Federal State Autonomous Educational Institution of Higher Education
«PEOPLES' FRIENDSHIP UNIVERSITY OF RUSSIA
NAMED AFTER PATRICE LUMUMBA»

published as a manuscript

JAWOOSH KARRAR SAHIB NASSRULLAH

SYMBOLIC REGRESSION ALGORITHM FOR CONTROL OF NON-HOLONOMIC WHEELED MOBILE ROBOTS

2.3.1. Systems analysis, management and information processing, statistics (technical sciences)

Dissertation

for the degree of Ph.D. candidate of technical sciences

Scientific supervisor:
Doctor of Biological Sciences, Professor
Ivan Viktorovich Stepanyan

Moscow – 2025

Федеральное государственное автономное
образовательное учреждение высшего образования
«РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ
ИМЕНИ ПАТРИСА ЛУМУМБЫ»

На правах рукописи

ЖАВУШ КАРРАР САХИБ НАССРУЛЛА

АЛГОРИТМ СИМВОЛЬНОЙ РЕГРЕССИИ ДЛЯ УПРАВЛЕНИЯ НЕГОЛОНОМНЫМИ МОБИЛЬНЫМИ РОБОТАМИ НА КОЛЁСАХ

2.3.1. Системный анализ, управление и обработка информации, статистика
(технические науки)

Диссертация

на соискание ученой степени кандидата технических наук

Научный руководитель:
Доктор биологических наук, профессор
Иван Викторович Степанян

Москва – 2025

TABLE OF CONTENTS

ACKNOELEDGMENTS

I express my gratitude to Allah, the Most Gracious and Merciful, for granting me the capability and motivation to finish this research work, assisting and safeguarding me throughout the study duration. Additionally, I extend my heartfelt thanks and appreciation to Prophet Muhammad and his noble family for their continuous blessings and support.

I express my ongoing appreciation and profound gratitude to my supervisor, Professor Ivan Viktorovich Stepanyan, for his invaluable support and unwavering guidance.

I genuinely wish to express my gratitude to all the professors who have guided my studies at RUDN University since my enrollment in 2021.

I am deeply indebted to my family and would like to express my utmost gratitude to my mother, sisters, and brothers for their lifelong support and encouragement.

I extend my heartfelt gratitude to my beloved wife for her unwavering support throughout my academic journey, especially for assuming significant responsibilities during my travels and study periods. My thanks and affection also go to my beloved son, Ali, for his endurance and patience during my years of study.

Ultimately, I am unable to adequately convey my gratitude to my classmates for their interest and unrelenting backing during the execution of this study.

ABSTRACT

The 20th century is renowned for the development of computer-based automatic control systems utilized in industrial plants and manufacturing processes. In the 21st century, the contemporary control systems necessitate the capacity to adapt, enhance, and acquire knowledge swiftly. Consequently, mobile robots have emerged as a focal point of considerable scholarly interest in recent years. The wheeled mobile robot (WMR) possesses an extensive variety of practical applications. However, despite their potential and prospects, mobile robots have not yet achieved the best performance due to the intrinsic challenges they encounter. Several critical problems have appeared in this domain, including navigation and path planning, localization, and obstacle avoidance. Tracking of trajectories and the problem of point stabilization are the two main control problems concerning this kind of robot.

The field of machine learning control (MLC) is well-suited to address these emerging difficulties. The objective of machine learning control entails the identification of an unknown control function. Through symbolic regression, control functions are automatically synthesized as closed-form mathematical formulations. These formulations provide a structured and efficient framework for guiding robotic motion toward target locations while circumventing environmental obstacles. Symbolic regression methods are the exclusive means by which one can explore the very structure and parameters associated with mathematical expressions.

This work is motivated by the construction of a control system for a pair of nonholonomic mobile robots. The successful execution of the proposed control necessitates the establishment of a dual feedback loop (two steps). In the internal loop (the first step), the robot is rendered stable concerning a specific point within the state space. In order to address this objective, the general synthesis problem can be solved by utilizing the numerical technique of symbolic regression, which is a machine learning technique, to find feedback control functions. In the external loop (the second step), the problem of achieving effective control over the robots is addressed by the utilization of an evolutionary algorithm to influentially change the location of the stable points of equilibrium. The problem of control synthesis is initially addressed using the suggested novel technique (variational synthesized genetic programming technique). The control object achieves stability when it reaches an equilibrium point inside the state space. These stabilization points can be changed, giving a chance to look up the coordinates of various stabilization points in order to get the mobile robot to go from its starting point to its destination with the improved quality criterion value and trajectory using the particle swarm optimization algorithm. The state space's required trajectory must exhibit an attractive property for suitable solutions within a certain vicinity.

The aforementioned methodology is referred to as the synthesized optimal control problem. This novel methodology not only presents a fresh perspective on addressing a widely recognized challenge in the field of optimal control but also introduces a novel problem statement that facilitates its numerical solution.

The proposed methodology has been applied to a pair of mobile robots. The mobile robots are tasked with modifying their planar coordinates to satisfy static phase conditions to achieve obstacle-free navigation, with an additional imperative: to maintain collision-free trajectories relative to one another throughout the mission. As demonstrated by the experimental outcomes, the two mobile robots successfully navigated to their target configurations under full compliance with phase constraints and without any occurrence of mutual collision, underscoring the efficacy of the control system. As seen from the findings, the effective control exerts an attractive influence on the relevant state-space component, without requiring kinematic matching with that component. It is widely recognized that the speed of state evolution is reduced in the immediate neighborhood of an equilibrium point compared to distant regions. Thus, for enhanced mobility, the control object should be maintained in the vicinity of that point without settling into it, allowing for continuous and faster motion.

INTRODUCTION

Relevance and level of development of the research topic

Contemporary developments in industrial automation have been driven by the integration of intelligent robotic systems that exhibit self-learning behaviors and a high degree of operational autonomy. These advanced robots are designed to function across a broad spectrum of tasks without requiring constant supervision. Specifically, mobile robots with non-holonomic constraints and wheel-driven locomotion are widely utilized in industrial automation, supporting activities such as assembly line processes, warehouse navigation, and facility maintenance.

This dissertation investigates a mobile nonholonomic robot characterized as a complex, nonlinear system designed for autonomous locomotion. The primary focus lies in the formulation and analysis of control algorithms for a pair of such robots, ensuring robust performance in heterogeneous operational settings and enabling task execution without human intervention. The relevance of this research is underscored by the increasing necessity for adaptive, intelligent robotic systems that can respond effectively to unpredictable environmental changes while maintaining autonomous functionality.

Numerous studies in the scientific domain focus on the synthesis of control architectures and the optimization of dynamic trajectories. Particular emphasis has been placed on analytical and computational methods for resolving control challenges—areas that have been profoundly shaped by the seminal contributions of renowned scholars, including S. Wolfram, W.R. Ashby, W. McCulloch, W. Pitts, P.K. Anokhin, L.S. Pontryagin, A.I. Diveev, N. Wiener, and A.N. Kolmogorov.

The implementation of optimal control strategies faces a key challenge: the inability to directly apply time-parameterized control functions to actual physical systems. This limitation arises from the open-loop configuration, which offers no correction mechanism in the presence of disturbances, potentially leading to substantial trajectory deviations and failure to meet performance criteria. In mobile robotics, effective control necessitates robust stabilization and high-fidelity trajectory tracking. The stability of the closed-loop system is commonly ensured by stabilizing the state trajectory near an equilibrium point within the state space, which serves as a foundation for robust autonomous operation.

The Purpose of the Dissertation Work

This work seeks to contribute to the field of intelligent control by developing and improving machine learning-based strategies for multi-agent systems, exemplified by a pair of non-holonomic wheeled mobile robots. The pursuit of this goal necessitates addressing the following specific tasks:

1. An investigation into genetic programming methods, evolutionary optimization techniques, and symbolic regression algorithms to advance automated model discovery and control system design.
2. Development of a numerical control approach that guarantees collision avoidance between two mobile robotic agents, as well as between each agent and the obstacles in the workspace.
3. Development of a symbolic regression-based control synthesis method that exploits the small variations principle to ensure stabilization of a robot towards a specified equilibrium point inside the state space.
4. Application of an evolutionary algorithm to dynamically reposition stable equilibrium points within a closed-loop control system that incorporates external feedback.
5. The outcome of the stabilization stage must be mathematically represented through a system of differential equations.

Object of Research

The focus of this study is on the maneuvering behavior of a two-robot system consisting of nonholonomic mobile platforms with differential drive actuation.

Subject of Research

The mathematical models and algorithmic support of the symbolic regression method, particularly as applied to identifying interpretable control function expressions and their numerical parameter values.

Methodology and Research Methods

The control object is endowed with a stabilization system that defines its essential dynamic property: a stable point of equilibrium within the state space. Robot control is accomplished by intelligently manipulating this point position, employing a methodological framework of an evolutionary algorithm, symbolic regression, and mathematical modeling through systems of differential equations.

The inner-loop control system, designed to stabilize the system around an operating point of equilibrium, is synthesized at an early stage and forms the cornerstone for the outer-loop control strategy that governs equilibrium point positioning. Such points can be set statically or modified online to accommodate environmental changes.

Through symbolic regression, control functions for mobile robots are automatically synthesized in the form of human-readable mathematical expressions. These formalized algorithms govern system behavior to meet mission objectives and maintain collision-free trajectories. Symbolic regression

facilitates the discovery of interpretable control functions by evolving both their functional form and tunable parameters. It follows from the universality of symbolic representations that, in the general case, symbolic regression can generate expressions that approximate the functional form of any neural network to a desired degree of accuracy [192].

Scientific Novelty of the Work

In the dissertation study, the following scientific novelty results are obtained:

1. An enhanced control problem formulation has been developed for nonholonomic mobile robotic systems, which includes additional design requirements to ensure the development of the stabilization system.
2. A novel machine learning approach—symbolic regression—has been introduced to facilitate the synthesis of control systems capable of achieving state-space stabilization.
3. The new approach synthesizes a dynamical system described by differential equations, leveraging the principle of small variations in the evolutionary processes of a genetic algorithm.
4. A new computational solution is contributed to the trajectory optimization problem for paired nonholonomic robots, explicitly accounting for geometric and kinematic constraints imposed by surrounding obstacles.
5. The fundamental problem of synthesizing control systems for nonlinear mobile robotic systems with identification of dynamic equations has been solved.

Theoretical Significance of the Work

An optimal control problem is established under extended constraint conditions, including the stipulation that the generated state-space trajectory must be attractive—that is, it must draw the system state into a given neighborhood. The proposed solution tackles the synthesis of a stabilizing feedback system for nonholonomic wheeled robots by engineering a stable point of equilibrium within the system's state space. And then, the control design is thereby reduced to the optimization of this point's location. The entire suite of computational tools employed is implemented as self-contained, automated numerical procedures.

Practical Significance of the Work

This study presents a synthesized optimal control methodology designed to solve trajectory and stability problems by explicitly controlling the location of the robot's stable point of equilibrium. The resulting methodology introduces a novel control paradigm based on equilibrium-point modulation.

The proposed methodology is specifically designed to address practical engineering challenges by reducing the gap between the theoretical mathematical model of the controlled system and its physical realization. This objective is accomplished through the integration of an inner-loop stabilization within the control architecture. Additionally, symbolic regression techniques exhibit broad applicability in the synthesis of control laws across diverse dynamical systems.

Main provisions to be defend

1. The developed control optimization method consists of two steps, where step one exemplifies stabilization step so that one nonholonomic mobile robot moved from 14 initial points to one terminal point; while step two exemplifies optimization step, where two nonholonomic mobile robots move from one initial point (different points) to a terminal one (also different points).

2. The variational synthesized genetic programming technique (VSGP) matrix consisting of 6 rows and 20 columns is used to define the control function of a nonholonomic mobile robot. The genetic algorithm parameters are: population size of 256, number of generations of 1024, number of crossovers in each generation of 128, variation depth of 10, and mutation probability of 0.75. A total of 30 functions are used, which make up the code space in the first stabilization stage. Two of these functions are binary operations, and 28 are unary.
3. To change the position of the robot's equilibrium point, a particle swarm optimization algorithm is used with control parameters : $\alpha = 0.5$, $\beta = 0.8$, $\gamma = 1.5$, and $\sigma = 1$, population size is 3500, number of generations is 150.

The Degree of Reliability of the Results

The proposed method’s effectiveness is supported by empirical results, including comparative assessments against Cartesian genetic programming [206] and parse-matrix evolution [208]. This study includes the development of a tailored mathematical model for simulating the dynamics of the Khepera II nonholonomic robot. Computational experiments were conducted to verify the accuracy and consistency of the dissertation’s outcomes.

Approbation of Research Results

The fundamental principles and results were deliberated upon and showcased at many international and Russian scientific conferences:

1. Using Symbolic Regression Methods for Machine Learning to Control Robot Motion: Advantages and Disadvantages. The XIV International Scientific and Practical

Conference “Modern strategies and digital transformations of sustainable development of society, education and science”. – Moscow: December 12, 2023.

2. Comparison of recurrent neural networks and symbolic regression methods. The XXII International Scientific and Practical Conference “Challenges of our time and development strategies of society in the conditions of the new reality”. – Moscow: December 15, 2023.
3. Problem of the Interpretability vs. Accuracy Trade-off in Symbolic Regression in robot motion: causes and solution. The II International Scientific and Practical Conference “Modern research: theory, practice, results”. – Moscow: December 29, 2023.
4. The 3rd International Conference on Engineering and Science, 3-4 May 2023 / Al-SAMAWA / IRAQ.

Furthermore, The principal findings, theoretical contributions, and practical recommendations derived from this dissertation have been disseminated through six peer-reviewed publications: four indexed in Scopus and two published in journals recognized by the Higher Attestation Commission (VAK).

Dissertation Structure

This dissertation is organized into several essential sections. Chapter 1 offers a thorough literature review of contemporary research regarding the wheeled mobile robots. Chapter 2 delineates the research methodology employed in this study, detailing the method of symbolic regression and the small variations principle. Chapter 3 presents the study's findings, which include a computational experiment of the synthesized optimal control strategy and its primary results. The thesis concludes with a summary of the research outcomes, along with conclusions and recommendations for future research directions.

CHAPTER 1. LITERATURE REVIEW

1.1 General Overview of Mobile Robots

The term "robot" can be traced back to its etymological origins in Slavic languages. The term "robota" in the Polish language signifies the concept of work or labor. However, it should be noted that in Czech or Slovenian, this word carries a more antiquated connotation and refers to statute labor or corvée. The term "robot" was originally introduced by the renowned Czech author Karel Capek in his science fiction drama titled R.U.R., an acronym for Rossum's Universal Robots. The robotic entities depicted in the theatrical production can be classified as a form of artificially created human-like beings. In contemporary parlance, the terms "cyborgs" or "androids" would be more suitable descriptors for these entities. The play experienced significant popularity, leading to the widespread adoption of the term "robot" in numerous global languages. Although the term "robot" has only been in existence for around a century, the concept of mechanical beings has a long and rich historical background. The term "mobile" originates from the Latin word "mobilis," which carries the same semantic connotation [1].

Mobile robots, as their designation suggests, possess the capacity for locomotion. These entities have the ability to traverse various mediums, including terrestrial surfaces, bodies of water, submerged environments, and aerial spaces. This stands in opposition to the prevalent use of fixed-base robotic manipulators in manufacturing operations, such as automobile assembly, electronic parts assembly, spray painting, and other related activities [2]. The significance of mobile robots is growing in various fields, including manufacturing and automated warehouses [3–5], domestic and medical aid [6–8], military uses [9-11], agricultural purposes [12-14], and rescue missions [15-16], and so on.

The emergence of mobile robots throughout the late 1960s and early 1970s marked the inception of a novel field of study known as autonomous navigation. It is noteworthy to mention that the initial navigation systems were presented during the inaugural International Joint Conference on Artificial Intelligence (IJCAI 1969). The methods mentioned earlier were founded upon critical concepts that have proven highly advantageous in the advancement of algorithms for robot motion planning. As an illustration, during the year 1969, the mobile robot Shakey employed a grid-based methodology to simulate and investigate the surrounding environment [17]. Similarly, in 1977, Jason utilized a visibility graph constructed from the corners of obstacles [18]. Furthermore, in 1979, Hilare employed a technique of decomposing the environment into convex cells that are free from collisions [19].

The concept of nonholonomic systems, derived from mechanics, was introduced in the literature [20] concerning robot motion planning, specifically in the context of car parking, around ten years later.

The problem at hand remained unresolved despite the ground-breaking research conducted in the field of mobile robot navigation. The scientific field of nonholonomic motion planning has gained significant attention [21].

1.2. Some Types of Mobile Robots

In order for a mobile robot to achieve unrestricted movement within its surroundings, it requires locomotion mechanisms. However, choosing a robot's strategy for locomotion is a crucial component of mobile robot development due to the wide range of alternative movement methods available. Within the laboratory setting, a diverse array of research robots has been developed with the capability to engage in various locomotion behaviors such as walking, jumping, running [22], sliding, skating, swimming [23], flying [24], and, naturally, rolling [25]. The majority of such locomotion mechanisms were originally derived from their biological equivalents.

With one notable exception, however: the actively propelled wheel, a human creation that achieves remarkable efficiency on level terrain. Biological systems already make use of something like this process. As can be seen in Figure 1.1, our walking bipedal system can be represented by a rolling polygon with sides of length *b* equal to the span of the step. The polygon evolves into a circular shape or wheel as the step size decreases. However, the technology required for wheeled locomotion—a completely spinning, dynamically propelled joint—was not developed by nature.

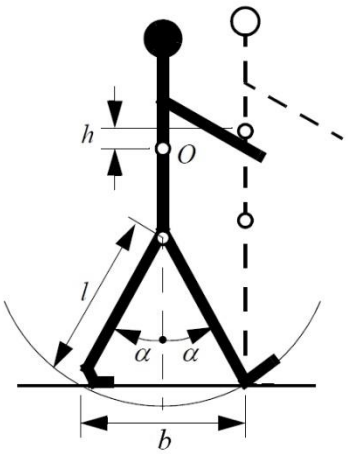

Figure 1.1. A walking bipedal system can be modelled using a rolling polygon [42]

Mobile robots often employ either wheeled systems, which are a widely recognized human invention for automobiles, or a limited number of articulating legs, representing the simplest basic form of biological locomotion.

In comparison to wheeled locomotion, legged locomotion often necessitates a higher number of mechanical degrees of freedom. Wheels are not only easy to use but also work very well on level surfaces. Rolling friction is reduced to a minimum on the railway's hard, flat steel surface, making it perfect for wheeled transportation. Wheeled locomotion, however, becomes increasingly inefficient when the surface softens as a result of rolling friction. At the same time, legged locomotion suffers significantly less as it consists entirely of contacts of points with the ground.

The effectiveness of wheeled locomotion is significantly influenced by environmental factors, specifically the levelness and firmness of the terrain. On the other hand, the effectiveness of legged locomotion is contingent upon the mass of the legs and the overall body mass, in both cases of which the robot should support them throughout different points of a legged gait.

It is comprehensible, hence, that nature exhibits a preference for locomotion, including legs, as natural locomotion systems must function on uneven and disorganized surfaces. Similarly, the human environment often has deliberately designed, polished surfaces found in both inside and exterior spaces. Hence, it is comprehensible that nearly all industrial implementations of mobile robotics employ a variant of wheeled mobility. In recent times, there has been notable advancement in the development of hybrid and legged industrial robots, particularly in the context of creating more organic outdoor settings. One prominent example of this improvement is the forestry robot.

1.2.1. Legged Mobile Robots

Legged mobile robots consist of many rigid bodies that are coupled through prismatic or, more commonly, revolute joints. Certain entities in the context possess anatomical structures that constitute the lower appendages, commonly referred to as feet, which intermittently make touch with the surface of the earth in order to facilitate the process of movement. This category encompasses a diverse array of mechanical structures, which frequently draw inspiration from the study of real organisms, known as biomimetic robotics. These structures span from biped human beings to hexapod robots, with the objective of emulating the biomechanical efficiency observed in insects.

The robot's legs make a variety of point contacts with the ground as it walks where two major benefits comprehend adaptability and maneuverability in challenging terrain. As long as the robot has

sufficient ground clearance, it doesn't matter how uneven the ground is between the set of point contacts. As a bonus, a walking robot can cross a gap as long as its reach is greater than the gap's breadth. The ability to deftly move objects in the surroundings is the icing on the cake of the benefits of legged locomotion. The dung beetle is a particularly impressive bug because of its ability to roll a ball with its deft front legs while moving.

The primary drawbacks associated with legged locomotion encompass problems pertaining to power consumption and mechanical intricacy. The leg, which may possess multiple degrees of freedom, should have the ability to support a portion of the robot's overall weight. Additionally, numerous robots must possess the capability to elevate and descend the robot. Moreover, the attainment of high maneuverability is contingent upon the presence of an adequate quantity of degrees of freedom in the legs, enabling the application of forces in various directions.

The legs must be lifted off the ground and set back down in order to move forward. Gait is the coordinated motion of the whole body, including the feet, as they are placed and lifted (in timing as well as place) to propel the walker forward.

In the context of legged mobile robots, it is often necessary to have at least two degrees of freedom in order to facilitate the forward movement of a leg. This involves the act of raising the leg and subsequently swinging it forward. The inclusion of an additional degree of freedom is an increasingly prevalent practice in order to facilitate more intricate motions. The recent advancements in the development of bipedal walking robots have resulted in the incorporation of an additional degree of freedom at the ankle joint. The ankle joint allows the robot to manipulate the resultant force vector generated by contact with the ground by controlling the position of the foot's sole through actuation.

In a broad sense, the incorporation of more degrees of freedom in a robotic leg enhances the maneuverability offered by the robot. This augmentation encompasses an expanded capacity to traverse diverse terrains and enables the robot to adopt various gaits during locomotion. The principal drawbacks associated with the incorporation of supplementary joints and actuators have been primarily related to energy consumption, control mechanisms, and overall bulk. The inclusion of supplementary actuators necessitates both energy and control while also contributing to the overall mass of the leg, hence amplifying the power and load demands placed on pre-existing actuators.

The coordination of legs for movement, often known as gait control, poses a significant challenge in the context of a mobile robot with several legs. The quantity of potential gaits is contingent upon the quantity of legs [26]. The main objective of early studies regarding multilegged walking robots concentrated on the design of robot locomotion for traversing smooth or somewhat rough terrain, navigating basic obstacles, moving on soft ground, and performing body maneuvers, among other related

aspects. These needs can be achieved by the implementation of periodic gaits and the utilization of binary (yes/no) information regarding contact with the ground. Recent research has focused on the development of quadrupedal robots capable of traversing challenging environments, including inaccessible roads and highly intricate terrains such as mountainous regions, ditches, pits, and locations affected by seismic activity. In such instances, it is important to possess further functionalities, together with comprehensive assistance in determining reactions and forecasting the stability of robots [27-30].

1.2.2. Tracked Mobile Robots

Tracked robots exhibit enhanced flexibility and possess the ability to navigate over challenging terrains. Nevertheless, their navigational capabilities are comparatively less precise when compared to those of a wheeled robot. Tracked robots necessitate the utilization of dual motors, with each motor assigned to a specific track located on either the left or right side. The locomotion of these robots is facilitated by a pair of tracks, which are set in motion through the rotation of wheels that are positioned within the sprockets of the robot [31].

Track mechanisms [32] are designed to provide precise linear motion and are well-suited for navigating uneven terrains, a common challenge encountered in off-road conditions. In contrast, these mechanisms exhibit a substantial size [33] and are distinguished by their somewhat lower energy efficiency for rotational motion in comparison to alternative driving mechanisms. The utilization of skid steering is prevalent in the operation of these vehicles. However, it is important for the reader to acknowledge that these maneuvers entail a complex interplay between the ground track and slippage events, which remains an area of an ongoing investigation within the discipline of ground mechanics. In order to effectively model and operate robots of this nature, it is important to conduct extensive experimentation prior to the formulation of the control scheme [34].

The skid steering principle, seen in Figure 1.2, operates by manipulating the relative velocities of the two tracks, similar to the manner in which differential drive vehicles with wheels function. Nevertheless, the task of controlling tracked locomotion presents a more intricate challenge due to the variance in the relative velocity of both tracks, which leads to slippage, soil shearing, and compaction as necessary mechanisms for achieving steering.

Tracked mobile robots have demonstrated their utility in various domains, including but not limited to the agricultural sector, rescue and search, military operations, forestry management, mining activities, and exploring other planets [35].

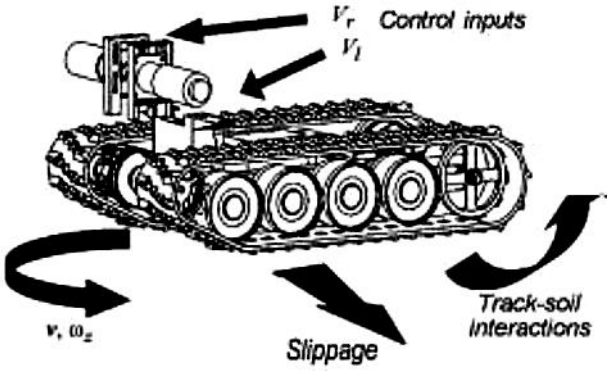

Figure 1.2. Principle of skid steering [35]

1.2.3. Wheeled Mobile Robots

The development of wheeled locomotion stands as a significant invention in human history. The invention of the wheel is estimated to have occurred about in 3000 BCE, whereas the development of the two-wheeled cart is believed to have taken place around 2000 BCE. Currently, the presence of four-wheeled cars is pervasive, with the global automobile population exceeding one billion. The efficacy and widespread usage of automobiles render them a logical selection as robotic platforms for terrestrial locomotion [36].

Wheeled mobile robots (WMR), commonly known as "ground mobile robots" in the field of robotics, typically have a stiff body, referred to as the base or chassis, and a wheel system that facilitates movement on the ground [37]. Additional rigid bodies, such as trailers, which are likewise equipped with wheels, can be linked to the base via revolute joints [38].

Wheeled robots are commonly used for the purpose of achieving mobility because of their numerous advantages, such as an uncomplicated structure, high energy efficiency, rapid speed, and inexpensive manufacturing cost, among others.

The wheel has emerged as the predominant mode of movement in the field of mobile robots and across many man-made vehicles. The system is capable of attaining high levels of efficiency while employing a rather straightforward mechanical design. Furthermore, the issue of balance is typically not a subject of research in the realm of wheeled robot designs. This is mostly due to the fact that wheeled robots are typically engineered in such a way that ensures continuous contact between all wheels and the ground throughout their operation. According to previous research [39-41], it has been established

that the use of three wheels is adequate to ensure stable balance. However, it should be noted that stability can also be achieved with two-wheeled robots.

In the context of robots intended for all-terrain conditions and those equipped with over three wheels, it is typically necessary to integrate a suspension system in order to ensure continuous contact between the wheels and the ground surface. One of the most straightforward methods for implementing suspension is incorporating a degree of elasticity directly into the wheel structure. In the context of certain indoor robots equipped with four wheels and castor wheels, the makers have implemented a rudimentary suspension system by incorporating a deformable tyre made of soft rubber onto the wheel. Naturally, this constrained method is unable to rival a developed suspension system in scenarios when the robot necessitates greater dynamic suspension to navigate considerably uneven terrain [42].

Each one of the wheels in a wheeled mobile robot (WMR) possesses the ability to rotate independently around an axis of its own. Consequently, there is a shared point that can be identified as the point of intersection of all the wheels' axes. The term used to refer to this concept is the instantaneous centre of rotation (ICR) or instantaneous centre of curvature. It designates a given point around which all of the wheels exhibit uniform angular velocity during their circular motion, as stated by ICR [43].

Rather than prioritizing balance, the field of wheeled robot research prefers to concentrate on addressing problems related to traction and stabilization, maneuverability, and control, which are contingent upon the types of wheels and configurations (drives) employed.

1.3. Wheel Types

WMRs, or Wheeled Mobile Robots, commonly employ two primary categories of wheels: conventional wheels as well as special wheels [38],[44]. There is significant variation in the kinematics of mobile robots, leading to a substantial impact on overall kinematics based on the chosen wheel type.

1.3.1. Conventional Wheels

There are three distinct categories of conventional wheels, as illustrated in Figures 1.3 and 1.4, accompanied by the corresponding icons that will be employed for their representation:

- *The fixed wheels or powered fixed wheels*: The propulsion of these wheels is facilitated by motors that are affixed to stationary locations on the vehicle. The wheel has the ability

to undergo rotation around an axis that passes through its centre and is perpendicular to the plane of the wheel. The wheel is affixed firmly to the chassis, resulting in a consistent orientation of the chassis with regard to the wheel.

- *The caster wheels*: These wheels lack power but possess a pair of axes of rotation. However, the vertical axis cannot cross the wheel's center, instead being consistently displaced by a fixed offset. This configuration induces the wheel to rotate spontaneously, swiftly matching it with the chassis' direction of motion. The introduction of this particular form of wheel serves the purpose of offering a backing point for static equilibrium while maintaining the maneuverability of the base. Caster wheels, for example, find widespread application in shopping carts and wheeled chairs.
- *The steerable wheels or powered steering wheels:* Each wheel operates under independent motorized drive and is capable of steering through rotation about an axis orthogonal to its rotational axis, enhancing navigational flexibility. The wheels in question have the potential to exist either equipped with offset or without offset, resulting in a scenario where the rotational and steering axes do not cross. There are a pair of axes of rotation present. The initial wheel is identical to a fixed wheel, whereas the subsequent wheel is oriented vertically and passes across the wheel's central axis. This mechanism enables the wheel to alter its orientation relative to the chassis.

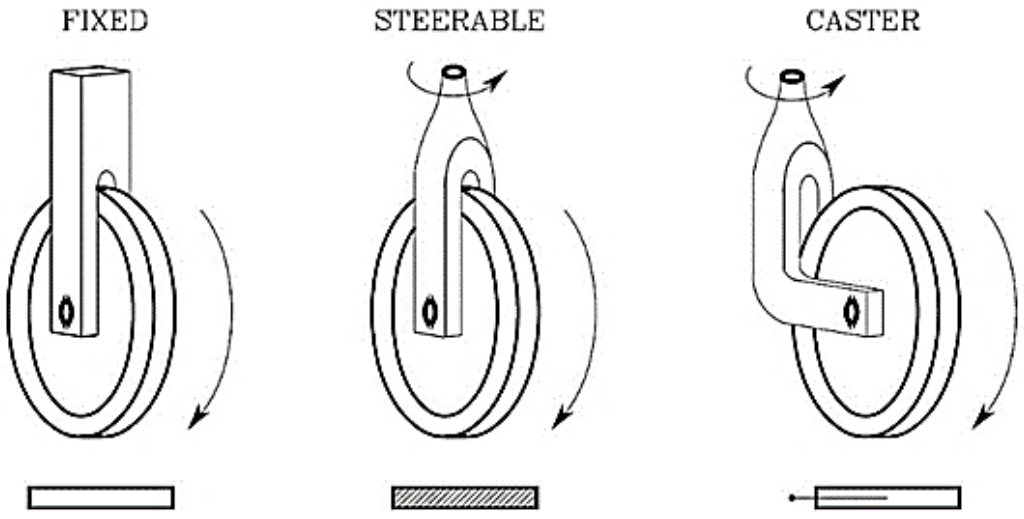

Figure 1.3. The three common types of conventional wheels and their corresponding icons [38]

Figure 1.4. Conventional wheels (A) fixed wheel, (B) caster wheel, (C) powered steering wheel without any offset, and (D) powered steering wheel with longitudinal offset [44]

In comparison to special wheel structures, conventional wheels have superior load capabilities and greater resilience towards terrain disturbances. However, owing to their nonholonomic restrictions, these wheels do not possess true omnidirectionality.

1.3.2. Special Wheels

The design of these wheels enables them to exhibit active traction in a particular direction and passive movement in another one, hence enhancing maneuverability in crowded conditions. There exist three primary categories of special wheels:

- Universal wheel
- Mecanum wheel or Swedish wheel
- Ball wheel or spherical wheel

Figure 1.5 illustrates the universal wheel, which offers a blend of restricted and unrestricted motion when turning. The wheel is equipped with little rollers positioned orthogonally to the axis of rotation around its external diameter. Additionally, to perform the normal rotation of the wheel, this mechanism enables the wheel to roll parallel to its axis.

Figure 1.5. Three configurations of universal wheel [44]

The mecanum wheel [45-49], also known as the Swedish wheel, as depicted in Figure 1.6, is a type of wheel that is analogous to the universal wheel with the exception that its rollers are positioned at an angle ψ, typically around ±45° and other than 90°.

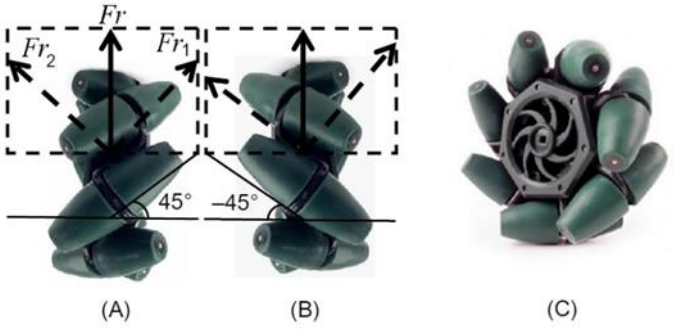

Figure 1.6. (A) Mecanum wheel with ψ = 45° (left wheel), (B) Mecanum wheel with ψ = -45° (right wheel), and (C) an actual functioning mecanum wheel [44]

The omnidirectional wheel is depicted in Figure 1.6 A and B, showcasing its appearance when observed from the bottom through a glass floor. The force *Fr* generated by the rotational motion of the wheel is transmitted to the ground through the roller that is in contact with the ground. It is assumed that the ground is sufficiently level and free from any abnormalities. At this particular roller, the applied force can be separated into two components: a parallel force, denoted as Fr_1, which parallels the axis of the roller, and a perpendicular force, denoted as Fr_2, which is oriented at a right angle to the axis of the roller. The force acting perpendicular to the axis of the roller induces a small rotational motion in the roller at a velocity denoted as v_v. Conversely, the force acting parallel to the axis of the roller applies a force on the wheel, consequently on the auto leading to hub velocity, v_h. The resultant velocity (v_t) of the auto is the sum of the horizontal velocity (v_h) and the vertical velocity (v_v). The actual functioning mecanum wheel is depicted in Figure 1.6 C.

The Swedish wheel operates similarly to a normal wheel, but it also offers reduced resistance in an additional direction, sometimes orthogonal to the traditional direction. The passive nature of the small rollers positioned along the periphery of the wheel is complemented by the active power exerted solely through the wheel's major axis joint. One notable benefit of this particular design is its ability to provide movement along various trajectories with minimal friction, despite the fact that the rotation of the wheel is solely propelled along the major axis (by the axle). This design enables the wheel to traverse not only forward and backward paths but also numerous other potential trajectories.

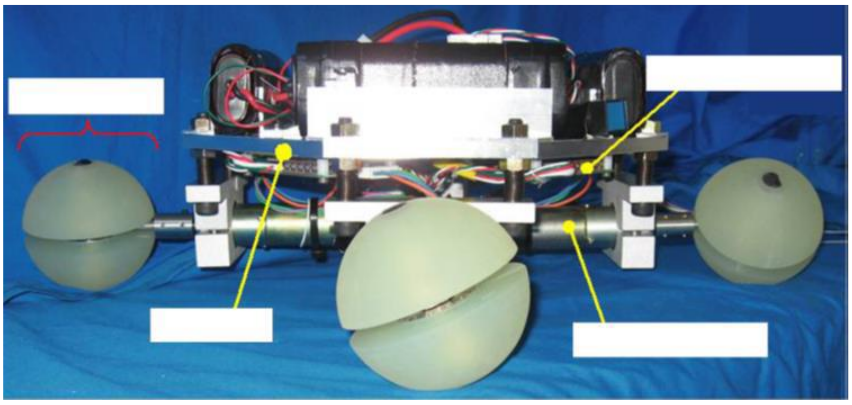

Figure 1.7. A practical application of a ball wheel [44]

The ball or spherical wheel does not impose direct limitations on motion, as it possesses omnidirectional capabilities similar to castor or special mecanum and universal wheels. Put otherwise, the wheel's rotational axis is capable of assuming any arbitrary orientation. One potential method for accomplishing this objective involves employing an active ring that is powered by a motor as well as a gearbox. This active ring serves to convey power to the ball through the utilization of rollers and friction. Notably, the ball possesses the ability to rotate freely in any direction without delay. Due to its intricate design, the ball wheel remains seldom employed in practical applications. Figure 1.7 illustrates a particular variant of a ball wheel. One approach to executing this spherical design involves emulating the functionality of a computer mouse, wherein powered rollers are employed to make contact with the upper surface of the sphere and generate rotating force.

Wheeled mobile robots (WMR) are extensively employed in various applications to accomplish robot locomotion. In a broad sense, wheeled robots tend to exhibit lower energy consumption and higher velocity compared to alternative locomotion systems such as legged robots or tracked vehicles. From a control perspective, the simplicity of their mechanisms and the lessened occurrence of stability issues result in a decreased need for control effort. Despite the inherent challenges posed by rugged terrain and uneven conditions of the ground, wheeled mobile robots have proven to be well-suited for a wide range of target situations in various practical applications [50].

The selection of wheel kinds for a mobile robot has become inextricably connected to the selection of wheel arrangement, often known as wheel geometry. When building the locomotion system of a wheeled robot, the mobile robot engineer must take into account two concurrent difficulties. What is the significance of wheel kind and wheel geometry? The decisions made in designing a robot have a direct impact on three essential attributes: controllability, maneuverability, and stability [51-52].

A plethora of design alternatives exist regarding wheeled mobile robots. The design problems associated with a single-body mobile robot encompass the choice of wheel types, the optimal positioning of wheels, and the precise determination of kinematic parameters. The specification of design targets should be contingent upon the specific target circumstances and tasks, in addition to considering the initial costs and operational expenses associated with the operation of a robot.

In contrast to automobiles, which are primarily engineered to operate inside a standardized environment such as the road network, mobile robots are specifically intended to cater to a diverse range of applications and scenarios. Automobiles exhibit commonality in their wheel configurations due to the existence of a specific region within the design space that optimizes their controllability, maneuverability, and stability within the typical environment they operate in, namely, the paved roadway. As a result, a notable drawback associated with wheeled robots is their reliance on a paved road or a level terrain for effective locomotion. Nevertheless, it is important to note that there is no singular wheel configuration that optimizes these characteristics for the diverse range of environments encountered by various mobile robots.

1.4. Drive Types

Wheeled robots, as demonstrated in Figure 1.8, represent a minimalistic yet effective design paradigm in mobile robotics, which commonly incorporates one or more powered wheels to facilitate motion, as seen by the solid rectangles in the illustration. Additionally, they may feature passive caster wheels, represented by hollow rectangles, which serve to enhance stability. Likewise, it is possible for these vehicles to possess steered wheels, often represented by wheels depicted within a circular shape to indicate their axis of rotation. Typically, the driving and steering mechanisms for a mobile robot necessitate the utilization of a pair of motors in the overall design [53-57].

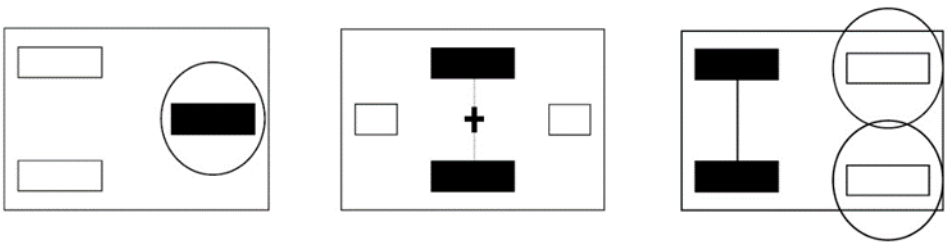

Figure 1.8. Various drive types used for the design of wheeled mobile robots [56]

The drives of Wheeled Mobile Robots (WMRs) can be categorized into the following:

1.4.1. Differential Drive

A differential drive system is characterized by two motors mounted in a fixed orientation on the robot's lateral sides, each providing independent propulsion to a single wheel. As this configuration provides only two points of ground contact, supplemental passive elements—such as caster wheels or sliders—are integrated to fulfill the minimum requirement of three ground-contact points for static stability. Furthermore, it is acknowledged that the interaction between the wheels of a robot and the ground is characterized by a state of non-slipping and pure rolling [58]. The differential drive system is considered to be mechanically more straightforward compared to the single-wheel drive system, as it eliminates the need for the rotating motion of a driving motor. Nevertheless, Compared to single-wheel drive systems, differential drive robots exhibit increased control complexity in directional navigation, attributable to the need for precise synchronization between the two driven wheels.

The presence of only one passive wheel in a differential drive system restricts the ability to position the driven wheels centrally, as such a configuration would compromise stability. Consequently, the robot rotates about a pivot point located between the two driven wheels, which is offset from the center. With two passive wheels (front and rear), however, the robot can achieve rotation about its center, enhancing directional control and operational efficiency. Nevertheless, this particular design may give rise to surface contact problems due to its utilization of four contact points rather than the more conventional three.

A differential drive robot's driving operations are depicted in Figure 1.9. When both motors operate at equal speeds, the robot moves in a linear path, either forward or backward. However, if one motor operates at a higher speed compared with the other, the robot follows a curved trajectory along the curve of an instantaneous circle. Moreover, To achieve a point turn, the control system commands the left and right motors to run at equal speeds in reverse directions, causing the platform to rotate about the center of its driving wheel axis.

- Driving linear forward or backward: $v_l = v_r,\ v_l > 0$
- Driving in a rightward curve: $v_l > v_r$
- Rotation in a clockwise direction on the spot: $v_l = -v_r,\quad v_l > 0$

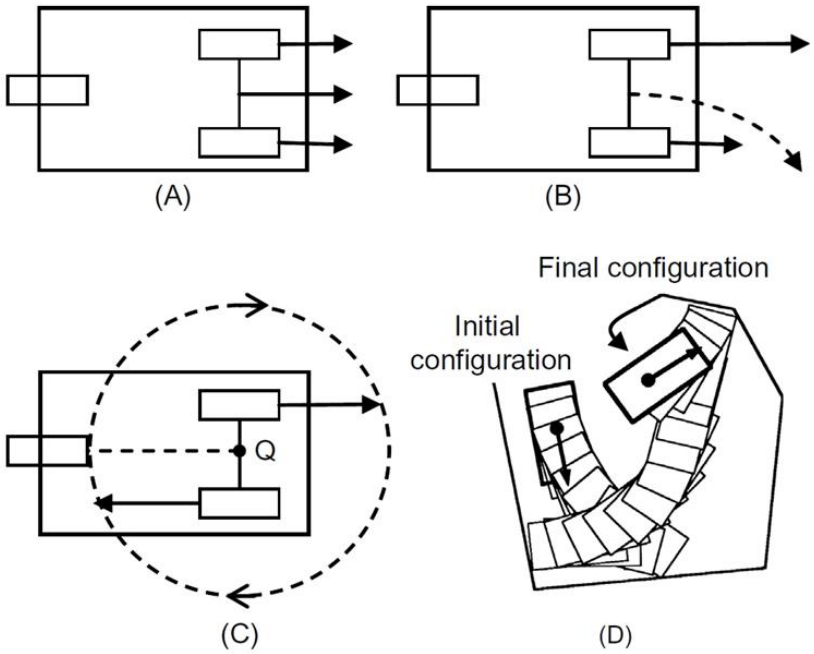

Figure 1.9. Driving operations of differential drive. (A) Straight path, (B) Curved path, (C) Circular path, and (D) Obstacle-free navigation to move from an initial to a final state [44]

1.4.2. Tricycle or Single Wheel Drive

The present mechanism is equipped with a solitary wheel that fulfills the dual functions of driving and steering. In order to ensure stability, a configuration involving two unpowered fixed wheels positioned at the rear is employed, hence maintaining the necessary three-point contact at all times. The system necessitates the utilization of two motors, with one motor dedicated to driving the vehicle's wheel and the other motor dedicated to turning. One notable benefit of this design is the complete decoupling of the driving and turning motions through the utilization of two distinct motors. Consequently, the control software designed for maintaining straight trajectories or executing curved paths will exhibit a high degree of simplicity. When driving in a straight line, the wheel is placed in the central position and operated at the desired velocity, see Figure 1.10A. Once the front wheel becomes inclined, the vehicle has a trajectory that is curved, see Figure 1.10B. When the front wheel is set at a 90° angle, the robot will undergo rotational motion along a circular trajectory. This circular path is centered at the midpoint of the rear wheels rather than the geometric center of the robot, as depicted in Figure 1.10C. This implies that the WMR lacks the ability to rotate in place. The minimal turning radius refers to the measurement of the distance separating the frontal wheel and the midway of the two rear wheels. Nonholonomic wheeled mobile robots (WMRs), such as tricycle or differential drive robots, are unable to execute parallel parking procedures directly. However, they can achieve parallel parking through a series of

maneuvers involving both forward and backward movements, as seen in Figure 1.10D. The utilization of tricycle drive is prevalent in the field of mobile robotics due to the inherent stability provided by the three wheels, enabling the robot to maintain an upright position autonomously.

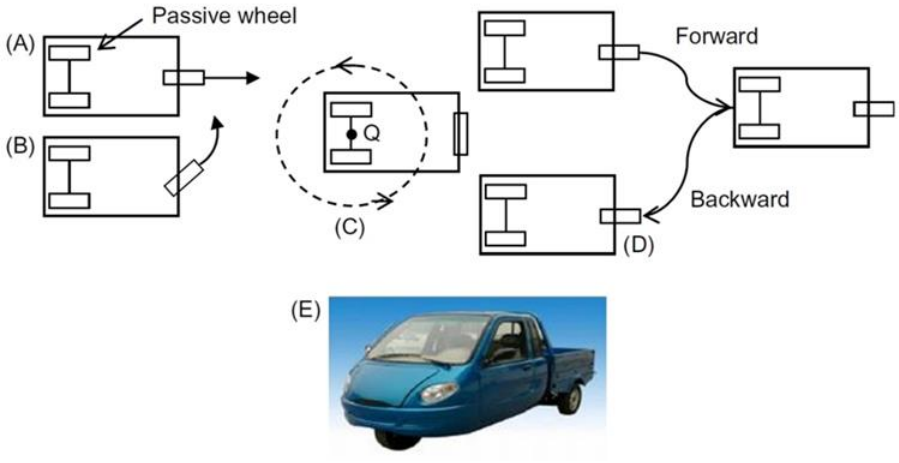

Figure 1.10. Tricycle WMR driving modes (A—D), (E) A tricycle example [44]

1.4.3. Synchro Drive

The drive in question consists of a minimum of three wheels that are interconnected in a manner that ensures simultaneous rotation in an identical direction and at an identical speed. Additionally, these wheels pivot collectively around their respective steering axes when executing a turn. A conventional synchro drive system is characterized by the presence of three wheels that are symmetrically positioned in an equilateral triangle configuration around the center of the vehicle. In this system, all wheels are guided in synchrony, resulting in their rotation axes consistently maintaining parallel alignment. Furthermore, the Instantaneous Center of Rotation (ICR) point is positioned at an infinite distance. There are different methods available for achieving mechanical steering synchronization, such as utilizing a belt, a chain, or a gear drive. The synchro drive system can be understood as an expansion of a single steered and driven wheel, hence maintaining a limited number of degrees of freedom, specifically two. Characterized by its near-holonomic kinematics, the synchro drive WMR exhibits omnidirectional mobility, allowing for movement along any path in the plane, typically requiring a cylindrical chassis to support omnidirectional movement.

Nevertheless, it is incapable of simultaneously driving and rotating. In order to transition from forward to lateral movement, the WMR (Wheeled Mobile Robot) must come to a halt and readjust the

alignment of its wheels. Figure 1.11A visually illustrates the movement and rotation of a three-wheel WMR (Wheeled Mobile Robot) equipped with a synchro drive.

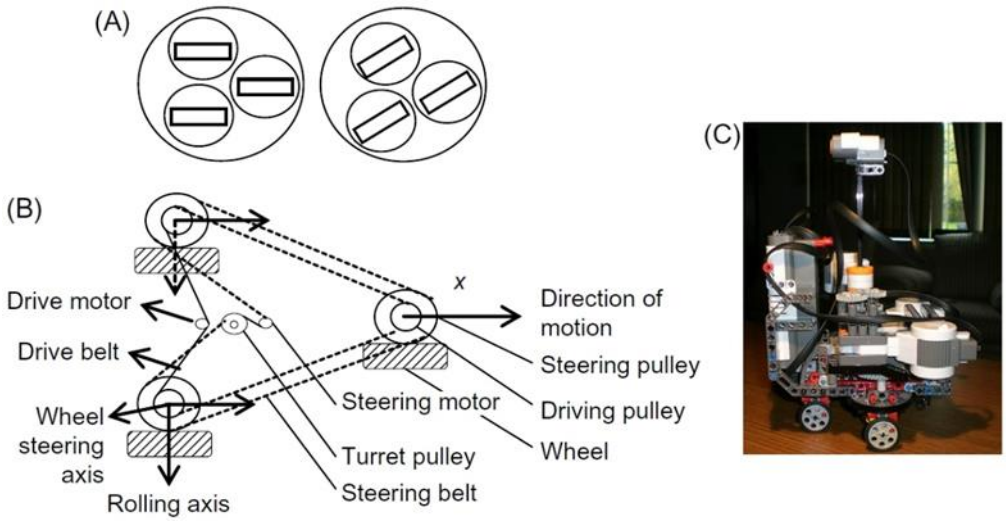

Figure 1.11. (A) An example of WMR motion with a synchro drive, (B) The two separate belts subsystems, and (C) A synchro drive example [44]

The utilization of a belt- or chain-based synchro drive results in diminished steering precision and alignment. The occurrence of this problem can be avoided by implementing a gear drive mechanism. In order to ensure proper functionality, it is necessary to employ two distinct motor-drive subsystems that operate using belt, chain, and gear mechanisms. These subsystems serve two distinct purposes: one of them for steering and another to control the driving shaft, as depicted in Figure 1.11B. The initial motor is responsible for regulating the rotational movement of the wheels along the horizontal axis, thereby supplying the force that drives (traction) the robot. The second motor governs the rotational movement of the wheels along the vertical axis, thus influencing their orientation.

It should be noted that the direction of the chassis remains constant throughout the action. Frequently, the inclusion of one more motor is observed in the design of such robots, with the purpose of enabling autonomous rotation of the upper section of the chassis, commonly referred to as a turret, in relation to the bottom section. This method could potentially be advantageous for the purpose of orienting a directional sensor, such as a camera, without any specific constraints or, alternatively, for correcting any errors in orientation [38].

One illustrative assignment that showcases the benefits of a synchro-drive system is the achievement of "complete area coverage" by a robot within a designated location. The practical use of this work can be observed in the context of cleaning or vacuuming floors.

1.4.4. Ackermann Steering

In a rear-wheel-drive vehicle, power is delivered to the rear wheels via a differential-connected motor, while the front wheels are passive and responsible for steering via synchronized actuation. The phenomenon referred to is commonly recognized as Ackermann steering, and offers comparative benefits and limitations when contrasted with the differential drive paradigm. A notable benefit is the ease of maintaining linear motion, as the rear wheels are driven through a single mechanical axis. However, this design limits maneuverability, as the vehicle cannot rotate in place and instead requires a minimum turning radius.

In Ackermann steering configurations, a distinct control interface is necessary because linear and angular velocities are produced by independent actuators, leading to full decoupling. This structural separation simplifies control, notably improving the accuracy and stability of straight-line travel. The driving library features dual independent control units: one responsible for the velocity and positional regulation of the rear wheels, and the other for the steering control of the front wheels. The inclusion of a position controller is necessary for the steering system since it is responsible for accurately setting the front wheels to a specific steering angle.

In contrast, the velocity controller is utilized to ensure a consistent speed is maintained by the back wheels. Slippage is observed in the rear driving wheels during the execution of turns. In order to accurately steer the front wheels, it is necessary to have supplementary sensors that can detect the zero position, as well as potentially the maximum right and left positions.

The Ackerman steering mechanism is specifically engineered to achieve a common cross point, known as the instantaneous center of rotation (ICR), for all-wheel axes during turns. This design feature aims to prevent wheel slippage resulting from geometric factors. It is able to find the equations from Figure 1.12 as follows:

$$cot\beta_s = \frac{a+E}{V}, \quad cot\beta_o = \frac{2a+E}{V}, \quad cot\beta_i = \frac{E}{V}, \tag{1.1}$$

By eliminating E, it can get

$$cot\beta_s = \frac{a}{V} + cot\beta_i\,, \quad or \quad cot\beta_s = cot\beta_o - \frac{a}{V}\,, \tag{1.2}$$

where β_s represents the automobile's true steering angle and β_o, β_i represent the outer and inner wheel's steering angles, respectively.

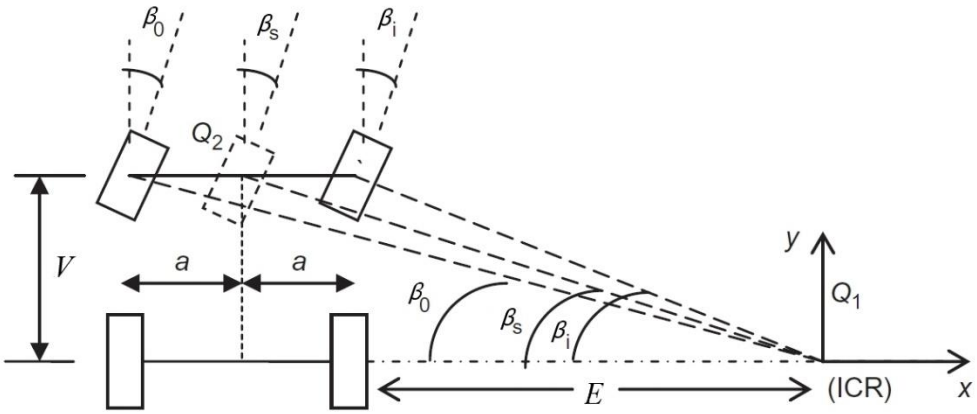

Figure 1.12. The intersection point (ICR) of the rotating axes across all wheels [44]

The limitation in the movement of an Ackerman-steered automobile is depicted in Figure 1.13. The automobile is currently situated in a location where there are two inaccessible circular areas, one on the left side and one on the right side. The reason for this limitation is that the robot is unable to execute turns (either to the right or left) when following a trajectory with a radius lower than a predetermined minimum value. Hence, the act of parallel parking necessitates a substantial degree of maneuvering.

It is noteworthy to mention that in order to prevent slippage, it is necessary for both of the front wheels to possess distinct orientations as the vehicle traverses a curve. Specifically, the internal wheel should be somewhat more steered in comparison to the external wheel.

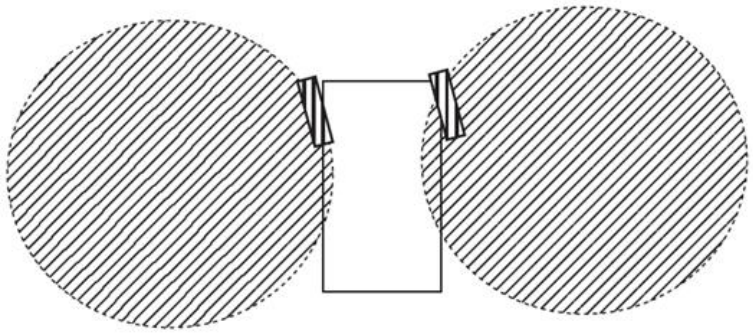

Figure 1.13. The inaccessible shaded areas by the Ackerman-steered robot [44]

1.4.5. Omni-Directional Robots (ODR)

As can be seen in Figure 1.14, a total of three, four, or even more omnidirectional wheels can be used to achieve this type of driving. As can be seen in Figure 1.14A, the universal wheels used on

three-wheeled WMRs all have a roller angle of 90 degrees (Figure 1.5). Mecanum wheels, as in Figure 1.6, are arranged like in Figure 1.14B on four-wheeled omnidirectional WMRs.

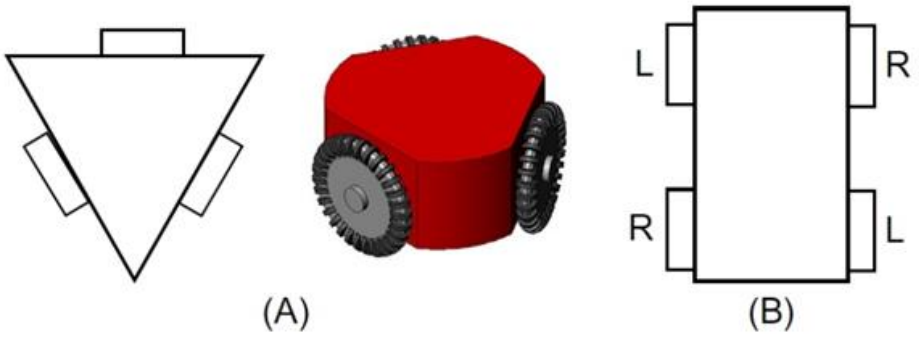

Figure 1.14. Omnidirectional WMRs. (A) Three-wheel example, (B) four-wheel example with roller angle various than 90° (typically ψ = ±45°) [44]

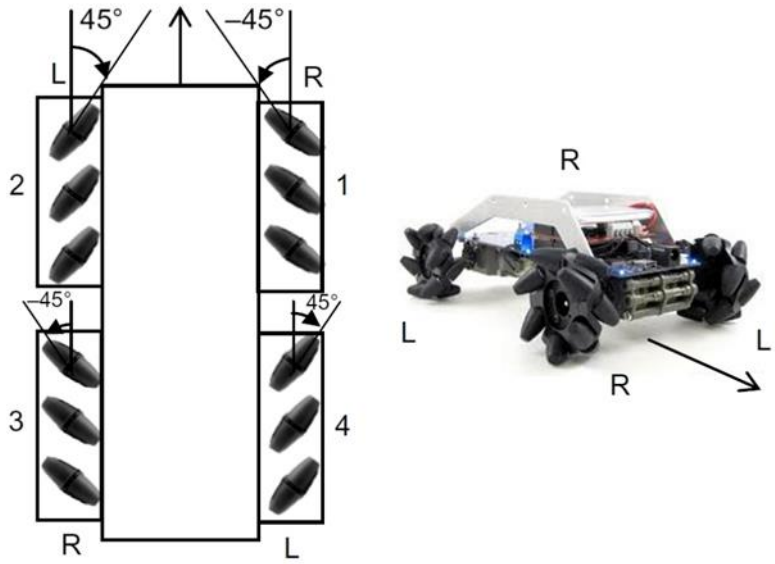

Figure 1.15. A conventional omnidirectional WMR uses a four-mecanum-wheel configuration [44]

Figure 1.15, it can observe four wheels total; two are designated as the left-side (L) wheels, while the other two are designated as the right-hand (R) wheels. Roller angle ψ = 45° on the left-side wheels and ψ = -45° on the right-hand wheels. As a result, the usual design of a four-wheel omnidirectional WMR is depicted in Figure 1.15.

Figure 1.16 illustrates the six fundamental movements performed by a four-wheel ODR, specifically denoted as (A) forward movement, (B) left shifting, (C) clockwise rotating (in place), (D) backward movement, (E) right shifting, and (F) anticlockwise rotating. The arrows located on both the

left and right sides of the vehicle serve to indicate the intended direction of motion for the respective wheels.

The arrows depicted on the automobile platform indicate the corresponding directions of motion for the WMR. Specifically, when the automobile is moving ahead, all wheels are required to travel in the forward direction, as shown in Figure 1.16A. In the case of left shifting, wheels 1 and 3 move forward while wheels 2 and 4 move backward, and so forth. The depicted locomotions in Figure 1.16 happen when all wheels are in motion at an identical velocity. The ability to achieve movement in any orientation on a two-dimensional plane using a WMR can be accomplished by adjusting the amount of the wheel speeds. Several instances are depicted in Figure 1.17.

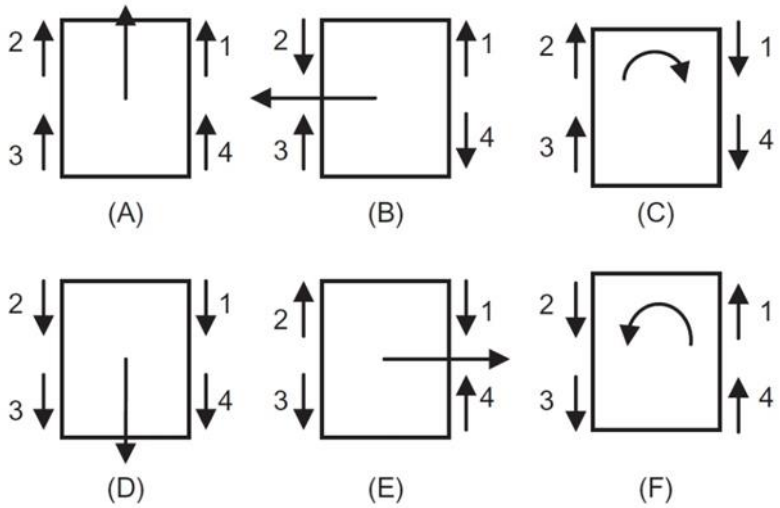

Figure 1.16. Six fundamental movements performed by a four mecanum wheels ODR [44]

The various movements observed in Figures 1.16 and 1.17 are easily elucidated by referring to the velocity or force diagrams depicted in Figure 1.6 A and B, respectively. For instance, due to the symmetrical configuration of the wheels on both sides (see Figure 1.15), when all wheels move in the forward direction, there exist four forward-pointing vectors that are combined, along with four sideways-pointing vectors—two towards the right and two towards the left—that mutually nullify each other. Therefore, altogether, the WMR demonstrates progress. The left (L) and right (R) wheels have the potential to be exchanged, meaning that the front wheels can be placed between each other as well as the back wheels. Furthermore, it should be noted that by incorporating the appropriate motion of the wheels, it is possible to achieve several kinds of omnidirectional mobility with this particular configuration.

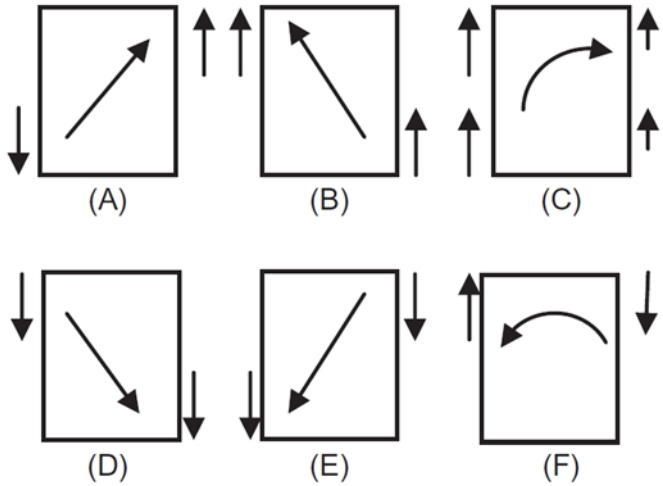

Figure 1.17. Six more movements: (A) forward-right, (B) forward-left, (C) curved right, (D) backward-right, (E) backward-left, and (F) lateral arc [44]

1.5. WMR Maneuverability

The maneuverability of WMRs, denoted as M_w, can be mathematically expressed as

$$M_w = G_m + G_s \tag{1.3}$$

where G_m represents the mobility degree, and G_s represents the steerability degree.

The mobility degree: The mobility degree the value of G_m is contingent upon the number of separate (independent) constraints that are imposed on the robot's motion capability by the kinds of wheels and configuration they have. The motion of the system is solely constrained by the presence of conventional wheels, whether they are fixed or steered. The utilization of omnidirectional wheels does not put any constraints on the mobility of the robot. A comprehensive understanding of the separate (independent) kinematic constraints concerning a wheeled mobile robot (WMR) can be achieved by examining the geometric characteristics of the robot, specifically focusing on the Instantaneous Center of Curvature (ICC) or Instantaneous Center of Rotation (ICR). As an illustration, it is worth noting that one conventional wheel lacks the capability to execute lateral movement, specifically along the line defined by its rotational axis. The line mentioned above is commonly referred to as the zero-motion line of the wheel. This implies that the wheel is limited to traversing an instantaneous circular path of radius R_{ai}, where the center of this circle is positioned on the zero-motion line. A bicycle is composed of two wheels: the front wheel, which is steered, and the rear wheel, which remains fixed (see Figure 1.18).

Every wheel produces its own distinct zero motion line, which operates independently. The intersection of the two lines occurs at the Instantaneous Center of Rotation (ICR). In the scenario of a differential drive-wheeled mobile robot (DDWMR), as depicted in Figure 1.19, it can be observed that the zero motion lines of the two wheels, which share a common axis, synchronize with each other.

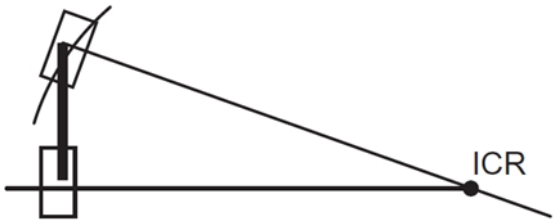

Figure 1.18. A bicycle's two wheels represent two independent constraints [44]

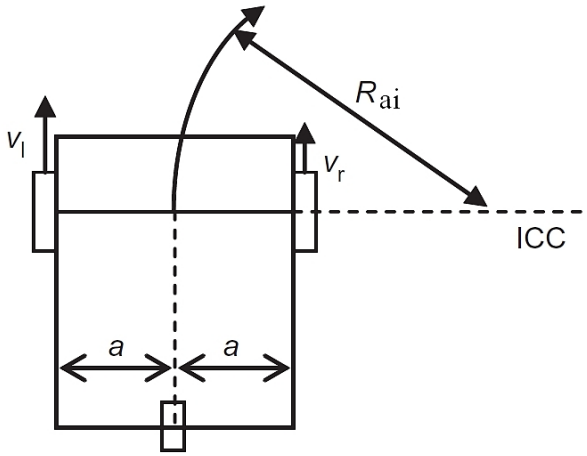

Figure 1.19. Determination of the WMR rotation's instantaneous radius R_{ai} [44]

Consequently, the motion of these wheels is not independent. This implies that there exists a single independent kinematic constraint. Any point located along the common zero motion line has the potential to serve as an Instantaneous Center of Rotation (ICR). The Ackerman steering mechanism is characterized by the presence of four conventional wheels on a wheeled mobile robot (WMR), with two separate (independent) kinematic constraints, as depicted in Figure 1.12. The presence of two rear wheels in a vehicle, such as in a differential drive system, introduces a singular constraint.

Additionally, the two front-steered wheels present a second singular kinematic constraint. This is due to the fact that these wheels intersect at an Instantaneous Center of Rotation (ICR), which lies on the zero-motion line specified by the common axis of the rear wheels. The highest mobility degree G_m is equal to 3 in cases if nonkinematic constraints are present. This scenario occurs when every one of

the wheels of the wheeled mobile robot (WMR) possesses omnidirectional capabilities. In a broad sense, the mobility degree can be considered as being equivalent to:

$$G_m = 3 - N_c \tag{1.4}$$

where N_c represents the number of separate (independent) constraints.

The steerability degree, G_s, is determined by the number of steering parameters that can be controlled separately (independently). The range of G_s is bounded by the period $0 \leq G_s \leq 2$. In the absence of any wheels capable of being steered, the value of G_s is equal to zero. The condition $G_s = 2$ is satisfied just in instances where the robot does not possess any fixed standard wheels. In this scenario, it is possible to implement a platform using two distinct steerable conventional wheels, such as those found in a two-steer bicycle or a three-wheeled two-steer wheeled mobile robot (WMR). According to the above information, a value of $G_s = 2$ indicates that the WMR (Wheeled Mobile Robot) possesses the capability to position its Instantaneous Center of Rotation (ICR) at any location inside the plane. The prevailing scenario occurs when G_s equals 1, a condition that arises when the robot's configuration incorporates at least one steerable conventional wheel. The utilization of a conventional wheel that is guided has the potential to reduce the overall mobility of the robot while simultaneously increasing its steerability. Indeed, while the immediate orientation of the steering wheel enforces a kinematic constraint, its capacity to modify the orientation may enable the exploration of supplementary trajectories. Table 1.1 displays the maneuverability (M_w), mobility degree (G_m), and steerability degree (G_s) for various common configurations of WMRs.

Two further distinguishing parameters of WMRs are the "differential degrees of freedom" ($DDOF$) and the "degrees of freedom" (DOF), which are related by the following relation:

$$DDOF \leq M_w \leq DOF \tag{1.5}$$

A bicycle possesses the capability to attain any given position (x, y, φ) within a plane through a series of maneuvers, so indicating that it possesses three degrees of freedom, $DOF = 3$. However, its differential degrees of freedom, $DDOF = G_m = 1$. The omnirobot under consideration possesses three omnidirectional wheels, resulting in $G_m = 3$, indicating a $DDOF = 3$. Additionally, the omnirobot manifests a DOF = 3. In a similar vein, it can be observed that a tricycle possesses a differential degree of freedom $DDOF = G_m = 1$, and a degree of freedom $DOF = 3$. This is due to its capability to attain any desired position (x, y, φ) with suitable maneuvering [44].

