Università degli Studi Mediterranea di Reggio Calabria

Dipartimento di Ingegneria dell'Informazione, delle Infrastrutture e dell'Energia Sostenibile

Dottorato in ingegneria dell'informazione

XXXV ciclo

PhD thesis

Feasible trajectory planning problem for mobile robot subject to uncertainties and disturbances: a set based approach.

Coordinator

Prof. Tommaso Isernia

Tutor

Prof. Valerio Scordamaglia

PhD candidate

Vito Antonio Nardi

Anno Accademico 2020-2021

Alla mia famiglia.

A chi non ho mai perso
a chi ho trovato
a chi ho perso
a chi ho ritrovato
a chi perderò
a chi ritroverò.

Contents

Introduction

This thesis proposes a solution to the problem of planning the motion of a vehicle, taking into account its closed-loop dynamics, in presence of disturbance and model uncertainty. In trajectory planning, the vehicle must be provided with a path and a timing law suitable for the accomplishment of a certain mission while abiding by a series of state and actuation constraints. Such problem requires the solution of a wide series of problems of scientific relevance in information engineering and control systems research. Currently, popular approaches focus upon the online computation of a control strategy which abides by constraints or heuristic solutions relying, mainly, on finding a proper timing law given a geometric path, which may fit some proper requirements.
In contrast, only a handful of work try considering closed-loop dynamics of the vehicle. Among them, this thesis introduces the idea of a solution which does not need complex online computations apart from the evaluation of a simple control law. Namely, in this thesis, the trajectory planning is tackled for a remotely controlled skid-steer tracked mobile robot. The proposed framework consists of two main elements: a constrained control problem, whose solution is checked in order to guarantee its disturbance invariance and a series of set-based problems, based upon semi-definite programming, aimed at checking whether a manoeuvre can be part of a trajectory, which are defined in terms of series of connected segments, to be crossed at a given velocity.
More in detail, a disturbance-invariant set, associated to a static feedback control law, which embeds the region of tracking error state-space where actuation constraints are satisfied is considered. Candidate trajectory segments are checked for their feasibility before being added to a trajectory. Such check is performed through an ellipsoidal embedding

of possible states, reached by the vehicle at the end of a candidate trajectory segment. If the embedding is included in the d-invariant region associated with the controller, the trajectory segment is deemed feasible.

This thesis is organized as follows: chapter 1 offers an overview of mobile robotics, with a description of its main examples, features and issues. Chapter 2 presents the problem of motion planning for a terrestrial mobile robot along with solutions featured in literature. In chapter 3 a kinematic model of the skid-steer tracked mobile robot used in this work is derived and the problem of remote control is tackled. Chapter 4 deals with the synthesis of a control law, then a series of set-based arguments provides a feasibility test which is the basis of the construction of a trajectory for the vehicle. Finally, chapter 5 provides and discusses a series of numerical and experimental simulations with the aim to prove the effectiveness of the proposed approach. Experimental validation is carried out on a Jaguar v4 mobile robotic platform.

1

Mobile robotics

According to the *Handbook of industrial robotics* [Kumar et al., 1999] a mobile robot "can be thought of as a platform with a certain characteristic configuration, and able to transport sensors, tools, and/or manipulators to any accessible and desirable location to perform a desired task". Such a wide definition has been applied to a vast series of heterogeneous systems, all of them sharing several aspects and problems of cardinal interest to information engineering.
In this chapter, an overview of mobile robots will be provided, with a focus on terrestrial rovers. A review of the most important challenges of mobile robotics will be also provided, highlighting those of particular interest for rovers.

1.1 Mobile robots: an overview

Several classifications of mobile robots can be used, relying upon different criteria, such as their operational environment, autonomy level, operational range, or locomotion schema. A wide classification is based upon the environment they operate, leading to the definition of the following kinds of mobile robots:

- Unmanned ground vehicles (UGVs)
- Unmanned aerial vehicles (UAVs)
- Unmanned surface vehicles (USVs)
- Unmanned underwater vehicle (UUVs)
- Micro and nano robots

1.1.1 UGVs classification

From an historical point of view, the term "robot" was created in a 1920 novel by Karel Čapek to refer to what today would be labelled as an unmanned ground vehicle, namely a legged robot.
The first example of an unmanned ground vehicle are probably tele-controlled cars dating back to 1920s, followed by Soviet Teletanks in 1930s, which were also used during the Winter war against Finland, in World War II. Similar devices were developed by other armies, such as the British radio-controlled Matilda II or the German Goliath.
A giant leap in UGVs was the Shakey robot developed by Stanford and DARPA in 1966-1972 period. Shakey encompassed a series of novelties and tools which are still of interest nowadays, such as visibility graphs, A* algorithm, and computer vision methods for features extraction.
Another boost to UGVs development came from spatial exploration. Alongside the risky and expensive manned space programs, relatively cheap and safe space exploration through robotic proxies has been carried out since the Soviet Luna program. In 1966 the probe Luna 9 was the first object to perform a soft landing on another celestial body, followed within months by the US Surveyor 1 mission which paved the way for the 1969 first men on the Moon. Following decades have witnessed increasing complex and ambitious missions, the Luna 17 mission included the Lunochod rover which explored the Moon for almost a year [Gao, 2016], in 2010, following a 7-years mission, the Hayabusa probe successfully took samples of an asteroid, finally, on 19th April 2021, the Ingenuity helicopter took off from the Perseverance Mars rover.
The most common classification of UGVs [Siciliano et al., 2009] is the following:

- wheeled mobile robots (WMRs)
- tracked mobile robots
- legged mobile robots
- soft mobile robots

a brief description of these classes follows here.

Wheeled mobile robots are made of a central body interacting with the soil through a series of wheels. Wheels may belong to three classes:

- fixed wheels which are rigidly connected to robot chassis, each wheel can only rotate around its axis;

- steerable wheels which introduce another degree of freedom (with respect to fixed wheels): rotation around a vertical axis passing through the centre of the wheel;
- caster wheels, when vertical rotation axis is offset wheel vertical axis, this causes the wheel to quickly align with motion direction

each of these kinds of wheel can vary for its structure and materials. Rigid plastic or rubber wheels are common in indoor environments, while inflatable wheels are more common in road-like environments.
Locomotion schemas also influence WMRs features and performances, the most common ones are:

- differential driving - the robot has two fixed wheels whose velocities can be set separately and a caster wheel providing mechanical support (no traction). This kind of schema has, notably, zero-radius turning capabilities, it is mechanically simple at the price of a quite complex kinematics [Wong, 2008] and less stability;
- synchro-drive - the robot has three rotating wheels providing traction force. Wheels vertical rotation is synchronized, in this schema robot heading does not change, unless a third motor is used to rotate the upper part of the chassis around a moving base;
- tricycle - in this schema two fixed wheels provide traction force spinning at the same velocity. A third steerable wheel changes vehicle motion direction.
- car like - the vehicle has four wheels, two fixed and two steerable. Traction force can be either provided by steering or fixed wheels or all of them. Slippage while steering can be avoided by the means of Ackermann steering schema, although it can be ignored at lower velocities.
- four wheels steerable (4WS) - a four-wheels locomotion schema [Oftadeh et al., 2013] where all wheels are steerable. It provides smooth and precise motion at the price of a higher mechanical complexity.
- omnidirectional - in such schema, all wheels (usually three) are steerable and provide traction force, this kind of vehicle can instantaneously change its moving direction.
- two wheeled-robots - such as the popular Segway and Overboard platforms, are differentially driven self-balancing platforms whose main features are reduced weight and size.

Tracked mobile robots - are a vast class of robots usually designed for field applications. Tracked robots can overcome rugged and uneven ter-

rains at the expense of a lower efficiency and a more complex motion in comparison to wheeled vehicles[Siciliano et al., 2008; Wong, 2008]. Historically, tracked vehicles have arisen from military usage, given the high traction force of tracked locomotion schema, along with its ability to traverse different terrains and the higher robustness of tracks to gunfire-induced damages. In following decades tracked robots became popular in supporting bomb squads operations and disaster recovery and assessment operations. A pioneering deployment in a very hostile disaster scenario was carried out by Whitaker et al. from Carnegie Mellon who built robots to explore and assess the status of the Three Mile Island nuclear plant following the 1979 incident, and the melted reactor core in the partially flooded and collapsed scenario of Chernobyl disaster [Yokokohji, 2021].
Tracks, both metal and rubber ones, are driven by a series of sprocket wheels, whose number and layout determine tracked vehicle properties. Steering action is usually differential, i.e., it is obtained by applying different velocities to tracks at the two sides of the vehicle, although half-track vehicles can have steering wheels. Differential steering causes slippage of tracks, which makes the vehicle motion quite complex. Some kinds of track systems are:

- two sprockets per track: a simple design used from small vehicles, traction on track is maximum, and weight distribution is uneven;
- track suspensions/tank tracks: several wheels distribute vehicle weight across the track, this design is typical to heavy load vehicles, such as caterpillars and tanks, usually only one or two sprocket wheels (drive wheels) per track are connected to engines;
- packbot-like systems/flippers: additional independent tracked (not necessarily motorized) arms are added at both ends of tracks, guaranteeing stability on very steep terrains or enabling crossing stairs[Tao et al., 2012];
- overlapping sprockets - a design used by German Tiger I and Panther tanks, which enables a very even distribution of weights. However, its maintenance was deemed too difficult and this design was discarded in following tank models;
- other polygonal sprockets designs are conceptually similar to tank tracks, but several arrangements are used to accommodate different applications, for example vertically inclined track endings (with soil-touching side being shorter than upper one) are used to improve

vehicles ability to overcome trenches in military contexts or bigger obstacles.

Legged mobile robots - robots whose motion is actuated by legs made of rigid elements connected by prismatic or revolute joints, whose lower surfaces periodically touch the ground. This locomotion schema is more complex than wheeled and tracked ones and usually less efficient in terms of maximum load and power consumption. However, it shows interesting performances on steep terrains. Common designs are bipeds such as Atlas by Boston Dynamics, quadrupeds as the renown military-oriented BigDog and bio-inspired vehicles such as hexapods robots which are quite popular in makers' community.

Soft mobile robots are labelled as "soft" as they include non-rigid links between their components [Laschi et al., 2016], they make up a vast class of bio-inspired devices [Calisti et al., 2017] being developed since 2000s.
Common locomotion schema in mobile, terrestrial, and soft robots include:

- crawling
  - peristaltic crawling - inspired by worms, a wave of contractions crosses the body of the vehicle, periodically shrinking the size of a relatively small portion of the robot;
  - two-anchor crawling - inspired by caterpillars, the robot has a friction coefficient which is lower in its locomotion direction than backwards. It then moves by shortening and elongating its body;
  - serpentine - inspired by snakes, the robot oscillates on a horizontal plane, achieving a similar effect to two-anchor crawling.
- jumping gait - in this kind of system, inspired by grasshoppers, some mammals, frogs and more, motion happens in two phases: a charge phase, when deformable elements (e.g. springs or pneumatic actuators) accumulate potential energy and a ballistic flight phase following the release of accumulated energy;
- legged robots - hopping, running, and walking robots are already introduced as a separate UGV class in this chapter;
- deformation-induced locomotion - in such schema, actuators move the centre of mass of the vehicle out of the support polygon, causing motion through tilting; Actuators can be either balloons or suspended rods, such as in tensegrity robots;

- vibration-induced locomotion - robot can move in space by changing an anisotropic friction which "shapes" an omnidirectional force induced by vibrating elements.

1.1.2 UAVs classification

Unmanned aerial vehicles are one of the earliest forms of mobile robots, with some early prototype, such as the *RAE 1921 Target* and *RAE Larynx* by English Royal Navy, dating back to 1922. Several concurrent classifications, many of them related to aviation regulations, are proposed, in this section the most common ones will be discussed.
An interesting way to classify UAVs relies upon their main performance characteristics [Arjomandi et al., 2006], namely:

classification by range and endurance - operational range (i.e. the maximum allowable distance in UAV operations) and endurance (i.e. the maximum duration of a mission) are closely coupled, given that the longer an UAV can operate, the larger becomes its operative radius. Range and endurance are a crucial factor in UAV design, determining the choice of locomotion schema, allowable weight and almost any design parameter.
The most common classification consists of three categories:

- long range UAVs - vehicles which can stay airborne for at least 24 hours (peaking at 25 days for the Zephyr stratospheric UAV), their range spans from 1500 km to 24 000 km (Orion UAS), most of recognition UAVs and several research-oriented ones belong to this class;
- medium range UAVs - vehicles whose endurance spans from 5 to 24 hours, they encompass most of military UAVs, such as the Predator military drone;
- short range UAVs - this class includes a plethora of heterogeneous aircraft, ranging from small rotating wing drones used in photography to battlefield reconnaissance systems.

classification by service ceiling Flight/service ceiling (highest allowable altitude) defines a series of characteristics of UAVs, such as propulsion, instrumentations of interest and wing structure. According to [Arjomandi et al., 2006]:

- high altitude UAVs - UAVs flying over 10 km, such as Predator, X-45, mostly military and research vehicles;

- medium altitude UAVs - UAVs flying between 1 and 10 km high, it includes the greatest number of military drones;
- low altitude UAVs - vehicles flying with a maximum altitude of 1 km, this category includes the vast majority of commercial vehicles.

