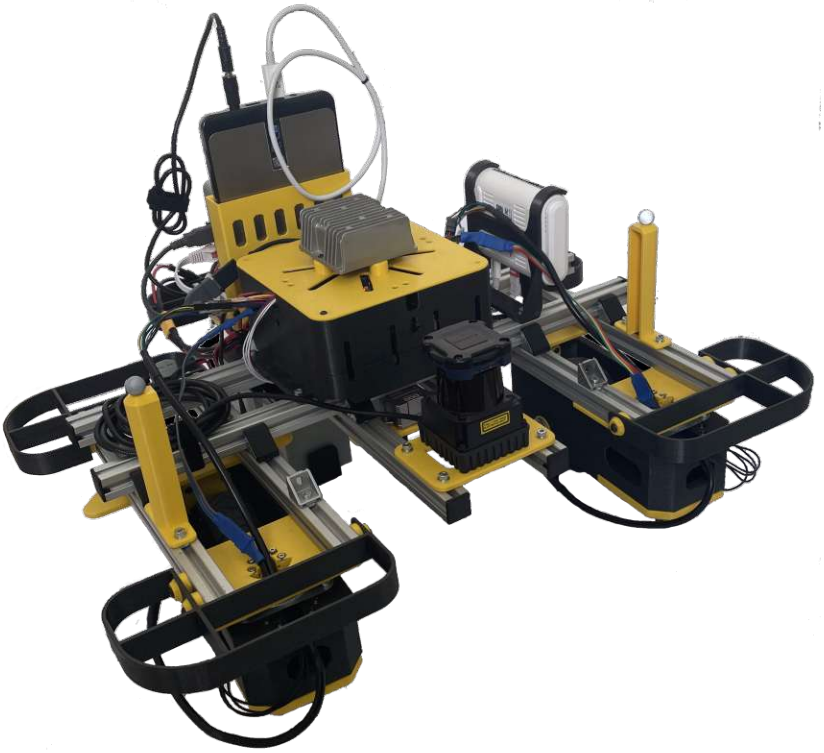


Figure 1.1: Picture of the prototype robot featuring IWOS.

The IWOS offers two major advantages over the DD.
Firstly, it allows the robot to have a facing direction contrary to its driving direction [CKTB19]. This capability is made possible by the additional joints of the IWOS and can not be achieved with the DD. It enables robots utilizing the IWOS to adopt a human-like approach of looking where it is heading, as observed in human gaze behavior [HPV02]. This feature enhances the potential for humans to intuitively predict the robot's path and therefore improves HRI.
Secondly, the IWOS may enable better collision detection and resolution as compared to conventional DD robots, as the additional joints can be utilized to detect collisions. Based on the forces measured on the joints, the collision location and intensity can be inferred. The only exception to this are direct frontal or rear collisions, which would not result in measurable forces applied to the joints. Furthermore, when detecting such a collision, the movement of the joints can be utilized to initiate a pivoting motion of the robot, resolving the collision and steering the robot away from the point of impact.

The notable benefits offered by the IWOS do come at the cost of heightened complexity in both hardware and software as compared to the DD system. This increase in complexity raises an important question: "Is this added intricacy a favorable trade-off, considering the advantages the IWOS has to offer?"

1.1 Problem Statement

Numerous approaches have been explored to enable robots equipped with the DD locomotion system to effectively convey their navigation intentions. However, these approaches often fall short in terms of intuitive human interaction or require bulky equipment attached to the robot. In contrast, the innovative IWOS introduces a novel locomotion system that not only enhances the capabilities of the DD, but also strives to maintain a lower level of complexity compared to omnidirectional drives. Omnidirectional drives enable robots to drive in any direction relative to their facing direction, granting them exceptional maneuverability. However, these systems often necessitate complex hardware and software solutions, which can increase the overall complexity of the robot.

The IWOS seeks to strike a balance by offering improved capabilities while keeping complexity at a manageable level. Despite the potential of the IWOS, the current lack of research on this specific locomotion system is due to its novelty and patent protection. As a result, there is a notable gap in studies comparing the DD and IWOS, which could provide valuable insights into the adaptability of technologies developed for the former to suit the latter. Investigating the similarities and differences between the DD and IWOS would greatly enhance our understanding of the IWOS and facilitate the identification of necessary technological adaptations. For this reason a dedicated research of the fundamentals of the novel IWOS locomotion system is required, providing a deeper understanding of the underlying principles and mechanisms that drive the functionality and performance of the IWOS system.

Furthermore, a functional prototype validating the concept of the IWOS and its purported benefits has yet to be developed. Creating such a prototype is a crucial initial step in acquiring empirical knowledge about the IWOS, its possibilities, and its limitations. However, fundamental challenges in the field of robotics need to be addressed for the novel IWOS, including but not limited to position tracking, control, and navigation, specific to this innovative type of robot locomotion system.

1.2 Research Questions

The objective of this thesis is to make theoretical and practical advancements towards understanding and furthering the capabilities of the IWOS. Based on the potential of the IWOS the following research questions shall be answered throughout this thesis:

Research Question 1: What are the key similarities and differences in the driving maneuvers the DD locomotion system and the IWOS locomotion system are capable of?

Research Question 2: How can accurate tracking of the position and orientation (pose) of an IWOS robot platform be achieved using sensor measurements from the controlled joints?

Research Question 3: How can Sebastian Thrun's OMM developed for the DD system [TBF05] be adapted to effectively capture the unique motion capabilities of the IWOS?

By delving into these aspects and leveraging the unique capabilities of the IWOS, our goal is to advance the field of robotics. We aim to pave the way for robotic systems that are capable of intuitively communicating their navigation intentions, thus enhancing safety for both humans and robots alike.

1.3 Structure of this Thesis

The primary objective of this thesis is to comprehensively explore and evaluate the potential of the IWOS system, both in theory and practical applications.

The thesis commences by delving into various approaches towards HRI, with focus on state-of-the-art approaches on path indication with various locomotion systems. It is then followed by an analysis of collision detection and resolution techniques. Furthermore, Chapter 2 provides an overview on contemporary approaches currently employed in the development of wheel mobile robots.

To address Research Question 1, Chapter 3 thoroughly examines the fundamental aspects of the IWOS. This includes a detailed analysis of both the DD system and its advancement, the IWOS, with a particular emphasis on their distinctive characteristics and maneuverability.

Chapter 4 presents an extensive discussion on the design and development of the aforementioned IWOS prototype.

Research Question 2 is addressed in Chapter 5 where an algorithm for joint state based odometry is introduced. This algorithm is the foundation for odometry calculation for the IWOS prototype.

In Chapter 6 the focus is on Reseach Question 3. The odometry motion model designed by Sebastian Thrun is adapted to suit the specific capabilities of the IWOS. Experimental validation of this novel motion model is conducted, and the structure of these experiments is elaborated upon towards the end of Chapter 6.

Chapter 7 critically assesses the outcomes of the experiments conducted, leading to the results of this thesis. Finally, Chapter 8 provides a comprehensive outlook for future research and concludes the thesis.

CHAPTER 2

Related work

This chapter investigates research in the field of robotics, focusing on the advancements related to IWOS. The IWOS introduces two significant advantages compared to other robotics systems.
Firstly, IWOS exhibits the potential to indicate the robots path, thus enhancing HRI. By enabling intuitive path indication, IWOS promotes seamless communication and collaboration between humans and robots.
Secondly, IWOS offers sophisticated collision detection and recovery techniques. This capability empowers the system to potentially detect and locate collisions and employ advanced techniques to recover from such situations. This could enhance the overall safety and reliability of the robotic system.

Consequently, this chapter delves into a discussion of various approaches towards path indication in HRI, aiming to provide a comprehensive overview of the current state-of-the-art. Additionally, it provides an overview of the advancements in collision detection and recovery. Furthermore, this chapter also presents an overview of the state-of-the-art in wheeled mobile robot development with focus on locomotion and odometer approaches.

2.1 Human Robot Interaction

The advancements in the field of robotics have led to the integration of autonomous mobile robots not only in warehouses and production sites but also in public spaces. These robots, operating autonomously in both indoor and outdoor environments, are becoming increasingly prevalent in our everyday lives. However, to ensure a seamless and safe interaction between humans and autonomous mobile robots in shared spaces, it is crucial that both humans and robots can predict each other's intentions [HS13]. This gives rise to twofold challenges: firstly, the robots must grasp human intentions, which is particularly complex due to humans often communicating with context [MSKH15], and secondly, humans need to comprehend the intentions of the robots.

Within the scope of this thesis, we are particularly interested in investigating the possibility of humans predicting the path of autonomous mobile robots.

In commercial and industrial settings, individuals who interact with robots are typically trained to effectively engage with these automated systems. However, in shared spaces where humans without prior training may encounter robots, it becomes even more crucial for the robot to communicate its intentions in an intuitive manner.
Researchers often draw inspiration from everyday life in order to solve problems, and in the case of HRI, human-human interaction serves as a valuable source of inspiration. Humans naturally tend to look in the direction they are going, thereby indicating their intended path through their gazing behavior [HPV02]. Therefore, the facing direction of a robot could serve as a useful tool to indicate the path it is traversing. However, apart from gazing behavior, factors such as the robot's speed when passing by a human and the distance between the human and the robot seem to have an impact on the subconscious and inherently social HRI [LS13]. Numerous approaches have been explored to indicate the path of robots, often relying on components attached to the robot to convey the direction of movement.

One relatively simple approach involves the use of flashing lights similar to turn indicators used in automobiles. However, studies have suggested that while this approach may seem familiar from driving experiences, it is not as effective in communicating the robot's intention [FJK+18]. Other studies have shown that intention indication through rotation outperforms the flashing lights approach [PRJ+20].

A promising approach to indicate the path of DD robots is to employ light projections on the floor. One method utilizes a laser pointer and a movable reflective surface to draw the robot's path on the ground. This approach allows to communicate both the robots path and velocity for a time window of three seconds. Regrettably, this approach has not undergone further experimentation to ascertain its impact on HRI [MKI06].

Alternatively, projectors, akin to the ones commonly utilized for home theaters, can be mounted on the robot to vividly display its path directly on the floor. These projections have been shown to improve HRI by making it easier to trace the robot's path [HCVdL21, CAKL15, WIM+15]. While these approaches are indeed functional, they frequently entail the incorporation of cumbersome and energy-intensive hardware that needs to be mounted on top of the robot. Even if solutions were found to address these challenges, the fundamental issue of effectively utilizing light for path indication in bright environments remains unresolved. Furthermore, these approaches suffer from a deficiency in establishing a natural and intuitive connection between humans and robots.

In an experimental setting, researchers evaluated a rotation indication motion, where a DD robot would stop and rotate in the direction it intends to move soon. This method of motion indication has shown superiority over light-based indication systems [PRJ+20]. Another study explored a more natural approach by attaching an object representing a head to the robot to indicate its intended movement. In this study, the head would gaze in the robot's driving direction, and the results were compared to indicator lights.

The findings suggest that indicator lights may be more beneficial than gazing behavior, although the results leave room for interpretation.
While the results of the conducted experiments were not statistically significant, likely due to a small number of participants involved [MDH15], it is important to consider that robot faces and heads are also associated with attention signaling, as highlighted in previous research [BNS02]. Therefore, conflicting goals can arise for the robot's gazing behavior. On the one hand, it needs to indicate that it has detected the presence of a human, while on the other hand, it also needs to gaze towards its intended path.

2.2 Collision Detection and Recovery

Collisions are regarded as the worst-case scenario in the field of robotics, posing significant challenges and risks. The most promising approach in addressing collisions is to prevent them from occurring in the first place. Collision avoidance, encompassing the avoidance of both static and dynamic obstacles, is a critical area of research for single robot applications as well as multi-robot scenarios [PPP17]. Collision avoidance includes hardware components detecting obstacles and software components dealing with detected obstacles. Research has demonstrated the efficiency of laser range finders in enabling collision avoidance, even in environments with multiple robots operating in unknown surroundings [HMS12]. The selection of an appropriate software approach on collision avoidance depends on the specific use case and the anticipated environment in which the robot will operate [WLZ+21].

In unfortunate instances where collision avoidance fails and a robot collides with an object or obstacle, the subsequent task involves handling the collision. Collision handling can be divided into distinct phases: precollision, detection, isolation, identification, classification, reaction, and postcollision. The precollision phase primarily focuses on collision avoidance and is not considered a direct part of the collision event. The detection phase refers to the moment when the robot detects the collision, while the isolation phase involves evaluating the precise point of collision. Subsequently, based on the gathered parameters pertaining to the collision, the collision is identified and classified. The subsequent step is to react to the collision, typically at the lowest level of control for the robot. The specific reaction depends on the classification of the collision, but a simple approach might involve stopping the robot. Finally, in the postcollision phase, the robot determines how to proceed after the collision [HDLAS17].

Research has been conducted on collision detection for omnidirectional robots, which possess the ability to navigate in any direction at any given time. For such robots, research has demonstrated the potential of leveraging torque sensing to accurately estimate collision parameters, including the location, direction, and magnitude of collision forces. By harnessing torque sensing technology, these robots can achieve effective collision detection and isolation, thereby enhancing their overall collision reaction capabilities [KLS16].

Preliminary simulation experiments with the IWOS have indicated that forces are exerted on the joints, which can be measured [CKTB19]. However, due to patent protection, no research has been conducted thus far specifically addressing collision detection and collision handling in relation to IWOS. A prototype of the IWOS has been developed for this thesis, offering an opportunity to further explore the collision detection and collision handling capabilities of IWOS.

2.3 Mobile Robot Design and Development

The presence of mobile robots in our everyday lives continues to expand, encompassing various domains. Among the popular choices are DD robots, known for their simplicity and effective locomotion approach, making them versatile for a wide range of applications. With proper design, these robots can handle heavy loads, while maintaining a cost-effective solution, making them particularly suitable for industrial settings such as warehouses and production sites, which has sparked significant research interest in this field [NFBR23]. Interestingly, the utility of DD robots has even transcended into the realm of art, where their capabilities have been harnessed creatively [SL17].
Another popular type of robots are omnidirectional robots, which offer enhanced maneuverability, enabling movements in directions independent of the robot's current heading [TB16]. However, this increased versatility comes at the cost of higher complexity, leading to a greater predisposition for errors.

The IWOS, a patented wheel configuration developed by the Vienna University of Technology, seeks to combine the simplicity of the DD with the versatility and aforementioned benefits of omnidirectional drives in terms of HRI, collision detection, and recovery [THM18]. Designing and constructing a robot with this novel locomotion system is a highly intricate task that necessitates expertise in mechanical engineering, electronic engineering, and software engineering. The development process encompasses tackling fundamental challenges in locomotion, perception, cognition, and navigation within the field of robotics [RVLA19]. To facilitate these endeavors, the open-source Robot Operating System (ROS) has emerged as the state-of-the-art framework for robotic applications, with its successor Robot Operating System 2 (ROS2) aimed at overcoming certain limitations of its predecessor [QCG$^+$09, MKA16].

Motion models play a vital role in various robotics applications, particularly in the domains of locomotion and navigation, including localization, motion planning, and motion control [RVLA19]. For DD robots, well-established motion models enable accurate estimation of the robot's pose based on wheel velocities [KZBS17]. This technique has gained widespread adoption owing to its reliability, even finding use in the Mars rover missions, where wheel encoder odometry besides Inertial Measurement Unit (IMU) and visual odometry are employed [CMM05].

To address the inherent drift in such systems, alternative sources for position tracking are actively sought. A promising technology seems to be visual odometry, relying on input stream from cameras to determine the change in position of a system. Such approaches offer the added advantage that these technologies not only provide information about the change in position but can also be utilized for simultaneous localization and mapping, further enhancing the robot's perception capabilitie [YBHH15]. Most recent approaches on this topic utilize machine learning approaches to further enhance the accuracy of such systems [VCBV23]. Various sources of odometry can be combined to enhance the overall accuracy [MS14].

Nonetheless, factors such as slippage and sensor errors can introduce divergence between the computed pose and the true pose of the robot over time, especially for systems relying on less sensors, such as joint encoder odometry [MHW+19].

To address this challenge from a statistical standpoint, Sebastian Thrun developed a velocity and odometry motion model that incorporates error and uncertainty considerations [TBF05]. This approach enhances the estimation of the robot's motion, thereby improving overall performance and accuracy. By integrating advanced motion models into the realm of robotics, researchers strive to augment the capabilities and reliability of autonomous systems, paving the way for further advancements in this dynamic field.

This section has shed light on various aspects important to the IWOS including HRI, collision detection and collision handling as well as design and development principles for mobile robots. In order to deduce from these findings and apply them to the IWOS, the subsequent section discusses the fundamental principles of the IWOS.

This section has provided valuable insights into related work for aspects crucial to the IWOS, including HRI, collision detection and handling, as well as design and development principles of mobile robots. Building upon these insights, the subsequent section discusses the fundamental principles of the IWOS. Exploring these fundamental principles, allows to draw meaningful conclusions from the previous findings and effectively apply them to the IWOS.

CHAPTER 3

Fundamentals of the Independent Wheel Offset Steering

The DD is a well-established locomotion system for mobile robots, finding applications in various fields such as industrial production, warehousing, indoor and outdoor delivery, and more. The success of the DD in robotics is attributed to its robustness and simplicity as a locomotion approach for mobile robot platforms. However, the DD has various limitations which include human-robot interaction and collision handling. These limitations led to the development of the IWOS, which is a derivative of the DD.

The innovative IWOS robot locomotion system expands the kinematic capabilities of the DD while aiming to maintain its simplicity. The goal of IWOS is to provide two significant improvements over the DD.

- Firstly, the IWOS allows the robot to look in a different direction than its actual movement direction, which might be beneficial for human-robot interaction, since this feature could be used to indicate the robot's path. The DD is not capable of driving in a direction different from its facing direction. While omnidirectional robot systems offer this capability, they are associated with high complexity in terms of both hardware and software, making them more prone to errors as compared to DD and IWOS robots.

- Secondly, in the event of a collision, the IWOS enables the robot to potentially resolve it without moving backwards. The feasibility of resolving a collision without the need of moving backwards depends on the specific circumstances of the collision and the type of robot involved into the collision. Nonetheless, IWOS provides an advantage over the DD when it comes to detecting and resolving possible collisions.

In the following section, a detailed discussion of IWOS will be presented. The characteristics and capabilities of the DD and IWOS will be explained and compared, laying a solid foundation for understanding IWOS.

3.1 Characteristics

In this section the basic characteristics of both the DD and the IWOS will be discussed. Since the IWOS can be seen as an extension of the DD, we will first take the DD into consideration and then discuss the characteristics of the IWOS based on that information.

3.1.1 Characteristics of the Differential Drive

The DD is a locomotion system commonly used in wheel-based mobile robots and tracked vehicles such as excavators or tanks. It is comprised of a separately-driven left and right wheel. These wheels are positioned in such a way that their axes are aligned. The distance between the wheels is denoted as wheelbase l.
In order to ensure the stability of the robot platform, a third wheel is typically incorporated [SNS11]. The third wheel has to be located such that the center of mass lies within the triangle formed by the three wheels. This third wheel usually is a spherical wheel or a caster wheel. In most applications this third wheel spins freely and is not actuated. By definition, the center of the robot's coordinate system is located in between the wheels steered. Figure 3.1 depicts the concept of the DD locomotion system.

3.1.2 Characteristics of the Independent Wheel Offset Steering

IWOS was inspired by DD and aims to address some of its limitations. To these ends, the IWOS extends the DD with an additional hinge joint for each driven wheel. These additional joints are independently steerable. Unlike steered standard wheels, the vertical axis of rotation of these joints does not intersect the contact point of the wheels with the surface below [SNS11], see Figure 3.2.
The distance between the intersection of the hinge joint axis and the contact point of the wheel with the surface below is denoted as wheel offset s. It is not necessary, but is beneficial if the wheel offset is identical for the left and right wheels [THM18]. In the following we will assume that the wheel-offset is identical for both the left and right sides of the robot.
The hinge joints allow for an arc like lateral motion of the wheels around the joints. Therefore, IWOS has a wider range of possible motions than the DD. In order to take advantage of the hinge joints it is not necessary that the joints are able to rotate in a full circular motion. Actually, it might be beneficial if the motion is limited, as it reduces the complexity of the system and therefore reduces the probability of hardware failure. The distance between the left and the right hinge joints is denoted as wheelbase l.

In order to prevent conflicting positions of the left and right hinge joint which might lead to collisions of the wheels it is beneficial to ensure $2 \cdot (s + r) < l$ where r denotes the radius (semidiameter) of the wheels.

Like the DD the IWOS requires a third wheel for stability. Again, this third wheel can either be a caster wheel or a spherical wheel and does not need to be actuated. The robot's center of mass has to be located within the triangle formed by the three wheels of the robot while in any given position of the robot's wheels.

The independently-steerable hinge joints and aforementioned wheel offset are the namesakes of the IWOS. These enhancements allow for a facing direction that does not correspond to the driving direction. Additionally, these enhancements enable the robot to pivot, which could resolve a collision without the need of reversing. The motions, specific to the IWOS, are discussed in Section 3.4.