Table 1.1. Mobility Degree and Steerability Degree (G_m, G_s) of popular WMRs [44]

Configuration	$\boldsymbol{G_m}$	$\boldsymbol{G_s}$	$\boldsymbol{M_w}$	**Notation**
Bicycle	1	1	2	(1,1)
Differential drive	2	0	2	(2,0)
Synchro drive	1	1	2	(1,1)
Tricycle	1	1	2	(1,1)
Ackerman steer	1	1	2	(1,1)
Two-steer	1	2	3	(1,2)
Omni-steer	2	1	3	(2,1)
Omnidirectional	3	0	3	(3,0)

1.6. WMR Stability

Interestingly, it has been found that a minimum of two wheels is sufficient to achieve static stability. A differential-drive robot with two wheels can attain static stability when the center of mass is positioned below the axle of the wheels [59]. Nevertheless, in typical scenarios, the implementation of such a resolution necessitates wheel widths that are excessively big and, therefore, not feasible. The presence of dynamics in a two-wheeled robot can result in the robot making contact with the floor at a third point, such as when there are strong motor torques applied from a stationary position. In accordance with conventional wisdom, static stability necessitates the presence of at least three wheels. It is important to note that the gravity center is required to be situated within the triangular region produced by the points of contact between the wheels and the ground. Enhancing stability can be achieved by increasing the number of wheels. However, it should be noted that when the number of connecting points surpasses three, the geometric configuration becomes hyperstatic, necessitating the implementation of a flexible suspension system to accommodate uneven terrain [42].

1.7. WMR Controllability

In general, there exists a negative link between controllability and maneuverability. An instance of this can be seen in omnidirectional designs, such as the configuration with four castor wheels, which necessitates substantial processing in order to turn the required rotational and linear velocities into specific orders for each wheel. Moreover, it is worth noting that omnidirectional designs frequently exhibit a higher number of degrees of freedom in the wheel mechanism. As an illustration, the Swedish wheel is equipped with a collection of unrestricted rollers positioned along the circumference of the wheel. The presence of the degrees mentioned above of freedom leads to the occurrence of slippage, which has a detrimental effect on the accuracy of dead-reckoning and also contributes to an increase in the complexity of the design.

The task of directing an omnidirectional robot towards a specific direction of motion is inherently more challenging and frequently exhibits lower levels of accuracy in comparison to less maneuverable robot designs. As an illustration, a vehicle equipped with an Ackerman steering mechanism is capable of maintaining a straight trajectory by immobilizing the steerable wheels while powering the drive wheels. In the context of a differential-drive vehicle, it is imperative to ensure that the two motors connected to the two wheels are driving at an identical velocity profile. However, achieving this synchronization can pose difficulties due to inherent variations among the wheels, motors, and the outside environment. The challenge becomes more complex when utilizing a four-wheel omni-drive system, exemplified by the Uranus robot equipped with four Swedish wheels. In this configuration, maintaining a precise straight trajectory necessitates the synchronization of all four wheels to operate at identical speeds [42].

In conclusion, it can be stated that no universally optimal drive design may effectively optimize stability, maneuverability, and controllability all at once. Every mobile robot application imposes specific constraints on the design challenge, and the designer's objective is to select the best suitable drive configuration among the range of trade-offs available.

1.8. Motion Modeling for Differential Drive Wheeled Mobile Robots

Motion models are utilized to describe the kinematics of robots. There has been a notable focus on the mathematical aspects of robot motion, disregarding the underlying causes like forces or torques. The kinematic model elucidates the inherent geometric relationships within the system. This statement

elucidates the correlation between the inputs (control parameters) and the behavior of a system as delineated by its state-space representation. The kinematic model pertains to the velocities of a system and is represented by a collection of differential equations of the first degree.

Dynamic models are utilized to depict the motion of a system in response to the application of forces. These models incorporate the principles of physics pertaining to motion, encompassing the utilization of forces, energy, mass of a system, velocity, and inertia parameters. The dynamic models' description can be expressed by second-order differential equations.

In the field of wheeled mobile robotics, it is well-accepted that kinematic models are typically adequate for the purpose of designing locomotion strategies. However, in the case of other systems involving robots operating in air, space, water, or walking robots, the inclusion of dynamic modeling becomes necessary [60].

As previously said, the Differential Drive Wheeled Mobile Robot (DDWMR) is considered to be a straightforward and efficient structure among the various types of mobile robots [61]. The DDWMRs possess the capability to navigate inside a predetermined operational environment in order to accomplish a specified path or trajectory [62]. The capacity for mobility renders them highly advantageous for a wide range of applications in both structured and non-structural environments. The differential drive system consists of a pair of wheels positioned at opposite ends of a mobile platform. These wheels can be operated independently in terms of both position and velocity [63]. In certain instances, the implementation of an additional wheel known as the Castor wheel can be used to maintain equilibrium in the event of any potential instability [64]. Various scenarios occur during the DDWMR rotation. When the two wheels of the DDWMR rotate in the same direction at equal speeds, the robot travels in a straight trajectory [65]. Additionally, when one wheel is in motion while the other remains stationary, the DDWMR exhibits circular motion, with the center of the circle being the pivotal point of the stationary wheel. Similarly, if the roles are reversed, the DDWMR follows a circular trajectory, with the center being the pivotal point of the rotating wheel [66].

The classification of Differential Drive Wheeled Mobile Robots (DDWMRs) encompasses four distinct models:

- The posture kinematic model
- The configuration kinematic model
- The posture dynamic model
- The configuration dynamic model

The kinematic models of the DDWMR characterize its behavior through a mathematical function that relates the velocity and orientation of its wheels. On the other hand, the dynamic models of the DDWMR describe its behavior by a mathematical function that relates the generalized forces exerted by the actuators. The posture models exclusively focus on the robot's position and orientation as state variables, in contrast to configuration models that incorporate other internal variables, such as the wheels' angular displacement [67-69].

1.8.1. Kinematics of Differential Drive Wheeled Mobile Robots

The field of robot kinematics encompasses the study of the configuration (arrangement) of robots within their operational environment, Robot kinematics encompasses the analysis of the geometric configuration of robotic systems, the functional dependencies between their structural parameters, and the kinematic constraints imposed on their trajectories—all of which are fundamentally determined by the robot's physical architecture. The choice of wheel type, the number of wheels, and the manner in which they are connected to the chassis of the robot have a substantial impact on the kinematics of mobile robots [70].

A comprehensive understanding of kinematics is an essential foundation for studying the principles of dynamics, the analysis of stability characteristics, and the implementation of control mechanisms in the field of robotics. Ongoing research is being conducted on the development of novel and special robotic kinematic structures, with the aim of creating robots capable of executing advanced and intricate tasks in various industrial and societal domains [71-76].

The nonholonomic mechanical system, known as the DDWMR, as depicted in Figure 1.20, serves as a representative illustration. The system under consideration comprises a rigid body, referred to as the base, which incorporates a pair of conventional fixed wheels that are driven by separate actuators, such as direct current motors. These wheels enable the system to achieve both movement and orientation. Additionally, a third wheel is present, and occasionally a fourth wheel, which are passive and solely serve the purpose of providing support to the DDWMR. The influence of these passive wheels on the dynamics of the DDWMR is considered to be trivial [77].

The posture vector, denoted as $\zeta = [x\ y\ \theta]^T$, represents the characteristics of the system. Here, x and y denote the coordinates of point C, which serves as both the mass center as well as the guidance point. These coordinates are defined within the inertial coordinate system OX_OY_O. Additionally, θ represents the angle of orientation of the mass center coordinate system of the DDWMR CX_CY_C relative

to the inertial coordinate system OX_OY_O. The OX_OY_O frame, referred to as the inertial reference frame, also represents the fixed frame of reference in the robot's surroundings within which it operates. On the other hand, the DDWMR CX_CY_C frame, known as the robot frame, denotes a local coordinate system that is affixed to the robot itself [78].

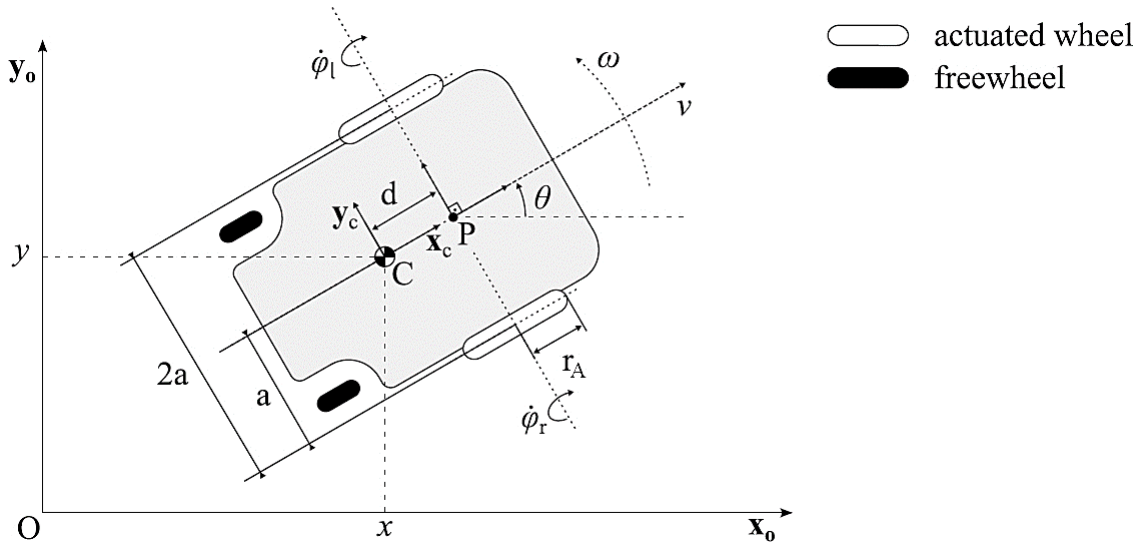

Figure 1.20. DDWMR and systems of coordinates [77]

The local coordinates of mechanical systems are capable of being represented using the generalized coordinate vector $\boldsymbol{Q}$, where $\boldsymbol{Q} = [Q_1, Q_2, \ldots, Q_n]^T \in \Re^n$. In numerous scenarios, the motion of mechanical systems is governed by a range of constraints that are consistently upheld throughout the movement. These constraints manifest as algebraic relationships between the velocities and positions of the system's points [79].

The parameter nomenclature employed in the DDWMR study, as depicted in Figure 1.20 and enumerated in Table 1.2.

The DDWMR depicted in Figure 1.20 exhibits three kinematic restrictions, as documented in references [80-82]. The initial constraint pertains to the inability of the DDWMR to undergo lateral sliding, thereby adhering to a non-slipping constraint. Consequently, the DDWMR is only capable of movement along the perpendicular direction to the actuated wheels' symmetry axis. The constraint mentioned above can be expressed as

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) = 0, \quad \text{for C = P,} \tag{1.6}$$

$$\dot{y}\cos(\theta) - \dot{x}\sin(\theta) - d\dot{\theta} = 0, \quad \text{for C} \neq \text{P.} \tag{1.7}$$

The next two constraints pertain to the wheel's rotation, namely the pure rolling constraints.

Table 1.2. DDWMR Parameters [77]

Parameter	Description
P	Intersection of the axis of the symmetry with the wheels' axis.
C	Center of mass or point of guidance.
d	The distance between point P and point C.
r_A	Radius of left or right wheel.
2a	The distance between the actuated wheels and the axis of symmetry.
$\dot{\varphi}_l, \dot{\varphi}_r$	The angular velocities of the left and right wheels.
ω, v	The angular and linear velocities of DDWMR.
$\boldsymbol{Q}$	The generalized coordinate vector.

These constraints ensure that the actuated wheels do not experience any incorrect rotation. They can be expressed as follows:

$$\dot{x}\cos(\theta) + \dot{y}\sin(\theta) + a\dot{\theta} - r\dot{\varphi}_r = 0, \quad \text{for C = P and C} \neq \text{P}, \tag{1.8}$$

$$\dot{x}\cos(\theta) + \dot{y}\sin(\theta) - a\dot{\theta} - r\dot{\varphi}_l = 0, \quad \text{for C = P and C} \neq \text{P}. \tag{1.9}$$

where φ_l and φ_r represent angular displacements of the left and right wheels, respectively.

Equations (1.8) and (1.9) can be expressed as follows:

$$v + a\omega = r\dot{\varphi}_r, \quad \text{for C = P and C} \neq \text{P}, \tag{1.10}$$

$$v - a\omega = r\dot{\varphi}_l, \quad \text{for C = P and C} \neq \text{P}, \tag{1.11}$$

since

$$v = \dot{x}\cos(\theta) + \dot{y}\sin(\theta), \tag{1.12}$$

$$\omega = \dot{\theta}. \tag{1.13}$$

By utilizing Equations (1.10) and (1.11), it is possible to establish a correlation between the angular velocities of the right and left wheels (φ_r, φ_l) of the DDWMR, and the angular and linear velocities of the mass center of the DDWMR (ω, υ). This correlation yields the following relationship:

$$\begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{a}{r} \\ \frac{1}{r} & \frac{-a}{r} \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix}, \tag{1.14}$$

and vice versa:

$$\begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = \boldsymbol{\Omega}^{-1} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix} = \begin{bmatrix} \frac{2}{r} & \frac{a2}{r} \\ b & -b \end{bmatrix} \begin{bmatrix} \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix}, \tag{1.15}$$

where $b = \frac{r}{2a}$.

Kinematic constraints are prevalent in a wide range of applications. Equations (1.6)–(1.9) represent linear relationships involving the generalized coordinate vector. These relationships can be described in matrix format as:

$$\boldsymbol{A}(\boldsymbol{Q})\dot{\boldsymbol{Q}} = 0. \tag{1.16}$$

The state vector is denoted by a set of five generalized coordinates,

$$\boldsymbol{Q} = [\boldsymbol{\zeta}^T \boldsymbol{\varphi}^T]^T = [x\ y\ \theta\ \varphi_r\ \varphi_l]^T, \tag{1.17}$$

the three constraints are able to be expressed in the format of Equation (1.16), i.e.,

$$\boldsymbol{A}(\boldsymbol{Q})\dot{\boldsymbol{Q}} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & 0 & 0 & 0 \\ -\cos(\theta) & -\sin(\theta) & -a & r & 0 \\ -\cos(\theta) & -\sin(\theta) & a & 0 & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix}, \quad \text{for C = P,} \tag{1.18}$$

$$\boldsymbol{A}(\boldsymbol{Q})\dot{\boldsymbol{Q}} = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & -d & 0 & 0 \\ -\cos(\theta) & -\sin(\theta) & -a & r & 0 \\ -\cos(\theta) & -\sin(\theta) & a & 0 & r \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi}_r \\ \dot{\varphi}_l \end{bmatrix}, \quad \text{for C} \neq \text{P,} \tag{1.19}$$

The referential of DDWMR velocity is determined by the angular velocity of the left and right wheels ($\dot{\varphi}_l$ and $\dot{\varphi}_r$), respectively,

$$\boldsymbol{v} = \begin{bmatrix} \dot{\varphi}_l \\ \dot{\varphi}_r \end{bmatrix}. \tag{1.20}$$

By assuming neglect of the inertia and mass of the motors and wheels, it may be postulated that the DDWMR adheres to the principles of pure rolling and non-slipping conditions [83]. Therefore, the matrix $\boldsymbol{A}(\boldsymbol{Q})$ that encompasses the nonholonomic constraints is simplified to:

$$\boldsymbol{A}(\boldsymbol{Q}) = [-\sin(\theta)\ \cos(\theta)\ \ 0], \quad \text{for C = P,} \tag{1.21}$$

$$\boldsymbol{A}(\boldsymbol{Q}) = [-\sin(\theta)\ \cos(\theta)\ -\ d], \quad \text{for C} \neq \text{P,} \tag{1.22}$$

in such a way that the displacements are limited to the direction of the axis of symmetry of the actuated wheels and

$$\boldsymbol{Q} = \boldsymbol{\zeta} = [x\ y\ \theta]^T. \tag{1.23}$$

Furthermore, it should be noted how the referential velocity of the DDWMR is determined by its linear velocity (υ) and angular velocity (ω), i.e.,

$$\boldsymbol{v} = \begin{bmatrix} \upsilon \\ \omega \end{bmatrix}. \tag{1.24}$$

It is crucial to highlight that the system's configuration space, denoted as n, consists of the generalized coordinate vector $\boldsymbol{Q}$ and the number of constraints represented by p. Consequently, the dimension of the velocity vector is m = n - p, where in this particular case, m = 2, indicating the system's degrees of freedom.

The objective is to eliminate the limitations imposed by the constraints [79] on the Jacobian matrix $\boldsymbol{S}(\boldsymbol{Q})$. This matrix, consisting of a collection of linearly independent and smooth vector fields, is of complete rank (n - p) and is dispersed inside the null space of $\boldsymbol{A}(\boldsymbol{Q})$, i.e.,

$$\boldsymbol{A}(\boldsymbol{Q})\boldsymbol{S}(\boldsymbol{Q}) = 0. \tag{1.25}$$

Based on Equations (1.16) and (1.25), it is feasible to determine an auxiliary velocity vector, denoted as $\boldsymbol{v} \in \Re^{p\times 1}$, which is dependent on time. This vector satisfies the condition for all values of t:

$$\dot{\boldsymbol{Q}} = \boldsymbol{S}(\boldsymbol{Q})\boldsymbol{v}\,. \tag{1.26}$$

The configuration kinematic model, denoted as matrix $\boldsymbol{S}(\boldsymbol{Q})$, is provided as follows:

- From Equations (1.18) and (1.19):

$$\boldsymbol{S}(\boldsymbol{Q}) = \begin{bmatrix} b\ a\cos(\theta) & b\ a\cos(\theta) \\ b\ a\sin(\theta) & b\ a\sin(\theta) \\ b & -b \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{for C = P,} \tag{1.27}$$

$$\boldsymbol{S}(\boldsymbol{Q}) = \begin{bmatrix} b(a\cos(\theta) - d\sin(\theta)) & b(a\cos(\theta) + d\sin(\theta)) \\ b(a\sin(\theta) + d\cos(\theta)) & b(a\sin(\theta) - d\cos(\theta)) \\ b & -b \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{for C} \neq \text{P}, \qquad (1.28)$$

where $b = \frac{r}{2a}$.

In a similar vein, the posture kinematic model, represented by the matrix $\boldsymbol{S}(\boldsymbol{Q})$, leads to:

- From equations. (1.21) and (1.22):

$$\boldsymbol{S}(\boldsymbol{Q}) = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix}, \qquad \text{for C = P}, \qquad (1.29)$$

$$\boldsymbol{S}(\boldsymbol{Q}) = \begin{bmatrix} \cos(\theta) & -d\sin(\theta) \\ \sin(\theta) & \text{d}\cos(\theta) \\ 0 & 1 \end{bmatrix}, \qquad \text{for C} \neq \text{P}, \qquad (1.30)$$

An alternative approach to expressing the posture kinematic model Equations (1.29) and (1.30) involves leveraging the velocity of the DDWMR in terms of $\dot{x}$, $\dot{y}$, and $\dot{\theta}$ as depicted in Figure (1.19), i.e.,

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \omega \end{cases}, \qquad \text{for C = P}, \qquad (1.31)$$

$$\begin{cases} \dot{x} = v\cos(\theta) - \omega d\sin(\theta) \\ \dot{y} = v\sin(\theta) + \omega d\cos(\theta) \\ \dot{\theta} = \omega \end{cases}, \qquad \text{for C} \neq \text{P}. \qquad (1.32)$$

It is imperative to underscore the necessity of conducting an investigation into the correlation between the wheels and the DDWMR in order to ascertain the matrices $\boldsymbol{A}(\boldsymbol{Q})$ and $\boldsymbol{S}(\boldsymbol{Q})$. The method for deducing these matrices has been analyzed in [67-69].

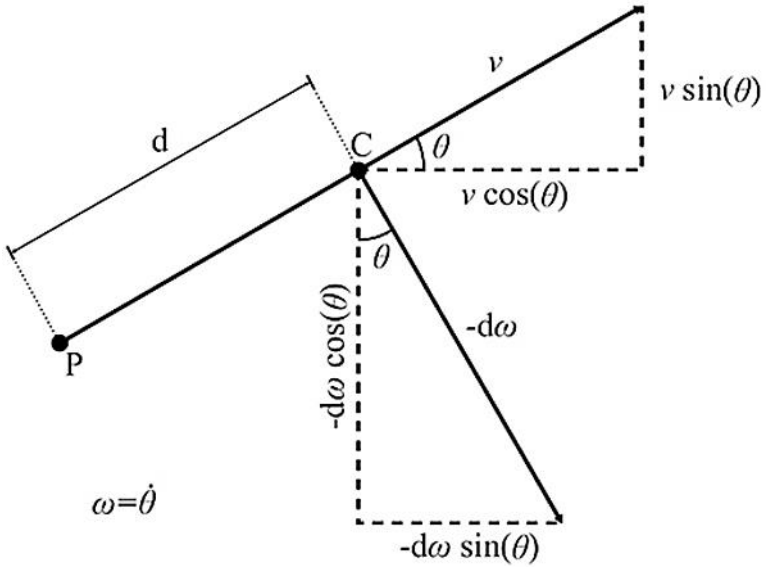

Figure 1.21. The depiction of the DDWMR velocities of Figure 1.20. [77]

1.9. Motion Constraints

All robotic systems are bound by different constraints on their motion, but not each of these is capable of being articulated as constraints on their configuration. An illustrative instance of this kind of system involves an automobile. At lower velocities, both automobile rear wheels exhibit unrestricted rotational movement in the direction they are oriented while impeding any lateral sliding motion in the direction perpendicular to them. This constraint indicates that the automobile is unable to move laterally. It has been empirically observed that the velocity constraint lacks any constraints on the configurations of the automobile. In other words, the automobile has the ability to attain any location or orientation inside the plane that does not contain obstacles. Indeed, the hindered lateral displacement can be estimated through the execution of parallel parking maneuvers.

The no-slip constraint can be classified as a nonholonomic constraint, which specifically pertains to the velocity of the system. Besides the condition of rolling without slipping, the principle of conservation of angular momentum is frequently encountered as a prevalent origin of nonholonomic constraints with mechanical systems.

Suppose we shift our perspective from perceiving the automobile as a system that adheres to a motion constraint. Instead, we acknowledge that merely two control inputs (the speed and steering angle) are available to govern the automobile's three degrees of freedom. In that case, it is plausible to categorize the system as underactuated. Underactuated systems are characterized by a fewer number of controls than the present degrees of freedom [84].

Kinematic constraints are able to be classified into two categories: holonomic constraints and nonholonomic constraints. The mobility of a robot is restricted in some directions due to nonholonomic constraints [85]. The concept of holonomic constraints is closely linked to the dimensionality associated with the system's state description, specifically in generalized coordinates.

Nonholonomic constraints manifest in two principal fashions [15]:

1. In the context of rolling motion without slipping constraints. An instance that illustrates the interdependence of translation and rotation occurs when a wheel undergoes rolling motion without sliding. Some examples that can be provided are a WMR, a unicycle, a vehicle, and a tractor-trailer.
2. In systems characterized by the conservation of angular momentum. Some examples of robotic applications include the utilization of satellites and space robots, as well as the development of robots for gymnastics, diving, and running.

The expression of holonomic constraints is achieved by the utilization of equations that incorporate generalized coordinates. The equations mentioned above can be employed to exclude a subset of generalized coordinates, reducing the number of necessary generalized coordinates for describing a given system. Nonholonomic constraints have no effect on the reduction of the dimensionality of the generalized coordinates; instead, they only affect the dimensionality of the generalized velocity space. The inclusion of nonholonomic constraints has a significant impact on the problem of path planning [60], [86].

The problem of motion planning for a nonholonomic system can be formulated as follows: given a representation of the environment containing obstacles in the workspace, a robot that is constrained by nonholonomic constraints, the initial position, and the final position, the objective is to determine a feasible path that is free from collisions between the initial and final positions. The resolution of this issue necessitates the consideration of both the constraints imposed by obstacles inside the configuration space as well as the nonholonomic constraints. The methods that have been created to tackle this issue effectively integrate techniques from both motion planning as well as control theory. Constraints arising from obstacles are explicitly represented in the configuration space, which is a manifold. However, nonholonomic constraints are described within the tangent space [87-91].

1.9.1. Holonomic Constraints

The holonomic constraints are contingent upon the utilization of generalized coordinates. In a system characterized by n generalized coordinates $\boldsymbol{Q} = [Q_1, \dots, Q_n]^T$, a holonomic constraint can be mathematically represented as follows:

$$f(\boldsymbol{Q}) = f(Q_1, \dots, Q_n) = 0, \tag{1.33}$$

where f and its associated derivatives are assumed to be continuous functions, this constraint establishes a subspace within the set of all the possible configurations in the generalized coordinates, wherein Equation (1.33) holds valid. Constraint (1.33) is capable of being employed to eliminate specific generalized coordinates, as it can be represented in terms of n − 1 other coordinates.

Typically, there exists a possibility of having m holonomic constraints, where (m < n). If the mentioned constraints are linearly independent, they establish a subspace of dimension (n − m), which corresponds to the genuine configuration space of the system with (n − m) degrees of freedom.

1.9.2. Nonholonomic Constraints

Nonholonomic constraints impose limitations on the possible system velocities or the possible motion directions. The formulation of the nonholonomic constraint is capable of being expressed by

$$f\left(\boldsymbol{Q}, \dot{\boldsymbol{Q}}\right) = f\left(Q_1, \dots, Q_n, \dot{Q}_1, \dots, \dot{Q}_n\right) = 0, \tag{1.34}$$

where f represents a smooth function possessing continuous derivatives, whereas $\dot{\boldsymbol{Q}}$ is the vector containing the velocities of the system within the generalized coordinates. If the system lacks constraints (1.34), it is unconstrained in its range of motion directions.

A kinematic constraint (1.34) becomes holonomic in the academic context if it satisfies the condition of integrability. Integrability implies that the velocities $\dot{Q}_1, \dots, \dot{Q}_n$ are able to be omitted from Equation (1.34), resulting in the constraint being represented in the format of Equation (1.33). If the constraint denoted by equation (1.34) lacks integrability, it can be classified as nonholonomic.

If there are m nonholonomic constraints in the form of Equation (1.34) that are linearly independent, then the dimension of the velocity space becomes (n − m). The system's velocities are constrained by nonholonomic constraints. An instance of a differential drive vehicle, such as a wheelchair, has the capability to travel in the direction determined by the current orientation of its wheels, but lacks the ability to move laterally.

Considering linear constraints in the equation $\dot{\boldsymbol{Q}} = [\dot{Q}_1, \dots, \dot{Q}_n]^T$, equation (1.34) is able to be expressed as

$$f\left(\boldsymbol{Q}, \dot{\boldsymbol{Q}}\right) = \boldsymbol{a}^T(\boldsymbol{Q})\dot{\boldsymbol{Q}} = [a_1(\boldsymbol{Q}) \ \dots \ a_n(\boldsymbol{Q})] \begin{bmatrix} \dot{Q}_1 \\ \vdots \\ \dot{Q}_n \end{bmatrix} = 0, \tag{1.35}$$

where $\boldsymbol{a}^T(\boldsymbol{Q})$ represents the parameter vector of the constraint. To obtain a constraint matrix for a system with m nonholonomic constraints, the following matrix is employed:

$$\boldsymbol{A}(\boldsymbol{Q}) = \begin{bmatrix} \boldsymbol{a}_1^T(\boldsymbol{Q}) \\ \vdots \\ \boldsymbol{a}_m^T(\boldsymbol{Q}) \end{bmatrix}, \tag{1.36}$$

and all of nonholonomic constraints are presented in matrix representation

$$\boldsymbol{A}(\boldsymbol{Q})\dot{\boldsymbol{Q}} = \boldsymbol{0}. \tag{1.37}$$

At every given time interval, the matrix denoting the set of attainable motion directions is represented as $\boldsymbol{S}(\boldsymbol{Q}) = [\boldsymbol{s}_1(\boldsymbol{Q}), \dots, \boldsymbol{s}_{n-m}(\boldsymbol{Q})]$. The number of constraints determines the number of attainable directions, which is equal to (n-m). The kinematic model is defined by this matrix in the following manner:

$$\dot{\boldsymbol{Q}}(t) = \boldsymbol{S}(\boldsymbol{Q})\boldsymbol{v}(t), \tag{1.38}$$

where $\boldsymbol{v}(t)$ represents the control vector. The resultant matrix obtained by multiplying the constraint matrix $\boldsymbol{A}$ and the kinematic matrix $\boldsymbol{S}$ is a matrix consisting entirely of zeros

$$\boldsymbol{AS} = \boldsymbol{0}\ . \tag{1.39}$$

The idea of holonomic motion is defined by the relative values of a robot's degree of freedom (DOF) and the mobility degree (G_m). A robot exhibits holonomic motion when the mobility degree is equal to degrees of freedom ($G_m = DOF$). Conversely, a robot demonstrates nonholonomic motion when the mobility degree is less than the degrees of freedom ($G_m < DOF$). A holonomic robot, such as the omni robot, possesses the capability to exert direct control over all of its degrees of freedom (DOF) without necessitating complex maneuvers. Figure 1.23 illustrates the relative ease with which the omnidirectional robot equipped with three Swedish wheels, as depicted in Figure 1.22, is able to execute parallel parking.

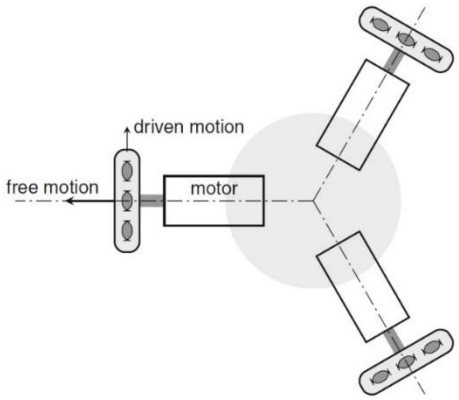

Figure 1.22. An omni-directional robot equipped with three Swedish wheels [92]

An automobile and a robot equipped with differential drive belong to non-holonomic systems due to their limited mobility degree (G_m). The automobile has a mobility degree of 1, while the robot has a mobility degree of 2. This is lower than their degrees of freedom (DOF), which is 3 for both systems. Due to this restricted mobility degree, these vehicles necessitate intricate steering maneuvers, such as those employed during parallel parking. A notable disparity exists between both of these vehicles.

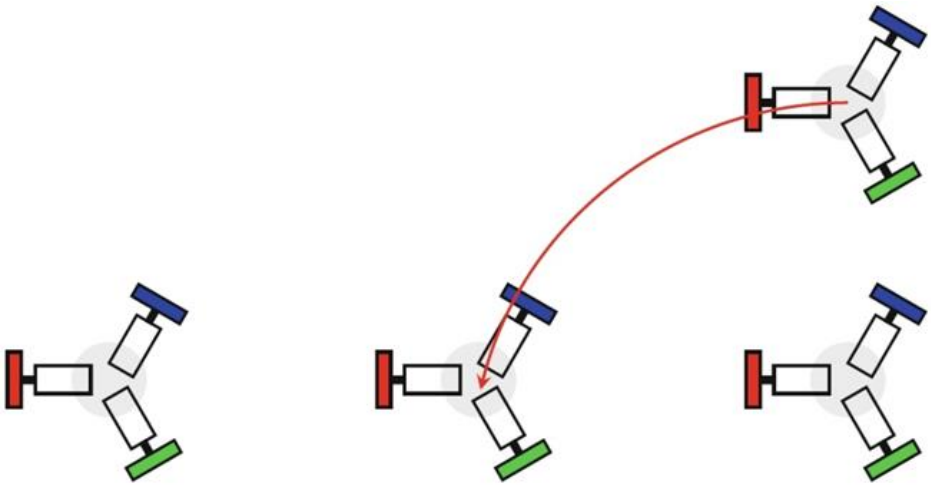

Figure 1.23. Parallel parking maneuvers by an omni-directional robot [92]

The DDWMR requires three distinct movements, which are characterized by simplicity: left rotation, backward motion, and right rotation, as depicted in Figure 1.24a. The automobile also necessitates three distinct motions, albeit executing them accurately proves exceedingly challenging (see Figure 1.24b). One must determine the optimal initiation point of the maneuver, the degree of curvature for each turn, and the distance to be covered between successive turns. The greater mobility degree (G_m) of the DDWMR provides a notable benefit in the given scenario [92].

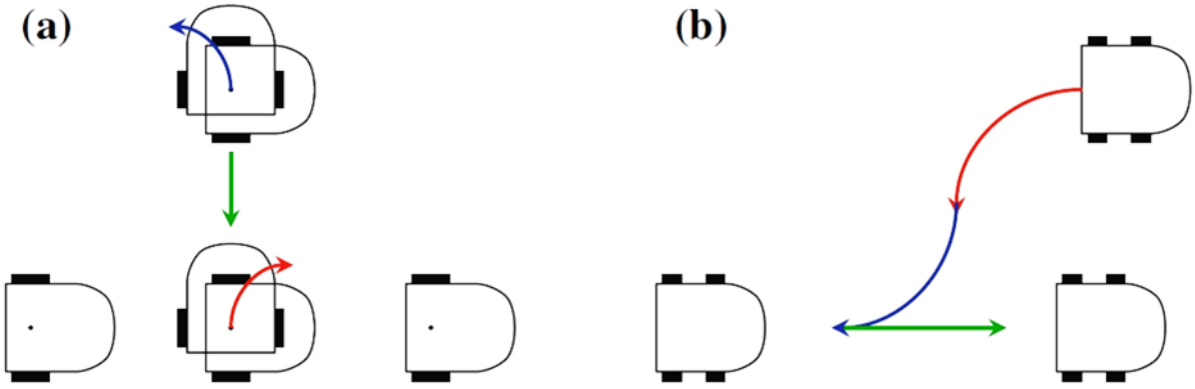

Figure 1.24. (a) Non-holonomic DDWMR parallel parking. (b) Non-holonomic automobile parallel parking [92]

1.10. Navigation of WMR

The process of robot navigation involves deliberate design strategies aimed at reaching a specified destination while simultaneously circumventing any encountered obstacles. The principal objective in the field of navigation involves either attaining a pre-established objective or following a predefined path while avoiding any instances of collision. A mobile robot possesses the capability to move intelligently throughout a wide range of environments, including static, dynamic, uncluttered, and unpredictable settings, among others [93]. Navigation is a fundamental methodology employed to facilitate the movement of a robotic entity across diverse environments, enabling it to traverse from an initial position to a desired destination [94]. The dependability of maps in navigational approaches is often called into question as a result of the dynamic and unexpected nature of applications in the real world [95]. The process depicted in Figure 1.25 [96-100] comprises four fundamental components: (i) perception, the system of perception refers to the ability of a robot to identify objects in its surrounding environment in real-time. This information is then transmitted to the decision-making system, which enables the robot to effectively reason concerning future actions necessary to accomplish the intended task [101]; (ii) localization, on the other hand, pertains to the robot's capability to accurately determine its precise position within a given map in the real world [102]; (iii) cognition and path planning, involve the process of determining a path that avoids collisions and optimizes specific objectives, such as minimizing the distance navigated or energy consumption, from an initial location to a desired goal location [103-104]; (iv) motion control, refers to the robot's ability to adjust its motor output in order to reach the intended route [105]. Furthermore, the successful navigation of a mobile robot necessitates the acquisition of supplementary skills, encompassing control aptitude, planning of trajectory, obstacle avoidance, and the establishment of secure distances to the intended destination. These competencies are imperative for mobile robots to execute optimal navigation performance. In order to ensure the successful completion of all tasks, it is essential for every navigation system to take into account those mentioned above basic design [106-107]. The navigation problem, as a whole, has been constructed based on the suitable answer to three fundamental questions. In which location am I currently situated? To what destination am I headed? Furthermore, what is the most efficient approach for reaching that destination? The fundamental philosophy of all research within this discipline is to provide answers to these three fundamental questions [108-109].

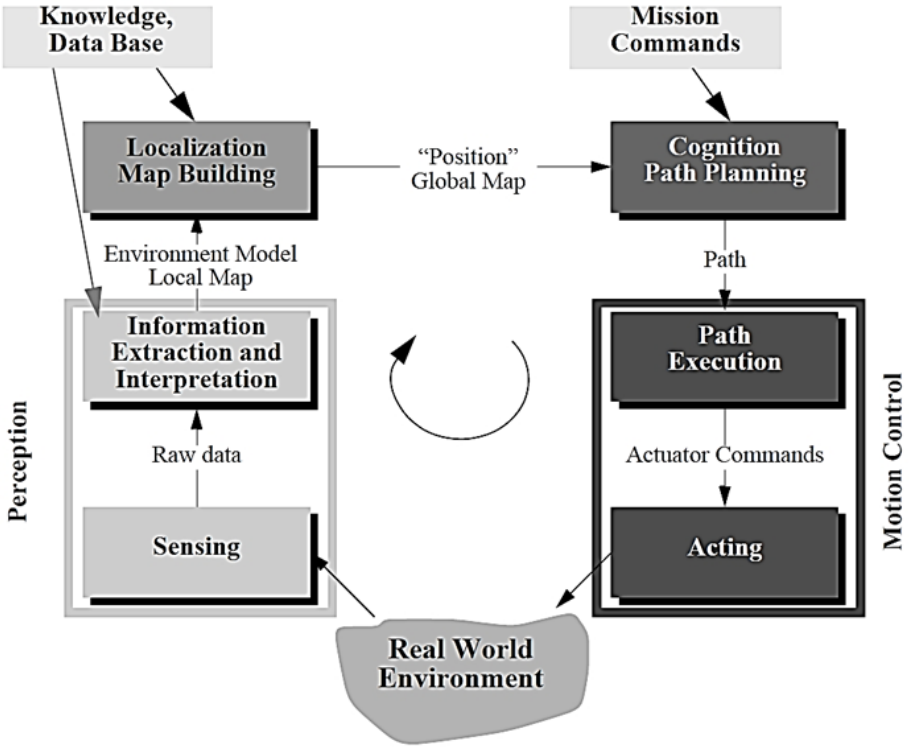

Figure 1.25. Mobile robot systems reference navigation scheme [42]

Since the primary focus of this study pertains to motion control, encompassing its various problem kinds and the most significant previous research in the field, all of these will be addressed sequentially.

1.10.1. Motion Control

In kinematic-based formulations of the motion control problem, it is assumed that the control inputs directly determine the generalized velocities of the wheeled mobile robot. There are two primary reasons for adopting this simplified assumption. Initially, given appropriate assumptions, it is feasible to nullify the dynamic influences by employing state feedback, so effectively shifting the control problem to the second-order kinematic model and then to the first-order kinematic model. Furthermore, in most cases of mobile robots, direct command over wheel torques is not feasible due to the presence of low-level loops of control that are embedded within either the hardware or the software construction. The loops in question are designed to receive a reference value through the angular speed of the wheel as input. This reference value is then replicated as precisely as possible via the use of typical regulation actions, such as PID controllers. In this scenario, the accessible inputs for high-level controls consist exclusively of the reference velocities [38]. Furthermore, it is worth noting that in most mobile robot

platforms, the internal torque controller is pre-installed, allowing users to focus on commanding the appropriate velocities of the system by considering its kinematics [60].

There exists a variety of tools that can be utilized for controlling nonholonomic systems. Nevertheless, up until now, there has been no definitive identification of a control method or collection of tools that exhibits superior performance compared to others. This can be attributed mostly to the following concrete proofs. A well-designed control law should possess two fundamental characteristics. Firstly, the system should be guided from its starting state to the destination state in a simple manner. Secondly, it should exhibit robustness against discrepancies between the model and the actual system, as well as measurements of noise and approximate knowledge regarding initial conditions. Open loop techniques have the capability to provide the initial item. However, their robustness remains uncertain, though they can be effectively utilized in the development of robust iterative designs.

Conversely, closed-loop techniques possess the likelihood for enhanced robustness, yet the inherent dynamics of the closed-loop system may lack naturalness. The closed-loop system may exhibit oscillatory behavior, which is not essential or demanded for reaching the required final point. It is worth noting that closed-loop techniques have the potential to be more robust compared to open-loop ones [110].

The motion of a mobile robot can be classified into one of the following scenarios:

1.10.1.1. Posture Control (Posture Stabilization)

The first potential approach is posture control, which refers to the ability to control both the position as well as the orientation of the robot in order to achieve the intended position and orientation. The term "posture" encompasses both the position as well as the orientation of the robot. In the context of the DDWMR, it is possible to divide the task into two distinct subtasks. The first subtask involves guiding the robot towards a destination position (x_d, y_d) within the navigation plane, beginning with an initial position (x_i, y_i) within the same plane. The second subtask entails rotating the robot around its vertical axis (θ_d) in order to correct its orientation subsequent to reaching the destination position.

It should be noted that this capability is only applicable to the particular robot under consideration, as it possesses the ability to rotate without altering its position. Another significant characteristic in this scenario is that the trajectory used by the robot to reach the destination position is inconsequential. The problem merely specifies the desired destination point as well as the intended orientation at this point for the robot without specifying any particular trajectory that should be taken

[111]. It is not possible for the nonholonomic mobile robot to get a point stabilization using a feedback law that is time-invariant and continuous in the variables of state. There are feedback laws that are time-varying and discontinuous, which have been shown to achieve the desired task [112-113]. Due to the provision of solely the initial and final postures, with the trajectory between these locations being arbitrary, novel opportunities arise, including the ability to select an "optimal" trajectory. It is imperative to emphasize the selection of a feasible trajectory that incorporates considerations of environmental, dynamic, and kinematic constraints. Typically, this results in an extensive array of possible trajectories, from which a specific trajectory is selected based on further criteria such as distance, curvature, time frame, energy consumption, and similar factors. The trajectory can be explicitly set and adjusted during movement, or it can be implicitly determined through the implementation of a control algorithm to get the desired position [60]. Figure 1.26 illustrates the specific task under consideration, wherein the DDWMR effectively tracks possible trajectories. To clarify, stabilizing a system can be understood as the process of attaining a particular point of equilibrium for the system's state [77].

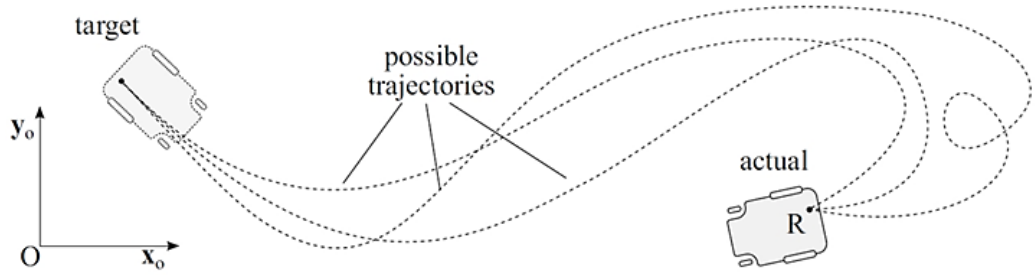

Figure 1.26. Example of posture control task [77]

1.10.1.2. Trajectory-Tracking Control

The second potential approach is trajectory-tracking control, wherein the robot is tasked with precisely tracking a specific trajectory. A trajectory refers to a vanishing point, which is a point located in the plane for ground robots or in 3D space for aerial robots. This point's position changes throughout time. This implies that the intended location is presently represented by the coordinates $(x_d(t), y_d(t))$ or $(x_d(t), y_d(t), \theta_d(t))$, which undergo motion at velocities $(\dot{x}_d(t), \dot{y}_d(t))$ or $(\dot{x}_d(t), \dot{y}_d(t), \dot{\theta}_d(t))$. In this scenario, the robot should increase its velocities beyond the ones of the trajectory in order to surpass them. Subsequently, the robot should decrease its velocity, so it matches the trajectory's velocity, thereby maintaining its position above it. Figure 1.27 depicts a trajectory tracking challenge in which a DDWMR, denoted as R, endeavors to reach a target position, denoted as M while adhering to specific timing constraints relative to a reference curve. In robotics, trajectory tracking is a commonly employed

technique to provide collision avoidance within a controlled environment. In such scenarios, DDWMRs are required to adhere to specific postures at certain time instances [77].

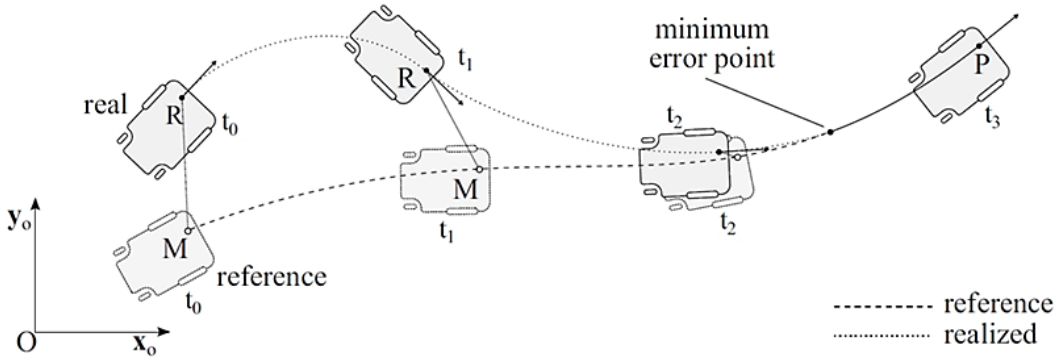

Figure 1.27. An illustration of a trajectory tracking control task [77]

It is worth noting that there are two variations of the tracking of trajectory problem that can be taken into consideration. The primary aim in the first scenario is to control the position of the robot solely; however, in the second scenario, the target extends to encompass simultaneously the orientation and position of the robot [114-115]. It should be noted that achieving flawless tracking is only possible when the trajectory used as a reference is feasible for a real robot [112], [116]. Additionally, it is essential to recognize that a trajectory that is feasible for a DDWMR may not necessarily be feasible for either a car-like robot. In the context of nonholonomic robots such as the DDWMR, the feasibility of a reference trajectory is contingent upon its generation by a reference robot that shares identical kinematic constraints with the real robot. For example, the majority of trajectories generated using an omnidirectional robot cannot be feasible for any nonholonomic mobile robot. Nevertheless, it should be noted that the lack of feasibility does not always mean that the reference trajectory is unable to be tracked to some extent, albeit with minor tracking errors that are not zero [116].

1.10.1.3. Path-Following Control

The third possible approach involves following a specified path, akin to the actions of an individual operating a car on a designated road. In order to remain on the road, it is imperative for the driver to maintain a target velocity that is consistently tangent relative to the path. Alternatively, if the velocity vector is not tangent to the vehicle's path, it will deviate from the road. To begin, the act of following a path entails initially identifying the point along the supplied path that is in closest proximity to the robot and subsequently directing the robot to attain the point in question. Secondly, the subsequent

procedure entails ensuring that the robot evolves a velocity consistently tangent to that path, irrespective of its magnitude. This implies that the value of the velocity is freely determined along the path. This implies that individuals have the freedom to choose any velocity, even zero, resulting in the vehicle coming to a temporary halt along its path, similar to when a car pauses at a red traffic light [111]. Hence, the primary goal is to control the robot's lateral displacement concerning the designated path by manipulating the robot's orientation or steering mechanism [114]. The path-following task is depicted in Figure 1.28. The objective of the task is for the DDWMR, denoted as R, to follow the reference curve and reach the point M, closest to the robot's current position on that curve, without any specific time constraints [77].

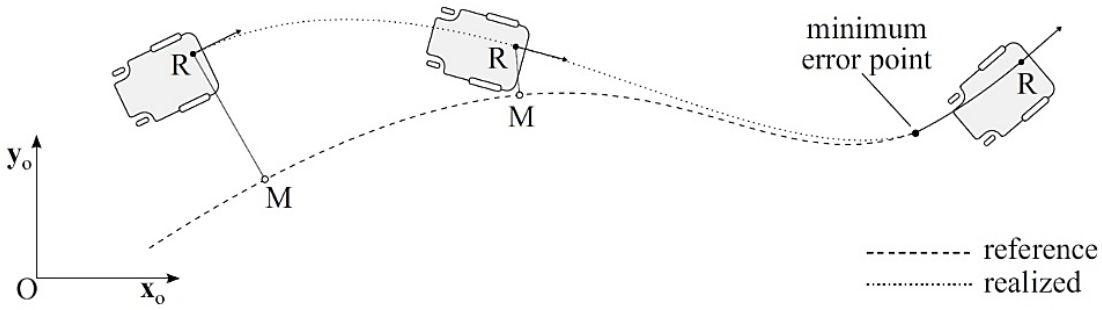

Figure 1.28. An illustration of a path-following control task [77]

1.10.1.4. Fault-Tolerant [77]

Fault tolerance can be conceptualized as the capacity of a system to successfully do a designated task, notwithstanding the existence of either software or hardware imperfections where the control system is reconfigured following the specific defect that has been isolated.

The primary distinction observed in the initial three instances is that posture control solely entails the definition of a desired destination without any specification of path or velocity. In contrast, trajectory tracking involves the determination of the vehicle's velocity, considering both its magnitude and direction, based on the trajectory currently being tracked. On the other hand, in path-following, the vehicle's velocity is determined by the designer and is constrained to be tangent to the followed path continuously [111].

From a control perspective, the unique characteristics of nonholonomic kinematics result in path following and trajectory tracking being comparatively simpler to control than posture stabilization. This is due to the availability of continuous time-invariant feedback laws that can stabilize the intended motions. Indeed, it is widely acknowledged that the attainment of feedback stabilization at a specific

posture is unattainable through the implementation of smooth time-invariant control strategies. This statement suggests that the issue at hand is genuinely nonlinear in nature, rendering linear control methods ineffectual, even locally. As a result, novel design approaches are required [111-112]. In addition, when considering various types of robot control methods, such as trajectory-tracking and path-following, posture stabilization is generally not regarded as a favorable solution for real-world environments. This is primarily due to the lack of predictability regarding the robot's movement trajectory and the potential occurrence of unforeseen collisions. It is evident that compelling the robot to navigate along or in proximity to a predetermined trajectory significantly diminishes the likelihood of collisions [38], [77].

These three cases are capable of being shown using commonplace everyday scenarios. When an individual boards a bus that reaches their workplace and subsequently falls asleep throughout the journey, they are left unaware of both the duration of the commute and the specific route taken by the bus. It is worth noting that the intended goal, namely the arrival at the workplace, has been successfully achieved, and this represents posture control. In the context of trajectory tracking, a pertinent illustration may be found in the role of a bus driver who possesses precise knowledge of the specific moments at which the bus is expected to reach each designated stop. This implies that the driver must possess knowledge of the timetable in order to effectively execute the responsibility of arriving at every bus stop punctually. To illustrate the concept of path-following, let us create a scenario where an individual operates a car on a roadway. Occasionally, individuals have the ability to increase their speed in order to travel at a faster speed. Still, at other times, it is necessary for them to decelerate in order to comply with prescribed speed limits on the road or to come to a complete halt when encountering a red traffic signal. The individual follows the designated path, specifically the roadway, during a variable timeframe, which is of negligible significance as temporal constraints are absent in this particular scenario.

1.11. Control Techniques for Wheeled Mobile Robots

In recent years, the rapid progress in mobile robotics and advancements in information processing and automation technologies have necessitated the development of control systems to enhance the autonomy of mobile robots across various work environments. The controlling of trajectories (trajectory-tracking) and the control of position and orientation (posture stabilization) have been prominent topics in the advancements within this field [117]. These topics have been approached through various traditional control methodologies, including nonlinear model predictive control, continuous time-varying adaptive controllers, back-stepping control with asymptotic stability, PID controllers, and others. In traditional control methodologies, the design of a controller often involves

utilizing the system's model and acquiring its parameters. Various current strategies can be employed to modify these parameters. The result of this process is an algorithm, commonly referred to as a control law, which obtains the inputs and computes the optimal action for effectively executing the control [118].

In contrast, Machine Learning (ML) techniques typically derive a law of control from data, wherein an agent autonomously adjusts its internal parameters to handle a given problem effectively. In recent times, there has been a shift towards the adoption of these novel paradigms in place of conventional ones. This topic can be approached using machine learning techniques, which offer alternative solutions and yield intriguing results. For instance, symbolic regression methods, neural networks, and fuzzy logic are among the several approaches that can be applied [118].

In this study, the strategies for posture and trajectory-tracking control have been classified into three categories based on previous research: artificial intelligence (machine learning) techniques, traditional techniques, and hybrid techniques.

1.11.1. Artificial Intelligence (Machine Learning) Techniques

1.11.1.1. Neural Network (NN)

A neural network comprises several individual units known as neurons, which form connections with one another. Every unique neuron possesses several inputs, a node of processing, and a solitary output. Every connection between two neurons is accompanied by a weight. The processing within a neural network occurs concurrently for all neurons [48]. The NN technique is well-suited for processes and systems that lack concise and precise mathematical models, such as mobile robot planning, identification, and control. The three main characteristics of neural networks are as follows: (i) the ability to utilize extensive sensory information effectively, (ii) the collective processing aptitude, and (iii) the capacity to learn and adapt [44], [119]. In [120-121], the authors' primary focus was the investigation of neural networks' role in controlling the trajectory tracking and posture stabilization of wheeled mobile robots. However, the adaptive control method for achieving point stabilization, as described in [122], is based on the utilization of backpropagation neural networks. The concept of identifying the two-wheeled robot was introduced. The study [123] introduced a novel control method based on adaptive neural networks. This control scheme was designed to address an omnidirectional mobile robot's trajectory tracking control problem, specifically in the face of uncertainty and external disturbance. In [124], a suggested neural network controller was presented for the trajectory-tracking of a mecanum-wheel mobile robot (MWMR). The controller featured a simple design and was based on a reference controller.

1.11.1.2. Fuzzy Logic (FL)

Fuzzy logic is a soft computing methodology that primarily addresses the challenge of uncertainty. It aims to capture and describe forms of knowledge that are unable to be adequately expressed using traditional Boolean algebra. The fuzzy logic concept was initially introduced by Lotfi A. Zadeh in the early 1990s [125-126]. The significance of fuzzy controllers is underscored by their ability to offer simplicity in control, inexpensiveness, and the potential for design even in the absence of precise mathematical models of the process. Due to such scenarios, fuzzy logic has emerged as a prominent and captivating topic within the field of robotics and computer science, finding extensive utilization across multiple applications, particularly in the realm of navigation control for autonomous mobile robots. The authors of [127] developed and executed a kinematic model for a tricycle robot, employing a trajectory tracking control system based on a fuzzy logic algorithm. The study [128] presented a novel methodology for tracking and position control in omnidirectional mobile robots (OMRs). This strategy incorporates type-2 fuzzy systems in order to efficiently control the responses and actions of these robots during intelligent navigation.

1.11.1.3. Reinforcement Learning (RL)

Reinforcement Learning (RL) pertains to a subfield within the domain of Machine Learning (ML), wherein an agent engages in interactions with its surroundings in order to obtain rewards in response to its actions. Based on this interaction, the agent is required to acquire the ability to effectively do a particular task by striking a harmonious equilibrium between the novel information acquired from the surroundings and its existing knowledge base [129]. In the context of addressing the motion control issue on non-holonomic restricted mobile robots, the study [130] presented a proposed kinematic control law of point stabilization for mobile robots. This control law is grounded in the principles of deep reinforcement learning. The articles [131], [118] have provided a comprehensive account of the process involved in designing, developing, and implementing an algorithm for controlling the position of a wheeled mobile robot. This algorithm utilizes Reinforcement Learning techniques and operates within a sophisticated 3D simulation environment. In the study [132], the application of reinforcement learning (RL) algorithms for the purpose of position control of a simulated Kephera IV mobile robot within a virtual environment was suggested. The study [133] presented a novel approach utilizing deep reinforcement learning to address the control challenges associated with non-holonomic-restricted mobile robots. The novel approach was accomplished by a proximal policy optimization learning algorithm to achieve end-to-end control, precisely posture control, of the mobile robot. The topic of

achieving robust trajectory tracking control for a three-mecanum wheeled mobile robot (MWMR) in the presence of external disturbance was investigated in [134] through the utilization of a model-based reinforcement learning (RL) algorithm. In [135], the authors have introduced a deep reinforcement learning algorithm known as the Deep Deterministic Policy Gradient algorithm for the purpose of controlling the posture of a DDWMR.

1.11.1.4. Symbolic Regression (SR)

Symbolic regression, a machine learning technique, enables the exploration of the effective structure and parameters of the desired function [136]. Symbolic regression techniques have shown significant advancements in the last ten years. Moreover, the broader scientific community has recently acknowledged the significance of interpretable machine learning. Symbolic regression approaches are predominantly employed in the context of supervised machine learning, specifically for the purpose of approximating given data [137–140]. In addition, symbolic regression methods can be employed as a form of unsupervised learning in situations where the machine learning issue for control lacks a training set. In such cases, finding a control function is guided by the objective of minimizing the quality criterion [141]. A novel numerical technique has been developed to address the optimal control issue while incorporating phase constraints in the realm of controlling wheeled mobile robots. This approach, referred to as synthesized optimal control, involves a two-step numerical technique. The problem combines two prominent tasks: the development of stabilizing control systems through symbolic regression (first step) and the optimization of control trajectories using optimal control theory, precisely the optimization step (second step), which utilizes evolutionary algorithms to tackle its objective [142]. A demonstration of the optimal control problem, specifically on the control of the equilibrium point's position, has been showcased in [143], focusing on a mobile robot equipped with mecanum wheels where the network operator method was used at the first step. The papers [144-146] have proposed a computational machine learning methodology for addressing the extended issue related to optimal control. This approach involves the utilization of a computationally synthesized optimal control technique, specifically targeting the controlling of the equilibrium point's position, in the context of DDWMR. Notably, the methodology accounts for perturbations in both models and initial conditions. The network operator method was used in [144-145], while the complete binary genetic programming method was used in [146] at the first step. The study [147] has examined the optimal control problem involving phase constraints for a collective of mobile robots. The problem is addressed by employing the synthesized optimal control approach, specifically in the context of control of the equilibrium point's position, where variational Cartesian genetic programming was used at the first step. The research

conducted in [148-150] has explored the application of machine learning via symbolic regression techniques to address the problem of trajectory tracking in optimal control. The proposed approach aims to enhance movement stability along the optimal trajectory, using the network operator method for all mentioned research.

1.11.2. Traditional Techniques

1.11.2.1. Proportional Integral Derivative (PID Controller)

The PID controller is generally recognized as a commonly used controller for a variety of control system purposes. The simplicity of the controller design and implementation is attributed to the ease of adjusting the gain parameters. Nevertheless, the model encounters significant obstacles pertaining to its non-linear nature, imprecise parameters, and erroneous parameter values. Hence, the utilization of a PID controller imposes constraints on the system's implementation design and affects its overall performance [151]. In the study [152], the implementation of PID control was explored as a means to enhance the response of a DDWMR during posture control, specifically when it is required to reach a predetermined position. The odometry approach was utilized in this context. A proposed controller in [153], with a simple PID-like structure, has been introduced for the purpose of posture control in a nonholonomic DDWMR. This controller exhibits smoothness and time-variant characteristics.

In the study [154], a PID controller was offered as a viable and efficient method to address the trajectory tracking issue of a DDWMR. In [155], a technique is shown for the development of a variable parameter PID controller for a DDWMR that is capable of tracking a NURBS trajectory with an intended velocity that varies over time. The design methodology for a PID controller containing time-varying parameters to obtain trajectory tracking control for a mecanum-wheeled robot has been provided in [156], wherein a little inaccuracy is observed.

1.11.2.2. Backstepping Controller

The utilization of backstepping control is a highly significant approach in the realm of stabilizing nonholonomic systems. The concept of backstepping involves employing a recursive approach to decompose a design problem pertaining to an overall system into a series of design problems pertaining to subsystems of lower order. The backstepping control law originates from the stability proof through the application of Lyapunov-like analysis, which ensures the asymptotic convergence of the posture error [157]. In the study [158], a generalized nontriangular normal form was introduced to aid in the

development of a recursive integral backstepping control strategy for a specific category of underactuated nonholonomic systems. Specifically, this control strategy was designed for wheeled mobile robots (WMRs) tasked with posture stabilization and tracking desired trajectories in obstacle-free environments. The authors of [159] have introduced a novel approach for posture stabilization of a WMR employing backstepping control with the inclusion of output feedback. The authors of [160-162] introduced a trajectory-tracking controller for a wheeled mobile robot that utilizes the backstepping approach. In the study [163], a time-varying feedback trajectory-tracking control method was introduced for stabilizing the trajectory of a WMR using the backstepping strategy.

1.11.2.3. Sliding Mode Controller (SMC)

The sliding mode control (SMC) is a nonlinear control strategy that has been primarily devised for the purpose of controlling variable-structure systems. The proposed approach involves the utilization of a time-varying state-feedback discontinuous control law that rapidly switches between different continuous structures based on the current position of the state variables in the state space. The primary goal is to ensure that the dynamics of the controlled system precisely follow what is needed and predefined [164]. The study [165] focuses on the finite-time posture stabilization of a unicycle mobile robot, specifically when merely position information is accessible. This is achieved through the development of a discrete-time sliding mode controller (DSMC). The authors of [166] have presented a trajectory-tracking robust algorithm solution for the perturbed kinematic model of a unicycle mobile robot. This algorithm employs the first-order sliding mode control approach. The authors of [167] have presented a robust adaptive trajectory tracking controller, specifically a sliding mode controller, intended to control an electric wheeled mobile robot operating in a scenario of dynamic disturbances. The problem pertaining to trajectory tracking control for nonholonomic mobile robots in the presence of unknown disturbances has been investigated in the study [168] using the approach of sliding mode control.

1.11.2.4. Model Predictive Control (MPC)

Model Predictive Control (MPC) has emerged as a technique utilized for the design and implementation of feedback control systems, which has demonstrated superior performance compared to other approaches in numerous scenarios. Furthermore, Model Predictive Control (MPC) offers a robust and versatile approach for the development of control systems applicable to a wide range of multiple-input, multiple-output (MIMO) systems [169]. In the study [170], the authors implemented two stabilizing nonlinear model predictive control (NMPC) designs, known as the final-state equality

constraint stabilizing design and the final-state inequality constraint stabilizing design, in order to accomplish control objectives for a two-wheeled mobile robot. These objectives included point stabilization and trajectory tracking. In the study [171], a novel approach to model predictive control was introduced for achieving point stabilization of wheeled mobile robots (WMRs) under nonholonomic constraints. The proposed method involved formulating a linearized error model by converting the position of the robot into a polar frame. The researchers in [172] conducted a study on Model Predictive Control strategies that do not incorporate stabilizing restrictions or costs to achieve set-point stabilization for holonomic mobile robots. The utilization of a nonlinear model predictive control technique was documented in [173]. This approach was applied to a DDWMR in order to address point-stabilization challenges while incorporating avoidance strategies for both static and dynamic impediments. In order to ensure system safety and achieve optimal performance within a limited prediction horizon, researchers in [174] have investigated the application of a control barrier function in a nonlinear model predictive control (NMPC) framework. This approach effectively decreases the computational burden associated with real-time NMPC implementation. In [175], a proposal was made for a model predictive control approach that is linear and time-varying. This technique is intended for the trajectory-tracking of a single-wheeled mobile robot, taking into account nonholonomic restrictions and control constraints.

1.11.2.5. Lyapunov-Based Controller

The utilization of a Lyapunov-based controller constitutes a prevalent approach within control theory for the purpose of designing nonlinear controllers specifically tailored for mechanical systems [176]. In the study [177], researchers introduced two kinematic control techniques that are not smooth in nature. These strategies were developed specifically for the purpose of posture stabilization of a wheeled mobile robot using a differential drive system. The approach that was formulated relied on the principles of kinematic coordinate transformation and the Lyapunov-like stability technique. The research conducted by [178] has introduced a robust switching control approach based on passivity for stabilizing the posture of wheeled mobile robots (WMRs) in the presence of model uncertainty. This control law was derived using the Lyapunov approach and energetic passivity.

Numerous traditional techniques exist, although the most significant and prevalent ones have been briefly discussed.

1.11.3. Hybrid Techniques

These techniques encompass a combination of exclusively artificial intelligence techniques as a first type, as documented in [179], utilized neural networks and fuzzy logic to achieve trajectory tracking for an omnidirectional mobile robot. Similarly, the study [180] presented the application of reinforcement learning and fuzzy logic for trajectory tracking control of an autonomous mobile robot. As a second type of these techniques is solely traditional techniques, as in [181], the authors have proposed the utilization of backstepping and nonlinear PID controller to achieve trajectory tracking control for a two-wheeled mobile robot. The study [182] also introduced the application of backstepping and second-order sliding mode for trajectory tracking of a car-like robot. Furthermore, as a third type, there are approaches that integrate both artificial intelligence techniques and traditional techniques, as exemplified in [183], where fuzzy logic and PID controller were suggested to address the precise trajectory-tracking issue of two nonholonomic WMRs with a variety of disruptions and noises.

Additionally, [184] proposed the use of a deep neural network and model predictive controller for trajectory-tracking of a car-like robot. In [185], a new intelligent controller (an adaptive neural network implemented within a nonlinear control framework based on Lyapunov) was proposed to enhance the accuracy of trajectory tracking in omnidirectional robots, particularly in the presence of unstructured uncertainty. The primary objective of the research study [186] was to ascertain the PID-controller coefficients through the implementation of reinforcement learning method in order to regulate the angular velocity of the turning motion of the two wheels of DDWMR for the purpose of trajectory tracking. The study by [187] introduced the application of a deep reinforcement learning method to adjust the PID controller gain parameters combined with fuzzy control. This approach aimed to improve the trajectory tracking performance of a wheeled mobile robot (WMR). In the study [188], researchers devised an advanced control methodology for trajectory-tracking tasks of an omnidirectional mobile robot. This methodology involved the use of an intelligent Proportional Integral Derivative (PID) neural network, and the weights of the controller were tuned using the Particle Swarm Optimization (PSO) algorithm. The research [189] introduced a novel approach for achieving path-tracking control of an omnidirectional robot using model predictive control (MPC) integrated with an adaptive neural-fuzzy inference system. The study [190] introduced a fuzzy adaptive sliding mode controller designed for an electrically driven WMR. The controller's purpose was to achieve trajectory tracking in an environment of uncertainties and disruptions. In the study conducted by [191], a novel fuzzy adaptive PID control approach was introduced. This method was specifically designed for the purpose of achieving trajectory tracking control in an eight-mecanum-wheeled omnidirectional mobile robot.

CHAPTER 2. METHODOLOGY

2.1. The Problems of Machine Learning

Machine learning systems, in nearly all practical implementations, are designed for estimation of functional relationships between input features and desired outputs. Neural networks, for instance, serve as powerful tools for modeling such relationships, enabling the discovery of mappings between feature spaces, However, these relationships are always represented as computational black box. The resultant functional relationship can be utilized for purposes such as classification, modeling, prediction, and so forth. However, a precise mathematical formulation of this function cannot be inferred.

An unknown function refers to a collection of computational approaches that convert a vector $\boldsymbol{x}$ in some input space **X** into a vector $\boldsymbol{y}$ in some output space **Y**. It is characterized by the absence of a mathematical statement $\boldsymbol{y} = f(\boldsymbol{x})$ to describe the relationship between the two vectors. The unknown function that relates the input vector $\boldsymbol{x}$ to the output vector $\boldsymbol{y}$ is denoted as

$$\boldsymbol{y} = \psi(\boldsymbol{x}). \tag{2.1}$$

Machine learning refers to the computational execution of a procedure aimed at finding an unknown function using computer systems. In order to effectively utilize machine learning techniques to address a variety of problems, these problems must be conceptualized as tasks aimed at inferring an unknown function

$$\boldsymbol{y} = \eta(\mathbf{x}, \mathbf{q}), \tag{2.2}$$

where $\mathbf{q}$ signifies the vector comprising the system's requisite parameters, $\mathbf{q} \in \mathbb{R}^{m_q}$, and η is a function that equals or approximated to ψ based on a specific criterion.

There exist two distinct methodologies for searching an unknown function: parametric and structural-parametric techniques.

The parametric approach involves the investigator defining the functional template of the unknown function, including its structural assumptions, while designating certain parameters, for example, η in Eq. (2.2) is specified. The machine learning task accordingly becomes one of parameter estimation—seeking the parameter vector $\mathbf{q}$ that satisfies the chosen criterion. In the context of unknown function approximation, neural networks belong to the class of parametric approaches. Their computational transformations are governed by a predefined functional structure, with performance dictated by the optimization of numerous internal parameters.

The structural-parametric methodology extends beyond conventional parametric modeling by treating not only the parameters but also the functional form itself of the unknown function. It seeks to determine the most suitable functional structure (η) and concurrently identify the desired values of its

internal parameters (**q**). Currently, the structural-parametric approach is being effectively employed through the utilization of symbolic regression techniques. These methodologies establish a foundational repertoire of elementary functions and associated structural encoding rules. Subsequently, employing a genetic algorithm, the system searches for the desired symbolic structure of the target function while concurrently tuning its numerical parameters within the predefined code space. Symbolic regression techniques exhibit variations in coding rules as well as the crossover and mutation processes employed by the genetic algorithm on the codes.