[Peter van Blyenburgh - UVS International, 2008]

classification by weight UAVs weight can vary across a wide range, playing a key role in defining the operational capabilities and features of the aircraft. UAVs can be classified, according to their weight, within the following categories:

- micro UAVs (sometimes *MAVs*), drones weighting less than 5 kg, these vehicles are usually powered by electric motors, with rotating wing design being the most common. This category spans from the popular hobbyist-oriented Parrot AR.Drone series to Raytheon Silent Eye reconnaissance glider.
- light weight UAVs, ranging from 5 kg to 50 kg
- medium weight UAVs, ranging from 50 kg to 200 kg, the long-range Zephyr Stratospheric UAV belongs to this category (140 kg)
- heavy weight UAVs, ranging from 200 kg to 2000 kg, an example is the Global Observer Stratospheric Persistent UAS, a general-purpose long-range vehicle which can perform both civil and military surveillance missions, carrying a significant payload of 181 kg
- super heavy weight UAVs, with a weight greater than 2000 kg, a class of (usually) military aircraft which includes massive vehicles such as X-45 and Global Hawk. It also includes the Orion Unmanned Aircraft System, which can serve also as an effective communication relay and has a maximum take-off weight of 5000 kg.

classification by engine type Given the variety of UAVs and the number of tasks they can fulfil several engines are used to power them. Namely, the following engines characterize UAVs:

- turbofans and turboprop engines - used mostly in HALE (high altitude long endurance) vehicles of military interest, they can guarantee a high reliability while achieving high velocity at high altitudes [Tsach et al., 1996].
- piston engines, are the most common design for bigger vehicles, although even some hobbyist gliders can be piston-powered;
- rotary engines - this class of engine inherits the reliability and cost-effectiveness of both military and civil experience. It powers tactical

military vehicles such as the Textron Shadow or strategic Leonardo's Falco;
- push and pull - although not a different kind of engine, push and pull configuration is used in some UAVs to reduce drag and increase reliability at the expense of a lower efficiency;
- electric engines - the most common design for smaller drones, it usually chosen for shorter range/lower altitude drones, although cutting-edge projects such as the Zephyr Stratospheric UAV (a solar-powered electric drone) outperformed most of non-electric competitors in terms of service ceiling and endurance.

An overall classification of UAVs has been proposed by UVS International since 2008 [Peter van Blyenburgh - UVS International, 2008] (see table 1.1) accounting for both commercial and military systems.

category	acronym	weight [kg]	range [km]	altitude [m]	endurance [hours]
micro	μ	<5	<10	250	1
mini (numbers according to national regulations)	M	<20/25/30/150	<10	150	<2
close range	CR	25-150	10-30	3000	2-4
short range	SR	50-250	30-70	3000	3-6
medium range	MR	150-500	70-200	5000	6-10
MR endurance	MRE	500-1500	>500	8000	10-18
low altitude deep penetration	LADP	250-2500	>250	50-9000	0,5-1
low altitude long endurance	LALE	15-25	>500	3000	>24
medium altitude long endurance	MALE	1000-1500	>500	5000-8000	24-48
high altitude long endurance	HALE	2500-5000	>2000	20000	24-48
stratospheric	Strato	>2500	>2000	>20000	>48
exo-stratospheric	EXO	-	-	>30500	-
unmanned combat AV	UCAV	>1000	1500	12000	2
lethal	LET	-	300	4000	3-4
decoys	DEC	150-500	0-500	50-5000	<4

Table 1.1. UAVs classification according to UVS.

1.1.3 USVs classification

Modern history of unmanned surface vehicles dates back World War II when the first mine sweeping remotely controlled surface vehicles were developed by different fighting navies. The field gained popularity in scientific research at the beginning of 1990s[Manley, 2008], well after UUVs, given the lower risk level associated to surface operations in

comparison to underwater ones. Some early examples include projects such as ARTEMIS by MIT which autonomously collected bathymetry data, or the SESAMO project for Antarctic oceanography[Caccia et al., 2005].
USVs are usually derived from different surface vehicles designs, sharing several properties with the corresponding manned vehicles.
Basing on hulls:

- rigid inflatable hulls - see [Motwani, 2012] are common in military applications because of their payload capacity and endurance;
- kayaks (single hull) - see for example SCOUT [Curcio et al., 2005] are quite common given their manoeuvrability and easiness to manufacture by modifying manned equipment;
- catamarans (twin hull) - such as SESAMO [Caccia et al., 2005], Springer [Naeem et al., 2008] and C-Enduro[Oh et al., 2014], this design guarantees lower wave resistance, higher roll stability and larger deck area[Silva et al., 2018]. Despite higher construction costs and lower manoeuvrability catamarans are quite popular given their resistance to capsizing;
- trimarans (triple hull) - although less common, autonomous trimarans are proposed as an evolution of twin hulls design. [Huang et al., 2016; Silva et al., 2018], expanding both advantages and disadvantages of twin hulls designs. See for reference [Qi et al., 2007; Peng et al., 2009].

Another property, which enables an useful classification of USVs, arises from propulsion and steering systems. Namely, USVs whose number of actuators/control inputs is lower than their number of degrees of freedom are under-actuated, otherwise they can be either fully or over-actuated.
Single hull vehicles heading and speed are usually controlled through rudders and propulsion systems, usually they lack of lateral propellers making this class of vehicles usually under-actuated.
Catamarans and trimarans are generally differentially driven, in other words heading is changed by applying a different propulsion force to the two sides of the USV, this usually leads to under-actuated systems, whose safe and accurate control is challenging. Some over-actuated designs are proposed, e.g. redundant orientable and invertible propellers, although they are more expensive[Breivik, 2010].

1.1.4 UUVs classification

Unmanned underwater vehicles, also known as underwater drones, is an umbrella term which encompasses a variety of vehicles used in scientific research, military, and many civilian applications [Antonelli, 2003].
The following paragraphs contain a brief description of main UUVs classes, along with their main properties and sub-classifications.

autonomous underwater vehicles

(AUVs) Self-Propelled Underwater Research Vehicle SPURV 1957

- long-range AUVs (LRAUVs);
- autonomous gliders;

remotely operated vehicles

(ROVs) - ROVs are UUVs whose control is remotely performed in real-time. ROVs are meant to be a replacement of manned intervention in hazardous environments or those environments which cannot be accessed by scuba divers.
A steerable torpedo was developed by Luppis-Whitehead automobile in 1864 in Imperial Austria, but it was in 1953 with POODLE device by Dimitri Rebikoff that a modern tethered ROV became fully operational. Military ROVs development was boosted by events such as the 1966 Palomares incident, when the Cable-Controlled Underwater Recovery Vehicle (CURV) was deployed by US Navy in order to locate a lost hydrogen bomb. In civil applications, following their start as "underwater eyes" [Hughes, 2015] in 1970s (i.e., inspection devices) ROVs gained prominence in oil industry for assessment and inspection of underwater infrastructures. In 1973 the crew of Royal Navy Pisces III submersible was successfully rescued by CURV III ROV, opening the doors to emergency interventions of tethered underwater vehicles. In 1980s, two famous wreckages of ships were found and inspected by remotely operated vehicles: British transatlantic cruise Titanic and German battleship Bismarck, both in the Atlantic Ocean. 1990s have witness a strong commitment by major navies on developing ROVs and AUVs that could disable sea mines, while civil sector focused upon underwater infrastructures deployment and maintenance.
A brief classification of ROVs can be carried out through three parameters[Christ and Wernli, 2008]: power source, degree of autonomy, and communication linkage.

- degree of autonomy [Huang, 2007]:
  - remotely controlled - the operator has no feedback on ROV operations, continuous surveillance and guidance are needed;
  - tele-operation - the operator is provided with video and/or sensory feedback for the continuous control of vehicle operation;
  - autonomous/semi-autonomous - a decreasing level of HRI (human-robot interaction) is needed, i.e., the vehicles performs operations of an increasing complexity with less human supervision, although operations to be carried out are chosen by the operator.
- power source:
  - surface-powered: power is provided by cable linking the vehicle to surface, this design guarantees a long endurance and the possibility to use energy-intensive tools, although it is less flexible in terms of operational range;
  - hybrid systems: the vehicle is provided with power storage devices but also a tethered power supply. Several schema can be used, e.g., batteries supply emergency power or an extra power for energy-intensive operations;
  - vehicle-powered: the vehicle relies upon its internal energy storage, it does not rely upon a cable but needs to periodically recharge batteries or fuel.
- communication systems
  - radio-frequency - robust to mechanical issues, more flexible but limited by water-induced attenuation of electromagnetic fields;
  - hard-wire - tethered communication which can be either electrical or fibre optic (usually more fragile but less power consuming and immune to electromagnetic interferences);
  - acoustic - communication through ultrasounds was used in the past, it has been dismissed with RF communications improvements.

gliders and autonomous gliders

a class of highly optimized vehicles meant to exploit, as much as possible, marine currents for motion. Autonomous gliders are equipped with buoyancy engines which allows them to periodically resurface to communicate with their remote-control station through satellite links. Gliders are used for oceanography [Sherman et al., 2001; Graver et al., 2003] but also military purposes, leading, for example, to NATO vast

series of models [Vicen-Bueno et al., 2020] and the PRC navy glider Haiyi UUVs series.

extra large UUVs

(XLUUVs) a category of vehicles brought into commercial production with the 2015 Echo Seeker by Boeing, characterized by long endurance, a significant maximum allowable depth, and a large payload. Recent XLUUV deployments such as US Navy Orca and RPC HSU-001 are de facto autonomous submersible with military applications.

1.1.5 Micro and nano robots classification

Microrobots (also microbots) are robots with characteristic dimensions smaller than 1 mm, while nanorobotics are those with characteristic dimensions smaller than 1 µm, i.e. devices whose components are within the scale of, respectively, micro and nanometres.
In 1959 Richard Feynman [Feynman, 2018] affirmed that *at the atomic level, we have new kinds of forces and new kinds of possibilities, new kinds of effects. The problems of manufacture and reproduction of materials will be quite different. The principles of physics, as far as I can see, do not speak against the possibility of manoeuvring things atom by atom.* sketching the main characteristic of small-scale robotics: exploiting physical principles which are negligible (or, at least, not usable) at the usual scales of technology.
In modern days there are several micro technologies in commercial use, while nano are still usually limited to research. The following sections will briefly describe both kinds of robots in mobile robotics context.

microrobots

Following the development of *micro-electro-mechanics* (MEMS) technologies at the end of 1980s, micrometre-scale actuators became widely available. Most of microrobots are made of silicon and powered through wires. They are usually steerable platforms carrying certain tools such as micro-catheters [Park and Esashi, 1999; Haga et al., 1998], microgrippers [Kim et al., 1992; Jager et al., 2000a,b]. Another class of mobile microrobots are the so-called smart pills [Zhou, 1989; Uchiyama, 1995; Carts-Powell, 2000], which carry sensors and cameras for diagnostic purposes. Furthermore, MEMS technology is also used for walking robots, through projects such as the European MINIMAN. In addition,

effort has been successful spent on flying [Miki and Shimoyama, 1999; Arai et al., 1995] and swimming [Fukuda et al., 1994] robots.

nanorobots

Nanorobotics encompasses tools, devices, actuators, and sensors at nanometre scale. Nanodevices are usually powered and controlled through magnetic fields and rely upon technologies such silicon-based NEMS (nano-electro-mechanical systems) and carbon structures such as nanotubes. Currently nanorobots are used as nano-scale measurement devices able to measure femtogram masses [Poncharal et al., 1999] or piconewton-order forces through changes of resonant frequencies detected within an electron microscope. Also they can work as thermometers [Gao and Bando, 2002], and they were successfully used for in vitro fecundations.

1.2 Mobile robotics: technological issues and challenges

Notwithstanding mobile robots variety, some interesting interdisciplinary scientific problems can be outlined. In this section, a description of the most important ones will be provided.

- power supply and locomotion
- communication systems
- localization and mapping
- motion planning
- fault detection and isolation
- multi-robot systems coordination
- decision algorithms for guidance

1.2.1 Locomotion schemas

From a mechanical point of view, locomotion schemas are chosen according to the mission the robot must perform and its operational scenario. Namely, locomotion schema choice is deeply influenced by the following elements:

1. kind and extent of operational scenario

2. load capacity and autonomy
3. speed and accuracy in motion
4. different mission-oriented requirements

Kind and extent of operational scenario Robot locomotion system should be adaptive with respect to operational scenario, guaranteeing a fully operational mobility, along with a sufficient uptime ratio. Flexibility and reliability are, thus, crucial to robot locomotion systems, expanding robot range and endurance as long as human intervention becomes less frequent.

Load capacity and autonomy robot deployment in operational scenarios are, usually, meant to replace or support human intervention, in order to increase mission effectiveness and reduce related risks for both humans and the environment. For this reason, load capacity as the allowable payload weight and autonomy (endurance) are two closely coupled elements which deeply influence the choice of robot locomotion system.

Speed and accuracy in motion many robotic applications are bound to certain requirements in terms of speed and accuracy in motion. Speed is closely coupled with endurance and usually affects vehicle autonomy. Accuracy in tracking the assigned position (or the assigned trajectory) is a crucial element for vehicles carrying manipulators (both UGVs an UUVs), while faster robots, such as UAVs, have laxer requirements in terms of motion accuracy.

Different mission-oriented requirements can hardly be described in general terms, anyway, their influence upon locomotion system choice should be highlighted. Moreover, some mission-related devices can expand vehicles capabilities also in terms of mobility for example, robotic arms in UGV can be used in aide of locomotion system in presence of obstacles, thus increasing robot operational range.