From here on, the additional joints will be denoted as the steering joints. The axes of these joints will be denoted as the steering axes. The orientation of the steering joints will be denoted as steering angles α_l and α_r. We define the zero position, also neutral position, of the steering angles as α_0 such that the steering axes are identical and the wheels are behind the steering joints, as depicted in Figure 3.2. Additionally, we want to

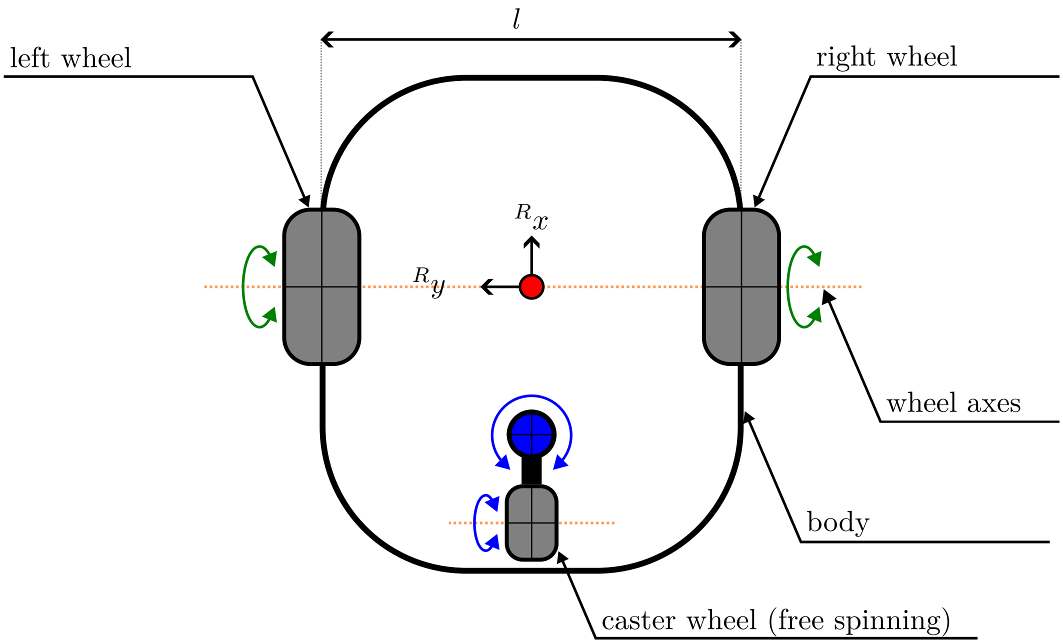


Figure 3.1: Structure of the DD. The large grey boxes depict the wheels driven. The small grey box depicts the caster wheel and the blue circle above represents the vertical axis of a caster wheel. Green arrows indicate the actuated motion and blue arrows indicate unactuated motion of the joints. The wheel axes are indicated with dotted orange lines. The red circle marks the center of the robot's coordinate system.

introduce the term revolute joints for the rotational wheel joints. The axes of the wheel joints will be denoted as the revolute axes.

The IWOS is introduced with a crucial component that connects the steering joint and the revolute joint, referred to as fork in subsequent discussions.
It is important to highlight that alternative methods can be employed for the construction of an IWOS. For instance, one possible approach involves utilizing a curved linear rail that enables rotation around a fictitious hinge joint. This alternative method opens up possibilities for diverse configurations and designs, enhancing flexibility in implementing IWOS systems to meet specific requirements and objectives.
Figure 3.2 depicts the concept of the IWOS locomotion system.

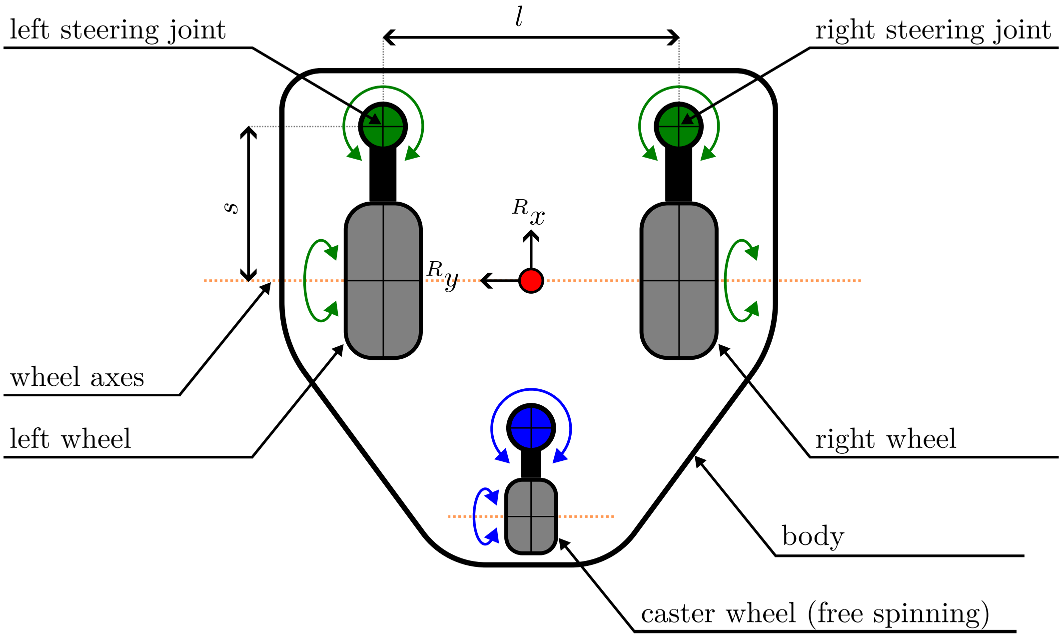


Figure 3.2: Structure of the IWOS. The large grey boxes depict the wheels driven and the green circle above depict the hinge joints actuated. The small grey box depicts the caster wheel and the blue circle above represents the swivel joint of the caster wheel. Green arrows indicate the actuated motion and blue arrows indicate unactuated motion of the joints. The wheel axes are indicated with dotted orange lines. The red circle marks the center of the robot's coordinate system.

3.2 Coordinate System

For a detailed discussion of the properties of the DD and the IWOS, suitable coordinate systems are necessary. More specifically, we need to define a coordinate system embedding the robot and a coordinate system on the robot itself. The former is commonly denoted as the world coordinate system and the latter one as the robot coordinate system. For the use cases in this thesis, a planar Cartesian coordinate system is the suitable option for both the world coordinate system and the robot coordinate system.

The pose of a robot is defined by its position and its orientation in the external coordinate system, denoted as world coordinate system. The robot's position in the world coordinate system is defined by its x-coordinate and y-coordinate and its orientation is defined by θ [TBF05, SNS11]. Thus, the robot's pose $\mathbf{x}$ has three parameters and is defined below.

$$\mathbf{x} = \begin{bmatrix} x, & y, & \theta \end{bmatrix}^T \tag{3.1}$$

The pose of a robot in the world coordinate system is depicted in Figure 3.3.

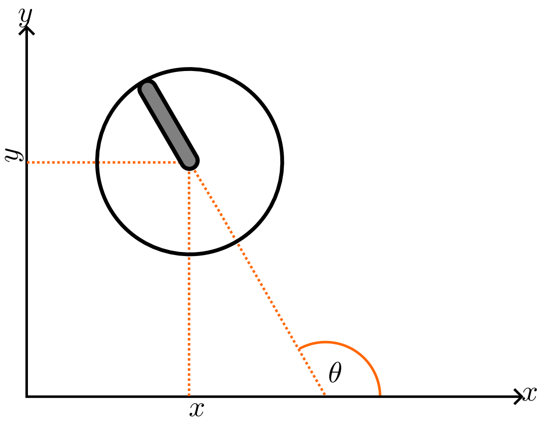


Figure 3.3: Pose of a robot in a planar space, defined by the vector $\mathbf{x} = \begin{bmatrix} x, & y, & \theta \end{bmatrix}^T$.

For robots which can face a different direction from their driving direction, we determine that the orientation θ refers to the direction the robot faces. How the offset between facing and driving direction is determined for a robot with IWOS will be discussed in Section 3.3.

For the DD and IWOS, the center of the robot's coordinate system is located between the wheels, where we assume that the steering angle of both the left and right steering joints are in their neutral position. The positive x-axis of the robot's coordinate system represents the forward direction and the negative x-axis represents backwards. We define the direction the robot faces as equivalent to the positive x-axis. The positive y-axis is to the left of the robot, and the negative y-axis, to the right of the robot. The positive z-axis points upwards from the robot and the negative z-axis points downwards. Angles are defined as positive if they are measured counter-clockwise and negative if they are measured clockwise. These definitions are in consensus with the ROS Enhancement Proposals (REP) 103[1]. Details regarding ROS will be discussed in Section 4.3.1.

[1] In REP 103 a reference for units and coordinate conventions is provided, for more details visit: `https://github.com/ros-infrastructure/rep/blob/master/rep-0103.rst` (accessed on 09.08.2023).

The Figures 3.1 and 3.2 depict a DD and IWOS robot respectively, in both graphics the robot's coordinate systems are depicted with a red dot.

3.3 Instantaneous Center of Curvature

In kinematics and dynamics, the Instantaneous Center of Curvature (ICC), also Instantaneous Center of Rotation (ICR), is a tool for describing planar motion. At any particular instant, planar motion of a body can be described as rotation around a fictional point, the ICC. Consequently, knowledge of the rotational direction and rotational velocity as well as the ICC is required in order to fully describe a planar motion. In the following section calculation of the ICC and resulting consequences for the DD and the IWOS will be discussed. Details on the rotational direction and velocity are discussed in Section 3.2.

3.3.1 Calculation of the Instantaneous Center of Curvature

The ICC is calculated by creating a line orthogonal to the motion vector of a point of the body in motion. For curved motions, the lines orthogonal to the motion vector of any two points intersect at the ICC. For linear motions, the lines are parallel and therefore do not intersect. In this case, the ICC is located at infinity and the rotation radius is also infinity.

The distance between the center of the body and the ICC defines rotational radius r. This point rotates around the ICC with an angular velocity ω and has a translation velocity v. The general relationship between angular velocity and translation velocity is defined as [TBF05]

$$v = \omega \cdot r. \tag{3.2}$$

The translation velocity is defined such that forward motion has a positive velocity and backwards motion has a negative velocity. According to the conventions, on coordinate systems angular velocity is defined as positive for counter-clockwise rotations and negative for clockwise rotations. According to these definitions radius is defined as positive if the ICC is located on the left side of the moving object and negative if the ICC is located on the right side of the moving object. Here we refer to the left hand right side relative to the robot's driving direction.

3.3.2 Differential Drive

The wheel axes of the DD are always identical. Thus, these axes have an infinite number of intersection points. Therefore, the ICC can be located anywhere on these axes and the radius of rotation can have any value from negative infinity to positive infinity. In the robot's coordinate system, the x-coordinate of the ICC is always zero and the y-coordinate is equivalent to the radius. The exact location of the ICC is linked to the ratio of the velocities of the left and right wheels.

The general relationship between angular velocity ω and translation velocity for the left wheel v_l and the right wheel v_r is described as

$$v_l = \omega \cdot \left(r - \frac{l}{2}\right) \text{ and } v_r = \omega \cdot \left(r + \frac{l}{2}\right). \tag{3.3}$$

From these relations, the radius of motion can be determined by

$$r = \frac{l}{2} \cdot \frac{(v_l + v_r)}{(-v_l + v_r)} \tag{3.4}$$

and the angular velocity can be written as

$$\omega = \frac{(-v_l + v_r)}{l}. \tag{3.5}$$

The third wheel, which is not driven, is pushed by frictional forces into a position such that the wheel axis intersects with the ICC.

A proper defined motion requires two of the three terms in Equation 3.2 to be defined. Equations 3.4 and 3.5 show that this is achievable for the DD by controlling the velocity of the left v_l and right v_r wheels.

3.3.3 Independent Wheel Offset Steering

One of the benefits of the IWOS over the DD is the capability of more complex motions. This can be described by the ICC and the possible locations of the ICC. In order to describe a motion, calculation of the ICC based on steering angles might be of interest. Figure 3.5 depicts the calculation of the ICC which will be described in detail in the following. A reverse calculation, steering angles based on the ICC might be of interest for motion planning, showcased in Figure 3.6.

In order to calculate the ICC with a given set of steering angles α_l and α_r equations for the functions defining the left l_l and l_r right revolute axes have to be found. The position of the the left and right steering joints S and the revolute joints R are defined below.

$$\vec{S}_l = \begin{bmatrix} s, & \frac{l}{2} \end{bmatrix}^T \tag{3.6}$$

$$\vec{S}_r = \begin{bmatrix} s, & -\frac{l}{2} \end{bmatrix}^T \tag{3.7}$$

$$\vec{R}_l = \begin{bmatrix} s\left(1 - \cos\left(\alpha_l\right)\right), & \frac{l}{2} - s \cdot \sin\left(\alpha_l\right) \end{bmatrix}^T \tag{3.8}$$

$$\vec{R}_r = \begin{bmatrix} s\left(1 - \cos\left(\alpha_r\right)\right), & -\frac{l}{2} - s \cdot \sin\left(\alpha_r\right) \end{bmatrix}^T \tag{3.9}$$

Therefore, the normalized vectors orthogonal to the left $\vec{n}_l$ and right $\vec{n}_r$ revolute axes are given by as defined below.

$$\vec{n}_l = \overline{S_l R_l} = \begin{bmatrix} \cos\left(\alpha_l\right), & \sin\left(\alpha_l\right) \end{bmatrix}^T \tag{3.10}$$

$$\vec{n}_r = \overline{S_r R_r} = \begin{bmatrix} \cos\left(\alpha_r\right), & \sin\left(\alpha_r\right) \end{bmatrix}^T \tag{3.11}$$

Based on the general normal form $(\vec{x} - \vec{p}) \cdot \vec{n} = 0$, functions for the revolute axes are given by the equations below.

$$l_l : \left(\vec{x} - \vec{R}_l\right) \cdot \vec{n_l} = (x - s)\cos(\alpha_l) + \left(y - \frac{l}{2}\right)\sin(\alpha_l) + s = 0 \tag{3.12}$$

$$l_r : \left(\vec{x} - \vec{R}_r\right) \cdot \vec{n_r} = (x - s)\cos(\alpha_r) + \left(y + \frac{l}{2}\right)\sin(\alpha_r) + s = 0 \tag{3.13}$$

The intersection of lines defined by the functions l_l and l_r is the ICC. There are three cases which are of interest when calculating the ICC.

1. Both the left and the right steering angle are at neutral position, $\alpha_l = \alpha_r = 0$. Applying these values to either of the Equations 3.12 and 3.13 results in $x = 0$. When calculating the intersection of l_l and l_r, the result is a tautology, and thus the revolute axes are identical and have infinite intersection points. The ICC can be located anywhere on these axes, such as is the case for the DD.

2. The left and the right steering angle are identical but not at neutral position $\alpha_l = \alpha_r \neq 0$. Employing these values in Equations 3.12 and 3.13 the resulting equations are deemed valid. Nevertheless, when calculating the intersection of l_l and l_r the result is a false statement, thus the revolute axes are parallel and no intersection points of the axes exist. This steering position places the ICC at infinity by definition.

3. The left and right steering angle are not equal, $\alpha_l \neq \alpha_r$ and either $\alpha_l \neq 0$ or $\alpha_r \neq 0$. In this case a well-defined intersection point of the left and right revolute axes exists, which is the ICC. The ICC's x-coordinate x_{ICC} and y-coordinate y_{ICC} within the robot's coordinate system are given by

$$x_{ICC} = \frac{s \cdot \left(\frac{\sin(\alpha_l)}{\sin(\alpha_r)} + \cos(\alpha_l) - \cot(\alpha_r) \cdot \sin(\alpha_l)\right) + l \cdot \sin(\alpha_l)}{\cos(\alpha_l) - \cot(\alpha_r) \cdot \sin(\alpha_l)} \tag{3.14}$$

and

$$y_{ICC} = \frac{s \cdot \left(1 - \frac{\cos(\alpha_l)}{\cos(\alpha_r)}\right) + \frac{l}{2}\left(\sin(\alpha_l) + \tan(\alpha_r) \cdot \cos(\alpha_l)\right)}{\sin(\alpha_l) - \tan(\alpha_r) \cdot \cos(\alpha_l)}. \tag{3.15}$$

The radius of the planar rotational motion is given by the Euclidean distance of the robot's center from the ICC

$$r = \sqrt{x_{ICC}^2 + y_{ICC}^2}. \tag{3.16}$$

Figure 3.5 shows the construction of the ICC based on a set of given steering angles α_l and α_r.

The online tool Desmos[2], a graphical online calculator has been utilized as an interactive visualization of the ICC. Both α_l and α_r can be set and the axes of the left (red) and right (green) wheel are visualized. The intersection of these lines is the ICC. Note that the robot is facing right, therefore the x-axis is horizontal and the y-axis is vertical. Figure 3.4 shows a QR-code which can be used to access the tool. Moreover, find the link to access the tool in the figures caption. The author is committed to ensuring that the tool remains accessible via the provided link and QR-code for as long as possible. However, it's important to note that the author cannot guarantee the perpetual availability of the tool through these means.

Figure 3.4: QR-code to access the tool showcasing the ICC construction. Alternatively, access the tool via the link: `https://qrcc.me/rycw29n9rgnn` (accessed on 09.08.2023).

In all three cases, the third wheel, which is not driven, is pushed by frictional forces into a position such that the wheel axis intersects with the ICC.

It is also possible to calculate steering angles for a given ICC. Calculation of steering angles based on a given ICC fall into one of two categories.

1. The ICC is located at infinity. In this case the revolute axes have to be parallel, thus the left and right steering angles are equal. It is not possible to gather any additional information about the steering angles.

2. The ICC has a finite x-coordinate and y-coordinate in the robot's coordinate system. It is possible to calculate at least one and at most two sets of steering angles leading to the given ICC. For this purpose, a line from the ICC to the left and right revolute joint is required. Circles with the diameter $\overline{ICCR_l}$ and $\overline{ICCR_r}$ can be constructed. Now we can make use of Thale's theorem [F+08]. The intersections of those circles with the respective circular arc of possible positions for the revolute joints are candidates for positions for the revolute joints to achieve the given ICC.

   For a valid ICC, where we know that a configuration of α_l and α_r exists, we get at least one intersection of the circle created with the aforementioned arc. It is

[2] Desmos is an online graphing calculator and math visualization tool, for more details visit `https://www.desmos.com/` (accessed on 09.08.2023)

possible that there are two intersection points for one of the two circles. In this case there are either two (if $2 \cdot (s+r) < l$) or four possible configurations leading to the given ICC. In general an ICC with more than on possible configuration is close to the left or right wheel in it's neutral position. Therefore, we can differentiate between an unambiguous and ambiguous ICC.

The number of possible configurations depends on the specific robot and the location of the ICC. The robot's specific parameters α_{MAX} and α_{MIN} as well as wheelbase l and wheel offset s determine the number of possible sets of steering angles for a given ICC.

Figure 3.6a depicts the calculation of α_l in the case of an unambiguous ICC. The calculation of α_r works in the same manner, but is omitted to ensure clarity of the visualization. Figure 3.6b depicts the calculation of α_l in the case of an ambiguous ICC. The calculation of α_r works in the same manner but is not ambiguous and is omitted to ensure clarity of the visualization..

In general, the design of IWOS imposes an upper and a lower limit for the steering angle. In the following section, the upper limit will be denoted by α_{MAX} and the lower limit will be denoted by α_{MIN}, it is assumed to be the same for the left and right steering joints. Additionally, we will assume that the neutral position of $\alpha_0 = 0$ and the limits are symmetric $\alpha_{\text{MAX}} = -\alpha_{\text{MIN}}$.

The resulting area of possible ICC locations is shown in Figure 3.7.

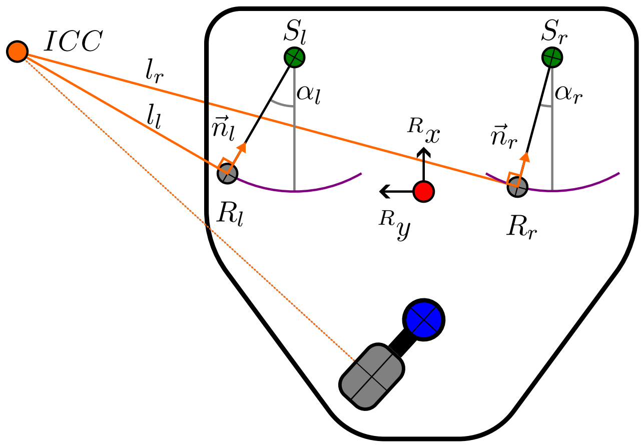
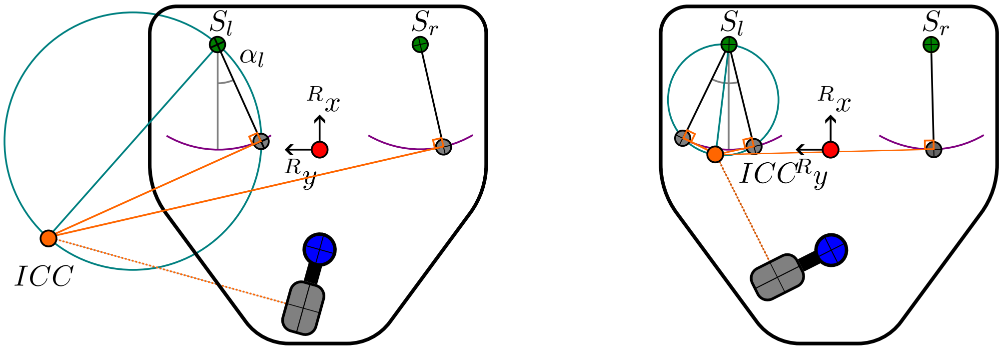


Figure 3.5: Construction of the ICC based on a set of given steering angles α_l and α_r where $\alpha_l \neq \alpha_r = 0$ holds. Green dots represent the steering joints. Grey dots represent the revolute joints and the purple arc depicts possible positions of the revolute joint. The caster wheel is depicted with a grey box for the wheel and a blue dot for the swivel joint. Lines l_l and l_r are depicted with solid orange lines, and the wheel axis of the caster wheel is depicted with a dotted orange line. The ICC is indicated by an orange circle. The red circle marks the center of the robot's coordinate system.