Machine learning's end goal is to seek out an unknown function, and this search must be guided by some sort of evaluative criterion. Machine learning problems are often broadly classified as either unsupervised or supervised, contingent upon the nature of the evaluative criterion being used. It's essential to keep in mind that the many different kinds of machine learning that exist today can be placed in one of these classes depending on the evaluation criteria used to classify them. An evaluation criterion is given in some problems as follows:

$$\varrho(\eta(\mathbf{x}, \mathbf{q})): \mathbf{X} \times \mathbb{R}^{m_q} \to \mathbb{R}^1. \tag{2.3}$$

2.1.1. Unsupervised machine learning

Unsupervised machine learning involves the find of a function (3.2) that satisfies a specified estimate (2.3), resulting in the fulfilment of the subsequent inequation

$$\|f^* - \zeta(\eta(\mathbf{x}, \mathbf{q}))\| \leq \delta, \tag{2.4}$$

where f^* represents a value that meets the estimate requirements, and δ represents a positive value of tiny magnitude.

2.1.2. Supervised machine learning

A second method for assessing the target function involves concocting a training set. A training set refers to a set of potential examples that are utilized in the process of learning to pinpoint an unknown function.

Two sets of dimensions that are compatible with each other

$$(\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}) \tag{2.5}$$

are known as a training set in a case

$$\widetilde{\mathbf{X}} = \{\boldsymbol{x}^1, \dots, \boldsymbol{x}^N\} \subseteq \mathbf{X}, \tag{2.6}$$

$$\widetilde{\mathbf{Y}} = \{\boldsymbol{y}^1 = \psi(\boldsymbol{x}^1), \dots, \boldsymbol{y}^N = \psi(\boldsymbol{x}^N)\} \subseteq \mathbf{Y}, \tag{2.7}$$

and it can be postulated that there exists a mapping of one-to-one between the elements of set **X** and set **Y**.

Supervised machine learning includes the creation of a training set (2.5) and the identification of a function (2.2) where if the overall error for said training set is smaller than the specified value $\boldsymbol{\varepsilon}$

$$\sum_{i=1}^{N} \left\| \boldsymbol{y}^i - \eta(\boldsymbol{x}^i, \mathbf{q}) \right\| \leq \boldsymbol{\varepsilon}, \tag{2.8}$$

then for every value of $\boldsymbol{x}^*$ that is not contained in said training set $\boldsymbol{x}^* \notin \widetilde{\mathbf{X}}$, the next inequation is satisfied

$$\| \boldsymbol{y}^* - \eta(\boldsymbol{x}^*, \mathbf{q}) \| \leq \delta, \tag{2.9}$$

where $\boldsymbol{y}^* = \psi(\boldsymbol{x}^*)$.

Machine learning control constitutes a paradigm wherein machine learning methods are employed to autonomously discover an unknown control function. The discipline of control encompasses several challenging problems, such as the optimization of control in different formulations, such as Pontryagin or Bellman formulations. Another significant problem is the control general synthesis, which involves designing a feedback function based on the object's state [192].

2.2. The Problem of Optimal Control

The optimal control issue holds a prominent position within the domain of control theory. The aforementioned issue has historically garnered the attention of mathematicians, leading to the integration of control theory as a distinct area within the study of mathematics.

The problem of optimal control involves the characterization of the control object through an ordinary differential equations system, wherein the right part of these equations contains an unknown control vector. The provided information includes the initial and terminal conditions, as well as the integral quality functional. The finding of the control as a time function is an imperative task in this problem. By substituting the given function into the right part of the differential equations, a non-stationary differential equations system is obtained, where the right part is a known function of time. The non-stationary differential equations system yields a specific solution that satisfies the initial

conditions and eventually arrives at terminal conditions. In this scenario, the value of the quality functional is satisfied.

Several reasons represent the rationale for presenting the problem of optimal control in this context here. Initially, in the context of the problem of optimal control, it is imperative to identify a function for one variable at least. Consequently, the utilization of machine learning techniques becomes viable in the pursuit of such a function. Furthermore, once the optimal control problem has been solved and the control as a function of time has been determined for the hands-on execution of the identified optimal solution, an effective stabilization system must be designed to constrain the motion of the controlled agent to the desired optimal trajectory. This gives rise to the challenge of identifying an additional control function, thereby necessitating a renewed focus on the machine learning problem. Ultimately, the problem of achieving optimal control is able to be handled subsequent to the solution of the stabilization problem of the object concerning the point of equilibrium inside the state space.

Give the following mathematical problem statement for the problem of optimal control. The control object is represented by a mathematical model in the form of an ordinary differential equations system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{2.10}$$

with $\boldsymbol{x}$ being a vector representing the state space, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u}$ denotes a vector representing the control, $\boldsymbol{u} \in \boldsymbol{U} \in \mathbb{R}^m$, and $\boldsymbol{U}$ representing a compact set, $m \leq n$.

The initial conditions for the system model (2.10) are provided

$$\boldsymbol{x}(0) = \boldsymbol{x}^0. \tag{2.11}$$

Terminal conditions are determined by

$$\boldsymbol{x}(t_f) = \boldsymbol{x}^f, \tag{2.12}$$

where t_f represents a terminal time for this system, which is an unspecified value that is assigned by the fulfillment of the terminal conditions.

The quality criterion can be expressed through the utilization of an integral and/or terminal functional

$$J = F\left(\boldsymbol{x}(t_f)\right) + \int_0^{t_f} \boldsymbol{f}_0(\boldsymbol{x}(t), \boldsymbol{u}(t))dt \rightarrow \min. \tag{2.13}$$

It is essential to ascertain the obtained control should be as a time function

$$\boldsymbol{u} = \boldsymbol{h}(t), \tag{2.14}$$

where $\boldsymbol{h}(t) \in \boldsymbol{U}$ for $t \in [0: t_f]$.

The control function $\boldsymbol{h}(t)$ that is obtained is commonly referred to as a program control. When the control function (2.14) is replaced into the right-hand side of the system (2.10), the resulting differential equations system appears as follows

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{h}(t)\big). \tag{2.15}$$

The system (2.10) possesses a partial solution $\boldsymbol{x}(t, \boldsymbol{x}^0)$ that satisfies the initial conditions (2.11) and leads to the attainment of the terminal conditions (2.12), while the quality criterion value (2.13) is satisfied.

Therefore, under the provided mathematical formulation of the problem of optimal control, the task entails finding the optimal function of control (2.14). This implies that the aforementioned problem can be classified as a problem of machine learning control and is capable of being solved through the utilization of machine learning techniques.

2.3. The Problem of Control Synthesis

The problem of control synthesis holds significant prominence within the field of control theory. Unlike the earlier optimal control problem, this formulation possesses a more application-oriented nature, as the control is synthesized as a state-dependent function. This results in a feedback control structure that dynamically responds to sensor-derived state information. This unit assures that the object attains the control objective, while the value of the quality criterion of this control for any object's current state is satisfactory. The problem of control synthesis is characterized by this particular feature. The solution of one problem of control synthesis can be considered tantamount to the solution of an infinite collection of problems of optimal control. Once the control synthesis problem has been solved, the derived control architecture inherently enables the solution of the optimal control problem across all feasible states of the system.

In the nascent phase of modern control theory, particularly during the 1960s, R. Bellman engaged in a rigorous mathematical examination of optimal control problems, which culminated in the formal articulation of the control synthesis problem and the derivation of the Bellman equation—a defining achievement in dynamic systems theory [193]. The stated equation represents a partial differential equation. The equation's solution is represented by the Bellman function, which takes the control vector as one of its arguments. The finding of this control that optimizes the Bellman function represents a

viable solution to the problem of control synthesis. It is essential to acknowledge the following: Partial differential equations exhibit a much higher level of complexity compared to ordinary differential equations, and in the majority of cases, they do not possess a universal solution at all. Bellman suggested a numerical approach to finding a solution using dynamic programming [194-195]. Using this approach on a vast array of numerical values representing state vectors generates a significant quantity of control vectors.

Many control synthesis problems had been effectively solved during that specific period via the Pontryagin maximum principle [196]. The result was favorable, as the analysis primarily focused on simple second-order models of the control objects. The problem of time-optimal has been successfully solved, resulting in the derivation of comprehensive solutions for the differential equations governing the control object as well as conjugate variables. Subsequently, the switching points of the control have been determined based on the derived solutions obtained from various initial conditions. It is evident that this approach is not universally applicable. However, when employing this approach, Boltyanskii [197] performed the formulation of the control general synthesis problem, which remains a pressing mathematical problem to this day since its mathematical formulation lacks comprehensive analytical and numerical techniques for solution at present.

Let us contemplate a traditional formulation of the problem of control synthesis. The differential equations system is characterized by a control object having a particular form (2.10). The initial conditions domain within the state space can be given by

$$\mathbf{X}_0 \subseteq \mathbb{R}^n. \tag{2.16}$$

The presence of the domain for the initial condition constitutes a fundamental characteristic of the general synthesis of the control problem. Boltyanskii first established the initial conditions domain as the entire space of states $\mathbf{X}_0 \subseteq \mathbb{R}^n$, as he searched for tackling this topic by analytical means. In this particular scenario, we adopt a numerical approach to address the problem at hand. Hence, the domain $\mathbf{X}_0$ can be considered a constrained subset within the state space.

The terminal conditions (2.12) are provided.

The quality criterion is provided

$$J_1 = \int_{\mathbf{X}_0} \dots \int \left(F\left(\boldsymbol{x}(t_f, \boldsymbol{x}^0)\right) + \int_0^{t_f} \boldsymbol{f}_0\big(\boldsymbol{x}(t, \boldsymbol{x}^0), \boldsymbol{u}(t)\big) dt \right) dx_1^0 \dots dx_n^0 \to \min_{\boldsymbol{u} \in \boldsymbol{U}}, \tag{2.17}$$

where $\boldsymbol{x}^0 = [x_1^0 \dots x_n^0]^T \in \mathbf{X}_0$, the value of t_f is not explicitly provided, but rather is calculated by fulfilling the terminal conditions (2.12). It is vital to note that the value of t_f can change depending on the change of initial conditions.

It is crucial for pinpointing of a control function in terms of the vector of state space

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}) \in \boldsymbol{U}, \quad \boldsymbol{g}(\boldsymbol{x}) : \mathbb{R}^n \to \mathbb{R}^m. \tag{2.18}$$

If the control function that has been acquired is included into the right-hand side of the mathematical equation (3.10), then the resulting system of stationary differential equations

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x})\big), \tag{2.19}$$

will possess a partial solution for any initial condition within the initial domain (2.16).

$$\boldsymbol{x}(0) = \boldsymbol{x}^0 \in \mathbf{X}_0, \tag{2.20}$$

which fulfills the terminal condition (2.12) and the quality criterion value (2.17) is satisfied. Therefore, the task of addressing the synthesis problem can be seen as the search for the control function (2.18), which coincides with the principles of machine learning control.

In order to computationally solve the problem of control synthesis (2.10), (2.16), (2.12), (2.17), the domain of initial condition (2.16) is substituted with a limited set of initial conditions

$$\boldsymbol{X}_0 = \{\boldsymbol{x}^{0,1}, \dots, \boldsymbol{x}^{0,\mathbf{L}}\}. \tag{2.21}$$

and the quality criterion multiple integral (2.17) is substituted with the corresponding summation for all of the initial conditions

$$J_2 = \sum_{i=1}^{\mathbf{L}} \left(F\left(\boldsymbol{x}\big(t_{f,i}, \boldsymbol{x}^{0,i}\big)\right) + \int_0^{t_{f,i}} \boldsymbol{f}_0\left(\boldsymbol{x}\big(t, \boldsymbol{x}^{0,i}\big), \boldsymbol{u}(t)\right) dt \right), \tag{2.22}$$

where $t_{f,i}$ represents the time at which the terminal condition is reached, starting from the initial one $\boldsymbol{x}^{0,i}$, $i = 1, \dots, \mathbf{L}$.

The equation provided determines the time at which the terminal condition gets achieved during the search procedure is

$$t_{f,i} = \begin{cases} t, & \text{if } \ t < t^+ \text{ and } \left\|\boldsymbol{x}^f - \boldsymbol{x}\right\| \le \varepsilon \\ t^+ & , \ \text{otherwise} \end{cases} \tag{2.23}$$

where ε represents the degree of accuracy required to achieve the terminal condition, while t^+ denotes the maximum time allowed for obtaining this condition. It is crucial to note that both ε and t^+ are positive numerical values.

There is a method for solving the synthesis problem, based on the Bellman equation

$$-\frac{d\mu(\boldsymbol{x})}{dt} = \min_{\boldsymbol{u}\in\boldsymbol{U}}\left\{\left(\frac{\partial\mu(\boldsymbol{x})}{\partial\boldsymbol{x}}\right)^T \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + \frac{\partial F(\boldsymbol{x})}{\partial\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + f_0(\boldsymbol{x},\boldsymbol{u})\right\}. \tag{2.24}$$

In case of the Bellman function $\mu(\boldsymbol{x})$ exists, the control function can be obtained by solving the Bellman equation (2.24)

$$\boldsymbol{u} = argmin\left\{\left(\frac{\partial\mu(\boldsymbol{x})}{\partial\boldsymbol{x}}\right)^T \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + \frac{\partial F(\boldsymbol{x})}{\partial\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + f_0(\boldsymbol{x},\boldsymbol{u})\right\}. \tag{2.25}$$

In order to address the synthesis problem utilizing the Bellman equation through machine learning, it needs to employ an approximation technique for the Bellman function. In order to implement the symbolic regression technique for the Bellman function, it is necessary to modify the functional to incorporate the various initial as well as terminal conditions

$$J_s = \sum_{j=1}^{\mathbf{L}}\left(\int_0^{t_{f,j}} f_0\left(\boldsymbol{x}\left(t,\boldsymbol{x}^{0,j}\right),\boldsymbol{u}(t)\right)dt + p_1\sqrt{\sum_{i=1}^{n}(x_i^f - x_i(t_{f,i},\boldsymbol{x}^{0,j}))^2}\right) \to \min_{\mu(\boldsymbol{x})}, \tag{2.26}$$

where p_1 represents a weight coefficient, and $\boldsymbol{x}(t,\boldsymbol{x}^{0,\mathbf{L}})$ denotes the system partial solution with control (2.25) beginning with the initial condition $\boldsymbol{x}^{0,j}$.

2.4. The Problem of Synthesized Optimal Control (The Problem Statement of This Study)

In an effective scenario, our ongoing objective remains the pursuit of developing systems that exhibit influential performance concerning the specified criterion. In this particular scenario, the problem of optimal control is addressed as a preliminary step. However, it should be noted that the solution to this problem cannot be implemented directly on a control object's board processor. This is because the optimal control function obtained is dependent on time, and its implementation would result in an open-loop control system. Consequently, Any temporal misalignment between the motion of the

controlled object and the application of control actions may result in the failure to achieve the desired control objective, leading to a deviation between the actual performance and the theoretically computed value of the quality criterion. To counteract deviations arising between the real-time trajectory of the system and its computed optimal counterpart, practical control architectures incorporate feedback stabilization mechanisms designed to maintain adherence to the optimal reference trajectory.

However, as a result of the implementation of the system of stabilization, we once again encounter a loss of optimality. Several pieces of evidence suggest this assertion:

1. The inclusion of a stabilization system leads to the transformation of the system's behavior and its mathematical representation, which may result in a control strategy that no longer satisfies optimality conditions for the updated dynamics.
2. The deviation of the system state from the trajectory may be characterized by temporal misalignment or spatial displacement; both forms yielding motion patterns that no longer satisfy the criteria for optimality.
3. The capacity of the stabilization system to restore the system state to the reference trajectory necessitates the reservation of sufficient control resources. Optimal control strategies should therefore incorporate this allocation, yet such accounting is routinely absent from standard computational frameworks.
4. Lastly, it should be noted that the object's motion in close proximity to the programmed trajectory can show notable deviations from the desired trajectory concerning the functional value.

Based on the suggested approach, it is necessary to first establish the stability of the control object within the state space before addressing the problem of optimal control. Hence, this approach is referred to as synthesized optimal control. The foundational premise is the derivation of a feedback control function that ensures the presence of a stable equilibrium point for the closed-loop system of differential equations. Moreover, the equilibrium's position is rendered controllable through a set of design parameters internal to the controller.

Contemplate About the Statement of the Problem of Synthesized Optimal Control:

Assuming the control object mathematical model, expressed as differential equations system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}), \tag{2.27}$$

with $\boldsymbol{x}$ being a vector representing the state space, $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{u}$ denotes a vector representing the control, $\boldsymbol{u} \in \boldsymbol{U} \in \mathbb{R}^m$, and $\boldsymbol{U}$ representing a compact set, $m \leq n$.

Provided the initial condition

$$\boldsymbol{x}(0) = \boldsymbol{x}^0. \tag{2.28}$$

Terminal conditions is determined by

$$\boldsymbol{x}(t_f) = \boldsymbol{x}^f, \tag{2.29}$$

where t_f represents the time at which the terminal condition is reached, t_f is not explicitly provided but is bounded

$$t_f \leq t^+, \tag{2.30}$$

and t^+ is provided.

Given the quality criterion

$$J_{so1} = \int_0^{t_f} f_0(\boldsymbol{x}, \boldsymbol{u})dt \to \min_{\boldsymbol{u}\in\boldsymbol{U}}. \tag{2.31}$$

It is crucial to pinpoint a control that matches to the subsequent form:

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}^*(t) - \boldsymbol{x}) \in \boldsymbol{U}, \tag{2.32}$$

where $\boldsymbol{x}^*(t)$ is a time function.

The function

$$\boldsymbol{g}(\boldsymbol{x}^*(t) - \boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}^m, \tag{2.33}$$

is sought in a manner that exhibits a property of feasibility [198], i.e. at each given time $t = t_k \leq t_f$, the system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x}^*(t_k) - \boldsymbol{x})\big), \tag{2.34}$$

possesses a stable point of equilibrium

$$\tilde{\boldsymbol{x}}(\boldsymbol{x}^*(t_k)) \in \mathbb{R}^n, \tag{2.35}$$

$$\boldsymbol{f}\big(\tilde{\boldsymbol{x}}, \boldsymbol{g}(\boldsymbol{x}^*(t_k) - \tilde{\boldsymbol{x}})\big) = 0, \tag{2.36}$$

$$\det(\mathbf{A} - \lambda\mathbf{E}) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = \prod_{j=1}^{n}\big(\lambda - \lambda_j\big) = 0, \tag{2.37}$$

where

$$\lambda_j = \alpha_j + i\beta_j, \;\; \alpha_j < 0, \; j = 1, \dots, n, \tag{2.38}$$

$i = \sqrt{-1}$,

$$\mathbf{A} = \frac{\partial \boldsymbol{f}\left(\tilde{\boldsymbol{x}}, \boldsymbol{g}(\boldsymbol{x}^*(t_k) - \tilde{\boldsymbol{x}})\right)}{\partial \boldsymbol{x}}. \tag{2.39}$$

The function (2.32) serves as a stabilization system for the object (2.27). Consequently, the object achieves stability in relation to a certain point inside the state space $\tilde{\boldsymbol{x}}$ (2.36). The positioning of this point of stabilization is contingent upon the parameters $\boldsymbol{x}^*$. The parameters $\boldsymbol{x}^*$ are able to serve as the direct coordinates of the stabilization point inside the state space. Alternatively, in the general situation, $\boldsymbol{x}^*$ can influence the positioning of a point of stabilization $\tilde{\boldsymbol{x}}$ $(\boldsymbol{x}^*)$ within the state space.

The solution to the problem of synthesized optimal control and the finding of the control function (2.32) is considered to be performed algorithmically in two steps, which are treated as sequential activities.

2.4.1. First Step: Synthesis of Stabilization System

In the first step of stabilization, the problem of control synthesis is addressed in order to establish the presence of a stable point of equilibrium inside the state space. The problem statement can be addressed using numerical solutions utilizing machine learning approaches.

The control object mathematical model (2.27) is given.

The initial conditions set is provided by

$$\boldsymbol{X}_0 = \{\boldsymbol{x}^{0,1}, \dots, \boldsymbol{x}^{0,\mathbf{L}}\}. \tag{2.40}$$

The terminal position is provided. Any point in the state space has the potential to serve as the terminal position, enabling the system to achieve stabilization at such a point. In the problem of optimal control, the position of the terminal condition (2.29) cannot be the exact position of the mentioned terminal position.

$$\boldsymbol{x}(t^*) = \boldsymbol{x}^* \in \mathbb{R}^n, \tag{2.41}$$

where the value of t^* is not provided, but bounded

$$t^* = \begin{cases} t, & \text{if } \; t < t^+ \text{ and } \|\boldsymbol{x}^* - \boldsymbol{x}(t, \boldsymbol{x}^0)\| \le \varepsilon \\ t^+, & \text{otherwise} \end{cases} \tag{2.42}$$

where $\boldsymbol{x}(t, \boldsymbol{x}^0)$ is the system partial solution (2.27), and ε and t^+ are provided positive numerical values.

It is crucial to pinpoint a control that matches to the subsequent form:

$$\boldsymbol{u} = \boldsymbol{g}(\boldsymbol{x}^* - \boldsymbol{x}), \tag{2.43}$$

that generally partial solution of the differential equations system

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\big(\boldsymbol{x}, \boldsymbol{g}(\boldsymbol{x}^* - \boldsymbol{x})\big), \tag{2.44}$$

from whatever initial condition inside the specified area (2.40)

$$\boldsymbol{x}^{0,i} \in \mathbf{X}_0, i = 1, \dots, \mathbf{L}. \tag{2.45}$$

will fulfil the terminal condition (2.41) by optimizing the value of the subsequent criterion:

$$J_{s1} = \sum_{i=1}^{\mathbf{L}} \big(t_i^* + p_1 \big\| \boldsymbol{x}^* - \boldsymbol{x}(t_i^*, \boldsymbol{x}^{0,i}) \big\|\big), \tag{2.46}$$

where

$$t_i^* = \begin{cases} t, & \text{if } \ t < t_1^+ \text{ and } \big\| \boldsymbol{x}^* - \boldsymbol{x}(t, \boldsymbol{x}^{0,i}) \big\| \le \varepsilon_1 \\ t_1^+, \text{otherwise} \end{cases} \tag{2.47}$$

$$\big\| \boldsymbol{x}^* - \boldsymbol{x}(t, \boldsymbol{x}^{0,i}) \big\| = \sqrt{\sum_{i=1}^{n} (\boldsymbol{x}^* - \boldsymbol{x}(t, \boldsymbol{x}^{0,i}))^2}\,, \tag{2.48}$$

where p_1 represents a weight coefficient, and ε_1 and t_1^+ are provided positive numerical values.

2.4.2. Second Step: Solution of the Problem of Optimal Control

As a second step in synthesized optimal control, following the solution of the problem of control synthesis (2.27), (2.40)–(2.48), the problem of optimal control (2.27)–(2.31) is addressed for the mathematical formula (2.44). This entails the finding of a control function using the subsequent form:

$$\boldsymbol{x}^*(t) = \boldsymbol{h}(t), \tag{2.49}$$

to minimize the specified criterion (2.31).

In the second step, it is essential to observe that the dimension of the sought function (2.49) is equivalent to that of the state space. In the context of this specific scenario, it is possible to search the function as a piecewise constant function

$$\boldsymbol{h}(t) = \boldsymbol{x}^{*,i}, \quad if \ \ (i-1)\Delta \le t \le i\Delta, \tag{2.50}$$

where $\boldsymbol{x}^{*,i}$ are obtained influential coordinates values of the point of equilibrium, $i = 1, \dots, K$, and Δ is a provided time interval,

$$K = \left\lfloor \frac{t^+}{\Delta} \right\rfloor. \tag{2.51}$$

Therefore, per this methodology, the primary objective of synthesized optimal control is first to guarantee the object's stability, which involves the emergence of an equilibrium point inside the phase space. In the vicinity of the point of equilibrium, the phase trajectories exhibit a contraction property, which plays a crucial role in determining the system's feasibility. The principal property of this framework, relative to conventional optimal control paradigms, lies in its intrinsic capacity to synthesize feedback-based control laws — thereby yielding closed-loop systems — as opposed to the open-loop nature of most optimal control solutions derived under idealized assumptions.

To realize the target behavior, the control strategy must be computed numerically as part of the stabilization synthesis process; once derived, these control expressions replace the original input terms in the system’s differential equations. In a usual scenario, the control object functions within a dynamic environment, necessitating the essential ability to compute the desired trajectory on board. The utilization of synthesized optimal control methodology enables the achievement of this objective. The stabilization synthesis problem is addressed during the design phase initially, yielding a parametric control structure wherein the location of the equilibrium point — acting as a design variable — may be precomputed or updated recursively in real time via onboard computation to ensure continued performance optimality.

2.5. The General Methodology of Symbolic Regression

Symbolic regression techniques refer to a set of approaches utilized in machine learning tasks to encode mathematical expressions. These techniques involve a range of algorithms that aim to identify the most influential mathematical expressions within the space that contains these encoded codes (see Figure 2.1).

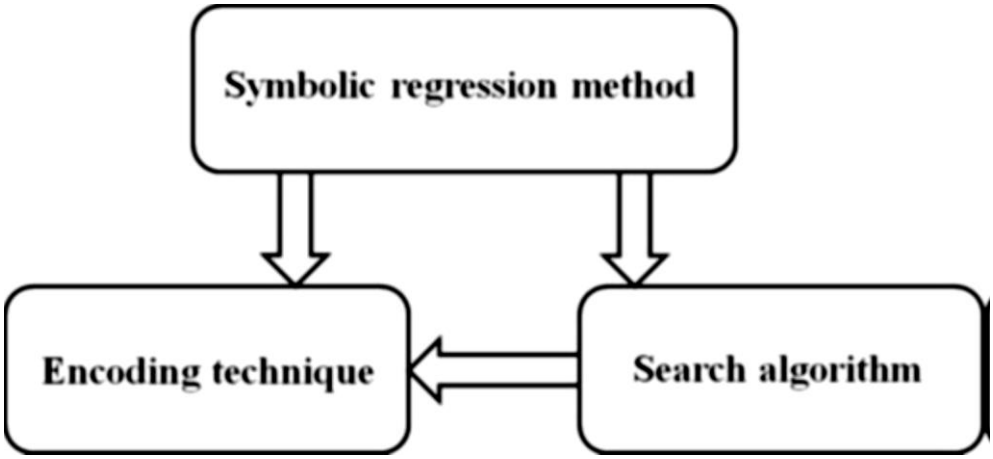

Figure 2.1. The overarching framework of symbolic regression approaches [192]

2.5.1. The Encoding Approach

The application of symbolic regression methodologies for encoding equations demands the prior initialization with a symbolic primitive alphabet — a finite set encompassing elementary mathematical functions and the independent variables of the target expression. The foundational methodology for the bidirectional transformation between symbolic expressions and their encoded representations entails the decomposition of the elementary function set into subsets based on the number of arguments they require.

The number of input arguments accepted by a function dictates how elementary function sets may be compositionally combined. The basic sets are as follows:

- The arguments set, or functions set without arguments

$$\boldsymbol{F_0} = \left\{f_{0,1}, \dots, f_{0,r+m_q+v}\right\} = \left\{x_1, \dots, x_r, \mathrm{q}_1, \dots, \mathrm{q}_{m_q}, e_1, \dots, e_v\right\}, \tag{2.52}$$

where $x_1, \dots, x_r$ stand for the variables, $\mathrm{q}_1, \dots, \mathrm{q}_{m_q}$ stand for the parameters, $e_1, \dots, e_v$ stand for the unit elements for two-argument functions;

- The functions set that is characterized by one argument

$$\boldsymbol{F_1} = \left\{f_{1,1}(z) = z, f_{1,2}(z), \dots, f_{1,w}(z)\right\}, \tag{2.53}$$

where $f_{1,1}(z)$ stands for a common identity function, which is typically required for coding;

- The functions set that is characterized by two arguments

$$\boldsymbol{F_2} = \{f_{2,1}(z_1, z_2), \dots, f_{2,n}(z_1, z_2)\}. \quad (2.54)$$

All two-argument functions must have the following features:

- commutative

$$f_{2,i}(z_1, z_2) = f_{2,i}(z_2, z_1), \ i = 1, \dots, n. \quad (2.55)$$

- associative

$$f_{2,i}\left(z_1, f_{2,i}(z_2, z_3)\right) = f_{2,i}\left(f_{2,i}(z_1, z_2), z_3\right), \quad (2.56)$$

- possess a unit element

$$f_{2,i}(z, e_i) = f_{2,i}(z_i, z) = z, \quad (2.57)$$

where e_i is a function's unit element, $i = 1, \dots, v$.

In order for solving the problems of machine learning control and while finding unknown functions, it becomes crucial to establish the principles governing the construction of mathematical expressions using primary functions. One widely applicable method for composing mathematical expressions using primary functions involves representing a series of primary functions in the form of a composition of such functions nested within one another. To illustrate,

$$f_{c_1,d_1} f_{c_2,d_2} f_{c_3,d_3} = f_{c_1,d_1} \circ f_{c_2,d_2} \circ f_{c_3,d_3} = f_{c_1,d_1}(f_{c_2,d_2}(f_{c_3,d_3}(\dots))). \quad (2.58)$$

It is essential to acknowledge that the representation utilized in symbolic regression techniques corresponds entirely with the principles outlined in the Kolmogorov-Arnold theory regarding the function's representation. According to this theory, if a function f is multidimensional and continuous, it can be expressed in the form of a finite composition consisting of continuous functions that involve a single variable and an addition binary operation [192].

2.5.2. The Search Algorithm

The methodology of symbolic regression within machine learning involves the identification of diverse functional architectures and optimization of their associated parameters concurrently for an unknown function. This dual-objective search is predominantly executed via genetic algorithms, which operate over a hybrid search space comprising symbolic structures and real-valued parameter vectors.

The genetic algorithm (Figure 2.2) has demonstrated its usefulness over a significant period of time [199–201]. One notable characteristic of the genetic algorithm includes its capacity to function inside a space of codes. The process of searching within the code space presents a challenge due to the discrepancy between the metric employed in the code space and the metric used in the calculation of the objective functional within the vector numeric space.

The genetic algorithm, due to its distinct structure, has the ability to conduct searches in non-numerical spaces. The foundational distinction of genetic algorithms resides in their operator set — comprising selection, crossover, and mutation — which operates independently of arithmetic or analytical operations. This enables their applicability to non-numerical optimization domains, including symbolic regression in machine learning and control systems.

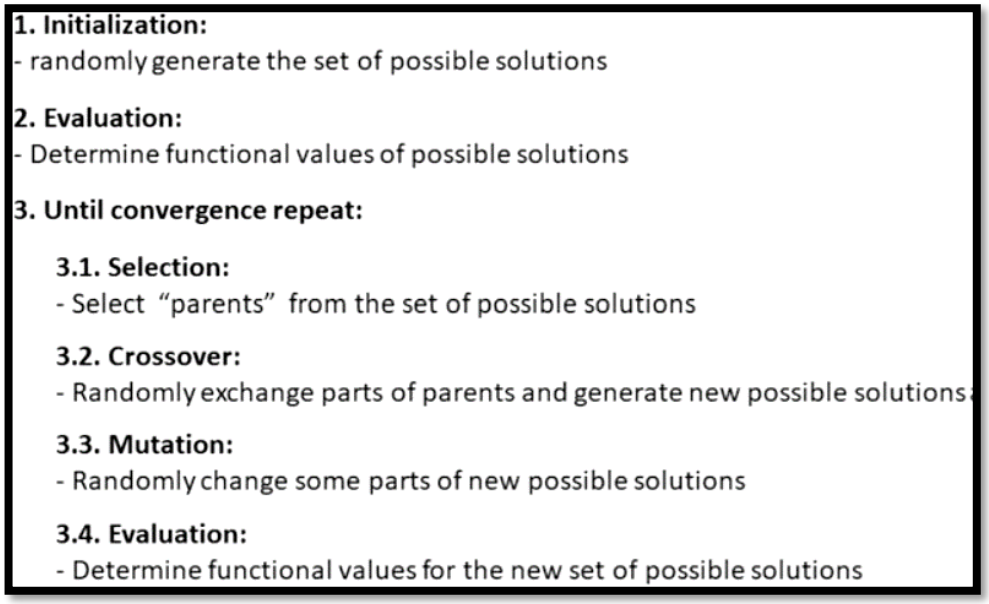

Figure 2.2. The genetic algorithm mechanism [192]

2.6. The Small Variations Principle within the Basic Solution

The complexity of finding an effective solution within the space of encoded codes is attributed to the classification of this work as a non-numerical optimization issue. The utilization of evolutionary algorithms involving arithmetic operations is not possible for search spaces of this nature. The predominant class of evolutionary algorithms employs arithmetic operators to manipulate candidate solutions numerically. In contrast, genetic algorithms constitute a principal search paradigm over symbolic or discrete-coded solution spaces, wherein all evolutionary operations are defined independently of arithmetic computation, relying instead on stochastic, structure-preserving genetic operators. Simultaneously, When symbolic regression employs sophisticated coding structures, the

derivation of appropriate crossover and mutation operators becomes a critical research problem. This has led to the formalization of the small variations principle, which constrains evolutionary variations to minimal changes from an established basic solution [202-203].

Contemplate a universal methodology for developing genetic algorithms to address non-numerical optimization issues. The methodology is centered upon the small variations' principle within the basic solution.

The fundamental tenets underlying this technique can be summarized as follows. An initial candidate solution, termed the basic solution, is often encoded to serve as a starting point for optimization. In complex scenarios, it is adequate to select this candidate solution that approximates the optimal outcome, as judged by the researcher, to reduce the time required for searching. Subsequently, The set of small variations is constructed so that every one applied to the basic solution code generates a structurally correct new solution. Furthermore, all such variations are represented in coded form for algorithmic execution. In the context of symbolic search, a small variation functions as a transformation operator applied to the code space of the basic solution, generating neighboring solutions through minimal structural modifications. Consequently, in all instances, a small variation code represents an integer vector that encompasses the essential information required to execute operations on the code in accordance with the operator of the small variation. The proposed approach benefits significantly from the existence of domain experts capable of constructing effective control systems through intuitive reasoning or extensive experience; these designed effective systems can be purposed as a basic solution.

In order to elucidate the concept of variation, it is necessary to introduce a vector denoting the extent of variation

$$\mathcal{W} = [\mathcal{w}_1 \dots \mathcal{w}_{dep}]^T, \tag{2.59}$$

where dep represents the dimension of the variation vector, specified by the information necessary to execute a small variation. This dimension is contingent upon the symbolic regression technique employed. As an illustration, let $\mathcal{w}_1$ represents an index denoting a small variation. Similarly, $\mathcal{w}_2$ and $\mathcal{w}_{dep-1}$ can be understood as indices indicating the element position in the code that define the variable element. Finally, $\mathcal{w}_{dep}$ represents the updated value of the defined element.

For instance, a small variation to the Cartesian genetic programming (CGP) code involves altering an element within the matrix. In order to implement a small variation, a three-element integer vector will do the trick

$$\mathcal{W} = [\mathcal{w}_1 \ \mathcal{w}_2 \ \mathcal{w}_3]^T, \tag{2.60}$$

where the variables $\mathcal{w}_1$, $\mathcal{w}_2$, and $\mathcal{w}_3$ correspond to the column identifier, the row position within that column, and the replacement value for the specified matrix element, respectively.

2.7. Variational Genetic Algorithm

The effective solution is sought using a genetic algorithm known as the variational genetic algorithm (VarGA), which operates in the ordered sets space of vectors with small variations to find the proper solution.

The genetic algorithm, in accordance with the small variations principle within the basic solution (VarGA), consists of the following sequential steps:

1. Define the basic solution such that this solution is deemed, based on the researcher's perspective, to be the most proximate to the potential effective solution.
$$\boldsymbol{b}^0 = [b_1^0 \dots b_n^0]^T. \tag{2.61}$$
2. Generate ordered multisets form consisting of variation vectors as the initial population
$$\boldsymbol{W}^i = \left(\mathcal{W}^{i,1}, \dots, \mathcal{W}^{i,D}\right), i = 1, \dots, l, \tag{2.62}$$
where in this given context, l represents the sequence of possible solutions within the initial population, whereas D is the total count of variation vectors present in a single set. The initial population is formed by subjecting the basic solution to a set of small variations, each yielding a candidate solution
$$\boldsymbol{b}^i = \boldsymbol{W}^i \circ \boldsymbol{b}^0 = \mathcal{W}^{i,D} \circ \mathcal{W}^{i,D-1} \circ \cdots \circ \mathcal{W}^{i,1} \circ \boldsymbol{b}^0, \tag{2.63}$$
where each potential solution inside the population is an element of the D-neighborhood of the basic solution
$$\boldsymbol{b}^i \in D(\boldsymbol{b}^0), i = 1, \dots, l. \tag{2.64}$$
3. Determine the objective function value for every possible solution within the population
$$F_i = J(\Psi(\boldsymbol{b}^i)), i = 1, \dots, l, \tag{2.65}$$
where $\Psi(\boldsymbol{b})$ serves as a decoding function that translates a structured, non-numeric representation into a computable real-valued function.
4. The evolution cycle is executed unless the condition of stop is met:
   I. Choose two sets of variations vectors at random
   $$\boldsymbol{W}_{\boldsymbol{\gamma}} = (\mathcal{W}^{\gamma,1}, \dots, \mathcal{W}^{\gamma,D}),\ \boldsymbol{W}_{\boldsymbol{\varphi}} = (\mathcal{W}^{\varphi,1}, \dots, \mathcal{W}^{\varphi,D}). \tag{2.66}$$
   II. The crossover's activation probability is computed based on the objective functional values of selected solution vectors

$$P_{r_c} = max\left\{\frac{F_{j-}}{F_\gamma}, \frac{F_{j-}}{F_\varphi}\right\} \quad (2.67)$$

If the generator of random number yields a value that is smaller than P_{r_c}, then the crossover process is executed.

Define the crossover point at random

$$k_c \in \{1, \dots, D\}. \quad (2.68)$$

Following the crossover point, swap the variations' vectors in the chosen sets to create two novel sets of variation vectors that signify two novel solutions from the basic solution's D-neighborhood

$$\boldsymbol{W_{\gamma+1}} = (\mathcal{W}^{\gamma,1}, \dots, \mathcal{W}^{\gamma,k}, \mathcal{W}^{\varphi,k+1}, \dots, \mathcal{W}^{\varphi,D}),$$
$$\boldsymbol{W_{\varphi+1}} = (\mathcal{W}^{\varphi,1}, \dots, \mathcal{W}^{\varphi,k}, \mathcal{W}^{\gamma,k+1}, \dots, \mathcal{W}^{\gamma,D}). \quad (2.69)$$

III. Execute the mutation process with a specified probability for the newly discovered possible solutions as sets of variations' vectors (2.69). Pick a mutation point at random, then create a new variations vector at that position.

IV. Each newly generated candidate solution undergoes evaluation using the objective functional. Its fate — inclusion (via replacement of the worst population member) or rejection — is determined by comparing its score against the current population's minimum performance threshold.

While built upon the small variations principle, this genetic algorithm preserves all canonical operations of traditional genetic algorithms, including crossover executed via tail-segment swap following a designated cut point. This algorithm can incorporate an additional loop to alter the basic solution. After executing a certain number of iterations to generate novel possible solutions, it is essential to substitute the basic solution with a possible solution chosen for the novel basis, which is determined to be the best option in terms of functionality.

The dual representation of this strategy —comprising both a basic solution and a vector of variations—may seem redundant but enables two critical improvements: (1) the basic solution enables rapid convergence in complex, multimodal optimization landscapes; (2) operating on variation vectors ensures that every evolved solution remains syntactically valid, thereby eliminating the risk of invalid outputs and reducing computational overhead associated with validation.

The principle of small variations constitutes a refinement strategy that may be systematically incorporated into any symbolic regression methodology to address the computational and structural complexities inherent in control synthesis problems.

2.8. Symbolic Regression Techniques

Multiple symbolic regression techniques are commonplace at the moment. So, here are a few examples: Genetic Programming (GP) [204], analytic programming [205], Cartesian GP [206], network operator method [207], parse-matrix evolution [208], and complete binary GP [209]. Unlike other symbolic regression techniques, the small-variations principle constitutes a novel contribution introduced exclusively within the network operator method, making it the sole representative of this class of evolutionary search. The extension of the small-variation principle to pre-existing symbolic regression paradigms gives rise to a class of augmented algorithms, each denoted by the prefix "variational," e.g., Variational Genetic Programming (VGP), Variational Cartesian GP. [203].

2.9. Synthesized Genetic Programming Technique (SGP)

The technique used in this study was created by the researcher. This technique is brand new, being the first instance in which this technique has been applied to solve the problem of control synthesis. Synthesized genetic programming (SGP) eschews the utilization of graphical representations for expressing codes of expressions.

2.9.1. Encoding Approach Using Synthesized Genetic Programming

The next mathematical expression is an example of how to encode it manually by synthesized genetic programming (SGP)

$$y = \exp(q_3 x_2^2 + q_1 x_3^2)\sin(q_2 x_1) + \cos(-q_3 x_3 + x_1). \tag{2.70}$$

where q_1, q_2 and q_3 exemplify the parameters, x_1, x_2 and x_3 exemplify variables, and both exemplify arguments of the mathematical expression.

The symbolic encoding of any mathematical expressions is rendered feasible through the utilization of the following predefined sets:

- The arguments set

$$\boldsymbol{F_0} = \{f_{0,1} = x_1, f_{0,2} = x_2, f_{0,3} = x_3, f_{0,4} = \mathrm{q}_1, f_{0,5} = \mathrm{q}_2, f_{0,6} = \mathrm{q}_3\}. \tag{2.71}$$

- The functions set that is characterized by one argument

$$\boldsymbol{F_1} = \{f_{1,1}(z) = z, f_{1,2}(z) = -z, f_{1,3}(z) = z^2, f_{1,4}(z) = \sin(z),$$

$$f_{1,5}(z) = \cos(z), f_{1,6}(z) = \exp(z)\}. \quad (2.72)$$

- The functions set that is characterized by two arguments

$$\boldsymbol{F_2} = \{f_{2,1}(z_1, z_2) = z_1 + z_2, f_{2,2}(z_1, z_2) = z_1 z_2\}. \quad (2.73)$$

In general, the mathematical expression's SGP code is a six-row integer matrix. The first row of the matrix denotes the indexes of functions belonging to the functions set that is characterized by two arguments (2.73). The indexes of functions from the functions set that is characterized by one argument (2.72) are represented by the second and fourth rows. The third and fifth rows represent the indexes of arguments from the arguments set (2.71). The sixth row represents the priority, which will thereafter be elucidated to elucidate its role. Within each column of the matrix, the second element (the one-argument function) and the third element (the argument) represent the first argument for the first element of the column (the two-argument function). Additionally, the fourth and fifth elements represent the second argument for the first element of the column. The term of the pivot for each column means either the first argument (the second and third elements) or the second argument (the fourth and fifth elements) of this column. The pivot can be determined by assigning the priority (the sixth element in the column) of 1 or 2 to opt for the desired pivot of the column. Even Nevertheless, in most contexts, its number is 1. It is important to acknowledge that the number of rows in the SGP matrix is contingent upon the number of arguments employed in the available functions. Specifically, when utilizing a three-argument function such as the if function, the number of rows is going to be 8. This is due to each argument being allocated two elements in the column, combined with the first element representing the three-argument function and the final element denoting the priority. After completing the calculation for each column, the result of this column should be appended to the set of arguments (2.71), progressively increasing the total number of arguments with each calculation.

In order to implement example (2.70) by this technique, let us get started by coding the expression $q_3 x_2^2$ as the first column of the SGP matrix. For the first element in this column, determine the index of multiplication function in the functions set that is characterized by two arguments (2.73); it is the number of 2, $f_{2,2}(z_1, z_2) = z_1 z_2$. For the second element, the function of the parameter q_3 is the identity function, $f_{1,1}(z) = z$, from the functions set that is characterized by one argument (2.72); the index of this function is 1. For the third element, the location of the parameter q_3 in the arguments set (2.71) is 6. For the fourth element, the variable x_2 function is the square $f_{1,3}(z) = z^2$, and its index is 3 in the set (2.72). For the fifth element, the location of the variable x_2 in the arguments set (2.71) is 2. The sixth element is the priority, and its number is 1. As a result, the code of the expression $q_3 x_2^2$ that represents the first column

in the matrix is [2 1 6 3 2 1]^{T}. After calculating this column, it will have been appended to the arguments set (2.71) as the seventh element, denoted as ($|\boldsymbol{F_0}| + 1 = 6 + 1 = 7$). The following expression $q_1 x_3^2$ will be the second column, which is the same idea as the first column, and its code is [2 1 4 3 3 1]^{T}. Consequently, it will be appended as the eighth element to the arguments set (2.71). The third column will demonstrate the amalgamation of the preceding two columns, specifically represented as $q_3 x_2^2 + q_1 x_3^2$. For the first element of the third column, the index of the addition function in the set (2.73) is 1. The identity function will be the primary function in this case for the first column ($q_3 x_2^2$) and the second column ($q_1 x_3^2$); therefore, the second and fourth elements will get number 1 as the identity function in the set (2.72). The indexes of the first column ($q_3 x_2^2$) and the second column ($q_1 x_3^2$) in the arguments set (2.71) are 7 and 8, respectively. So, the code of the third column is [1 1 7 1 8 1]^{T} and will have been appended to the arguments set (2.71) as the ninth element.

The final code of the SGP matrix, for example (2.70), can be expressed as:

$$\boldsymbol{R}_{SGP} = \begin{bmatrix} 2 & 2 & 1 & 2 & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 6 & 5 \\ 6 & 4 & 7 & 5 & 6 & 11 & 9 & 12 \\ 3 & 3 & 1 & 1 & 1 & 1 & 4 & 1 \\ 2 & 3 & 8 & 1 & 3 & 1 & 10 & 13 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2.74}$$

2.9.2. Search Algorithm for Synthesized Genetic Programming

Let us analyze the sequential procedures of the algorithm for SGP.

Initially, a set of codes is generated in a random manner, representing possible solutions

$$S = \{\boldsymbol{R}_1, \dots, \boldsymbol{R}_M\}, \tag{2.75}$$

where $\boldsymbol{R}$ is the code matrix, $\boldsymbol{R}_i = (\boldsymbol{r}^{i,1}, \dots, \boldsymbol{r}^{i,H})$, $i = 1, \dots, M$, $\boldsymbol{r}$ is a placeholder for any column in the matrix — each is a vector. So $\boldsymbol{r}^1$ is the first column-vector, $\boldsymbol{r}^2$ the second, and so on, H tells you how many such vectors (columns) there are.

For every possible structure associated with the mathematical expression, there is a random generation of a parameters vector

$$\mathrm{q}_i^j = \boldsymbol{\xi}(\mathrm{q}_i^+ - \mathrm{q}_i^-) + \mathrm{q}_i^-, \;\; i = 1, \dots, m_q, \;\; j = 1, \dots, M. \tag{2.76}$$

where $\boldsymbol{\xi}$ represents a random value drawn from the interval [0:1]; while q_i^+ and q_i^- denote the higher and lower bounds of the parameters, respectively. Additionally, m_q represents the dimensionality of the parameters vector.

The evaluation of every possible solution is conducted through the utilization of an objective function or a function of fitness

$$\boldsymbol{G} = \{\boldsymbol{g}_1 = J(\boldsymbol{R}_1, \mathbf{q}^1), \dots, \boldsymbol{g}_M = J(\boldsymbol{R}_M, \mathbf{q}^M)\}, \tag{2.77}$$

where $J\left(\boldsymbol{R}_i, \mathbf{q}^i\right)$ represents an objective function, and $\mathbf{q}^i$ represents a parameters vector, $i = 1, \dots, M$.

In the context of the problem of synthesis, the objective function is represented by the functional (2.46). In order to determine the objective function value, the formulated code of the control function, in addition to the parameter vector, must be entered into the control object model (2.27). Subsequently, the system (2.44) is subjected to integration, followed by the calculation of the functional value (2.46).

The best solution R_{i^-} is specified

$$\boldsymbol{g}_{i^-} = min\{\boldsymbol{g}_1, \dots, \boldsymbol{g}_M\}. \tag{2.78}$$

For the operation of crossover, random two possible solutions $(\boldsymbol{R}_\gamma, \mathbf{q}^\gamma)$ and $(\boldsymbol{R}_\varphi, \mathbf{q}^\varphi)$ are chosen, $\gamma, \varphi \in \{1, \dots, M\}$.

A probability of executing the operation of crossover is determined

$$P_c = \max\left\{\frac{\boldsymbol{g}_i -}{\boldsymbol{g}_\gamma}, \frac{\boldsymbol{g}_i -}{\boldsymbol{g}_\varphi}\right\}. \tag{2.79}$$

A random value $\boldsymbol{\xi}$ is produced, generally distributed between 0 and 1. If this value is below P_c, the operation of crossover is executed.

Two crossover operation points are selected at random

$$k_1 \in \{1, \dots, H\}, \quad k_2 \in \{1, \dots, m_q\}, \tag{2.80}$$

the first point pertains to the structural portion, while the other one pertains to the parametric portion.

The application of a crossover operation yields a total of four novel possible solutions

$$\mathbf{q}^{M+1} = [\mathrm{q}_1^\gamma, \dots, \mathrm{q}_{k_2}^\gamma, \mathrm{q}_{k_{2+1}}^\varphi, \dots, \mathrm{q}_{m_q}^\varphi]^T,$$

$$\boldsymbol{R}_{M+1} = (\boldsymbol{r}^{\gamma,1}, \dots, \boldsymbol{r}^{\gamma,k_1}, \boldsymbol{r}^{\varphi,k_{2+1}}, \dots, \boldsymbol{r}^{\varphi,H}),$$

$$\mathbf{q}^{M+2} = [\mathrm{q}_1^\varphi, \dots, \mathrm{q}_{k_2}^\varphi, \mathrm{q}_{k_{2+1}}^\gamma, \dots, \mathrm{q}_{m_q}^\gamma]^T,$$

$$\boldsymbol{R}_{M+2} = (\boldsymbol{r}^{\varphi,1}, \dots, \boldsymbol{r}^{\varphi,k_1}, \boldsymbol{r}^{\gamma,k_{2+1}}, \dots, \boldsymbol{r}^{\gamma,H}),$$

$$\mathbf{q}^{M+3} = [\mathrm{q}_1^{\gamma}, \dots, \mathrm{q}_{k_2}^{\gamma}, \mathrm{q}_{k_{2+1}}^{\varphi}, \dots, \mathrm{q}_{m_q}^{\varphi}]^T,$$

$$\boldsymbol{R}_{M+3} = \boldsymbol{R}_{\gamma},$$

$$\mathbf{q}^{M+4} = [\mathrm{q}_1^{\varphi}, \dots, \mathrm{q}_{k_2}^{\varphi}, \mathrm{q}_{k_{2+1}}^{\gamma}, \dots, \mathrm{q}_{m_q}^{\gamma}]^T,$$

$$\boldsymbol{R}_{M+4} = \boldsymbol{R}_{\varphi}. \tag{2.81}$$

Four offspring are generated: two resulting from the simultaneous recombination of structural and parametric elements, and two from the recombination of parametric elements only, with structural components held constant.

Following the crossover, a subsequent mutation operation is executed with a certain probability P_{μ}. A random value $\boldsymbol{\xi}$ is produced, generally distributed between 0 and 1. If this value is fewer than P_{μ}, the mutation operation is implemented.

The selection of mutation points is conducted with regard to both structural and parametric portions

$$\mu_1 \in \{1, \dots, H\}, \quad \mu_2 \in \{1, \dots, m_q\}, \tag{2.82}$$

New values at μ_1 and μ_2 points are being generated

$$r_{e1}^{M+1,\mu_1} \in |\boldsymbol{F_2}|,$$

$$r_{e2}^{M+1,\mu_1} \quad \text{and} \quad r_{e4}^{M+1,\mu_1} \in |\boldsymbol{F_1}|,$$

$$r_{e3}^{M+1,\mu_1} \quad \text{and} \quad r_{e5}^{M+1,\mu_1} \in |\boldsymbol{F_0} + \mu_1 - 1|,$$

$$\mathrm{q}_{\mu_2}^{M+1} = \boldsymbol{\xi}\left(\mathrm{q}_{\mu_2}^{+} - \mathrm{q}_{\mu_2}^{-}\right) + \mathrm{q}_{\mu_2}^{-}. \tag{2.83}$$

where r_e represents any element in the column of the code matrix, for instance, r_{e1} is the first element in the column, $\boldsymbol{F_0}$ represents the arguments set, $\boldsymbol{F_1}$ represents the functions set that is characterized by one argument, $\boldsymbol{F_2}$ represents the functions set that is characterized by two arguments.

Afterwards, the first novel possible solution gets estimated based on the presented criterion

$$f_{M+1} = J(\boldsymbol{R}_{M+1}, \mathbf{q}^{M+1}). \tag{2.84}$$

Following that, the worst solution within the population is identified

$$f_{j^+} = \max\{f_1, \dots, f_M\}. \tag{2.85}$$

If the first novel solution exhibits better results compared to the worst solution within the population

$$f_{M+1} < f_{j^+}, \tag{2.86}$$

then, the first novel solution is substituted with the worst solution within the population

$$\mathbf{q}^{j^+} \leftarrow \mathbf{q}^{M+1},$$

$$\boldsymbol{R}_{j^+} \leftarrow \boldsymbol{R}_{M+1}. \tag{2.87}$$

The previously mentioned actions (5.84)–(5.87), are iteratively performed for other novel possible solutions $(\boldsymbol{R}_{M+2}, \mathbf{q}^{M+2})$, $(\boldsymbol{R}_{M+3}, \mathbf{q}^{M+3})$ and $(\boldsymbol{R}_{M+4}, \mathbf{q}^{M+4})$.

2.10. Variational Synthesized Genetic Programming (VSGP)

Mirroring the encoding scheme of Cartesian GP, this technique employs a minimalistic a triplet of integers vector to typically identify each applied small variation

$$\mathcal{W} = [\mathcal{w}_1 \;\; \mathcal{w}_2 \;\; \mathcal{w}_3]^T, \tag{2.88}$$

where the variables $\mathcal{w}_1$, $\mathcal{w}_2$, and $\mathcal{w}_3$ correspond to the column identifier, the row position within that column, and the replacement value for the specified matrix element, respectively. If $\mathcal{w}_2$ equals 1, the subsequent number ($\mathcal{w}_3$) must either be zero or modified based on the functions set that is characterized by two arguments (2.73). If $\mathcal{w}_2$ is equal to either 2 or 4, then $\mathcal{w}_3$ will be modified to either zero or selected from the functions set that is characterized by one argument (2.72). If $\mathcal{w}_2$ is equal to either 3 or 5, then $\mathcal{w}_3$ can either be set to zero or can only be determined by the combination of the number of arguments (2.71) and the number of columns minus one. Certain conditions dictate the implementation of small variations to the SGP matrix based on the pivot and priority. These requirements can be elucidated by implementing the following variations to the matrix (2.74):

$$\begin{aligned}
\mathcal{W}^1 &= [3 \;\; 6 \;\; 2]^T, \\
\mathcal{W}^2 &= [5 \;\; 2 \;\; 0]^T, \\
\mathcal{W}^3 &= [4 \;\; 1 \;\; 1]^T, \\
\mathcal{W}^4 &= [6 \;\; 5 \;\; 0]^T, \\
\mathcal{W}^5 &= [3 \;\; 1 \;\; 0]^T, \\
\mathcal{W}^6 &= [8 \;\; 2 \;\; 3]^T, \\
\mathcal{W}^7 &= [6 \;\; 6 \;\; 2]^T,
\end{aligned} \tag{2.89}$$

The updated matrix of the SGP will look like

$$\mathcal{W}^1 \circ \mathcal{W}^2 \circ \mathcal{W}^3 \circ \mathcal{W}^4 \circ \mathcal{W}^5 \circ \mathcal{W}^6 \circ \mathcal{W}^7 \circ R_{SGP} = \begin{bmatrix} 2 & 2 & \mathbf{0} & \mathbf{1} & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 6 & \mathbf{3} \\ 6 & 4 & 7 & 5 & 6 & 11 & 9 & 12 \\ 3 & 3 & 1 & 1 & 1 & 1 & 4 & 1 \\ 2 & 3 & 8 & 1 & 3 & \mathbf{0} & 10 & 13 \\ 1 & 1 & \mathbf{2} & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \quad (2.90)$$

The first variation $\mathcal{W}^1$ has changed the third column's priority (the 6th element) from 1 to 2. Consequently, it has changed the pivot from the first argument (the 2nd and 3rd elements) to the second argument (the 4th and 5th elements) of this column. It is worth noting that the second variation $\mathcal{W}^2$ did not affect the 5th column since it is not possible to alter any element of the pivot to zero in each column of the matrix, where the pivot of the 5th column is the first argument (the 2nd and 3rd elements) because of the number of the priority is 1 in this column. In contrast, the fourth variation $\mathcal{W}^4$ can be accomplished since the 5th element of the 6th column is not an element of the pivot in this column, where the expression of this column was $-q_3x_3 + x_1$, the variable x_1 has been neglected. In this case, the unit element of the function is used as the second argument where the unit element of the addition function is 0 and for the multiplication function is 1, so the expression has become $(-q_3x_3 + 0)$. Interestingly, the fifth variation $\mathcal{W}^5$ has fulfilled a primary change, where the expression of this column was $q_3x_2^2 + q_1x_3^2$. This variation cancelled the addition function (changed the first element to 0). The number of priority turned out to be 2 as a result of the first variation, so the new expression of this column has got the expression of the pivot (the 2nd argument), whose code is $[1\ \ 8]^{\mathrm{T}}$ that represents $q_1x_3^2$ (the identity function and the expression of the second column). The seventh variation $\mathcal{W}^7$ was not accomplished since the fourth variation has changed the 5th element in the 6th column to 0, and changing the priority means changing the pivot of the column, and the pivot element is not allowed to be zero. Eventually, the third and sixth variations $\mathcal{W}^3$ and $\mathcal{W}^6$ can be performed directly.

This new matrix can be expressed mathematically as

$$y = \exp(q_1x_3^2)\sin(q_2 + x_1) + (-q_3x_3)^2. \quad (2.91)$$

As mentioned above, the analysis highlights the crucial importance of the priority, which may be summed up as follows: the main task of the priority is to pick the pivot for each column. Moreover, it effectively avoids zero values in the pivot elements due to small variations. Additionally, it has the likelihood to decrease the length of mathematical expressions.

The application of steps of variational genetic algorithm to synthesized genetic programming technique is as mentioned in section 2.7.

2.11. The synthesized genetic programming as a distinct and modern technique

The Synthesized Genetic Programming technique (SGP) is regarded as a distinct technology separate from Genetic Programming (GP) and Cartesian Genetic Programming (CGP). This differentiation arises from divergences in encoding and decoding processes, code type, and the approach to implementing the principle of small variations within the basic solution. The significant distinctions among these three techniques can be exemplified as follows:

The next mathematical expression is an example of how to encode it manually by three methods of symbolic regression

$$z = \sin(\cos(\exp(q_1 x_1 + q_2 x_2^2))). \tag{2.92}$$

To encode this mathematical equation, the following fundamental sets need to be used:

- The arguments set

$$\boldsymbol{F_0} = \{f_{0,1} = x_1,\ f_{0,2} = x_2,\ f_{0,3} = \mathrm{q}_1, f_{0,4} = \mathrm{q}_2,\ f_{0,5} = 0, f_{0,6} = 1\}; \tag{2.93}$$

- The functions set that is characterized by one argument

$$\boldsymbol{F_1} = \{f_{1,1}(y) = y, f_{1,2}(y) = y^2, f_{1,3}(y) = \sin(y),$$

$$f_{1,4}(y) = \cos(y), f_{1,5}(y) = \exp(y)\}; \tag{2.94}$$

- The functions set that is characterized by two arguments

$$\boldsymbol{F_2} = \{f_{2,1}(y_1, y_2) = y_1 + y_2, f_{2,2}(y_1, y_2) = y_1 y_2\}. \tag{2.95}$$

2.11.1. Genetic Programming Technique (GP)

In this technique, the mathematical expression's structure is represented as a computational tree. In this structure, functions are represented by nodes, while the arguments of mathematical expressions are represented by leaves. The fundamental sets (2.93)-(2.95) are required for depicting the computational tree corresponding to the mathematical expression (2.92), as illustrated in Figure 2.3.

Furthermore, the mathematical expression (2.92) can be reformulated utilizing the fundamental sets (2.93)-(2.95) as the record:

$$z = f_{1,3}\left(f_{1,4}\left(f_{1,5}\left(f_{2,1}\left(f_{2,2}(f_{0,3}, f_{0,1}), f_{2,2}\left(f_{0,4}, f_{1,2}(f_{0,2})\right)\right)\right)\right)\right). \quad (2.96)$$

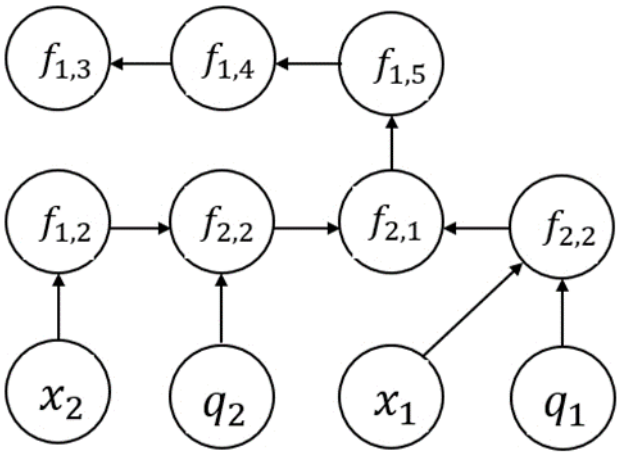

Figure 2.3. The computational tree of example (2.92) by GP

In the context of a mathematical expression, if an argument is present multiple times, it necessitates an equivalent number of occurrences on the leaves within the computational tree structure.

The computational tree is preserved within the memory of the computer as arranged sets of integer vectors, each comprising two elements. The initial element designates the count of arguments, while the subsequent element delineates the sequence of functions. The GP code corresponding to example (2.92) is as follows:

$$\boldsymbol{R}_{GP} = \left(\begin{bmatrix}1\\3\end{bmatrix}, \begin{bmatrix}1\\4\end{bmatrix}, \begin{bmatrix}1\\5\end{bmatrix}, \begin{bmatrix}2\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\3\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}2\\2\end{bmatrix}, \begin{bmatrix}0\\4\end{bmatrix}, \begin{bmatrix}1\\2\end{bmatrix}, \begin{bmatrix}0\\2\end{bmatrix}\right). \quad (2.97)$$

The indices of elements across all branches of the computational tree, extending from the top node to the leaves, serve as the genetic programming code.

2.11.2. Cartesian Genetic Programming Technique (CGP)

Cartesian genetic programming (CGP) employs a non-graphical representation for expressing codes of expressions. This technique involves the integration of the two sets of fundamental functions into a unified set.

$$\boldsymbol{F} = \boldsymbol{F_1} \cup \boldsymbol{F_2}. \quad (2.98)$$

As a result, the sets (2.94) and (2.95) will be as one set and as shown:

$$\boldsymbol{F} = \{f_1(y) = y, f_2(y) = y^2, f_3(y) = \sin(y), f_4(y) = \cos(y),$$
$$f_5(y) = \exp(y), f_6(y_1, y_2) = y_1 + y_2,$$
$$f_7(y_1, y_2) = y_1 y_2\}. \quad (2.99)$$

CGP codes for mathematical expressions typically take the form of a three- or four-row integer matrix. The initial row of the matrix denotes the indexes of functions obtained from the set of fundamental functions (2.99). The fundamentals functions' set (2.99) consists of functions that have a maximum of two arguments. Consequently, encoding the matrix requires only three rows. It should be noted that the number of rows in the matrix varies depending on the number of arguments used for the available functions, where in the case of using the one-argument function, the third element in the column does not have any practical application. The remaining rows stand for the argument indices (2.93). When a column's calculation is complete, its result should be appended to the arguments (2.93). So, after each computation, there will be more arguments. As a result, the total number of arguments will grow with each new calculation.

In order to encode the example (2.92) by this technique, firstly, it is needed to encode the expression $q_1 x_1$ as the first column. The function is multiplication, as its sequence is 7 in set (2.99), $f_7(y_1, y_2) = y_1 y_2$. Then, from the set of arguments (2.93), the sequence of parameter q_1 is 3, and the sequence of variable x_1 is 1. As a result, the code of the first column for the matrix is $[7\ 3\ 1]^T$. The result of each elementary function determination is added to the list of arguments (2.93) every time, increasing the total number of arguments with each calculation. Subsequently, the sequence of the first column will be ($|\boldsymbol{F_0}| + 1 = 6 + 1 = 7$) in the set of arguments. Then, the code of expression x_2^2 is calculated as the second column, from the set (2.99), the sequence of $f_2(y) = y^2$ is 2. And the variable x_2 has the series of 2 in the set of arguments (2.93). The third element of this column is not utilized since the argument of this function is only one. So, it can be 2. The code of the second column for the matrix is $[2\ 2\ 2]^T$, and it will be added to the arguments set (2.93), and the sequence of this column will be ($|\boldsymbol{F_0}| + 2 = 6 + 2 = 8$) in the arguments set (2.93).

Typically, the solution of CGP's mathematical equation (2.92) is coded as

$$\boldsymbol{R}_{CGP} = \begin{bmatrix} 7 & 2 & 7 & 6 & 5 & 3 & 6 \\ 3 & 2 & 4 & 7 & 10 & 12 & 13 \\ 1 & 2 & 8 & 9 & 6 & 1 & 5 \end{bmatrix}. \quad (2.100)$$

2.11.3. Synthesized Genetic Programming Technique (SGP)

All pertinent details concerning this technique are elaborated upon in Section 2.9 of the current chapter. In order to implement example (2.92) by this technique, let us get started by coding the expression $q_1 x_1$ as the first column of the SGP matrix. For the first element in this column, determine the index of multiplication function in the functions set that is characterized by two arguments (2.95); it is the number of 2, $f_{2,2}(y_1, y_2) = y_1 y_2$. For the second element, the function of the parameter q_1 is the identity function, $f_{1,1}(y) = y$, from the functions set that is characterized by one argument (2.94); the index of this function is 1. For the third element, the location of the parameter q_1 in the arguments set (2.93) is 3. For the fourth element, the variable x_1 function is the identity function, $f_{1,1}(y) = y$, and its index is 1 in the set (2.94). For the fifth element, the location of the variable x_1 in the arguments set (2.93) is 1. The sixth element is the priority, and its number is 1. As a result, the code of the expression $q_1 x_1$ that represents the first column in the matrix is $[2\ \ 1\ \ 3\ \ 1\ \ 1\ \ 1]^{\mathrm{T}}$. After calculating this column, it will have been appended to the arguments set (2.93) as the seventh element, denoted as ($|\boldsymbol{F_0}| + 1 = 6 + 1 = 7$).

The final code of the SGP matrix, for example (2.92), can be expressed as:

$$\boldsymbol{R}_{SGP} = \begin{bmatrix} 2 & 2 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 5 & 4 & 3 \\ 3 & 4 & 7 & 9 & 10 & 11 \\ 1 & 2 & 1 & 1 & 1 & 1 \\ 1 & 2 & 8 & 5 & 5 & 5 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}. \tag{2.101}$$

The procedural application of the small variations' principle within the basic solution of genetic programming and Cartesian genetic programming techniques was delineated in reference [203]. Furthermore, the application of this principle to the synthesized genetic programming technique was expounded upon in Section 2.10 of the current chapter. Consequently, a discernible contrast emerges in the application of this principle across each of these three techniques.

Table 2.1 presents a comparative analysis of the key features pertaining to symbolic regression across the three techniques under consideration.

Table 2.1. The main features of three symbolic regression techniques (GP, CGP & SGP)

No.	The feature	The technique of symbolic regression		
		GP	**CGP**	**SGP**
1	Appearance	1992	2000	2024
2	The structure of mathematical expression	computational tree	no graph	no graph
3	The stored code in the memory of the computer	as arranged sets of integer vectors, each comprising two elements	a three- or four-row integer matrix.	a six-row integer matrix
4	The length of code for any mathematical expression and after each crossover operation	various (Needs more time for calculation)	constant	constant
5	Terminology employed within the code	No terms	No terms	innovative coding terms such as the pivot and the priority

It is noteworthy that the efficiency and rapid solution discovery capabilities of the synthesized genetic programming technique (SGP) in addressing the problems of control general synthesis have been demonstrated in comparison to Cartesian genetic programming and parse-matrix evolution techniques [210-211].

2.12. The Search for the Effective Position of Points

Following the solution of the problem of synthesis, it becomes essential to identify the effective position of points (2.50) inside the states' space. Evolutionary algorithms are employed in order to address this objective, as the quality criterion (2.31) exhibits non-convexity and non-unimodality in the coordinates space of points (2.50). The utilization of the particle swarm optimization (PSO) algorithm [212-213] serves as the basis for this study. The mentioned algorithm is currently well-recognized as a prominent evolutionary algorithm.