1.2.2 SLAM

Simultaneous localization and mapping is a fundamental topic in mobile robotics, arousing a wide research effort in different contexts of mobile robotics [Thrun et al., 1998], [Nieto et al., 2003], [Liu and Thrun, 2003], [Newman et al., 2002], [González-García et al., 2020], [Latif and Saddik, 2019]
Although several robotic application deal with a structured environment, with known topology, the capability of the robot to autonomously

draw a map of its operational scenario is a strong plus for mobile robots, an early example are the robotic tourist guides in [Thrun et al., 1999]. Solving the problem of SLAM requires that a mobile robot, by starting from an unknown location in an unknown environment, can draw a map of its surrounding environment while, at the same time, being able to assess its position within the map. In the process of SLAM, the robot can perform SLAM with knowledge about its kinematic model and by selecting proper markers in the environment[Thrun et al., 2002]. Vehicle position and motion is, thus, evaluated in real-time with respect to visible markers. In order to assess markers position the robot must be equipped with suitable sensors, and, if needed the proper intelligence to select environment features which are suitable as markers. Sensors used in SLAM are different according to the kind of robotic application: computer vision-based ones are common in UAVs, UGVs, and, although with some open technological challenges, UUVs [Ferrera et al., 2019]; sonars are the main tool for UUV mapping, but they are also popular in indoor applications [Buonanno et al., 2011], while laser-based sensors are commonly used for both UGV and UAV. Further sensors such as inertial measurement units [Viquez et al., 2016], GPS, infrared sensor, etc. [Thrun et al., 2002] when available can improve the effectiveness of SLAM algorithms. The most common SLAM approach is the probabilistic approach [Aulinas et al., 2008]. Probabilistic approach appears, indeed, quite a natural choice since the whole process of mapping and localization through sensors observations is deeply affected by stochastic disturbances due to sensors noise. The impact of these phenomena upon SLAM is tackled by such algorithms through stochastic modelling of both disturbance and its effect upon measurement process [Thrun et al., 2002]. In [Jensfelt et al., 2006] the SLAM problem is solved through Kalman filters. Kalman filter-based SLAM solutions are, however, curbed by their computational complexity, which depends on the square of the number of reference points. Complexity hampers the possibility to use Kalman filters in a real-time context, at least when using computational platforms whose weight, power consumption, and cost are of interest for mobile robotics applications. Nonetheless some effective KF-based solutions are proposed [Rodriguez-Losada et al., 2005] in order to improve KF-based solution computational efficiency, usually by carefully selecting a small subset of reference points for real-time computations, while the whole map is updated at a lower priority and frequency [Thrun et al., 2002] than motion control.

Another common probabilistic approach for SLAM are particle filters. Particle filters are Monte Carlo methods which, in contrast to Kalman filter-based solutions, can tackle strong nonlinearities in sensors and non-Gaussian noises [Hahnel et al., 2003], notably, a particle filter is used in ROS AMCL package which is used to perform LIDAR-based SLAM for experimental validation of the results of this thesis, in section 5.
Given SLAM complexity, several techniques can be found in a very vast literature. A grid occupation technique is proposed in [Elfes, 1987], while an *expectancy minimization* method is used in [Thrun et al., 1998], probabilistic [Gutmann and Konolige, 1999] topologic [Choset and Nagatani, 2001] approaches are also examples of widely used solutions to SLAM problem.

1.2.3 Trajectory planning

Another fundamental task in mobile robotics is trajectory planning, i.e. giving both a path and a timing law to the vehicle [Kant and Zucker, 1986; Steer, 1989; Aydin and Temeltas, 2002; Brezak and Petrović, 2011]. Path planning algorithms are used in order to find a path that the robot can safely use to reach its target taking into account known obstacles and safety prescriptions. Path optimality and smoothness are strongly advisable although they significantly increase the required computational effort of trajectory planning. Variational approach is probably the most straightforward and exact approach in search for an optimal trajectory, anyway, its complexity hampers the possibility to use it for solving less than trivial problems. Several papers deal with trajectory planning through direct and indirect numerical methods in a non-linear optimization context [Geiger et al., 2008] [Chang and Yamamoto, 2009]. In indirect methods, optimality is pursued through the a series of necessary conditions derived as a consequence of Pontryagin maximum principle. Direct methods, instead, are based upon a discrete representation of inputs and state variables leading to a nonlinear programming problem whose complexity, however, quickly makes it computationally intractable for real-time applications. Another popular approach makes use of attraction potential fields [Pacheco et al., 2009] [Gavrilova et al., 2008], in this kind of solution, operational scenario is associated to a potential field, where obstacles are given a repulsive force, while destination radiates an attraction force. Gradient descent is then used for optimal trajectory planning. Local minima can,

however, entrap the robot, therefore hampering the possibility to use this method. The local minima problem has been tackled by [Kim and Khosla, 1992] using harmonic potential functions, in order to generate potential fields with a single minimum. Harmonic function method has been successfully deployed in a series of real-world scenario [Fahimi et al., 2003] and [Mattei and Scordamaglia, 2010a]. Another family of solutions relies upon the representation of operational scenario through a graph, making use of graph optimization techniques [Mattei and Blasi, 2010; Mattei and Scordamaglia, 2010b], while constraints are taken into account during graph exploration and/or construction.
Another vast family of trajectory planning algorithms consists of *nature-inspired* ones, such as *ant colony optimization algorithms* for multi-robot systems. Also, several hybrid approaches based upon soft-computing and genetic algorithms can find optimal or suboptimal trajectories for mobile robots [Yokoyama and Suzuki, 2005] [Tian and Collins, 2004].

1.2.4 Guidance, navigation and control algorithms

Obstacle avoidance capabilities are a fundamental skill for a successful mission involving mobile robots. Motion planning can be divided [Fujimura, 2012] in a *static* and *dynamic* part. In static motion planning obstacles are known in advance, while, in dynamic planning, obstacles, both fixed and moving ones, are not known a priori. Accordingly, operational scenarios can be classified as static, when the environment is a priori known, and dynamic, when the scenario is only partially known in advance. Collision avoidance algorithms must be implemented in dynamic environments, given that an optimal, collision-free, planning requires a complete knowledge of the operational scenario. Hence, the robot must be capable of a prompt and proper reaction to sudden changes in its surrounding environment or to the presence of previously unknown obstacles. Local optimization algorithms are the only possible solution to this problem, they can make use of data gathered by vehicle sensors and partial maps obtained through SLAM procedures. Reactive algorithms are those which can accommodate unknown obstacles, both fixed and mobile, an example is the reactive control schema using potential fields described in [Khatib, 1986]. Potential fields method is notably efficient for usage in reactive control schemas, given its computational efficiency, although the previously mentioned local minima problem could hamper algorithm convergence. Another reactive control schema is *vector field histogram*, as described in [Ulrich and Borenstein,

2000] which operates a greedy choice of robot motion direction by considering density of obstacles. Although efficient, this method introduces some drawbacks, such as the risk of unduly pruning of possible paths. Other reactive control methods are the *curvature velocity method* [Simmons, 1996] and the *dynamic window approach* [Fox et al., 1997] which can arrange both kinematic and dynamic constraints of the vehicle, while minimizing a suitable cost index. These kinds of approaches can be merged with a planning algorithm to improve the overall optimality of the chosen trajectory (*model based dynamic window approach* [Fox et al., 1998]). The *elastic bands method* [Quinlan and Khatib, 1993] is another method which alters a pre-calculated trajectory found in a planning step using artificial forces depending on the shape of dynamic obstacles. This kind of approach is used in [Sekhavat and Chyba, 1999] which also offers a solution to the problem of kinematic and dynamic constraints.
The problem of reactive control should be considered in scenarios where mobile robot path is predetermined by its mission. In such circumstance the only available degree of freedom is the speed of the robot, which is adjusted to take into account the presence of obstacles. Autonomous navigation in terms of the sum of a path planning problem in a static environment and the problem of finding a suitable timing law is tackled in [Griswold and Eem, 1990], while the synthesis of a collision-free timing law and its optimization in terms of overall time length is addressed, for example, in [Dahl and Nielsen, 1990].
Finally, the problem of coexistence of unmanned and manned vehicles should be tackled, trying to make the behaviour of the autonomous vehicle predictable to nearby human actors, a examples are [Casalino et al., 2009] and [D'Amato et al., 2021b] where the possibility to embed the *Convention on the International Regulations for Preventing Collisions at Sea* (COLREGs) collision avoidance rules into trajectory planning of a USVs is considered.

1.2.5 Fault detection and isolation

Mobile robots operational scenario can make human intervention on a faulty robot hard or even dangerous. Also a fault can bring to the loss of the vehicle or put surrounding environment or infrastructures at risk. For example, UAVs fault in dense areas can pose a risk to safety, especially in densely populated areas, UUVs operating in oil

production can damage plants if lost while UGV can be hard to recover during SAR operations. Last decades have witnessed a significant research effort in developing proper *fault detection and isolation* (FDI) techniques for mobile robots, often enhanced with proper reconfiguration capabilities with the aim to allow the mobile robot to fulfil its mission even in presence of faults. In [Michaelson and Jiang, 2000] a cooperative FDI schema for a robotic swarm is proposed, in such scenario faults are tackled by reallocating tasks of the faulty vehicle to fellow swarm members, hence mitigating fault impact upon swarm performances. [Scheding et al., 1998] provides a review of redundancy-based isolation techniques. An FDI schema (often labelled as SFDI, *simultaneous fault detection and isolation* schema) is efficient if it can detect faults quickly enough to prevent performance degradation through reconfiguration. The main tool in SFDI is redundancy, which can be used in one of its two possible forms, namely *hardware redundancy* and *analytical* or *software redundancy*. Hardware redundancy is the golden standard for aeronautics, using a set of redundant sensors to validate measurement data [Zhong et al., 2019], namely triple or more hardware redundancies which can strongly enhance safety and reliability [Goupil, 2011] through voting-based solutions. This schema is quite convenient from the point of view of implementation and management, but it comes with the drawback of increased weight, power consumption, and costs. Analytical redundancy is commonly realized through the use of state estimators [Gertler, 1998]. In literature, various estimator schemas for FDI have been used, most of them underpinning the generation and analysis of the so-called *residuals*, i.e. the difference between observed and calculated measurements according to a proper prediction algorithm. In *dedicated observer scheme* (DOS) [Clark et al., 1975], a bank of observers, driven by a different set of measurements, is utilized. In presence of a fault, one of the observers will return inaccurate estimates, leading to fault isolation. In *generalized observer scheme* (GOS) [J. Wünnenberg and P. M. Frank, 1987] each observer is fed by all measurements but one. In a faulty condition, just a single estimate will be accurate. Another case of DOS is the *simplified detection scheme* [Clark, 1978] which employs a single observer, driven by only one measurement. Thus, if another measurement is faulty, the relevant residual is non-zero. Obviously, if the sensor used by the observer is faulty its residuals will differ from zero and no information can be derived.

Several solutions try replacing hardware redundancy-based techniques with analytical redundancy-based ones [Rudin et al., 2014; Berdjag et al., 2012; Rosa and Silvestre, 2013; Notaro et al., 2014; Ariola et al., 2013; Goupil, 2011; Alwi and Edwards, 2006; Varga et al., 2014; Mattei et al., 2016, 2005]. A Kalman filtering [Kalman, 1960] approach is proposed in [Berdjag et al., 2012] to protect sensors used in flight control law calculation from oscillatory failures. An FDI method that relying upon set-valued observers is proposed in [Rosa and Silvestre, 2013] dealing with uncertain linear parameter-varying systems. The proposed technique uses a bank of filters without the need of thresholds to state a fault, on the contrary of residual-based architectures. Some Kalman-filter and sliding mode-based techniques are [Ariola et al., 2013; Alwi and Edwards, 2006]. Other robust and nonlinear techniques, such as H_∞ filtering, can improve robustness to faults of physically redundant schemes [Mattei et al., 2004, 2005]. Also particle filters [Crassidis et al., 2007; Cheng and Crassidis, 2004; Oshman and Carmi, 2006; Carmi and Oshman, 2009; Xue-Yuan Jiang and Guang-Fu Ma, 2005; Chagas and Waldmann, 2012] are suitable to the role of observers in FDI schemas. A PF-based fault detection scheme is used by [Kadirkamanathan et al., 2000], while [Kadirkamanathan et al., 2002] stresses PF advantages in carrying out FDI for non-linear systems. PF-based FDI is discussed in [Dearden and Clancy, 2002]. FDI problem for the autonomous integrity monitoring of a GPS receiver is investigated in [Wang et al., 2018]. Two alternative PF-based FDI techniques and a comparison with KF ones are proposed in [Wei et al., 2009]. The FDI problem for a wheeled mobile robot is addressed in [Duan et al., 2006], principal component analysis driving a Gaussian Mixture Model is investigated in [Yu, 2012].

Further examples of FDI in robotics are [Washington, 2000] which deals with rupture of wheels in an UGV, [Nardi et al., 2021] where the problem of FDI for the IMU sensors of a space rover is tackled and [D'Amato et al., 2021a] which proposes a fault-tolerant attitude estimation for an UUV.

Finally, in [Albore et al., 2021] fault tolerance is achieved by the means of a decision layer which can perform skill-based mission reconfigurations.