Facing and Driving Direction

Facing direction is defined by the robot's coordinate system, see Section 3.2. Using the ICC the driving direction can formally be defined. Driving direction is orthogonal to the line that connects the ICC to the center of the robot's coordinate system. We designate the opening angle between facing and driving direction as κ. In Section 3.2 we define the



(a) Unambiguous calculation of α_l with a given ICC.

(b) Ambiguous calculation of α_l with a given ICC.

Figure 3.6: Calculation of the steering angle with a given unambiguous ICC and a given ambiguous ICC. Green dots represent the steering joints. Grey dots represent the revolute joints and the purple arc depicts possible positions of the revolute joint. The caster wheel is depicted with a grey box for the wheel and a blue dot for the swivel joint. The constructed circles are depicted in teal with the diameter depicted by a teal line. Lines l_l and l_r are depicted with solid orange lines, and the wheel axis of the caster wheel is depicted with a dotted orange line. The ICC is indicated by an orange circle. The red circle marks the center of the robot's coordinate system.

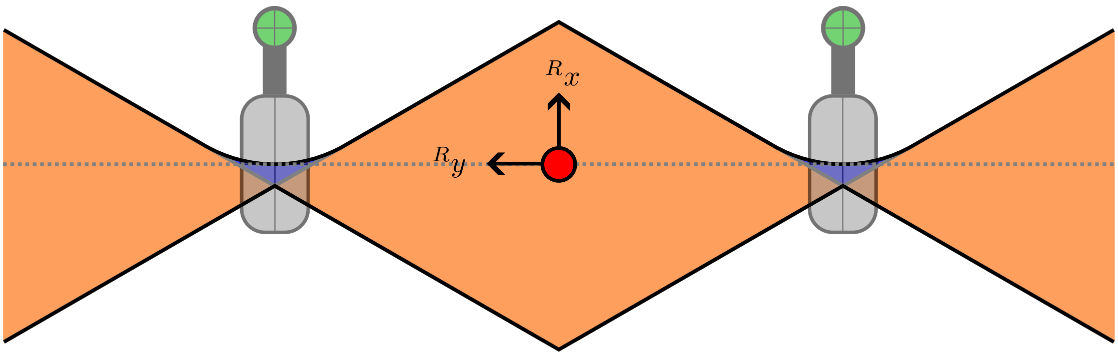


Figure 3.7: Area of possible ICC locations for the IWOS. The area of unambiguous ICC locations is depicted in orange, and the are of ambiguous ICC locations is depicted in blue. The area of possible ICC locations for the DD and the IWOS with the restriction $\alpha_l = \alpha_r = 0$ is depicted with a grey dotted line.

orientation term of the pose as the robot's facing direction. Therefore, for a complete description of the robot, besides the pose the opening angle κ is necessary.

Note that it is not possible to find an unambiguous ICC based on the robot's pose $\mathbf{x}$ and the opening angle κ. All points on the line orthogonal to driving direction that are within the area of valid ICC positions are applicable candidates for the ICC.

When the robot is rotating on spot the ICC is located in the center of the robot's coordinate system and $r = 0$ as well as $v = 0$. In this case the aforementioned calculation of the driving direction is not applicable. Therefore, we define that the driving direction is equal to the facing direction if the ICC is located at the origin of the robot's coordinate system. In doing so, we align the definition of the facing and driving direction with the DD.

For path and motion planning it might be beneficial if the orientation term θ of pose would refer to the driving direction instead of the facing direction. Nevertheless, many well established tools assume that the orientation term θ of pose refers to the robot's facing direction. These tools would require an additional coordinate system transformation in order to operate properly. Thus, to ensure seamless compatibility of the IWOS with existing tools the pose refers to the robot's facing direction.

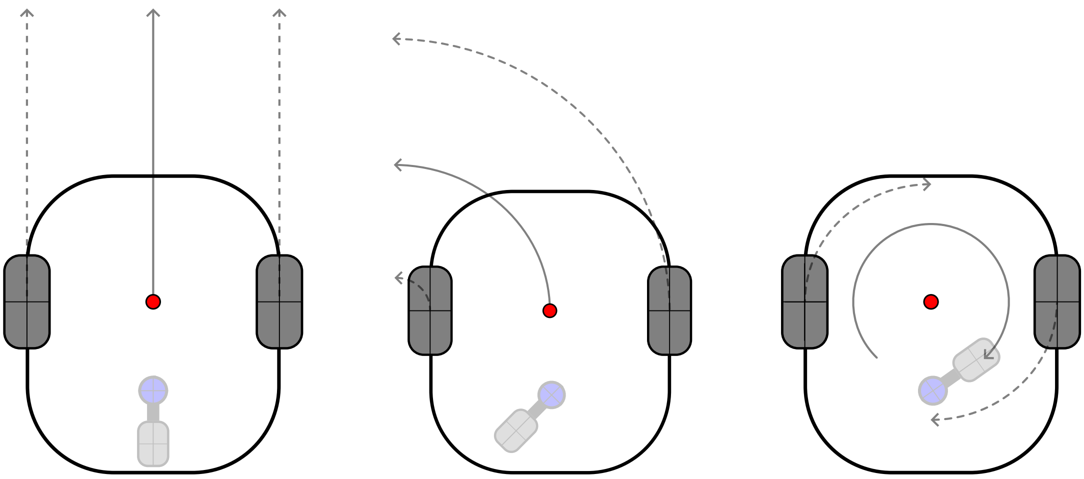
3.4 Maneuverability

The basic motions of the DD and IWOS are capable of can be identified and classified based on the location of the ICC.

For the DD there are three basic motions.

1. Straight line motions, where the ICC's y-coordinate is infinite. This is achieved by driving both wheels with the same velocity $v_l = v_r$. This motion is depicted in Figure 3.8a.
2. Arc line motions, where the ICC's y-coordinate is finite. This is achieved by a different velocity for the left and right wheels. The relation between the radius of the arc line the relation in Equation 3.4 has to hold for the ratio between the left and the right wheel velocity. A left curve motion is driven when $v_l < v_r$ and a right curve motion is driven when $v_l > v_r$ In Figure 3.8b this type of motion is illustrated.
3. Rotation on spot, where the ICC is located exactly in between the left and the right wheel. For this kind of motion the wheels have to spin with equal velocity but in opposite directions, $v_l = -v_r$. A counter-clockwise rotation requires $v_l < v_r$ and a clockwise rotation requires $v_l > v_r$. Figure 3.8c depicts this kind of motion.

The IWOS is capable of the motions listed above, since the set of possible ICC locations of the DD is a subset of the possible ICC locations of the IWOS. In addition to these motions, we can identify the following motions for the IWOS.



(a) Straight motion with $v_l = v_r$.

(b) Arc line motion with $v_l < v_r$.

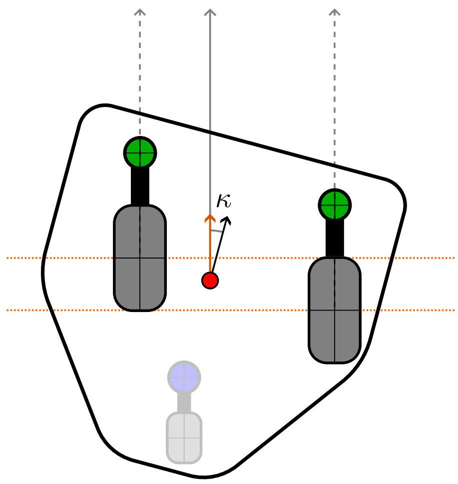
(c) Rotation on spot with $v_l = -v_r$ and $v_l < v_r$.

Figure 3.8: Motion categories for the DD. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion.

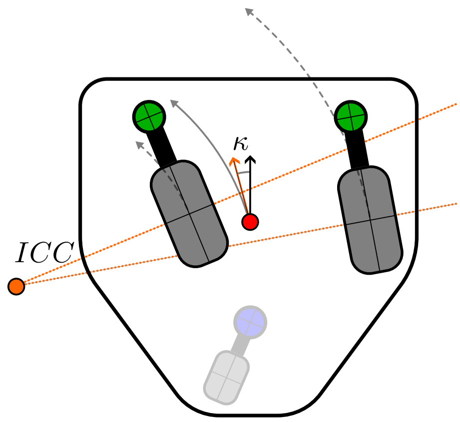
(a) Straight line motions in which the steering joints are not in their neutral position and the steering angle for both steering joints is equal. These are straight line motions where the facing and the driving direction have an offset equal to the negative steering angle, $\kappa = -\alpha_l = -\alpha_r$. By definition, the ICC is located at infinity. Figure 3.9a depicts this kind of motion.

(b) Arc line motions where the ICC's x-coordinate is not equal to zero and the ICC's y-coordinate is finite. These motions can be split into two subcategories based on the ICC's x-coordinate, ICC_x.
Firstly, $ICC_x > \frac{l}{2}$ or $ICC_x < -\frac{l}{2}$, where the κ can have values within the range $\alpha_{MIN} < \kappa < \alpha_{MAX}$. For such a motion the wheels spin in the same direction but with different velocity. Note that the cases $\kappa = 0$ and $\kappa = \alpha_{MIN}$ or $\kappa = \alpha_{MAX}$ would fall into other categories of this classification. A motion of this kind is depicted in Figure 3.9b.
Secondly, $-\frac{l}{2} \leq ICC_x \leq \frac{l}{2}$, where κ can have values within the range $-\pi \leq \kappa < 0$ and $0 < \kappa \leq \pi$. Again, the case $\kappa = 0$ would fall into another category. If the ICC is located in the area $-\frac{l}{2} < ICC_x < \frac{l}{2}$ and for the steering angles $\alpha_l = -\alpha_r$ holds, a valid motion requires $v_l = -v_r$. Such a motion is similar to a rotation on spot for the DD, but for the IWOS such a motion changes both the orientation and position of the robots center. A motion of this kind is depicted in Figure 3.9c.

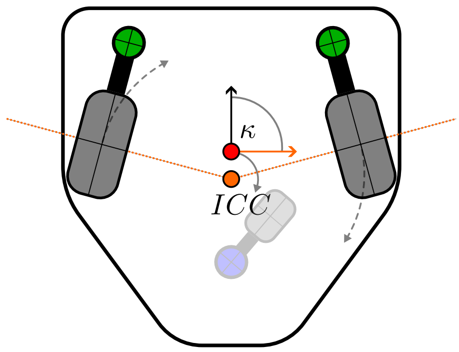
For the described motions, the wheels move solely along the wheel plane, which is perpendicular to the wheel axis adhere to the sliding constraint [SNS11].



(a) Straight motion with steering joints not in neutral position and infinite ICC.



(b) Arc line motion with steering joints not in neutral position and finite ICC and $v_l \neq -v_r$.



(c) Arc line motion with steering joints not in neutral position and finite ICC and $v_l = -v_r$.

Figure 3.9: Motion categories for the IWOS. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. Wheel axes of wheels driven are depicted by dotted orange lines. The black arrow represents the robot's facing direction and the orange arrow represents the driving direction.

However, empirical know-how gathered in the terms of this thesis, indicates that the IWOS is also capable of executing motions that do violate the sliding constraint. These specific motions are necessary for transitioning the robot's state between the aforementioned motions. While these motions may temporarily deviate from the sliding constraint, they play a crucial role in enabling the robot to achieve desired movements and perform complex maneuvers effectively.

Addressing one of the research goals of this thesis, we aim to summarize the differences in maneuverability between the IWOS and DD locomotion systems.

The DD locomotion system provides a strong foundation in terms of motion capabilities. Its range of motions allows a robot utilizing this system to navigate with versatility. By adjusting the velocity of the steered wheels, the robot can easily change its motion by shifting the location of the ICC. With just three simple motions - rotation, translation, and rotation - the robot can reach any position for any other position in an unoccupied environment.

The maneuverability of the IWOS depends on the position of its steering joints. When the steering joints are in their neutral position, the IWOS is capable of the same kinds of motions as the DD. However, if the steering joints deviate from their neutral position, the IWOS is limited in terms of motion. In such cases, changing the position of the ICC involves adjusting the position of the steering joints, which may violate the sliding constraints of the wheels. The possibility of violating these constraints varies depending on the specific robot and its environment. Consequently, further research focusing on the ICC adjustment in the IWOS locomotion system is necessary.

3.5 Degrees of Freedom

In Section 3.3 we introduce the ICC, a tool useful to describe planar motions. With the ICC and either the angular velocity or the translation velocity, a planar motion is defined. The ICC requires an x-coordinate and a y-coordinate to be specified. The angular velocity ω is valid for the whole robot in motion, while the translation velocity v refers to the robot's center. Thus, such a motion has three degrees of freedom: the x-coordinate and y-coordinate of the ICC and either the angular velocity ω or the translation velocity v.

The ICC of the DD is restricted to a one-dimensional space, as discussed in Section 3.3.2. One degree of freedom is retained by the robot's structure and any planar motion has two degrees of freedom. Therefore, the two actuated joints of the DD are sufficient to control the robot's planar motion.

As discussed in Section 3.3.3, the ICC of the IWOS is also restricted, but still has two degrees of freedom. Thus, for the IWOS, planar motions have three degrees of freedom. The number of joints that can be actuated for the IWOS is four, in contrast to a motion with three degrees of freedom. Consequentially, it is sufficient, but not necessary, to actuate only three of the four joints. Of the four joints of the IWOS, either a revolute joint or a steering joint could operate unactuated in any given moment.

Firstly, we can assume that one revolute joint is not actuated while the other revolute joint and the steering joints are actuated. The ICC's position would be defined by the steering angle and the velocity would be determined by the single revolute joint. The other revolute joint would follow the motion, such that Equation 3.17 is satisfied.

$$\left|\overline{ICCR_l}\right| = \left|\frac{v_l}{\omega}\right| \quad \text{and} \quad \left|\overline{ICCR_r}\right| = \left|\frac{v_r}{\omega}\right| \tag{3.17}$$

Secondly, we can assume that one revolute joint is not actuated while the other revolute joint and the steering joints are actuated. The actuated steering joint would determine a line of possible locations for the ICC by steering the position of the revolute joint and therefore the position and orientation of the revolute axis. By controlling the velocity of the left and right revolute joints, the exact location of the ICC would be determined, such that the Equations 3.17 hold. The unactuated steering joint would be pushed into correct orientation by frictional forces on the wheel.

Nevertheless, it is also possible to actuate all four joints when taking the constraints into consideration, ensuring that the actuation is coherent for all joints. An incoherent actuation of the four joints can result in excessive stress on the mechanical components, potentially leading to their failure and breakage.
The benefits and drawbacks of these control modes are beyond the scope of this work and are a topic open to further research.

In this section the fundamentals of the IWOS have been discussed. The theoretical knowledge gathered finds application in a prcatical implementation of this novel robot locomotion system. In the following chapter the construction of a prototpye of the IWOS will be discussed.

In this section, we have discussed the fundamental principles underlying the IWOS. The theoretical knowledge gained serves as a solid foundation for the practical implementation of this robot locomotion system. Building upon the theoretical insights, the subsequent chapter will focus on the construction and realization of a prototype of the IWOS. The following chapter will provide an account of the design considerations and integration of the various hardware and software components that constitute the IWOS prototype. By exploring the practical implementation, we aim to bridge the gap between theory and application, showcasing the feasibility and potential of this novel locomotion system in a tangible and demonstrable manner.

CHAPTER 4

Proof of Concept for the Independent Wheel Offset Steering

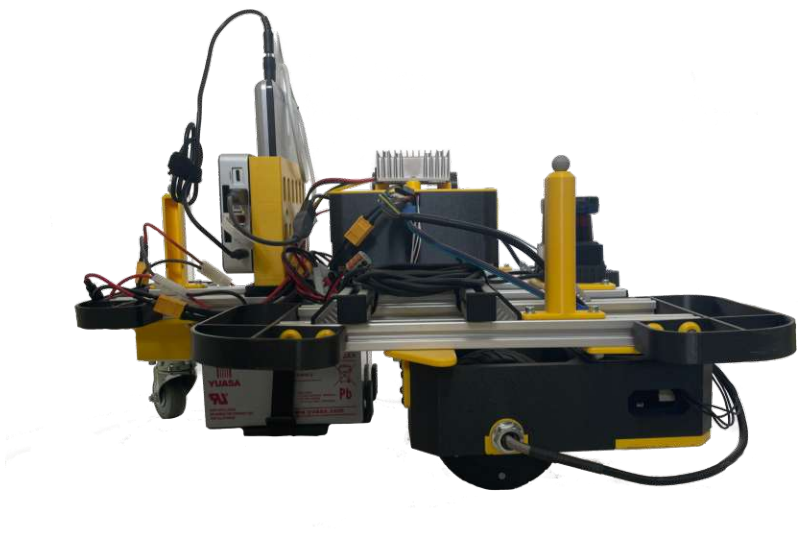
The IWOS is an innovative propulsion mechanism designed for mobile robots, which was invented and subsequently patented by the Technical University of Vienna. With permission granted to the author of this thesis, a prototype of this protected design was constructed. On the one hand, the prototype is intended to showcase the functionality and possibilities of the IWOS. On the other hand, it serves as an indispensable tool, enabling a range of experiments to explore and evaluate the capabilities of this propulsion mechanism.

The construction of this prototype showed that robotics is indeed an interdisciplinary field since challenges stemming from diverse fields such as machine engineering, electronic engineering and of course software engineering had to be solved. Figure 4.1a as well as Figure 4.1b show pictures capturing the proof of concept robot, depicting its visual appeal.

In this chapter, the construction of the prototype will be envisioned. Firstly, we will delve into the requirements and non-requirements associated with the prototype. Secondly, we will explore the hardware selection for the robot. Lastly, we will delve into the software responsible for operating the robot.

4.1 Requirements

The objective of this prototype is to create the opportunity to conduct experiments using this novel locomotion mechanism and, at the same time, provide proof of concept for the IWOS.



(a) Side view picture of the IWOS prototype.

(b) Front view picture of the IWOS prototype.

Figure 4.1: Pictures showing the side and front of the prototype robot featuring IWOS.

For this purpose the following requirements for the IWOS robot prototype were formulated:

- **Power Autonomy:** The robot shall be equipped with a power supply system to ensure operation without the need for connection to an external power source. The power supply system should provide sufficient capacity to support the robot's intended purpose. It should also be easily rechargeable or replaceable to minimize downtime during charging cycles.

- **Predetermined Breaking Point:** The prototype shall possess a frame with predetermined breaking points acting as a cushioning to prevent damage to both, the robot and it's environment.

- **Framework:** ROS is a state-of-the-art framework for robotics applications, especially for research purposes. Therefore, the prototype shall utilize ROS for operation and communication.

- **Operability:** The robot's joints should be independently controllable, where the steering joints take a desired position as input and the revolute joints take a desired velocity as input.

- **Joint State Information:** Providing information about the state of the joints, including the rotational position and rotational velocity is required. Information about the force applied to the joints is beneficial.

- **Odometer:** Providing the position and orientation of the robot based on the state of the joints is necessary. This basic approach serves a solid foundation for odometry data with room for improvements. Incorporating additional sensor information into the odometry calculation could result in enhanced accuracy [MHW+19]. Nevertheless, for this thesis a naive approach for odometry is sufficient leaving more advanced odometry approaches open to further research.

- **Extensibility:** The robot should be flexible and adaptable, allowing for changes such as repositioning or replacing components, as well as expanding its capabilities by incorporating additional components.

In addition to these requirements, it is important to clarify the non-requirements associated with this particular prototype. The following list enumerates aspects that may commonly be associated with a robotic prototype but are explicitly considered as non-requirements for this specific prototype:

- **Autonomous operation:** The prototype is not required to operate autonomous. Autonomous operation is not necessary for the experiments for this thesis and is open to future research.

- **Precise maneuverability:** It is not necessary that the robot performs exactly the same under variable circumstances, such as different load, different temperatures and varying battery level. Reliability and consistency are a necessity for robots in industrial applications but non-essential, but potentially beneficial, for research robots.

- **Sensors:** While additional sensors beyond those necessary for the steering joints and revolute joints may not be essential, it is worth considering their inclusion, particularly for future research endeavors.

The hardware and software for this prototype was tailored to these requirements and non-requirements. During the design and implementation phase, special attention was given to ensuring that the prototype remains open to future changes and improvements.

4.2 Hardware

The hardware for this prototype can be categorized into three groups. Firstly, the hardware that was available and consequently selected. Secondly, the hardware that was sourced specifically for this prototype, carefully chosen to align with the unique characteristics of the robot. During the selection process, the potential for its use in other robotic applications was also considered. Lastly, the hardware that was custom-built for this robot, encompassing parts created from scratch. These components were designed using Computer Aided Design (CAD)[1] software and fabricated through 3D-printing techniques.

4.2.1 Processor

The inclusion of a processor is an indispensable component in enabling a robot to operate and potentially operate autonomously. In the case of mobile robots, an onboard computer assumes the responsibility of controlling the joints and managing the sensor data. Advanced tasks, such as path planning, can be executed either on the onboard computer or delegated to a cloud-based counterpart. Nonetheless, the onboard computer must possess sufficient power to ensure proper operation of the joints and effectively handle sensor data, whether through processing or forwarding it.

For the IWOS prototype, an Intel NUC was available. The Intel NUC houses an Intel i3-5010U processor with 8GB of RAM, rendering it an optimal compromise between compactness, power efficiency, and computational prowess. Operating on Ubuntu 20.04.6[2] and running Docker[3], the onboard computer effectively handles the computational demands of the IWOS prototype.

[1]CAD software, is a powerful computer application utilized for creating precise and detailed two-dimensional and three-dimensional digital models. Such models can then be seamlessly transferred and utilized for production through technologies such as 3D printing.

[2]For more details about Ubuntu 20.04.6, visit: `https://www.releases.ubuntu.com/focal/` (accessed an 09.08.2023).