The PSO algorithm comprises the subsequent steps. Initially, the generation of a possible solution's initial set is done

$$\mathrm{q}_i^j = \boldsymbol{\xi}(\mathrm{q}_i^+ - \mathrm{q}_i^-) + \mathrm{q}_i^-, \;\; i = 1, \dots, m_q, \;\; j = 1, \dots, M. \tag{2.102}$$

where $\boldsymbol{\xi}$ represents a random value drawn from the interval [0:1]; while q_i^+ and q_i^- denote the higher and lower bounds of the parameter evolution for vector values, respectively. Additionally, m_q represents the dimensionality of the parametric vector and M represents the total number of vectors inside the initial population.

For every possible solution, a history vector is generated, initialized with a value of zero

$$v_i^j = 0, i = 1, \dots, m_q, \;\; j = 1, \dots, M. \tag{2.103}$$

Subsequently, the objective function values are estimated for every possible solution

$$f_j = F(\mathbf{q}^j), \;\; j = 1, \dots, M. \tag{2.104}$$

where $F(\mathbf{q})$ represents the objective function of this problem of optimization.

In addition, an evaluation is determined for every possible solution

$$\tilde{\mathrm{q}}_i^j = \begin{cases} \mathrm{q}_i^+, \text{if } \mathrm{q}_i^j + \sigma v_i^j > \mathrm{q}_i^+ \\ \mathrm{q}_i^-, \text{if } \mathrm{q}_i^j + \sigma v_i^j < \mathrm{q}_i^+ \\ \mathrm{q}_i^j + \sigma v_i^j, \text{otherwise} \end{cases}, \tag{2.105}$$

where σ represents an algorithm constant parameter,

$$v_i^j \leftarrow \alpha v_i^j + \gamma \boldsymbol{\xi}(\mathrm{q}_i^{j-} - \mathrm{q}_i^j) + \beta \boldsymbol{\xi}(\mathrm{q}_i^{j(r)} - \mathrm{q}_i^j), \tag{2.106}$$

α, γ, β represent algorithm constant parameters, $\mathrm{q}-$ represents the most efficient possible solution for now

$$f_{j^-} = \min\{f_1, \dots, f_M\}. \tag{2.107}$$

$\mathrm{q}^{j(c)}$ represents the most efficient possible solution, out of c randomly chosen possible solutions

$$f_{j(c)} = \min\{f_{j1}, \dots, f_{jc}\}, \;\; i = 1, \dots, m_q, \;\; j = 1, \dots, M. \tag{2.108}$$

The optimization problem's solution is the desired one that can be found after the provided evolution loops.

CHAPTER 3. RESULTS

3.1. Introduction

This chapter verifies the feasibility of the suggested synthesized optimal control technique described in chapter two using a mathematical model of a nonholonomic wheeled mobile robot (Khepera II).

The technique of variational synthesized genetic programming (VSGP) involves achieving stability of the control object concerning a specific point in the space of states and controlling the objects by altering the positions of the points of equilibrium.

3.2. Computational Experiment

The optimal control problem is defined for a system comprising a pair of non-holonomic mobile robots, whose positions must be dynamically adjusted within the plane to circumvent environmental obstacles such that their environment encompasses several static phase constraints. However, the complexity of the task is heightened by the existence of the dynamic phase constraints, as it necessitates ensuring the avoidance of collisions between the two robots.

The following is the form of the nonholonomic mobile robot mathematical model [214]:

$$\begin{aligned} \dot{x}_1 &= 0.5(u_1 + u_2)\cos(x_3), \\ \dot{x}_2 &= 0.5(u_1 + u_2)\sin(x_3), \\ \dot{x}_3 &= 0.5(u_1 - u_2), \end{aligned} \tag{3.1}$$

In the first step, the task of synthesizing a stabilization system is addressed in order to establish a stable state for the object.

For numerically solving the synthesis problem, it is necessary to establish a predetermined set of initial states:

$$\boldsymbol{X}_0 = \{\boldsymbol{x}^{0,1}, \dots, \boldsymbol{x}^{0,\mathbf{L}}\}. \tag{3.2}$$

One terminal state is established:

$$\mathbf{x}^* = [x_1^* \;\; x_2^* \;\; x_3^*]^T. \tag{3.3}$$

The subsequent quality criterion is

$$J_{syn} = \sum_{i=1}^{\mathbf{L}} (t_{f,i} + p_1 \|\mathbf{x}^* - \mathbf{x}(t_{f,i}, \mathbf{x}^{0,i})\|) \rightarrow \min, \tag{3.4}$$

where $t_{f,i}$ is a period characterized by the attainment of the terminal state (3.10) starting from the initial states (3.9), $i = 1, \ldots, \mathbf{L}$, **L** represents the total number of initial states, p_1 represents a weight coefficient, and

$$t_{f,i} = \begin{cases} t, & \text{if } t \le t^+ \text{ and } \|x^f - x(t)\| \le \varepsilon \\ t^+, & \text{otherwise} \end{cases}, \tag{3.5}$$

and

$$\|x^f - x(t)\| = \sqrt{\sum_{i=1}^{3} (x_i^f - x_i(t))^2}. \tag{3.6}$$

The first step represents the stabilization system synthesis that involves the search for and creation of a single control function:

$$\mathbf{u} = \mathbf{g}(\mathbf{x}^* - \mathbf{x}), \tag{3.7}$$

which guarantees the attainment of the minimum functional value (3.4) for all provided initial states (3.2).

One robot can solve the problem of control synthesis (3.1)–(3.7) because the pair of robots are identical to one another. This problem is solved using the variational synthesized genetic programming (VSGP) symbolic regression technique.

Case one: Eight initial states are provided:

$$X_0 = \left\{ \boldsymbol{x}^{0,0} = \left[-3 \;\; 3.5 \;\; \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3 \;\; -3.5 \;\; \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3 \;\; -3.5 \;\; \frac{5\pi}{16}\right]^T, \; \boldsymbol{x}^{0,3} = \right.$$
$$\left[3 \;\; 3.5 \;\; \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3 \;\; 3.5 \;\; -\frac{5\pi}{16}\right]^T, \; \boldsymbol{x}^{0,5} = \left[-3 \;\; -3.5 \;\; -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3 \;\; -3.5 \;\; -\frac{5\pi}{16}\right]^T, \; \boldsymbol{x}^{0,7} = $$
$$\left. \left[3 \;\; 3.5 \;\; -\frac{5\pi}{16}\right]^T \right\}. \tag{3.8}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^* \;\; x_2^* \;\; x_3^*]^T = [0 \;\; 0 \;\; 0]^T. \tag{3.9}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.10}$$

where

$$\tilde{u}_1 = \left(x_2^f - x_2\right)^{\frac{1}{3}}\left(x_1^f - x_1\right)q_1\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right)$$

$$+ \sin\left(\left(x_2^f - x_2\right)^{\frac{1}{3}}\left(x_1^f - x_1\right)q_1\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right)\right), \tag{3.11}$$

$$\tilde{u}_2 = \rho\left(q_3^2\left(x_1^f - x_1\right)\right) + 0.5 * q_3^2\left(x_1^f - x_1\right), \tag{3.12}$$

$$\rho(\mu) = \begin{cases} 0, & \text{if } |\mu| < \delta \\ sgn(\mu), & \text{otherwise} \end{cases} \tag{3.13}$$

$q_1 = 2.07946$, $q_2 = 2.63935$, $q_3 = 2.96333$, $\delta = 10^{-8}$. The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 2.26092$, where $\varepsilon = 0.01, t^+ = 2.5$ sec, $\mathbf{L} = 8$, and , $p_1 = 1$.

Figure 3.1 shows the trajectories taken by one robot as it moved from eight initial states (3.2) to the terminal state (3.3).

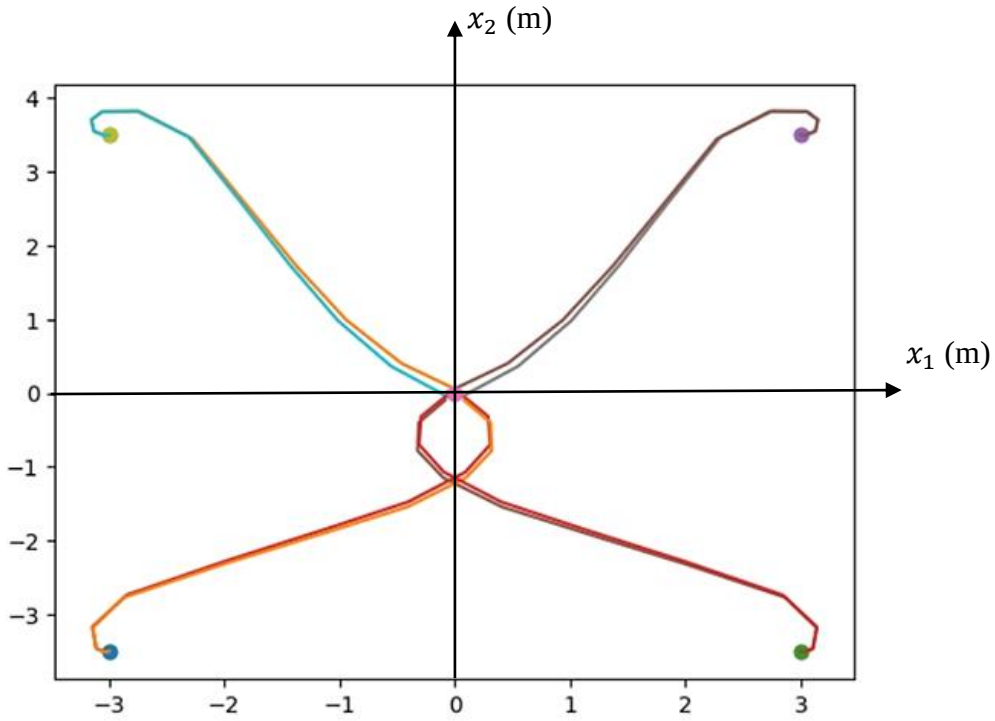

Figure 3.1. Robot trajectories with VSGP control function

The control functions (3.10) that have been acquired to guarantee the stability of the object are inserted into the model equations (3.1). The solution to the problem of control synthesis yields the emergence of a stable point of equilibrium in the space of state. The equilibrium point position is contingent upon the terminal vector (3.3).

Figures 3.2 through 3.9 depict the simulation results of a nonholonomic mobile robot, showcasing its behavior for displacement — meter, velocity — m/s, and control — m/s, in the y-axes and time — second in the x-axes, by the VSGP method.

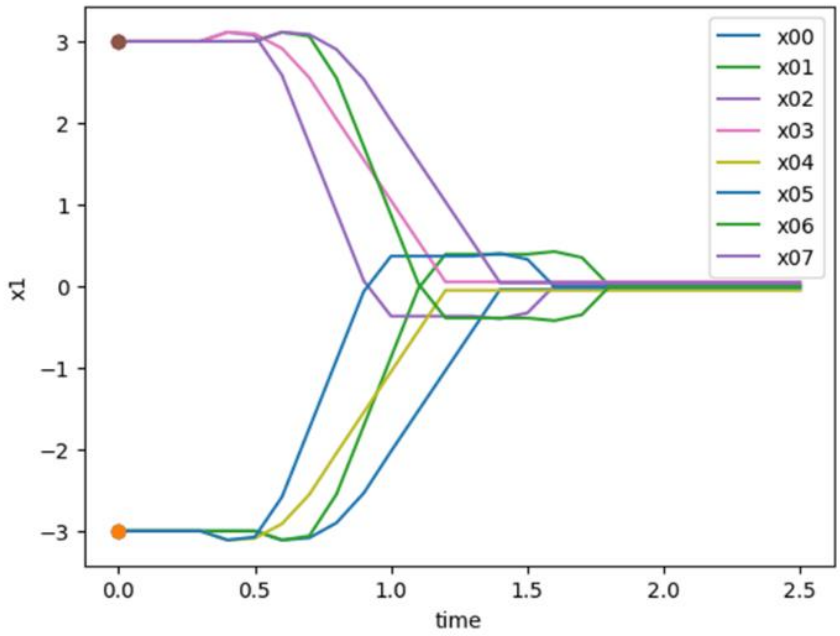

Figure 3.2. The robot displacements in direction x_1 from all initial conditions by VSGP method

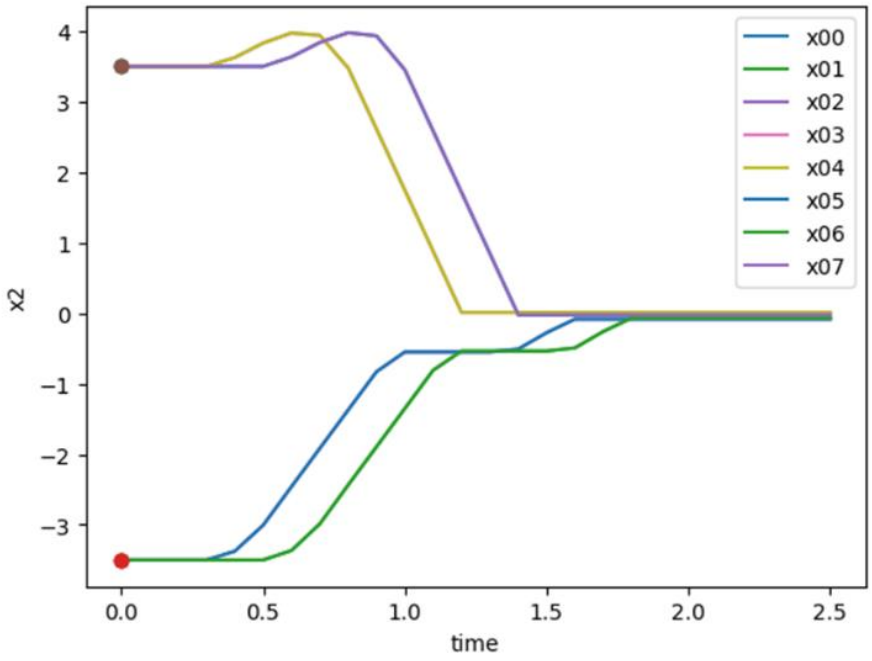

Figure 3.3. The robot displacements in direction x_2 from all initial conditions by VSGP method

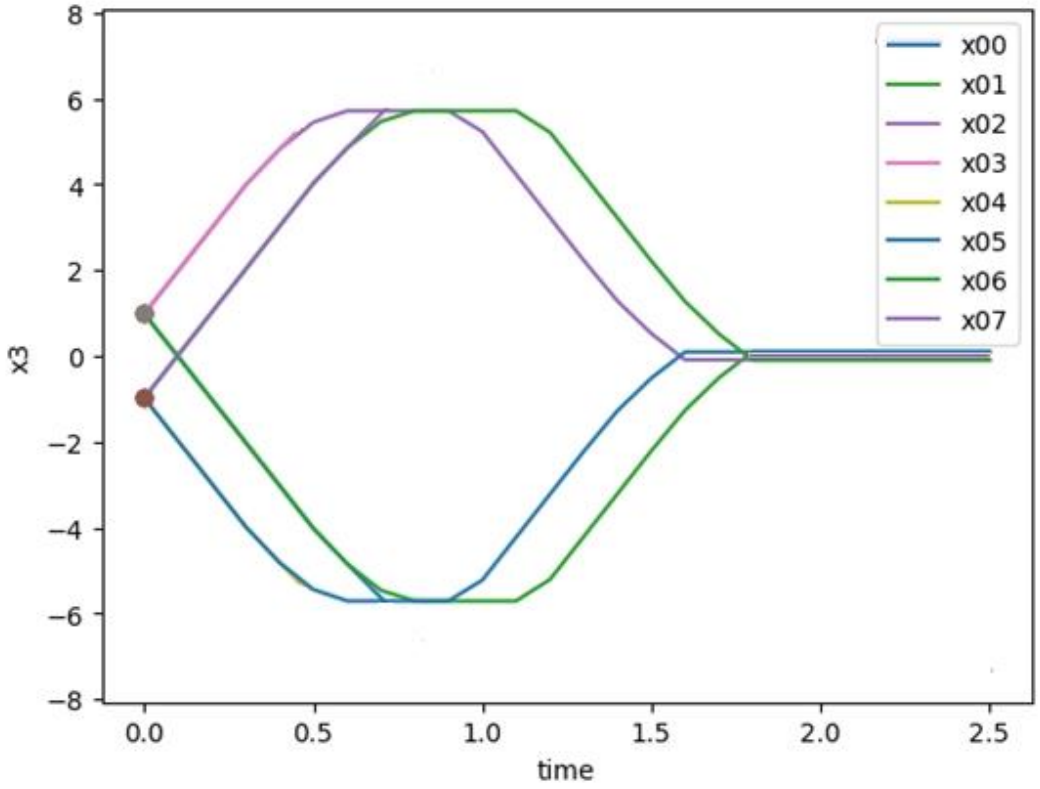

Figure 3.4. The robot displacements in direction x_3 from all initial conditions by VSGP method

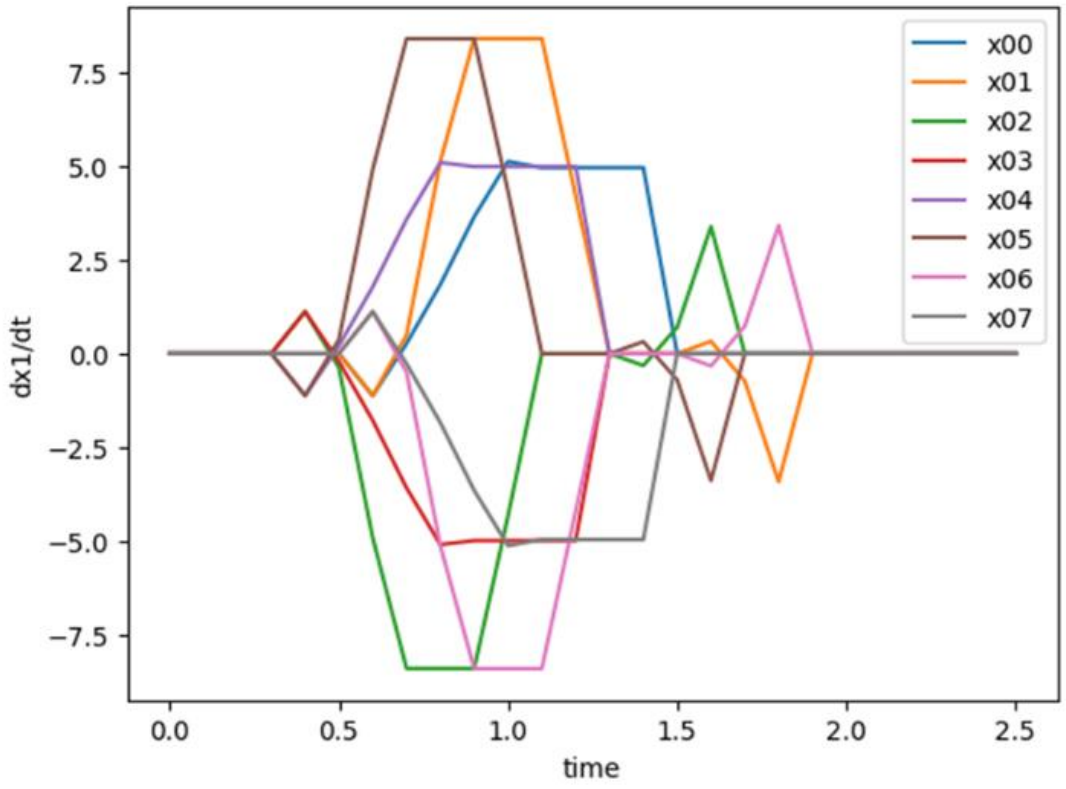

Figure 3.5. The robot velocities in direction x_1 from all initial conditions by VSGP method

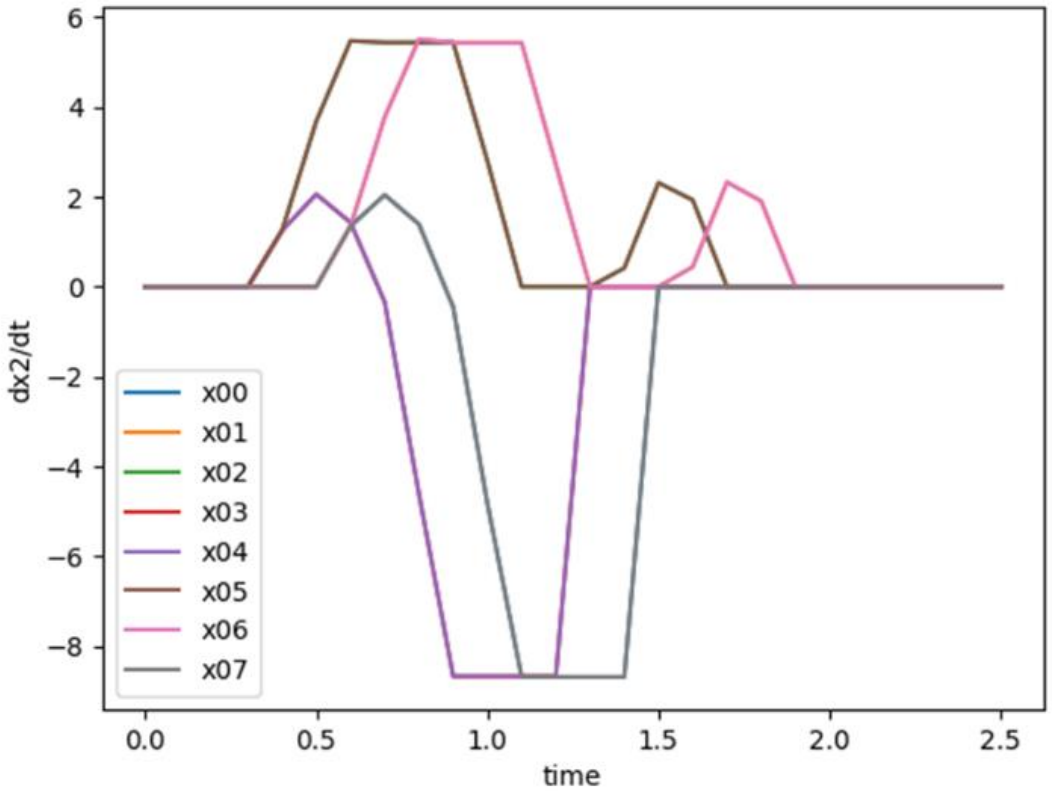

Figure 3.6. The robot velocities in direction x_2 from all initial conditions by VSGP method

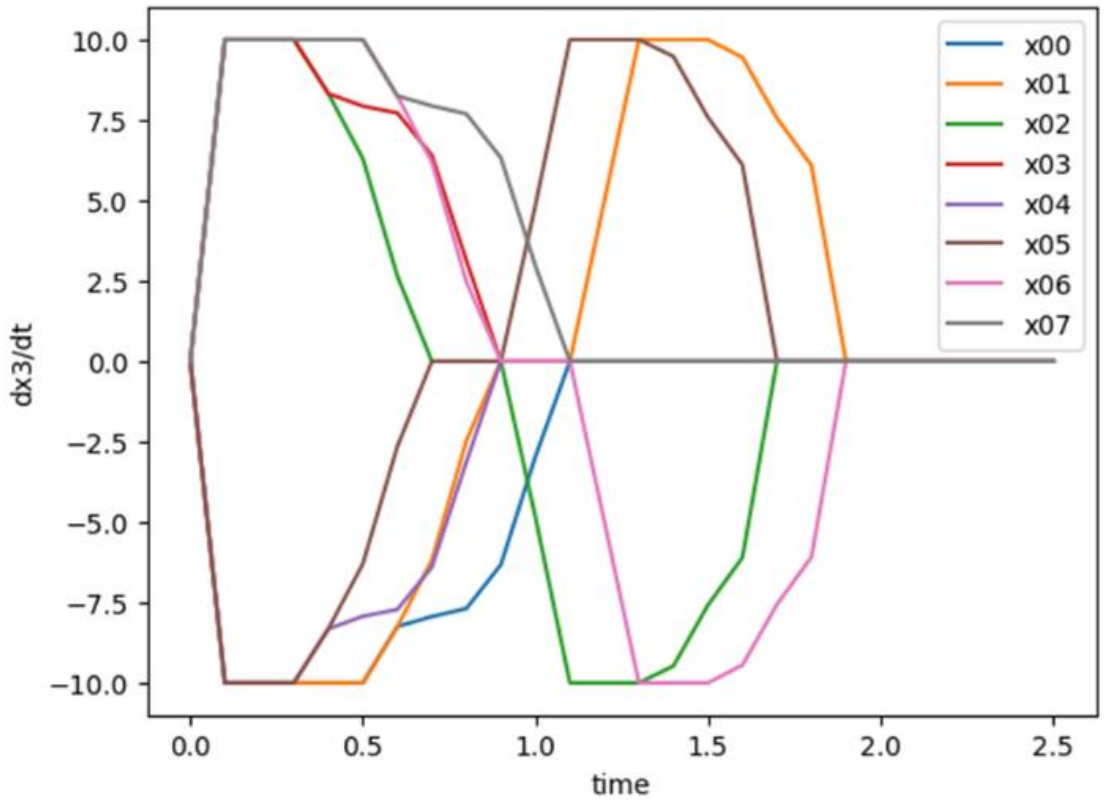

Figure 3.7. The robot velocities in direction x_3 from all initial conditions by VSGP method

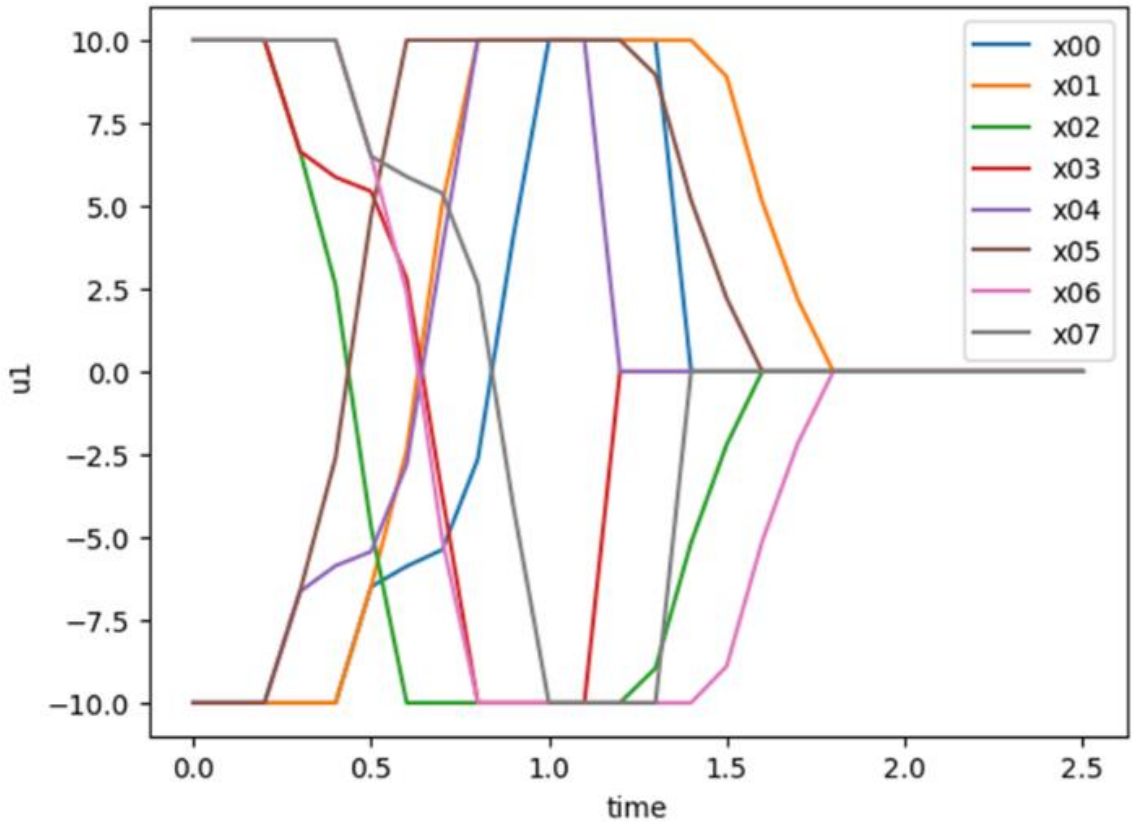

Figure 3.8. The robot control u_1 from all initial conditions by VSGP method

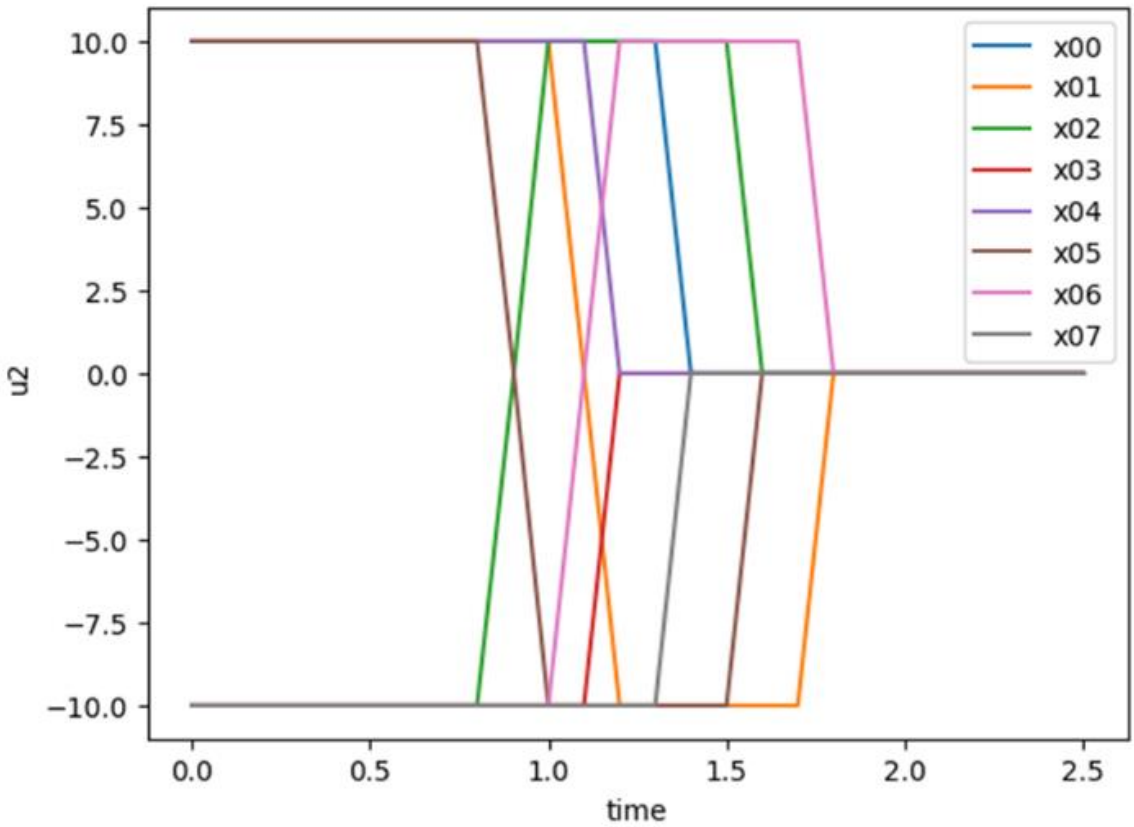

Figure 3.9. The robot control u_2 from all initial conditions by VSGP method

Case two: Eight initial states are provided:

$$X_0 = \left\{\boldsymbol{x}^{0,0} = \left[-3\ \ 3.5\ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3\ \ -3.5\ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ -3.5\ \frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,3} = \left[3\ \ 3.5\ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3\ \ 3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,5} = \left[-3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3\ \ -3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,7} = \left[3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T\right\}. \tag{3.14}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^*\ \ x_2^*\ \ x_3^*]^T = [0\ \ 0\ \ 0]^T. \tag{3.15}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.16}$$

where

$$\tilde{u}_1 = A * (1 - \exp(-A^2)) * \left(1 - \exp\left(-\left(A * (1 - \exp(-A^2))\right)^2\right)\right), \tag{3.17}$$

$$\tilde{u}_2 = (((2\left(x_1^f - x_1\right) * \left(1 - \exp\left(-\left(x_1^f - x_1\right)^2\right)\right) * \rho\left(q_2 + \exp\left(\left(x_1^f - x_1\right) + \left(x_1^f - x_1\right)^2\right) + \tanh(0.5 * q_3) * \left(x_2^f - x_2\right)\right)^3)^3)/2, \tag{3.18}$$

$$A = q_1\left(x_1^f - x_1\right) + \left(x_3^f - x_3\right)\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right), \tag{3.19}$$

$$\rho(\mu) = \begin{cases} 1, & \text{if } \mu \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.20}$$

$q_1 = 7.18441,\ \ q_2 = 7.12227,\ \ q_3 = 7.87202.$ The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 2.35184$, where $\varepsilon = 0.01, t^+ = 2.5$ sec, $\mathbf{L} = 8$, and , $p_1 = 1$.

Figure 3.10 shows the trajectories taken by one robot as it moved from eight initial states (3.14) to the terminal state (3.15).

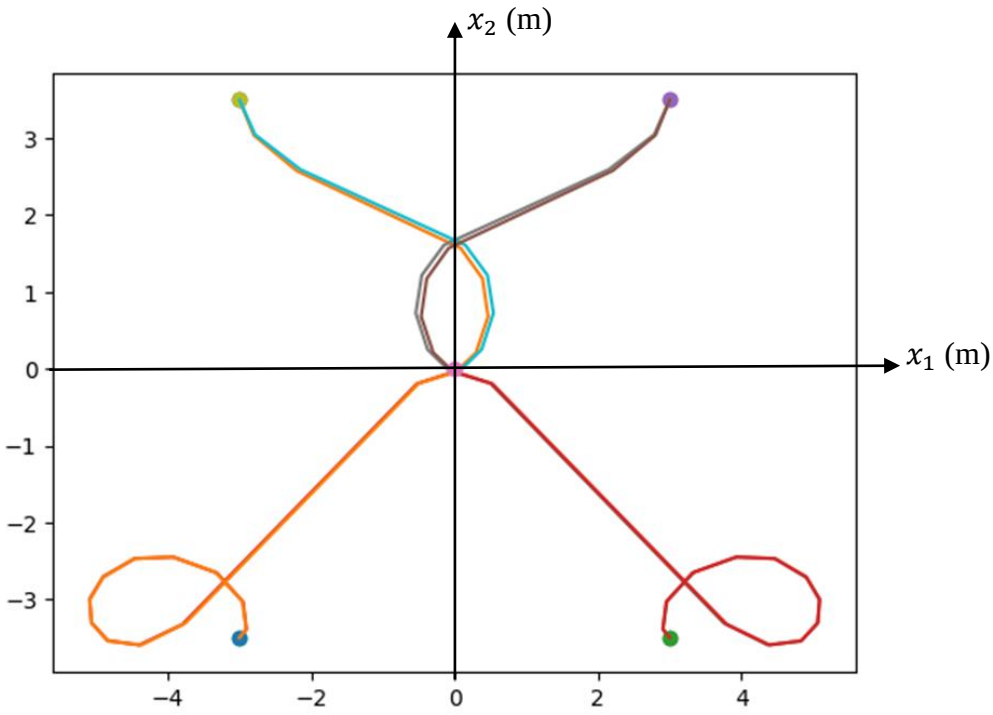

Figure 3.10. Robot trajectories with VSGP control function

Figures 3.11 through 3.16 depict the simulation results of a nonholonomic mobile robot, showcasing its behavior for displacement — meter, velocity — m/s, and control — m/s, in the y-axes and time — second in the x-axes, by the VSGP method.

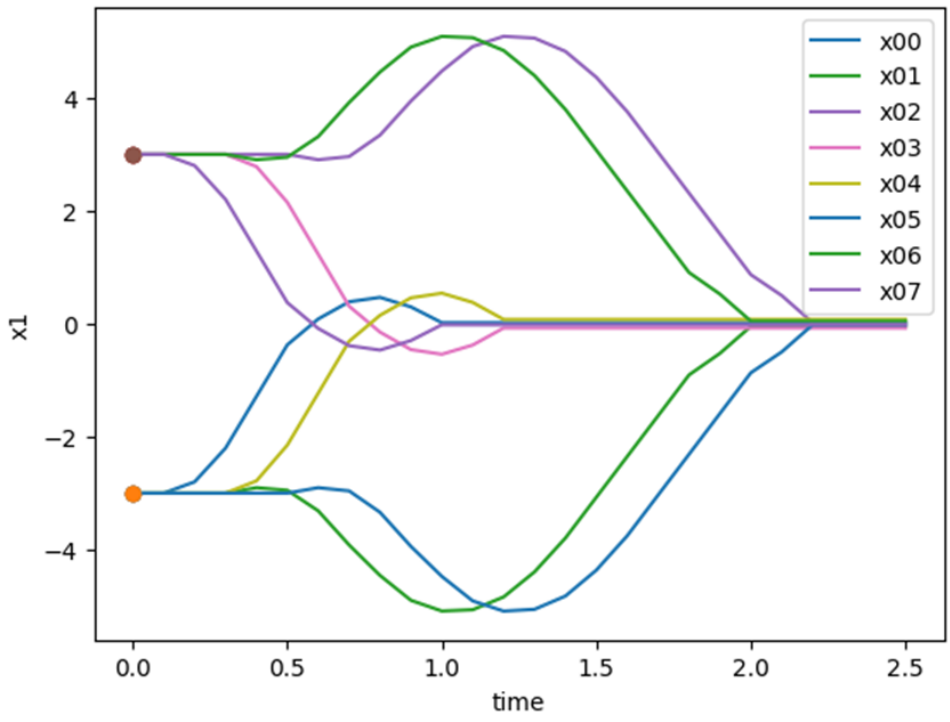

Figure 3.11. The robot displacements in direction x_1 from all initial conditions by VSGP method

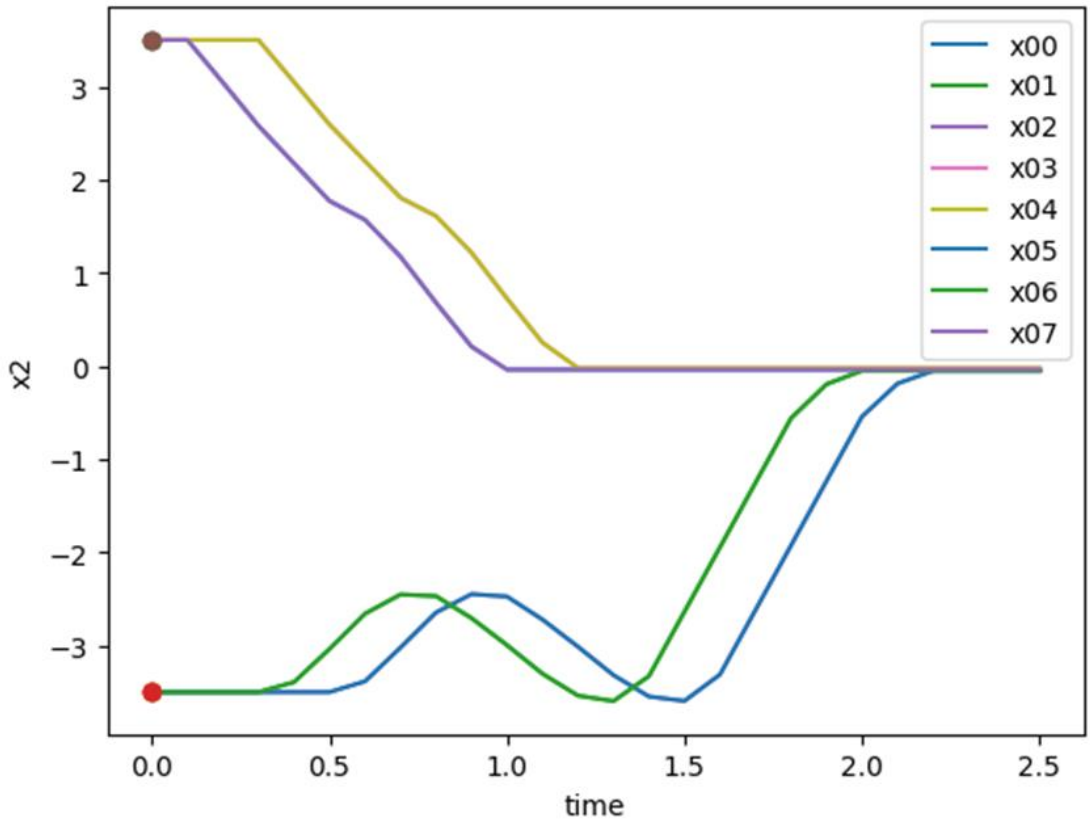

Figure 3.12. The robot displacements in direction x_2 from all initial conditions by VSGP method

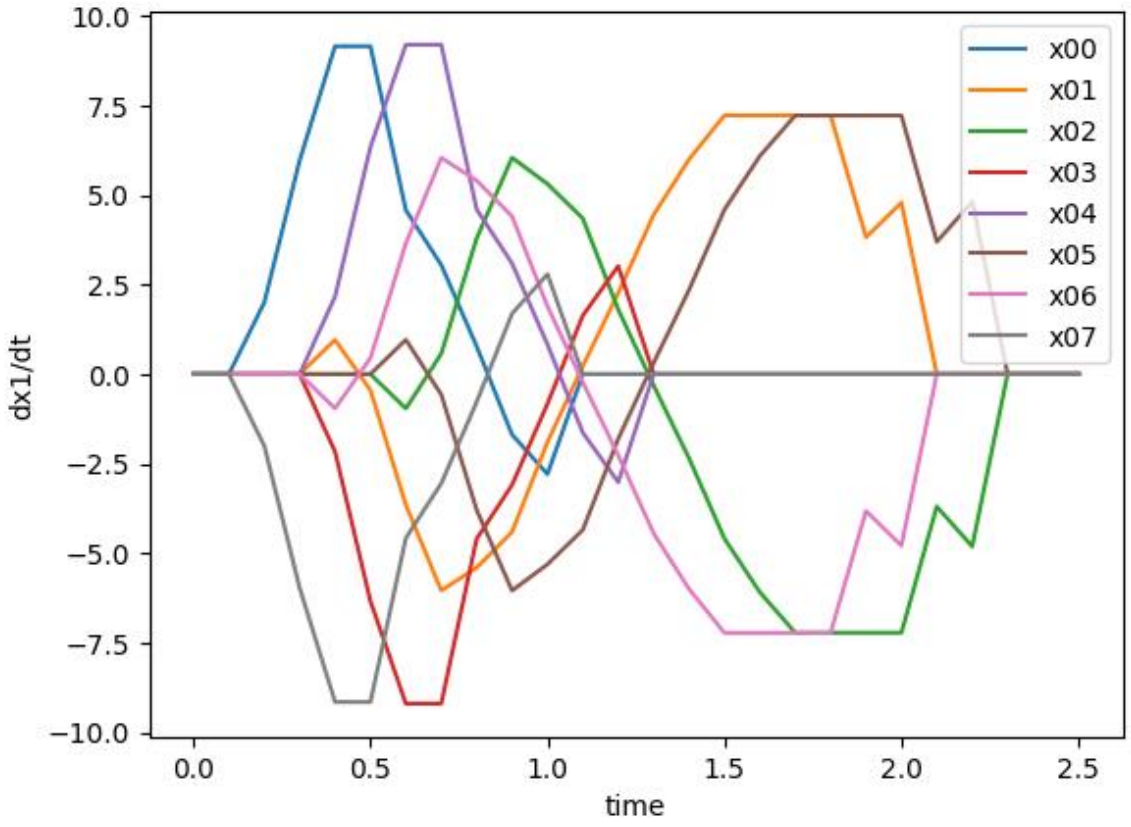

Figure 3.13. The robot velocities in direction x_1 from all initial conditions by VSGP method

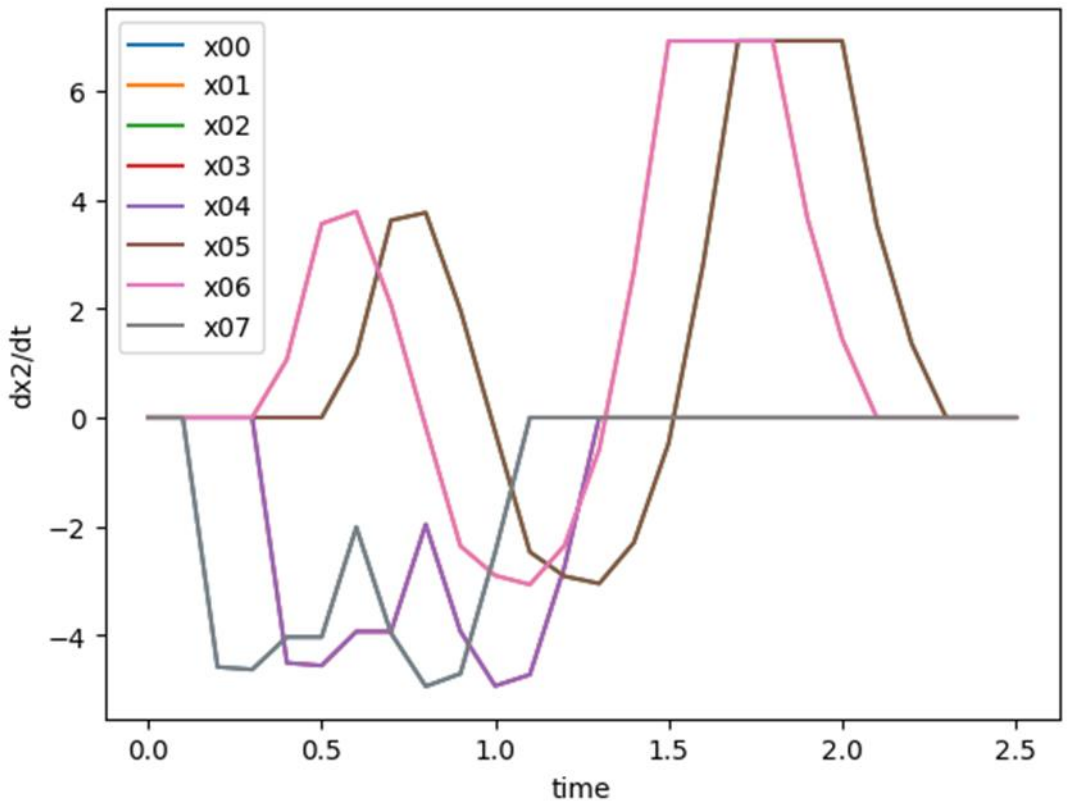

Figure 3.14. The robot velocities in direction x_2 from all initial conditions by VSGP method

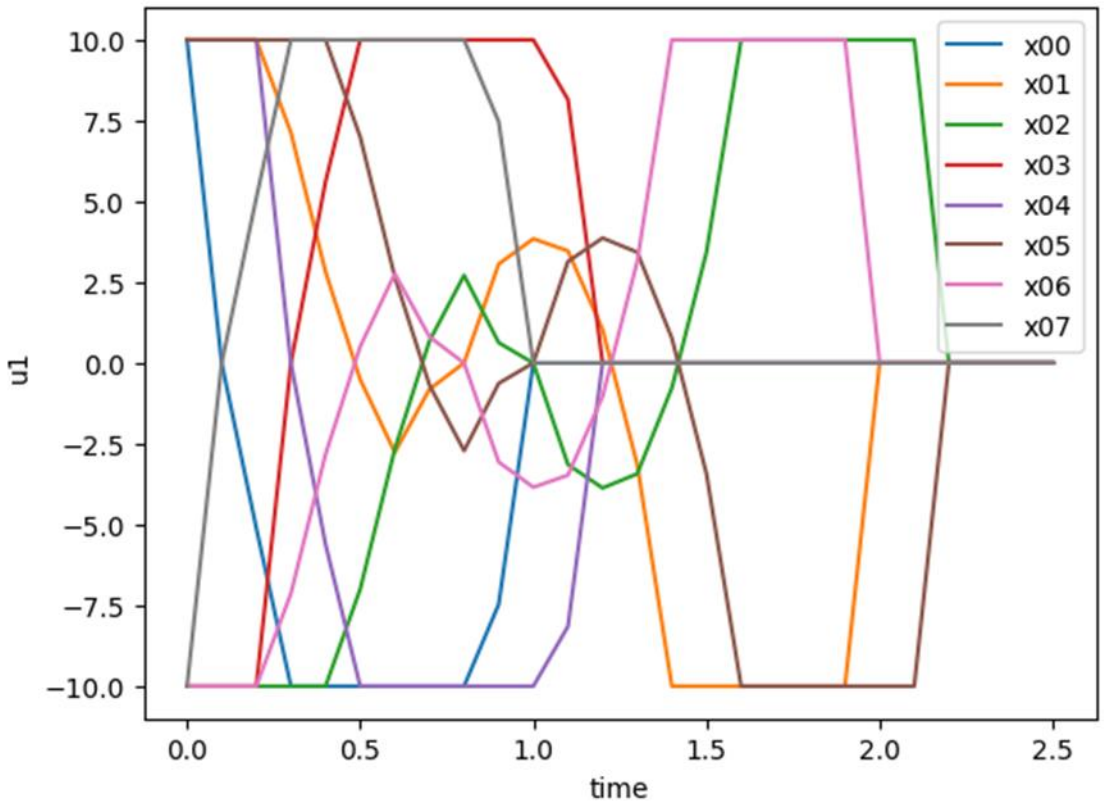

Figure 3.15. The robot control u_1 from all initial conditions by VSGP method

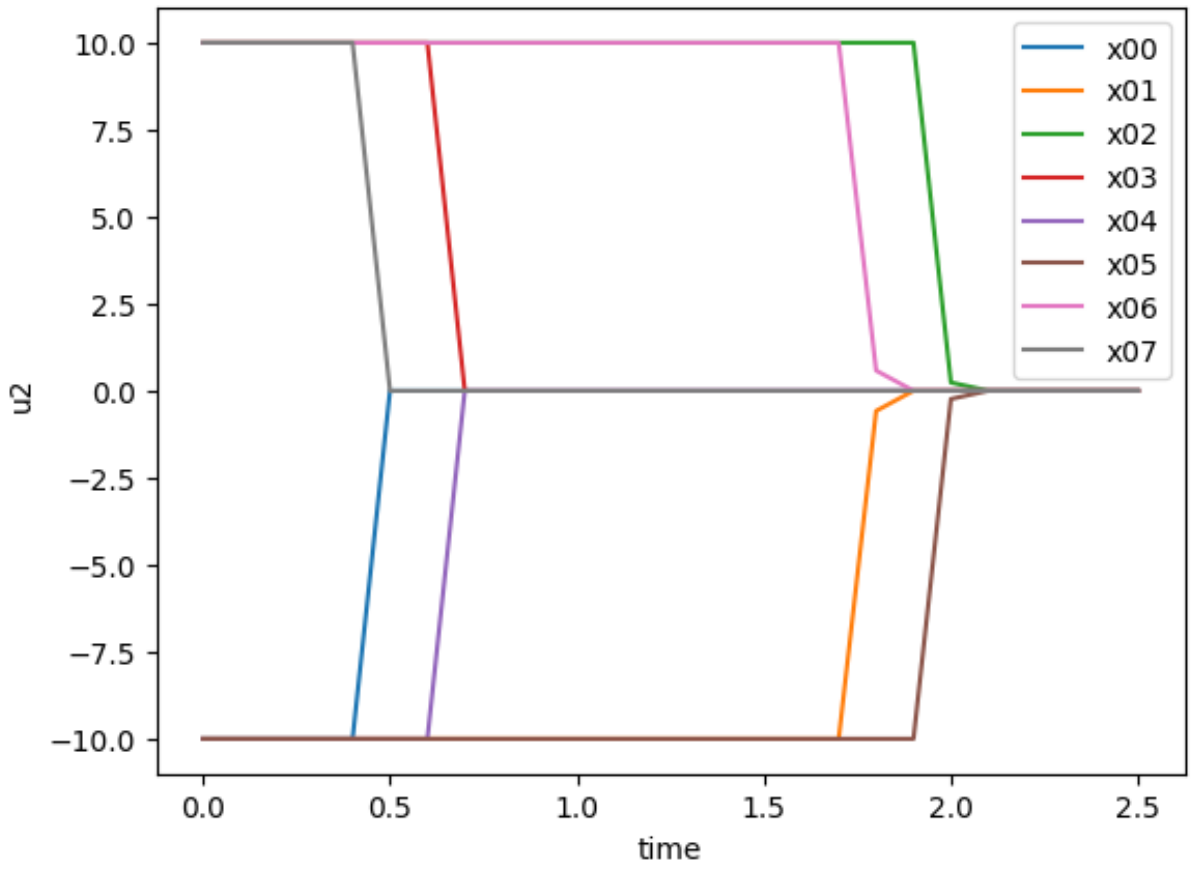

Figure 3.16. The robot control u_2 from all initial conditions by VSGP method

Case Three: Twelve initial states are provided:

$$X_0 = \left\{\boldsymbol{x}^{0,0} = \left[-3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,3} = \left[3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3\ \ 3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,5} = \left[-3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3\ \ -3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,7} = \left[3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,8} = [-3\ \ 3.5\ \ 0]^T, \boldsymbol{x}^{0,9} = [-3\ \ -3.5\ \ 0]^T, \boldsymbol{x}^{0,10} = [3\ \ -3.5\ \ 0]^T,\ \boldsymbol{x}^{0,11} = [3\ \ 3.5\ \ 0]^T\right\}. \tag{3.21}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^*\ \ x_2^*\ \ x_3^*]^T = [0\ \ 0\ \ 0]^T. \tag{3.22}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.23}$$

where

$$\tilde{u}_1 = 2 * \left(q_1\left(x_1^f - x_1\right) + \left(x_3^f - x_3\right)\left(x_2^f - x_2\right) + \left(q_3\left(x_3^f - x_3\right)\right)^3 \right), \qquad (3.24)$$

$$\tilde{u}_2 = q_2\left(x_1^f - x_1\right) * \ln\left(\left|2q_4\left(x_2^f - x_2\right)\right|\right), \qquad (3.25)$$

$q_1 = 5.94890$, $q_2 = 8.05063$, $q_3 = 0.86430$ and $q_4 = 1.64740$. The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 1.88384$, where $\varepsilon = 0.01$, $t^+ = 2.5$ sec, $\mathbf{L} = 12$, and , $p_1 = 1$.

Figure 3.17 shows the trajectories taken by one robot as it moved from twelve initial states (3.21) to the terminal state (3.22).

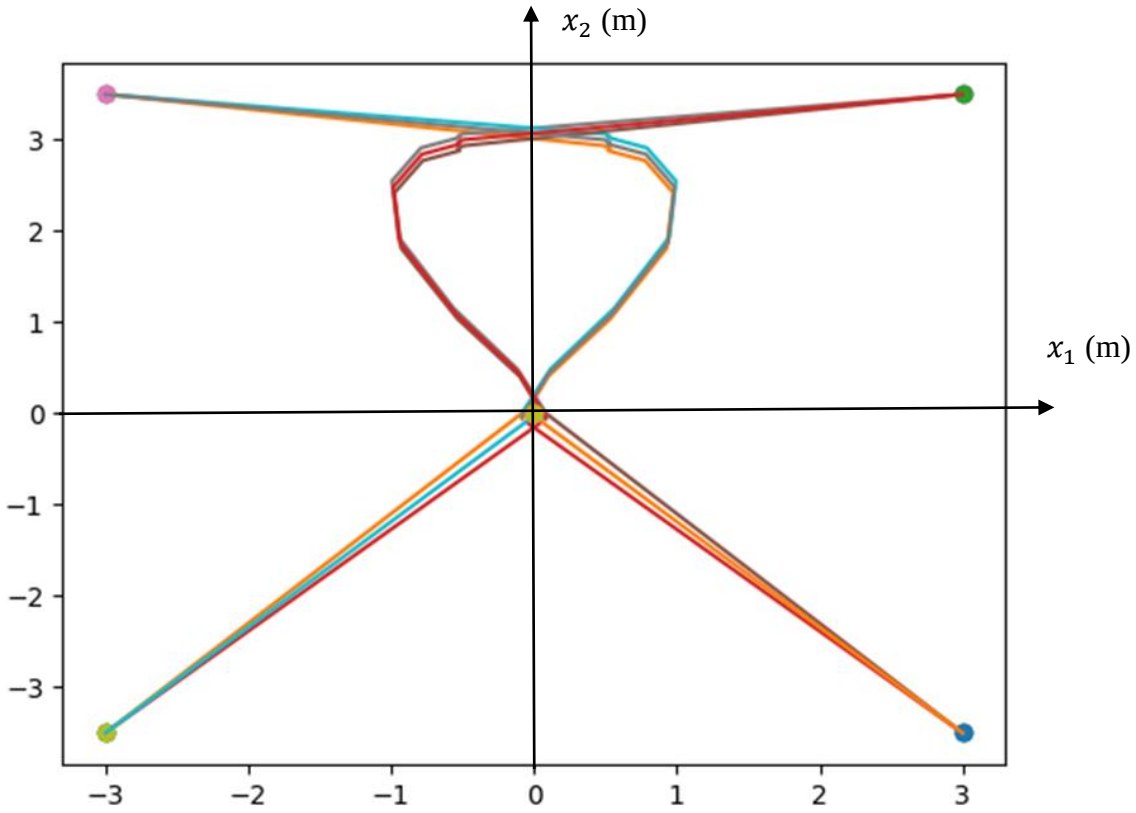

Figure 3.17. Robot trajectories with VSGP control function

Figures 3.18 through 3.29 depict the simulation results of a nonholonomic mobile robot, showcasing its behavior for displacement — meter, velocity — m/s, and control — m/s, in the y-axes and time — second in the x-axes, by the VSGP method.

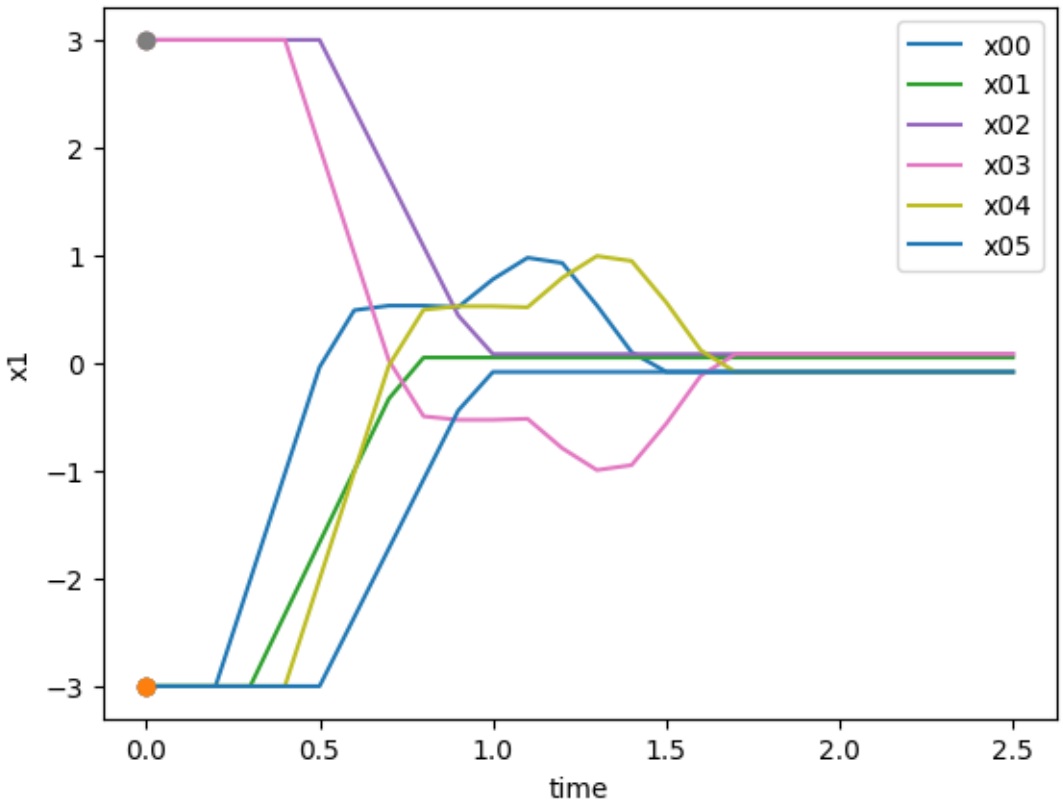

Figure 3.18. The robot displacements in direction x_1 from first six initial conditions by VSGP

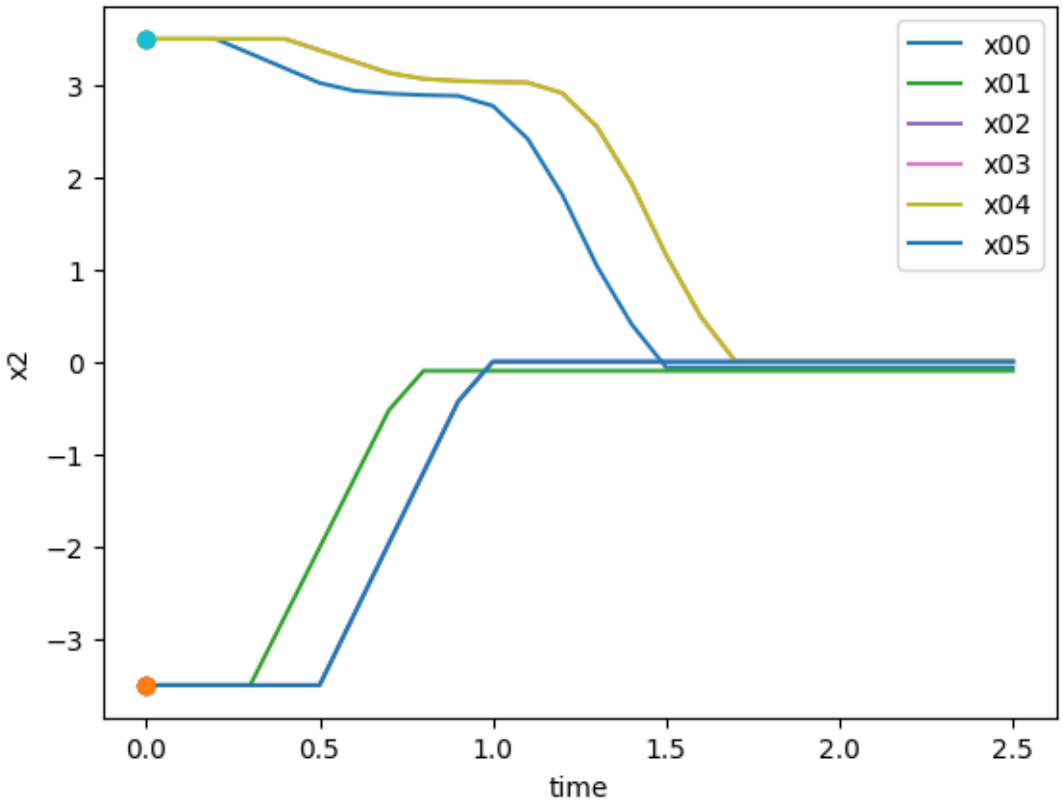

Figure 3.19. The robot displacements in direction x_2 from first six initial conditions by VSGP

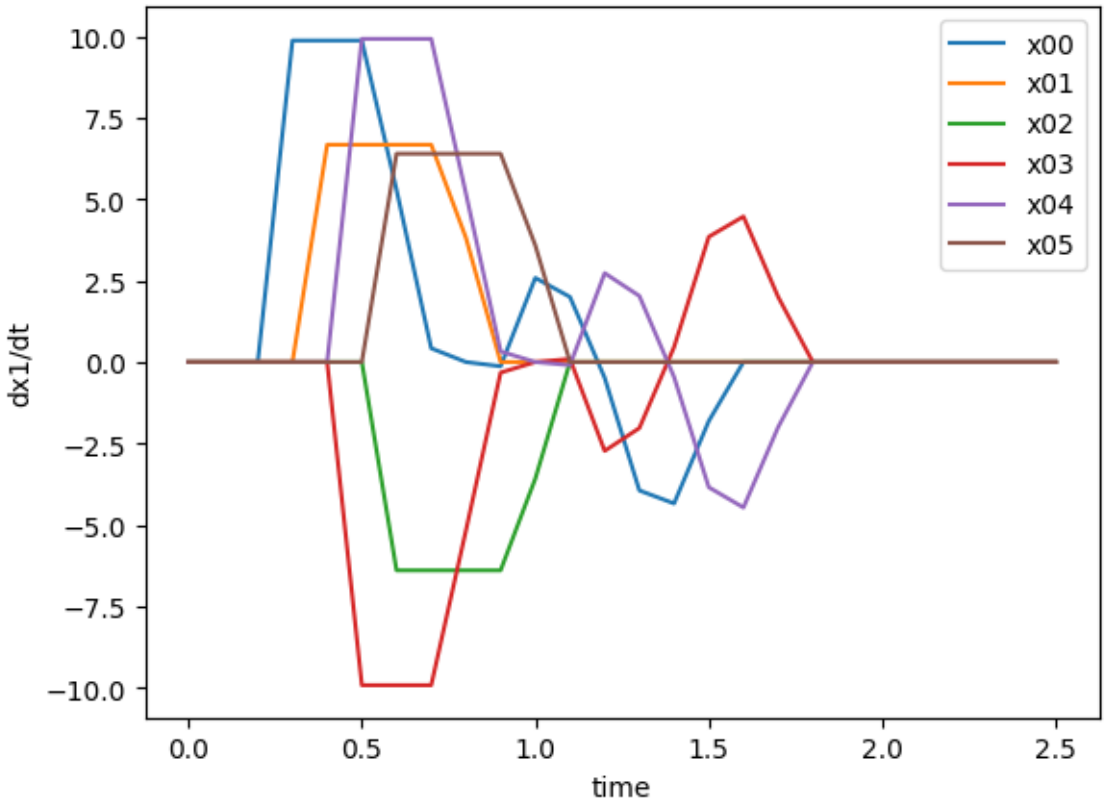

Figure 3.20. The robot velocities in direction x_1 from first six initial conditions by VSGP

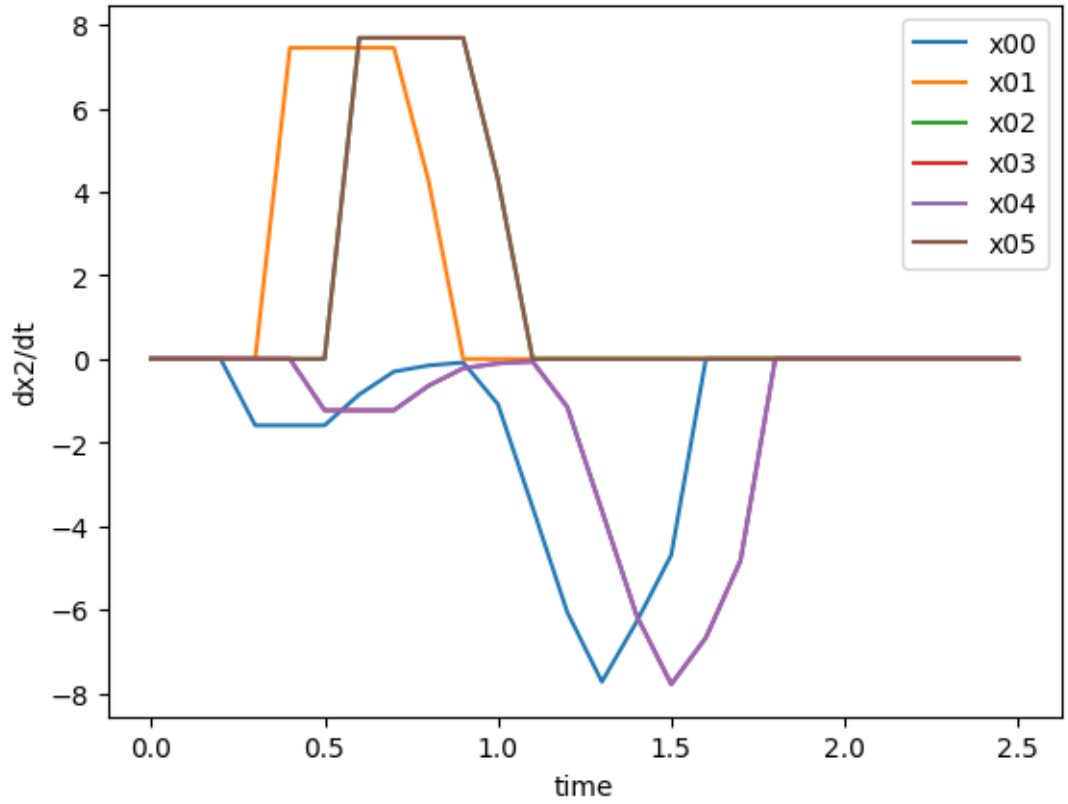

Figure 3.21. The robot velocities in direction x_2 from first six initial conditions by VSGP