1.2.6 Coordinated control

Recently, multi-robot systems have drawn an increasing interest both in research and industrial field. *Multi robot systems* (MRSes) are, in

short, cooperative systems of autonomous mobile robots. MRSes show several advantages upon single or non-cooperative robotic applications in terms of time requirements and results achieved while executing their mission. Furthermore some missions cannot be accomplished by single robots but, also, MRSes can benefit from their distributed sensing and actuation capabilities. Another upside of MRSes lies in their lower cost in comparison to single, high-performance, vehicles. Heterogeneous multi-robot systems can be classified according to the kinds of vehicles they encompass and their mission. In [Dudek et al., 1996] multi-agent robotic systems are classified according to: communication network; swarm size; members autonomy; swarm composition; reconfiguration capabilities. In [Gerkey and Matarić, 2004] coordination technique is the key for MRSes classification.
Several fields of application can benefit from MRS usage, for example cooperative robots can explore an unknown scenario performing multi-robot SLAM. In such scenario using an MRS can increase mapping speed and accuracy. In [Rekleitis et al., 2001] a two-robots team perform a cooperative SLAM by reciprocally detecting and correcting robots position. A cooperative multi-vehicle SLAM with constrained channels in developed in [Burgard et al., 2005] which makes use of three heterogeneous robots. Several military-oriented multi-robot systems have been studied [Balch and Arkin, 1998] with a focus upon reconnaissance, surveillance, and high-risk missions [Chaimowicz et al., 2004]. MRS effectiveness strongly depends on the possibility for robots to share information about their state and strategy, therefore communications and control architecture are vital element of MRS. In [Fukuda et al., 1989] a bio-inspired hierarchic distributed control system is proposed (CEBOT) trying bringing biologic cells self-organizing capabilities into an MRS. This level of flexibility where robots can be seamlessly added and removed from the team also adds a further layer of fault-tolerance given the possibility to replace some robot with minimum impact upon MRS operations. In [Asama et al., 1989] the ACTRESS architecture is proposed for heterogeneous multi-robot system communication management. Also [Orebäck and Christensen, 2003] compares some middlewares for multi-robot systems. Notably the popular ROS (robotic operating system) open-source framework for robotics offer the possibility to use robots and even control software entities as logical nodes, in a publisher-subscriber framework, to be shared in a MRS. MRS control frameworks could, anyway, be classified into two main

categories, namely centralized and decentralized ones. Decentralized approaches implement the idea that robot should rely upon their sensors to gather the relevant information about their own state and the environment. Each vehicle has its own motion strategy and communication with every vehicle of the fleet, apart from closer ones, is not needed. Conversely, centralized systems are equipped with a central unit whose role includes the coordination of other vehicles and information collection and dispatch, ensuring mission accomplishment. In such architecture, the central unit is a hotspot, both in terms of required capabilities and impact of faults, moreover it must be able to communicate with every element of the team, at least in a relayed communication architecture. Centralized architectures are usually more efficient in terms of resources usage, benefiting from the possibility of a global optimization of the strategies and task allocation of single vehicles. Also, faults on non-central units can be better handled by exploiting centralized task allocation [Arrichiello, 2006]. Although decentralized systems are less efficient by these points of view, they have no hotspots and, also, require less expensive vehicles. Finally, mixed centralized/decentralized solutions are proposed [Tanner et al., 2004] with the aim to combine advantages offered by both kinds of control architectures.

2

Motion planning for a mobile terrestrial vehicle

In this chapter a brief overview of motion planning for terrestrial vehicles is proposed.
Motion planning solutions can be divided in many ways, among others, between path planning and trajectory planning solutions. While path planning consists in the search of a sequence of configurations which the vehicle can cross to fulfil its mission, trajectory planning binds these configurations to specific time instants [Valero et al., 2006].
Given the vector of coordinates $q = \begin{bmatrix} x\ y\ \theta \end{bmatrix}$, with θ the heading, and the parametric timing law $s(t)$ the nonholonomic constraints of a robot moving on a plane can be expressed as:

$$F(q)\frac{dq}{ds}\frac{ds}{dt} = 0 \tag{2.1}$$

Paths which are admissible from a geometrical point of view can be defined as the solutions [Siciliano et al., 2009] of the non-linear system:

$$\frac{dq}{ds} = G(q)i(s)\frac{s(t)}{dt} \tag{2.2}$$

with $i(t)$ the sequence of *geometric inputs* coupled with the timing law $s(t)$.
The two problems (finding a path and a timing law) can be either tackled at the same time or in a *decoupled* approach.
Trajectory planning requires a mathematical model of the vehicle, in this thesis a kinematic model for a differentially driven vehicle subject to disturbance phenomena is found in chapter 3.2. An interesting comparison of the effects of robot mathematical choice upon different planning techniques is carried out in [Polack et al., 2017].

2.1 Classifications of planning methods

In [Kumar et al., 2007] a classification of planning methods is proposed.
A first dichotomy is highlighted in *implicit* versus *explicit* planning methods. Explicit methods are those which perform the explicit computation of the trajectory in a certain space (either controller or configuration space). Implicit methods, instead, do not pre-compute motion (or the sequence of inputs) but define a set of rules which are followed by the vehicle during its motion.
Explicit methods can be further divided between *discrete* and *continuous* ones, the first class focuses on finding a sequence of discrete configurations connecting starting and ending points, abiding by constraints, and bypassing obstacles. Conversely, continuous methods define an open-loop reference trajectory and related control sequences for the vehicle. Implicit methods are, in general, related to closed-loop control strategies, which define a tracking error from a nominal trajectory which must be controlled to zero through proper closed-loop solutions.

2.2 Implicit methods

The main class of implicit methods is the potential fields one.

2.2.1 Potential fields methods

The possibility to associate the operational scenario with a potential field, where destination point is associated to an attractive potential and obstacles and forbidden areas are characterized by repulsive forces has long been used. This class of solutions appears attractive given the possibility to exploit fast and reliable methods of optimization on fields such as gradient descent. For instance, [Khatib, 1986] provides a potential method suitable for both manipulators in direct control (rather than in joint space) and mobile robots, with real-time collision avoidance capabilities.

In [Chang and Yamamoto, 2009] a hybrid planner based upon potential methods and Voronoi Diagrams is used to perform simultaneous navigation and mapping. The main drawback of gradient-based methods, the possibility of the robot ending up trapped in a local minimum,

is tackled through the usage of Voronoi diagrams to set proper subgoals. In [Pacheco et al., 2009] the problem of trajectory planning of a WMR is solved by using attraction potential fields combined with model predictive control for trajectory tracking. [Rao et al., 1991] presents a solution framework which is based upon retraction of free space onto Voronoi diagram. This method can handle, at the same time, terrain mapping and the problem of visiting a series of subgoals. Notably, such method can encompass any graph structure which guarantees finiteness, connectivity, and local constructability.

In [Masehian et al., 2003] a mixed potential-field/Voronoi graph is proposed. The method is complete, i.e., it can result in stating that no suitable trajectory exists and is suitable for online usage.

Another mean to tackle the problem of local minima is presented in [Kim and Khosla, 1992] where an artificial potential is built using harmonic functions, resulting in a field with no local minima even in a cluttered environment, panel method is used to represent arbitrarily shaped obstacles allowing a real-time control of the vehicle.

In [Janabi-Sharifi and Vinke, 1993] a potential field approach is used; however, the local minima problem is tackled through a heuristic strategy, namely a simulated annealing method which consists in escaping from local minima by choosing random neighbours with a probability inversely proportional to the number of unsuccessful attempts to escape from a minimum.

2.3 Explicit methods

One of the first solution to path planning consists in properly interpolating the initial and final values of vehicle generalized coordinates, through Cartesian polynomials [Siciliano et al., 2009] which can guarantee path smoothness.
[Rao et al., 1993] provides a survey of early, non-heuristic, solutions.

2.3.1 Curvature-based methods

[Steer, 1989] is focused upon the problem of altering the speed and the steer angle of a path curvature limited robot-vehicle in order to attain convenient changes in its position and orientation. The proposed solution consists in identifying a series of functions (mainly Gaussian envelopes) modulating robot steering angles.

A smooth trajectory planning algorithm is proposed in [Aydin and Temeltas, 2002], it uses the classic visibility graph approach to obtain a shortest path which is then modified to take into account robot dynamic constraints, according to robot natural behaviour, i.e., the behaviour of the vehicle while tracking a non-smooth path.

Smoothness in trajectories is also advisable in order to simplify trajectory tracking, requiring, for instance, less commands. Such advantages are useful in real-time contexts and are discussed in [Aydin and Temeltas, 2002].

Smoothness is the focus of [Bruyninckx and Reynaerts, 1997] which decomposes the path into a series of Pythagorean hodograph curves conjoining the inflection points of the path. Such solution is favoured over other splines since it offers advantages, given the amount of mathematical closed solutions of energy and curvature parametrization offered by this class of curves.

In [Perez et al., 2000] a Nomad 200 robot is driven through a series of steering commands calculated from a decomposition of the path into a series of simple curves, whose discontinuities are eliminated through geometrical arguments.

A series of grossly selected main points, deemed subgoals, for a car-like vehicle, is selected in [Zhang and Knoll, 1995], subgoals are then connected using b-splines, obtained by solving an optimization problem in terms of cost function of time, length, and curvature. The optimization procedure can tackle the presence of local minima. Car-like vehicles are provided with continuous curvature paths in [Fraichard and Ahuactzin, 2001] which proposes a verifiable complete planner through topological considerations.

In [Agarwal et al., 2002] an $O(n^2 \log n)$ algorithm for the search of an optimal shortest-path in a convex space is proposed, the given solution is curvature-constrained and a series of properties of curvature-constrained solutions is found. [Wu et al., 2000] proposes a time-optimal solution in absence of obstacles for a WMR. This solution is, namely, a decouple approach where the path planning is taken in a purely geometric fashion while still taking into account kinematic constraints, resulting in a succession of circular arcs and straight lines, followed by the generation of an optimal velocity profile considering dynamic constraints. [Villagra et al., 2012] designs a global path planner aimed at guaranteeing passengers comfort in an automatic public transport vehicle. The planner uses bounded continuous curvature and bounded

curvature derivative to ensure smooth driving. The approach is enriched with semantic information provided by the path planner to find a continuous velocity profile with bounded accelerations and jerks. Finally, in [Brezak and Petrović, 2011] a time-optimal framework is developed for any smooth trajectory, seen as an input to a decoupled trajectory planner which takes into account dynamic constraints.
In [Lumelsky and Stepanov, 1986] the case of a robot moving in an unstructured environment is studied. The proposed solution relies upon local perception of environment still guaranteeing reaching of a target through suboptimal paths. Also, the method allows the dynamic integration of further information sources.

[Khatib et al., 1997] combines planning and reactive control for a car-like vehicle. A reachable set is defined as the starting point of a feasible elastic band-based trajectory. Found trajectories are smoothed through Bezier curves whose minimum curvature radius is chosen according to kinematic constraints.

2.3.2 Graph-based and grid-based methods

The possibility to use graphs as a tool for planning is attractive both for the vast amount of available algorithm and its effectiveness in providing a natural representation to planning in terms of a finite set of configurations connected by possible manoeuvres.
Kavraki et al. [1996] introduces a two-phases method, which starts with a learning phase when a probabilistic roadmap is built and represented as a graph of collision-free configurations. Paths are, then, found through local planning. The second phase is a query phase consisting of extracting the desired path.

A vast class of methods relies upon the concept of visibility graphs and its variations (e.g., extended visibility graphs, natural visibility graphs, etc.). The core idea lies in reducing the cardinality of state-space following a simple consideration: shortest path between two points passes through the vertices of obstacles, an example is reported in figure 2.1. VGs provide non-smooth paths which usually require a proper smoothing procedure.

In [Krishnaswamy and Newman, 1992] a visibility graph is used to drive the robot along the edges of a workspace containing convex polygonal obstacles. [Oommen et al., 1987] instead constructs a visibility graph incrementally, using information gathered on already crossed

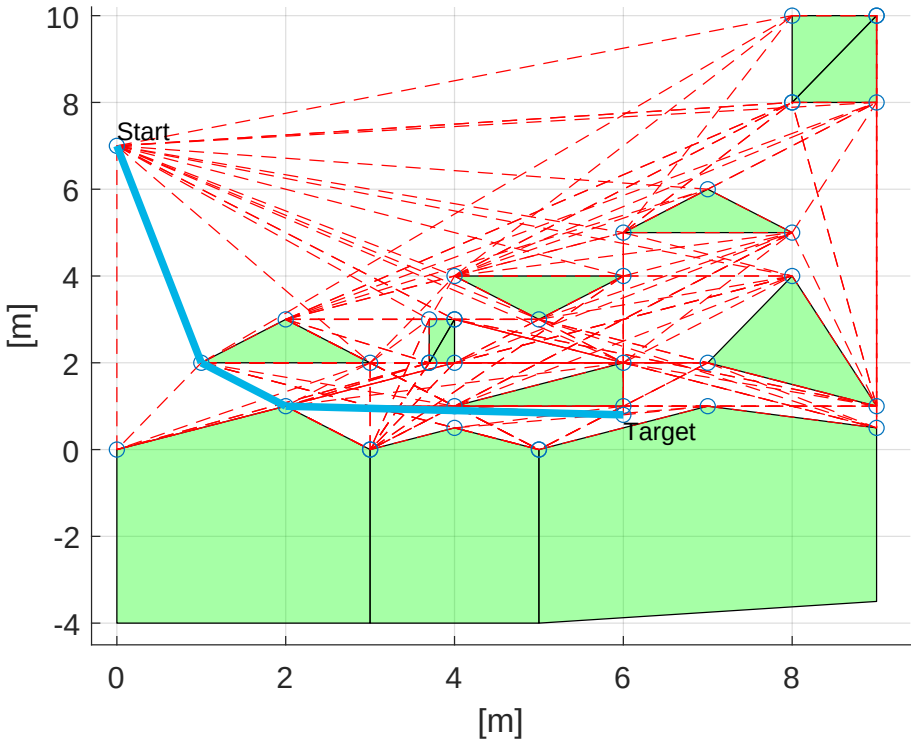


Fig. 2.1. An example of visibility graph in a cluttered environment. The optimal path from starting to ending point is highlighted by the bold blue line.

paths, then a global optimization is performed, leading to a global optimum for subsequent explorations.
An interesting classification of graph-based solutions is proposed in [Liu et al., 2017], identifying two main families of solutions, namely *planning by construction* (PBC) and *planning by modification* (PBM) methods. In the first class of solutions, trajectories are incrementally built until the goal is not reached. These methods usually produce non-smooth trajectories which should be smoothed by proper means. In [Mattei and Scordamaglia, 2010c] a series of quadratic programming problems is solved to create the arcs of a graph whose nodes are discretized positions in the operational scenarios and whose arcs represents possible trajectory segments.