[3]Docker is a platform that allows you to package and distribute applications along with their dependencies in a lightweight, isolated container. For more details visit: `https://www.docker.com/` (accessed on 09.08.2023)

4.2.2 Steering Joints

The steering joints play a central role in the IWOS, setting it apart from the conventional DD mechanism. The basic requirement for the steering joint is a servo motor capable of supporting position control with sufficient torque to maneuver the fork and wheel effectively. In order to fully exploit the potential of the maneuverability of the IWOS, the integration of accurate position sensing and control is essential. This enables precise regulation of the steering angle, facilitating precise positioning of the ICC. Furthermore, incorporating torque control and torque sensing in the steering joints proves advantageous. These features allow for the detection of potential collisions based on the applied torque to the steering joints, contributing to enhanced collision detection and collision handling. It is advantageous, but not necessary, that the servo motor is small enough to be integrated directly into the fork.

The Robotis Dynamixel MX-106R[4] servo motor was selected for the steering joints due to its alignment with the aforementioned requirements and its capability to sense position, velocity and current.

To facilitate control of these servo motors, the Robotis Dynamixel U2D2[5] USB communication converter is employed, connecting it to the computer of the robot. The U2D2 limits the operation frequency of motors connected to approximately $30Hz$. This includes the availability of joint state information, such as position, velocity and effort. Operating the U2D2 with the Dynamixel-SDK[6] provides compatibility with C++ and Python besides other programming languages.

4.2.3 Revolute Joints

The wheels, designated as revolute joints within the IWOS propulsion mechanism, hold paramount importance alongside the steering joints. On one hand, large wheels prove advantageous in robotics applications as they can effectively traverse surface bumps compared to smaller wheels. On the other hand, employing small wheels permits positioning them beneath the robot's main surface, despite their lateral motion which is a characteristic feature of the IWOS. For this prototype, hub motor wheels with a diameter of approximately ten centimeters have been selected. This diameter allows the robot to be designed in a manner, where its base surface is positioned above the wheels. These motors are brushless direct current motors, requiring a 24-volt power

[4] For more information on the Robotis Dynamixel MX-106R, refer to: `https://emanual.robotis.com/docs/en/dxl/mx/mx-106-2/` (accessed on 09.08.2023).

[5] For detailed specifications of the Robotis Dynamixel U2D2, visit: `https://emanual.robotis.com/docs/en/parts/interface/u2d2/` (accessed on 09.08.2023).

[6] The Dynamixel-SDK is open source and available on GitHub, for more details visit: `https://github.com/ROBOTIS-GIT/DynamixelSDK` (accessed on 09.08.2023).

supply for operation. The motors have a build in hall sensor[7] allowing to receive detailed information about the motor position and velocity.

The revolute joints are controlled using the Trinamic TMCM-1640[8], which serves as an appropriate choice for the provided motors. The drive module is capable of reading data from the hall sensor at the motor. By leveraging the sensor readings and integrating current sensing, the TMCM-1640 has the capability to effectively monitor and measure the position, velocity, and current of the connected motor. The Trinamic TMCM-1640 can be operated with $120Hz$, which includes the availability of joint state information, such as position, velocity and effort.
Trinamic controllers support operation with both C++ and Python programming languages, facilitating seamless integration into the robot's control system.

4.2.4 Frame and Frame Components

The robot's frame, serving as the main surface for this prototype, is constructed using Bosch Rexroth aluminum profiles. These profiles offer excellent structural integrity, durability, and versatility rendering them a commendable choice for research applications in robotics.

The fork, encompassing the steering joint and revolute joints, was designed and build for this prototype solely. The design is tailored for the requirements of this robot using FreeCAD[9]. A picture of the fork designed in FreeCAD is shown in Figure 4.2a. For the fabrication of the designed parts a Prusa 3D-printer[10] was utilized, printing with Polyethylene Terephthalate Glycol (PETG)[11] material in Fused Deposition Modeling (FDM) technique, resulting in a robust and precise component. Figure 4.2b depicts a screenshot of the processing of the 3D model with PrusaSlicer[12].

For the purpose of facilitating further research, the CAD file of the fork can be made available upon request.
Beside the fork, other components such as a hardware enclosure and various mounts have been designed and produced with FreeCAD and 3D-printing.

[7] A Hall sensor for measuring motor position and velocity is a device mounted on a rotating axis that detects changes in the magnetic field caused by magnets on the axis, allowing it to determine the position and speed of the motor.

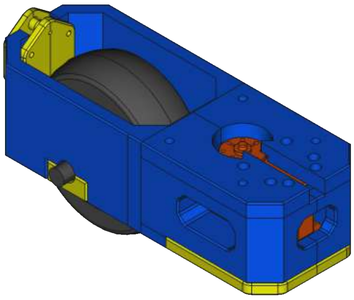
[8] For detailed specifications of the Trinamic TMCM-1640 drive module, visit: `https://www.trinamic.com/products/modules/details/tmcm-1640/` (accessed on 09.08.2023)

[9] FreeCAD is a free and open source CAD software, for more details visit `https://www.freecad.org` (accessed on 09.08.2023).

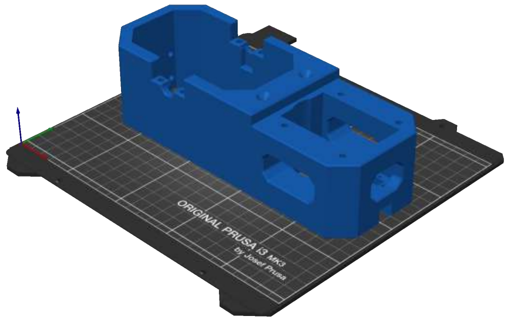
[10] Prusa 3D-printer are reliable open source devices, for more details visit: `https://www.prusa3d.com/` (accessed on 09.08.2023).

[11] PETG is a thermoplastic material commonly used in 3D-printing due to its high impact resistance, flexibility and durability.

[12] The powerful PrusaSlicer converts a 3D object into layers and finally into motion commands for the 3D-printer, for more details visit `https://www.prusa3d.com/page/prusaslicer_424/` (accessed on 09.08.2023)



(a) Picture of the fork created in FreeCAD. The main part of the fork is shown in blue, attachment parts are shown in yellow. The wheel and the servo are also shown in this picture in dark grey and red.



(b) Screenshot of the processing of the fork with PrusaSlicer. Note that the main part of the fork is placed upside down for printing. Attachment parts are not shown in this screenshot.

Figure 4.2: Pictures illustrating the design and production process of the fork.

An important aspect to consider is the location of the center of mass on the mobile robot. Placing the lead batteries, which are typically the heaviest components, in the central position behind the wheels is a strategic choice. This positioning ensures that the center of mass is encircled with the triangle formed by the contact points of the three wheels with the underlying surface. The benefit of this configuration lies in its inherent ability to ensure optimal stability for the robot.

The frame and components of the robot define important properties of the robot such as the wheelbase and the wheel offset as defined in Section 3.1.2. Table 4.1 shows the values of these properties.

Table 4.1: Properties of the Prototype

Property	**Value**
Wheelbase (l)	$0.35(m)$
Wheel Offset (s)	$0.1(m)$

4.3 Software

The software for the prototype was designed to be hardware independent allowing future iterations to exchange components or change the robots characteristics such as the wheelbase or the wheel-offset.

4.3.1 Robot Operating System

ROS is a flexible and widely adopted open-source framework for the development and control of robotic systems, encompassing both stationary and mobile platforms. The

most recent version, Noetic Ninjemys[13], was released in 2020 and has established itself as a mature and stable framework in the field of robotics. At the time this thesis was initiated in 2020, ROS2, the successor of ROS, was still in its early stages of development, lacking comprehensive support and some essential components crucial for this particular thesis. Therefore, ROS was chosen as the preferred framework, offering a more suitable environment for conducting the research and development required.
Since then, remarkable progress has been achieved with ROS2, reaching a commendable level of stability and comprehensive features. As a result, it is now feasible to migrate the software developed for this thesis to ROS2. This demonstrates the evolving nature of the ROS ecosystem and the feasibility of migrating projects from ROS to ROS2 for future developments.

ROS provides a rich set of libraries, tools, and conventions that ease the development of robot software. It promotes a modular and reusable approach, allowing developers to seamlessly integrate diverse hardware and software components.
The central component of ROS is the ROS core, serving as the centralized hub for communication in a ROS-based system. It coordinates the interaction between various components and facilitates the exchange of data among them. The core infrastructure of ROS encompasses several key elements, including the ROS Master, which manages the naming and registration of nodes, and the ROS Parameter Server, which acts as a centralized repository for configuration parameters.

A fundamental concept in ROS are packages, which encapsulate related functionality and resources. A ROS package typically consists of libraries, executables, configuration files, and other resources. Within a package, individual executable components are referred to as nodes.

Nodes in ROS are autonomous processes that perform specific tasks. They can communicate with one another via messages. ROS nodes can send or receive messages by publishing and subscribing to topics, which represent a stream of data. This publish-subscribe mechanism enables efficient and decoupled communication between nodes. Additionally, nodes can provide or consume services, allowing for request-response interactions.

4.3.2 Implementation

For the IWOS prototype devised for this thesis several ROS packages were developed. These nodes are designed to provide a solid and stable interface to operate the robot. Upon request, these packages can be made available for research purposes.

The `tuw_iwos_hardware_broker` Package

The `tuw_iwos_hardware_broker_node` is a central element, the gatekeeper, to the control of the robot's hardware. This node actively awaits incoming commands for the

[13] For details about ROS Noetic Ninjemys, visit: `https://wiki.ros.org/noetic` (accessed on 09.08.2023).

robot and processes them by splitting them up and forwarding them. Incoming commands have to utilize the JointsIWS[14] message type. This versatile message type contains a information types field for the steering and revolute message type and information lists for the according values. An information type can either be a command or a measurement of position, velocity, acceleration or effort. An information value is expected to be represented as a decimal number.

The `tuw_iwos_hardware_broker` node expects to receive messages with position, velocity or effort commands and an according target value. After receiving a message the node splits the message into commands for each joint. The node verifies the compatibility between the received command type, target value, and the corresponding joint hardware and operation type for each joint. The operation type of a joint depends on the `ros_controller` currently responsible for the joint. On a successful verification the command is forwarded to the `ros_controller` of the designated joint.

Due to weak definitions in the messages, the `tuw_iwos_hardware_broker` node has to make two assumptions. Firstly, the unit of commands is not defined, and it is assumed that messages follow the conventions of standard units of measurements as defined in REP 103[15]. The expected command types for the steering joints are listed in Table 4.2 and for revolute joints in the Table 4.3.

Table 4.2: Units for Commands for Steering Joints

Quantity	**Unit**
Position	radians (rad)
Velocity	rad per second ($\frac{rad}{s}$)
Acceleration	meters per second squared ($\frac{rad}{s^2}$)
Effort	amperes (A)

Table 4.3: Units for Commands for Revolute Joints

Quantity	**Unit**
Position	meter (m)
Velocity	meters per second ($\frac{m}{s}$)
Acceleration	meters per second squared ($\frac{m}{s^2}$)
Effort	amperes (A)

Secondly, the commands are not accompanied by an identifier indicating the corresponding joint to which the command belongs. The commands are split into lists of steering and

[14] For detailed information about the structure of JointState messages, visit: `https://github.com/tuw-robotics/tuw_msgs/blob/ros2/tuw_nav_msgs/msg/JointsIWS.msg` (accessed on 09.08.2023).

[15] In REP 103 a reference for units and coordinate conventions is provided, for more details visit: `https://github.com/ros-infrastructure/rep/blob/master/rep-0103.rst` (accessed on 09.08.2023).

revolute commands but no identifier for the specific steering joints and revolute joints are provided. For this reason it is assumed that the values in the list correspond to the joints in alphabetical order. This could be enhanced by a service providing the information about the intended attribution of the commands to the joints.

The `tuw_iwos_hardware` Package

This package contains and includes the nodes which are responsible for controlling the robots hardware. The `tuw_iwos_hardware` package makes use of the `ros_control`[16,17], a ROS package for hardware abstraction and hardware management.

The `ros_control` package is a collection of executables and interfaces aiming to provide a standardized way to control robot hardware. It consits of two main components, the `controller_manager` which manages the execution of various controllers, and the `hardware_interface::RobotHW` which connects the robot's hardware to the control system. Figure 4.3 depicts the data flow of `ros_control`.

The `controller_manager` is responsible to provide a controller for each joint. These controllers can take two forms: a simple pass-through controller, which maintains the input command as is, or a more intricate controller that employs advanced control techniques to modify the input commands. The `ros_controllers`[18] package contains pass trough controllers and proportional–integral–derivative (PID)[19] controllers, as well as the interfaces necessary to implement controllers tailored to specific use-cases. For the prototpye of the IWOS the pass trough controller is a sufficient choice since both, the steering joint hardware and the revolute joint hardware have built in PID controllers. The `controller_manager` as well as `ros_controller` are separate nodes allowing to change the specific controller for a joint at run time.

The actual hardware is operated in a single node, the `tuw_iwos_hardware` node. For each hardware component the `hardware_interface::RobotHW` has to be implemented, inlcuding the joint command interface and the joint state interface. This interfaces provide an abstraction for the tasks of writing commands to the hardware and reading the hardware state, allowing to operate hardware without the knowledge of hardware specific details. Reading the joint state is crucial for the controllers to operate properly. Additionally, the `tuw_iwos_hardware` node assumes the crucial duty of enforcing predefined limits on the joints, guaranteeing their safe operation. Such limits can be defined in the robot's description, a definition of the robot's joints and links in

[16]For details on the implementation of the `ros_control` package, visit: `https://wiki.ros.org/ros_control` (accessed on 09.08.2023).

[17]The official documentation on how to implement the `ros_control` is sparse, for an unofficial but more detailed documentation visit: `https://fjp.at/posts/ros/ros-control/` (accessed on 09.08.2023).

[18]For details on the implementation of the `ros_controllers` package, visit: `https://wiki.ros.org/ros_controllers` (accessed on 09.08.2023).

[19]A PID controller is a feedback control mechanism that provides an output signal based on the proportional, integral and derivative terms, aiming to achieve desired system behavior.

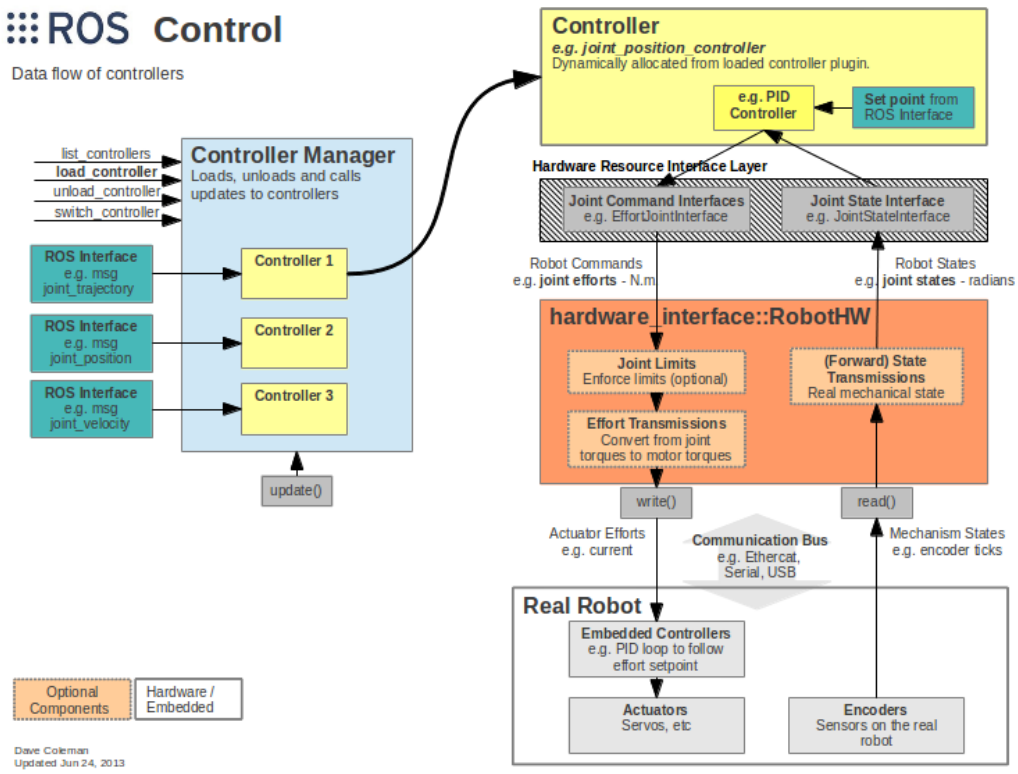


Figure 4.3: Picture of data flow of `ros_control` by Dave Coleman [CMEM+17].

form of the Unified Robot Description Format (URDF). This format allows to define hard and soft limits for the position, velocity, acceleration, jerk and effort of a joint.

The `tuw_iwos_hardware` node is designed to be launched with the `robot_state_publisher` in parallel.
The `tuw_iwos_hardware` node is responsible for reading the robot's description file and to make the content available to other nodes. Furthermore, this node can access the joint state interface and thus read the current state of the robot's joints. The node provides the gathered information to the `robot_state_publisher`.
The `robot_state_publisher` publishes the joint state information in form of the joint state messages[20] to the `joint_state` topic.

The coupling of `ros_control` and `robot_state_publisher` favors the use of the joint state message over the JointsIWS message to publish the joint state. An additional node could translate the joint state messages to JointsIWS messages.

[20]For detailed information about the structure of joint state messages, visit: `https://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/JointState.html` (accessed on 09.08.2023).

The prototype publishes the joint state with a frequency of $30Hz$. The frequency at which this information can be published is limited by the hardware operating with the lowest frequency. Therefore, the steering joint hardware is the bottle neck for the joint state publishing frequency.

The `tuw_iwos_odometer` Package

The `tuw_iwos_odometer` package encompasses two nodes. Firstly, the `tuw_iwos_odometer_node`, which subscribes to the joint state topic and publishes the calculated odometry data. Secondly, the `tuw_iwos_odometer_service` node, which provides a service for odometry calculation. Upon requesting this service, the client must provide two joint states and an initial pose to serve as the starting point for the odometry calculation. The service responds with the estimated pose based on the provided joint states. Both nodes provide the odometry data in form of odometry messages[21].

The calculation of the odometry data is discussed in detail in Section 5.1.

The `tuw_iwos_motion_model` Package

The `tuw_iwos_motion_model` node consumes odometry data in order to apply the OMM introduced in Section 6.2.

This package contains a node offering a service which applies the OMM sampling algorithm. A request to this services has to provide a starting pose including the offset between facing and driving direction as well as starting and ending odometry data and the offset between facing and driving direction for both these data points. Additionally, noise parameters and the number of desired samples has to be provided.

This package contains a node offering a ROS service, implementing the OMM sampling algorithm. When making a request to this service, several essential parameters need to be provided. Firstly, a starting pose must be included, specifying the offset between the facing and driving directions. Secondly, starting and ending odometry data, along with the respective offsets between facing and driving directions, are required. Thirdly, it is necessary to provide noise parameters. Finally, specifying the desired number of samples will enable the algorithm to generate the desired quantity of data points.

4.3.3 Deployment

In order to achieve platform independence and enhance the reproducibility, the software for the proof of concept robot is encapsulated within a Docker container [WC17]. By utilizing Docker, the robot's software can be deployed across different versions of ROS, ensuring compatibility and eliminating potential compatibility issues associated with

[21] For detailed information about the structure of odometry messages, visit: `https://docs.ros.org/en/noetic/api/nav_msgs/html/msg/Odometry.html` (accessed on 09.08.2023).

varying software environments. This approach greatly simplifies the management of dependencies.

The container housing the ROS packages mentioned earlier is built upon Ubuntu 20.04.06 and encompasses ROS Noetic Ninjemys. It necessitates access to the host system for the optimal operation of the robot's hardware.

4.4 Overview

In this chapter, an overview of both the requirements and non-requirements for the introduction of the IWOS is presented. Building upon these prerequisites, a examination of the hardware and software components of the IWOS prototype is provided. In this section, an overview of how these interconnected hardware and software elements collaborate is offered to enhance understanding. Figure 4.4 provides an overview of the hardware and software components and their communication. The components of this overview and their interaction are explained below.

The operating system of the robot is Ubuntu 20.04.6, utilizing Docker for software execution. The robot's operational software runs within a Docker container. Given that the robot's operation involves controlling hardware, the Docker container necessitates access to the host system.

The initial arbiter of the robot's software operation is the `tuw_iwos_hardware_broker` node. This node is crucial for receiving and processing motion commands for the robot in the form of JointIWS messages. These messages contain commands for each of the robot's joints, including target positions, velocities, efforts, or accelerations. The prototype can effectively realize target positions with its steering joints and target velocities with its revolute joints. As implied by its name, the `tuw_iwos_hardware_broker` node segregates incoming commands and forwards them to their respective joint controllers.

Each joint possesses its own dedicated controller. In this prototype, these controllers are simple pass-through controllers that do not modify the commands. However, if necessary, these controllers could be substituted with more sophisticated ones, potentially enhancing the motion of the corresponding joints. The controller also has access to the joint state, enabling comprehensive control and coordination.

The `tuw_iwos_hardware` node serves as the central entity responsible for managing all hardware components. This node facilitates communication with the hardware, encompassing the task of writing commands to the onboard controller and retrieving joint state information, including position, velocity, and effort. For this specific prototype, the Dynamixel U2D2 interfaces with both steering joints, while for each, the left and right revolute joint, a single Trinamic TMCM-1640 is connected. Despite being a single node, each joint possesses its own memory space and thread. This node also provides the joint state information to other components of the robot.