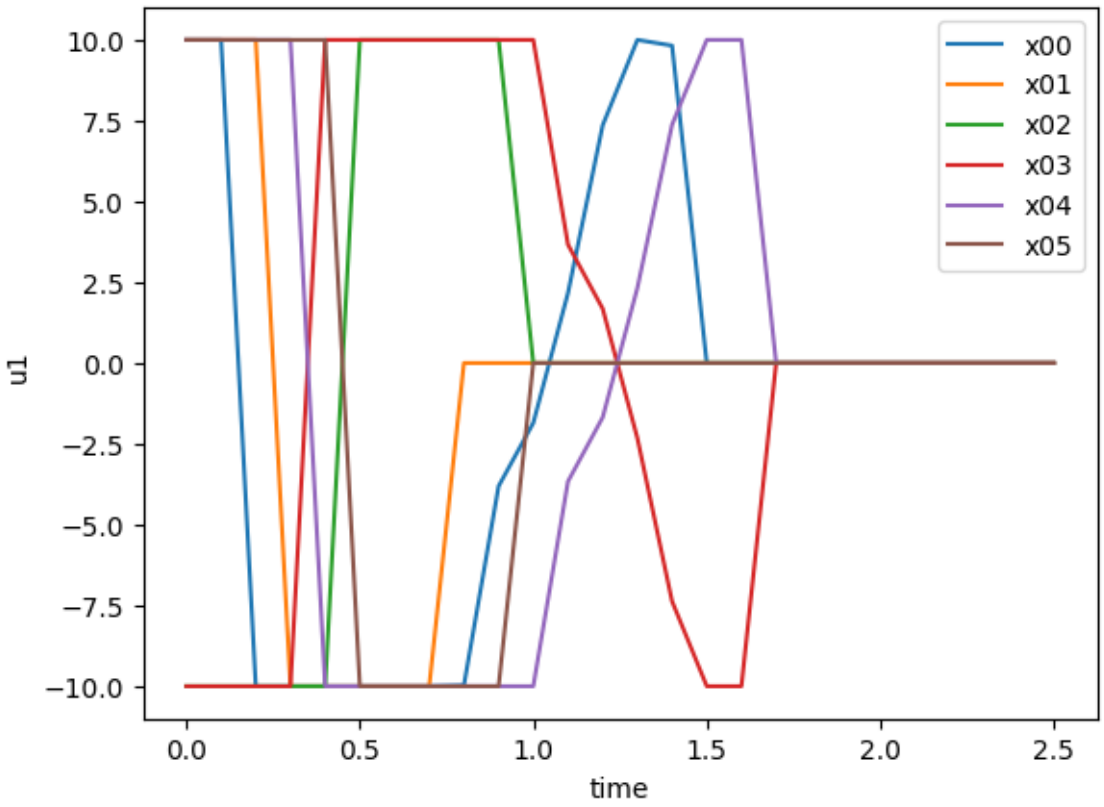

Figure 3.22. The robot control u_1 from first six initial conditions by VSGP

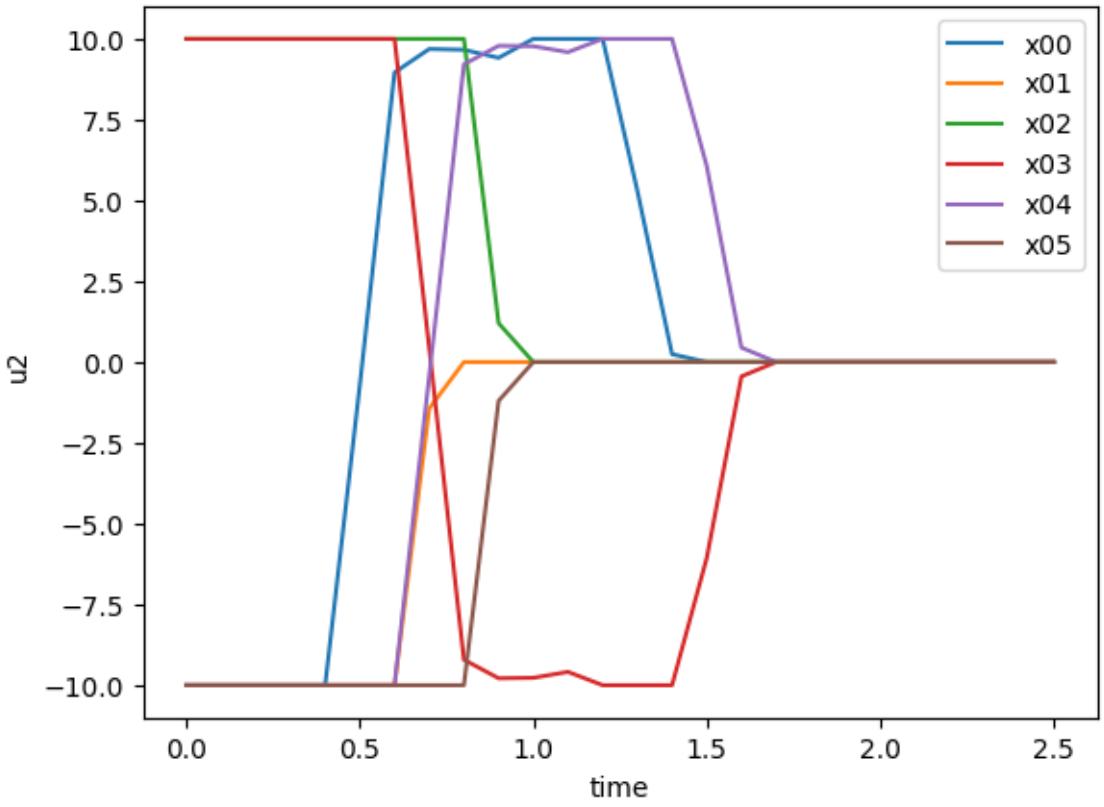

Figure 3.23. The robot control u_2 from first six initial conditions by VSGP

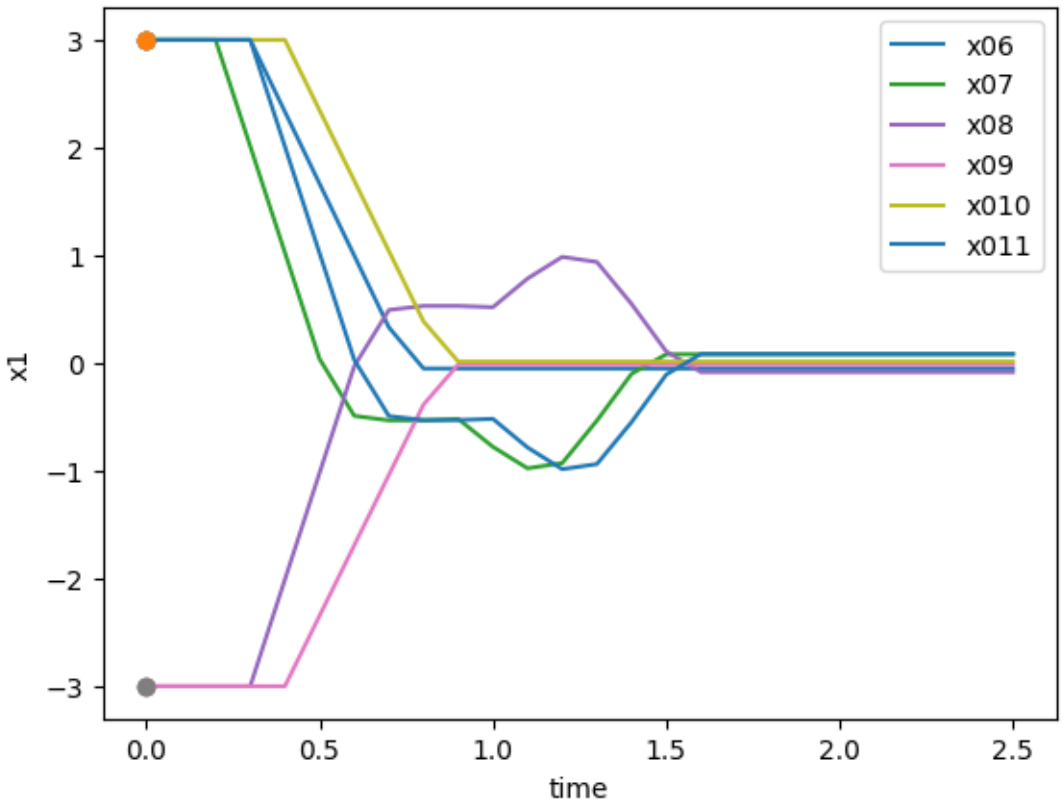

Figure 3.24. The robot displacements in direction x_1 from second six initial conditions by VSGP

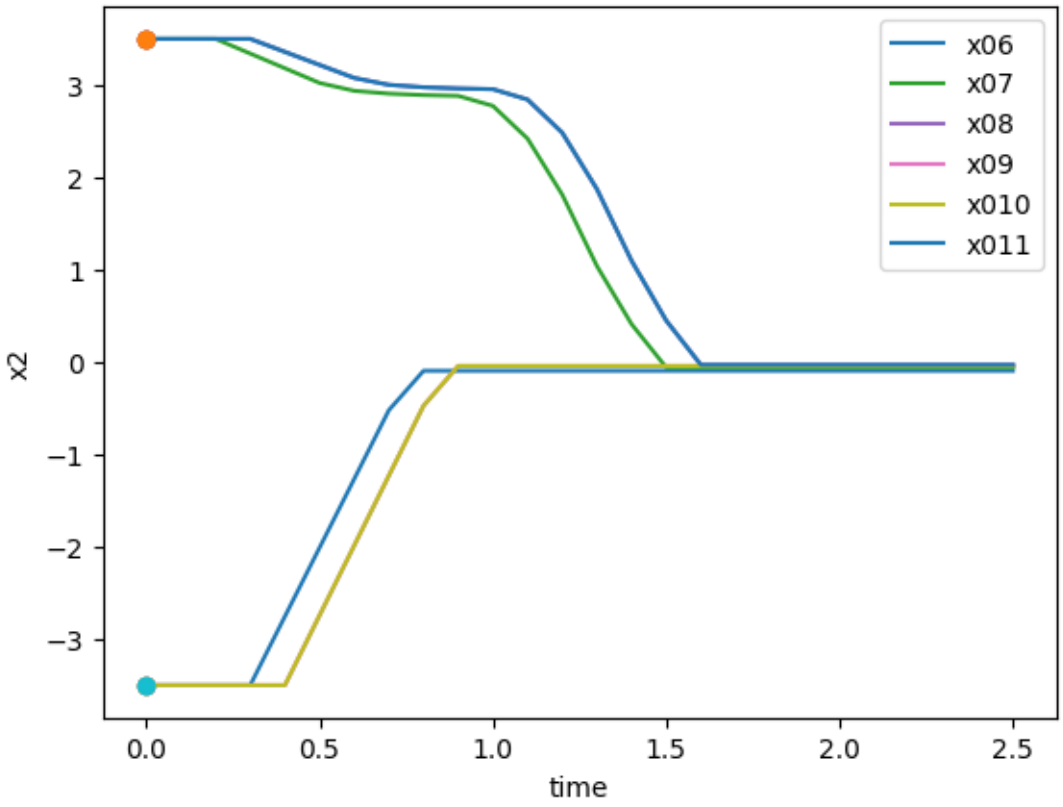

Figure 3.25. The robot displacements in direction x_2 from second six initial conditions by VSGP

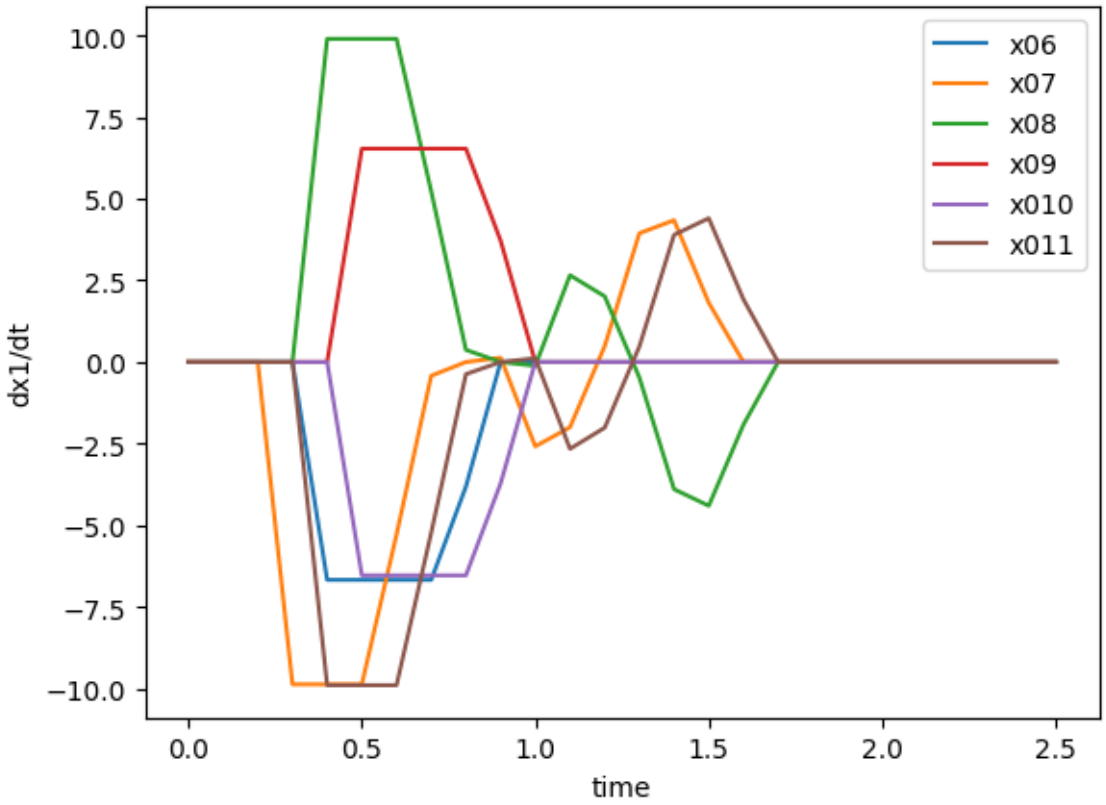

Figure 3.26. The robot velocities in direction x_1 from second six initial conditions by VSGP

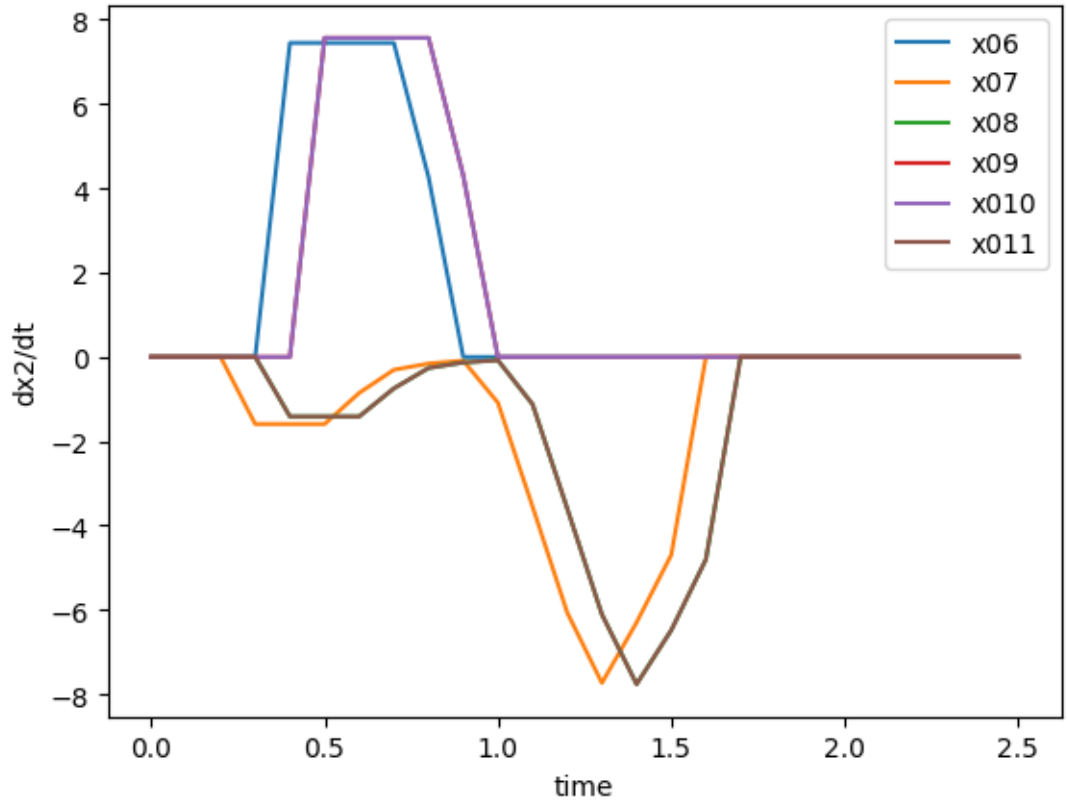

Figure 3.27. The robot velocities in direction x_2 from second six initial conditions by VSGP

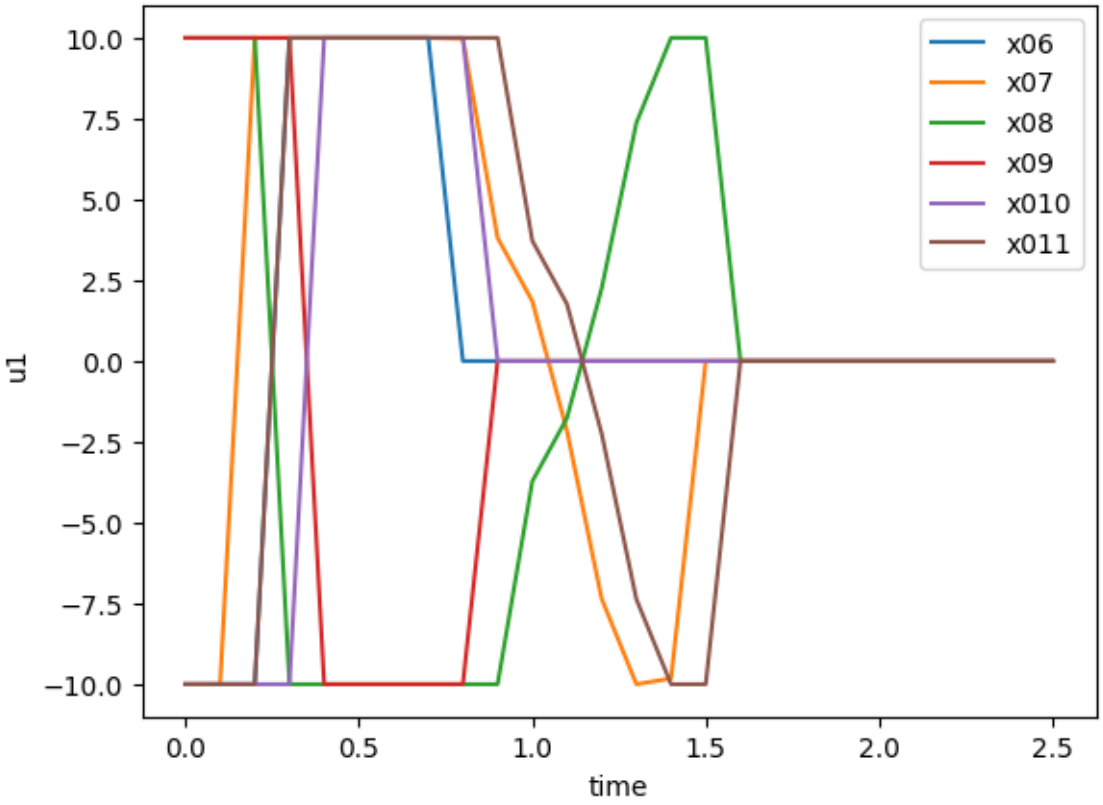

Figure 3.28. The robot control u_1 from second six initial conditions by VSGP

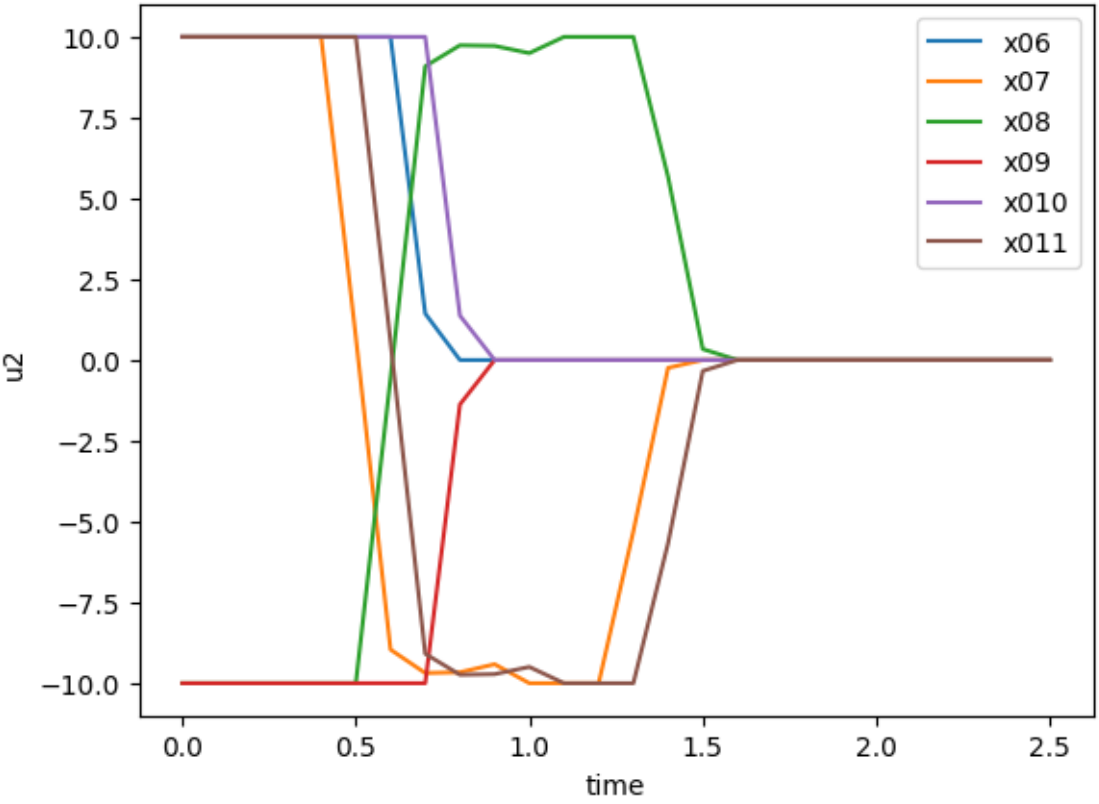

Figure 3.29. The robot control u_2 from second six initial conditions by VSGP

Case Four: Twelve initial states are provided:

$$X_0 = \left\{\boldsymbol{x}^{0,0} = \left[-3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,3} = \left[3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,5} = \left[-3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,7} = \left[3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,8} = [-3\ \ 3.5\ \ 0]^T, \boldsymbol{x}^{0,9} = [-3\ \ -3.5\ \ 0]^T, \boldsymbol{x}^{0,10} = [3\ \ -3.5\ \ 0]^T,\ \boldsymbol{x}^{0,11} = [3\ \ 3.5\ \ 0]^T\right\}. \tag{3.26}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^*\ \ x_2^*\ \ x_3^*]^T = [0\ \ 0\ \ 0]^T. \tag{3.27}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.28}$$

where

$$\tilde{u}_1 = B * (1 - \exp(-B^2)) + \left(x_3^f - x_3\right), \tag{3.29}$$

$$\tilde{u}_2 = \left(\left(\left(x_2^f - x_2\right)\left(x_1^f - x_1\right)\right)^3 + \rho\left(\left(x_2^f - x_2\right)\left(x_1^f - x_1\right)\right)\right)^3, \tag{3.30}$$

$$B = sgn(q_1) * \ln(|q_1| + 1)\left(x_1^f - x_1\right) + sgn(q_1) * \ln(|q_1| + 1)\left(x_1^f - x_1\right)\left(x_2^f - x_2\right) + q_2\left(x_3^f - x_3\right), \tag{3.31}$$

$$\rho(\mu) = \begin{cases} 0, & \text{if } |\mu| < \delta \\ sgn(\mu), & \text{otherwise} \end{cases} \tag{3.32}$$

$q_1 = 7.60454$, $q_2 = 5.86686$ and $\delta = 10^{-8}$. The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 2.37900$, where $\varepsilon = 0.01, t^+ = 2.5$ sec, $\mathbf{L} = 12$, and , $p_1 = 1$.

Figure 3.30 shows the trajectories taken by one robot as it moved from twelve initial states (3.26) to the terminal state (3.27).

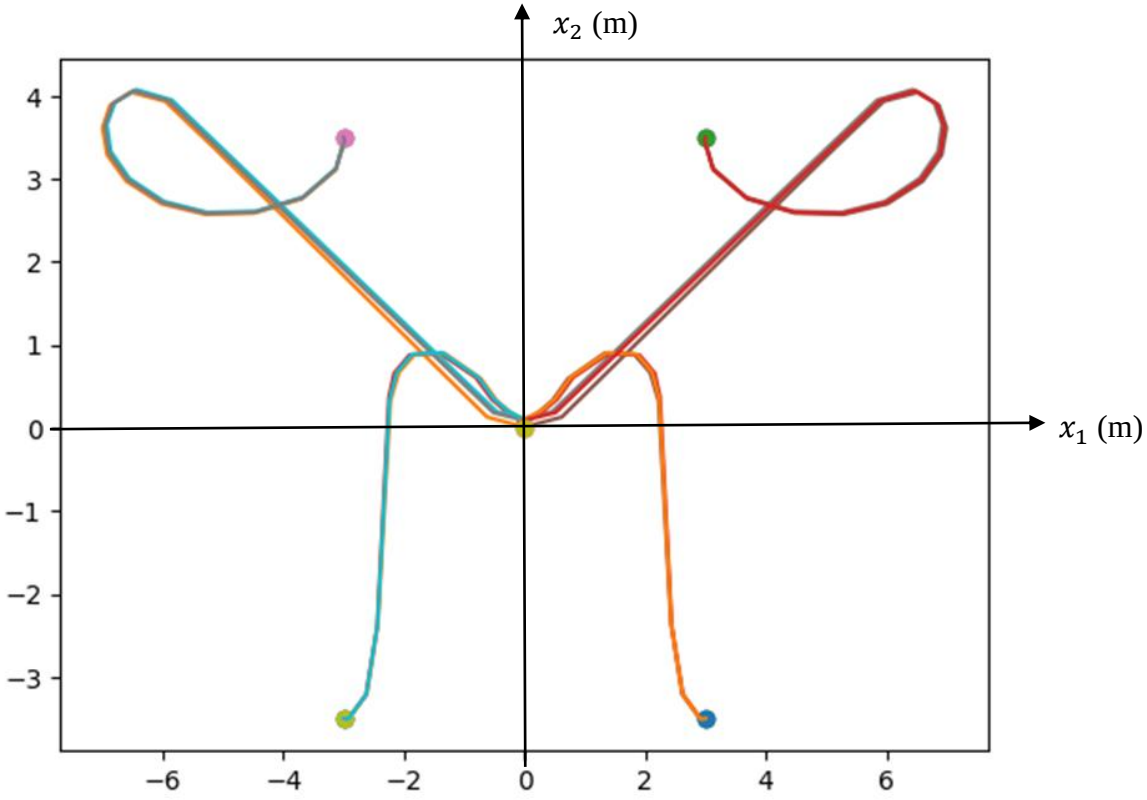

Figure 3.30. Robot trajectories with VSGP control function

Figures 3.31 through 3.42 depict the simulation results of a nonholonomic mobile robot, showcasing its behavior for displacement — meter, velocity — m/s, and control — m/s, in the y-axes and time — second in the x-axes, by the VSGP method.

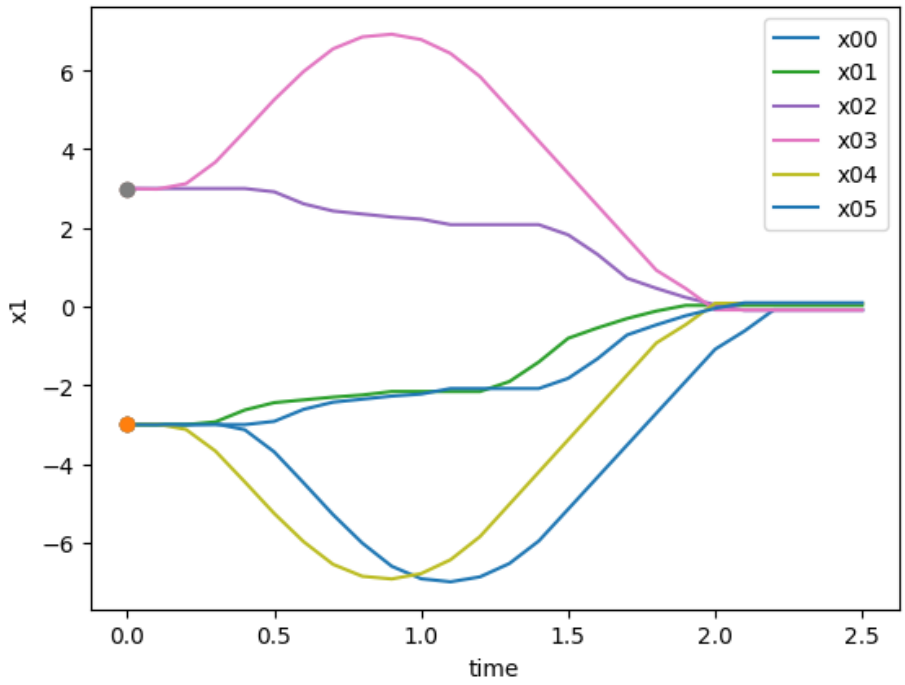

Figure 3.31. The robot displacements in direction x_1 from first six initial conditions by VSGP

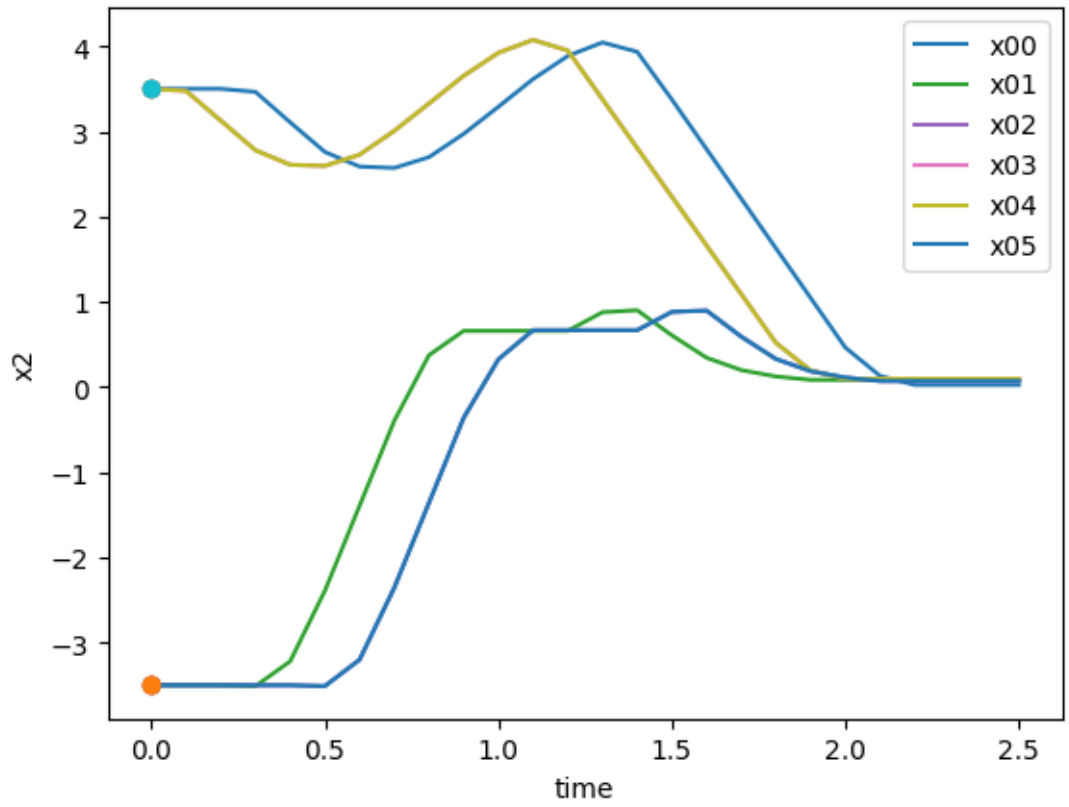

Figure 3.32. The robot displacements in direction x_2 from first six initial conditions by VSGP

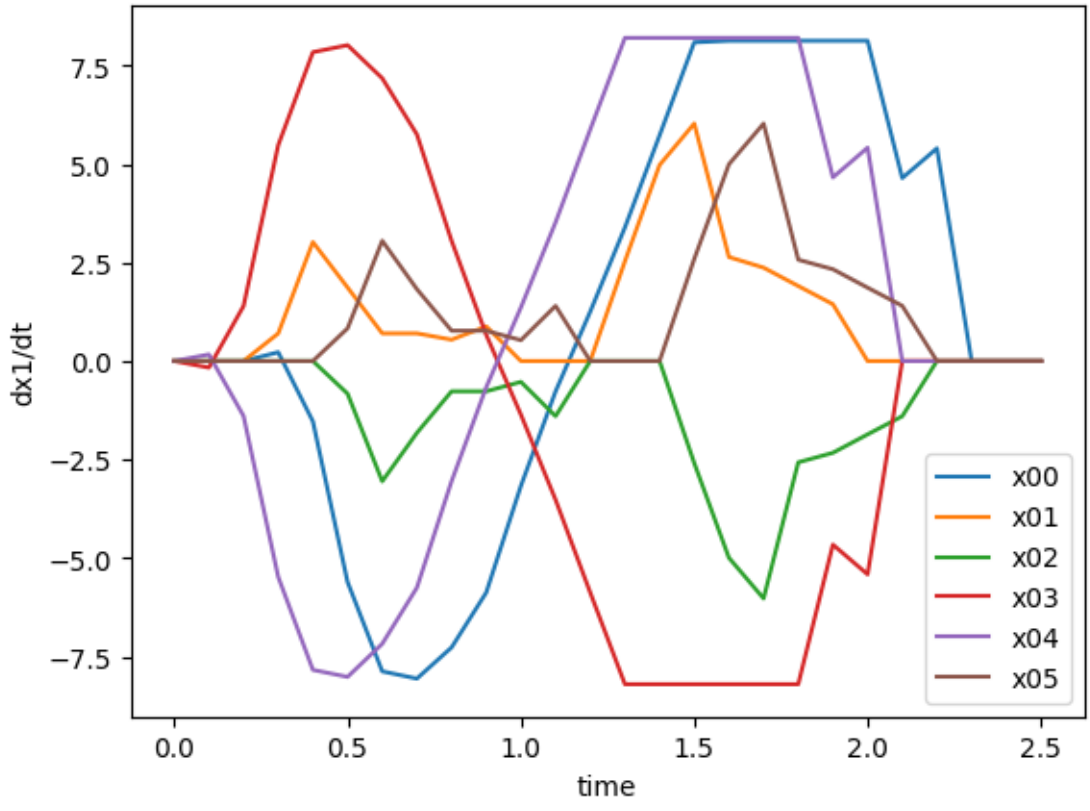

Figure 3.33. The robot velocities in direction x_1 from first six initial conditions by VSGP

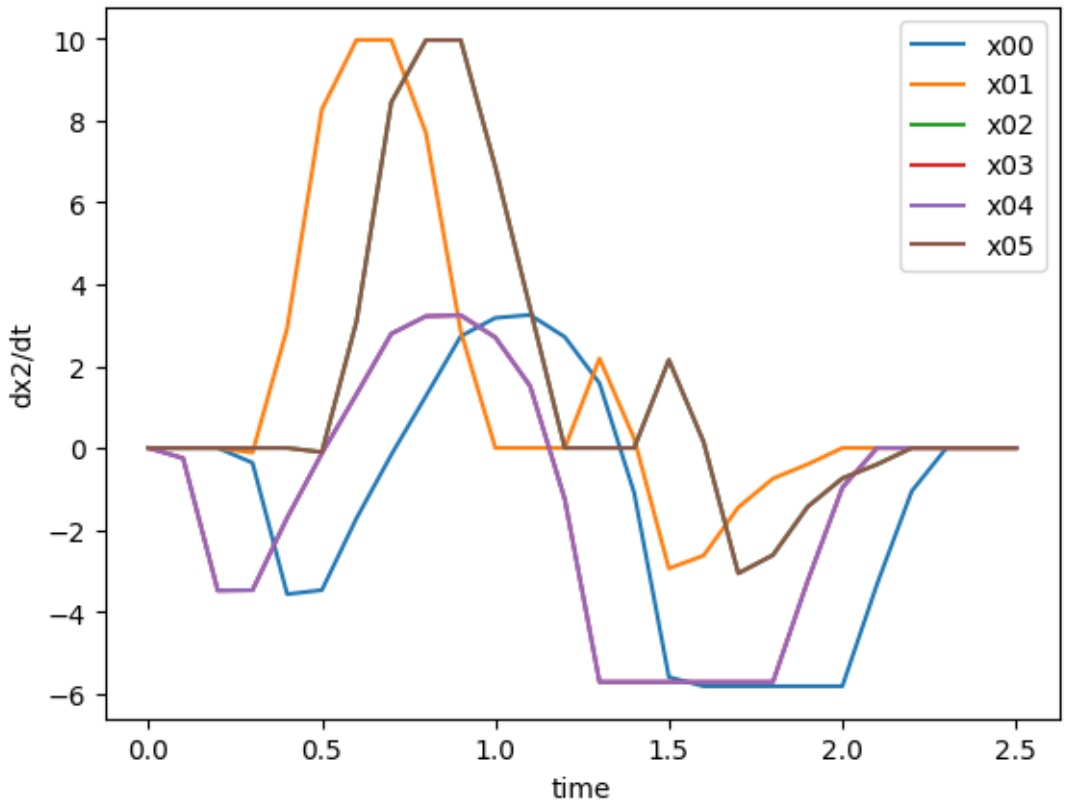

Figure 3.34. The robot velocities in direction x_2 from first six initial conditions by VSGP

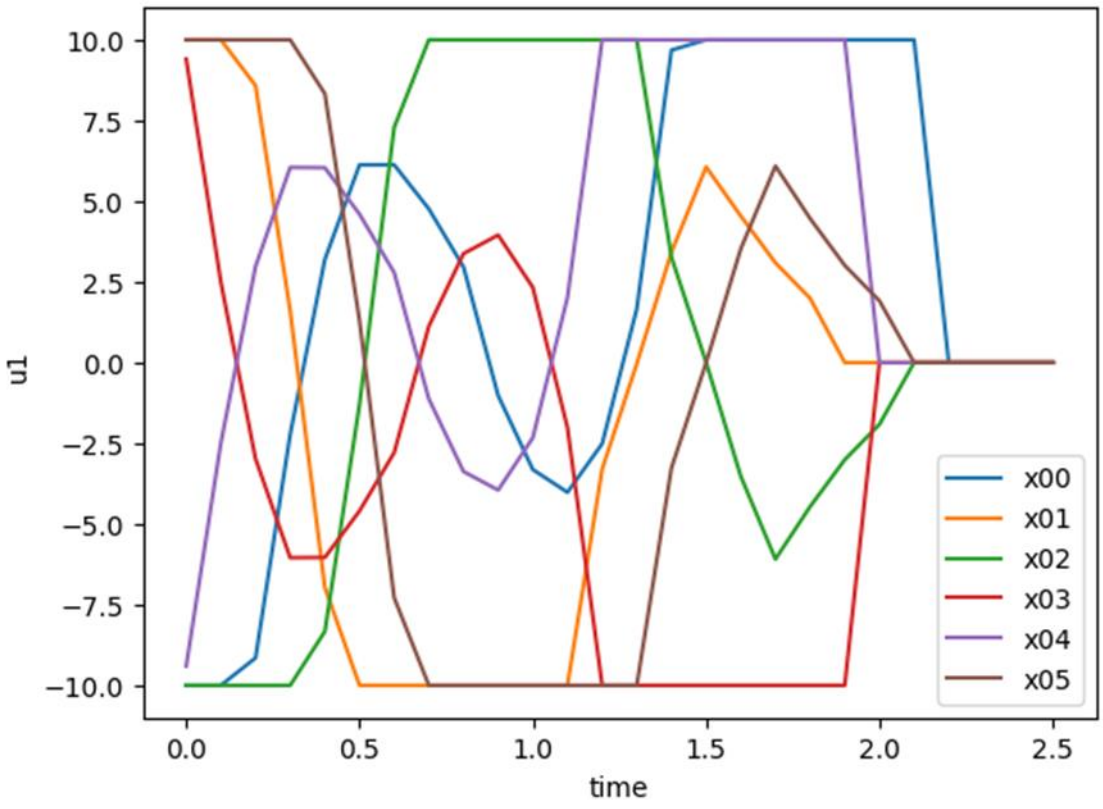

Figure 3.35. The robot control u_1 from first six initial conditions by VSGP

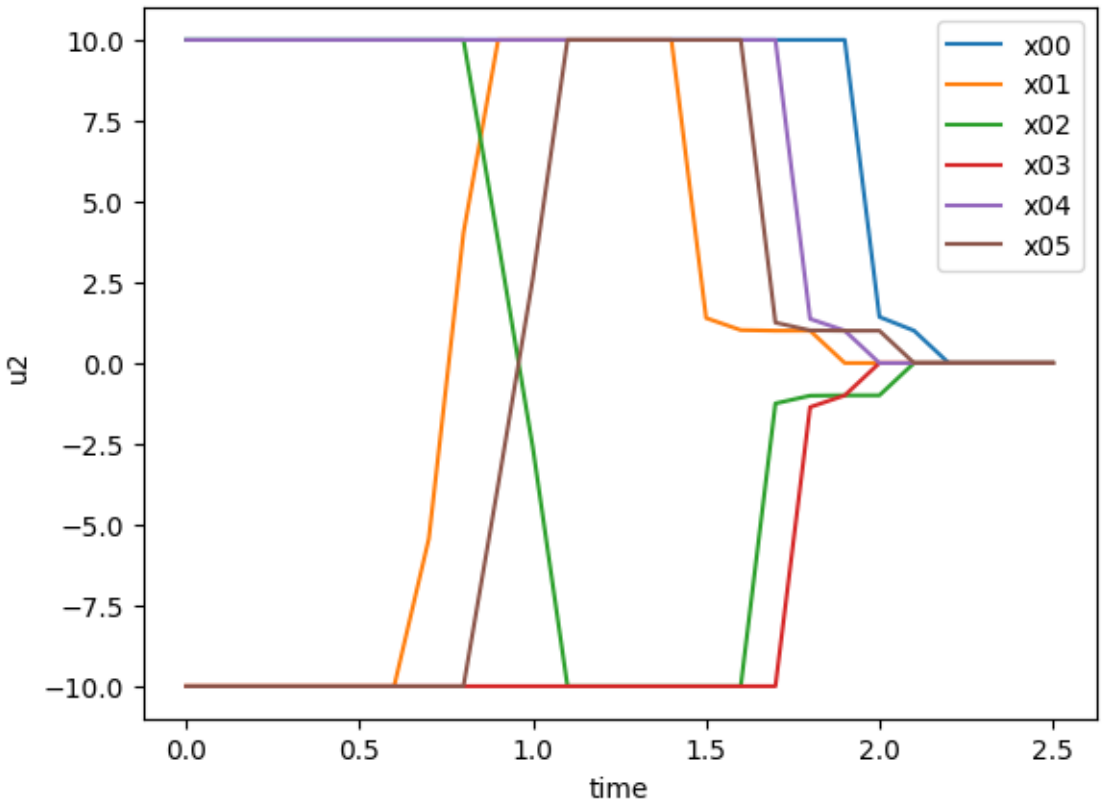

Figure 3.36. The robot control u_2 from first six initial conditions by VSGP

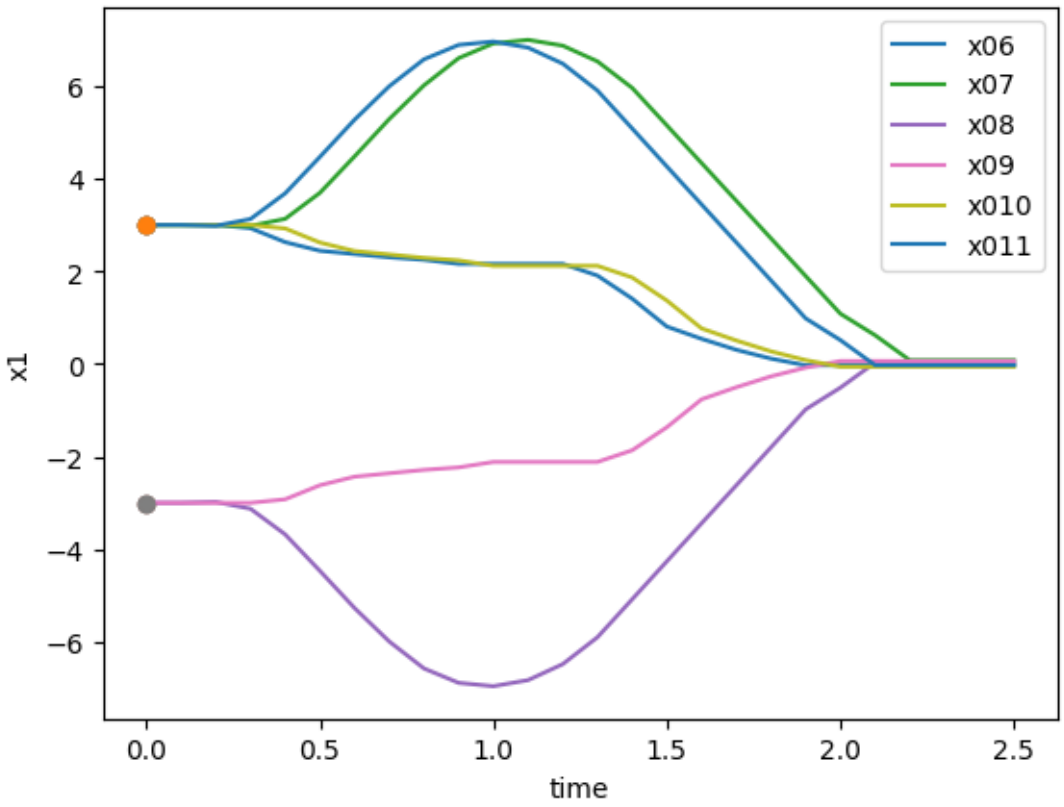

Figure 3.37. The robot displacements in direction x_1 from second six initial conditions by VSGP

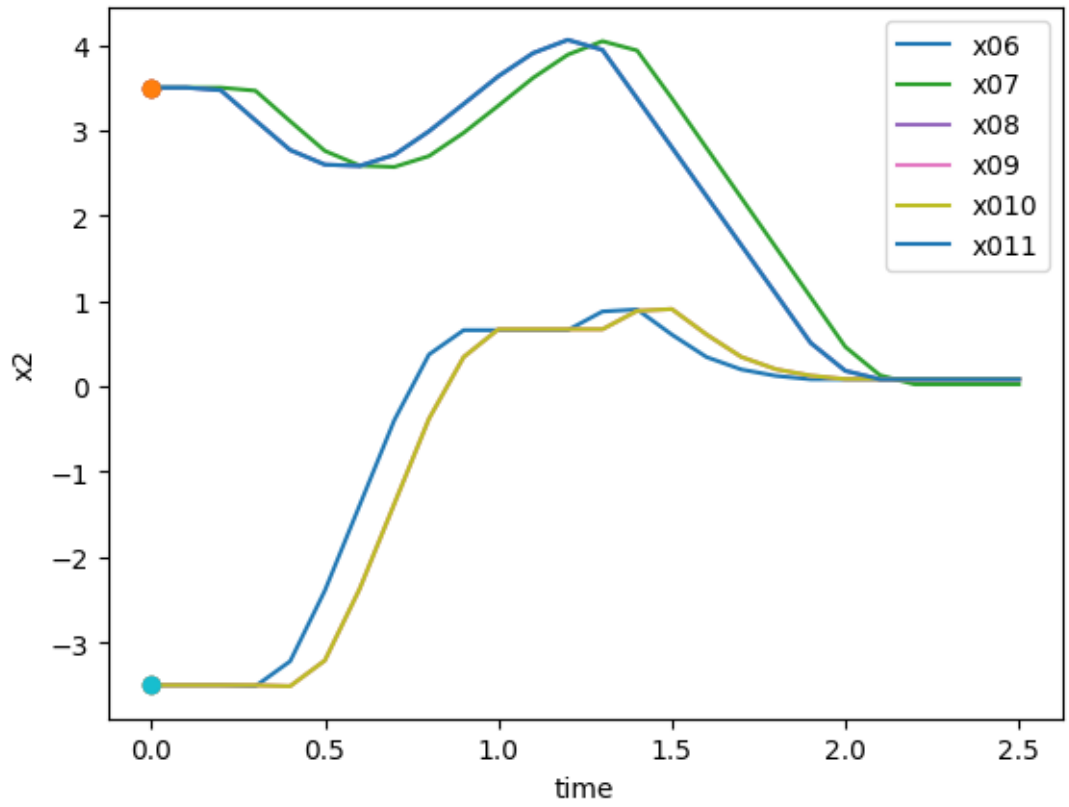

Figure 3.38. The robot displacements in direction x_2 from second six initial conditions by VSGP

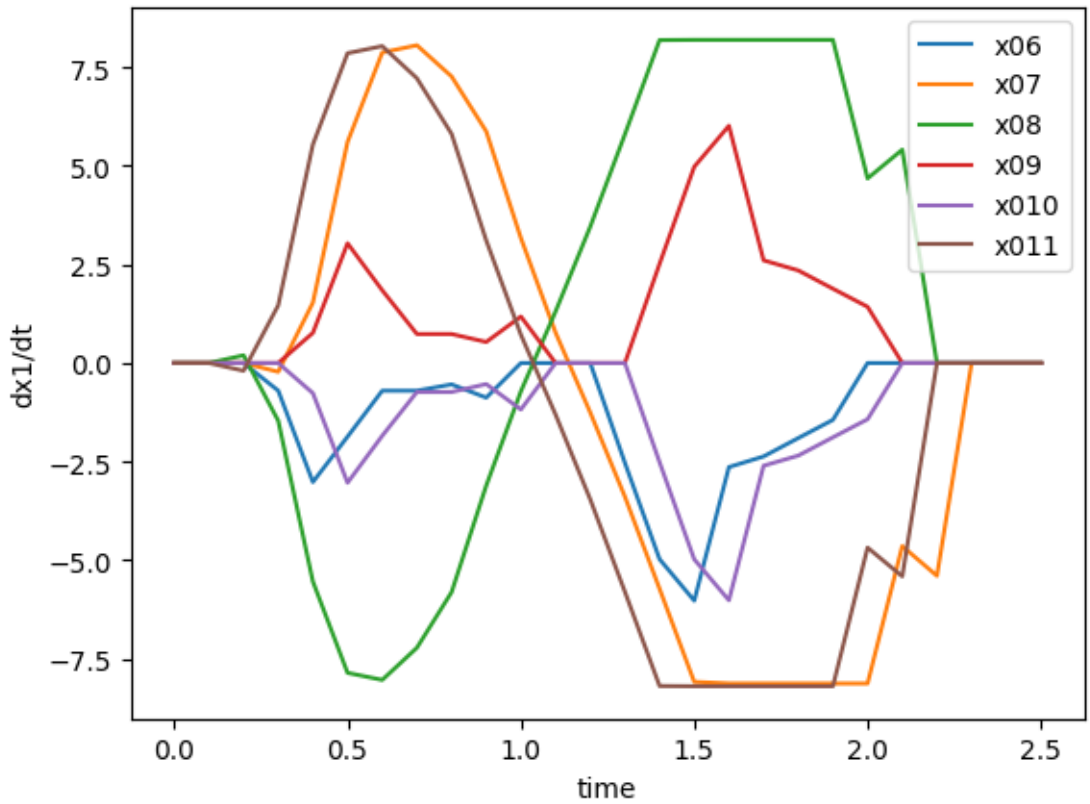

Figure 3.39. The robot velocities in direction x_1 from second six initial conditions by VSGP

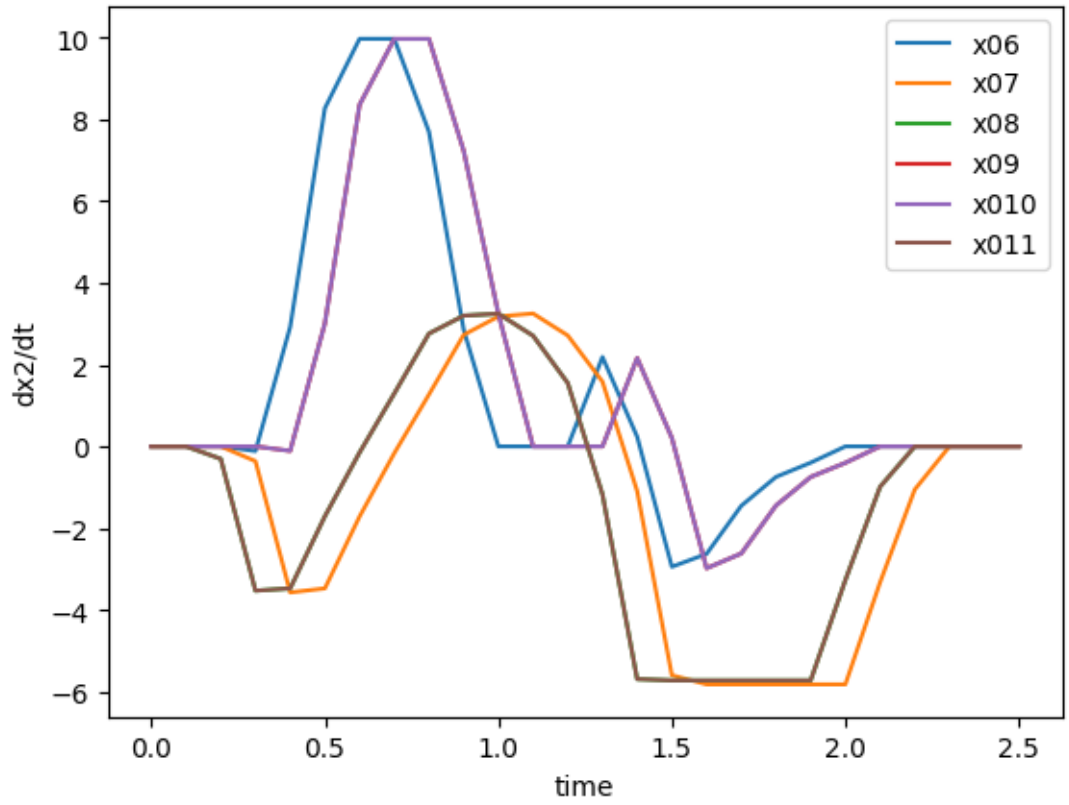

Figure 3.40. The robot velocities in direction x_2 from second six initial conditions by VSGP

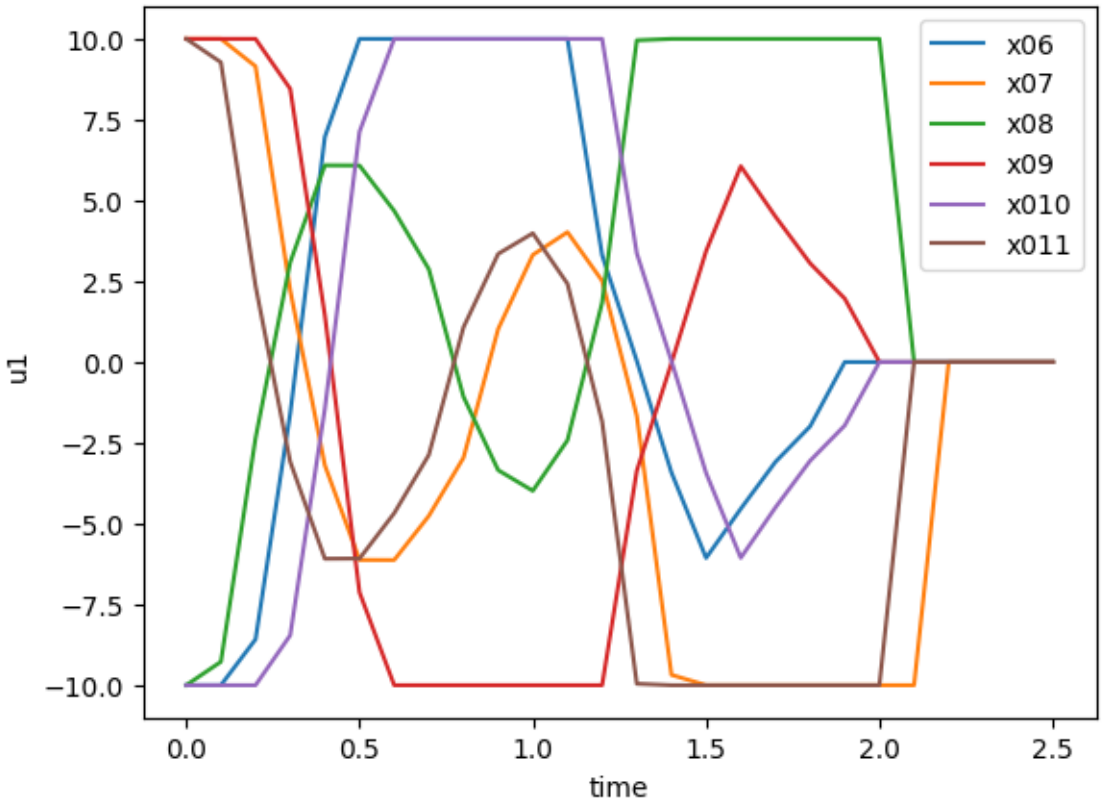

Figure 3.41. The robot control u_1 from second six initial conditions by VSGP

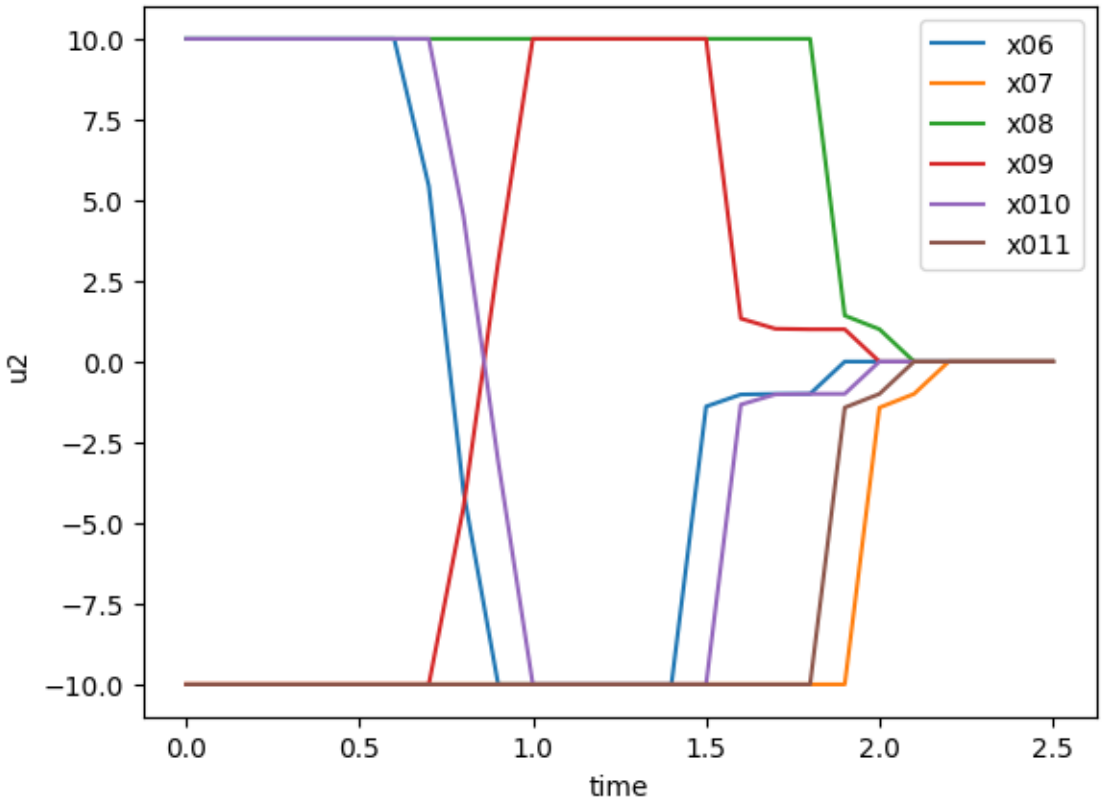

Figure 3.42. The robot control u_2 from second six initial conditions by VSGP

Case Five: Twelve initial states are provided:

$$X_0 = \left\{\boldsymbol{x}^{0,0} = \left[-3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,3} = \left[3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,5} = \left[-3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,7} = \left[3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,8} = [-3\ \ 3.5\ \ 0]^T, \boldsymbol{x}^{0,9} = [-3\ \ -3.5\ \ 0]^T, \boldsymbol{x}^{0,10} = [3\ \ -3.5\ \ 0]^T,\ \boldsymbol{x}^{0,11} = [3\ \ 3.5\ \ 0]^T\right\}. \tag{3.33}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^*\ \ x_2^*\ \ x_3^*]^T = [0\ \ 0\ \ 0]^T. \tag{3.34}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.35}$$

where

$$\tilde{u}_1 = C * (1 - \exp(-C^2)) + arctan(q_2(x_3^f - x_3)), \tag{3.36}$$

$$\tilde{u}_2 = sgn\left(q_1(x_1^f - x_1)\right) * \left(\exp\left(\left|q_1(x_1^f - x_1)\right|\right) - 1\right) + sgn\left(q_3\left((x_3^f - x_3) * \left(1 - \exp\left(-\left((x_3^f - x_3)^2\right)\right)\right)\right)\right) * \left(\left|q_3((x_3^f - x_3) * \left(1 - \exp\left(-((x_3^f - x_3)^2)\right)\right))\right|\right)^{0.5}, \tag{3.37}$$

$$C = (x_2^f - x_2)(x_1^f - x_1) + (x_2^f - x_2)(x_1^f - x_1)(x_2^f - x_2) + q_2(x_3^f - x_3), \tag{3.38}$$

$q_1 = 7.62259$, $q_2 = 6.27694$ and $q_3 = 7.40398$. The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 2.38341$, where $\varepsilon = 0.01, t^+ = 2.5$ sec, $\mathbf{L} = 12$, and , $p_1 = 1$.

Figure 3.43 shows the trajectories taken by one robot as it moved from twelve initial states (3.33) to the terminal state (3.34).

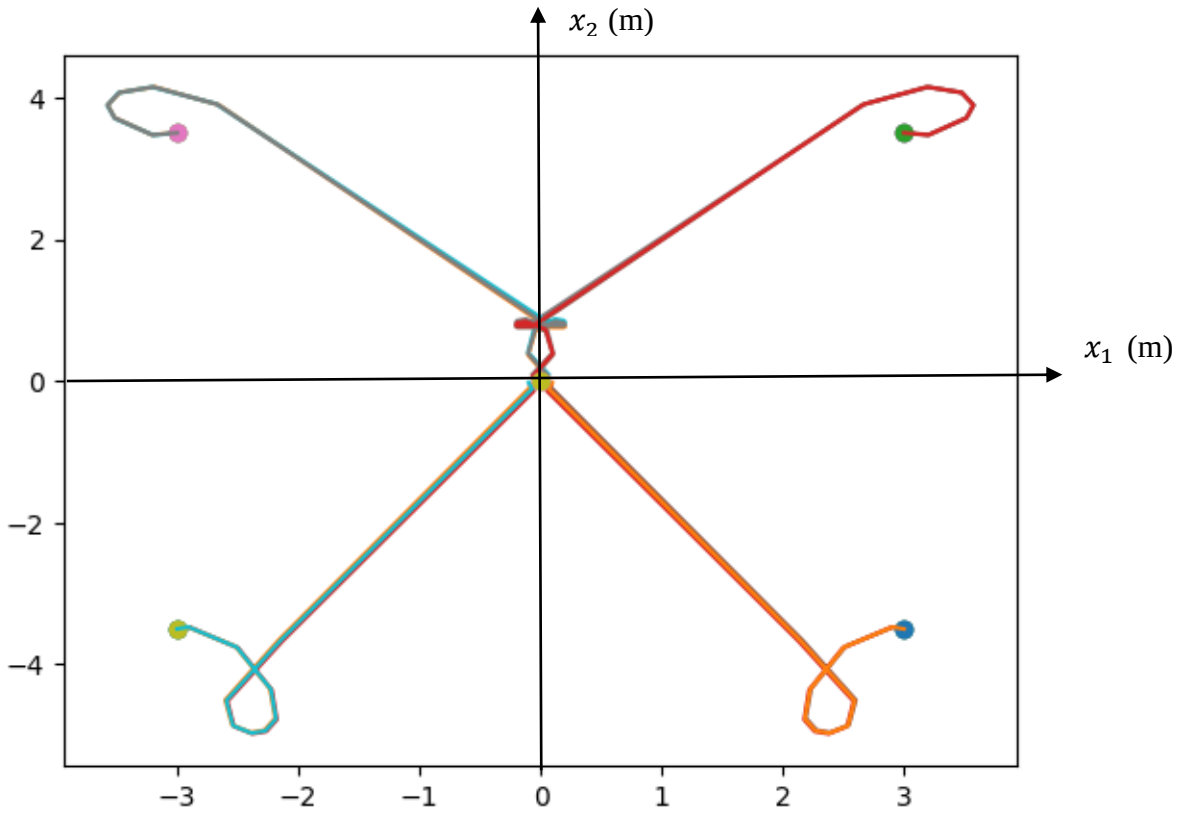

Figure 3.43. Robot trajectories with VSGP control function

Figures 3.44 through 3.55 depict the simulation results of a nonholonomic mobile robot, showcasing its behavior for displacement — meter, velocity — m/s, and control — m/s, in the y-axes and time — second in the x-axes, by the VSGP method.