A wide class of PBC methods are *rapidly-exploring random trees*, these methods create spanning trees in the operational scenario giving

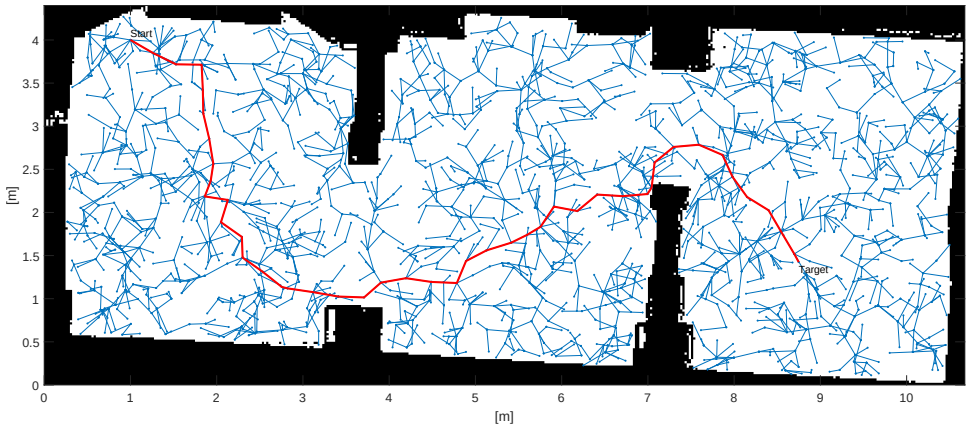


Fig. 2.2. An example of rapidly-exploring random tree in a cluttered environment. The optimal path from starting to ending point is highlighted by the bold red line.

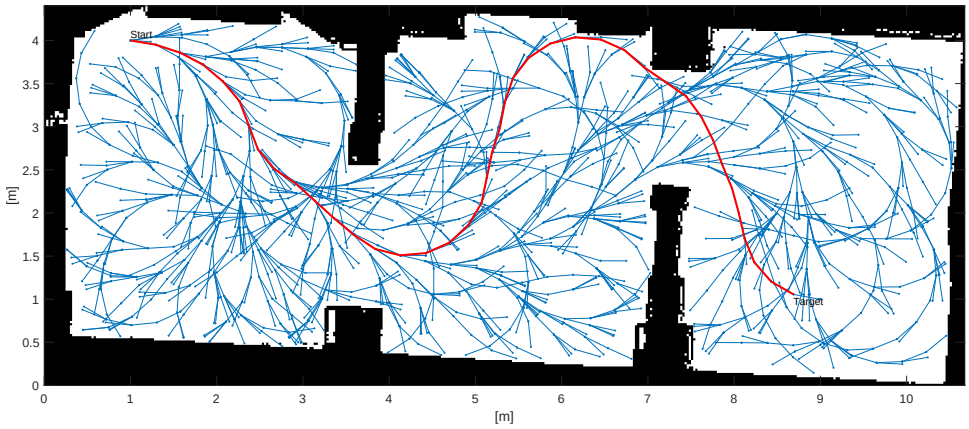


Fig. 2.3. An example of planning on RRT of Dubin curves in a cluttered environment. The optimal path from starting to ending point is highlighted by the bold red line, minimum curvature radius has been set to 0.9 m.

precedence to unexplored areas which are filled with new branches of the tree (see 2.2 for an example), in [Kuffner and LaValle, 2000] this method, originated from [LaValle et al., 1998], is used starting simultaneously from starting and ending nodes, thus halving the computation time. RRTs are also used in [Hsu et al., 2002] on a probabilistic roadmap of sampled state versus time points, called milestones, connected by short admissible trajectories.
In [Sisbot et al., 2010] a human-aware planning is introduced, meaning that trajectory planning explicitly takes into account the presence of

humans and their behaviour with respect to the presence of the robot. Planning by modification methods starts from a geometrical path which is modified until constraints are satisfied. An example is [Liu et al., 2017], which proposes an optimization-based trajectory smoothing method for robot motion planning, which is characterized by convexity and usage of quadratic programming methods. In [Howard et al., 2010] a model-predictive approach on an A* path search algorithm is proposed. Another tool for creating smooth trajectories are Dubin curves (see fig. 2.3 for an example): the shortest curves connecting two points with a minimum turning radius. Dubin curves can be combined in different means to create smooth trajectories which can even handle kinematic constraints.
In [Chen et al., 2008] a graph represents the Voronoi points of operational space. It uses a hybrid control approach using HQL. [Choset et al., 2000] proposes an algorithm which allows the navigation of a circular robot in an unknown environment through the iterative visit of the vertices of a Voronoi diagram. Moreover, it includes a method for incremental construction of a hierarchical generalized Voronoi diagram (HGVG) and a method to preserve connectivity in high-dimensional space.

Finally, visibility graphs are used in [Ghosh et al., 2008] offering an effective solution for the complete exploration of an environment, with the aim of perform search and rescue operations. Outer boundary and the boundaries of obstacles are approximated as piecewise linear, allowing the construction of a proper VG for the environment.

2.3.3 Heuristic methods

In seek of online planning capabilities, a variety of heuristic algorithms has been developed. Such algorithms do not guarantee to find a solution, but when they do, they outperform classic motion planning approaches in terms of speed. Popular metaheuristic approaches include genetic algorithms and simulated annealing. An example of the first approach can be found in [Ahuactzin et al., 1991], such approach consists into random growing of candidate solutions composed of elementary moves, ranked according to a fitness function (i.e., a heuristic of their suitability as best solution). Genetic algorithms can benefit from computational parallelism which comes with modern computation platforms. [Eldershaw and Cameron, 1999] formulates the path planning problem in terms of

a global optimization problem solved through a genetic approach, using a spherical approximation of forbidden regions of configurations space. Another example of a genetic solution is [Tian and Collins, 2004], it exploits a polynomial approximation of the time history of the trajectory in task space which is based on Hermite cubic interpolation. Points to be interpolated are determined through a genetic algorithm leading to an optimal trajectory.

In [Masehian and Amin-Naseri, 2008] a taboo search is performed. Such class of navigation algorithms encompasses a first phase which is similar to local optimization algorithms, but circumvents the problem of local minima by including a phase in which the operational space is explored even by performing moves which actually worsen the current optimal solution. Such moves are called taboo moves and are stored in a list so to avoid endless loops. Subsequent iterations allow to obtain a map of most promising areas to scan for a global optimum.

[Madhevan et al., 2017] uses different trajectory planning methods for a fire-fighting mobile robots, among them a fuzzy logic method.

2.3.4 Other sampling methods

Further sampling methods operate in the control space (i.e., the space of possible control actions) however such methods can become intractable when manoeuvres become more complex, leading to a high dimensional control space which makes state-space solutions easier to compute. [Howard et al., 2008] presents an algorithm for state space sampling relying upon a model-based trajectory generation approach in robot surroundings.
Another control space solution is proposed in [Ramirez and Zeghloul, 2000], where obstacles are mapped as linear constraints in velocity space, so that a convex polyhedral solution space is obtained and, thus, convex optimization is possible.

2.3.5 Set-based methods

The idea of considering properly defined sets of possible system states in planning arises from the widely used MPC control framework. The high performance guaranteed by such methods attracted interest upon them in the motion planning of aerial vehicles (see for example [Kuwata et al., 2005]), but also in relation with terrestrial vehicles.
In [Abate and El Ghaoui, 2004] model predictive control is performed

through convex programming to ensure a robust path planning within ellipsoidal sets. MPC is also used in [Gao et al., 2014], which realizes a robust planner based upon a nonlinear force model. [Falcone et al., 2007] proposes two MPC approaches, which, at each sampling step, compute the steering angle of an autonomous vehicle over a finite horizon. In the first approach a nonlinear MPC is used, while in the second one a linearization is performed each time. Although quite expensive from a computational point of view this approach is proved effective on an icy road at a sustained velocity. Another MPC solution is proposed in [Di Cairano et al., 2016], forcing the vehicle to remain within a polyhedral robust control-invariant region.
[Kayacan et al., 2016] proposes a linear model predictive control which allows precise trajectory tracking for a tractor-trailer vehicle. A model-based approach is used in [Hegedüs et al., 2017] to generate the reference signal which allows a local trajectory planning, taking into account state and actuation constraints for highly automated road vehicles. In [Cariou et al., 2009] an agricultural robot is driven on a trajectory which is adjusted on a prediction horizon, so to take into account delays in actuation and skid and slip phenomena.

[Schouwenaars et al., 2004] defines polyhedral sets of safe states through mixed-integer linear programming for a single vehicle moving in a cluttered environment which is only known within a certain detection radius around the vehicle. Hard terminal constraints on a receding horizon ensure MILP feasibility and, at each iteration, the vehicle never leaves a safe set of states where it can remain for an indefinite period of time.

In [Horn and Janschek, 2010] a dynamic window approach is proposed, using set theoretic methods which can tackle noisy measurements. The proposed solution performs three operations through sets: the estimation of robot state until standstill, the calculation of obstacle map and the computation of admissible velocities.

Another class of set-based methods relies upon the definition of invariably safe sets, which are regions where the vehicle can safely remain for an indefinite timespan, in [Pek and Althoff, 2018] an under-approximation of invariably safe sets is efficiently computed for a fleet of vehicles. An automated highway driving case study is tackled in [Danielson et al., 2020b] through a set-based approach which can tackle model uncertainty and disturbance. Notably it introduces a bound in time re-

quired to move among invariant sets, so to consider moving obstacles.

A set Λ in state-space is defined as invariant (or positively invariant) for the system $x_{k+1} = \Phi\, x_k$ if $x_0 \in \Lambda \implies x_k \in \Lambda \quad \forall k > 0$ (a longer discussion of invariance properties is carried out in chapter 4). The possibility to use such sets in trajectory planning has been drawing increasing attention given its robustness and its possibility to rely upon coarse sampling. A quite interesting category of set-based works takes into account closed-loop dynamics see for example Danielson et al. [2020a] which provides an efficient framework which considers closed loop dynamics of the system. In [Majumdar and Tedrake, 2013] a library of finite-time invariant regions (*funnel*s) is calculated, allowing to compose, on-line, trajectories by properly combining library elements. [Berntorp et al., 2017] develops an algorithm for safe lane changes through positively invariant sets, which can guarantee collision-free closed-loop trajectory tracking. The system moves through a series of equilibrium points towards the desired configuration, without exiting from a proper succession of invariant sets.

[Nardi et al., 2018] proposes a planning by construction procedure which takes into account closed loop dynamics and disturbance phenomena arising from skid and slip of the tracks of a differentially driven tracked mobile robot. Moreover, [Scordamaglia et al., 2019a,b] tackle model uncertainty through the evaluation of proper invariant sets in a polytopic representation of closed loop system.

3

Mathematical modelling of a skid-steer tracked mobile robot

In this chapter the trajectory tracking error dynamics required for carrying out the feasible trajectory planning procedure proposed in chapter 4 is analysed. Skid-steered robot motion analysis is performed, along with modelling of the networked control system.

The following assumptions are made:

1. a rigid vehicle is moving on a horizontal plane;
2. the reference frame **E** is inertial;
3. nonholonomic kinematics equations of differentially driven tracked mobile robot hold.

3.1 Overview

This work proposes a trajectory planning algorithm in a-priori known static environment for the autonomous skid-steered tracked mobile robot Jaguar V4 by Dr.Robot that can be seen in Fig. 5.3. This experimental setup poses some challenging facets from trajectory tracking control design point of view. In skid-steering locomotion schema, steering action is obtained by setting different velocities for left and right tracks of the vehicle[Martínez et al., 2005]. Such schema is characterized by simplicity and a robust mechanical structure [Wu et al., 2013] and, among others, zero radius-turning capabilities. Moreover, it is a reliable mobility solution which is suitable for all-terrain missions [Krishnamurthy, 2008]. However, this class of UGV is affected by several problems from motion control perspective due to non-negligible skidding and slipping phenomena [Mandow et al., 2007]. The simple Coulomb model for friction is used, for example in [Petrov et al., 2000],

but for such kind of systems, several *model-free* approaches for control design are proposed in literature [Fliess and Join, 2013; Depraetere et al., 2014; Zhang, 2019]. [Yang et al., 2019] proposes an intelligent-PID which can handle the complex wheel-soil interactions for a wheeled vehicle. In contrast, *model-based* techniques are the main tool for constrained control design problem [Afonso et al., 2013; Pivtoraiko et al., 2008; Kamyar and Taheri, 2014; Wolek and Woolsey, 2017].

The two kinds of approach are compared in [Depraetere et al., 2014; Kayacan et al., 2018].