The `tuw_iwos_odometer` node employs the joint state information to calculate the robot's change in position, employing a straightforward approach that serves as a baseline

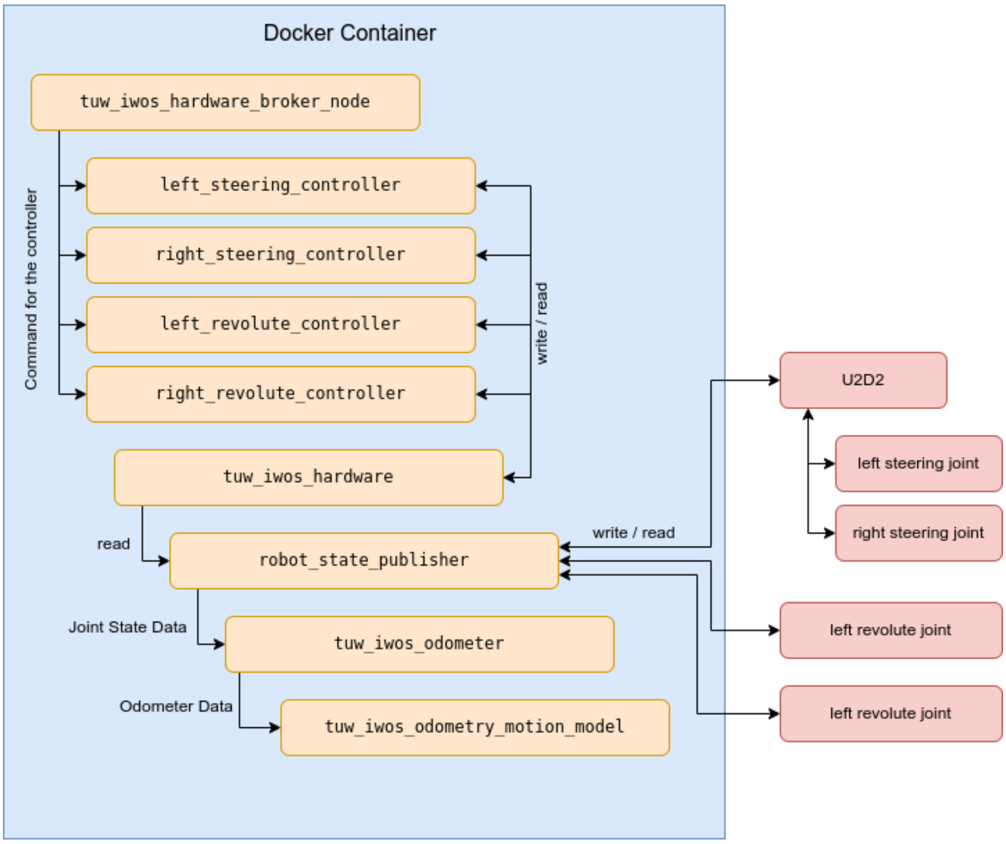


Figure 4.4: Overview of the prototype's software and software controlled hardware components. The blue box represents the Docker container embedding the ROS nodes, which are depicted by orange boxes. The red boxes depict hardware components. Arrows represent data flow.

for the robot's odometer functionality. The `tuw_iwos_motion_model` node consumes the data emitted by the odometer, introducing noise to compensate for errors in the odometric measurements.

In the upcoming chapter the calculation of odometry data is presented. Additionally an algorithm to calculate odometry data for the IWOS is introduced. This algorithm is the centerpiece of the aforementioned `tuw_iwos_odometer` node.

CHAPTER 5

Odometry

In mobile robotics odometry, also known as dead reckoning, is a fundamental method to determine the position and orientation of a system based on its locomotion. This process relies on a combination of diverse sources, incorporating either motion commands or sensor readings from a wide range of sensors, or even both. These inputs are then subject to further processing to achieve the desired outcomes. In the field of robotics, odometry plays a critical role in tasks such as localization and mapping, among others [BAM17]. As a result, it is essential for a robot to provide accurate odometry information.

In this chapter, we will delve into general aspects of odometry and present a odometry calculation method for the IWOS which finds application in the IWOS prototype.

There are various sensor-based approaches that can be employed for odometry calculation. A straightforward method involves using sensor readings from the robot's joints, which can provide information on position, velocity, effort, and sometimes acceleration. Another approach utilizes inertial measurement units (IMUs) to estimate acceleration and orientation, enabling odometry calculation. Additionally, visual odometry relies on visual data captured by cameras and depth sensors to calculate changes in position [YBHH15]. Novel approaches toward visual odometry utilize machine learning techniques to enhance the accuracy [VCBV23].

To enhance the accuracy of reported odometry data, it is possible to combine data from these or other sources. The `robot_localization`[1] package, part of the ROS framework, is specifically designed for this purpose. It utilizes extended Kalman filters[2] for data fusion, which results in improved odometry [MS14].

[1] For more details on the `robot_localization` package, visit: `https://docs.ros.org/en/noetic/api/robot_localization/html/index.html` (accessed on 09.08.2023).

[2] The extended Kalman filter is an estimation algorithm that combines nonlinear system dynamics with linearized measurements to estimate the state of a system in the presence of uncertainties.

5.1 Odometry Calculation

In general, odometry data in the form of a pose $\mathbf{x} = \begin{bmatrix} x, & y, & \theta \end{bmatrix}^T$ relative to a starting point can be calculated based on the linear velocity $v(t)$ and the angular velocity $\omega(t)$ of a system [KZBS17].

$$x(t) = \int_0^t v(t) \cdot \cos(\theta(t)) \cdot dt \tag{5.1}$$

$$y(t) = \int_0^t v(t) \cdot \sin(\theta(t)) \cdot dt \tag{5.2}$$

$$\theta(t) = \int_0^t \omega(t) \cdot dt \tag{5.3}$$

However, in practice, this calculation is not straightforward. The functions $v(t)$, $\omega(t)$, and $\theta(t)$ are not known at all times t. Nevertheless, with sensor readings, the values of these functions can be calculated for the time of the sensor reading. Assuming constant linear and angular velocities for a given time interval between sensor readings of duration Δt and a given set of $v(t)$ and $\omega(t)$, the odometry calculation simplifies to terms below [KZBS17].

$$x(t + \Delta t) = x(t) + v(t) \cdot \cos(\theta(t)) \cdot \Delta t \tag{5.4}$$

$$y(t + \Delta t) = y(t) + v(t) \cdot \sin(\theta(t)) \cdot \Delta t \tag{5.5}$$

$$\theta(t + \Delta t) = \theta(t) + \omega(t) \cdot \Delta t \tag{5.6}$$

Here, the motion parameters v and ω at time t are used for the calculation. It is worth noting that in other literature, v and ω might be considered at time $t + \Delta t$ for the calculation [TBF05].

For large time intervals, the calculation above may not be accurate since the angular velocity is only considered for the orientation and not the position. To improve accuracy, it is essential to incorporate the change in orientation into the position calculation, thereby accounting for both the angular and linear motion of the system. This refinement enhances the overall precision of the odometry calculation.

5.1.1 Odometry Calculation for the IWOS

For robots with a DD, knowing the velocity of the left and right wheel is sufficient to calculate odometry data. These robots are equipped with wheel encoders that measure the wheels' position and infer their velocity. This method is known as wheel encoder odometry.

However, the IWOS is capable of more complex motions than the DD. Therefore, wheel velocity alone is not sufficient to determine the robot's change in position. The position of the steering joints also needs to be considered. Consequently, odometry using joint position, velocity, or effort will be referred to as joint state odometry. Unlike the DD the facing and driving direction are not identical for the IWOS. This has to be considered when calculating the robots change in position.

To address the specific characteristics of the IWOS system, Algorithm 5.1 was developed. This algorithm calculates odometry data based on joint states, including the updated pose $\mathbf{x}\,(t + \Delta t)$ and the offset of the facing and driving direction $\kappa\,(t + \Delta t)$.

To perform the calculation, it is assumed that the robot's joint states contain a timestamp, which is used to determine the duration in seconds. The algorithm considers the change in the robot's orientation during this duration. To ensure accuracy, a maximum time interval for iterations is introduced, and the robot's orientation used in the calculations is updated at least every maximum time interval.

The motion parameters v, ω, and κ required for this algorithm, based on the joint state, are calculated using the method described in Section 3.3.3. These calculations enable the algorithm to return the robot's pose $\mathbf{x}_{t+1}$ and the offset between facing and driving direction κ_{t+1}.

Algorithm 5.1: Odometry Calculation for the IWOS.

Input: robot pose $\mathbf{x}_t$, robot joint states s_t and s_{t+1} as well as a minimum time interval for iterations Δt_{MIN}

Output: robot pose $\mathbf{x}_{t+1}$

$\Delta t = (s_{t+1})\,.timestamp - (s_t)\,.timestamp$

$\mathrm{i} = \mathrm{ceil}(\frac{\Delta t}{\Delta t_{MIN}})$

$\Delta t' = \frac{\Delta t}{\mathrm{i}}$

v_{t+1} = calculate v at s_{t+1}

ω_{t+1} = calculate ω at s_{t+1}

κ_{t+1} = calculate κ at s_{t+1}

$x = x$ component of $\mathbf{x}_t$

$y = y$ component of $\mathbf{x}_t$

$d = (\theta$ component of $\mathbf{x}_t) + \kappa_{t+1}$

for $i \leftarrow 1$ **to** *iterations* **do**

  $x = x + v_{t+1} \cdot \cos(d) \cdot \Delta t'$

  $y = y + v_{t+1} \cdot \sin(d) \cdot \Delta t'$

  $d = d + \omega_{t+1} \cdot \Delta t'$

end

$\mathbf{x}_{t+1} = (x, y, d - \kappa_{t+1})$

return $\mathbf{x}_{t+1}, \kappa_{t+1}$;

5.2 Limitations

Odometry relies on measurements that inherently contain errors, which can distort the accuracy of the results. These errors can be categorized as either systematic errors, such as biases, or random errors, such as noise.

When it comes to odometry relying on joint state data, it is particularly prone to errors and drift over time. Several sources of errors and drift can be identified for a joint state-based odometry system:

- **Sensors:** The sensors that measure the joint state, including position, velocity, and effort, play a crucial role in odometry. However, noise present in the sensor measurements can introduce drift in the odometry data.
- **Mechanics:** The mechanical aspects of the robot, such as wheel diameter and wheelbase, significantly impact the robot's motion. Inaccuracies in these mechanical parameters can lead to errors in odometry calculations and result in drift over time.
- **Motion Inaccuracies:** Inaccuracies in the robot's motion can stem from various sources. The type of surface the robot is moving on can influence its movements. Factors like slips and bumps can cause the joints to report data that does not accurately represent the robot's true motion. These motion inaccuracies introduce errors that accumulate over time and affect the estimation of odometry.

These are just a few examples of the causes of errors and drift in odometry, particularly in joint state-based odometry. It is essential to be aware of these sources of errors and take appropriate measures to mitigate their effects in order to improve the accuracy and reliability of odometry estimation. An approach to address these types of errors is introduced in the following chapter.

CHAPTER 6

Odometry Motion Model

Odometric data can be obtained from various sources, each with varying degrees of accuracy [MHW+19]. To improve the stability and reliability of pose estimation, it is often beneficial to fuse data from multiple sources [MS14].

However, achieving perfect odometry data can be challenging due to several factors. One major challenge is the presence of statistical errors in the sensor readings that form the basis of odometry calculations [TBF05]. To address these errors, Sebastian Thrun developed the OMM, which incorporates the errors by assuming a random distribution for the resulting errors [TBF05].

The OMM, takes into account the noise in odometry data in order to improve its quality. Despite being based on sensor measurements, odometry data is treated as control data. This simplification is made to reduce the dimensions of the state space [TBF05]. As a result, the odometry data $\bar{\mathbf{x}}_{t-1}$ and $\bar{\mathbf{x}}_t$ are treated as a command:

$$u_t = \begin{bmatrix} \bar{\mathbf{x}}_{t-1} \\ \bar{\mathbf{x}}_t \end{bmatrix} \tag{6.1}$$

where

$$\bar{\mathbf{x}}_{t-1} = \begin{bmatrix} \bar{x} & \bar{y} & \bar{\theta} \end{bmatrix}^T \quad \text{and} \quad \bar{\mathbf{x}}_t = \begin{bmatrix} \bar{x}' & \bar{y}' & \bar{\theta}' \end{bmatrix}^T . \tag{6.2}$$

This implies that odometry suggest the transition of the robot's pose $\bar{\mathbf{x}}_{t-1}$ to $\bar{\mathbf{x}}_t$ within the time interval $(t-1, t]$. Hence, we can employ u_t to characterize the change in the robot's pose during this time interval.

The OMM introduces an algorithm to express the probability distribution of the robot's pose $\mathbf{x}_t$ when executing a motion command u_t at state $\mathbf{x}_{t-1}$. The pose vectors $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are defined below.

$$\mathbf{x}_{t-1} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \quad \text{and} \quad \mathbf{x}_t = \begin{bmatrix} x' & y' & \theta' \end{bmatrix}^T . \tag{6.3}$$

Additionally the OMM proposes an algorithm to draw random samples from this probability distribution. The former algorithm is useful when interested in the most likely position of the robot, while the latter is especially useful for the applications including particle filters.

When calculating the most likely position of the robot, we utilize the previous position and the odometry data. We are interested in the likelihood of $\mathbf{x}_t$ given the conditions $\mathbf{x}_{t-1}$ and u_t. For this calculation Bayes' theorem is employed, which calculates the probability of an event A based on prior knowledge conditioned on event B that may influence event A. The general form of Bayes' theorem is [TBF05]:

$$P\left(A|B\right) = \frac{P\left(B|A\right) \cdot P\left(A\right)}{P\left(B\right)} \tag{6.4}$$

where:

- $P\left(A|B\right)$ is the probability of event A given that event B has occurred,
- $P\left(B|A\right)$ is the probability of event B given that event A has occurred,
- $P\left(A\right)$ is the a priori probability of event A,
- $P\left(B\right)$ is the a priori probability of event B.

The a priori probability is the probability of an event occurring based on general knowledge about that event. The a posteriori probability is the probability of event A occurring given the knowledge of event B that influences the outcome of event A.

The odometry motion model provides an algorithm to calculate the a posterior probability of a state $\mathbf{x}_t$ given the knowledge of $\mathbf{x}_{t-1}$ and u_t. Formally, this probability is described as:

$$p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t\right). \tag{6.5}$$

Additionally, the odometry motion model provides an algorithm drawing samples from this probability distribution.

6.1 Odometry Motion Model for the Differential Drive

The OMM was initially designed for robots with a DD. A fundamental principle of the OMM is that the transition of a robot utilizing the DD locomotion system from a pose $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ can be accomplished through three individual motion components:

- δ_{r1}: an initial rotation of the robot such that it can drive from the initial position to the final position,
- δ_t: a transition, in form of a linear motion, from the initial position to the final position,
- δ_{r2}: a final rotation, at the final position, such that the robots orientation is the desired final orientation.

Any motion with a DD locomotion system in the time interval $(t-1, t]$ can be approximated with these partial motions. Figure 6.1 depicts the motion approximation for the DD.
It is important to note that the actual motion of the robot may deviate from the estimated motion based on odometry data. Odometry data provides an approximation of the robot's motion, but it may not accurately reflect the true motion. In an ideal scenario, odometry data would be published at a high frequency, and the estimated motion would closely align with the actual motion of the robot.

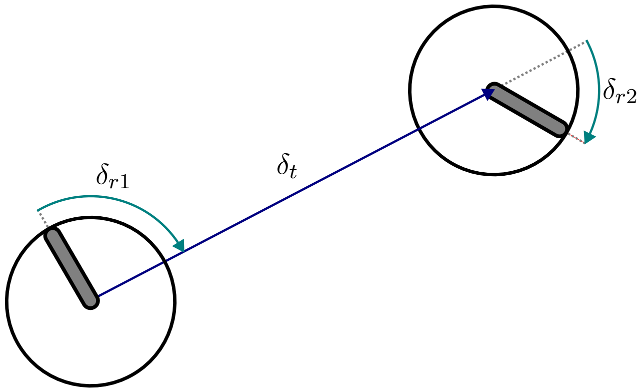


Figure 6.1: Motion approximation for a robot with DD locomotion system from a pose $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$, with an initial rotation δ_{r1}, a translation δ_t and a final rotation δ_{r2}. Rotational motions are depicted with teal arrows and translational motions are depicted with a blue arrow.

Algorithm 6.1, designed by Sebastian Thrun [TBF05], delineates the calculation of $p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t)$. This algorithm calculates the probability of an error for each of the partial motions extracted from the odometry data. For the calculation of the error probability an error distribution is presumed. Proposed error distributions are normal distribution or triangular distribution with a zero mean. The probability functions are denoted as $\mathbf{prob}(a, b^2)$, which provides the probability of value a in the either a normal or triangular distribution probability function with the variance b^2. The product of the three error probabilities is the resulting probability for the final pose $\mathbf{x}_t$.

For the usage with particle filters drawing a sample from the probability $p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t)$ might be of interest. In this case we need to take a sample from either the normal or triangular distribution. The function taking such sample is denoted as $\mathbf{sample}(\mathbf{b^2})$, again with a zero mean and a variance of b^2. The sample taken for each of the three partial motions is substarcted from the according partial motion from the odometry data. This step is meant to calculate the real motion of the robot by subtracting the noise from the odometry data. This approach is realized in Algorithm 6.2, designed by Sebastian Thrun [TBF05], which returns a random sample from the aforementioned probability distribution.

Algorithm 6.1: OMM for DD [TBF05].

Input: robot pose $\mathbf{x}_{t-1}$ and pose $\mathbf{x}_t$ as well as odometry data u_t
Output: probability $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t\right)$

$\delta_{\mathrm{r1}} = \mathbf{atan2}\left(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}\right) - \bar{\theta};$

$\delta_{\mathrm{t}} = \sqrt{\left(\bar{x}' - \bar{x}\right)^2 + \left(\bar{y}' - \bar{y}\right)^2};$

$\delta_{\mathrm{r2}} = \bar{\theta}' - \bar{\theta} - \delta_{\mathrm{r1}};$

$\hat{\delta}_{\mathrm{r1}} = \mathbf{atan2}\left(y' - y, x' - x\right) - \bar{\theta};$

$\hat{\delta}_{\mathrm{t}} = \sqrt{\left(x' - x\right)^2 + \left(y' - y\right)^2};$

$\hat{\delta}_{\mathrm{r2}} = \theta' - \theta - \hat{\delta}_{\mathrm{r1}};$

$p_1 = \mathbf{prob}\left(\delta_{\mathrm{r1}} - \hat{\delta}_{\mathrm{r1}}, \alpha_1 \hat{\delta}_{\mathrm{r1}}^2 + \alpha_2 \hat{\delta}_{\mathrm{t}}^2\right);$

$p_2 = \mathbf{prob}\left(\delta_{\mathrm{t}} - \hat{\delta}_{\mathrm{t}}, \alpha_3 \hat{\delta}_{\mathrm{t}}^2 + \alpha_4 \hat{\delta}_{\mathrm{r1}}^2 + \alpha_4 \hat{\delta}_{\mathrm{r2}}^2\right);$

$p_3 = \mathbf{prob}\left(\delta_{\mathrm{r2}} - \hat{\delta}_{\mathrm{r2}}, \alpha_1 \hat{\delta}_{\mathrm{r2}}^2 + \alpha_2 \hat{\delta}_{\mathrm{t}}^2\right);$

return $p_1 \cdot p_2 \cdot p_3$;

Algorithm 6.2: Sample OMM for DD [TBF05].

Input: robot pose $\mathbf{x}_{t-1}$ as well as odometry u_t
Output: a random sample for $\mathbf{x}_t$ from the probability distribution $p\left(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t\right)$.

$\delta_{\mathrm{r1}} = \mathbf{atan2}\left(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}\right) - \bar{\theta};$

$\delta_{\mathrm{t}} = \sqrt{\left(\bar{x}' - \bar{x}\right)^2 + \left(\bar{y}' - \bar{y}\right)^2};$

$\delta_{\mathrm{r2}} = \bar{\theta}' - \bar{\theta} - \delta_{\mathrm{r1}};$

$\hat{\delta}_{\mathrm{r1}} = \delta_{\mathrm{r1}} - \mathbf{sample}\left(\alpha_1 \delta_{\mathrm{r1}}^2 + \alpha_2 \delta_{\mathrm{t}}^2\right);$

$\hat{\delta}_{\mathrm{t}} = \delta_{\mathrm{t}} - \mathbf{sample}\left(\alpha_3 \delta_{\mathrm{t}}^2 + \alpha_4 \delta_{\mathrm{r1}}^2 + \alpha_4 \delta_{\mathrm{r2}}^2\right);$

$\hat{\delta}_{\mathrm{r2}} = \delta_{\mathrm{r2}} - \mathbf{sample}\left(\alpha_1 \delta_{\mathrm{r2}}^2 + \alpha_2 \delta_{\mathrm{t}}^2\right);$

$x' = x + \hat{\delta}_{\mathrm{t}} \cos\left(\theta + \hat{\delta}_{\mathrm{r1}}\right);$

$y' = y + \hat{\delta}_{\mathrm{t}} \sin\left(\theta + \hat{\delta}_{\mathrm{r1}}\right);$

$\theta' = \theta + \hat{\delta}_{\mathrm{r1}} + \hat{\delta}_{\mathrm{r2}};$

return $\mathbf{x}_t = \begin{pmatrix} x' & y' & \theta' \end{pmatrix}^T$;

These algorithms contain noise parameters α_1 to α_4. The noise parameters influence the variance of the **prob** and **sample** functions by weighting the noise applied to the partial motions. These parameters are tied to the individual robot's characteristics within a given environment.

6.2 Odometry Motion Model for the Independent Wheel Offset Steering

Based on the similarities of the DD locomotion system and the IWOS locomotion system an adaption of the OMM for the IWOS seems natural. As discussed in Section 3.1 and Section 3.4 the IWOS is capable of the same motions as the DD and even extends the set of motions. These additional motions have to be considered when adapting the OMM. This includes the crucial consideration of the offset between the facing and driving direction, denoted as κ, as it significantly impacts the robot's motion.