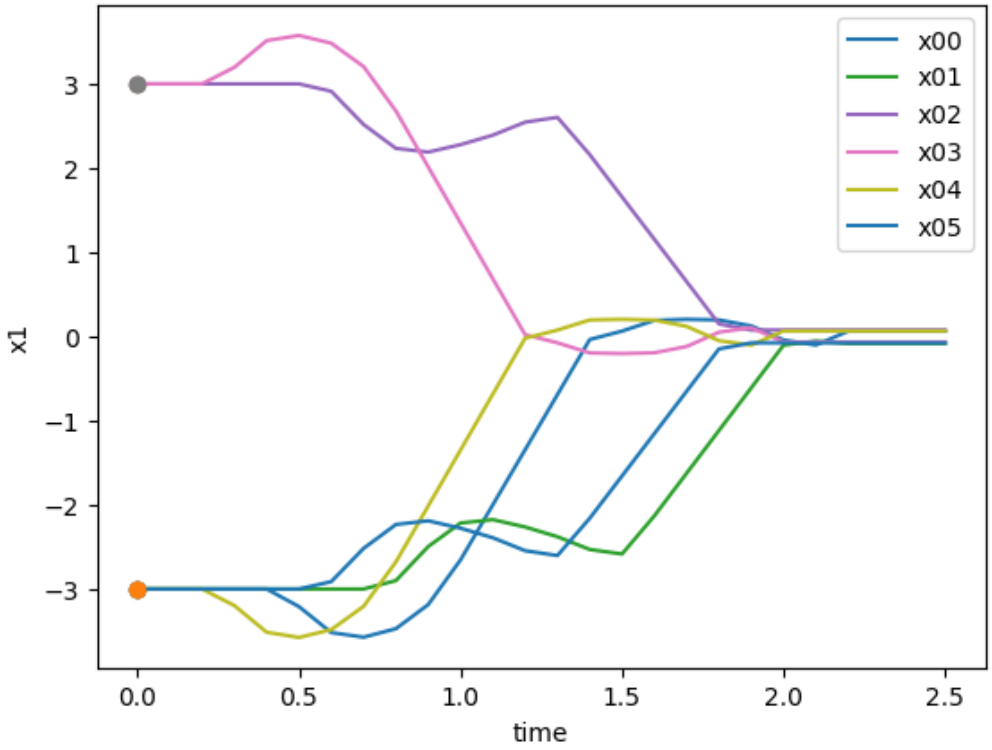

Figure 3.44. The robot displacements in direction x_1 from first six initial conditions by VSGP

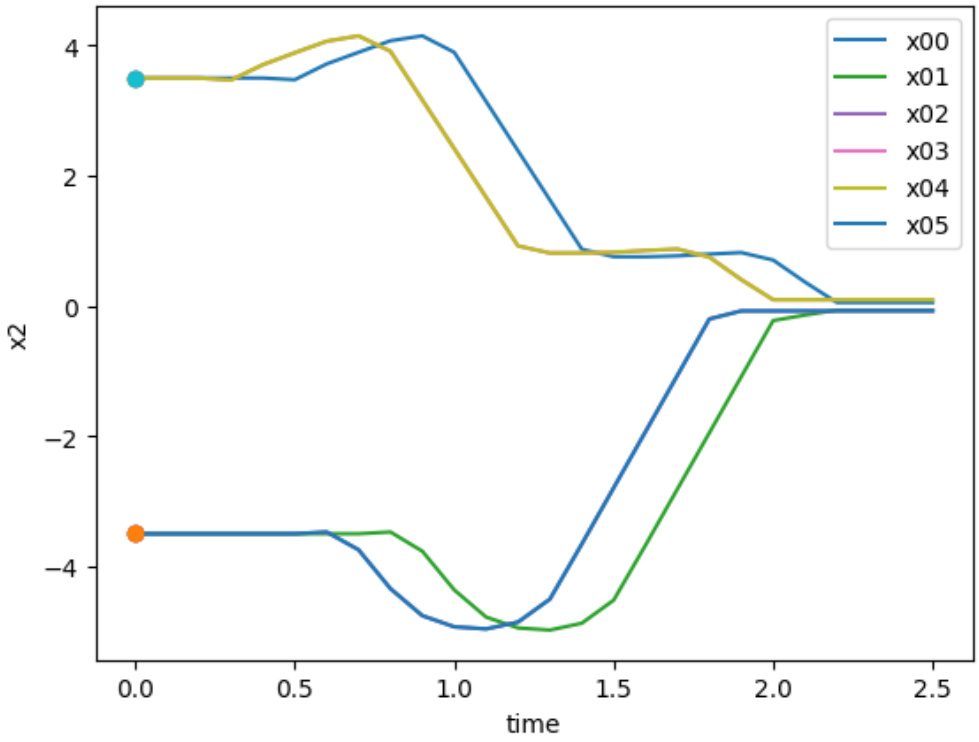

Figure 3.45. The robot displacements in direction x_2 from first six initial conditions by VSGP

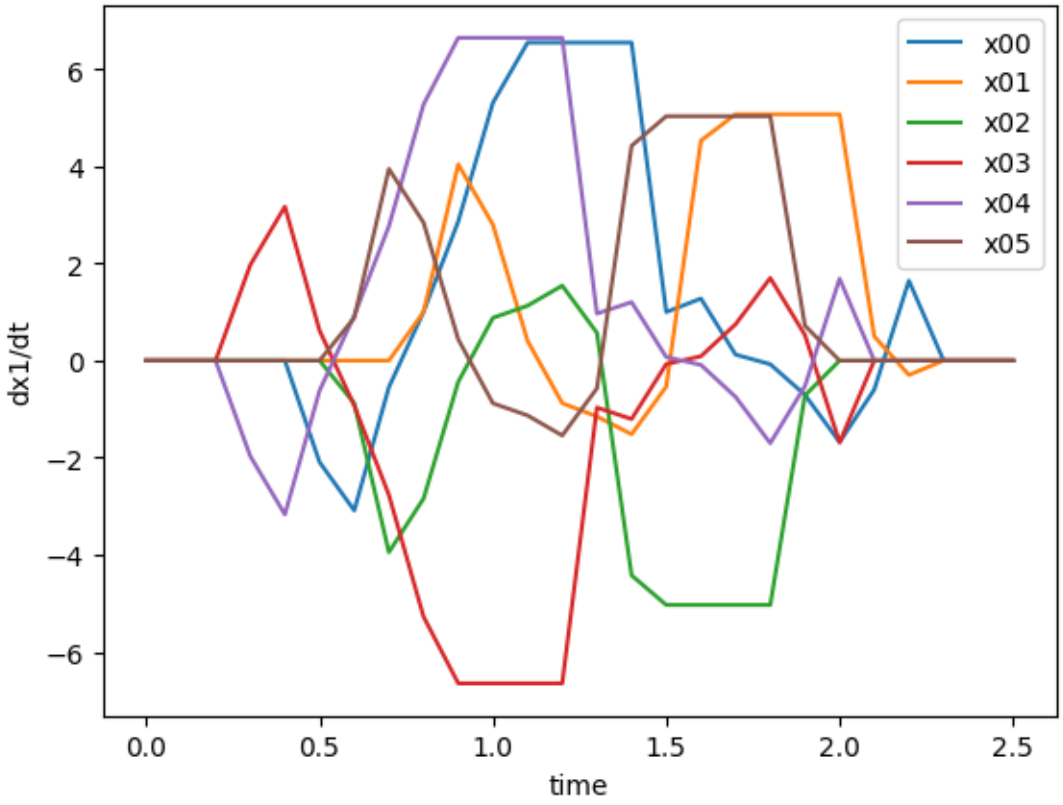

Figure 3.46. The robot velocities in direction x_1 from first six initial conditions by VSGP

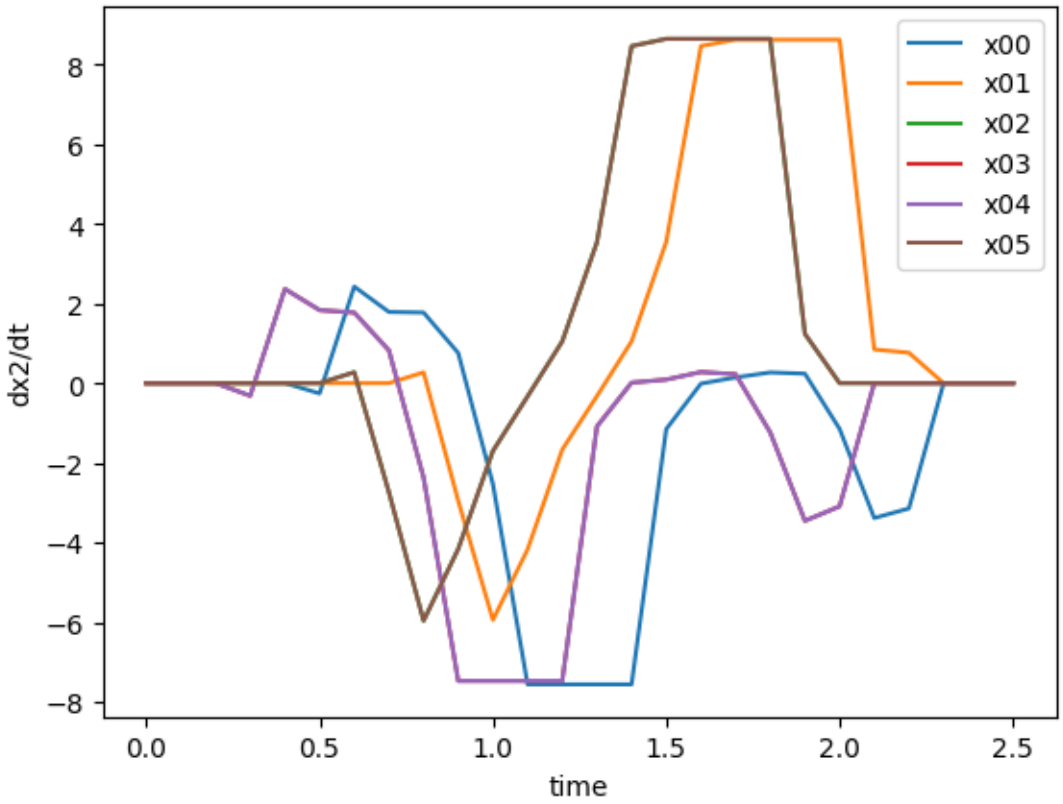

Figure 3.47. The robot velocities in direction x_2 from first six initial conditions by VSGP

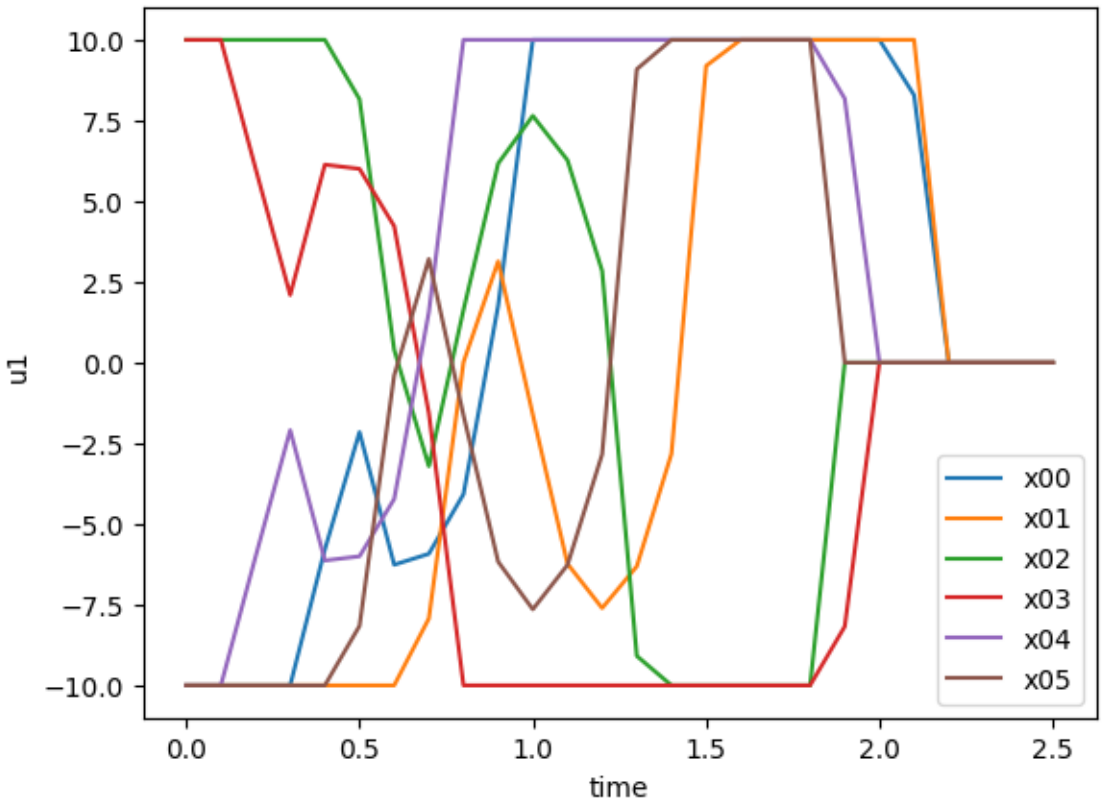

Figure 3.48. The robot control u_1 from first six initial conditions by VSGP

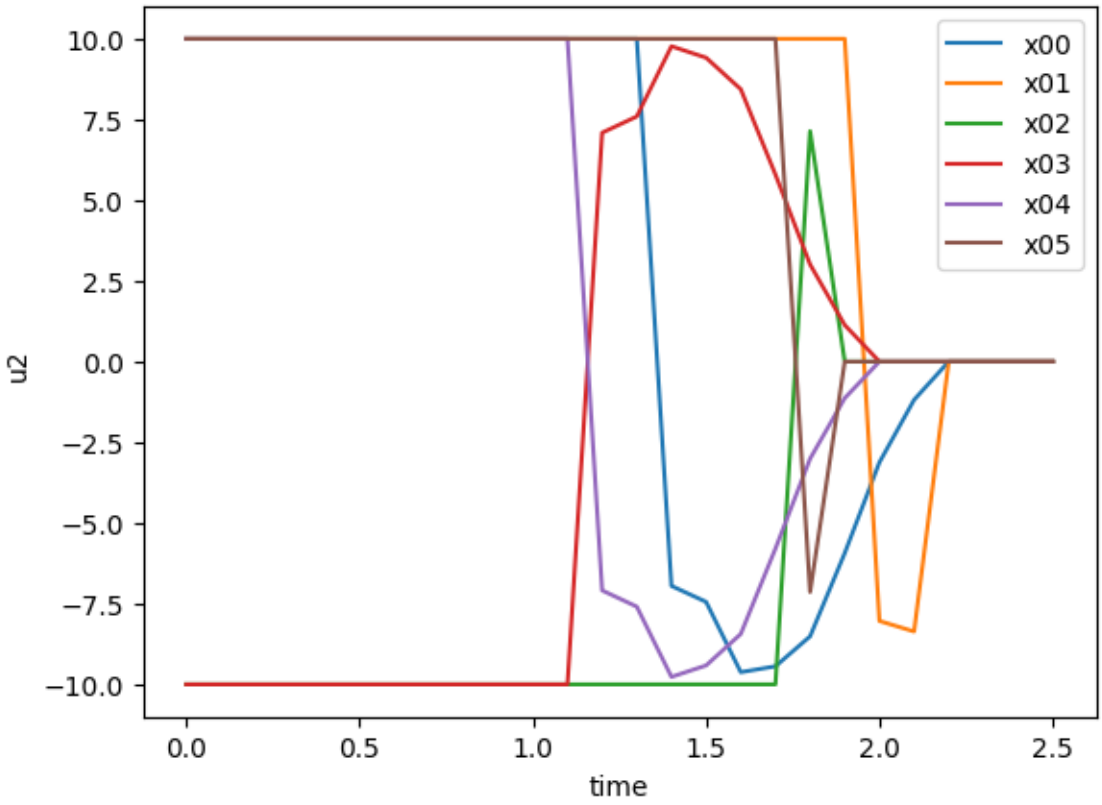

Figure 3.49. The robot control u_2 from first six initial conditions by VSGP

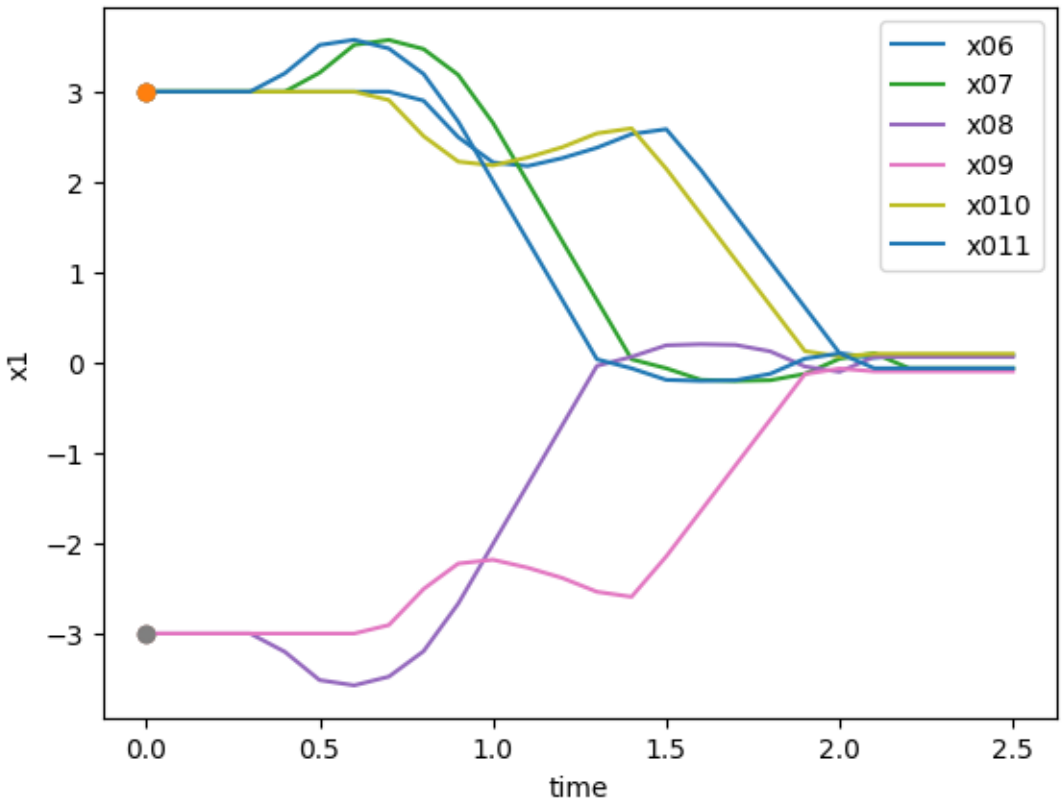

Figure 3.50. The robot displacements in direction x_1 from second six initial conditions by VSGP

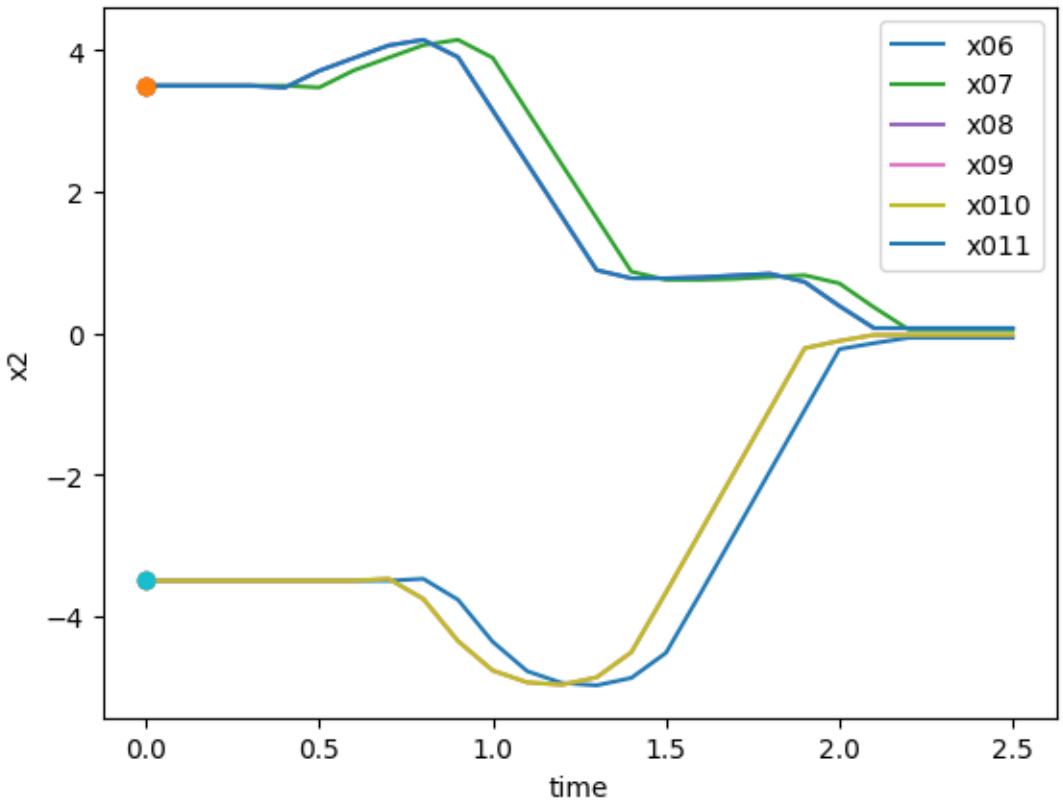

Figure 3.51. The robot displacements in direction x_2 from second six initial conditions by VSGP

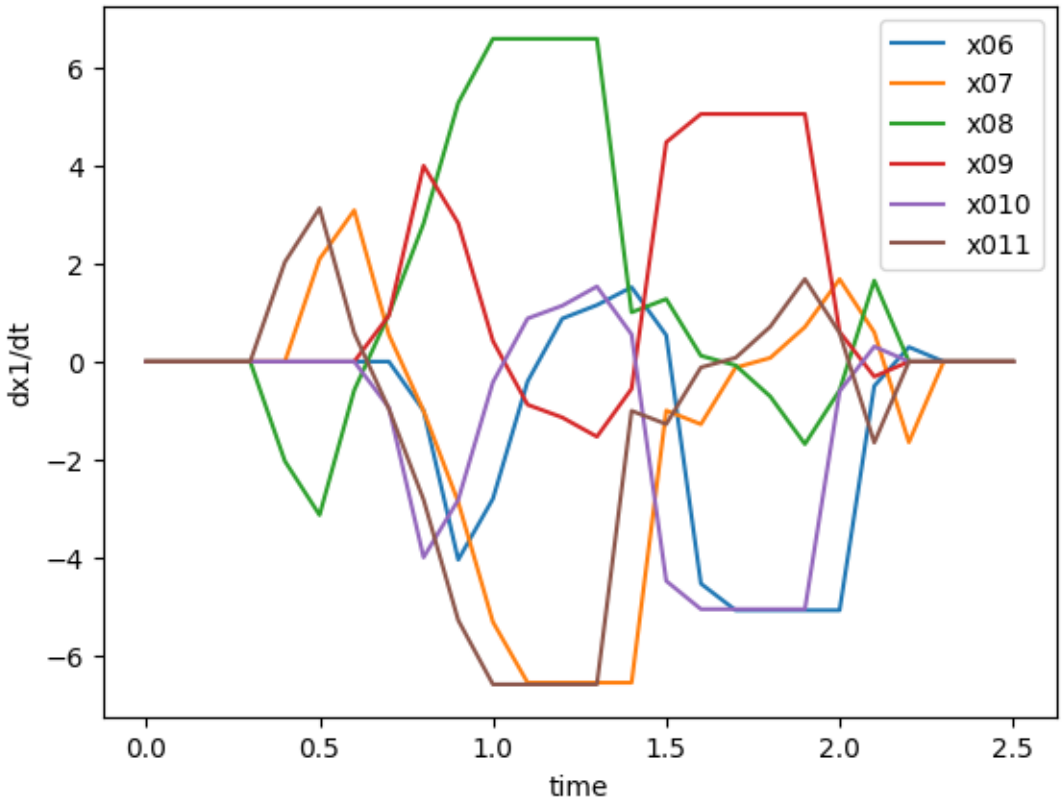

Figure 3.52. The robot velocities in direction x_1 from second six initial conditions by VSGP

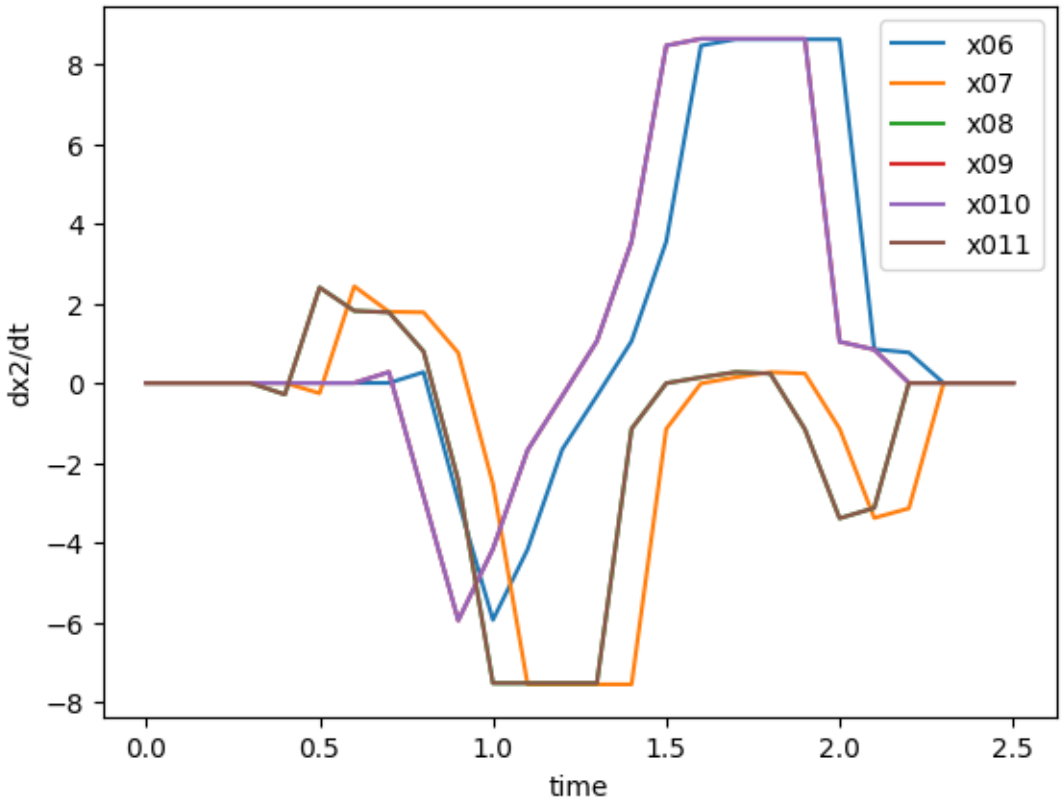

Figure 3.53. The robot velocities in direction x_2 from second six initial conditions by VSGP

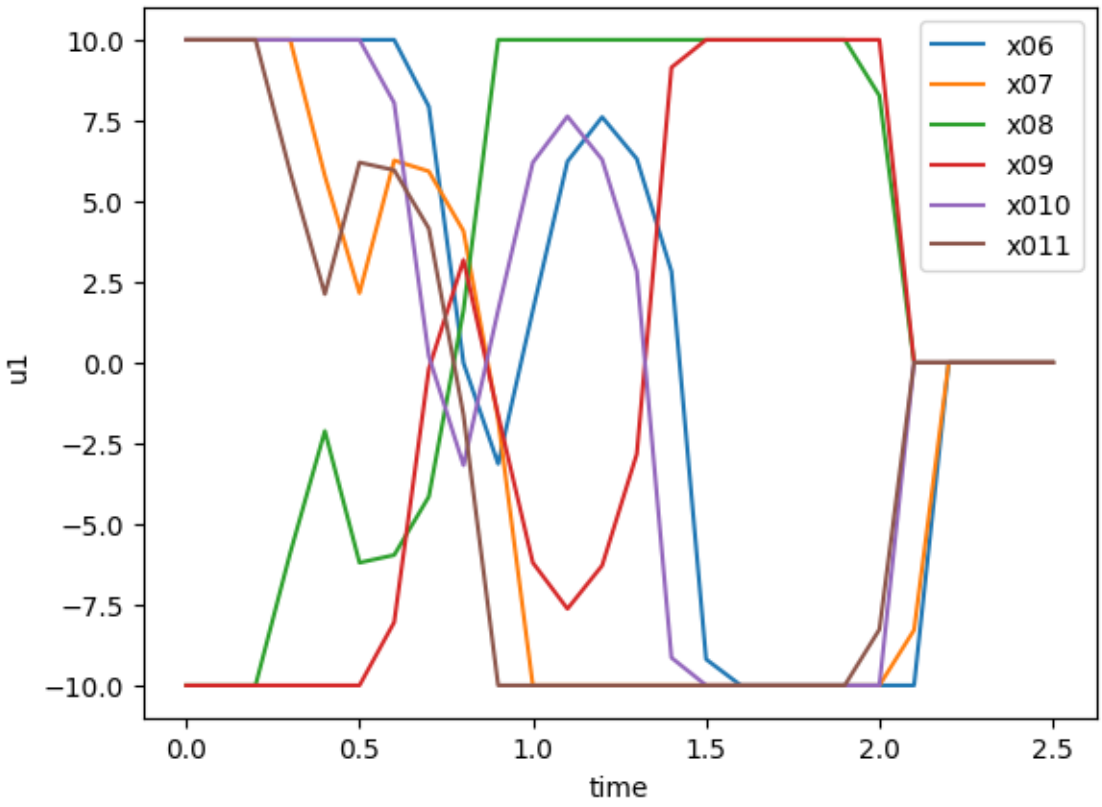

Figure 3.54. The robot control u_1 from second six initial conditions by VSGP

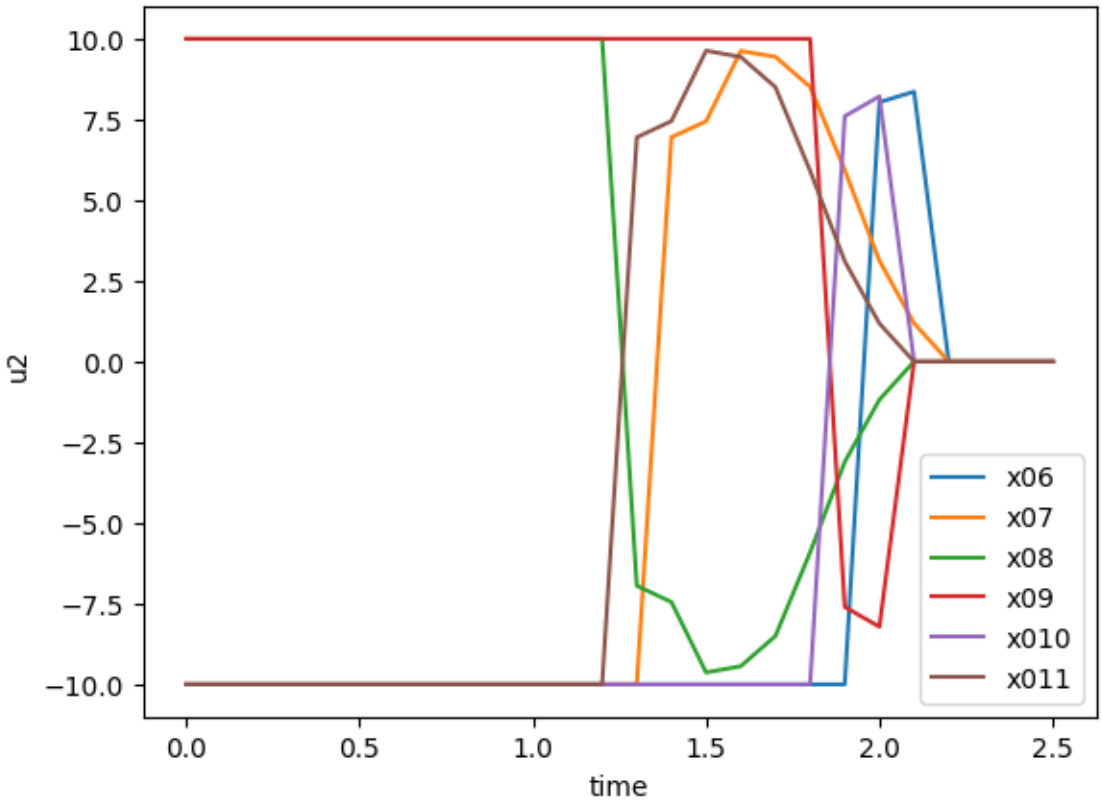

Figure 3.55. The robot control u_2 from second six initial conditions by VSGP

Case Six: Fourteen initial states are provided:

$$X_0 = \left\{\boldsymbol{x}^{0,0} = \left[-3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,1} = \left[-3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,2} = \left[3\ \ -3.5\ \ \frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,3} = \left[3\ \ 3.5\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,4} = \left[-3\ \ 3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,5} = \left[-3\ \ -3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,6} = \left[3\ \ -3.5\ -\frac{5\pi}{16}\right]^T,\ \boldsymbol{x}^{0,7} = \left[3\ \ 3.5\ \ -\frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,8} = [-3\ \ 3.5\ \ 0]^T, \boldsymbol{x}^{0,9} = [-3\ \ -3.5\ \ 0]^T, \boldsymbol{x}^{0,10} = [3\ \ -3.5\ \ 0]^T,\ \boldsymbol{x}^{0,11} = [3\ \ 3.5\ \ 0]^T, \boldsymbol{x}^{0,12} = \left[0\ \ 0\ \ \frac{5\pi}{16}\right]^T, \boldsymbol{x}^{0,13} = \left[0\ \ 0\ -\frac{5\pi}{16}\right]^T\right\}. \tag{3.39}$$

The terminal states are established as one point

$$\mathbf{x}^* = [x_1^*\ \ x_2^*\ \ x_3^*]^T = [0\ \ 0\ \ 0]^T. \tag{3.40}$$

Consequently, the ensuing mathematical expression for the control function is constructed

$$u_i^1 = \begin{cases} u_i^-, \text{if } \tilde{u}_i < u_i^- \\ u_i^+, \text{if } \tilde{u}_i > u_i^+ \\ \tilde{u}_i, \text{otherwise} \end{cases}, i = 1,2, \tag{3.41}$$

where

$$\tilde{u}_1 = \Big(\big(x_2^f - x_2\big)\big(x_1^f - x_1\big)\Big(q_1 + \big(x_2^f - x_2\big)\Big) + q_2\big(x_3^f - x_3\big) + sgn\Big(\big(x_2^f - x_2\big)\big(x_1^f - x_1\big)\Big(q_1 + \big(x_2^f - x_2\big)\Big) + q_2\big(x_3^f - x_3\big)\Big) \ln\Big(\Big|\big(x_2^f - x_2\big)\big(x_1^f - x_1\big)\Big(q_1 + \big(x_2^f - x_2\big)\Big) + q_2\big(x_3^f - x_3\big)\Big| + 1\Big) + \rho\Big(q_2\big(x_3^f - x_3\big)\Big)\Big)/2, \tag{3.42}$$

$$\tilde{u}_2 = (4 * sin(q_3) * \big(x_1^f - x_1\big))^3, \tag{3.43}$$

$$\rho(\mu) = \begin{cases} 0, & \text{if } |\mu| < \delta \\ sgn(\mu), & \text{otherwise} \end{cases} \tag{3.44}$$

$q_1 = 0.23307,\ q_2 = 6.87832, q_3 = 8.36356,\ \delta = 10^{-8}$. The quality criterion (3.4) for the variational SGP solution is $J_{syn} = 1.75102$, where $\varepsilon = 0.01, t^+ = 2.5$ sec, $\mathbf{L} = 14$, and , $p_1 = 1$.

Figure 3.56 shows the trajectories taken by one robot as it moved from fourteen initial states (3.39) to the terminal state (3.40).

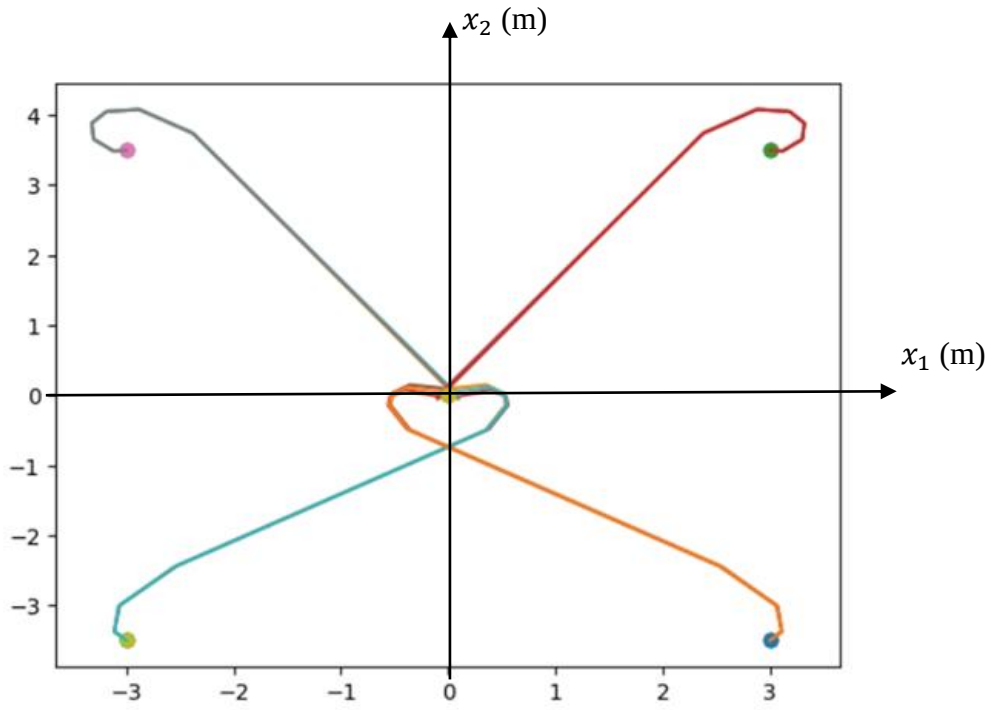

Figure 3.56. Robot trajectories with VSGP control function

The control functions (3.41) that have been acquired to guarantee the stability of the object are inserted into the model equations (3.1). The solution to the problem of control synthesis yields the emergence of a stable point of equilibrium in the space of state. The equilibrium point position is contingent upon the terminal vector (3.40).

In the second step, the particle swarm optimization (PSO) algorithm is employed to find the stabilization points, resulting in the discovery of these points for every single mobile robot.

Mathematical models of two mobile robots [214] are presented

$$\begin{aligned} \dot{x}_1^j &= 0.5\left(u_1^j + u_2^j\right)\cos\left(x_3^j\right), \\ \dot{x}_2^j &= 0.5\left(u_1^j + u_2^j\right)\sin\left(x_3^j\right), \\ \dot{x}_3^j &= 0.5\left(u_1^j - u_2^j\right), \end{aligned} \tag{3.45}$$

where $\mathbf{x}^j = [x_1^j\ x_2^j\ x_3^j]^T$ represents a state vector of robot j, $\mathbf{u}^j = [u_1^j\ u_2^j]^T$ represents a control vector of robot j, $j = 1,2$.

The control vectors elements are subject to specific constraints

$$u_i^- = -10 \le u_i^j \le 10 = u_i^+,\ \ j = 1,2,\quad i = 1,2. \tag{3.46}$$

The initial states are established

$$\mathbf{x}^1(0) = \mathbf{x}^{0,1} = [0\ \ 0\ \ 0]^T,$$
$$\mathbf{x}^2(0) = \mathbf{x}^{0,2} = [10\ \ 10\ \ 0]^T. \tag{3.47}$$

The terminal states are established

$$\mathbf{x}^1(t_f) = \mathbf{x}^{f,1} = [10\ \ 10\ \ 0]^T,$$
$$\mathbf{x}^2(t_f) = \mathbf{x}^{f,2} = [0\ \ 0\ \ 0]^T, \tag{3.48}$$

where

$$t_f = \begin{cases} t, & \text{if } \ t \le t^+ \text{ and } \left\|x^f - x(t)\right\| \le \varepsilon \\ t^+, & \text{otherwise} \end{cases} \tag{3.49}$$

and

$$\left\|x^f - x(t)\right\| = \sqrt{\sum_{i=1}^{3}(x_i^f - x_i(t))^2}. \tag{3.50}$$

The constraints of static phase are presented

$$r_{st} - \sqrt{\left(x_1^j - x_{1,st}\right)^2 + \left(x_2^j - x_{2,st}\right)^2} \le 0,\ j = 1,2. \tag{3.51}$$

where $r_{st}, x_{1,st}, x_{2,st}$ are provided parameters (radius and coordinates of center) of the constraints of static phase, $st = 1, \dots, P_t$, P_t represents the total number of phase constraints.

The constraints of dynamic phase are provided

$$r_d - \sqrt{(x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2} \le 0, \tag{3.52}$$

where r_d the minimal acceptable secure distance between robots, $r_d = 2$.

The next quality functional is defined for the solution of optimal control:

$$\begin{aligned} J_{opt} = t_f + c_1 \sum_{st=1}^{5}\sum_{j=1}^{2}\int_0^{t_f} \vartheta\left(r_{st} - \sqrt{\left(x_1^j - x_{1,st}\right)^2 + \left(x_2^j - x_{2,st}\right)^2}\right) \\ + c_2 \int_0^{t_f} \vartheta\left(r_d - \sqrt{(x_1^1 - x_1^2)^2 + (x_2^1 - x_2^2)^2}\right) \\ + c_3 \sum_{j=1}^{2}\sqrt{\sum_{i=1}^{3}(x_i^{f,j} - x_i^j)^2} \end{aligned} \tag{3.53}$$

where $\vartheta(A)$ represents the Heaviside step function

$$\vartheta(A) = \begin{cases} 1, & \text{if } A > 0 \\ 0, & \text{otherwise} \end{cases}, \tag{3.54}$$

The problem (3.45)–(3.54) can be solved by the utilization of the technique of synthesized optimal control and for all next cases, the control functions (3.42)-(3.44) are used.

Case 1: The particle swarm optimization (PSO) algorithm is employed for the purpose of finding the stabilization points and it is possible to search the control function (the equilibrium points) as a piecewise constant function. The constraints pertaining to the elements of points are as follows:

$$\begin{aligned} -2 \le x_1^{j,*,i} \le 12, \\ -2 \le x_2^{j,*,i} \le 12, \\ -\frac{\pi}{2} \le x_3^{j,*,i} \le \frac{\pi}{2}. \end{aligned} \tag{3.55}$$

where $c_1 = 2,\ c_2 = 4,\ c_3 = 2.5,\ P_t = 4, r_1 = 1.5, r_2 = 2, r_3 = 2, r_4 = 1.5, x_{1,1} = 1.5, x_{1,2} = 2, x_{1,3} = 8, x_{1,4} = 8.5, x_{2,1} = 2.5, x_{2,2} = 7.5, x_{2,3} = 2.5, x_{2,4} = 7.5,\ \varepsilon = 0.01$ and $t^+ = 2.7$ sec.

The three points for each mobile robot have the subsequent coordinates in the state space $\{x_1, x_2, x_3\}$:

$$\begin{aligned} \mathbf{x}^{1,*,1} &= [4.462 \quad -1.8995 \quad 1.5701]^T, \\ \mathbf{x}^{1,*,2} &= [10.7687 \quad 11.8605 \quad -0.6636]^T, \\ \mathbf{x}^{1,*,3} &= [9.7937 \quad 11.4512 \quad 0.2058]^T, \\ \mathbf{x}^{2,*,1} &= [-0.6247 \quad 9.2577 \quad -0.382\ ]^T, \\ \mathbf{x}^{2,*,2} &= [2.248 \quad 10.6802 \quad -0.2329]^T, \\ \mathbf{x}^{2,*,3} &= [0.1365 \quad 5.131 \quad -0.477\ ]^T. \end{aligned} \tag{3.56}$$

where the first three points are for the first robot and the other points for the second one.

In Figures (3.57)-(3.63) the findings of the simulation are laid out. Figure 3.57 displays suitable trajectories generated by the pair of mobile robots in the $\{x_1, x_2\}$ plane. In the graphical representation, the red curve traces the motion of the first robot, and the yellow curve the second. Purple circles highlight locations where phase constraints are enforced. Furthermore, the equilibrium configurations for the first and second robots are denoted by small green and black squares, respectively. These markers correspond bijectively to the three equilibrium points per agent that were identified

through the solution of the underlying optimal control problem (3.56). As evident from the observation, the two robots have successfully attained their terminal states without violating phase constraints. The functional value in Equation (3.53) was $J_{opt} = 2.7000$.

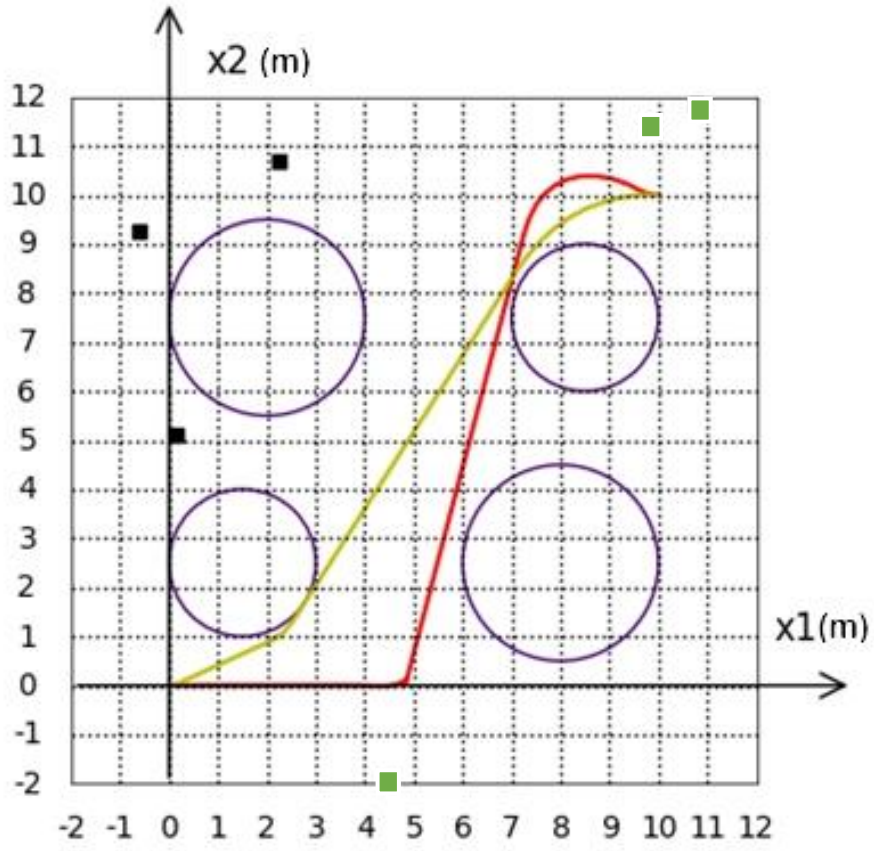

Figure 3.57. Synthesized optimal control trajectories for two robots in the $\{x_1, x_2\}$ plane

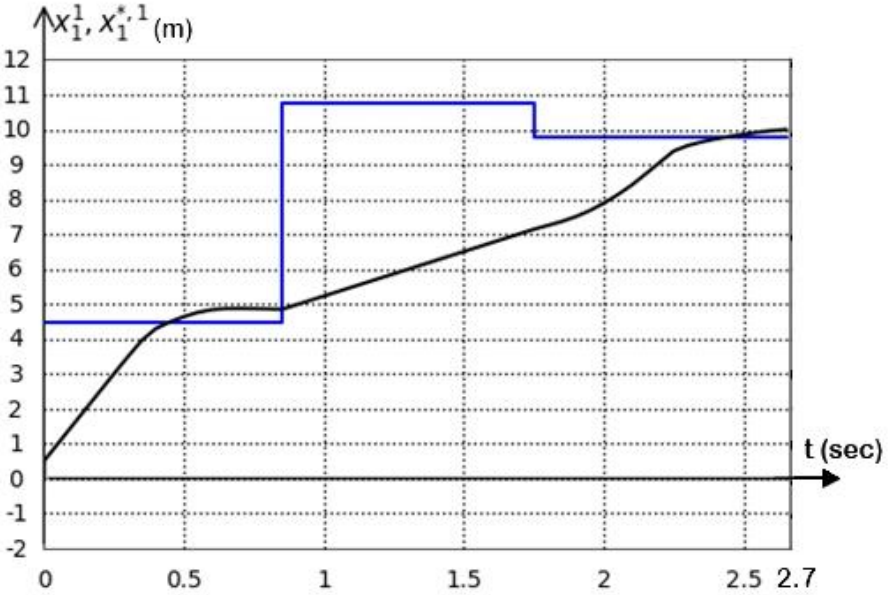

Figure 3.58. The variable x_1^1 in black and effective control $x_1^{*,1}$ in blue

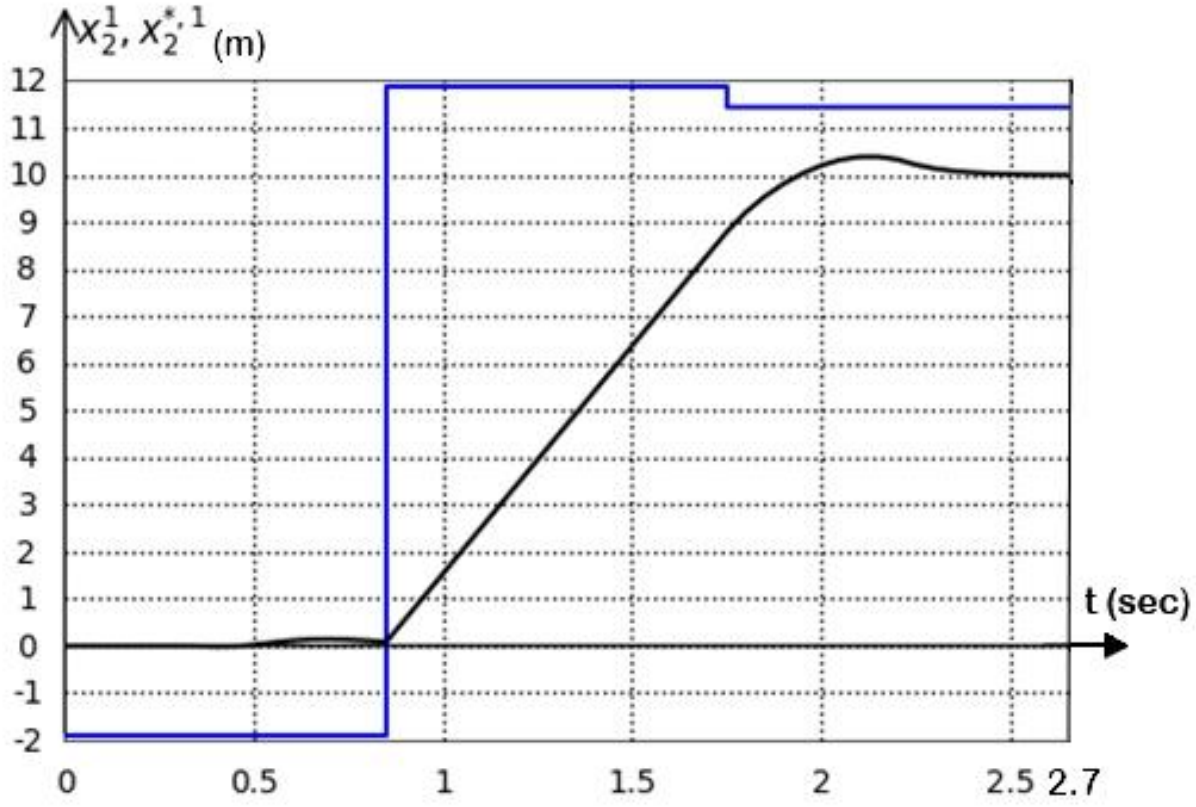

Figure 3.59. The variable x_2^1 in black and effective control $x_2^{*,1}$ in blue

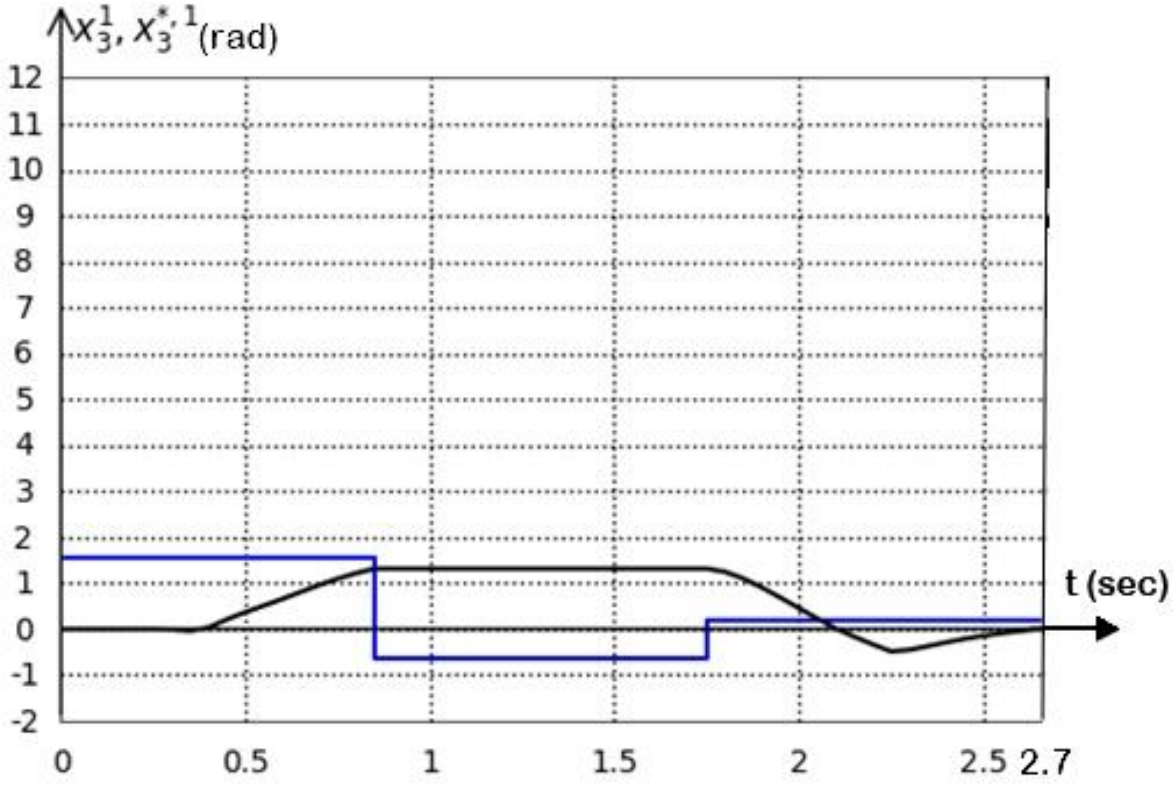

Figure 3.60. The variable x_3^1 in black and effective control $x_3^{*,1}$ in blue

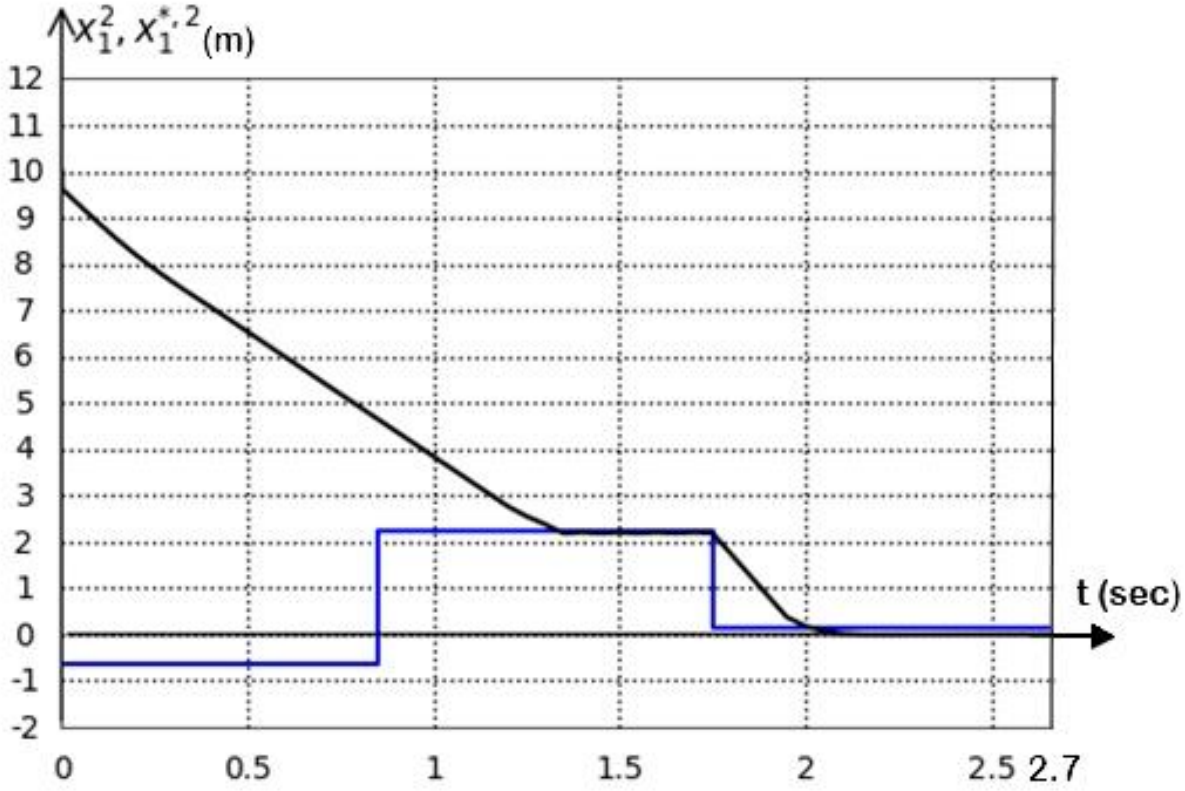

Figure 3.61. The variable x_1^2 in black and effective control $x_1^{*,2}$ in blue

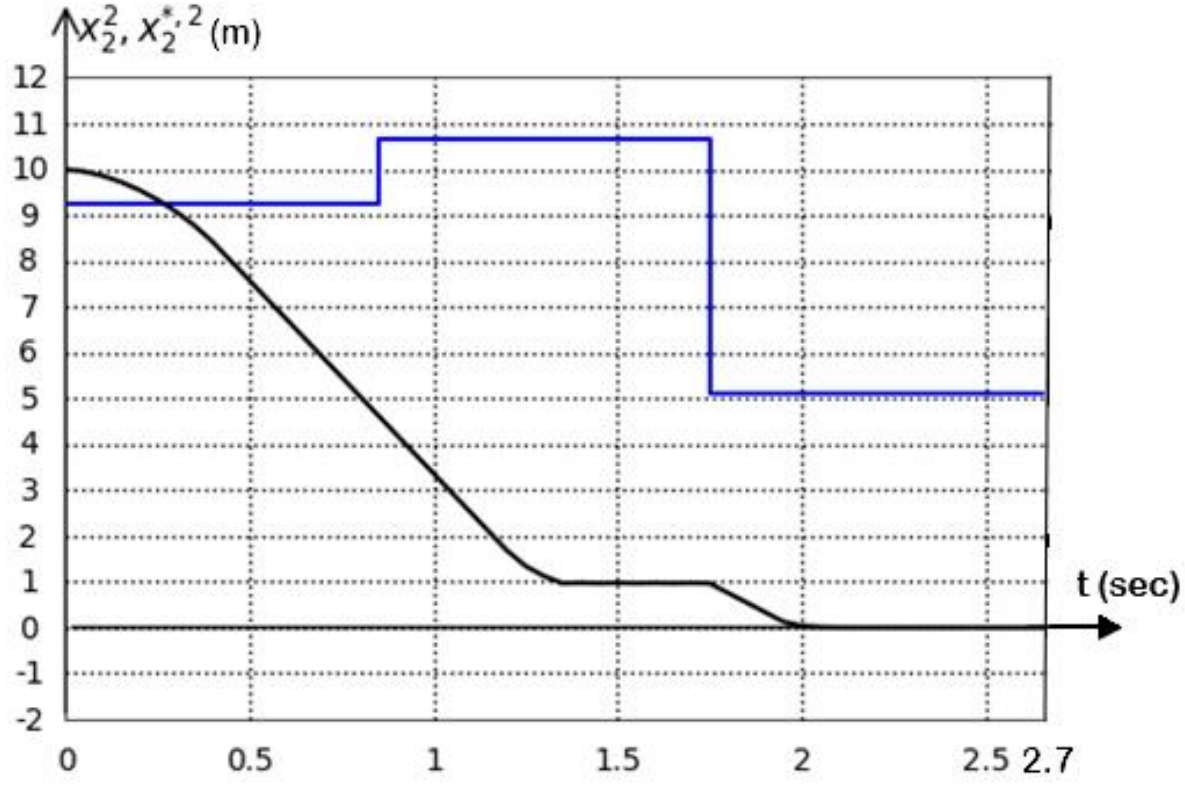

Figure 3.62. The variable x_2^2 in black and effective control $x_2^{*,2}$ in blue

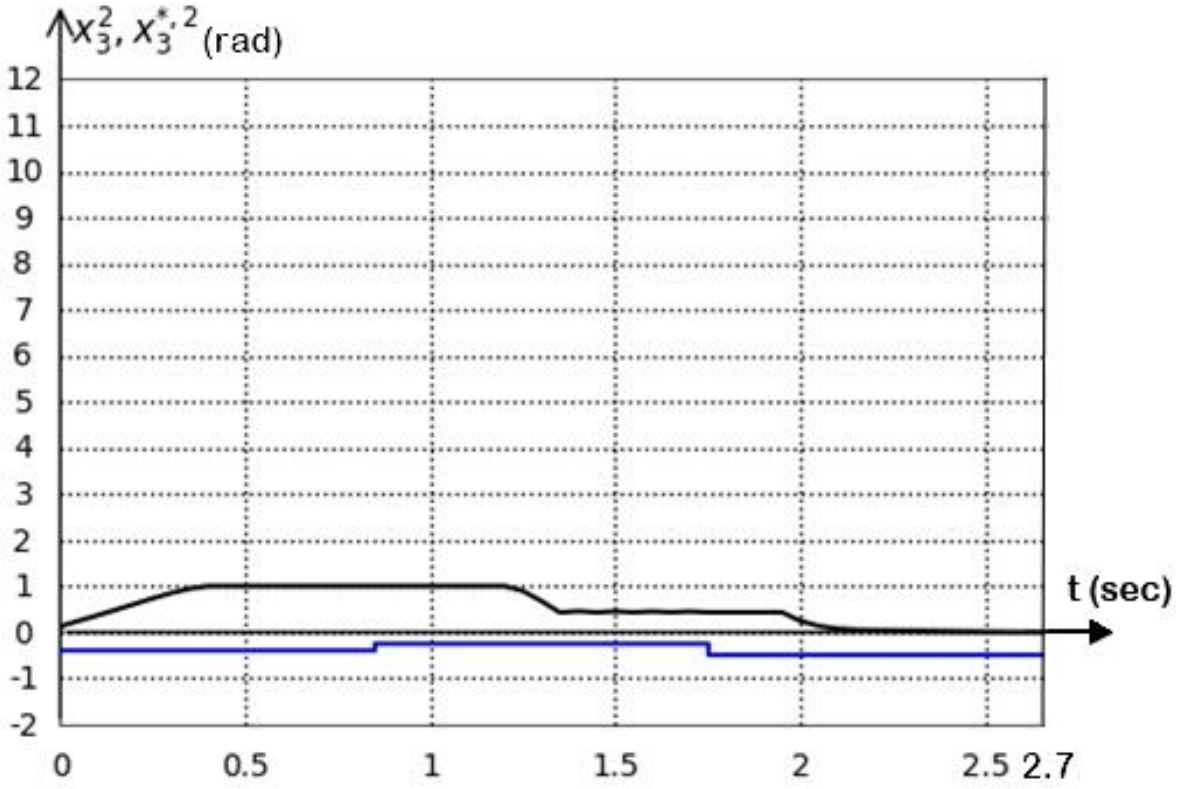

Figure 3.63. The variable x_3^2 in black and effective control $x_3^{*,2}$ in blue

Case 2: The particle swarm optimization (PSO) algorithm is employed for the purpose of finding the stabilization points and it is possible to search the control function (the equilibrium points) as a piecewise constant function. The constraints pertaining to the elements of points are as follows:

$$\begin{aligned} -2 \le x_1^{j,*,i} \le 12, \\ -2 \le x_2^{j,*,i} \le 12, \\ -\frac{\pi}{2} \le x_3^{j,*,i} \le \frac{\pi}{2}. \end{aligned} \tag{3.57}$$

where $c_1 = 2, \quad c_2 = 4, \quad c_3 = 2.5, \quad P_t = 5, r_1 = 1.5, r_2 = 1.5, r_3 = 1.5, r_4 = 1.5, r_5 = 1.5, x_{1,1} = 1.5, x_{1,2} = 1.5, x_{1,3} = 8.5, x_{1,4} = 8.5, x_{1,5} = 5, x_{2,1} = 2.5, x_{2,2} = 7.5, x_{2,3} = 2.5, x_{2,4} = 7.5, x_{2,5} = 5, \varepsilon = 0.01$ and $t^+ = 2.7$ sec.

The three points for each mobile robot have the subsequent coordinates in the state space $\{x_1, x_2, x_3\}$:

$$\mathbf{x}^{1,*,1} = [2.6034 \quad -2 \quad 1.5708]^T,$$

$$\mathbf{x}^{1,*,2} = [12 \quad 8.3517 \quad 0.0366]^T,$$

$$\mathbf{x}^{1,*,3} = [9.9742 \quad 10.1032 \quad 1.5708]^T,$$

$$\mathbf{x}^{2,*,1} = [12 \quad 5.5246 \quad 1.5708\ ]^T,$$

$$\mathbf{x}^{2,*,2} = [7.648 \quad -2 \quad 0.2991]^T,$$

$$\mathbf{x}^{2,*,3} = [0.1696 \quad 11.9731 \quad -0.8627]^T. \tag{3.58}$$

where the first three points are for the first robot and the other points for the second one.

In Figures (3.64)-(3.70) the findings of the simulation are laid out. Figure 3.64 displays suitable trajectories generated by the pair of mobile robots in the $\{x_1, x_2\}$ plane. In the graphical representation, the red curve traces the motion of the first robot, and the yellow curve the second. Purple circles highlight locations where phase constraints are enforced. Furthermore, the equilibrium configurations for the first and second robots are denoted by small green and black squares, respectively. These markers correspond bijectively to the three equilibrium points per agent that were identified through the solution of the underlying optimal control problem (3.58). As evident from the observation, the two robots have successfully attained their terminal states without violating phase constraints. The functional value in Equation (3.53) was $J_{opt} = 2.7335$.