In the given setup, the trajectory tracking control is performed on a remote computer connected to the mobile robot through a communication network, see fig. 5.4. Integration of networks in control systems is a crucial topic in the field, opening the doors to remote control, coordination of multi-agent systems and providing complex control capabilities even to the simplest platforms. Solutions where feedback control loop is closed through a communication network are widely adopted in mobile robotics[Arcara and Melchiorri, 2002; Caruntu et al., 2013; Okamura, 2004; Samad et al., 2007; Zhang et al., 2001; Scordamaglia et al., 2019a]. Networked control systems present substantial benefits such as, for example, higher system resource employment, maintainability and testability, a flexible control architecture design, reduction of cost and simplified deployment [Bemporad et al., 2010b]. Nevertheless, developing a control system using a communication network is a non-trivial task given that a communication network induces non negligible stochastic effects i.e., delays, jitter, and packet loss [Antsaklis and Baillieul, 2004; Bemporad et al., 2010a; Hespanha et al., 2007]. These effects make the networked mobile robot be an uncertain non-linear system subject to external disturbances.

In the field of mobile robotics is usual to work with kinematic models [Asensio and Montano, 2002] to obtain stable motion control while satisfying even complex, dynamic constraints [Khatib et al., 1997] and complex obstacle-avoidance goals [Ramirez and Zeghloul, 2000]. Alongside, dynamic models are used to take into account inner phenomena of vehicles. In [Lhomme-Desages et al., 2007] performances for a fast four wheels skid-steer vehicle are improved through the use of a non-linear model of wheel slippage, to be tackled by a lower-level controller. In [Zong et al., 2006] a sliding mode observer is used to explicitly estimate slip parameters relying upon the kinematics model of a skid-steering vehicle and trajectory measurements. The approach is compared to use

of both EKF and direct mathematical inversion of the kinematic equations. In [Motte and Campion, 2000] a control law is developed such that the robot ends up in a slow manifold where an ideal dynamic model (i.e., single point contact of tyre and soil) can be used, while in [Peng et al., 2004] singular perturbation theory allows the replacement of the actual wheel subsystem model with its quasi-steady state, for model reduction. Furthermore, [Espinosa et al., 1998; Fierro and Lewis, 1997] express control laws in terms of set points to the servos, while [Topalov et al., 1998] provides a model in which servos torques (useful for several classes of electric motors) are the input vector. Analogously. [Yun and Yamamoto, 1997] ties servos torque with vehicle coordinates and turned angles of wheels. Anyway, the dynamic approach is characterized by higher complexity both in terms of model identification and associated computational effort [Michalek et al., 2009]. The kinematic representation is attractive, from a practical point of view, when a sufficiently precise regulation of track speeds is available i.e., motor controllers are driven by controllers with adequately large bandwidths. Interestingly, control techniques can be extended with low effort to every differentially driven vehicle.

In skid-steered mobile robots, the steering manoeuvre depends on the difference of the velocities of tracks. This leads to left and right tracks to rotate around different *instantaneous centres of rotation* (ICR) [Pentzer et al., 2014] causing skidding or slipping. Both phenomena are non-negligible and substantially deteriorate motion accuracy of the tracked mobile robot, which is controlled under the assumption of nonholonomically constrained motion, whose kinematic laws cannot be expressed in terms of the sole generalized coordinates[Zohar et al., 2011].

A variety of works describes skid and slip phenomena in terms of ICR movements, see [Martínez et al., 2005], [Mandow et al., 2007] and [Wu et al., 2013] for example. In [Corradini et al., 1999] and [Lenain et al., 2006] skid and slip phenomena are modelled in terms of disturbances on control inputs. Finally, [Wang and Low, 2011] classifies mobile robots according to the influence of tracks skid and slip on their kinematic model. Modelling skid and steer is not straightforward, owning the fact that it would require a description of forces acting between tracks and a motion surface influenced by highly nonlinear physical dynamic effects [Michałek et al., 2009]. Some works focus upon track-soil interaction providing the possibility to represent different phenomena in terms of

adimensional coefficients relating robot velocities with assigned tracks speeds [Wong, 2008]. Several solutions for the identification of the values of these coefficients are available in literature [Endo et al., 2007; Nagatani et al., 2008; Reinstein et al., 2013; Sevil et al., 2012; Moosavian, 2008; Dogru and Marques, 2017].

3.2 Skid-steer tracked mobile robot model

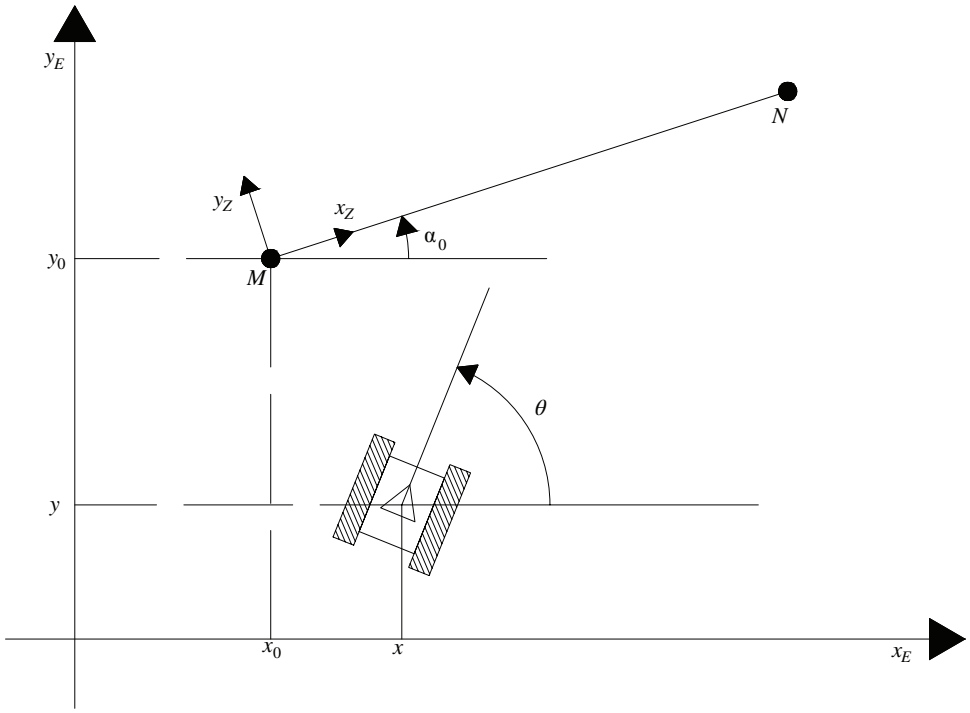


Fig. 3.1. Reference frame for the skid-steered tracked vehicle.

Let assume a generic differentially driven tracked mobile robot and let **E** be an inertial reference frame, see Fig.3.1. Let

$$q(t) = \left[x(t)\, y(t)\, \theta(t)\right]^T \tag{3.1}$$

be the pose vector of the robot at the time instant t. Let assume that robot is controlled in terms of forward $V(t)$ and rotational $\omega(t)$ velocities. Let define the vector of robot velocities $U(t) = \left[V(t)\ \omega(t)\right]^T$.

The classic nonholonomic kinematic first-order model can be written [Kelly, 2004] as

$$\dot{q}(t) = G(q) \cdot U(t) \tag{3.2}$$

with

$$G(q) = \begin{bmatrix} \cos\theta(t) & 0 \\ \sin\theta(t) & 0 \\ 0 & 1 \end{bmatrix} \tag{3.3}$$

In this thesis, slip and skid phenomena are modelled through two time-varying terrain-dependent friction positive coefficients $\mu_L(t)$ and $\mu_R(t)$ for left and right track respectively. Specifically, in order to embed terrain-dependent friction coefficients, be the following relationship between robot velocities $U(t)$ and tracks sprockets angular velocities

$$U(t) = J \cdot H(t) \cdot w(t) \tag{3.4}$$

being

$$w(t) = \begin{bmatrix} w_R(t) & w_L(t) \end{bmatrix}^T \tag{3.5}$$

$$J = \begin{bmatrix} R/2 & R/2 \\ R/D & -R/D \end{bmatrix} \tag{3.6}$$

and

$$H(t) = \begin{bmatrix} \mu_R(t) & 0 \\ 0 & \mu_L(t) \end{bmatrix} \tag{3.7}$$

where D is the distance between tracks, R is the radius of track sprocket, $w_R(t)$ and $w_L(t)$ are the right and left track sprocket angular velocities respectively.

Following these assumptions, let assume

$$\hat{U}(t) = \begin{bmatrix} \hat{V}(t) & \hat{\omega}(t) \end{bmatrix}^T \tag{3.8}$$

where $\hat{V}(t)$ and $\hat{\omega}(t)$ are the forward and rotational control velocities. Ideally, given the hypothesis that $\bar{\mu}_R$ and $\bar{\mu}_L$ are the nominal values of friction coefficients, differing velocities of the two tracks, required to obtain (3.8), are

$$\hat{w}(t) = \bar{H}^{-1} \cdot J^{-1} \cdot \hat{U}(t) \tag{3.9}$$

where

$$\bar{H} = \begin{bmatrix} \bar{\mu}_R & 0 \\ 0 & \bar{\mu}_L \end{bmatrix} \tag{3.10}$$

However, since $H(t)$ might differ from $\bar{H}$ at the time instant t, following (3.4), the actual robot velocities are

$$U(t) = J \cdot H(t) \cdot \hat{w}(t) \tag{3.11}$$

By merging (3.9) and (3.11), eq. (3.2) can be reformulated in the following form:

$$\dot{q}(t) = G(q) \cdot J \cdot H(t) \cdot H_N^{-1} \cdot J^{-1} \cdot \hat{U}(t) \tag{3.12}$$

The estimation of difference between real and planned robot poses, hereinafter named the trajectory tracking error, can be achieved by considering the trajectory segment $\overline{MN}$ represented in Fig.3.1. Let suppose a reference frame **Z** having its base point in M and x_Z oriented as the $\overline{MN}$ segment. Robot pose $q(t)$ defined in **E** can be promptly expressed in **Z** by the subsequent roto-translation

$$q_Z(t) = R_{E,Z}(\alpha_0) \cdot (q(t) - q_0) \tag{3.13}$$

being

$$R_{E,Z}(\alpha_0) = \begin{bmatrix} cos(\alpha_0) & sin(\alpha_0) & 0 \\ -sin(\alpha_0) & cos(\alpha_0) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.14}$$

the matrix expressing the rotation from **E** to **Z** while $q_0 = \begin{bmatrix} x_0 \; y_0 \; \alpha_0 \end{bmatrix}^T$ represents the origin $[x_0, y_0]$ and orientation α_0 of **Z** expressed in **E**.

Be $\bar{q}_Z(\cdot)$ the nominal (planned) robot pose according to **Z** over the timespan $[0, t]$ while crossing the $\overline{MN}$ trajectory segment at a constant forward velocity $\bar{V}$. According to (3.2) the subsequent equation is held:

$$\dot{\bar{q}}_Z(t) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \cdot \bar{U} \tag{3.15}$$

with $\bar{q}_Z(0) = \begin{bmatrix} 0 \; 0 \; 0 \end{bmatrix}^T$ where $\bar{U} = \begin{bmatrix} \bar{V} \; 0 \end{bmatrix}^T$.

The real robot pose expressed in **Z** can be recast in the following form

$$q_Z(t) = \bar{q}_Z(t) + e(t) \tag{3.16}$$

thus, the trajectory tracking error dynamic can be rewritten as

$$\dot{e}(t) = \dot{q}_Z(t) - \dot{\bar{q}}_Z(t) \tag{3.17}$$

Assume the following conditions

$$\hat{U}(t) = \bar{U} + u(t) \tag{3.18}$$
$$\mu_R(t) = \bar{\mu}_R + d_{\mu_R}(t) \tag{3.19}$$
$$\mu_L(t) = \bar{\mu}_L + d_{\mu_L}(t) \tag{3.20}$$

By applying classical linearization procedure around nominal condition, the following linear approximation of (3.17) can be written:

$$\dot{e}(t) = Ae(t) + Bu(t) + B_{D_C}d(t) \tag{3.21}$$

$$A = \frac{\partial \dot{e}(t)}{\partial q_Z(t)}\bigg|_{\bar{q}_Z(t),\bar{U},\bar{\mu}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & V_L^N \\ 0 & 0 & 0 \end{bmatrix} \tag{3.22}$$

$$B = \frac{\partial \dot{e}(t)}{\partial \hat{U}(t)}\bigg|_{\bar{q}_Z(t),\bar{U},\bar{\mu}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.23}$$

$$B_{D_C} = \frac{\partial \dot{e}(t)}{\partial \mu(t)}\bigg|_{\bar{q}_Z(t),\bar{U},\bar{\mu}} = \begin{bmatrix} \frac{V_L^N}{2\mu_R^N} & \frac{V_L^N}{2\mu_L^N} \\ 0 & 0 \\ \frac{V_L^N}{D\mu_R^N} & -\frac{V_L^N}{D\mu_L^N} \end{bmatrix} \tag{3.24}$$

being $\mu(t) = \begin{bmatrix}\mu_R(t)\ \mu_L(t)\end{bmatrix}^T$, $\bar{\mu} = \begin{bmatrix}\bar{\mu}_R\ \bar{\mu}_L\end{bmatrix}^T$ and $d(t) = \begin{bmatrix}d_{\mu_R}\ d_{\mu_L}\end{bmatrix}^T$.