For those motions that the IWOS is capable of but the DD is not, the ICC is positioned in distinct locations that are beyond the reach of the DD locomotion system. This implies that for these motions the robot's facing direction and driving direction are not equal and thus $\kappa \neq 0$.

In order to adapt the OMM designed for the DD we have to identify the partial motions necessary for the robot to advance from pose $\mathbf{x}_{t-1}$ to pose $\mathbf{x}_t$. The IWOS is capable of all motions the DD is capable of, if the steering angle of the left and right steering joint is at its neutral position. Therefore, the IWOS robot first advances from its starting position with to $\kappa \neq 0$ to a position with $\kappa = 0$, it can then use the same partial motions as the DD, and finally move to a position with $\kappa \neq 0$ again.

It is important to ensure that facing and driving direction are equal in order to split the advancement from pose $\mathbf{x}_{t-1}$ to pose $\mathbf{x}_t$. If facing and driving direction are not equal there are two possible configurations. On the one hand the ICC could be at infinity and the steering angle for the left and right steering joint is not at its neutral position. In this case the robot is not capable of rotations without violation of the sliding constraints of the wheels. On the other hand the ICC could be at a finite location. In this case the robot has to make an arc like motion around the ICC, potentially leading the robot to another position than the desired final position.

The following partial motions for an robot with IWOS to move from pose $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ can be identified:

- δ_{o1}: an initial orientation motion for the robot to bring the steering joints into their neutral position,
- δ_{r1}: an initial rotation of the robot such that it can drive from the initial position to the final position where facing and driving direction are equal,
- δ_t: a transition from the initial position to the final position,
- δ_{r2}: a final rotation such that the robots orientation is the desired final orientation,
- δ_{o2}: a final orientation motion such that the robot's offset between the facing and driving direction aligns with the desired configuration, without altering the robot's facing direction.

Any motion with a IWOS locomotion system in the time interval $(t-1, t]$ can be approximated with these motions. Figure 6.2 depicts the motion approximation for the IWOS.

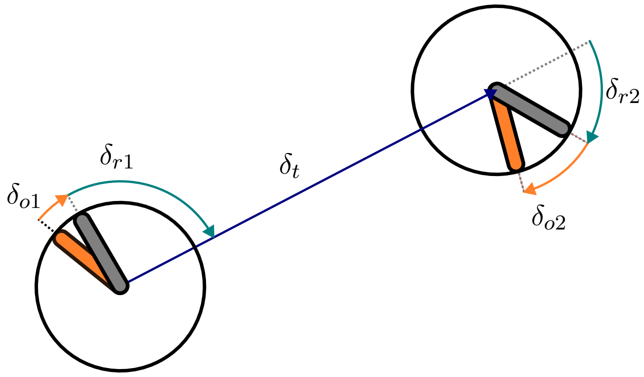


Figure 6.2: Motion approximation for a robot with IWOS locomotion system from a pose $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$, with an initial orientation motion δ_{r1}, an initial rotation δ_{r1}, a translation δ_t and a final rotation δ_{r2} as well as a final orientation motion δ_{r2}. Orientation motions are depicted with an orange arrow, rotational motions are depicted with teal arrows and translational motions are depicted with a blue arrow. The robots facing direction is depicted in grey and the robots driving direction is depicted in orange.

Note that the partial motions δ_{o1} and δ_{o2} might be complex motions. These motions might include movement in both, the steering joints and the revolute joints. Additionally, the robot may be required to change position and orientation for this partial motion. Thus, δ_{o1} and δ_{o2} are a simplification of potentially complex motions which are necessary to split the advancement from pose $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ into partial motions.

To address the additional partial motions the OMM for the IWOS has to introduce additional noise parameters, resulting in noise parameters α_1 to α_9. Like for the original OMM these noise parameters are specific to a robot and its environment.

The calculation of the a posterior probability with this novel OMM for the IWOS is broken down in Algorithm 6.3.

The sampling process for the IWOS on the probability function $p(\mathbf{x}_t|\mathbf{x}_{t-1}, u_t)$ is described in Algorithm 6.4.

6.3 Experimental Verification

To verify the concept of the OMM, a series of experiments are conducted using the IWOS prototype developed for this thesis. These experiments aim to compare the actual movement of the robot with the odometry data reported by the robot itself and the odometry data reported by the OMM.

Algorithm 6.3: OMM for IWOS, inspired by the original OMM by Thrun [TBF05].

Input: robot pose $\mathbf{x}_{t-1}$ and pose $\mathbf{x}_t$ as well as odometry information u_t
Output: probability $p\left(\mathbf{x}_t | u_t, \mathbf{x}_{t-1}\right)$

$\delta_{\mathrm{o1}} = -\bar{\kappa};$
$\delta_{\mathrm{r1}} = \textbf{atan2}\left(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}\right) - \bar{\theta};$
$\delta_{\mathrm{t}} \ = \sqrt{\left(\bar{x}' - \bar{x}\right)^2 + \left(\bar{y}' - \bar{y}\right)^2};$
$\delta_{\mathrm{r2}} = \bar{\theta}' - \bar{\theta} - \delta_{\mathrm{r1}};$
$\delta_{\mathrm{o2}} = \bar{\kappa}';$
$\hat{\delta}_{\mathrm{o1}} = -\kappa;$
$\hat{\delta}_{\mathrm{r1}} = \textbf{atan2}\left(y' - y, x' - x\right) - \bar{\theta};$
$\hat{\delta}_{\mathrm{t}} \ = \sqrt{\left(x' - x\right)^2 + \left(y' - y\right)^2};$
$\hat{\delta}_{\mathrm{r2}} = \theta' - \theta - \hat{\delta}_{\mathrm{r1}};$
$\hat{\delta}_{\mathrm{o2}} = \kappa';$
$p_1 = \textbf{prob}\left(\delta_{\mathrm{o1}} - \hat{\delta}_{\mathrm{o1}}, \alpha_1\hat{\delta}_{\mathrm{o1}}^2 + \alpha_2\hat{\delta}_{\mathrm{r1}}^2 + \alpha_3\hat{\delta}_{\mathrm{t}}^2\right);$
$p_2 = \textbf{prob}\left(\delta_{\mathrm{r1}} - \hat{\delta}_{\mathrm{r1}}, \alpha_4\hat{\delta}_{\mathrm{o1}}^2 + \alpha_5\hat{\delta}_{\mathrm{r1}}^2 + \alpha_6\hat{\delta}_{\mathrm{t}}^2\right);$
$p_3 = \textbf{prob}\left(\delta_{\mathrm{t}} \ - \hat{\delta}_{\mathrm{t}}, \alpha_7\hat{\delta}_{\mathrm{o1}}^2 + \alpha_7\hat{\delta}_{\mathrm{o2}}^2 + \alpha_8\hat{\delta}_{\mathrm{r1}}^2 + \alpha_8\hat{\delta}_{\mathrm{r2}}^2 + \alpha_9\hat{\delta}_{\mathrm{t}}^2\right);$
$p_4 = \textbf{prob}\left(\delta_{\mathrm{r2}} - \hat{\delta}_{\mathrm{r2}}, \alpha_4\hat{\delta}_{\mathrm{o2}}^2 + \alpha_5\hat{\delta}_{\mathrm{r2}}^2 + \alpha_6\hat{\delta}_{\mathrm{t}}^2\right);$
$p_5 = \textbf{prob}\left(\delta_{\mathrm{o2}} - \hat{\delta}_{\mathrm{o2}}, \alpha_1\hat{\delta}_{\mathrm{o2}}^2 + \alpha_2\hat{\delta}_{\mathrm{r2}}^2 + \alpha_3\hat{\delta}_{\mathrm{t}}^2\right);$
return $p_1 \cdot p_2 \cdot p_3 \cdot p_4 \cdot p_5;$

The goal of these experiments is to provide a proof of concept for the OMM in the context of the IWOS. Since the robot executing the experiments is a prototype, the results are interpreted qualitatively rather than quantitatively.

For accurate measurements of the robot's position during the movement, ground truth data is required. The OptiTrack[1] system, available at the Vienna University of Technology, is used to record this ground truth data. This system utilizes eight cameras with depth sensors, specifically the OptiTrack Prime 13[2] cameras, to triangulate the position of the robot. The prototype is equipped with a set of three markers, allowing precise quantification of the robot's position and orientation. The OptiTrack system, compatible with ROS, is able to publish the position and orientation of the robot at a frequency of up to 240Hz using the `vrpn_client_ros`[3] package.

[1] OptiTrack is a motion capture technology that uses specialized cameras and markers to precisely track and analyze the movements of objects or individuals in real-time.
[2] For more details on the OptiTrack Prime 13, visit: `https://optitrack.com/cameras/prime-13/` (accessed on 09.08.2023).
[3] For more details on the `vrpn_client_ros` package, visit: `https://wiki.ros.org/vrpn_client_ros` (accessed on 09.08.2023).

Algorithm 6.4: Sample OMM for IWOS, inspired by the original OMM by Thrun [TBF05].

Input: robot pose $\mathbf{x}_{t-1}$ as well as odometry information u_t

Output: a sample from the probability distribution $p\left(\mathbf{x}_t | u_t, \mathbf{x}_{t-1}\right)$

$\delta_{\mathrm{o}1} = -\bar{\kappa};$

$\delta_{\mathrm{r}1} = \mathbf{atan2}\left(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}\right) - \bar{\theta};$

$\delta_{\mathrm{t}} = \sqrt{\left(\bar{x}' - \bar{x}\right)^2 + \left(\bar{y}' - \bar{y}\right)^2};$

$\delta_{\mathrm{r}2} = \bar{\theta}' - \bar{\theta} - \delta_{\mathrm{r}1};$

$\delta_{\mathrm{o}2} = \bar{\kappa}';$

$\hat{\delta}_{\mathrm{o}1} = \delta_{\mathrm{o}1} - \mathbf{sample}\left(\alpha_1 \delta_{\mathrm{o}1}^2 + \alpha_2 \delta_{\mathrm{r}1}^2 + \alpha_3 \delta_{\mathrm{t}}^2\right);$

$\hat{\delta}_{\mathrm{r}1} = \delta_{\mathrm{r}1} - \mathbf{sample}\left(\alpha_4 \delta_{\mathrm{o}1}^2 + \alpha_5 \delta_{\mathrm{r}1}^2 + \alpha_6 \delta_{\mathrm{t}}^2\right);$

$\hat{\delta}_{\mathrm{t}} = \delta_{\mathrm{t}} - \mathbf{sample}\left(\alpha_7 \delta_{\mathrm{o}1}^2 + \alpha_7 \delta_{\mathrm{o}2}^2 + \alpha_8 \delta_{\mathrm{r}1}^2 + \alpha_8 \delta_{\mathrm{r}2}^2 + \alpha_9 \delta_{\mathrm{t}}^2\right);$

$\hat{\delta}_{\mathrm{r}2} = \delta_{\mathrm{r}2} - \mathbf{sample}\left(\alpha_4 \delta_{\mathrm{o}2}^2 + \alpha_5 \delta_{\mathrm{r}2}^2 + \alpha_6 \delta_{\mathrm{t}}^2\right);$

$\hat{\delta}_{\mathrm{o}2} = \delta_{\mathrm{o}2} - \mathbf{sample}\left(\alpha_1 \delta_{\mathrm{o}2}^2 + \alpha_2 \delta_{\mathrm{r}2}^2 + \alpha_3 \delta_{\mathrm{t}}^2\right);$

$x' = x + \hat{\delta}_{\mathrm{t}} \cos\left(\theta + \hat{\delta}_{\mathrm{r}1}\right);$

$y' = y + \hat{\delta}_{\mathrm{t}} \sin\left(\theta + \hat{\delta}_{\mathrm{r}1}\right);$

$\theta' = \theta + \hat{\delta}_{\mathrm{r}1} + \hat{\delta}_{\mathrm{r}2};$

$\kappa' = \kappa + \hat{\delta}_{\mathrm{o}1} + \hat{\delta}_{\mathrm{o}2};$

$\mathbf{x}_t = \begin{pmatrix} x' & y' & \theta' \end{pmatrix}^{\mathrm{T}};$

return $\mathbf{x}_t$ and κ';

To record data during the experiments, the powerful `rosbag` package is utilized. This package allows the recording of messages published by any ROS node, including joint states, odometry, data from the OMM, motion commands and ground truth pose from the OptiTrack system. Note that the OptiTrack system does not directly measure the offset between the facing and driving direction. Therefore, the driving direction is inferred from the change in position, which is a suitable solution for motions involving linear velocity but may lack accuracy for motions involving only rotational motion.

Six different types of experiments are defined. These experiments represent the various motions achievable by the IWOS. On the one hand, this includes three types of motion that both the DD and the IWOS are capable of (line, curve and rotation), denoted as DD-Line, DD-Curve, and DD-Rotation, respectively. One the other hand, this includes motions specific to the IWOS and not feasible with the DD, involving an offset between the facing and driving direction. These motions include a linear motion, a curved motion and a motion similar to a rotation on spot where the ICC is located between the wheels and the left and right wheel spin with same velocity but different direction. These experiments are denoted as IWOS-Line, IWOS-Curve, and IWOS-Rotation. It should be noted that these motions involve movement of the revolute joints while keeping the steering joints at a fixed position.

These aforementioned motions are discussed in detail in Section 3.4.

A total of eighty experiments were conducted to evaluate the OMM. Among these experiments, DD-Line, DD-Curve, IWOS-Line, and IWOS-Curve were each executed ten times, while DD-Rotation and IWOS-Rotation were executed twenty times. The latter two experiments yielded fewer usable samples compared to the former, leading to the decision to conduct them twice as frequently. This results in a total of eighty conducted experiments.

Each executed experiment is composed of three distinct stages. Firstly, the robot was positioned upright with the steering joints set to the desired position for the specific experiment. Secondly, the robot received the motion command for the revolute joints. Lastly, the robot received the command to stop. Data recording began when the robot was stationary before receiving the first motion command and ended when the robot was stationary again after receiving the last motion command.

The motion commands for the six types of experiments are listed in Table 6.1. The stop command involved setting the target velocity for the revolute joints to zero while keeping the target position of the steering joints unchanged. To ensure comparability between executions, the aim was to keep the applied velocity within a small range, with the target values as close to $0.5\frac{m}{s}$ as possible.

Motion Type	α_l (rad)	α_r (rad)	v_l $(\frac{m}{s})$	v_r $(\frac{m}{s})$	κ (rad)
DD-Line	0.00	0.00	0.50	0.50	0.00
DD-Curve	0.00	0.00	0.25	0.75	0.00
DD-Rotation	0.00	0.00	0.40	−0.40	0.00
IWOS-Line	0.40	0.40	0.50	0.50	0.40
IWOS-Curve	0.40	0.32	0.40	0.50	0.39
IWOS-Rotation	−0.40	0.40	0.40	−0.40	−1.57

Table 6.1: Motion commands for the steering joints, and the revolute joints as well as the resulting κ for the experiments conducted.

The recordings of the eighty experiment executions were subsequently subjected to further processing. Firstly, data sets with a duration of one second have been extracted from these recordings. These data sets are allowed to overlap, resulting in a higher count of data sets, with the drawback of partially redundant data in these data sets. For each data set the odometry and data from the odometry motion model where provided with the ground truth data of the robot at the start of the data set. Consequentially the data sets contain the ground truth data, the odometry and the data from the odometry motion model for one second.

Secondly, the data set had to be filtered. Based on the execution of these experiments the data sets can depict the robot standing still, starting, driving and stopping. For this reason, only those data sets representing motions were filtered:

- **Line motions:** The DD-Line and IWOS - Line experiments were filtered for those data sets with a ground truth distance of at least $0.3m$ and at most $0.7m$. Additionally, only those data sets where the target velocity for the left and right wheel is not equal to zero were selected.
- **Curve motions:** The DD-Curve and IWOS-Curve experiments were filtered for those data sets with a ground truth distance of at least $0.2m$ and at most $0.6m$. Additionally, only those data sets where the target velocity for the left and right wheel is not equal to zero were selected.
- **Rotational motions:** For the DD-Curve and IWOS-Curve experiments, only those data sets where the target velocity for the left and right wheel is not equal to zero were selected.

Finally, the data sets for each kind of experiment were combined and plotted. For the plot a filter has been applied, limiting the plotted robot positions to $0 \leq x \leq 0.8$ and $-0.4 \leq y \leq 0.4$ for the DD-Line, DD-Curve, IWOS-Line, and IWOS-Curve experiments. For the DD-Rotation and IWOS-Rotation experiments the plot was limited to $-0.4 \leq x \leq 0.4$ and $-0.4 \leq y \leq 0.4$. This step was carried out to enhance the graphics and facilitate the comparability of the visuals. This filtering affected the portion of the graphics representing the OMM data, as these positions, due to the added noise, could potentially lie outside the aforementioned boundaries.

The results of the conducted experiments are presented and discussed in the following chapter.

CHAPTER 7

Results

In this chapter, we examine the results obtained from the conducted experiments. The experimental setup and execution are elaborately described in Section 6.3. Firstly, we will present the results in a comprehensive manner. Secondly, we will engage in a discussion of the results. Lastly, we will address the limitations of the experiments and explore the deductions drawn from them.

7.1 Result Presentation

To demonstrate the functionality of the IWOS prototype, as well as the proposed odometry approach and the sampling algorithm of the OMM, a series of experiments were conducted.

Six different types of experiments were designed, with ten to twenty executions carried out for each experiment type. In this chapter, we provide a qualitative analysis of the experimental results. The experimental results are presented separately for each of the six experiment types, but the data sets from each kind of experiment are fused together.

For each kind of experiment a figure depicts the executed motion. Additionally, up to three types of figures are provided showing the position data, orientation data and offset between driving and facing direction from the experiments. Each of these figures consists of three sub-figures. These subfigures show ground truth data on the left-hand side, odometry data in the center and samples from the OMM on the right-hand side.

- **Position figures:** These depict the final positions of all data sets for the ground truth data, the odometry data and the sampling algorithm of the OMM. The figures are displayed in robot coordinates, with the x-axis vertical and the y-axis horizontal. They are centered on the first position of the sample set. In the case of perfect odometry and sampling from the OMM, these illustrations would resemble the ground truth. However, as expected, we will observe deviations from the ground truth.

- **Orientation figures:** These depict the final orientations of the robot, which can range from $-\pi$ to π. The final orientations of the robot from all data sets are binned and plotted over the aforementioned range.
- **Offset figures:** These depict the offset between the facing and driving directions of the final position. This offset can range from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. Similarly, this data is binned across all data sets and plotted over this range.

Furthermore, a video on the platform YouTube shows an example for each kind of motion. Figure 7.1 shows a QR-code which can be used to access the video. Moreover, find the link to access the video in the figures caption. The author is committed to ensuring that the video remains accessible via the provided link and QR-code for as long as possible. However, it's important to note that the author cannot guarantee the perpetual availability of the video through these means.

Figure 7.1: QR-code to access the video showcasing the experimental motions. Alternatively, access the video via the link: `https://qrcc.me/ryckxcytt5vg` (accessed on 09.08.2023).

For all experiment executions the same noise parameters for the OMM have been used. These parameters are shown in Table 7.1.

α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9
0.05	0.2	0.1	0.05	0.2	0.1	0.05	0.025	0.5

Table 7.1: Noise parameters for the OMM sampling algorithm used for the experiments.

The noise parameters shown in Table 7.1 have been carefully selected through a grid search process. The objective of this grid search was to minimize the mean square error between the samples generated by the OMM and the final ground truth positions of the data sets.
Given the large number of noise parameters and the resulting search space, the grid search was performed using a relatively coarse grid with larger step sizes. This approach

helped to explore the parameter space more efficiently.
After the grid search, the noise parameters obtained were further adjusted based on the experience gained from processing the experimental data. This iterative process allowed for fine-tuning the parameters to improve the alignment between the OMM samples and the actual ground truth positions.

7.1.1 DD-Line Experiment

This experiment focuses on evaluating straight line motion. Both, the DD as well as the IWOS are capable of this motion. For a robot with the IWOS drive system this implies that both steering joints are in their neutral position. The motion is depicted in Figure 7.2.

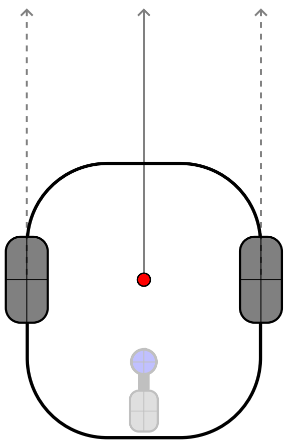


Figure 7.2: Motion for the DD-Line experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion.

The position results of this experiment are presented in Figure 7.3. The ground truth data exhibits a range of final positions, ranging from $x > 0.3m$ to $x < 0.6m$ for most data points, with some outliers reaching up to $x < 0.7m$. The range of positions in the driving direction for the odometry data is approximately $0.2m$ to $0.5m$, with some outliers reaching up to $0.6m$. Nevertheless, the odometry data exhibits a noticeable lag, trailing approximately seventeen percent behind the ground truth measurements. However, this lack of accuracy in the odometry data is partially compensated for in the sampling algorithm of the OMM. The shape of the samples generated by the OMM is similar to the results presented by Thrun [TBF05] and fully covers the ground truth and odometry data.