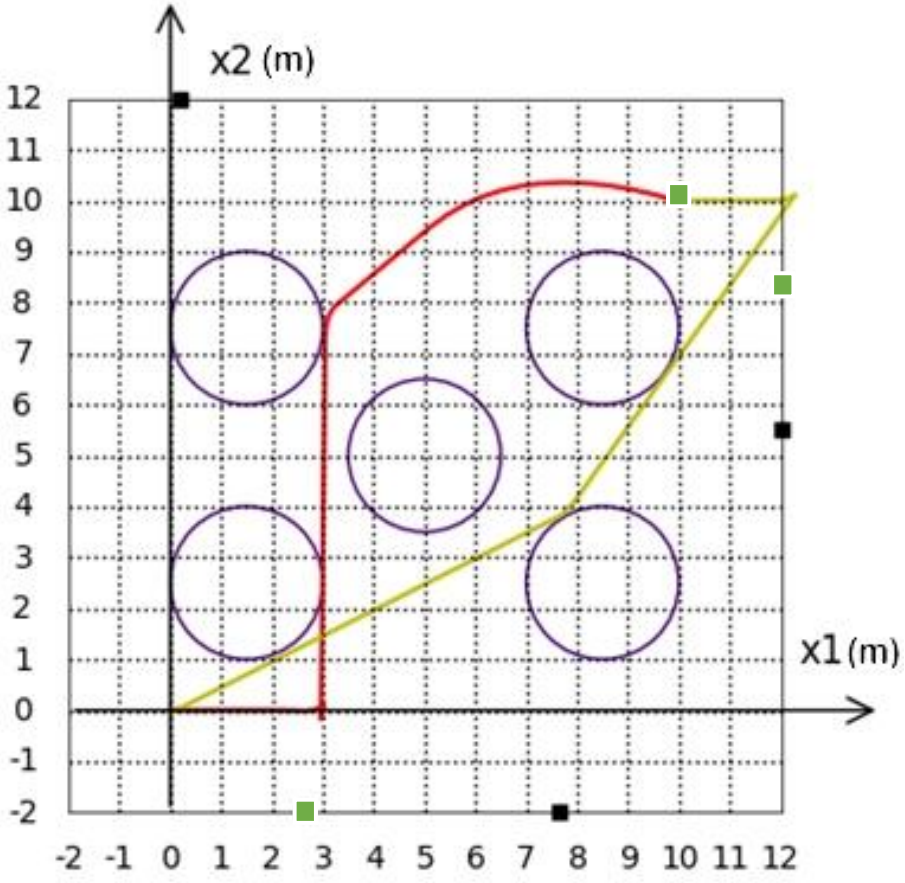

Figure 3.64. Synthesized optimal control trajectories for two robots in the $\{x_1, x_2\}$ plane

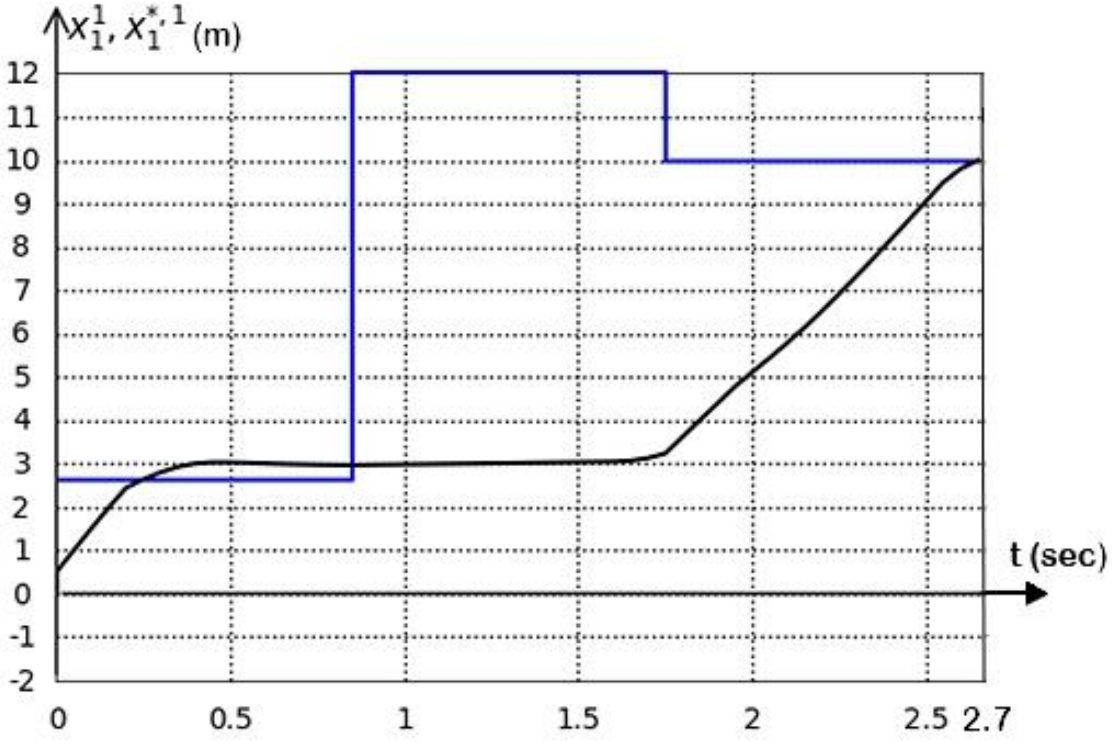

Figure 3.65. The variable x_1^1 in black and effective control $x_1^{*,1}$ in blue

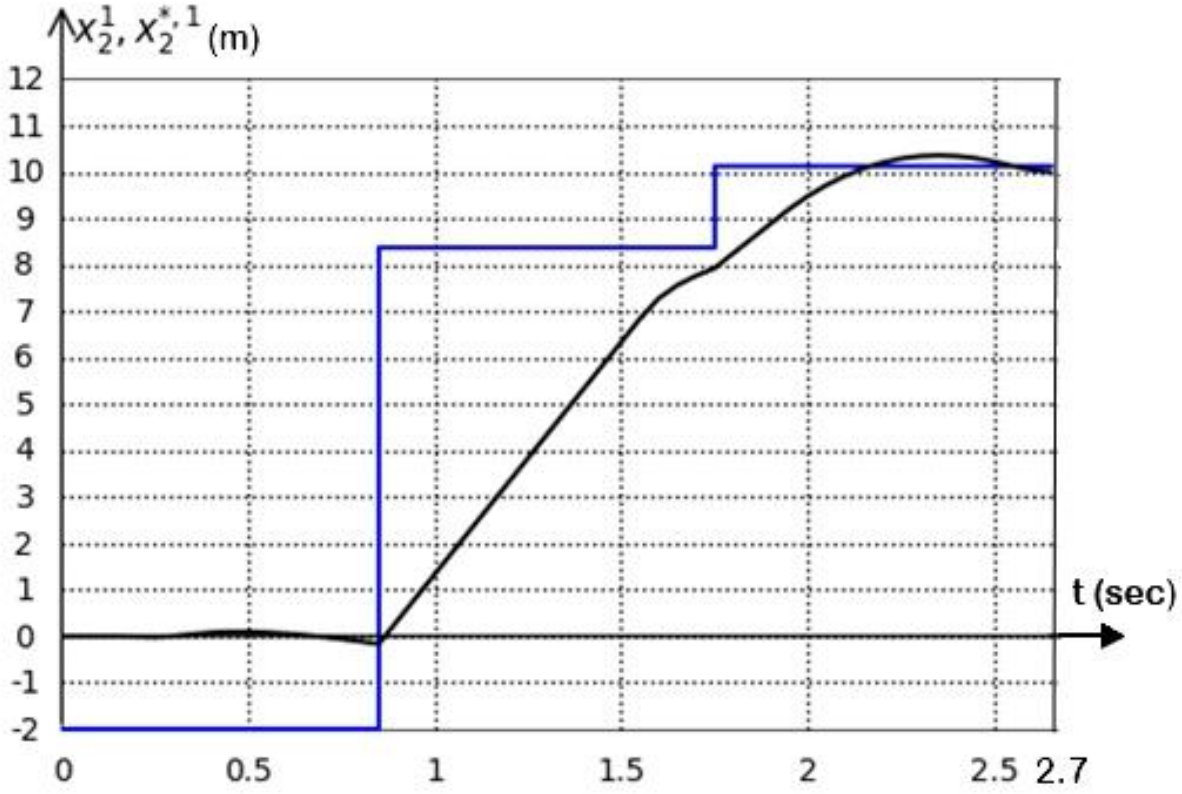

Figure 3.66. The variable x_2^1 in black and effective control $x_2^{*,1}$ in blue

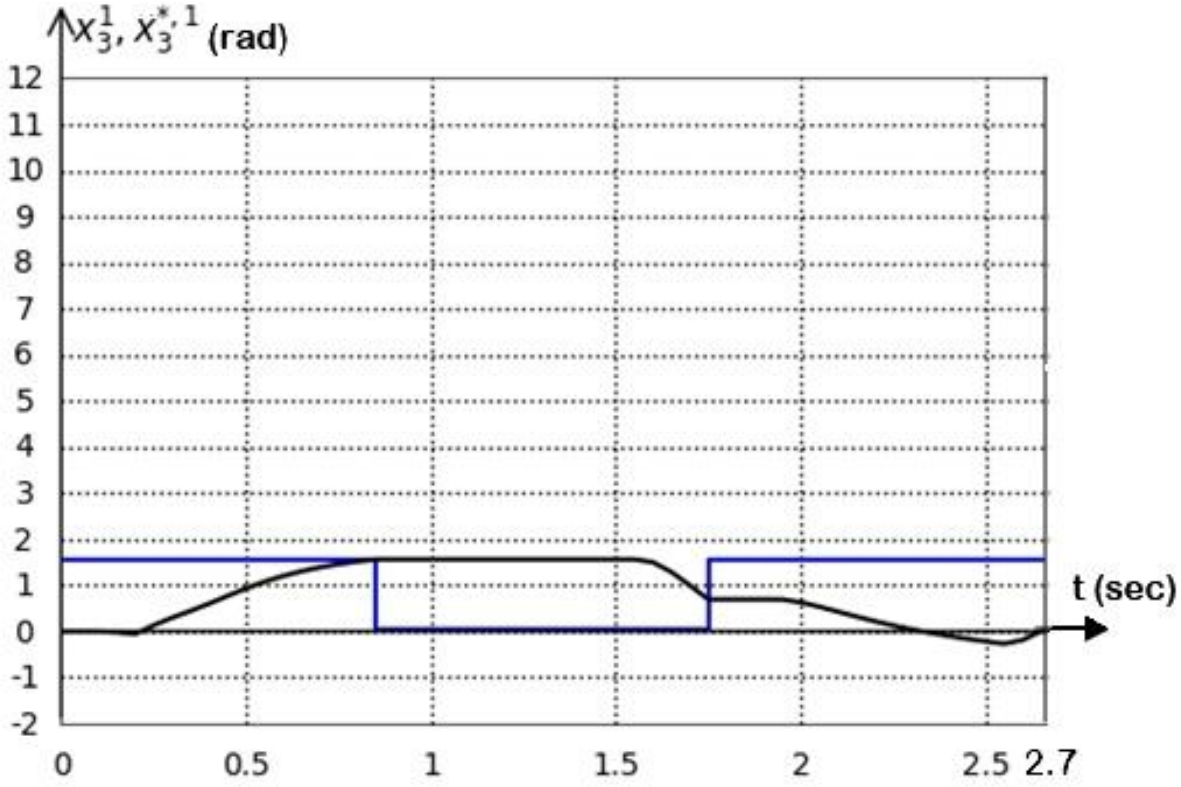

Figure 3.67. The variable x_3^1 in black and effective control $x_3^{*,1}$ in blue

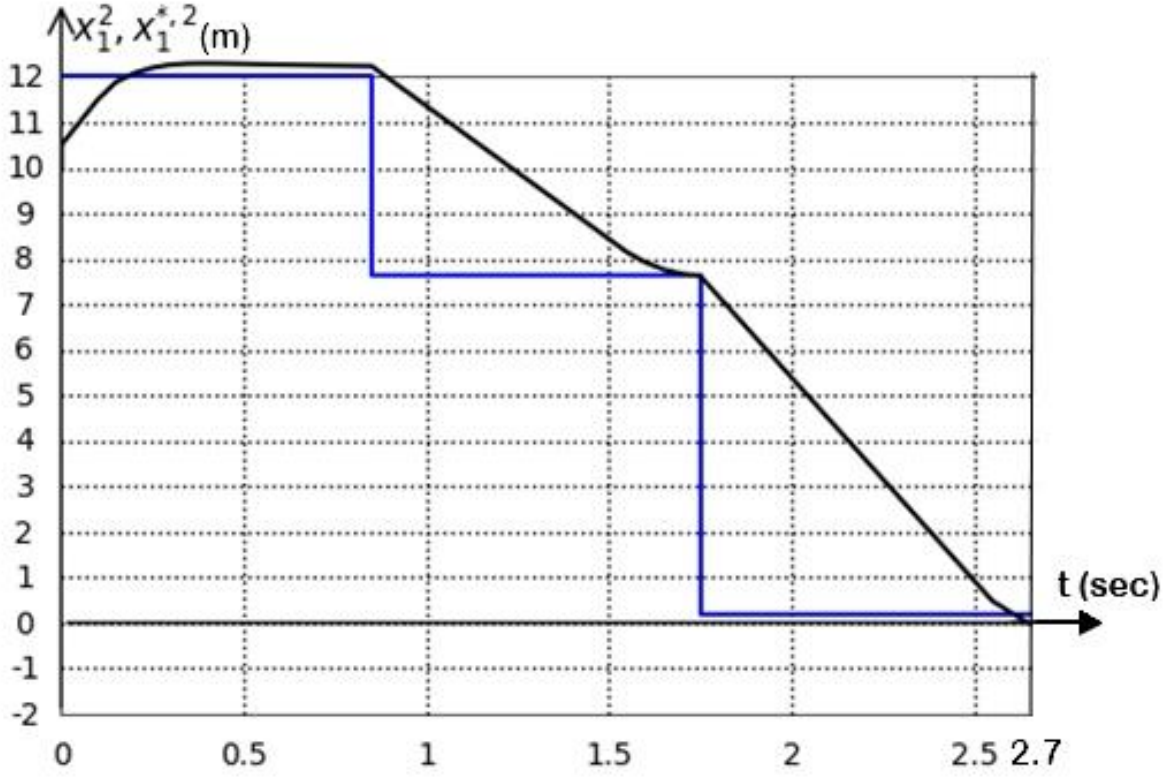

Figure 3.68. The variable x_1^2 in black and effective control $x_1^{*,2}$ in blue

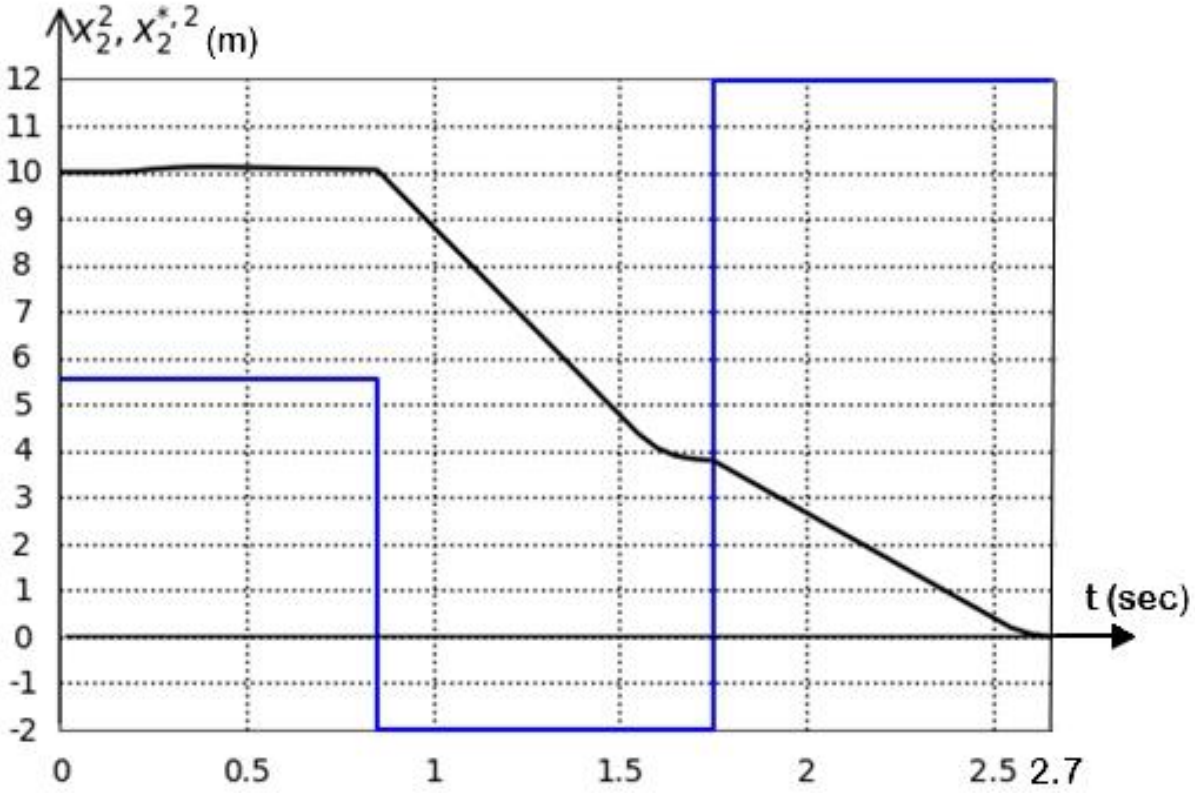

Figure 3.69. The variable x_2^2 in black and effective control $x_2^{*,2}$ in blue

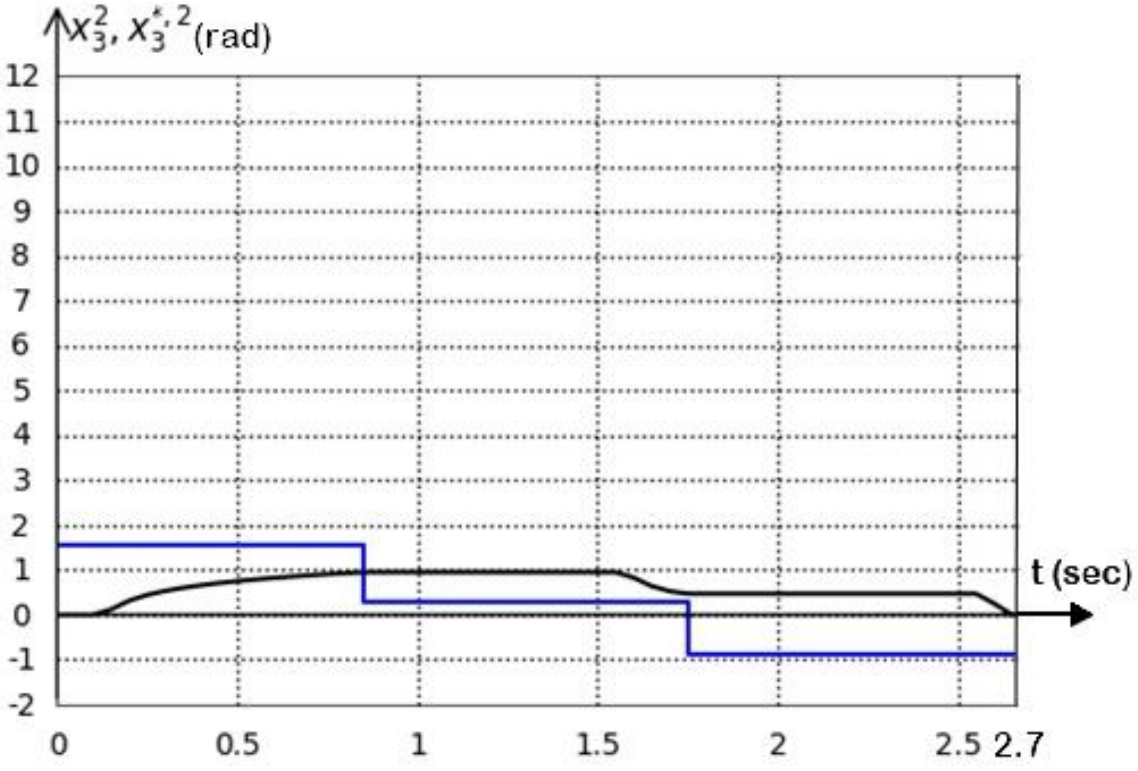

Figure 3.70. The variable x_3^2 in black and effective control $x_3^{*,2}$ in blue

Case 3: The particle swarm optimization (PSO) algorithm is employed for the purpose of finding the stabilization points and it is possible to search the control function (the equilibrium points) as a piecewise constant function. The constraints pertaining to the elements of points are as follows:

$$\begin{aligned} -2 \leq x_1^{j,*,i} \leq 12, \\ -2 \leq x_2^{j,*,i} \leq 12, \\ -\frac{\pi}{2} \leq x_3^{j,*,i} \leq \frac{\pi}{2}. \end{aligned} \tag{3.59}$$

where $c_1 = 2, \quad c_2 = 4, \quad c_3 = 2.5, \quad P_t = 5, r_1 = 2, r_2 = 1.5, r_3 = 2, r_4 = 2, r_5 = 2, x_{1,1} = 0, x_{1,2} = 5, x_{1,3} = 10, x_{1,4} = 5, x_{1,5} = 5, x_{2,1} = 5, x_{2,2} = 5, x_{2,3} = 5, x_{2,4} = 0, x_{2,5} = 10, \varepsilon = 0.01$ and $t^+ = 2.8$ sec.

The four points for each mobile robot have the subsequent coordinates in the state space $\{x_1, x_2, x_3\}$:

$$\begin{aligned} \mathbf{x}^{1,*,1} &= [-2 \quad 4.313 \quad 1.5681]^T, \\ \mathbf{x}^{1,*,2} &= [0.8889 \quad -0.4234 \quad 1.5649]^T, \\ \mathbf{x}^{1,*,3} &= [5.5287 \quad 7.4658 \quad 1.5708]^T, \\ \mathbf{x}^{1,*,4} &= [12 \quad 9.9375 \quad 1.5708]^T, \\ \mathbf{x}^{2,*,1} &= [12 \quad 6.0535 \quad 1.3051\ ]^T, \\ \mathbf{x}^{2,*,2} &= [-2 \quad -0.3394 \quad 1.5708]^T, \\ \mathbf{x}^{2,*,3} &= [4.181 \quad 0.1354 \quad 0.1268]^T \\ \mathbf{x}^{2,*,4} &= [0.1342 \quad 8.9329 \quad -1.5708]^T. \end{aligned} \tag{3.60}$$

where the first four points are for the first robot and the other points for the second one.

In Figures (3.71)-(3.77) the findings of the simulation are laid out. Figure 3.71 displays suitable trajectories generated by the pair of mobile robots in the $\{x_1, x_2\}$ plane. In the graphical representation, the red curve traces the motion of the first robot, and the yellow curve the second. Purple circles highlight locations where phase constraints are enforced. Furthermore, the equilibrium configurations for the first and second robots are denoted by small green and black squares, respectively. These markers correspond bijectively to the four equilibrium points per agent that were identified through the solution of the underlying optimal control problem (3.60). As evident from the observation,

the two robots have successfully attained their terminal states without violating phase constraints. The functional value in Equation (3.53) was $J_{opt} = 2.8000$.

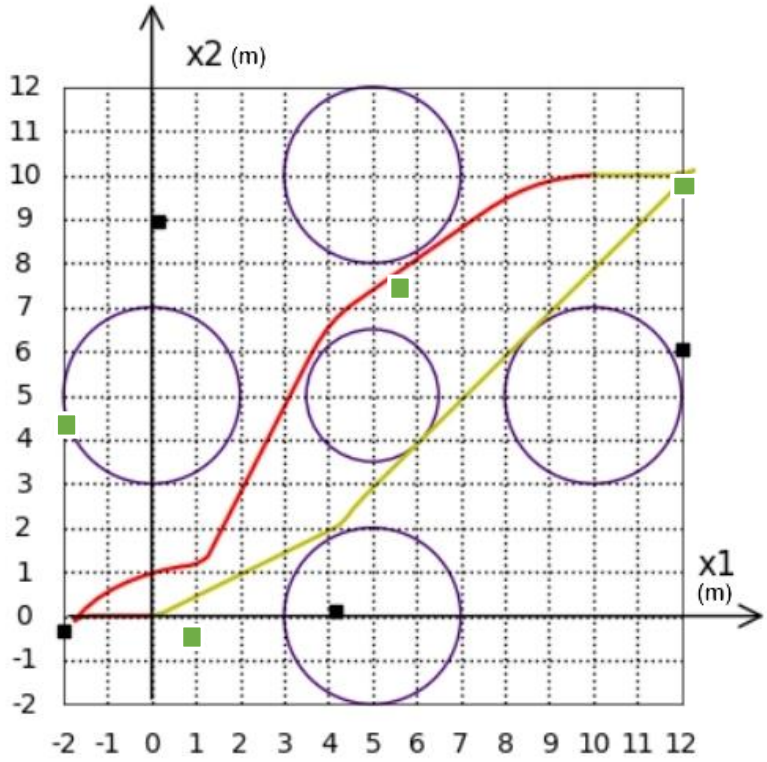

Figure 3.71. Synthesized optimal control trajectories for two robots in the $\{x_1, x_2\}$ plane

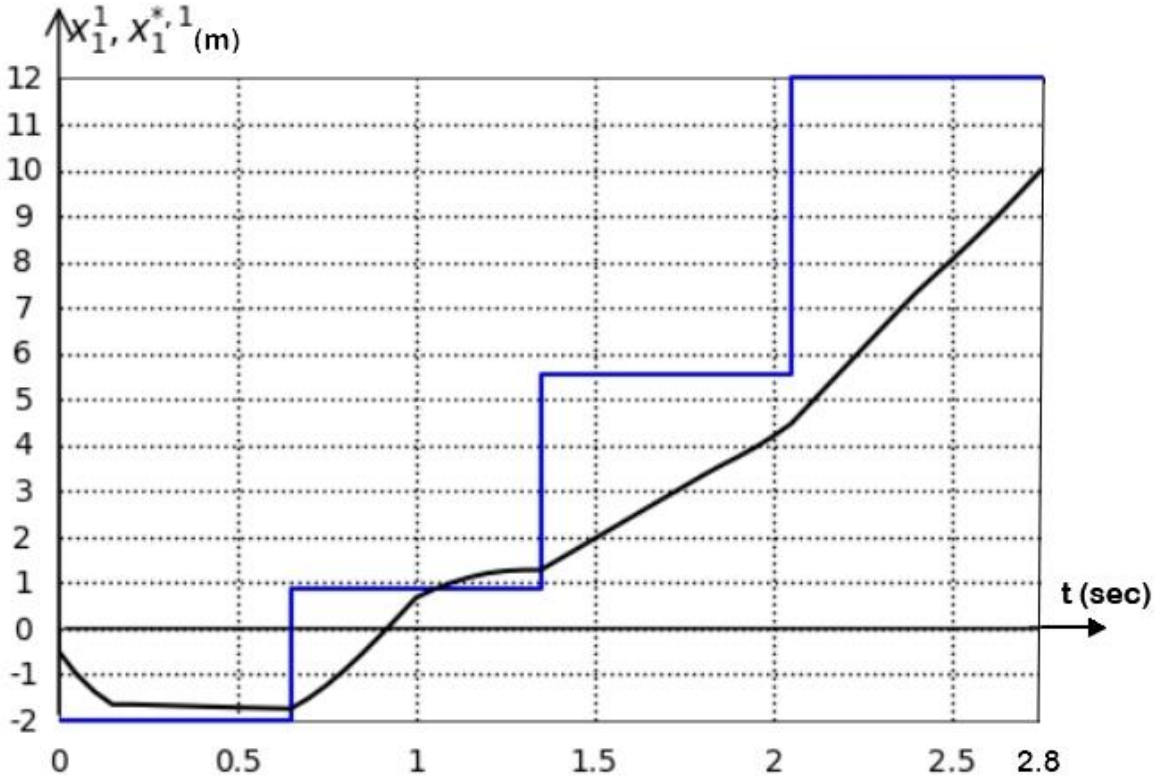

Figure 3.72. The variable x_1^1 in black and effective control $x_1^{*,1}$ in blue

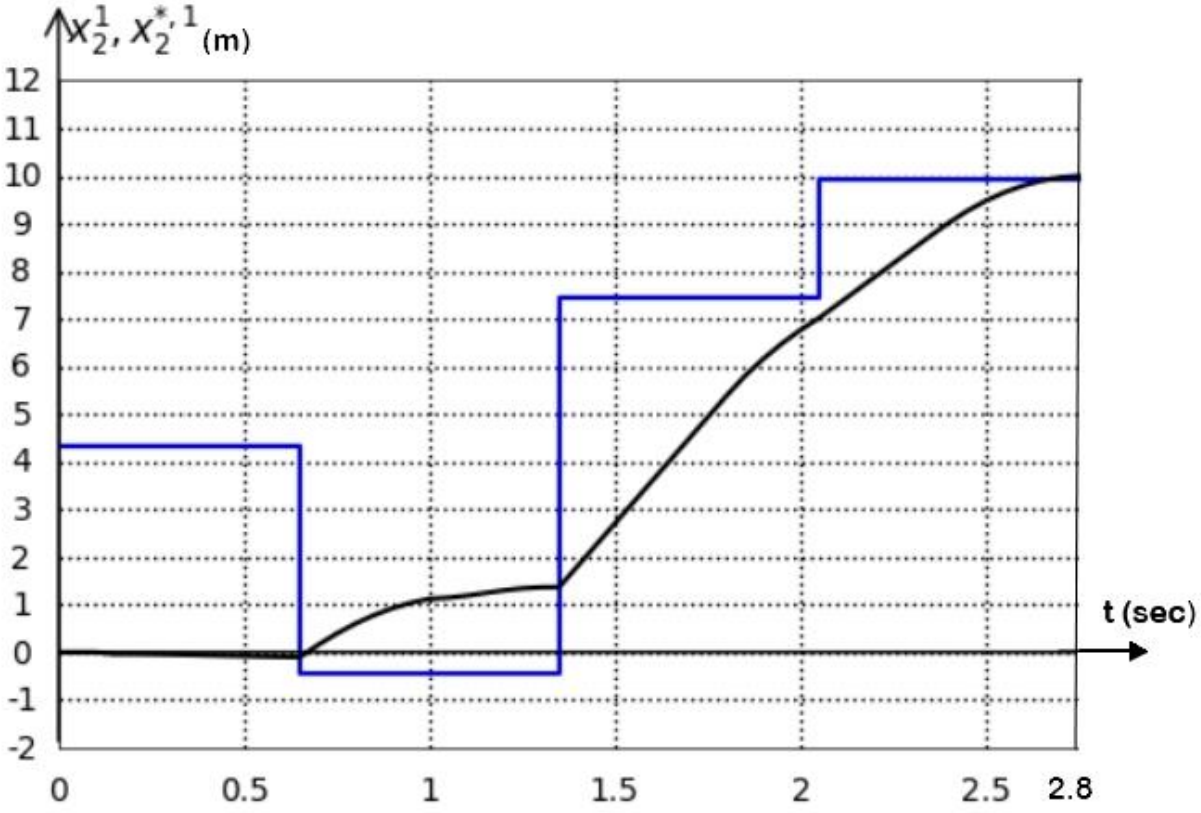

Figure 3.73. The variable x_2^1 in black and effective control $x_2^{*,1}$ in blue

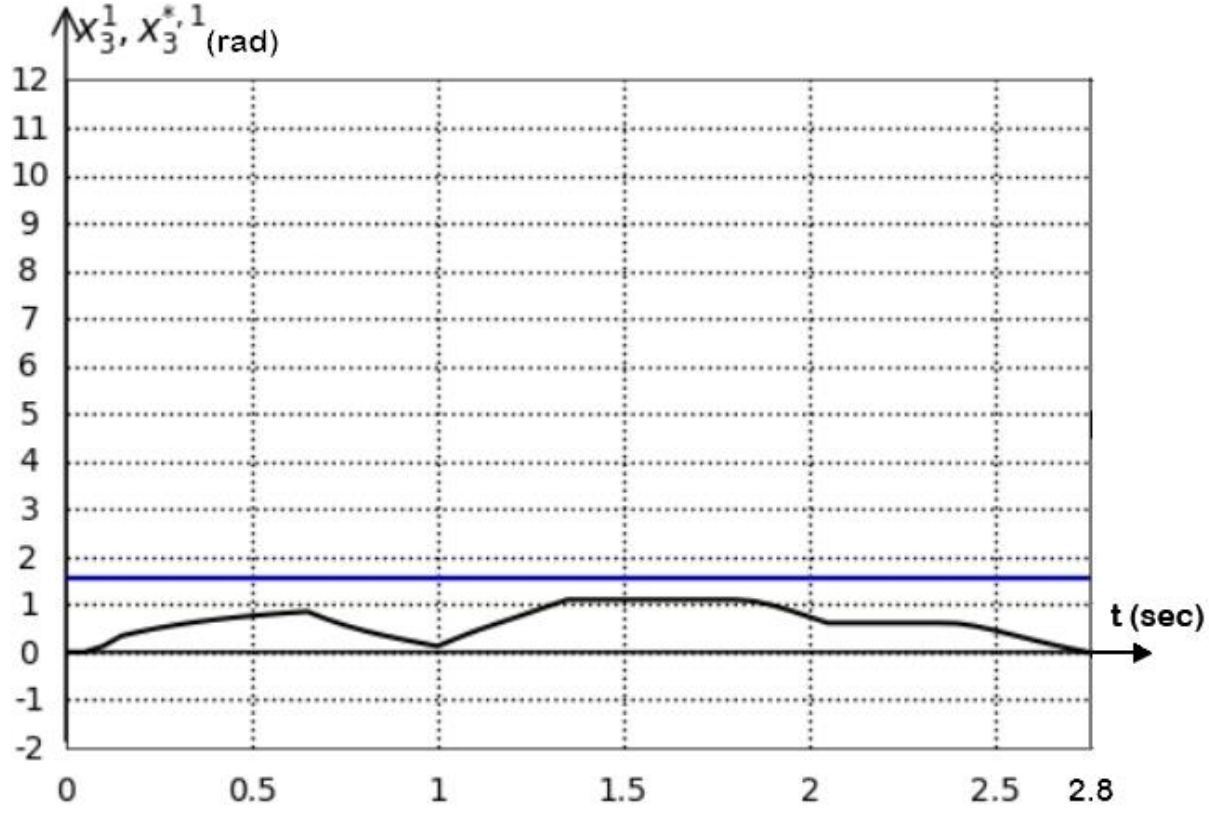

Figure 3.74. The variable x_3^1 in black and effective control $x_3^{*,1}$ in blue

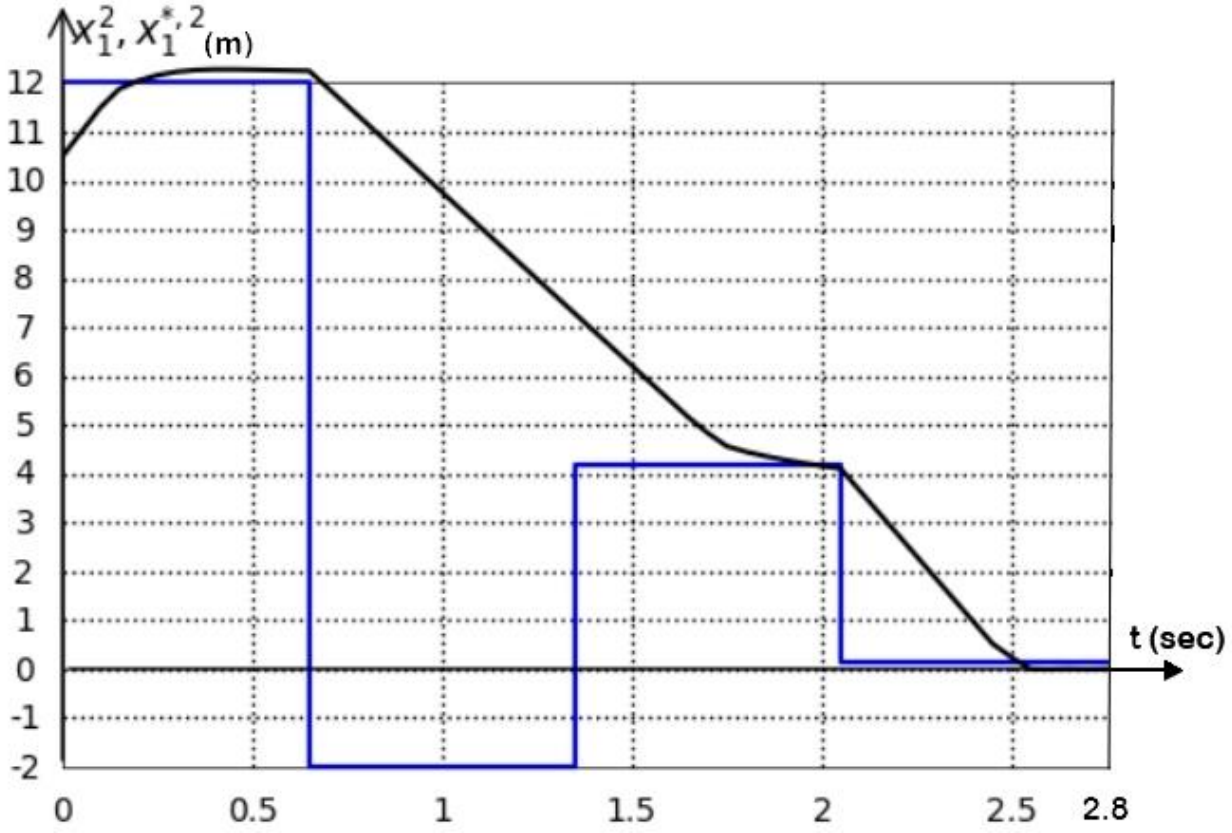

Figure 3.75. The variable x_1^2 in black and effective control $x_1^{*,2}$ in blue

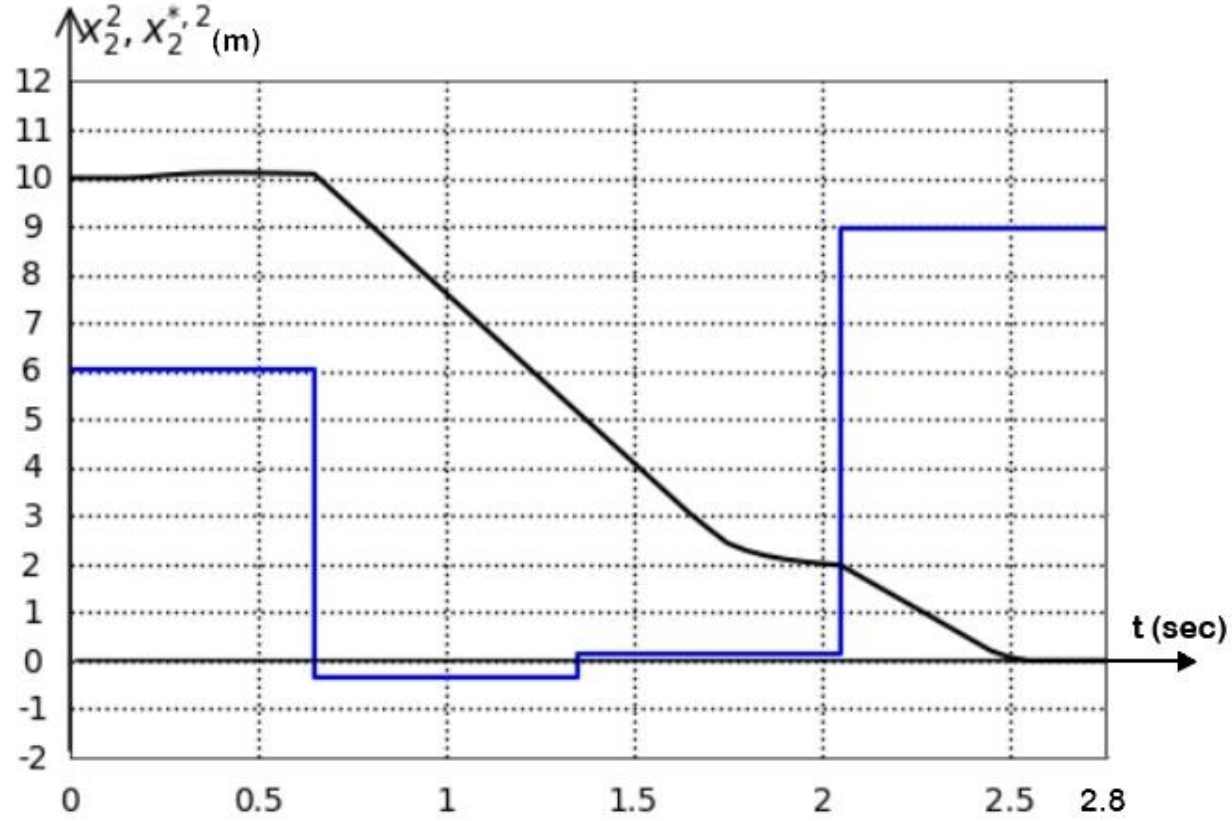

Figure 3.76. The variable x_2^2 in black and effective control $x_2^{*,2}$ in blue

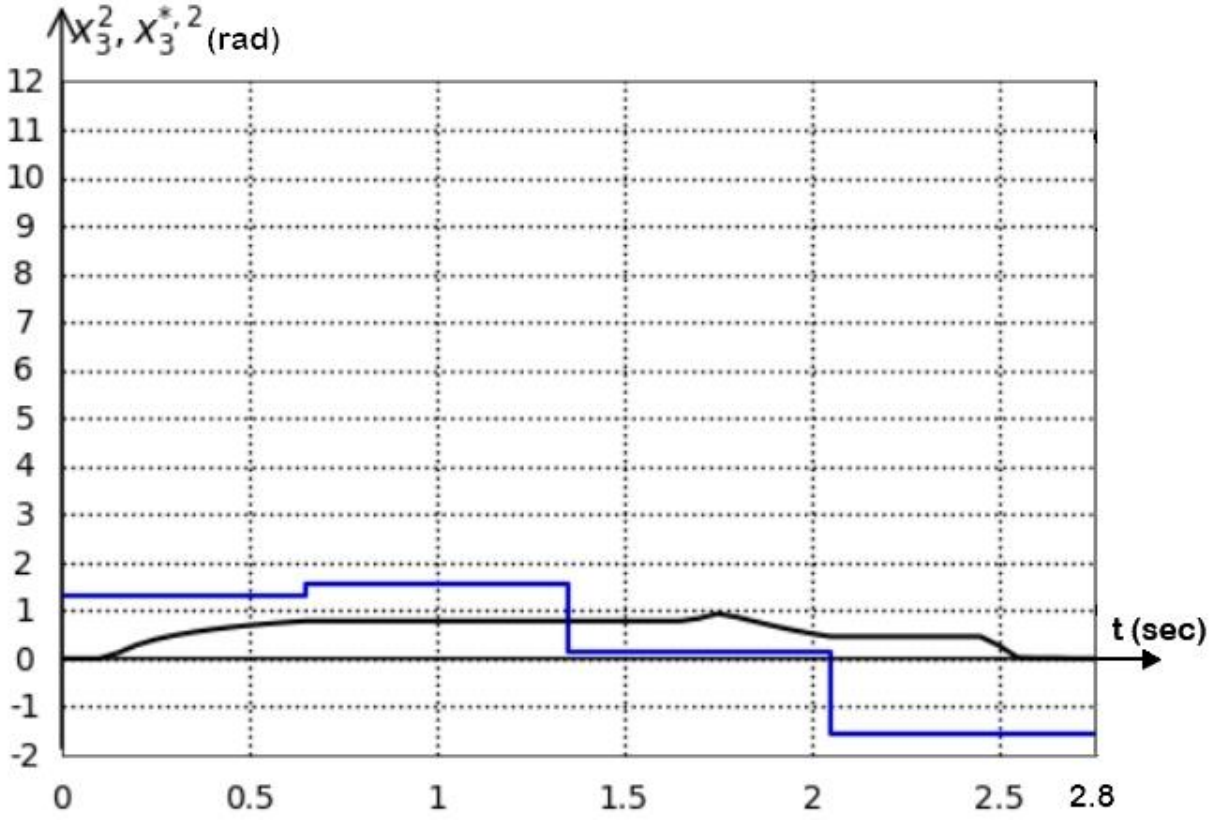

Figure 3.77. The variable x_3^2 in black and effective control $x_3^{*,2}$ in blue

Case 4: The particle swarm optimization (PSO) algorithm is employed for the purpose of finding the stabilization points and it is possible to search the control function (the equilibrium points) as a piecewise constant function. The constraints pertaining to the elements of points are as follows:

$$\begin{aligned} -2 \leq x_1^{j,*,i} \leq 12, \\ -2 \leq x_2^{j,*,i} \leq 12, \\ -\frac{\pi}{2} \leq x_3^{j,*,i} \leq \frac{\pi}{2}. \end{aligned} \tag{3.61}$$

where $c_1 = 2$, $c_2 = 4$, $c_3 = 2.5$, $P_t = 5, r_1 = 2, r_2 = 2, r_3 = 2, r_4 = 2, r_5 = 2, x_{1,1} = 0, x_{1,2} = 5, x_{1,3} = 10, x_{1,4} = 5, x_{1,5} = 5, x_{2,1} = 5, x_{2,2} = 5, x_{2,3} = 5, x_{2,4} = 0, x_{2,5} = 10,$ $\varepsilon = 0.01$ and $t^+ = 2.7$ sec.

The three points for each mobile robot have the subsequent coordinates in the state space $\{x_1, x_2, x_3\}$:

$$\mathbf{x}^{1,*,1} = [-0.0372 \quad 3.2964 \quad 1.4551]^T,$$

$$\mathbf{x}^{1,*,2} = [5.304 \quad 8.5296 \quad 0.6911]^T,$$

$$\mathbf{x}^{1,*,3} = [10.2478 \quad 5.3404 \quad -0.7197]^T,$$

$$\mathbf{x}^{2,*,1} = [7.4218 \quad 10.7499 \quad 0.5217]^T,$$

$$\mathbf{x}^{2,*,2} = [4.3332 \quad 11.0171 \quad 1.3788]^T,$$

$$\mathbf{x}^{2,*,3} = [0.1775 \quad 4.6068 \quad -0.518\ ]^T. \tag{3.62}$$

where the first three points are for the first robot and the other points for the second one.

In Figures (3.78)-(3.84) the findings of the simulation are laid out. Figure 3.78 displays suitable trajectories generated by the pair of mobile robots in the $\{x_1, x_2\}$ plane. In the graphical representation, the red curve traces the motion of the first robot, and the yellow curve the second. Purple circles highlight locations where phase constraints are enforced. Furthermore, the equilibrium configurations for the first and second robots are denoted by small green and black squares, respectively. These markers correspond bijectively to the three equilibrium points per agent that were identified through the solution of the underlying optimal control problem (3.62). As evident from the observation, the two robots have successfully attained their terminal states without violating phase constraints. The functional value in Equation (3.53) was $J_{opt} = 2.7032$.

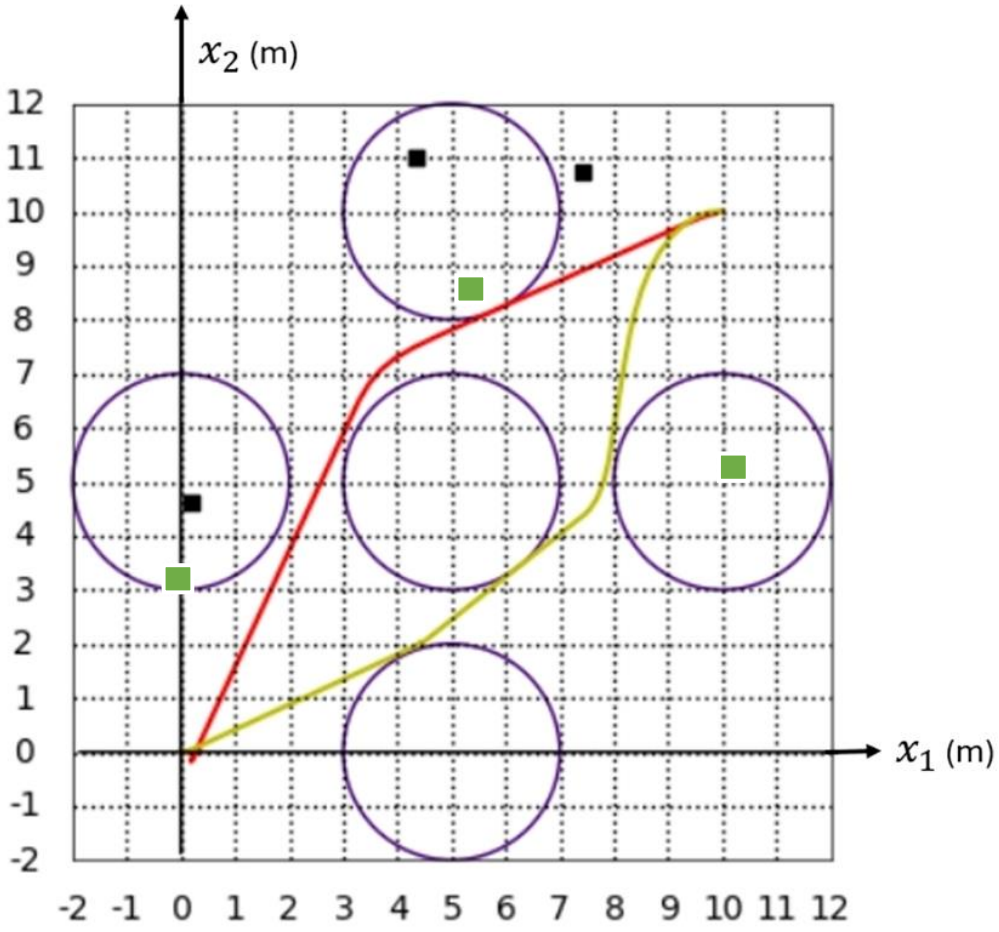

Figure 3.78. Synthesized optimal control trajectories for two robots in the $\{x_1, x_2\}$ plane

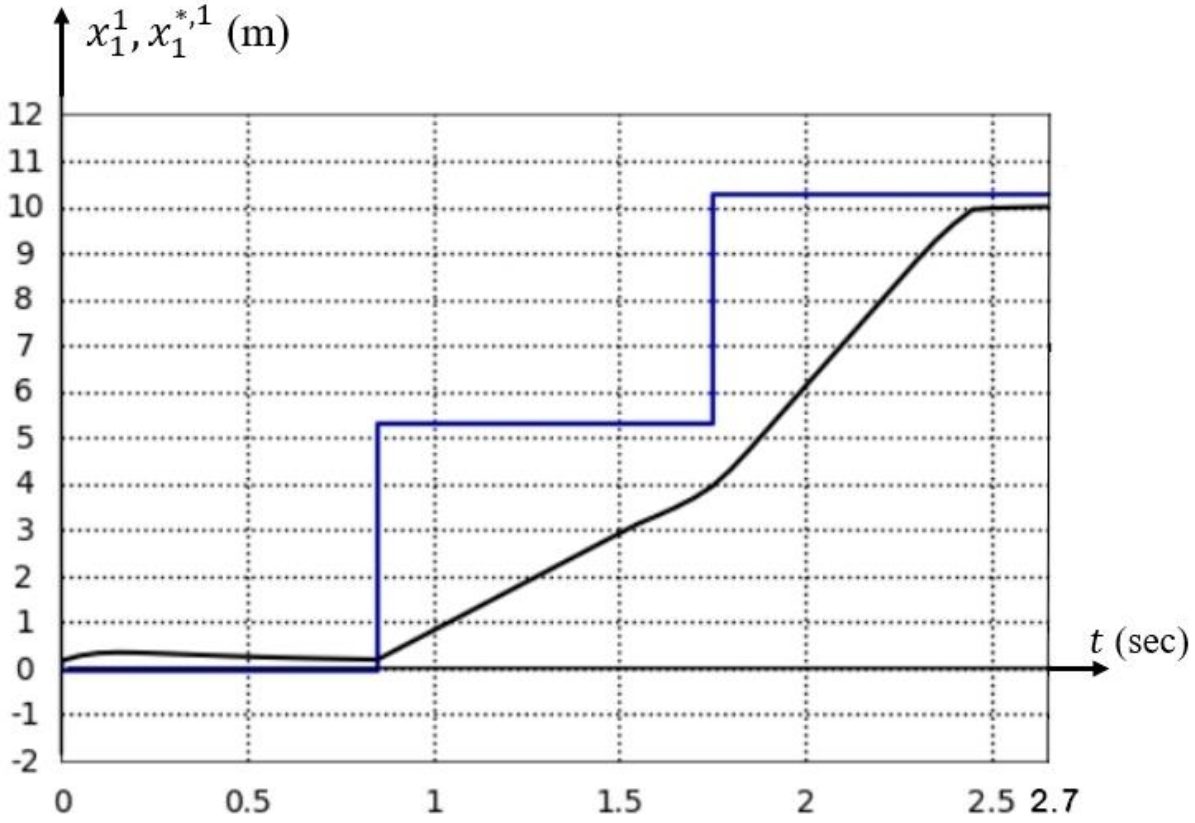

Figure 3.79. The variable x_1^1 in black and effective control $x_1^{*,1}$ in blue

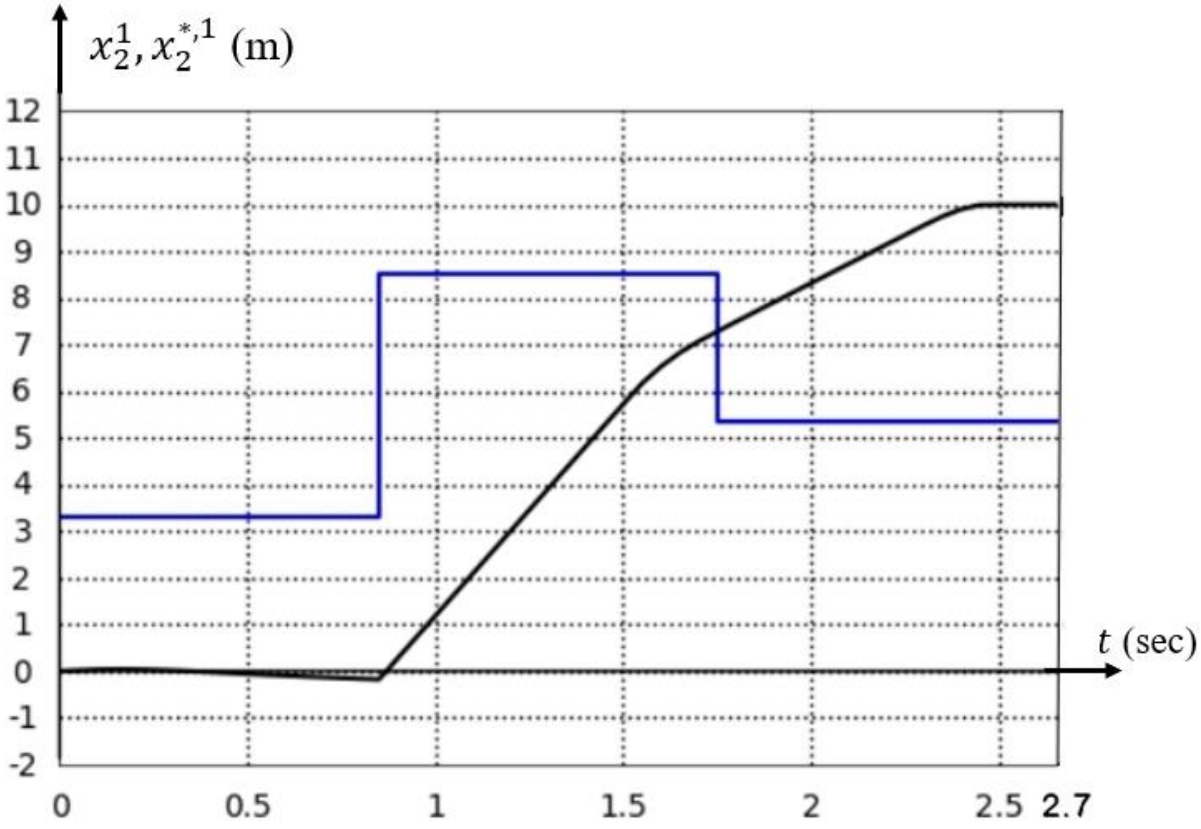

Figure 3.80. The variable x_2^1 in black and effective control $x_2^{*,1}$ in blue

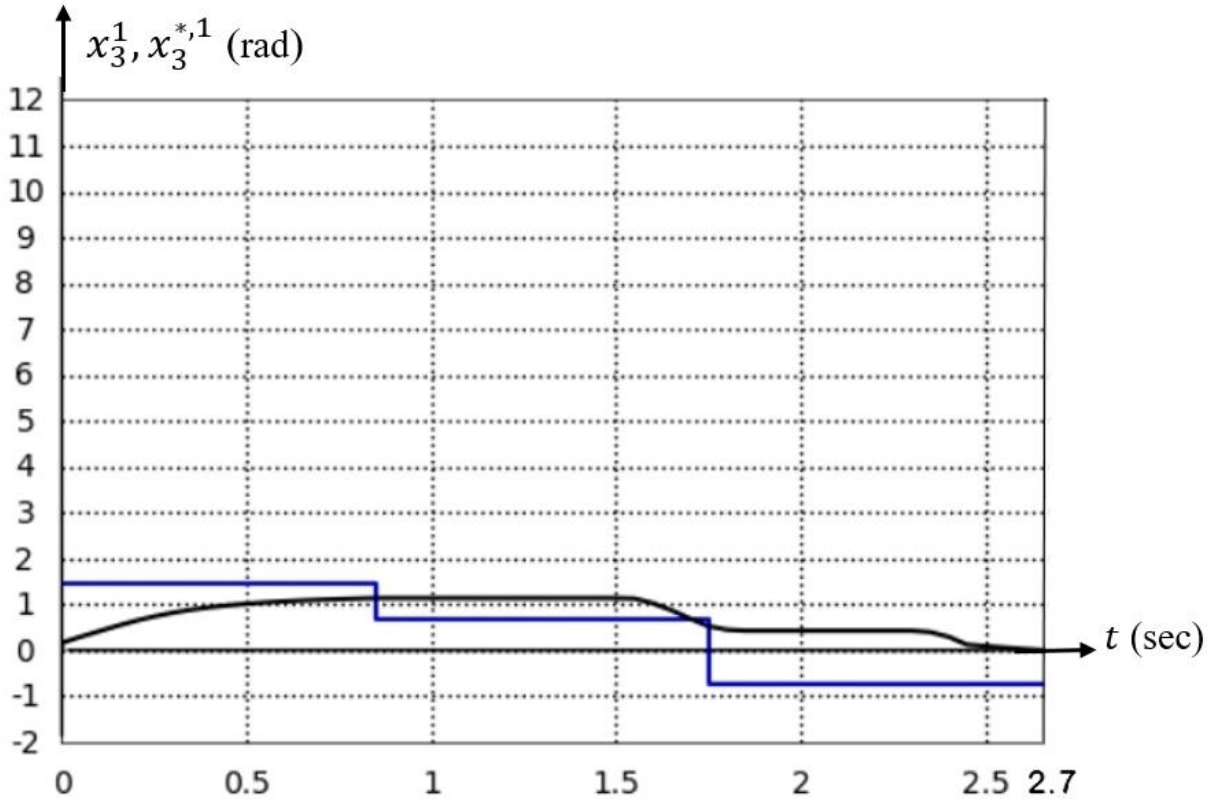

Figure 3.81. The variable x_3^1 in black and effective control $x_3^{*,1}$ in blue

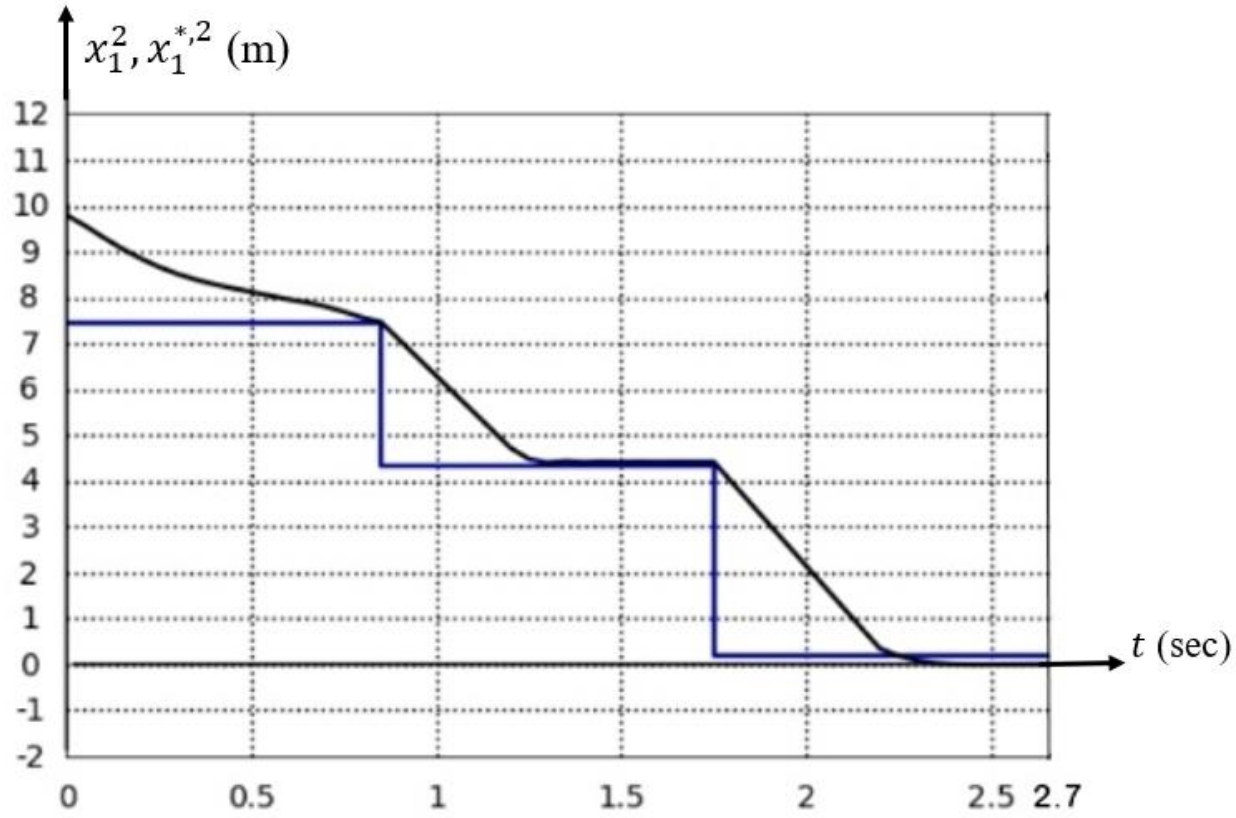

Figure 3.82. The variable x_1^2 in black and effective control $x_1^{*,2}$ in blue

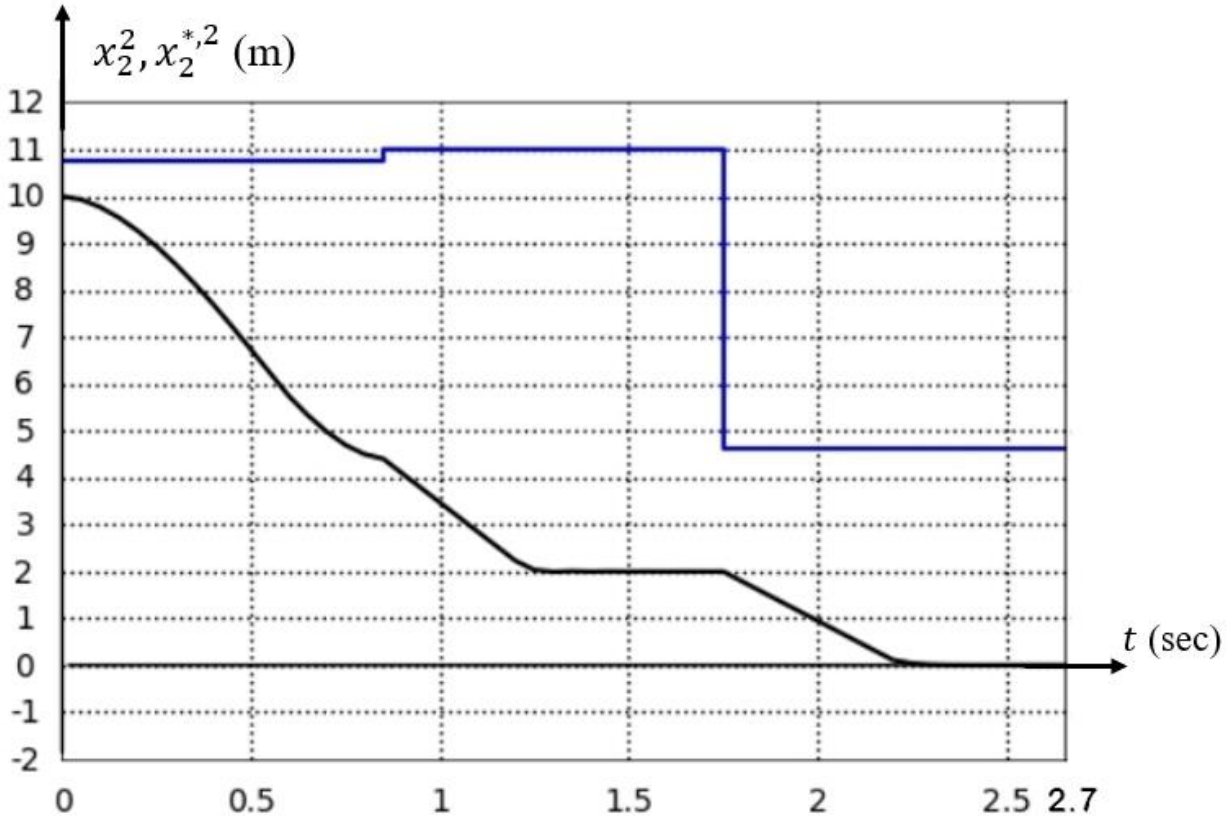

Figure 3.83. The variable x_2^2 in black and effective control $x_2^{*,2}$ in blue

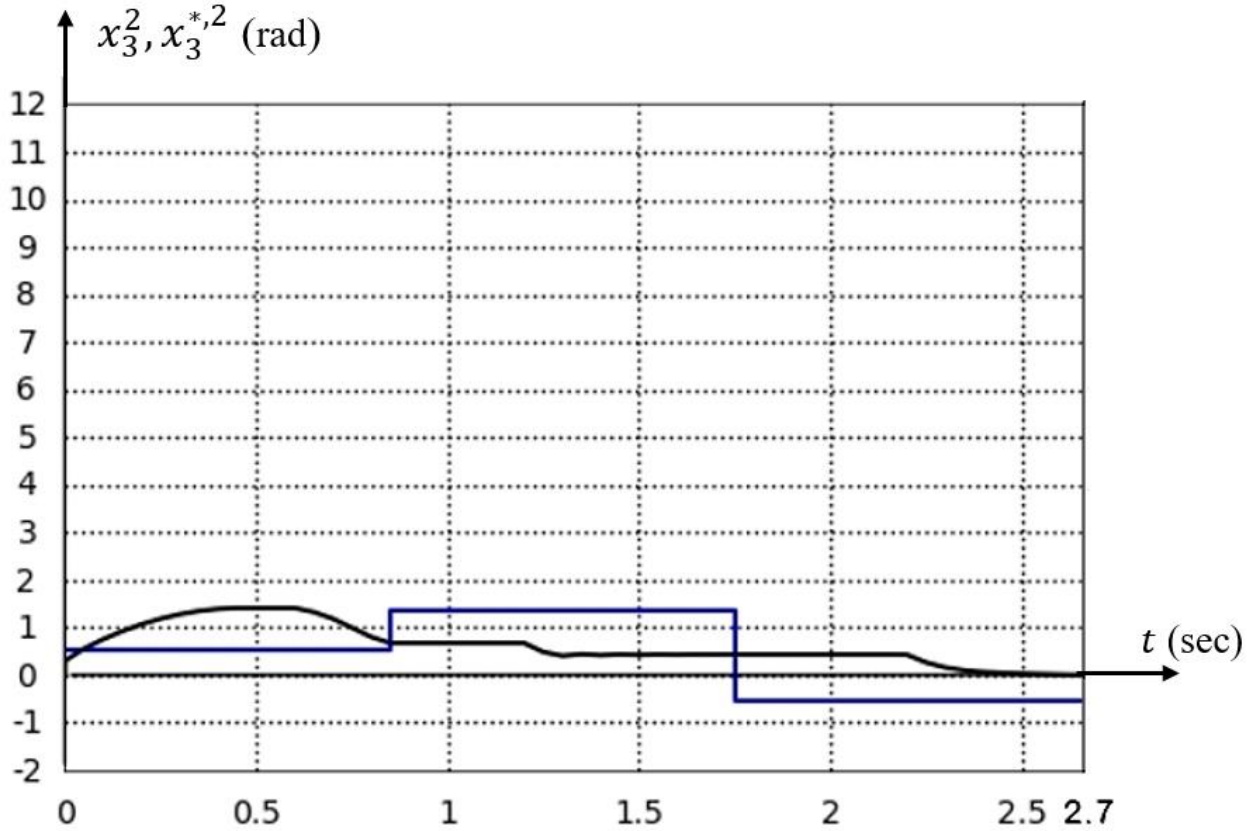

Figure 3.84. The variable x_3^2 in black and effective control $x_3^{*,2}$ in blue

The coordinates of a stable point of equilibrium are determined by the effective control $x_i^{*,j}(t), j = 1, 2,\ i = 1, 2, 3$, in every second of time, based on the construction of a control system for objects. As seen from the findings, the effective control exerts an attractive influence on the relevant state-space component, without requiring kinematic matching with that component. It is widely recognized that the speed of state evolution is reduced in the immediate neighborhood of an equilibrium point compared to distant regions. Thus, for enhanced mobility, the control object should be maintained in the vicinity of that point without settling into it, allowing for continuous and faster motion.

3.3. Summary

It can be summarized this chapter as follows:

1. It has provided an introduction to the problem of optimal control concerning two nonholonomic mobile robots.
2. The environment in question encompassed several static phase constraints, as well as dynamic phase constraints arising from the collision between these two robots.
3. The topic of synthesis of a stabilization system, which could have effectively been addressed using a single robot, was tackled using the variational synthesized genetic programming (VSGP) technique.
4. The solution to the problem of control synthesis yielded the emergence of a stable point of equilibrium in the space of states.
5. The particle swarm optimization (PSO) algorithm was employed to find the previously mentioned points, resulting in the discovery of three points for every single mobile robot.

CONCLUSION

Conclusion and Discussion

This dissertation proposed a synthesized optimal control technique for a pair of nonholonomic wheeled mobile robots operating in a complicated environment, which includes both static and dynamic phase constraints. The study employed a synthesized optimal control approach, which included a further step of synthesis of a stabilization feedback control system. This control system aimed to achieve a steady state for the robot with respect to a specific point in the space of states. The proposed approach incorporates tunable stabilization points as decision variables in the optimization process. Their effective coordinates are determined such that the resulting closed-loop trajectory satisfies initial and terminal boundary conditions, adheres to obstacle-avoidance constraints, and minimizes a researcher-defined quality criterion. The proposed methodology presented a novel strategy to address a widely known problem in optimal control. However, it additionally introduced a novel problem statement in the field of optimal control, subsequently facilitating its numerical solution. The results have shown that employing this methodology enabled the computer to generate innovative and remarkable solutions, surpassing the expectations of engineers in certain instances.

The problem of synthesized optimal control has been solved by a two-step process, namely the stabilization step and optimization step. The stabilization step represented the first step. The primary challenge encountered in addressing the mentioned synthesized optimal control problem was mostly associated with the first step. Solving the problem of control synthesis has consistently posed a more intricate challenge compared to the problem of optimal control. The synthesis problem has been solved via the utilization of controllers in feedback, wherein control is sought as a function involving the robot's state. However, this approach necessitated an accurate model of the controlled robot. It is essential to acknowledge that solving the problem of control synthesis in the first step has brought about substantial modifications to the control robot mathematical model. A more generalized technique has relied on the utilization of symbolic regression, a computer technique known as variational synthesized genetic programming (VSGP), to address the synthesis problem. The control synthesis problem has been solved with the objective of guaranteeing the control robot's stability with respect to a specific point inside the state space. The present step of the stabilization system synthesis has facilitated the incorporation of control within the robot, ensuring that the differential equations system possesses the essential attribute of feasibility. The implementation of this form of control in actual systems was well accepted due to its ability to minimize model errors through the utilization of feedback control. This methodology belongs to the broader family of machine learning algorithms; however, it transcends the limitations of neural networks by enabling the search over both the space of possible functional architectures and their

parameter values of the control function—thereby supporting interpretable, equation-based modeling. The VSGP implements an evolutionary framework that evolves the structural-parametric search of candidate control functions, evaluating their performance solely through the quality functional's output. The VSGP technique has been utilized to obtain a solution without relying on explicit model equations. The first step yielded the acquisition of the control function's structure and parameters. Consequently, the researcher has automatically obtained the efficient controller function structure and its proper parameters. The first step of synthesis of the stabilization system was a crucial concept within this methodology, leading to improved results in tasks involving intricate environments. At present, the problem of general synthesis can only be effectively solved by employing symbolic regression-based machine learning techniques that offer approximate solutions.