3.3 Modelling of a NCS skid-steer tracked mobile robot

Given a trajectory tracking control law, let suppose it is evaluated on a remote ground-station computer connected to the mobile robot through a communication network. Classical communication networks problems arising from unpredictable delays and packets loss are taken into account by adopting the approach proposed in [Cloosterman et al., 2010b]. Consider the classical Networked Control System (NCS) schema depicted in Fig.3.2. Two time-varying delays induced by the communication network are considered, namely the *sensors-to-control* delay τ_k^{SC} and the *control-to-actuator* delay τ_k^{CA}. Furthermore, a complete measurement of state vector $x(t)$ is assumed to be available at the sampling time instant

$$t_k = \sum_{i=0}^{k-1} T_S \;\; \forall k \geq 1, \;\; t_0 = 0 \tag{3.25}$$

being $T_S > 0$ the constant sampling time. Assume the following notations: $x_k = x(t_k)$ ($u_k = u(t_k)$) is the *k-th* sampled value of x (u). The *zero-order holder* (ZOH) holds the control input u_k until a new control action is successfully received from the controller. Variable computation time needed to evaluate the control law is denoted by τ_k^C. In this work, it is assumed that sensors are sampled at the time instants t_k defined in (3.25) whereas controller and actuator act as event-driven systems. Under the above assumptions all three time-variable delays can be embedded in a single delay [Bernardini et al., 2010]

$$\tau_k = \tau_k^{SC} + \tau_k^{CA} + \tau_k^C \tag{3.26}$$

hereinafter called network delay. Then, the continuous input signal $u(t)$ can be defined as $u(t) = u_k$ if $t \in [t_k + \tau_k, \, t_{k+1} + \tau_{k+1}) \; \forall k \in \mathbb{N}$.

All the considered delays are supposed bounded. Moreover, commu-

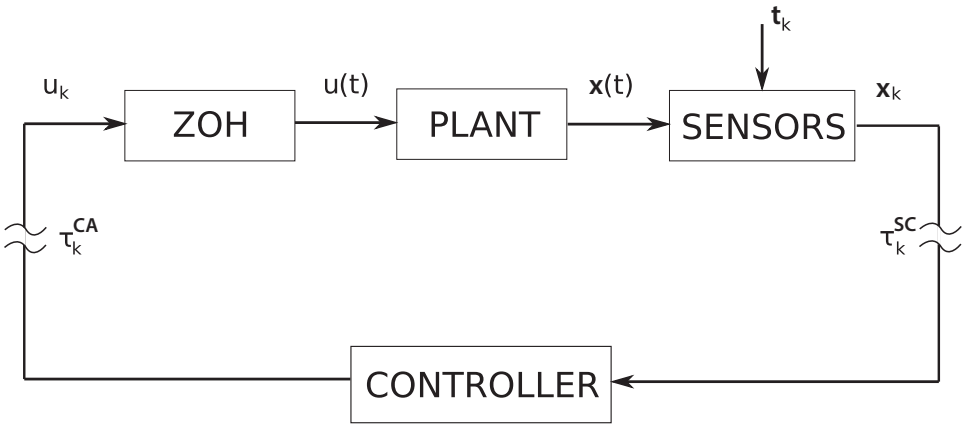


Fig. 3.2. Classical NCS control schema.

nication along the control loop is carried out, in this work, using TCP over IP protocol stack [Open Source Robotics Foundation, 2013]. TCP is a connection-oriented and ordered communication protocol, thus, packet loss can be assumed to be zero if τ_k^{SC} and τ_k^{CA} are never longer than the relevant TCP max retransmission window duration. Be the integer number $\bar{d}$ the smallest value for which the following inequality holds

$$\bar{d} \geq \frac{\tau_{max}}{T_s} \tag{3.27}$$

where τ_{max} is the upper bound for network delay. Be the integer $\underline{d}$ the largest value such that

$$\underline{d} \leq \frac{\tau_{min}}{T_s} \tag{3.28}$$

where τ_{min} is the lower bound for network delay. Linear time-invariant model of trajectory tracking error dynamic (3.21) can be expressed by the following discrete form

$$e_{k+1} = e^{AT_s} e_k + \sum_{j=\underline{d}}^{\bar{d}} \int_{t_k^{\bar{d}-j}}^{t_{\bar{d}-j+1}} e^{A(t_{k+1}-\sigma)} d\sigma B u_{k-j} + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-\sigma)} d\sigma B_{D_C} d_k \tag{3.29}$$

Let θ_k be the vector of actuation update instants, see Fig.3.3

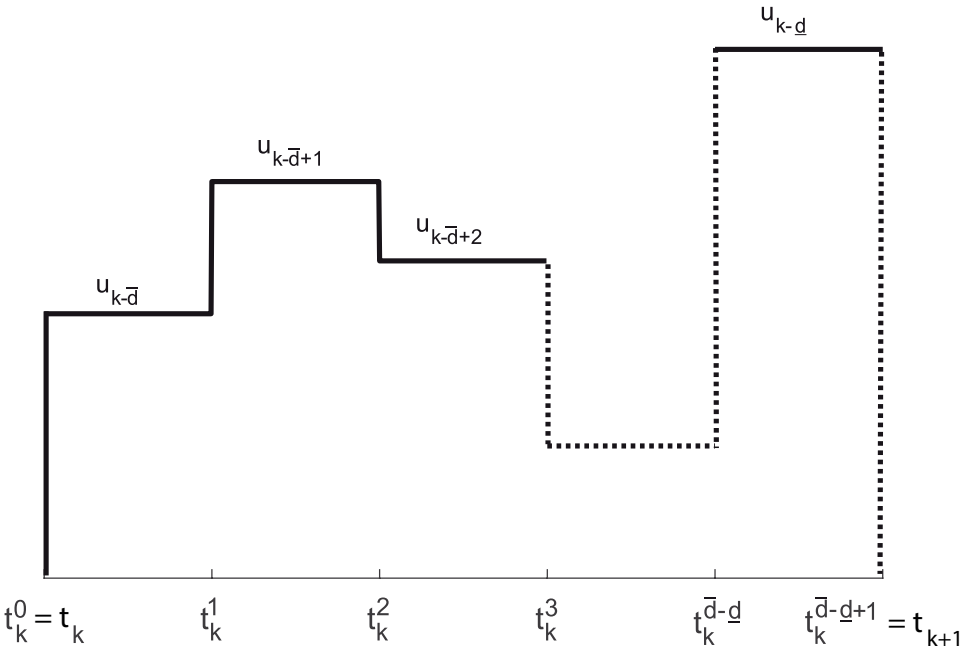


Fig. 3.3. Actuation update time instants t_k^j

$$\theta_k := \left[t_k^1 \cdots t_k^{\bar{d}-\underline{d}} \right]^T \tag{3.30}$$

where

$$t_k = t_k^0 \leq t_k^1 \leq \cdots \leq t_k^{\bar{d}-\underline{d}+1} = t_{k+1} \tag{3.31}$$

Be ξ_k the lifted state vector

$$\xi_k = \begin{bmatrix} e_k^T & u_{k-1}^T & \cdots & u_{k-\bar{d}}^T \end{bmatrix}^T \tag{3.32}$$

eq. (3.29) can be readily recast in the following form

$$\xi_{k+1} = \tilde{A}(\theta_k)\xi_k + \tilde{B}(\theta_k)u_k + \tilde{B}_D d_k \tag{3.33}$$

being

$$\tilde{A}(\theta_k) = \begin{bmatrix} e^{AT_s} & M_1(\theta_k) & M_2(\theta_k) & \cdots & M_{\bar{d}-1}(\theta_k) & M_{\bar{d}}(\theta_k) \\ 0 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & I & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & 0 & I & 0 \end{bmatrix} \tag{3.34}$$

$$\tilde{B}(\theta_k) = \begin{bmatrix} M_0(\theta_k) \\ I \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{3.35}$$

where

$$M_j(\theta_k) = \begin{cases} \int_{t_k^{\bar{d}-j}}^{t_{\bar{d}-j+1}} e^{A(t_{k+1}-\sigma)} d\sigma B & \text{if } \underline{d} \leq j \leq \overline{d} \\ 0 & \text{otherwise} \end{cases} \tag{3.36}$$

As shown in [Posthumus - Cloosterman, 2008] this structure can be manipulated to remove the dependence from θ_k through real Jordan form decomposition. The Jordan form Q of state transition matrix $\tilde{A}(\theta_k)$ is $Q = T\tilde{A}(\theta_k)T^{-1}$, with the invertible matrix T containing $\tilde{A}(\theta_k)$ generalized eigenvectors. Let $\lambda_q \in \begin{bmatrix} \lambda_1 \ldots \lambda_p \end{bmatrix}$ be the eigenvalues of $\tilde{A}(\theta_k)$, it must be noted $\tilde{A}(\theta_k)$ can have only real eigenvalues. Therefore Q has a diagonal-block structures, with $Q_1, \ldots, Q_q, \ldots, Q_p$ blocks. The *Jordan block* q for the λ_q eigenvalue of $\tilde{A}(\theta_k)$:

$$Q_q = \begin{bmatrix} \lambda_q & 1 & 0 & \dots & 0 \\ 0 & \lambda_q & 1 & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \dots & \lambda_q & 1 \\ 0 & 0 & \dots & 0 & \lambda_q \end{bmatrix} \tag{3.37}$$

$Q_q \in \mathbb{R}^{\kappa \times \kappa}$, with κ the algebraic multiplicity of λ_q[Posthumus - Cloosterman, 2008; Cloosterman et al., 2010a].
It follows that

$$e^{A\sigma} = T \begin{bmatrix} e^{G_1\sigma} & 0 & \dots & 0 & \\ 0 & \dots & 0 & \dots & 0 \\ \vdots & & e^{G_q\sigma} & & \vdots \\ 0 & 0 & \dots & \dots & 0 \\ 0 & 0 & \dots & 0 & e^{G_p\sigma} \end{bmatrix} T \tag{3.38}$$

where:

$$G_q = e^{\lambda_q \sigma} \begin{bmatrix} 1 & \sigma & \frac{\sigma^2}{2!} & \cdots & \frac{\sigma^{(k-1)}}{(k-1)!} \\ 0 & 1 & \sigma & \cdots & \frac{\sigma^{(k-2)}}{(k-2)!} \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & 1 & \sigma \\ 0 & 0 & & 0 & 1 \end{bmatrix} \tag{3.39}$$

With $G_q \in \mathbb{R}^{\kappa \times \kappa}$. This allows to express 3.36 with the following form [Heemels et al., 2010]:

$$M_j(\theta_k) = A_0 + \sum_{j=1}^{\zeta} \alpha_j(\theta_k) A_j \tag{3.40}$$

with $\zeta = (\bar{d} - \underline{d})r$ and r is the degree of the minimal polynomial of A [Heemels et al., 2010] i.e., the degree of the polynomial P holding the condition $P(A) = 0$ having the smallest degree. As shown in [Bemporad et al., 2010b], by considering the maximum and minimum values of $\alpha_j(\theta_k)$ $\forall \theta_j \in \Theta$, (3.40) can be overapproximated with the convex hull

$$Co\{\psi_1, \psi_2, \cdots \psi_{N_v}\} := \left\{ \sum_{n=1}^{N_v} \alpha_n \psi_n \middle| \sum_{n=1}^{N_v} \alpha_n = 1 \right\} \tag{3.41}$$

with N_v the number of vertices of the convex hull and the n-th vertex $\psi_n = (\tilde{A_n}, \tilde{B_n}, \tilde{B_D})$. Finally, according to [Lombardi et al., 2009], the

conservativity of this approach can be reduced by dividing sampling interval T_s into g subintervals, at the price of linearly increasing the number of polytope vertices. It follows that

$$N_v = 2^{(\bar{d}-\underline{d})r} \cdot g \tag{3.42}$$

. Hence obtaining the following discrete polytopic linear time invariant system which represents trajectory tracking error dynamic

$$\xi_{k+1} = \tilde{A}_n \xi_k + \tilde{B}_n u_k + \tilde{B}_D d_k, \;\; n = 1 \cdots N_v \tag{3.43}$$

In order to improve trajectory tracking performances, a Proportional-Integral (PI) control law action will be synthetized, such control law has the following structure:

$$u_k = Kp \cdot \xi_k + Ki \cdot \sum_{q=0}^{k-1} S \cdot \xi_q \cdot Ts \tag{3.44}$$

being Kp and Ki PI gain matrices while $S \in \mathcal{R}^{2\times 7}$ is a matrix selecting tracking error along the x and y dimensions

$$S = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0\,0\,0\,0 \end{bmatrix} \tag{3.45}$$

This control law structure (3.44) can be obtained by the means of a static state-feedback control law, which will be synthetized in section 4.1, by considering an augmented form of (3.43). Let $w \in \mathbb{R}^{2\times 1}$ be the two *integral states* i.e., the integral in time of x and y tracking errors:

$$w_{k+1} = w_{k+1} + S \cdot \xi_k \cdot Ts \tag{3.46}$$

hence, the trajectory tracking error polytopic model can be expressed as

$$\hat{\xi}_{k+1} = \hat{A}_n \hat{\xi}_k + \hat{B}_n u_k + \hat{B}_D d_k, \;\; n = 1 \cdots N_v \tag{3.47}$$

being $\hat{\xi} = \begin{bmatrix} \xi \\ w \end{bmatrix}$, $\hat{A}_n = \begin{bmatrix} \tilde{A}_n & 0 \\ S \cdot T_S & I \end{bmatrix}$, $\hat{B}_n = \begin{bmatrix} \tilde{B}_n \\ 0 \end{bmatrix}$ and $\hat{B}_D = \begin{bmatrix} \tilde{B}_D \\ 0 \end{bmatrix}$.
For the sake of completeness, it must be underlined that this representation can be also conservatively approximated with a norm-bound representation[Boyd et al., 1994]. Such kind of representation comes, in practice, in the following form:

$$\begin{aligned}\tilde{\xi}_{k+1} &= \hat{A}\,\hat{\xi}_k + \hat{B}\,u_k + \hat{B}_D\,d_k + B_p\,\Delta(k)\,p_k \\ p_k &= C_p\,\xi_k \\ \|\Delta(k)\| &\leq 1\end{aligned} \tag{3.48}$$

with B_p and C_p matrices of proper dimensions. This is, indeed, a system with a time-varying feedback matrix.
The representation 3.48 can be obtained by solving the following eigenvalues problem:

$$\min_{O,T,W} tr(T) + tr(W) \tag{3.49}$$

s. t.

$$\begin{bmatrix} T & (\hat{A}_n - O)^T \\ * & W \end{bmatrix} \geq 0\, n = 1, \ldots, N_v \tag{3.50}$$

this representation offers the practical advantage of being more concise than 3.47, leading to scaling down the complexity of related computations by a factor close to N_V since each problem can be solved for a single, slightly more complex, mathematical structure. This advantage comes at the price of a certain conservativity. In this work, given the focus on the off-line planning phase, the polytopic structure will be used.