The orientation data for this experiment is displayed in Figure 7.4. The ground truth data shows that the orientation of the robot is aligned with the coordinate system, as expected. However, a small spread in the data suggests some inaccuracies in the execution

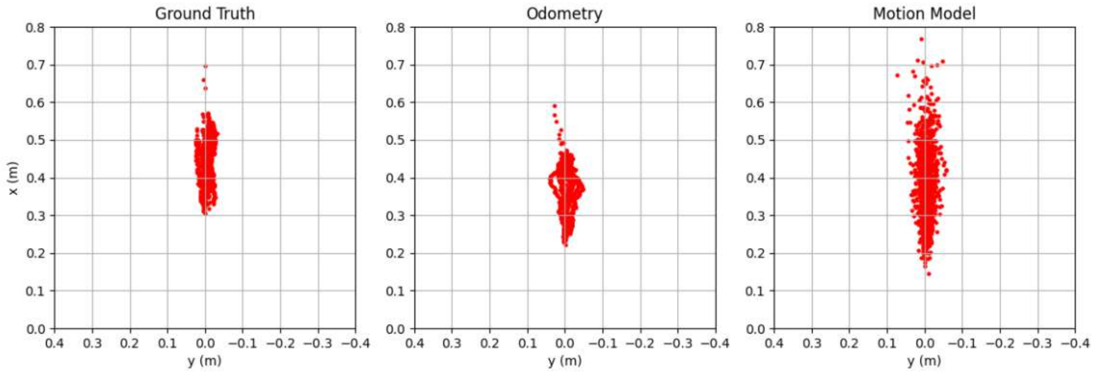


Figure 7.3: Position data of the DD-Line experiments.

of the motion. The odometry data is in consonance with the ground truth data. Also the samples from the OMM show the same spread as the ground truth and odometry data.

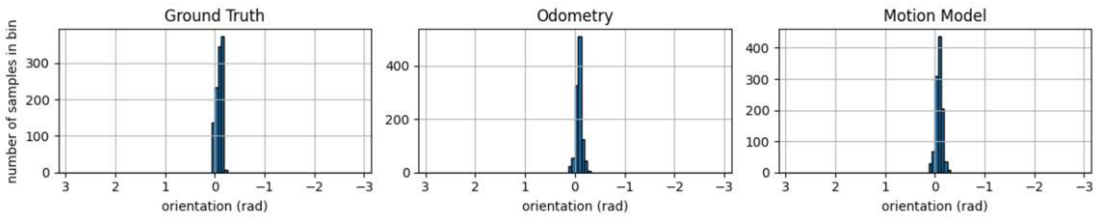


Figure 7.4: Orientation data of the DD-Line experiments.

For this experiment, the offset between the facing and driving direction is zero. This means that the robot is moving in a straight line without any deviation or rotation. This characteristic is also reflected in the resulting data, as there is no noticeable offset observed between the facing and driving direction in the position data. The robot maintains a consistent alignment between its direction of movement and its orientation throughout the experiment.

7.1.2 DD-Curve Experiment

The objective of this experiment is to examine the curve motion the DD and IWOS are capable of. In order for a robot with the IWOS drive system to execute this motion both steering joints have to be in their neutral position. This motion is depicted in Figure 7.5.

The position results of this experiment are presented in Figure 7.6. Please note that only the endpoints of the data sets are depicted in the figure. The final positions of the data are centered around $0.3m$ in the x-direction and from zero to $0.1m$ in the y-direction. Similar to previous observations, the odometry data lags behind the ground truth positions by about twenty-six percent. The data from the OMM represents the

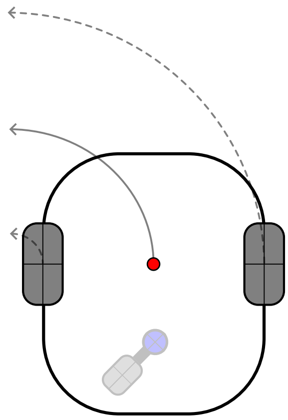


Figure 7.5: Motion for the DD-Curve experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion.

odometry data quite accurately but does not fully compensate for the errors introduced by the odometry.

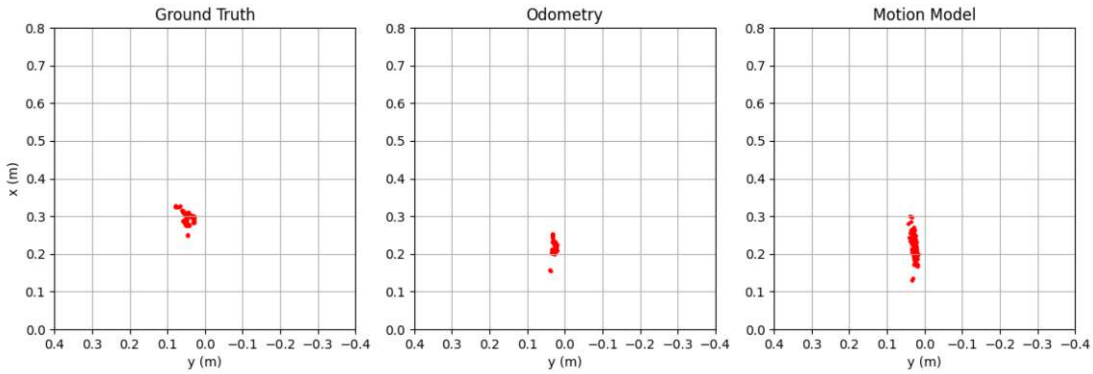


Figure 7.6: Position data of the DD-Curve experiments.

Figure 7.7 presents the final orientation of the robot for this experiment. The coordinate system used in the figure is aligned with the robot's position at the start of each data set. As expected for a curved motion, the ground truth data indicates a counter-clockwise rotation. These results align with the original OMM design by Thrun [TBF05]. The data from the odometry and the OMM exhibit a similar spike in orientation but lack in distribution of the final orientations.

In this experiment, the offset between the facing and driving direction is zero. This means that the robot maintains a consistent alignment between its direction of movement and its orientation throughout the motion. This characteristic is reflected in the data,

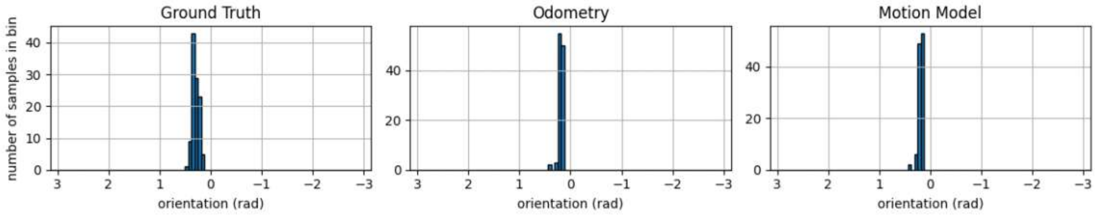


Figure 7.7: Orientation figure for the DD-Curve experiments.

including the ground truth, odometry, and the data generated by the OMM. All three sources of data indicate that there is no offset between the facing and driving direction.

7.1.3 DD-Rotation Experiment

This experiment focuses on the rotational motion of the robot, which can be executed by both the DD and IWOS. For a robot with the IWOS drive system this implies that both steering joints are in their neutral position. The robot's position is steady but the robot's orientation changes during this motion. Such a motion is depicted in Figure 7.8.

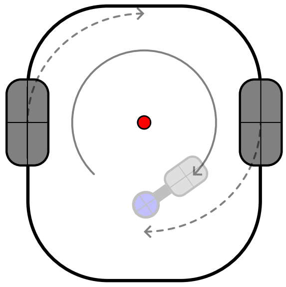


Figure 7.8: Motion for the DD-Rotation experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the change in the robots orientation. The caster wheel is depicted according to the motion.

Figure 7.9 displays the final positions of the robot for this type of experiment. It is observed that the rotational motion is not executed perfectly, leading to a change in the position of the robot's center. The odometry data captures this change in position. The samples drawn from the OMM show a wide spread centered to the positions of the ground truth and odometry data. Overall, the positional data suggests that the robot's rotational motion lacks complete precision, exhibiting a certain degree of inaccuracy in execution. Although the odometry data partially captures these discrepancies, it fails to fully depict the extent of the inaccuracy in the motion. Notably, the OMM displays a pronounced overshooting effect, with samples showing a significantly wider spread compared to both the ground truth and odometry data.

Figure 7.10 presents the final orientation of the robot from the rotational motion experiment. The ground truth data aligns with our expectations, displaying a noticeable shift

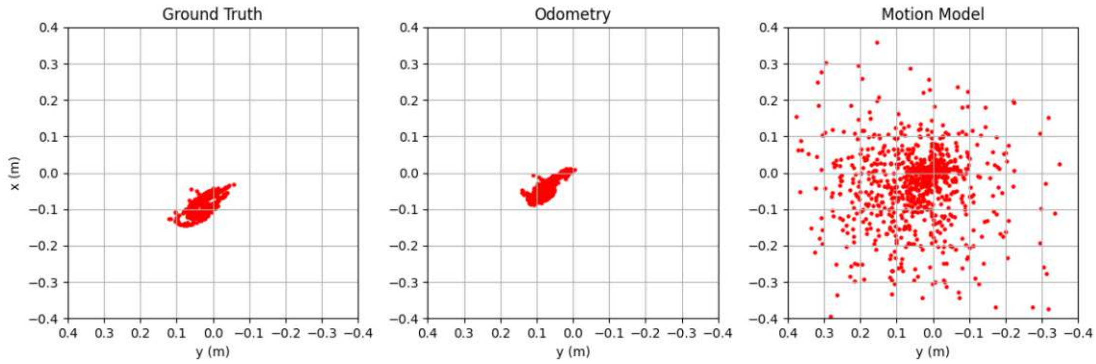


Figure 7.9: Position figure for the DD-Rotation experiments.

in orientation towards a negative angle, which is consistent with the rotational motion. Comparatively, the odometry data exhibits a similar spread in the final orientation, but it underestimates the mean final orientation. Meanwhile, the samples obtained from the OMM showcase a wide spread encompassing the means of both ground truth and odometry data. However, once again, we observe an excess in the spread of the OMM data samples, indicating a potential discrepancy in its precision.

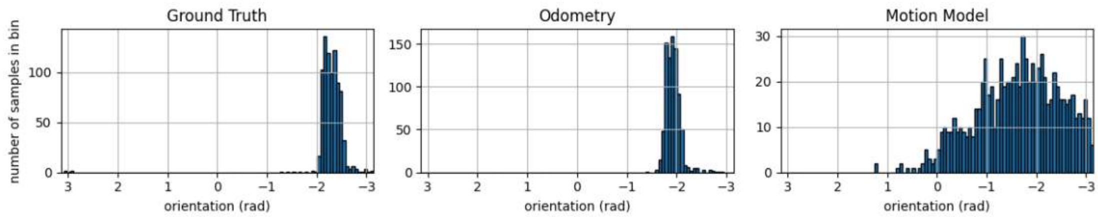


Figure 7.10: Orientation figure for the DD-Rotation experiments.

In this experiment, trustworthy ground truth data for the offset between the facing and driving direction could not be obtained, due to he limitations of the ground truth capturing system. Therefore, it is not possible to provide accurate information regarding the offset in this particular experiment.

7.1.4 IWOS-Line Experiment

This experiment evaluates on a motion unique to the IWOS locomotion system. The robot drives in a straight line with a maximum offset between the facing and driving direction. Figure 7.11 showcases the robot's motion for this kind experiment.

In Figure 7.12, the coordinate system is aligned with the robot's driving direction to provide intuitive data visualization. The position data from this experiment shows results similar to the position data from the DD-Line experiment. The ground truth data ranges

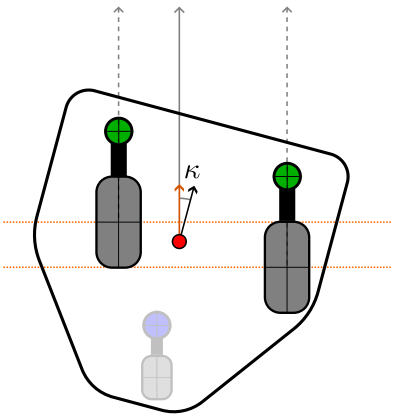


Figure 7.11: Motion for the IWOS-Line experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion. The orange arrow indicates the driving direction and the black arrow indicates the facing direction.

from $0.3m$ to approximately $0.5m$ and the odometry data ranges from $0.2m$ to almost $0.5m$. The odometry data lags about eighteen percent behind the ground truth data. The shape of the samples generated by the OMM is similar to the results presented by Thrun [TBF05] and fully covers the ground truth and odometry data.

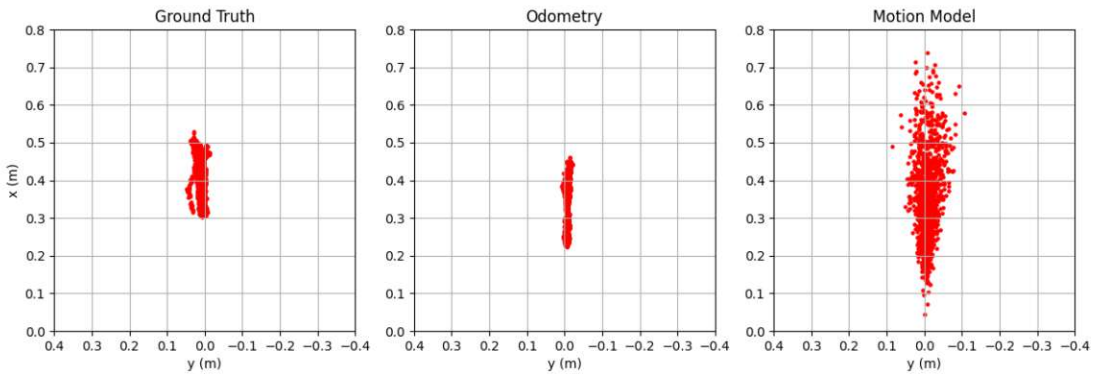


Figure 7.12: Position figure for the DD-Rotation experiments.

Figure 7.13 illustrates the orientation of the final positions in the data sets for the IWOS-Line experiment. The ground truth data indicates an orientation with a negative angle close to half a radian. This aligns with the expected behavior, as the coordinate system is aligned with the driving direction. With a maximum offset between the facing and driving direction, a negative orientation of exactly that value is expected. Inaccuracies in motion execution lead to a spread of the ground truth data around the expected final orientation. The odometry data, however, falls short in accurately representing the spread of variance in the orientation. It fails to capture the full range of possible orientations. The sampling from the OMM compensates for this omission and provides a better representation of the orientation variance observed in the ground truth data. The OMM sampling compensates for the limitations of the odometry data in terms of capturing the orientation spread.

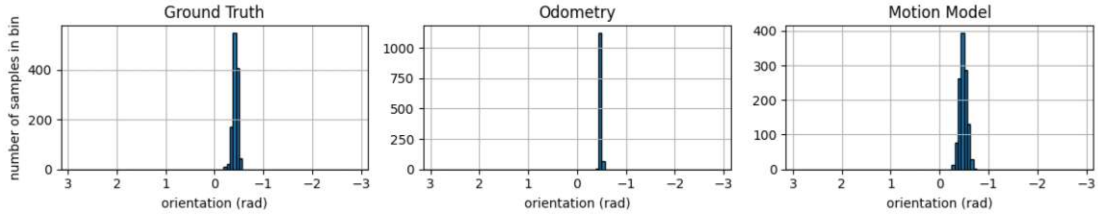


Figure 7.13: Orientation figure for the IWOS-Line experiments.

The offset between facing and driving direction is depicted in Figure 7.18. The ground truth data represents the stated offset of approximately half a radian. The odometry data shows results similar to the ground truth data with neglectable differences. This aligns reasonably well with the expected offset, indicating that the odometry captures the offset between facing and driving direction as desired. As anticipated, the OMM introduces additional noise to the offset between facing and driving direction. This is expected since the OMM sampling algorithm incorporates stochastic elements, resulting in a wider spread of offset values compared to the odometry data.

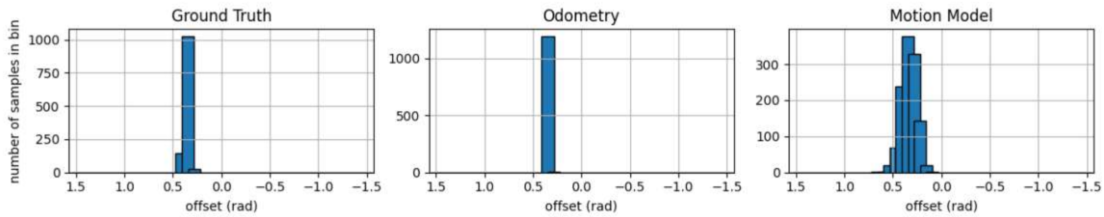


Figure 7.14: Offset figure for the IWOS-Line experiments.

7.1.5 IWOS-Curve Experiment

In this experiment, the curved motion of the IWOS locomotion system is examined. The steering joints are positioned in such a way that the ICC is located to the left-hand side of the robot. This configuration creates a curved motion with a radius of approximately two meters. In Figure 7.15 such a motion is depicted. Note that the ICC is depicted closer to the robot than it actually is in the experiment for visualization purposes.

Figure 7.16 displays the final positions of the data sets for the IWOS-Curve experiment. To provide an intuitive presentation, the coordinate system is aligned with the robot's facing direction. The ground truth data for this motion aligns with the expected behavior.

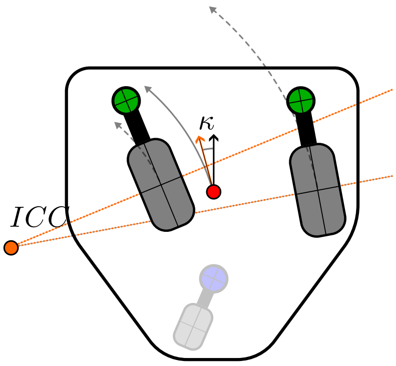


Figure 7.15: Motion for the IWOS-Line experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion. The orange arrow indicates the driving direction and the black arrow indicates the facing direction.

However, the odometry data falls behind the ground truth for about $0.1m$, failing to accurately depict the true motion of the robot. The odometry data lags about twenty-three percent behind the ground truth data. This observation aligns with the results of the other experiments. The samples from the OMM reflect the odometry data with added noise and cover the odometry data fully but not the ground truth data. These samples exhibit a wider spread compared to the DD-Curve experiment, indicating greater variability in the estimated positions.

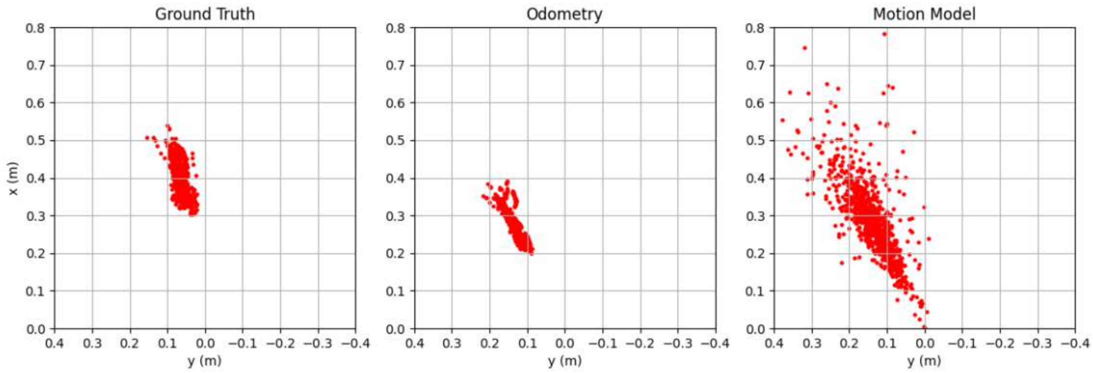


Figure 7.16: Position figures for the IWOS-Curve experiments.

The orientation of the robot at the end point of the data sets is shown in Figure 7.17. As anticipated, the final orientation displays a slight offset of approximately one-third of a radian within the positive range. The odometry data shows results for the final orientation similar to the ground truth data. In comparison, the samples drawn from the OMM also exhibit a spike akin to the ground truth and odometry data, yet with a broader spread, offering a more comprehensive depiction of the potential orientations at the endpoint.

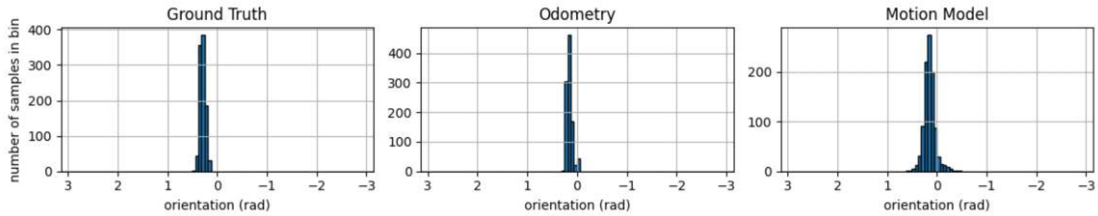


Figure 7.17: Orientation figures for the IWOS-Curve experiments.

In Figure 7.18 the offset between facing and driving direction is depicted. The ground truth data shows the expected offset as specified for this experiment. Odometry data exhibits results that are similar to the ground truth data, indicating a reasonable estimation of the offset between facing and driving direction. The samples generated by the OMM exhibit a wider spread in the offset values compared to both the ground truth and odometry data. This wider spread indicates the additional variability introduced by the OMM sampling algorithm.

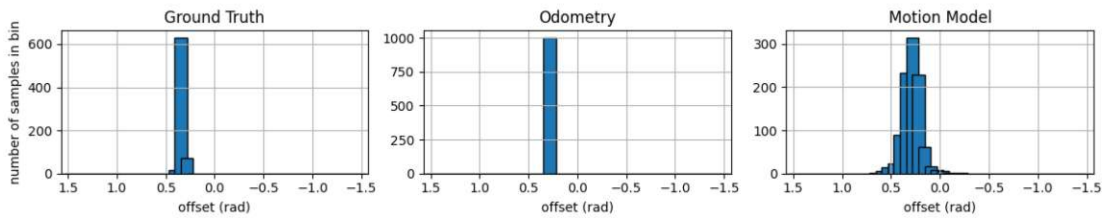


Figure 7.18: Offset figures for the IWOS-Line experiments.