The optimization step was the second step in this proposed approach. Following the previous step, which guaranteed a steady system movement to a stabilization point, a series of stabilization points were meticulously sought to transition among them at specified times sequentially. This strategic approach enabled the robots to ultimately attain the terminal state, besides the quality criterion improved estimation. During this step, the optimal control problem was addressed by utilizing the robot's stability points' coordinates as control. In order to ensure the existence of adjacent areas with attractive properties for the effective solution, it was necessary to carefully select the stability points' position within the state space. This positioning was done in such a way that specific solutions originating from a specific area of initial states, which are attracted to such stability points, would exhibit nearness to each other as they progress towards the terminal state. The equilibrium point exhibited attractor features in an algorithmic manner, as it was seen that all solutions converged in close vicinity to this point, so satisfying the principle of feasibility. This methodology implemented a control mechanism for the robot by transitioning among stable equilibrium points. However, it is essential to note that these equilibrium points were not coincident with the reference trajectory. The positions of such points were determined by the utilization of an evolutionary algorithm known as Particle Swarm Optimization (PSO), which was applied based on the criterion of the problem of optimal control. It is essential to observe that at the stable point of equilibrium within the state space, the velocity of the robot was equal to zero. Consequently, the placement of stable points over the reference trajectory resulted in ineffective mobility characterized by stops at such points. The points have the potential to be located at any position inside the state space. By strategically switching these points, the robot could accomplish an efficient movement on the reference trajectory without any stops. The computer memory was set up with the found stabilization points' coordinates and a designated time interval for transitioning among these points, thus establishing the suitable trajectory. The proposed methodology introduced a novel control strategy that involved altering the position of a stable point of equilibrium. This approach compelled the

robot's stabilization system to drive it towards the equilibrium point. By altering the position of the equilibrium point over time, it became possible to guide the robot to its intended terminal state while improving the quality criterion. In the second step of the applied technique for synthesized optimal control, we conducted a search for the positions of the points of equilibrium using a piece-wise constant function.

This technique possesses numerous advantages. One great thing about this technique was that it did not depend on a specific model of the control object. This meant that the symbolic regression technique could be used to search the feedback function of control automatically. The primary advantage of this technique consisted of its versatility and capacity to be applied to diverse, dynamic models of control objects. One additional benefit resulted from the establishment of systems of optimal control that possess the property of feasibility. This characteristic emerges as a result of the control object stabilization during the first step. This system of stabilization has facilitated the establishment of an equilibrium point for the robot inside the space of states. This implies that the system was designed for its attraction to a specific equilibrium point. Another benefit of this technique was the implementation of control through the alteration of equilibrium points. The ability to achieve optimal control over an object has been made possible such that the control parameters' effective values could be rapidly computed employing numerical optimization techniques; moreover, it has been possible to update these parameters in real-time, even on board. Interestingly, it could be noted that all techniques employed for the purpose of calculation were automated numerical techniques, obviating the need for manual calculations. This pivotal aspect facilitated the automation and universalization of the control system development process. One of several primary characteristics of the synthesized technique was the hands-on feasibility of obtaining numerical solutions for the problem of optimal control in intricate systems. One objective of the process reformulation for the known problem represented that its solution was able to be directly applied to a real object. The problem of refined optimal control incorporated one extra requirement for the suitable trajectory, namely that this trajectory possessed an attractive close vicinity. In order to achieve this objective, it is necessary for a control function to be dependent not just on time but also on the vector of state space.

In summary, the methodology of synthesized optimal control presented in this study was a novel approach to solving optimal control problems by focusing on controlling a stable robot's equilibrium point. The methodology consisted of two different steps. In the initial design phase, a stabilization system was embedded within the control architecture of the robotic system, thereby inducing a structurally stable equilibrium point in its phase space. This was motivated by the established principle that such an equilibrium is a necessary condition for ensuring desirable control properties in the robot's mathematical model. Secondly, Although the equilibrium point could be reconfigured over time, the system remained

stable at all times because of the underlying stabilization system, which allowed for control through manipulating the position of the equilibrium point. This technique possesses the ability to be universal, enabling a numerical solution of the synthesis problem within a broad context, devoid of the necessity to construct a training set. Instead, it relies just on the evaluation of the quality criterion, so exemplifying the utilization of unsupervised machine learning.

Suggested Future Works

The subsequent recommendations are proposed for future works:

1. A two-stage methodology is proposed for solving the optimal control problem: (i) numerical solution of the optimal control problem over a set of initial conditions to generate a collection of optimal trajectories; (ii) application of symbolic regression to approximate the resulting trajectories with an interpretable expression. In this context, supervised machine learning is employed rather than unsupervised machine learning.
2. One possible way to execute the proposed synthesized optimal control technique is to employ a holonomic mobile robot rather than a nonholonomic one.
3. The suggested technique can potentially be applied in various forms of motion control for mobile robots, such as trajectory tracking, as an alternative to the current approach of altering the stable point of equilibrium.
4. The proposed technique can be employed to address the optimal control problem and evaluate its efficacy in the existence of uncertainties, which may arise due to considerations such as model inaccuracies, noise, initial conditions uncertainty, and other similar sources.
5. It is essential to persist in the exploration of other evolutionary algorithms, such as the Grey Wolf Optimization Algorithm (GWO) or hybrid algorithms, such as (GA and PSO or GA and GWO), to solve the problem of optimal control rather than relying solely on the Particle Swarm Optimization Algorithm (PSO), as mentioned in this dissertation.

LIST OF ABBREVIATIONS

Abbreviation	Definition
R.U.R	Rossum's Universal Robots
MLC	Machine learning control
WMR	Wheeled mobile robot
SGP	Synthesized genetic programming
VSGP	Variational synthesized genetic programming
PSO	Particle Swarm Optimization Algorithm
WMRs	Wheeled mobile robots
ICR	Instantaneous center of rotation
ICC	Instantaneous Center of Curvature
DDWMR	Differential drive wheeled mobile robot
DOF	Degrees of freedom
DDOF	Differential degrees of freedom
DDWMRs	Differential drive wheeled mobile robots
PID	Proportional Integral Derivative
ML	Machine learning
NN	Neural Network
MWMR	Mecanum-wheel mobile robot
FL	Fuzzy Logic
OMRs	Omnidirectional mobile robots
RL	Reinforcement Learning
SR	Symbolic Regression
SMC	Sliding Mode Control
MPC	Model Predictive Control
MIMO	Multiple-Input Multiple-Output System
NMPC	Nonlinear Model Predictive Control
GA	Genetic Algorithm
GAs	Genetic Algorithms
VarGA	Variational Genetic Algorithm
GP	Genetic Programming
CGP	Cartesian Genetic Programming
GWO	Grey Wolf Optimization Algorithm

LIST OF SYMBOLS

Symbol	Definition
$\alpha, \gamma, \beta, \sigma$	The PSO algorithm constant parameters.
β_i	The inner wheel's steering angle in the Ackerman steering mechanism.
β_o	The outer wheel's steering angle in the Ackerman steering mechanism.
β_s	The automobile's true steering angle in the Ackerman steering mechanism.
δ	A positive value of tiny magnitude.
Δ	A provided time interval.
ε and t^+	positive numerical values.
ε_1 and t_1^+	Provided positive numerical values.
ζ	The posture vector of differential drive wheeled mobile robot (DDWMR).
η	A function that equals or approximated to ψ based on a specific criterion.
θ	the angle of orientation of the mass center coordinate system of the DDWMR CX_CY_C relative to the inertial coordinate system OX_OY_O.
θ_d	The destination orientation of the robot within the navigation plane.
$\vartheta(A)$	The Heaviside step function.
μ_1 , μ_2	The random mutation points for SGP technique.
$\mu(\boldsymbol{x})$	The Bellman function.
$\boldsymbol{\xi}$	A random value drawn from the interval [0:1].
ϱ	An evaluation criterion.
$\dot{\varphi}_l, \dot{\varphi}_r$	The angular velocities of the left and right wheels.
ψ	The angle of the roller in the Swedish wheel.
ψ	The unknown function.
$\Psi(\boldsymbol{b})$	The function that transforms a non-numerical structure's code into an actual function.
ω, υ	The angular and linear velocities of differential drive wheeled mobile robot.
$\boldsymbol{\Omega}$	A correlation between the angular velocities of the right and left wheels (φ_r, φ_l) of the DDWMR, and the angular and linear velocities of the mass center of the DDWMR (ω, υ).
$\boldsymbol{A}(\boldsymbol{Q})$	The matrix that encompasses nonholonomic constraints.
a	The distance from the center of wheel to the center of automobile in the Ackerman steering mechanism.
2a	The distance between the actuated wheels and the axis of symmetry.

$\boldsymbol{a}^T(\boldsymbol{Q})$	The parameter vector of the constraint.
b	The length of rolling polygon side.
$\boldsymbol{b}^0$	The basic solution.
$\boldsymbol{b}^i$	The new basic solution.
C	Center of mass or point of guidance.
CX_CY_C	The mass center coordinate system of differential drive wheeled mobile robot.
c_1, c_2, c_3	The weight coefficients for the quality criterion of the synthesized optimal control for mobile robot example.
c	Randomly chosen possible solutions.
D	The total count of variation vectors that presents in a single set.
d	The distance between point P and point C.
dep	The dimension of the variation vector.
E	The distance from the instantaneous center of rotation (ICR) to the nearest wheel in the Ackerman steering mechanism.
$e_1, \dots, e_v$	The unit elements for two-argument functions.
$\boldsymbol{F}$	A unified set of the two sets of fundamental functions
$\boldsymbol{F_0}$	The arguments set.
$\boldsymbol{F_1}$	The functions set that is characterized by one argument.
$\boldsymbol{F_2}$	The functions set that is characterized by two arguments.
$F(\mathbf{q})$	The objective function of the optimization problem in PSO algorithm.
F_i	The objective function value for genetic algorithm.
F_{j-}	The best objective function value for genetic algorithm.
Fr	The generated force by the rotational motion of the omnidirectional wheel.
Fr_1	A parallel force of *Fr*, which parallels the axis of the roller.
Fr_2	A perpendicular force of *Fr*, which is oriented at a right angle to the axis of the roller.
F_γ	The objective function value for the set of variations vector $\boldsymbol{W_\gamma}$.
F_φ	The objective function value for the set of variations vector $\boldsymbol{W_\varphi}$.
$f(\boldsymbol{Q})$	The holonomic constraint.
$f(\boldsymbol{Q}, \dot{\boldsymbol{Q}})$	The nonholonomic constraint.
f^*	A value that meets the estimate requirements.
$\boldsymbol{G}$	The evaluation of an objective function or a function of fitness for SGP technique.
G_m	The mobility degree of wheeled mobile robots (WMRs).

G_s	The steerability degree of wheeled mobile robots (WMRs).
$\boldsymbol{g}(\boldsymbol{x})$	The control function in terms of the vector of state space.
$\boldsymbol{g}(\boldsymbol{x}^* - \boldsymbol{x})$	The control function that obtained from the stabilization step.
$\boldsymbol{g}_{\boldsymbol{\gamma}}$, $\boldsymbol{g}_{\boldsymbol{\varphi}}$	The objective functions values of random two possible solutions of SGP technique.
H	The number of columns (vectors) in the code matrix.
$\boldsymbol{h}(t)$	The control function that is obtained is commonly referred to as a program control.
i	General serial counter.
J	The general quality criterion of the optimal control.
J_1	The general quality criterion of the control synthesis system with a domain of initial condition.
J_2	The general quality criterion of the control synthesis system with a limited set of initial conditions.
$J(\boldsymbol{R}_i, \mathbf{q}^i)$	The objective function for SGP technique.
J_{opt}	The quality criterion of the synthesized optimal control for mobile robot example.
J_s	The general quality criterion of the control synthesis system with a limited set of initial conditions for symbolic regression techniques.
J_{s1}	The general quality criterion of system with a limited set of initial conditions in the stabilization step.
J_{so1}	The general quality criterion of the synthesized optimal control.
J_{syn}	The general quality criterion of system with a limited set of initial conditions in the stabilization step for mobile robot example.
j	General serial counter.
K	Number of intervals or number of points of equilibrium.
k_1 , k_2	The random crossover points for SGP technique.
k_c	The random crossover point for genetic algorithm.
L	The number of initial conditions.
l	The sequence of possible solutions within the initial population.
M	The total number of the set of codes that representing the possible solutions.
M_w	The maneuverability of wheeled mobile robots (WMRs).
m_q	The dimensionality of the parameters vector.
m	The dimension of the velocity (control) vector.
N_c	The number of separate (independent) constraints for wheeled mobile robot.
n	The dimension of the state space.

OX_OY_O	The inertial coordinate system of differential drive wheeled mobile robot.
P	Intersection of the axis of the symmetry with the wheels' axis.
P_c	The crossover probability for SGP technique.
P_{r_c}	The crossover probability.
P_t	The total number of phase constraints.
P_μ	The mutation probability for SGP technique.
p	The number of nonholonomic constraints.
p_1	A weight coefficient for the quality criterion of the control synthesis system.
$\boldsymbol{Q}$	The generalized coordinate vector.
$\dot{\boldsymbol{Q}}$	The vector containing the velocities of the system within the generalized coordinates.
$\mathbf{q}$	The parameters vector.
$q_1, \ldots, q_{m_q}$	The parameters of the mathematical expression.
q_i^+ and q_i^-	The higher and lower bounds of the parameters.
$q^{j(c)}$	The most efficient possible solution in PSO algorithm.
$\boldsymbol{R}$	The code matrix.
$\boldsymbol{R}_{i^-}$ or $\boldsymbol{g}_{\boldsymbol{i}^-}$	The best solution for the code of SGP technique.
$\boldsymbol{R}_{CGP}$	The code matrix of Cartesian genetic programming technique.
$\boldsymbol{R}_{GP}$	The code matrix of genetic programming technique.
$\boldsymbol{R}_{SGP}$	The code matrix of synthesized genetic programming technique (SGP).
$\boldsymbol{R}_\gamma, \mathbf{q}^\gamma,$ $\boldsymbol{R}_\varphi, \mathbf{q}^\varphi$	Random two possible solutions of SGP technique for the crossover operation.
R_{ai}	The radius of an instantaneous circular path for wheeled mobile robot.
$\mathbb{R}^n$	The state space.
$\mathbb{R}^m$	The control space.
$\boldsymbol{r}$	The column in the code matrix such that it can be considered the column as a vector.
r_A	Radius of left or right wheel.
r_d	The minimal acceptable secure distance between robots.
r_e	An element in the column of the code matrix.
$r_{st}, x_{1,st},$ $x_{2,st}$	Radius and coordinates of center of the constraints of static phase.
$\boldsymbol{S}(\boldsymbol{Q})$	The Jacobian matrix.
S	A set of codes that representing the possible solutions.

t_f	The time at which the terminal condition is reached, starting from the initial one in the optimal control.
t_k	A given time.
t^*	The general time at which the terminal condition is reached, starting from the initial one in the stabilization step.
t_i^*	The time at which the terminal condition is reached, starting from the initial one in the stabilization step and used in the general quality criterion J_{s1}.
$\boldsymbol{U}$	A compact set.
$\boldsymbol{u}$	The vector representing the control, $\boldsymbol{u} \in \boldsymbol{U}$.
$\tilde{u}_1, \tilde{u}_2$	The effective control functions.
V	The distance from the center of front wheel to the center of rear wheel in the Ackerman steering mechanism.
$\boldsymbol{v}$	The auxiliary velocity vector.
v_h	A hub velocity in the roller of the omnidirectional wheel.
v_l	The velocity of the left wheel in a differential drive robot.
v_r	The velocity of the right wheel in a differential drive robot.
v_t	The sum of the horizontal velocity (v_h) and the vertical velocity (v_v).
v_v	A small rotational velocity in the roller of the omnidirectional wheel.
v_i^j	A history vector in PSO algorithm.
$\boldsymbol{W}^i$	The ordered multiset consisting of variation vectors as the initial population.
$\boldsymbol{W_\gamma}, \boldsymbol{W_\varphi}$	Two sets of variations vectors.
$\boldsymbol{W_{\gamma+1}}$, $\boldsymbol{W_{\varphi+1}}$	The new sets of variations' vectors generated from the crossover.
$\mathcal{W}$	The vector of small variations.
w_1	An index denoting a small variation.
w_2 , w_{dep-1}	Indices indicating the element position in the code that define the variable element.
w_{dep}	The updated value of the defined element.
$\mathbf{X}$	The input space.
$\mathbf{X}_0$	The initial conditions domain within the state space.
$\widetilde{\mathbf{X}}, \widetilde{\mathbf{Y}}$	The training sets.
$\boldsymbol{x}$	The input vector (state space vector).
$\boldsymbol{x}^0$	The initial conditions of the control object model.
$\dot{\boldsymbol{x}}$	The mathematical model of the control object in the form of an ordinary differential e

	system.
$\boldsymbol{x}^f$	The terminal conditions of the control object model.
$\boldsymbol{x}(t, \boldsymbol{x}^0)$	A partial solution of the control object system.
$\boldsymbol{x}(t^*)$	The terminal position, enabling the system to achieve stabilization at such a point.
$\tilde{\boldsymbol{x}}(\boldsymbol{x}^*(t_k))$	A stable point of equilibrium.
$\boldsymbol{x}^*(t)$	A time control function.
x , y	The coordinates of point C.
$x_1, \ldots, x_r$	The variables of the mathematical expression.
x_i, y_i	The initial position of the robot within the navigation plane.
x_d, y_d	The destination position of the robot within the navigation plane.
$x_d(t), y_d(t),$ $\theta_d(t)$	The destination position of the robot within the navigation plane throughout time.
$\dot{x}_d(t), \dot{y}_d(t),$ $\dot{\theta}_d(t)$	The destination velocity of the robot within the navigation plane throughout time.
$x_i^{*,j}(t)$	The coordinates of a stable point of equilibrium.
Y	The output space.
$\boldsymbol{y}$	The output vector.

REFERENCES

[1] G. Klancar, A. Zdesar, S. Blazic and I. Skrjanc, *Wheeled Mobile Robotics*. Butterworth-Heinemann, 2017.

[2] G. Cook and F. Zhang, *Mobile Robots*. John Wiley & Sons, 2020, doi:10.1002/9781119534839.

[3] M. Javaid, A. Haleem, R. P. Singh and R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," *Cognitive Robotics*, vol. 1, pp. 58–75, 2021, doi: 10.1016/j.cogr.2021.06.001.

[4] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa and J. O. Strandhagen, "Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of Operations Research*, vol. 308, no. 1–2, pp. 125–143, Feb. 2020, doi: 10.1007/s10479-020-03526-7.

[5] F. D'Souza, J. Costa and J. N. Pires, "Development of a solution for adding a collaborative robot to an industrial AGV," *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 5, pp. 723–735, May 2020, doi: 10.1108/ir-01-2020-0004.

[6] J. Holland *et al.*, "Service Robots in the Healthcare Sector," *Robotics*, vol. 10, no. 1, p. 47, Mar. 2021, doi: 10.3390/robotics10010047.

[7] M. Stasevych and V. Zvarych, "Innovative Robotic Technologies and Artificial Intelligence in Pharmacy and Medicine: Paving the Way for the Future of Health Care—A Review," *Big Data and Cognitive Computing*, vol. 7, no. 3, p. 147, Aug. 2023, doi: 10.3390/bdcc7030147.

[8] H. Najim, I. Kareem and W. Abdul-Lateef, "Design and implementation of an omnidirectional mobile robot for medicine delivery in hospitals during the covid-19 epidemic," *AIP Conference Proceedings*, vol. 2380, no. 1, 2023, doi:10.1063/5.0156862

[9] S. Jameel Al-Kamil and R. Szabolcsi, "Optimizing path planning in mobile robot systems using motion capture technology," *Results in Engineering*, p. 102043, Mar. 2024, doi: 10.1016/j.rineng.2024.102043.

[10] N. Sharma, J. K. Pandey and S. Mondal, "A Review of Mobile Robots: Applications and Future Prospect," *International Journal of Precision Engineering and Manufacturing*, vol. 24, no. 9, pp. 1695–1706, Aug. 2023, doi: 10.1007/s12541-023-00876-7.

[11] M. Z. U. Rahman, U. Raza, M. A. Akbar, M. T. Riaz, A. H. Gumaei and N. Ahmad, "Radio-Controlled Intelligent UGV as a Spy Robot with Laser Targeting for Military Purposes," *Axioms*, vol. 12, no. 2, p. 176, Feb. 2023, doi: 10.3390/axioms12020176.

[12] K. Bazargani and T. Deemyad, "Automation's Impact on Agriculture: Opportunities, Challenges, and Economic Effects," *Robotics*, vol. 13, no. 2, p. 33, 2024, doi: 10.3390/robotics13020033.

[13] C. Cheng, J. Fu, H. Su and L. Ren, “Recent Advancements in Agriculture Robots: Benefits and Challenges,” *Machines*, vol. 11, no. 1, p. 48, 2023, doi: 10.3390/machines11010048.

[14] D. Xie, L. Chen, L. Liu, L. Chen and H. Wang, “Actuators and Sensors for Application in Agricultural Robots: A Review,” *Machines*, vol. 10, no. 10, p. 913, Oct. 2022, doi: 10.3390/machines10100913.

[15] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*. John Wiley & Sons, 2020.

[16] M. Mihelj *et al*., “Mobile Robots,” *Robotics*, pp. 189–208, Jul. 2018, doi: 10.1007/978-3-319-72911-4_13.

[17] N. J. Nilsson and A. M. Automaton, “An Application of Artificial Intelligence Techniques,” In *Proc. of IJCAI*, vol. 509. 1969, doi: 10.21236/ADA459660.

[18] A. M. Thompson, *The navigation system of the JPL robot*. No. NASA-CR-154123. 1977.

[19] G. Giralt, R. Sobek and R. Chatila, “A multi-level planning and navigation system for a mobile robot: a first approach to Hilare,” In *Proceedings of the 6th international joint conference on Artificial intelligence*, vol. 1, pp. 335-337. 1979.

[20] L. Jean-Paul, “Feasible trajectories for mobile robots with kinematic and environment constraints,” *Proceeding International Conference Intelligent Autonomous Systems*, pp. 346-354, 1986.

[21] Z. Li and J. F. Canny, *Nonholonomic Motion Planning*. Springer Science & Business Media, 2012.

[22] M. Yue, C. An and Z. Li, “Constrained Adaptive Robust Trajectory Tracking for WIP Vehicles Using Model Predictive Control and Extended State Observer,” in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 733-742, May 2018, doi: 10.1109/TSMC.2016.2621181.

[23] C. Shen, Y. Shi and B. Buckham, “Nonlinear model predictive control for trajectory tracking of an AUV: A distributed implementation,” *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec. 2016, doi: 10.1109/cdc.2016.7799190.

[24] Y. -C. Huang and H. -Y. Li, “Receding Horizon Optimal controller for reference trajectory tracking in Mars entry guidance,” *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp. 2442-2449, 2016, doi: 10.1109/CGNCC.2016.7829176.

[25] M. Neunert *et al*., “Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking,” *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1398-1404, 2016, doi: 10.1109/ICRA.2016.7487274.

[26] D. J. Todd, *Walking Machines*. Springer Science & Business Media, 2013.

[27] P. Cizek, M. Zoula and J. Faigl, “Design, Construction, and Rough-Terrain Locomotion Control of Novel Hexapod Walking Robot with Four Degrees of Freedom Per Leg,” *IEEE Access,* vol. 9, pp. 17866–17881, 2021, doi: 10.1109/access.2021.3053492.

[28] A. Mahapatra, S. S. Roy and D. K. Pratihar, “Multi-legged robots—A review,” *Multi-body Dynamic Modeling of Multi-legged Robots*, pp. 11-32, 2020, doi: 10.1007/978-981-15-2953-5_2.

[29] B. Chong *et al*., “Geometry of contact: contact planning for multi-legged robots via spin models duality,” *arXiv preprint arXiv:2302.03019*, 2023, doi: 10.48550/arXiv.2302.03019.

[30] N. Mahkam, T. B. Yılmaz and O. Özcan, “Smooth and Inclined Surface Locomotion and Obstacle Scaling of a C-Legged Miniature Modular Robot, ” *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, pp. 9-14, Apr. 2021, doi: 10.1109/RoboSoft51838.2021.9479218.

[31] G. Rigatos, “A Nonlinear Optimal Control Approach for Tracked Mobile Robots,” *Journal of Systems Science and Complexity*, vol. 34, no. 4, pp. 1279–1300, Feb. 2021, doi: 10.1007/s11424-021-0036-1.

[32] L. Bruzzone, S. E. Nodehi and P. Fanghella, “Tracked Locomotion Systems for Ground Mobile Robots: A Review,” *Machines*, vol. 10, no. 8, p. 648, Aug. 2022, doi: 10.3390/machines10080648.

[33] M. Ahmad, V. Polotski and R. Hurteau, “Path tracking control of tracked vehicles,” *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol.3, pp. 2938-2943, 2000, doi: 10.1109/ROBOT.2000.846474.

[34] T. Bräunl, *Embedded robotics: from mobile robots to autonomous vehicles with Raspberry Pi and Arduino*. Springer Nature, 2022, doi: 10.1007/978-981-16-0804-9.

[35] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, “Approximating Kinematics for Tracked Mobile Robots,” *The International Journal of Robotics Research*, vol. 24, no. 10, pp. 867–878, Oct. 2005, doi: 10.1177/0278364905058239.

[36] P. Corke, *Robotics and control: fundamental algorithms in MATLAB®*. vol. 141. springer Nature, 2021, doi: 10.1007/978-3-030-79179-7.

[37] P. Morin, “Control of Mobile Robots,” in *Encyclopedia of Robotics*, M. H. Ang, O. Khatib, and B. Siciliano, Eds. Berlin, Heidelberg: Springer, 2023, doi: 10.1007/978-3-642-41610-1_60-1.

[38] B. Siciliano *et al*., “Mobile robots,” in *Robotics: Modelling, Planning and Control*, pp. 469-521, 2010, doi: 10.1007/978-1-84628-642-1_11.

[39] D. R. Jones and K. A. Stol, “Modelling and stability control of two-wheeled robots in low-traction environments,” in *Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2010.

[40] R. Beniak and T. Pyka, “Stability analysis of a tri-wheel mobile robot,” *2016 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 1094-1097, 2016, doi: 10.1109/MMAR.2016.7575290.

[41] D. Cui, X. Gao, W. Guo and H. Dong, “Design and Stability Analysis of a Wheel-Track Robot,” *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pp. 918-922, 2016, doi: 10.1109/ICISCE.2016.200.

[42] R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*. MIT Press, 2011.

[43] G. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge University Press, 2010, doi: 10.1017/CBO9780511780929.

[44] S. G. Tzafestas, *Introduction to Mobile Robot Control*. Elsevier, 2014, doi: 10.1016/B978-0-12-417049-0.00005-5.

[45] S. Jonsson, “New AGV with Revolutionart Movement,” In *3rd International Conference on Automated Guided Vehicles*, pp. 135-144. 1985.

[46] B. Carlisle, “An omni-directional mobile robot,” *Development in robotics*, 1983.

[47] J. Agulló, S. Cardona and J. Vivancos, “Kinematics of vehicles with directional sliding wheels,” *Mechanism and Machine Theory*, vol. 22, no. 4, pp. 295-301, 1987, doi: 10.1016/0094-114X(87)90018-8.

[48] S. L. Dickerson and B. D. Lapin, “Control of an omni-directional robotic vehicle with Mecanum wheels,” In *NTC'91-National Telesystems Conference Proceedings*, pp. 323-328, 1991, doi: 10.1109/NTC.1991.148039.

[49] L. Ferriere, B. Raucent and G. Campion, “Design of omnimobile robot wheels,” *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3664-3670, 1996, doi: 10.1109/ROBOT.1996.509271.

[50] W. Chung and K. Iagnemma, “Wheeled robots,” *Springer Handbook of Robotics*, pp. 575-594, 2016, doi: 10.1007/978-3-319-32552-1_24.

[51] N. Shiroma, Y. -h. Chiu, Z. Min, I. Kawabuchi and F. Matsuno, “Development and Control of a High Maneuverability Wheeled Robot with Variable-Structure Functionality,” *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4000-4005, 2006, doi: 10.1109/IROS.2006.281839.

[52] S. D. Lee and S. Jung, “A recursive least square approach to a disturbance observer design for balancing control of a single-wheel robot system,” *2016 IEEE International Conference on Information and Automation (ICIA)*, pp. 1878-1881, 2016, doi: 10.1109/ICInfA.2016.7832125.

[53] H. Cho and J. J. Lee, *Proceedings of the 2002 FIRA World Congress*, 2002.

[54] J. Borenstein, H. R. Everett, and L. Feng, *Navigating Mobile Robots: Systems and Techniques*. Wellesley, MA: AK Peters, Ltd., 1998.

[55] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA: MIT Press, 1998.

[56] J. L. Jones, B. A. Seiger and A. M. Flynn, *Mobile Robots: Inspiration to Implementation*. Wellesley, MA: AK Peters/CRC Press, 1998, doi: 10.1201/9781439863985.

[57] P. Mckerrow, *Introduction to Robotics*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1991.

[58] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*, vol. 37, no. 1, pp. 89-103, 2013, https://doi.org/10.1016/j.arcontrol.2013.03.004.

[59] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and Autonomous Systems*, vol. 55, no. 6, pp. 460-469, 2007, doi: 10.1016/j.robot.2007.01.002.

[60] G. Klancar, A. Zdesar, S. Blazic, and I. Skrjanc, *Wheeled Mobile Robotics: From Fundamentals Towards Autonomous Systems*. Oxford: Butterworth-Heinemann, 2017.

[61] P. Mckerrow, *Introduction to Robotics*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1991.

[62] P. Glotfelter and M. Egerstedt, "A Parametric MPC Approach to Balancing the Cost of Abstraction for Differential-Drive Mobile Robots," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 732-737, 2018, doi: 10.1109/ICRA.2018.8461234.

[63] A. A. Rodriguez *et al*., "Modeling, design and control of low-cost differential-drive robotic ground vehicles: Part II — Multiple vehicle study," *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 161-166, 2017, doi: 10.1109/CCTA.2017.8062457.

[64] L. -Y. Hsu and T. -L. Chen, "An Optimal Wheel Torque Distribution Controller for Automated Vehicle Trajectory Following," in *IEEE Transactions on Vehicular Technology*, vol. 62, no. 6, pp. 2430-2440, July 2013, doi: 10.1109/TVT.2013.2246593.

[65] C. Myint and N. N. Win, "Position and velocity control for two-wheel differential drive mobile robot," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 5, no. 9, pp. 2849-2855, 2016.

[66] A. A. Mahfouz, A. A. Aly and F. A. Salem, "Mechatronics design of a mobile robot system," *International Journal of Intelligent Systems and Applications*, vol. 5, no. 3, pp. 23-36, 2013, doi: 10.5815/ijisa.2013.03.03.

[67] G. Campion, G. Bastin and B. D'Andrea-Novel, "Structural properties and classification of kinematic and dynamic models of wheeled mobile robots," *[1993] Proceedings IEEE*

International Conference on Robotics and Automation, vol. 1, pp. 462-469, 1993, doi: 10.1109/ROBOT.1993.292023.

[68] G. Campion and W. Chung, "Wheeled robots," in *Springer Handbook of Robotics*, pp. 391-410, Springer, Berlin, Heidelberg, 2008, doi: 10.1007/978-3-540-30301-5_18.

[69] R. Dhaouadi and A. Abu Hatab, "Dynamic modelling of differential-drive mobile robots using Lagrange and Newton-Euler methodologies: A unified framework," *Advances in Robotics & Automation*, vol. 2, no. 2, pp. 1-7, 2013, doi: 10.4172/2168-9695.1000107.

[70] M. Mihelj *et al*., "Mobile robots," in *Robotics*, pp. 189-208, 2019, doi: 10.1007/978-3-319-72911-4_13.

[71] J. A. Angelo, *Robotics: A Reference Guide to the New Technology*. Westport: Greenwood Press, 2007.

[72] P. F. Muir and C. P. Neuman, "Kinematic modeling of wheeled mobile robots," *Journal of Robotic Systems*, vol. 4, no. 2, pp. 281-340, 1987, doi: 10.1002/rob.4620040209

[73] J. C. Alexander and J. H. Maddocks, "On the kinematics of wheeled mobile robots," *The International Journal of Robotics Research*, vol. 8, no. 5, pp. 15-27, 1989, doi: 10.1177/027836498900800502.

[74] D.-S. Kim, W.-H. Kwon and H.-S. Park, "Geometric kinematics and applications of a mobile robot," *International Journal of Control, Automation, and Systems*, vol. 1, no. 3, pp. 376-384, 2003.

[75] R. Rajagopalan, "A generic kinematic formulation for wheeled mobile robots," *Journal of Robotic Systems*, vol. 14, no. 2, pp. 77-91, 1997, doi: 10.1002/(SICI)1097-4563(199702)14:2%3C77::AID-ROB3%3E3.0.CO;2-Q.

[76] T. Phairoh and K. Williamson, "Autonomous mobile robots using real time kinematic signal correction and global positioning system control," in *Proceedings of IAJC-IJME International Conference on Industrial Technology*, Nov. 17-19, 2008.

[77] N. A. Martins and D. W. Bertol, *Wheeled Mobile Robot Control: Theory, Simulation, and Experimentation*, vol. 380, Springer Nature, 2022, doi: 10.1007/978-3-030-77912-2.

[78] M. Crenganis and O. Bologa, "Implementing PID Controller for a Mobile Platform," *Buletinul AGIR*, suppl. 1, pp. 143-148, 2015.

[79] G. Campion, B. d'Andrea-Novel and G. Bastin, "Modelling and state feedback control of nonholonomic mechanical systems," *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, vol. 2, pp. 1184-1189, 1991, doi: 10.1109/CDC.1991.261553.

[80] P. Coelho and U. Nunes, "Path-following control of mobile robots in presence of uncertainties," in *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 252-261, April 2005, doi: 10.1109/TRO.2004.837240.

[81] Y. Yamamoto and Xiaoping Yun, “Coordinating locomotion and manipulation of a mobile manipulator,” in *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1326-1332, June 1994, doi: 10.1109/9.293207.

[82] S. Khatoon, M. Istiyaque, S. A. Wani and M. Shahid, “Design kinematics and control for a differential drive mobile robot,” in *Renewable Power for Sustainable Growth: Proceedings of International Conference on Renewable Power (ICRP 2020)*, pp. 189-196, Springer Singapore, 2021, doi: 10.1007/978-981-33-4080-0_18.

[83] R. Fierro and F. L. Lewis, “Control of a nonholonomic mobile robot using neural networks,” in *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 589-600, July 1998, doi: 10.1109/72.701173.

[84] H. Choset *et al*., *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[85] I. Kolmanovsky and N. H. McClamroch, “Developments in nonholonomic control problems,” in *IEEE Control Systems Magazine*, vol. 15, no. 6, pp. 20-36, Dec. 1995, doi: 10.1109/37.476384.

[86] A. M. Bloch, “Nonholonomic Mechanics,” in *Nonholonomic Mechanics and Control*, P. Krishnaprasad and R. Murray, Eds., Interdisciplinary Applied Mathematics, vol. 24, Springer, New York, NY, 2015, doi: 10.1007/978-1-4939-3017-3_5.

[87] J. Minguez, F. Lamiraux and J. P. Laumond, “Motion Planning and Obstacle Avoidance,” in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., Springer Handbooks, Springer, Cham, 2016, doi: 10.1007/978-3-319-32552-1_47.

[88] Z. Li and J. Canny, Eds., *Nonholonomic Motion Planning*. Springer Science & Business Media, 1993, doi: 10.1007/978-1-4615-3176-0.

[89] N. Correll, B. Hayes, C. Heckman and A. Roncone, *Introduction to Autonomous Robots: Mechanisms, Sensors, Actuators, and Algorithms*. MIT Press, 2022.

[90] J. J. Craig, *Introduction to robotics*. Pearson Educacion, 2006.

[91] W. Abbasi, “Stabilization of Nonholonomic Systems,” Doctoral dissertation, Capital University of Science and Technology, Islamabad, 2018.

[92] M. Ben-Ari and F. Mondada, *Elements of Robotics*. Springer Nature, 2017, doi: 10.1007/978-3-319-62533-1.

[93] M. Gnanaprakash, “Study on Mobile Robot Path Planning–A Review,” *International Journal of Applied Engineering Research*, vol. 10, no. 57, p. 2015, 2015.

[94] A. N. A. Rafai, N. Adzhar and N. I. Jaini, “A review on path planning and obstacle avoidance algorithms for autonomous mobile robots,” *Journal of Robotics*, 2022, doi: 10.1155/2022/2538220.

[95] A. Atyabi, S. Phon-Amnuaisuk and C. K. Ho, “Navigating a robotic swarm in an uncharted 2D landscape,” *Applied Soft Computing*, vol. 10, no. 1, pp. 149-169, 2010, doi: 10.1016/j.asoc.2009.06.017.

[96] P. Raja and S. Pugazhenthi, “Optimal path planning of mobile robots: A review,” *International Journal of Physical Sciences*, vol. 7, no. 9, pp. 1314-1320, 2012, doi: 10.5897/IJPS11.1745.

[97] N. A. K. Zghair and A. S. Al-Araji, “A one decade survey of autonomous mobile robot systems,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, p. 4891, 2021, doi: 10.11591/ijece.v11i6.pp4891-4906.

[98] M. A. H. Ali and I. H. Shanono, “Path planning methods for mobile robots: A systematic and bibliometric review,” *ELEKTRIKA-Journal of Electrical Engineering*, vol. 19, no. 3, pp. 14-34, 2020, doi: 10.11113/elektrika.v19n3.225.

[99] H. S. Hewawasam, M. Y. Ibrahim and G. K. Appuhamillage, “Past, Present and Future of Path-Planning Algorithms for Mobile Robot Navigation in Dynamic Environments,” in *IEEE Open Journal of the Industrial Electronics Society*, vol. 3, pp. 353-365, 2022, doi: 10.1109/OJIES.2022.3179617.

[100] T. T. Hoang, V. C. Thanh, N. N. A. Quan and T. L. T. Dong, “Stabilization Controller Design for Differential Mobile Robot Using Lyapunov Function and Extended Kalman Filter,” in *Industrial Networks and Intelligent Systems: 8th EAI International Conference, INISCOM 2022, Proceedings*, pp. 201-213, Cham: Springer International Publishing, 2022, doi: 10.1007/978-3-031-08878-0_14.

[101] A. Jokić, M. Petrović and Z. Miljković, “Real-Time Mobile Robot Perception Based on Deep Learning Detection Model,” in *International Conference “New Technologies, Development and Applications”*, pp. 670-677, Cham: Springer International Publishing, 2022, doi: 10.1007/978-3-031-05230-9_80.

[102] M. N. Ab Wahab, S. Nefti-Meziani and A. Atyabi, “A comparative review on mobile robot path planning: Classical or meta-heuristic methods?,” *Annual Reviews in Control*, vol. 50, pp. 233-252, 2020, doi: 10.1016/j.arcontrol.2020.10.001.

[103] P. K. Mohanty, A. K. Singh, A. Kumar, M. K. Mahto and S. Kundu, “Path Planning Techniques for Mobile Robots: A Review,” in *International Conference on Soft Computing and Pattern Recognition*, pp. 657-667, Cham: Springer International Publishing, 2021, doi: 10.1007/978-3-030-96302-6_62.

[104] P. T. Kyaw *et al*., “Energy-Efficient Path Planning of Reconfigurable Robots in Complex Environments,” in *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2481-2494, Aug. 2022, doi: 10.1109/TRO.2022.3147408.

[105] M. S. Abed, O. F. Lutfy and Q. F. Al-Doori, “A review on path planning algorithms for mobile robots,” *Engineering and Technology Journal*, vol. 39, no. 5A, pp. 804-820, 2021, doi: 10.30684/etj.v39i5A.1941.

[106] S. Nurmaini and B. Tutuko, “Intelligent robotics navigation system: Problems, methods, and algorithm,” *International Journal of Electrical and Computer Engineering*, vol. 7, no. 6, p. 3711, 2017, doi: 10.11591/ijece.v7i6.pp3711-3726.

[107] S.-H. Joo *et al*., “Autonomous navigation framework for intelligent robots based on a semantic environment modeling,” *Applied Sciences*, vol. 10, no. 9, p. 3219, 2020, doi: 10.3390/app10093219.

[108] S. Khan and M. K. Ahmmed, “Where am I? Autonomous navigation system of a mobile robot in an unknown environment,” *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 56-61, 2016, doi: 10.1109/ICIEV.2016.7760188.

[109] M. Dirik, O. Castillo and F. Kocamaz, *Vision-Based Mobile Robot Control and Path Planning Algorithms in Obstacle Environments Using Type-2 Fuzzy Logic*, vol. 407. Springer Nature, 2021, doi: 10.1007/978-3-030-69247-6.

[110] F. L. Lewis and S. S. Ge, Eds., *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*. CRC Press, 2018.

[111] M. Sarcinelli-Filho and R. Carelli, *Control of Ground and Aerial Robots*, vol. 103, Springer Nature, 2023, doi: 10.1007/978-3-031-23088-2.

[112] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge, UK: Cambridge University Press, 2017.

[113] A. De Luca, G. Oriolo and M. Vendittelli, “Control of wheeled mobile robots: An experimental overview,” in *RAMSETE: Articulated and Mobile Robotics for Services and Technologies*, pp. 181-226, 2002, doi: 10.1007/3-540-45000-9_8.

[114] P. Morin, “Control of Mobile Robots,” *Encyclopedia of Robotics*, pp. 1-9, 2023, doi: 10.1007/978-3-642-41610-1_60-1.

[115] P. Morin and C. Samson, “Motion control of wheeled mobile robots,” in *Springer Handbook of Robotics*, vol. 1, pp. 799-826, 2008, doi: 10.1007/978-3-540-30301-5_35.

[116] C. Samson, P. Morin and R. Lenain, “Modeling and control of wheeled mobile robots,” in *Springer Handbook of Robotics*, pp. 1235-1266, 2016, doi: 10.1007/978-3-319-32552-1_49.

[117] C. Caceres, J. M. Rosario and D. Amaya, “Approach of Kinematic Control for a Nonholonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms,” *2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI)*, pp. 1-6, 2017, doi: 10.1109/IWOBI.2017.7985533.

[118] G. Farias *et al*., "Position control of a mobile robot using reinforcement learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17393-17398, 2020, doi: 10.1016/j.ifacol.2020.12.2093.

[119] S. G. Tzafestas, "Mobile robot control and navigation: A global overview," *Journal of Intelligent & Robotic Systems*, vol. 91, pp. 35-58, 2018, doi: 10.1007/s10846-018-0805-9.

[120] A. Noormohammadi-Asl, M. Saffari and M. Teshnehlab, "Neural Control of Mobile Robot Motion Based on Feedback Error Learning and Mimetic Structure," *Electrical Engineering (ICEE), Iranian Conference on*, pp. 778-783, 2018, doi: 10.1109/ICEE.2018.8472657.

[121] H. Huang, J. Zhou, Q. Di, J. Zhou and J. Li, "Robust neural network–based tracking control and stabilization of a wheeled mobile robot with input saturation," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 2, pp. 375-392, 2019, doi: 10.1002/rnc.4396.

[122] H. Niu, N. Wang and N. Li, "The adaptive control based on BP neural network identification for two-wheeled robot," *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 2437-2442, 2016, doi: 10.1109/WCICA.2016.7578658.

[123] X. Feng and C. Wang, "Adaptive neural network tracking control of an omnidirectional mobile robot," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 237, no. 3, pp. 375-387, 2023, doi: 10.1177/095965182211359.

[124] T. T. K. Ly, N. T. Thanh, H. Thien and T. Nguyen, "A Neural Network Controller Design for the Mecanum Wheel Mobile Robot," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10541-10547, 2023, doi: 10.48084/etasr.5761.

[125] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," in *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi A Zadeh*, pp. 775-782, 1996, doi: 10.1142/9789814261302_0040.

[126] L. A. Zadeh, "Soft computing and fuzzy logic," in *IEEE Software*, vol. 11, no. 6, pp. 48-56, Nov. 1994, doi: 10.1109/52.329401.

[127] A. El Farnane *et al*., "Trajectory tracking of autonomous driving tricycle robot with fuzzy control," *International Review of Automatic Control*, vol. 15, no. 2, pp. 80-86, 2022, doi: 10.15866/ireaco.v15i2.21719.

[128] F. Cuevas, O. Castillo and P. Cortés-Antonio, "Design of a Control Strategy Based on Type-2 Fuzzy Logic for Omnidirectional Mobile Robots," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 37, 2021.

[129] L. Busoniu, R. Babuska, B. De Schutter and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, 2017, doi: 10.1201/9781439821091.

[130] R. Gao *et al*., "Motion Control of Non-Holonomic Constrained Mobile Robot Using Deep Reinforcement Learning," *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 348-353, 2019, doi: 10.1109/ICARM.2019.8834284.

[131] G. Farias, G. Garcia, G. Montenegro, E. Fabregas, S. Dormido-Canto and S. Dormido, "Reinforcement Learning for Position Control Problem of a Mobile Robot," in *IEEE Access*, vol. 8, pp. 152941-152951, 2020, doi: 10.1109/ACCESS.2020.3018026.

[132] F. Quiroga, G. Hermosilla, G. Farias, E. Fabregas and G. Montenegro, "Position control of a mobile robot through deep reinforcement learning," *Applied Sciences*, vol. 12, no. 14, p. 7194, 2022, doi: 10.3390/app12147194.

[133] J. Xie and Q. Wang, "Intelligent Control for a Non-holonomic Constrained Mobile Robot with Proximal Policy Optimization," *2022 34th Chinese Control and Decision Conference (CCDC)*, pp. 2913-2918, 2022, doi: 10.1109/CCDC55256.2022.10033883.

[134] D. Zhang, G. Wang and Z. Wu, "Reinforcement Learning-Based Tracking Control for a Three Mecanum Wheeled Mobile Robot," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 1, pp. 1445-1452, Jan. 2024, doi: 10.1109/TNNLS.2022.3185055.

[135] J. Bernat, P. Czopek and S. Bartosik, "Analysis of Mobile Robot Control by Reinforcement Learning Algorithm," *Electronics*, vol. 11, no. 11, p. 1754, 2022, doi: 10.3390/electronics11111754.

[136] A. Diveev and E. Shmalko, "Machine Learning Feedback Control Approach Based on Symbolic Regression for Robotic Systems," *Mathematics*, vol. 10, no. 21, p. 4100, 2022, doi: 10.3390/math10214100.

[137] S.-M. Udrescu and M. Tegmark, "AI Feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, art. eaay2631, 2020, doi: 10.1126/sciadv.aay2631.

[138] Y. Jin, W. Fu, J. Kang, J. Guo and J. Guo, "Bayesian symbolic regression," *arXiv preprint arXiv:1910.08892*, 2019, doi: 10.48550/arXiv.1910.08892.

[139] W. La Cava and J. H. Moore, "Learning feature spaces for regression with genetic programming," *Genetic Programming and Evolvable Machines*, vol. 21, no. 3, pp. 433-467, 2020, doi: 10.1007/s10710-020-09383-4.

[140] B. K. Petersen, M. Landajuela, T. N. Mundhenk, C. P. Santiago, S. K. Kim and J. T. Kim, "Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients," *arXiv preprint arXiv:1912.04871*, 2019, doi: 10.48550/arXiv.1912.04871.

[141] E. Shmalko and A. Diveev, "Control synthesis as machine learning control by symbolic regression methods," *Applied Sciences*, vol. 11, no. 12, p. 5468, 2021, doi: 10.3390/app11125468.

[142] A. Diveev and E. Shmalko, "Optimal Feedback Control through Numerical Synthesis of Stabilization System," *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 112-117, 2020, doi: 10.1109/CoDIT49905.2020.9263787.

[143] E. Shmalko, “Computational Approach to Optimal Control in Applied Robotics,” in *Frontiers in Robotics and Electromechanics*, pp. 387-401, Singapore: Springer Nature Singapore, 2023, doi: 10.1007/978-981-19-7685-8_25.

[144] E. Shmalko and A. Diveev, “Extended Statement of the Optimal Control Problem and Machine Learning Approach to Its Solution,” *Mathematical Problems in Engineering*, 2022, doi: 10.1155/2022/1932520.

[145] E. Shmalko, “Feasibility of synthesized optimal control approach on model of robotic system with uncertainties,” in *Electromechanics and Robotics: Proceedings of 16th International Conference on Electromechanics and Robotics "Zavalishin's Readings" (ER (ZR) 2021)*, pp. 131-143, Springer Singapore, 2022, doi: 10.1007/978-981-16-2814-6_12.

[146] A. Diveev and E. Sofronova, “Automation of synthesized optimal control problem solution for mobile robot by genetic programming,” in *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*, pp. 1054-1072, Springer International Publishing, 2020, doi: 10.1007/978-3-030-29513-4_77.

[147] A. Diveev and G. Balandina, “Optimal Trajectories Synthesis of a Mobile Robots Group Using Cartesian Genetic Programming,” *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 130-135, 2020, doi: 10.1109/CoDIT49905.2020.9263782.

[148] A. Diveev and E. Shmalko, “Research of Trajectory Optimization Approaches in Synthesized Optimal Control,” *Symmetry*, vol. 13, no. 2, p. 336, 2021, doi: 10.3390/sym13020336.

[149] A. Diveev and E. Sofronova, “Synthesized Control for Optimal Control Problem of Motion Along the Program Trajectory,” *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 475-480, 2022, doi: 10.1109/CoDIT55151.2022.9803924.

[150] A. Diveev, “The Refined Optimal Control Problem and Synthesized Control Method for its Solution,” *2022 30th Mediterranean Conference on Control and Automation (MED)*, pp. 176-181, 2022, doi: 10.1109/MED54222.2022.9837245.

[151] F. Gul, S. S. N. Alhady and W. Rahiman, “A review of controller approach for autonomous guided vehicle system,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 552-562, 2020, doi: 10.11591/ijeecs.v20.i1.pp552-562.

[152] A. N. Albab, E. Rahmawati, M. Yantidewi, I. Sucahyo and R. R. Firmansyah, “Control position of mobile robot based on odometry method and PID controller,” in *Journal of Physics: Conference Series*, vol. 1491, no. 1, p. 012039, IOP Publishing, 2020, doi: 10.1088/1742-6596/1491/1/012039.

[153] J. G. Romero, E. Nuño, E. Restrepo, R. Cisneros and M. Morales, “A Smooth Time–Varying PID Controller for Nonholonomic Mobile Robots Subject to Matched Disturbances,” *Journal of Intelligent & Robotic Systems*, vol. 105, no. 1, p. 13, 2022, doi: 10.1007/s10846-022-01622-3.

[154] U. Zangina, S. Buyamin, M. S. Zainal Abidin, M. S. Azimi and H. S. Hasan, “Non-linear PID controller for trajectory tracking of a differential drive mobile robot,” *Journal of Mechanical Engineering Research and Developments*, vol. 43, no. 1, pp. 255-270, 2020.

[155] N. H. Thai, T. T. K. Ly, H. Thien and L. Q. Dzung, “Trajectory tracking control for differential-drive mobile robot by a variable parameter PID controller,” *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 8, pp. 614-621, 2022, doi: 10.18178/ijmerr.11.8.614-621.

[156] N. H. Thai and T. T. K. Ly, “Trajectory tracking control for mecanum wheel mobile robot by time-varying parameter PID controller,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 4, pp. 1902-1910, 2022, doi: 10.11591/eei.v11i4.3712.

[157] S. Vaidyanathan and A. T. Azar, Eds., *Backstepping Control of Nonlinear Dynamical Systems*. Academic Press, 2021.

[158] M. J. Rabbani and A. Y. Memon, “Trajectory tracking and stabilization of nonholonomic wheeled mobile robot using recursive integral backstepping control,” *Electronics*, vol. 10, no. 16, p. 1992, 2021, doi: 10.3390/electronics10161992.

[159] M. J. Rabbani and A. Y. Memon, “Output Feedback Stabilization of Nonholonomic Wheeled Mobile Robot Using Backstepping Control,” *2022 IEEE 12th International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 119-124, 2022, doi: 10.1109/ICCSCE54767.2022.9935650.

[160] W. M. E. Mahgoub and I. M. H. Sanhoury, “Back stepping tracking controller for wheeled mobile robot,” *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, pp. 1-5, 2017, doi: 10.1109/ICCCCEE.2017.7867663.

[161] I. Hassani, I. Maalej and C. Rekik, “Backstepping tracking control for nonholonomic mobile robot,” *2020 4th International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, pp. 63-68, 2020, doi: 10.1109/IC_ASET49463.2020.9318221.

[162] S. Fadlo, A. Ait Elmahjoub and N. Rabbah, “Optimal trajectory tracking control for a wheeled mobile robot using backstepping technique,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, p. 5979, 2022, doi: 10.11591/ijece.v12i6.pp5979-5987.

[163] W. M. E. Mahgoub and I. M. H. Sanhoury, “Tracking Control of Unicycle-type Wheeled Mobile Robot Utlizing Backstepping Approach,” *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-5, 2021, doi: 10.1109/ICCCEEE49695.2021.9429613.

[164] S.-C. Tan, Y.-M. Lai and C.-K. Tse, *Sliding Mode Control of Switching Power Converters: Techniques and Implementation*. CRC Press, 2018, doi: 10.1201/9781315217796.

[165] M. Thomas, B. Bandyopadhyay and L. Vachhani, "Finite-time posture stabilization of the unicycle mobile robot using only position information: A discrete-time sliding mode approach," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 6, pp. 1990-2006, 2019, doi: 10.1002/rnc.4480.

[166] M. Mera, H. Ríos and E. A. Martínez, "A sliding-mode based controller for trajectory tracking of perturbed unicycle mobile robots," *Control Engineering Practice*, vol. 102, art. 104548, 2020, doi: 10.1016/j.conengprac.2020.104548.

[167] B. Moudoud, H. Aissaoui and M. Diany, "Robust adaptive trajectory tracking control based on sliding mode of electrical wheeled mobile robot," *International Journal of Mechanical Engineering and Robotics Research*, vol. 10, no. 9, 2021, doi: 10.18178/ijmerr.10.9.505-509.

[168] H. Yu, N. Sheng and Z. Ai, "Sliding mode control for trajectory tracking of mobile robots," *2021 40th Chinese Control Conference (CCC)*, pp. 13-17, 2021, doi: 10.23919/CCC52363.2021.9550404.

[169] S. V. Rakovic and W. S. Levine, Eds., *Handbook of Model Predictive Control*. 2018, doi: 10.1007/978-3-319-77489-3.

[170] M. W. Mehrez, G. K. I. Mann and R. G. Gosine, "Comparison of stabilizing NMPC designs for wheeled mobile robots: An experimental study," *2015 Moratuwa Engineering Research Conference (MERCon)*, pp. 130-135, 2015, doi: 10.1109/MERCon.2015.7112333.

[171] Y. Gao and K. T. Chong, "Point Stabilization for Wheeled Mobile Robots Using Model Predictive Control," *International Journal of Control and Automation*, vol. 9, no. 5, pp. 67-78, 2016, doi: 10.14257/ijca.2016.9.5.07.

[172] M. W. Mehrez, K. Worthmann, J. P. V. Cenerini, M. Osman, W. W. Melek and S. Jeon, "Model predictive control without terminal constraints or costs for holonomic mobile robots," *Robotics and Autonomous Systems*, vol. 127, art. 103468, 2020, doi: 10.1016/j.robot.2020.103468.

[173] M. Sani, B. Robu and A. Hably, "Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control," *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1-6, 2021, doi: 10.1109/IECON48115.2021.9589658.

[174] N. N. Minh, S. McIlvanna, Y. Sun, Y. Jin and M. Van, "Safety-critical model predictive control with control barrier function for dynamic obstacle avoidance," *arXiv preprint arXiv:2211.11348*, 2022, doi: 10.48550/arXiv. 2211.11348.

[175] J. Wei and B. Zhu, "Model predictive control for trajectory-tracking and formation of wheeled mobile robots," *Neural Computing and Applications*, vol. 34, no. 19, pp. 16351-16365, 2022, doi: 10.1007/s00521-022-07195-4.

[176] M. S. de Queiroz, D. M. Dawson, S. P. Nagarkatti and F. Zhang, *Lyapunov-Based Control of Mechanical Systems*. Birkhäuser Boston, 2000, doi: 10.1007/978-1-4612-1352-9.

[177] P. Panahandeh, K. Alipour, B. Tarvirdizadeh and A. Hadi, "A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots," *Mechanical Systems and Signal Processing*, vol. 134, art. 106319, 2019, doi: 10.1016/j.ymssp.2019.106319.

[178] D. Jung and S. Bang, "Posture stabilization of wheeled mobile robot based on passivity-based robust switching control with model uncertainty compensation," *Applied Sciences*, vol. 9, no. 23, p. 5233, 2019, doi: 10.3390/app9235233.

[179] T. Zhao, P. Qin and Y. Zhong, "Trajectory Tracking Control Method for Omnidirectional Mobile Robot Based on Self-Organizing Fuzzy Neural Network and Preview Strategy," *Entropy*, vol. 25, no. 2, p. 248, 2023, doi: 10.3390/e25020248.

[180] M. Q. Zaman and H. -M. Wu, "Fuzzy Reinforcement Learning Based Trajectory-tracking Control of an Autonomous Mobile Robot," *2022 22nd International Conference on Control, Automation and Systems (ICCAS)*, pp. 840-845, 2022, doi: 10.23919/ICCAS55662.2022.10003839.

[181] F. Fufa, L. Duguma and E. Ayenew, "Trajectory Tracking of a Two-Wheeled Mobile Robot Using Backstepping and Nonlinear PID Controller," in *International Conference on Advances of Science and Technology*, Cham, Switzerland: Springer Nature Switzerland, pp. 290-304, 2022, doi: 10.1007/978-3-031-28725-1_18.

[182] C. Mireles-Perez, D. Cruz-Ortiz, I. Salgado and I. Chairez, "Backstepping second order sliding mode control for a car-like robot," *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 463-467, 2022, doi: 10.1109/CoDIT55151.2022.9803917.

[183] R. Rouhi Ardeshiri, M. Gheisarnejad, M. R. Tavan, N. Vafamand and M.-H. Khooban, "A robust intelligent controller-based motion control of a wheeled mobile robot," *Transactions of the Institute of Measurement and Control*, vol. 44, no. 15, pp. 2911-2918, 2022, doi: 10.1177/01423312221088389.

[184] K. Yeom, "Design of deep neural network based model predictive controller for a car-like mobile robot," *International Journal of Mechanical Engineering and Robotics Research*, vol. 11, no. 8, pp. 606-613, 2022, doi: 10.18178/ijmerr.11.8.606-613.

[185] G. da Silva Lima, V. R. Firmo Moreira and W. M. Bessa, "Accurate trajectory tracking control with adaptive neural networks for omnidirectional mobile robots subject to unmodeled dynamics," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 45, no. 1, p. 48, 2023, doi: 10.1007/s40430-022-03969-y.

[186] T. Kim and R. Prakapovich, “Automatic Tuning of the Motion Control System of a Mobile Robot Along a Trajectory Based on the Reinforcement Learning Method,” in *International Conference on Pattern Recognition and Information Processing*, Cham, Switzerland: Springer International Publishing, pp. 234-244, 2021, doi: 10.1007/978-3-030-98883-8_17.

[187] C.-T. Lee and W.-T. Sung, “Controller Design of Tracking WMR system based on deep reinforcement learning,” *Electronics*, vol. 11, no. 6, p. 928, 2022, doi: 10.3390/electronics11060928.

[188] A. Al-Jodah *et al*., “PSO-based optimized neural network PID control approach for a four wheeled omnidirectional mobile robot,” *International Review of Applied Sciences and Engineering*, vol. 14, no. 1, pp. 58-67, 2023, doi: 10.1556/1848.2022.00420.

[189] F. Pang *et al.*, “Path tracking control of an omni-directional service robot based on model predictive control of adaptive neural-fuzzy inference system,” *Applied Sciences*, vol. 11, no. 2, p. 838, 2021, doi: 10.3390/app11020838.

[190] B. Moudoud, H. Aissaoui and M. Diany, “Fuzzy adaptive sliding mode controller for electrically driven wheeled mobile robot for trajectory tracking task,” *Journal of Control and Decision*, vol. 9, no. 1, pp. 71-79, 2022, doi: 10.1080/23307706.2021.1912665.

[191] G. Cao, X. Zhao, C. Ye, S. Yu, B. Li and C. Jiang, “Fuzzy adaptive PID control method for multi-mecanum-wheeled mobile robot,” *Journal of Mechanical Science and Technology*, vol. 36, no. 4, pp. 2019-2029, 2022, doi: 10.1007/s12206-022-0337-x.

[192] A. Diveev and E. Shmalko, *Machine Learning Control by Symbolic Regression*. Berlin/Heidelberg, Germany: Springer International Publishing, 2021, doi: 10.1007/978-3-030-83213-1.

[193] R. Bellman, I. Glickberg and O. Gross, “Some Aspects of the Mathematical Theory of Control Processes,” Rand Corporation, Report R-313, Santa Monica, California, 1958.

[194] R. Bellman and R. E. Kalaba, *Dynamic Programming and Modern Control Theory*, vol. 81. New York: Academic Press, 1965.

[195] R. E. Bellman and S. E. Dreyfus, *Applied Dynamic Programming*, vol. 2050. Princeton, NJ: Princeton University Press, 2015.

[196] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze and E. F. Mishchenko, “The Mathematical Theory of Optimal Processes,” *Interscience*, New York, vol. 171, pp. 276-294, 1962.

[197] V. G. Boltyanskii, K. N. Trirogoff, I. Tarnove and G. Leitmann, “Mathematical Methods of Optimal Control,” 1971, doi: 10.1115/1.3426517.

[198] A. Diveev and E. Shmalko, “Multi-point Stabilization Approach to the Optimal Control Problem with Uncertainties,” in *International Conference on Optimization and Applications*, Cham,

Switzerland: Springer International Publishing, pp. 129-142, 2020, doi: 10.1007/978-3-030-65739-0_10.

[199] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1998, doi: 10.7551/mitpress/3927.001.0001.

[200] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley, vol. 102, p. 36, 1989.

[201] M. Kumar, M. Husain, N. Upreti and D. Gupta, "Genetic Algorithm: Review and Application," *SSRN Electronic Journal*, 2010, doi: 10.2139/ssrn.3529843.

[202] A. I. Diveev, "Small Variations of Basic Solution Method for Non-numerical Optimization," *IFAC-PapersOnLine*, vol. 48, no. 25, pp. 28–33, 2015, doi: 10.1016/j.ifacol.2015.11.054.

[203] E. Sofronova and A. Diveev, "Universal Approach to Solution of Optimization Problems by Symbolic Regression," *Applied Sciences*, vol. 11, no. 11, p. 5081, May 2021, doi: 10.3390/app11115081.

[204] J. R. Koza, *Genetic Programming II*, vol. 17. Cambridge, MA: MIT Press, 1994.

[205] I. Zelinka, Z. Oplatkova and L. Nolle, "Analytic programming–Symbolic regression by means of arbitrary evolutionary algorithms," *International Journal of Simulation: Systems, Science and Technology*, vol. 6, no. 9, pp. 44-56, 2005.

[206] J. F. Miller and S. L. Harding, "Cartesian Genetic Programming," in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pp. 3489-3512, 2009, doi: 10.1145/1570256.1570428.

[207] A. I. Diveev and E. A. Sofronova, "Numerical method of network operator for multiobjective synthesis of optimal control system," *2009 IEEE International Conference on Control and Automation*, pp. 701-708, 2009, doi: 10.1109/ICCA.2009.5410619.

[208] C. Luo and S.-L. Zhang, "Parse-matrix evolution for symbolic regression," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 6, pp. 1182–1193, Sep. 2012, doi: 10.1016/j.engappai.2012.05.015.

[209] A. Diveev and E. Sofronova, "Automation of synthesized optimal control problem solution for mobile robot by genetic programming," in *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*, Springer International Publishing, pp. 1054-1072, 2020, doi: 10.1007/978-3-030-29513-4_77.

[210] K. S. Nassrullah, I. V. Stepanyan, H. S. Nasrallah, N. J. Mendez Florez, A. M. Zidoun and S. R. Mohammed, "Unsupervised Machine Learning Control Techniques for Solving the General Synthesis of Control System Problem," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 3, pp. 401-416, 2024, doi: 10.22266/ijies2024.0630.32.

[211] K. S. Nassrullah, I. V. Stepanyan, A. A. Ali, H. S. Nasrallah, and NJ. M. Florez, “Problem of Control Synthesis of Stabilization System for a Nonholonomic Mobile Robot: An Autonomous Solution via Modified Synthesized Genetic Programming Method,” *International Journal of Intelligent Engineering and Systems*, vol. 18, no. 6, pp. 350-365, 2025, doi: 10.22266/ijies2025.0731.22.

[212] J. Kennedy and R. Eberhart, “Particle swarm optimization,” *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995, doi: 10.1109/ICNN.1995.488968.

[213] A. I. Diveev and S. V. Konstantinov, “Study of the Practical Convergence of Evolutionary Algorithms for the Optimal Program Control of a Wheeled Robot,” *Journal of Computer and Systems Sciences International*, vol. 57, no. 4, pp. 561–580, Jul. 2018, doi: 10.1134/s106423071804007x.

[214] P. Šuster and A. Jadlovská, “Tracking Trajectory of the Mobile Robot Khepera II Using Approaches of Artificial Intelligence,” *Acta Electrotechnica et Informatica*, vol. 11, no. 1, Jan. 2011, doi: 10.2478/v10198-011-0006-y.

APPENDIX I.

The operations of addition and multiplication were used as binary operations, and a set of 28 smooth elementary functions was used as unary operations. The total number of functions was 30, and they were used as the space for codes in the first step (the stabilization step) and as follows:

$f_1(z) = z$	$f_2(z) = z^2$
$f_3(z) = -z$	$f_4(z) = sgn(z)\sqrt{\lvert z\rvert}$
$f_5(z) = z^{-1}$	$f_6(z) = exp(z)$
$f_7(z) = ln\,(\lvert z\rvert)$	$f_8(z) = tanh\,(0.5z)$
$f_9(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & otherwise \end{cases}$	$f_{10}(z) = sgn(z)$
$f_{11}(z) = cos\,(z)$	$f_{12}(z) = sin\,(z)$
$f_{13}(z) = arctan\,(z)$	$f_{14}(z) = z^3$
$f_{15}(z) = \sqrt[3]{z}$	$f_{16}(z) = \begin{cases} z, & \text{if } \lvert z\rvert < 1 \\ sgn(z), & otherwise \end{cases}$
$f_{17}(z) = sgn(z)\,ln(\lvert z\rvert + 1)$	$f_{18}(z) = sgn(z)\,(exp(\lvert z\rvert) - 1)$
$f_{19}(z) = sgn(z)\,exp(-\lvert z\rvert)$	$f_{20}(z) = 0.5z$
$f_{21}(z) = 2z$	$f_{22}(z) = 1 - exp(-\lvert z\rvert)$
$f_{23}(z) = z - z^3$	$f_{24}(z) = \dfrac{1}{1 + exp(-z)}$
$f_{25}(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & otherwise \end{cases}$	$f_{26}(z) = \begin{cases} 0, & \text{if } \lvert z\rvert < \varepsilon \\ sgn(z), & otherwise \end{cases}$
$f_{27}(z) = sgn(z)(1 - \sqrt{1 - z^2})$	$f_{28}(z) = z(1 - exp(-z^2))$
$f_{29}(z_1, z_2) = z_1 + z_2$	$f_{30}(z_1, z_2) = z_1 z_2$

APPENDIX II.

The final code matrix of the first control function ($\tilde{u}_1$) in Eq. (3.16) was as follows:

$$\tilde{u}_1 = \begin{bmatrix} 2 & 1 & 2 & 2 & 1 & 2 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 20 & 1 \\ 2 & 5 & 6 & 10 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 12 & 22 & 23 & 21 & 25 & 26 & 27 & 28 \\ 3 & 3 & 3 & 1 & 1 & 28 & 0 & 7 & 17 & 0 & 17 & 0 & 18 & 0 & 28 & 0 & 0 & 26 & 0 & 18 \\ 1 & 2 & 3 & 11 & 12 & 0 & 0 & 0 & 0 & 11 & 15 & 19 & 0 & 0 & 0 & 0 & 0 & 24 & 15 & 0 \\ 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The final code matrix of the second control function ($\tilde{u}_2$) in Eq. (3.17) was as follows:

$$\tilde{u}_2 = \begin{bmatrix} 2 & 2 & 0 & 0 & 1 & 2 & 1 & 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 \\ 12 & 1 & 1 & 21 & 1 & 1 & 1 & 1 & 21 & 14 & 1 & 1 & 1 & 1 & 11 & 1 & 1 & 1 & 1 & 1 \\ 7 & 8 & 9 & 10 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 19 & 24 & 25 & 26 & 27 & 21 \\ 3 & 3 & 17 & 1 & 16 & 0 & 0 & 0 & 0 & 7 & 25 & 0 & 2 & 0 & 22 & 9 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 11 & 0 & 0 & 0 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 6 & 3 & 0 & 0 & 4 & 0 \\ 1 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 1 & 1 \end{bmatrix}$$

The final matrix of the small variations (SV) was as follows:

$$SV = \begin{bmatrix} 2 & 2 & 6 & 2 \\ 2 & 15 & 2 & 11 \\ 2 & 16 & 6 & 2 \\ 1 & 4 & 3 & 10 \\ 2 & 1 & 2 & 12 \\ 1 & 15 & 6 & 2 \\ 1 & 9 & 4 & 17 \\ 1 & 10 & 5 & 11 \\ 2 & 15 & 4 & 22 \\ 2 & 14 & 6 & 2 \end{bmatrix}$$