7.1.6 IWOS-Rotation Experiment

This experiments focuses on examination of a motion similar to the DD-Rotation. For this experiments the wheels rotate with same velocity but different direction. Also the steering angle is identical for both steering joints in magnitude but with different direction. Note that for this experiment the robots center changes position, nevertheless, this motion is similar to the rotational motion of the DD. Figure 7.19 showcases such a motion.

The position data of the IWOS rotation experiments is shown in Figure 7.20. The ground truth data shows a small but noticeable change in the position of the robot, which is expected given that the ICC is not located at the robot's center for this motion. The odometry data, however, falls short in accurately reproducing this change in the robot's position. Like for the DD-Rotation experiment the samples from the OMM show a excessive spread.

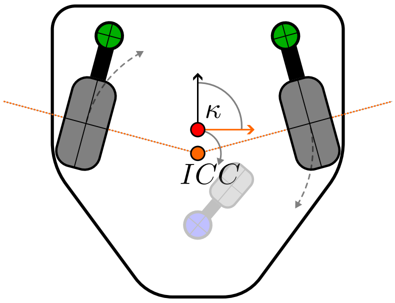


Figure 7.19: Motion for the IWOS-Line experiment. The dotted grey line depicts the trajectory of the wheels and the solid grey line depicts the trajectory of the robot's center. The caster wheel is depicted according to the motion. The orange arrow indicates the driving direction and the black arrow indicates the facing direction.

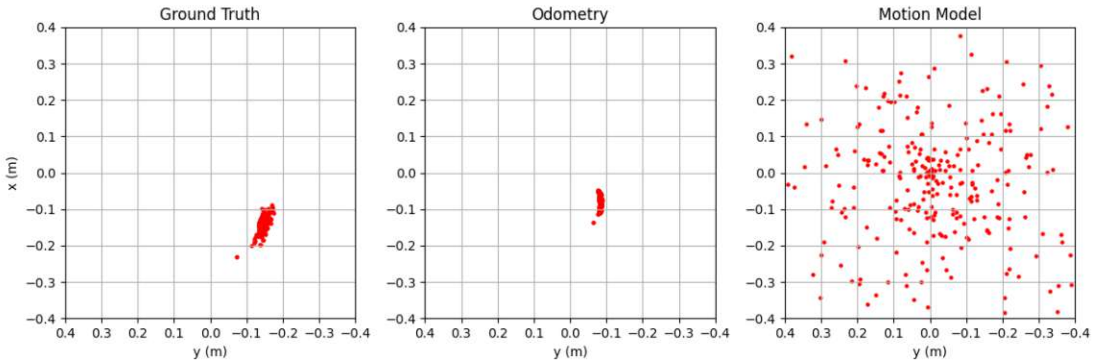


Figure 7.20: Position figures for the IWOS-Line experiments.

In Figure 7.21 the robot's orientation at the final position of the data sets is shown. The ground truth data shows a small spread in final orientations with a mean of negative two radians. Odometry data shows a similar spread with the mean falling short behind the ground truth data. The samples drawn from the OMM show a wider spread than both the ground truth and odometry data. The mean of the ground truth and odometry data both fall in the spread of the OMM samples. These observations are consistent with the results of DD-Rotation experiment.

Due to the unavailability of reliable ground truth data for the offset between the facing and driving direction in this kind of motion, accurate information regarding this offset cannot be provided for this specific experiment.

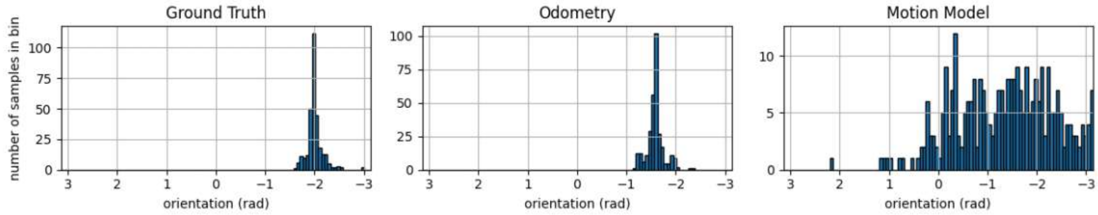


Figure 7.21: Orientation figures for the IWOS-Rotation experiments.

7.2 Result Summary and Discussion

In the previous section, the results of the conducted experiments, representing six different kinds of motions, are presented. These motions were identified in the discussion of the maneuverability of the IWOS locomotion system, aiming to address Research Question 2, see Section 3.4. Three of these motions are common to both the DD and IWOS, while the other three are exclusive to IWOS. For each motion, an experimental setup was created, consisting of ten to twenty executions, and their data is combined.

The data suggests that the execution of motion commands in IWOS is inaccurate, with the robot moving slower than desired. This is acceptable considering the prototype state of the IWOS robot used in these experiments. For the purposes of this thesis, this difference in motion execution does not significantly impact the comparison of ground truth, odometry, and OMM sampling data.

The experimental results indicate that the odometry approach for IWOS is able to represent the motion of the robot based on joint commands. However, the data shows that the odometry data falls behind the ground truth data for both linear and curved motion experiments. This lag ranges from from seventeen to twenty-six percent. This discrepancy may be caused by factors such as inaccurate sensor readings or imprecise measurements of the robot's characteristic such as wheel diameter, wheelbase or wheel offset. Thus, further work is needed to improve the quality of the reported odometry data. The odometry calculation proposed in this thesis, aiming to answer Research Question 2, complies the aim of representing the robots motion but lacks in accuracy. Further work targeting the quality of the reported odometry data is pending.

For linear and curved motions that both DD and IWOS are capable of, the OMM sampling algorithm yields results similar to the original OMM by Thrun. The results of the OMM for the linear motion are much closer to the results provided by Thrun [TBF05], while the results for curved motions lack in spread. This suggests that the adaptions made to the algorithm do not compromise its accuracy for linear and curved motions but limit the spread for samples. The experiments involving linear and curved motions with an offset between facing and driving direction demonstrate promising results for the OMM sampling algorithm for the IWOS. Notably, the algorithm introduces significant noise, even with relatively small values for the noise parameters affecting the offset between facing and driving direction and the partial motion δ_{o1} and δ_{o2}.

The experiments targeting rotational motions exhibit comparable data between the ground truth and odometry data. Again, the odometry data shows a slight lag behind the ground truth, yet still captures the essence of the motion. As mentioned earlier, this discrepancy could arise from faulty sensor readings or inaccurate measurements of the robot's characteristics. The OMM results for these motions display an excessive spread, which could be mitigated by employing a higher number of samples to better represent the true motion of the robot. It's worth noting that the original OMM for the DD does not provide data specifically for rotational motions. Consequently, these results cannot be directly compared to the original OMM by Thrun [TBF05].

In general, the samples from the OMM adequately cover the ground truth positions and align well with the odometry data. Notably, this coverage is particularly robust for line and curve motions, but it exhibits a lower level of accuracy for rotational motions. It remains uncertain whether this discrepancy is inherent in the original OMM or was introduced during the modifications made to adapt it to the IWOS. Addressing this potential flaw may be achievable by operating with a higher number of samples, offering a promising avenue for resolution.

An interesting and noteworthy insight, not revealed in the results, emerged due to a software bug. While conducting the IWOS-curve experiment, an inadvertent swapping of wheel velocities occurred, resulting in an invalid configuration. Specifically, the wheel velocity intended for a left curve was assigned to a right curve in terms of the steering angle, despite both curves having the same absolute value of the intended motion radius. Consequently, the robot unexpectedly drove in a slight right curve during this erroneous trial. This observation suggests that the dominance of the ICC by the steering position surpasses that of the wheel velocity. These findings possess potential significance for future investigations into maneuverability and the degrees of freedom, which are elaborated in Section 3.4 and Section 3.5.

In conclusion, this research has made a promising step towards adapting the OMM by Thrun [TBF05] to suit the IWOS locomotion system, addressing Research Question 3. The qualitative evaluation of the results highlights the effectiveness of the proposed approach in handling straight and curved motion. However, it also reveals that rotational motions are not as precise, underscoring the need for a higher number of samples to improve accuracy in those scenarios.

Overall, the odometry approach has the potential to benefit a system that fuses data from multiple sources, particularly if the reported short distances are resolved. The OMM sampling algorithm, which utilizes odometry data to compensate for noise, shows promising results in representing the odometry data with added noise.

7.3 Limitations

The conducted experiments in this thesis serve as a proof of concept for several elements, the prototype of the IWOS, the joint state odometry calculation for the IWOS as well as the OMM for the IWOS. However, there are limitations and areas for further improvement that should be acknowledged.

Firstly, the robot used in the experiments is the initial prototype of its kind. As such, it requires further hardware and software refinements to enhance the accuracy of motion execution and increase the frequency of joint state reports.

Secondly, the odometry approach employed in this thesis is derived from existing approaches known to suffer from drift over time. While a qualitative discussion suggests that the odometry falls behind the actual motion. This discrepancy could be attributed to factors such as inaccurate sensor readings from the joints or imprecise measurements of the robot's characteristics, including wheel diameter, wheelbase, or wheel offset, which are crucial for the odometry calculations. Moreover, the odometry estimation could benefit from higher joint state report frequencies.

Thirdly, no quantitative research has been conducted on the OMM for the IWOS locomotion system. Although the qualitative discussion of the experimental results indicates promising results for line and curve motion experiments, the experiments targeting rotational motions show less accurate results in comparison.

Finally, the conducted experiments contain data sets that include the starting and stopping processes of the robot. This flaw arises due to the limitations of the experimental setup, which restricts the distance traveled. Future experiments should aim to cover larger distances to eliminate the influence of the starting and stopping processes on the data sets.

In conclusion, while the conducted experiments have laid a solid foundation and demonstrated the potential of the IWOS prototype, the odometry approach, and the OMM sampling algorithm for the OMM, further improvements and investigations are necessary to address the limitations outlined above and unlock the full capabilities of the IWOS locomotion system.

CHAPTER 8

Conclusion

This thesis marks the debut of the innovative robot locomotion system, IWOS, which is a derivative of the well-established DD system. The DD system comprises two independent steerable wheels and finds widespread use in various applications such as warehousing, production, and transportation in crowded environments.

The IWOS was conceived to enhance the already-successful DD system, particularly in terms of HRI, collision detection, and collision handling. To achieve this, additional steered joints were integrated into the IWOS, allowing the steered wheels to pivot around a vertical axis.

Path indication has proven to be a challenging task for robots utilizing the DD locomotion system. However, the additional joints in the IWOS enable robots utilizing this locomotion system to face directions independent of their driving direction. This capability has the potential to enhance HRI, specifically in terms of path indication. While omnidirectional locomotion systems allow robots to face any direction while driving, they come with increased complexity in hardware and software. The IWOS aims to fill this gap by providing a relatively simple robot locomotion system capable of executing complex motions.

The offset between the driving and facing directions in the IWOS can be leveraged, enabling the robot to indicate its intended path by aligning its gaze with upcoming changes in motion direction. This approach, inspired by human gazing behavior, facilitates intuitive navigation intention communication.

Moreover, the additional joint, if equipped with hardware capable of measuring applied forces, can be utilized for collision detection by sensing external forces applied thereto. It should be noted that collisions directly to the front or back of the robot are exceptions to this detection method. However, collisions other than that type can not only be measured with this system but also be potentially resolved as well. Given the additional joints have not been steered to their limits, they can swivel the robot away from the point of contact.

This thesis delved into the intricacies of the IWOS locomotion system, providing a comprehensive understanding of its underlying principles. The fundamentals of the IWOS were discussed in detail, drawing upon foundational knowledge derived from the DD system, of which the IWOS is an evolution. A comparison of the maneuverability of the DD and IWOS described the motions that the IWOS is capable of without violating the sliding constraints of the robot's wheels.

Building upon the knowledge acquired, a prototype of the IWOS was developed, the first of its kind. This prototype serves as both a proof of concept for the IWOS system as well as a starting point for further research. The design processes of both the hardware and software for this prototype were described.

An odometry system was introduced for this prototype, which enables the tracking of changes in the robot's pose. However, experiment results revealed that the odometry approach is prone to drift.

In addition to the odometry system, this thesis introduced an advancement of the OMM by Thrun, adapting that approach for the IWOS. Experiment verification of this OMM yielded promising results for linear and curved motions, showcasing its potential for accurate motion estimation.

These findings contribute to a deeper understanding of the IWOS locomotion system and pave the way for further advancements and applications in the field.

8.1 Further Work

Since this thesis explores a novel robot locomotion system and involves the construction of the first prototype, it not only addresses research questions but also highlights areas for further development. The following list presents future opportunities for advancing the IWOS system, prioritized based on the author's opinion:

- Further improvements to the IWOS are necessary, mainly in terms of both hardware and software enhancements. Enhancing joint operation in terms of frequency and accuracy should be a primary goal. Additionally, considering porting the robot's software from ROS to ROS2 might prove beneficial.
- The IWOS has the potential to enhance collision detection. While this claim lacks a proof of concept for the IWOS, specifically, research has shown that it is possible for other robot systems. Investigating and providing evidence for collision detection capabilities in the IWOS would be valuable. If this proves challenging, exploring simpler solutions for an offset between the facing and driving directions could be considered.

- Research suggests that the offset between the driving and facing directions, enabled by the IWOS, could be advantageous for HRI. However, no empirical evidence has been presented thus far to support this claim. Conducting experiments to validate the potential benefits of the IWOS in HRI scenarios would be valuable.
- Currently, there is no existing path-planning or path-following approach specifically designed for the IWOS. Developing a dedicated tool or algorithm for this purpose is highly desirable and essential to enabling autonomous operation of robots with the IWOS locomotion system.

By addressing these tasks, future research and development has the potential to advance the IWOS locomotion system, taking it closer to its full potential and expanding its applicability in various domains.

List of Figures

List of Tables

List of Algorithms

Acronyms

CAD Computer Aided Design. 30, 32

DD Differential Drive. xi, xiii, 1–4, 6, 8, 11–18, 21–25, 31, 42, 43, 46–49, 52–54, 57–62, 64–68, 71, 72, 75, 79

FDM Fused Deposition Modeling. 32

HRI Human-Robot Interaction. xi, xiii, 2, 4–6, 8, 9, 71, 73

ICC Instantaneous Center of Curvature. 16–26, 31, 49, 52, 63, 65, 68, 75

ICR Instantaneous Center of Rotation. 16

IMU Inertial Measurement Unit. 8

IWOS Independent Wheel Offset Steering. xi, xiii, 1–5, 7–9, 11–17, 20–28, 30, 31, 34, 36, 39–43, 49–55, 57, 58, 60–69, 71–73, 75, 76, 79

OMM Odometry Motion Model. xi, xiii, 4, 38, 45, 46, 48–69, 72, 77, 79

PETG Polyethylene Terephthalate Glycol. 32

PID proportional–integral–derivative. 36

REP ROS Enhancement Proposals. 15, 35

ROS Robot Operating System. 8, 15, 29, 33, 34, 36, 38–41, 51, 52, 72

ROS2 Robot Operating System 2. 8, 34, 72

URDF Unified Robot Description Format. 37

Bibliography

[BAM17] Mordechai Ben-Ari and Francesco Mondada. *Elements of robotics*. Springer Nature, 2017.

[BNS02] Allison Bruce, Illah Nourbakhsh, and Reid Simmons. The role of expressiveness and attention in human-robot interaction. In *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, volume 4, pages 4138–4142. IEEE, 2002.

[CAKL15] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. That's on my mind! robot to human intention communication through on-board projection on shared floor space. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[CKTB19] Firat Berk Cakar, Eugen Kaltenegger, George Todoran, and Markus Bader. Independent Offset Wheel Steering: An Improved Approach to Wheeled Mobile Robot Locomotion. In *Proceedings of the ARW & OAGM Workshop 2019*, pages 32–37, 2019.

[CMEM+17] Sachin Chitta, Eitan Marder-Eppstein, Wim Meeussen, Vijay Pradeep, Adolfo Rodríguez Tsouroukdissian, Jonathan Bohren, David Coleman, Bence Magyar, Gennaro Raiola, Mathias Lüdtke, and Enrique Fernández Perdomo. ros_control: A generic and simple control framework for ros. *The Journal of Open Source Software*, 2017.

[CMM05] Yang Cheng, Mark Maimone, and Larry Matthies. Visual odometry on the mars exploration rovers. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 1, pages 903–910. IEEE, 2005.

[F+08] Thomas Friedrich et al. *Elementary geometry*, volume 43. American Mathematical Soc., 2008.

[FJK+18] Rolando Fernandez, Nathan John, Sean Kirmani, Justin Hart, Jivko Sinapov, and Peter Stone. Passive demonstrations of light-based robot signals for improved human interpretability. In *2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 234–239. IEEE, 2018.

[HCVdL21] Nicholas J Hetherington, Elizabeth A Croft, and HF Machiel Van der Loos. Hey robot, which way are you going? nonverbal motion legibility cues for human-robot spatial interaction. *IEEE Robotics and Automation Letters*, 6(3):5010–5015, 2021.

[HDLAS17] Sami Haddadin, Alessandro De Luca, and Alin Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identification. *IEEE Transactions on Robotics*, 33(6):1292–1312, 2017.

[HMS12] Michael Hoy, Alexey S Matveev, and Andrey V Savkin. Collision free cooperative navigation of multiple wheeled robots in unknown cluttered environments. *Robotics and Autonomous Systems*, 60(10):1253–1266, 2012.

[HPV02] Mark A Hollands, Aftab E Patla, and Joan N Vickers. “look where you’re going!”: gaze behaviour associated with maintaining and changing the direction of locomotion. *Experimental brain research*, 143:221–230, 2002.

[HS13] Bradley Hayes and Brian Scassellati. Challenges in shared-environment human-robot collaboration. *learning*, 8(9), 2013.

[KLS16] Kwan Suk Kim, Travis Llado, and Luis Sentis. Full-body collision detection and reaction with omnidirectional mobile platforms: a step towards safe human–robot interaction. *Autonomous Robots*, 40:325–341, 2016.

[KZBS17] Gregor Klancar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc. *Wheeled mobile robotics: from fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.

[LS13] David V Lu and William D Smart. Towards more efficient navigation for robots and humans. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1707–1713. IEEE, 2013.

[MDH15] Alyxander David May, Christian Dondrup, and Marc Hanheide. Show me your moves! conveying navigation intention of a mobile robot to humans. In *2015 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[MHW+19] Sherif AS Mohamed, Mohammad-Hashem Haghbayan, Tomi Westerlund, Jukka Heikkonen, Hannu Tenhunen, and Juha Plosila. A survey on odometry for autonomous navigation systems. *IEEE access*, 7:97466–97486, 2019.

[MKA16] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. Exploring the performance of ros2. In *Proceedings of the 13th International Conference on Embedded Software*, pages 1–10, 2016.

[MKI06] Takafumi Matsumaru, Takashi Kusada, and Kazuya Iwase. Mobile robot with preliminary-announcement function of forthcoming motion using light-ray. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1516–1523. IEEE, 2006.

[MS14] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[MSKH15] Yoichi Morales, Satoru Satake, Takayuki Kanda, and Norihiro Hagita. Building a model of the environment from a route perspective for human–robot interaction. *International Journal of Social Robotics*, 7:165–181, 2015.

[NFBR23] Linus Nwankwo, Clemens Fritze, Konrad Bartsch, and Elmar Rueckert. Romr: A ros-based open-source mobile robot. *HardwareX*, 14:e00426, 2023.

[PPP17] Anish Pandey, Shalini Pandey, and DR Parhi. Mobile robot navigation and obstacle avoidance techniques: A review. *Int Rob Auto J*, 2(3):00022, 2017.

[PRJ+20] Oskar Palinko, Eduardo Ruiz Ramírez, William Kristian Juel, Norbert Krüger, and Leon Bodenhagen. Intention indication for human aware robot navigation. In *VISIGRAPP (2: HUCAPP)*, pages 64–74, 2020.

[QCG+09] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[RVLA19] Francisco Rubio, Francisco Valero, and Carlos Llopis-Albert. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16(2):1729881419839596, 2019.

[SL17] Ching-Long Shih and Li-Chen Lin. Trajectory planning and tracking control of a differential-drive mobile robot in a picture drawing application. *Robotics*, 6(3):17, 2017.

[SNS11] Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza. *Introduction to Autonomous Mobile Robots*. The MIT Press, 2011.

[TB16] George Todoran and Markus Bader. Expressive navigation and local path-planning of independent steering autonomous systems. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4742–4749. IEEE, 2016.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[THM18] George Todoran Horatio and Bader Markus. Mobiles system. Austrian Patent Office, 2018. Docket Number: A50232/2018.

[VCBV23] Niclas Vödisch, Daniele Cattaneo, Wolfram Burgard, and Abhinav Valada. Covio: Online continual learning for visual-inertial odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2463–2472, 2023.

[WC17] Ruffin White and Henrik Christensen. Ros and docker. *Robot Operating System (ROS) The Complete Reference (Volume 2)*, pages 285–307, 2017.

[WIM+15] Atsushi Watanabe, Tetsushi Ikeda, Yoichi Morales, Kazuhiko Shinozawa, Takahiro Miyashita, and Norihiro Hagita. Communicating robotic navigational intentions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5763–5769. IEEE, 2015.

[WLZ+21] Yong Wang, Xiaoxiao Li, Juan Zhang, Shuai Li, Zhihao Xu, and Xuefeng Zhou. Review of wheeled mobile robot collision avoidance under unknown environment. *Science Progress*, 104(3):00368504211037771, 2021.

[YBHH15] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4):289–311, 2015.