4

Trajectory planning algorithm

In this chapter the problem of trajectory planning is tackled. A constrained control problem is solved, thus defining the closed-loop dynamics of the robot. The operational scenario is represented through a graph connecting safe positions in a discretized operational scenario. Subsequently, set-based arguments are used to define and provide a practical procedure to check whether arc nodes can be combined in a trajectory which is compatible with robot dynamics.
In this thesis the following definitions are used [Brockman and Corless, 1998; Blanchini, 1999]:

- *(positively) invariant set* - a set Λ is said *invariant*, or *positively invariant* for the system $x_{k+1} = \Phi\, x_k$ if $x_0 \in \Lambda \implies x_k \in \Lambda \quad \forall k > 0$;
- *d-invariant set* - invariance property is held in presence of an input belonging to a bounded set. Thus, the set Λ is said the be d-invariant or disturbance (positively) invariant for the system $x_{k+1} = \Phi\, x_k + B_D\, d_k$, if $x_0 \in \Lambda \implies x_k \in \Lambda \quad \forall d_k \in \Omega_D$;
- *robustness* of a set - the properties of invariance or d-invariance are referred to an uncertain system.

4.1 Constrained control problem

As already outlined, this work proposes a solution to the trajectory planning problem which takes into account closed loop dynamics of the vehicle. So, the first step of planning lies in control synthesis, given a set of properly defined constraints. Namely, the control velocities introduced in (3.18) and lifted state vector (3.32) are supposed, in this

thesis, to be bounded according to the subsequent ellipsoidal constraints

$$\xi \in \Omega_\xi,\ \Omega_\xi \triangleq \{\xi \in \mathcal{R}^{n_\xi} : \xi^T \xi \leq \xi_{max}^2\} \tag{4.1}$$

$$u \in \Omega_u,\ \Omega_u \triangleq \{u \in \mathcal{R}^2 : u^T u \leq u_{max}^2\} \tag{4.2}$$

Let assume, moreover, that disturbances vector d, which embeds skid and slip phenomena, is bounded according to the ellipsoidal constraint

$$\Omega_D \triangleq \{d \in \mathcal{R}^2 : d^T d \leq d_{max}^2\} \tag{4.3}$$

Constrained Control Problem (CCP) - *find a stabilizing state feedback regulator*

$$u_t = K \cdot \xi_t \tag{4.4}$$

fulfilling constraints (4.1)-(4.2), given (3.43), for any possible realization of external disturbance (4.3)

A solution of **CCP** can be obtained by the following two step procedure.

- According to Kothare et al. [1996], synthesize a quadratic stabilizing state-feedback control law (4.4) fulfilling (4.1) and (4.2) inside the ellipsoidal positively invariant region Γ_0 with radius γ_0:

$$\Gamma_0 = \{\xi \in \mathcal{R}^{n_\xi} : \xi^T P_0 \xi \leq \gamma_0 \ \ P_0 \geq 0\} \tag{4.5}$$

  details are provided in 4.1
- Check whether (4.10) is disturbance invariant (i.e., it is invariant for any admissible realization of disturbance d) through theorem 4.1, otherwise repeat control synthesis by properly adjusting synthesis parameters. A detailed explanation of this step is presented in 4.1.

Step 1

The control law 4.4 can be written as $K = YQ^{-1}$, with matrices Q and Y that can be obtained by solving the following SDP problem:

$$\min_{\gamma,Q,Y} \gamma \tag{4.6}$$

s. t.

$$\begin{bmatrix} 1 & \xi_0^T \\ \xi_0 & Q \end{bmatrix} \geq 0 \tag{4.7}$$

$$\begin{bmatrix} Q & Q\tilde{A}_n^T + Y^T\tilde{B}_n^T & Q\chi^{\frac{1}{2}} & Y^T R^{\frac{1}{2}} \\ \tilde{A}_n Q + \tilde{B}_n Y & Q & 0 & 0 \\ \chi^{\frac{1}{2}} Q & 0 & \gamma I & 0 \\ R^{\frac{1}{2}} Y & 0 & 0 & \gamma I \end{bmatrix} \geq 0 \tag{4.8}$$

$$\begin{bmatrix} V_{max}^2 & 0 & Y \\ 0 & u_{max}^2 & \\ & & \\ Y^T & & Q \end{bmatrix} \geq 0 \tag{4.9}$$

with $n = 1 \dots N_v$ and $P_0 = Q^{-1}$.
$\chi \in \mathbb{R}^{n_\xi}$ and $R \in \mathbb{R}^2$ are, respectively, the LQR symmetric weight matrices for state and input tracking performances.
V_{max} and u_{max} are the maximum allowable values of actuation, for forward and rotational velocities, respectively. Finally, ξ_0 is the initial tracking error which must belong to Γ_0. Given the state-feedback matrix K, the closed loop system can be rewritten in the following polytopic form

$$\xi_{k+1} = \tilde{\phi}_n \, \xi_k + \tilde{B}_D \, d_k, \;\; n = 1 \cdots N_v \tag{4.10}$$

being $\tilde{\phi}_n = \tilde{A}_n + \tilde{B}_n \, K$ with $n = 1 \cdots N_v$.

Step 2

Theorem 4.1. *The invariant set Γ_0 is also a robustly disturbance-invariant set for the bounded disturbance $d \in \Omega_D$ if, for each vertex of the polytope 3.43, the following LMI feasibility problem in τ and P_Z holds:*

$$\tau \geq 0 \tag{4.11}$$

$$P_Z \geq 0 \tag{4.12}$$

$$\begin{bmatrix} P_Z - \tilde{\phi}_n^T \, P_Z \, \tilde{\phi}_n^T & -\tilde{\phi}_n^T \, P_Z \, \tilde{B}_D \\ * & \tau \, M - \tilde{B}_D^T \, P_Z \, \tilde{B}_D \end{bmatrix} \geq 0 \tag{4.13}$$

$$\begin{bmatrix} \gamma_0 \, (P_Z - \tilde{\phi}_n^T \, P_Z \tilde{\phi}_n^T) - \tau \, P_0 & -\tilde{\phi}_n^T \, P_Z \, \tilde{B}_D \sqrt{\gamma_0} \\ * & -\tilde{B}_D^T \, P_Z \, \tilde{B}_D + \tau \, M \end{bmatrix} \geq 0 \tag{4.14}$$

Proof - Let introduce the region Γ_C which holds the following property:

$$V_{k+1} - V_k \leq 0 \qquad \forall d \in \Omega_D \tag{4.15}$$

with V_k the quadratic Lyapunov function $V_k = \xi^T P_Z \xi$. The condition 4.15 can be expressed, for each vertex of the polytope representing the networked closed loop system, as follows:

$$(\tilde{\phi}_n\,\xi_k + \tilde{B}_D\,d_k)^T P_Z(\tilde{\phi}_n\,\xi_k + \tilde{B}_D\,d_k) - \xi_k^T\,P_Z\,\xi_k \leq 0 \qquad \forall d^T M d^T \leq 1 \tag{4.16}$$

with $M > 0$ the matrix defining the ellipsoidal region 4.3. This condition can be reformulated as:

$$\xi_k^T\,\tilde{\phi}_n^T\,P_Z\,\tilde{\phi}_n\,\xi_k + 2\,\xi_k^T\,\tilde{\phi}_n\,\tilde{B}_D\,d_k + d_k^T\,\tilde{B}_D\,P_Z\,\tilde{B}_D\,d_k - \xi^T\,P_Z\,\xi \leq 0$$

which can be expressed as:

$$\begin{bmatrix}1\;d_k^T\end{bmatrix}\begin{bmatrix}\xi_k^T\,(P_Z - \tilde{\phi}_n^T\,P_Z\,\tilde{\phi}_n)\xi_k & * \\ -\xi_k^T\,\tilde{\phi}_n^T\,P_Z\,\tilde{B}_D & -\tilde{B}_D^T\,P_Z\,\tilde{B}_D\end{bmatrix}\begin{bmatrix}1\\ d_k\end{bmatrix} \geq 0$$

$$\begin{bmatrix}1\;d_k^T\end{bmatrix}\begin{bmatrix}1 & 0\\ * & -M\end{bmatrix}\begin{bmatrix}1\\ d_k\end{bmatrix} \geq 0 \tag{4.17}$$

both LMIs can be combined through the *s-procedure*, leading to the following:

$$\begin{bmatrix}\xi_k^T\,(P_Z - \tilde{\phi}_n^T\,P_Z\,\tilde{\phi}_n)\xi_k - \tau & -\xi_k^T\,\tilde{\phi}_n^T\,P_Z\,\tilde{B}_D \\ * & -\tilde{B}_D^T\,P_Z\,\tilde{B}_D + \tau\,M\end{bmatrix} \geq 0$$

using the Schur complement, this LMI can be recast in the following set of inequalities:

$$\xi_k^T(P_Z - \tilde{\phi}_n^T\,P_Z\,\tilde{\phi}_n - \tilde{\phi}_n^T\,P_Z, \tilde{B}_D(-\tilde{B}_D^T\,P_Z\,\tilde{B}_D + \tau\,M)^{-1}\tilde{B}^T\,P_Z\,\tilde{\phi}_n)\xi_k \geq \tau \tag{4.18}$$

$$\tau\,M - \tilde{B}_D^T\,P_Z\,\tilde{B}_D \geq 0 \tag{4.19}$$

$$\tau \geq 0 \tag{4.20}$$

$$P_Z \geq 0 \tag{4.21}$$

$$\xi_k^T\,P_0\,\xi_k \leq \gamma_0 \tag{4.22}$$

condition 4.18 can be rewritten, in terms of Schur's complement as:

$$\begin{bmatrix}P_Z - \tilde{\phi}_n^T\,P_Z\,\tilde{\phi}_n^T & -\tilde{\phi}_n^T\,P_Z\,\tilde{B}_D \\ * & \tau\,M - \tilde{B}_D^T, P_Z\,\tilde{B}_D\end{bmatrix} \geq 0$$

interestingly, equation 4.18-4.22 define Γ_C, if exists, as the intersection between the external region of an ellipsoid with Γ_0.

The meaning of Γ_C can be understood by induction.
Γ_0 is, by definition, non-degenerate, i.e., it contains both zero and elements which are not zero. Let suppose that there exist a non-empty region $\Gamma_F \coloneqq \Gamma_0 - \Gamma_C$. It follows that such region is the region of states whose one-step contraction towards zero is smaller than the effect of disturbance on a single sampling step.
Let consider the evolution in time of the $\xi_k = 0$. At the next sampling step, the associated value ξ_{k+1} is $\xi_{k+1} = \tilde{B}_D\, d_k$, which means that the property 4.15 is not held, thus $0 \in \Gamma_F$. Analogously, small values of ξ_k will show the same behaviour, while in Γ_C the control-induced contraction of Lyapunov function is greater than $\tilde{B}_D\, d_k$.
The next point in state evolution is $\xi_{k+2} = \tilde{\Phi}_n\, \tilde{B}_D\, d_k + \tilde{B}_D\, d_{k+1}$ which can either belong to Γ_C or Γ_F. If it belongs to Γ_C the associated value of the Lyapunov quadratic function will decrease at the next sampling step, otherwise, it can either leave Γ_F, ending up in Γ_C or stay in Γ_F. Figure 4.1 provides a visual overlook of this property. So, the existence of Γ_C proves that Γ_0 is d-invariant.
For reference, the problem has been tackled by [Blanchini, 1992] through polyhedra, while [Kolmanovsky and Gilbert, 1998] deals with invariance in terms of Pontryagin difference, offering tools for constructing polyhedral invariant sets. Sufficient and necessary conditions, involving nonlinear programming, for d-invariance are presented in [Kolmanovsky and Gilbert, 1998], while [Schweppe, 1973] proposes a sufficient condition for ellipsoidal sets for continuous-time systems which is proved to be also necessary by [Brockman and Corless, 1998], although it requires the check of a feasibility LMI condition on the bounded range of a parameter. Finally, a survey of relevant solution is performed by [Blanchini, 1999].
Given the above 4.18 can be expressed as $\xi_k^T\, Q_C(\tau)\, \xi_k^T \geq \tau$, with $Q_C(\tau) \geq 0.$, this allows the enforcement of the following condition:

$$\frac{1}{\tau}\, Q_C(\tau) \geq \frac{1}{\gamma_0}\, P_0 \tag{4.23}$$

which can be expressed as:

$$\begin{bmatrix} \gamma_0\, (P_Z - \tilde{\phi}_n^T\, P_Z \tilde{\phi}_n^T) - \tau\, P_0 & -\tilde{\phi}_n^T\, P_Z\, \tilde{B}_D \sqrt{\gamma_0} \\ * & -\tilde{B}_D^T\, P_Z\, \tilde{B}_D + \tau\, M \end{bmatrix} \geq 0 \tag{4.24}$$

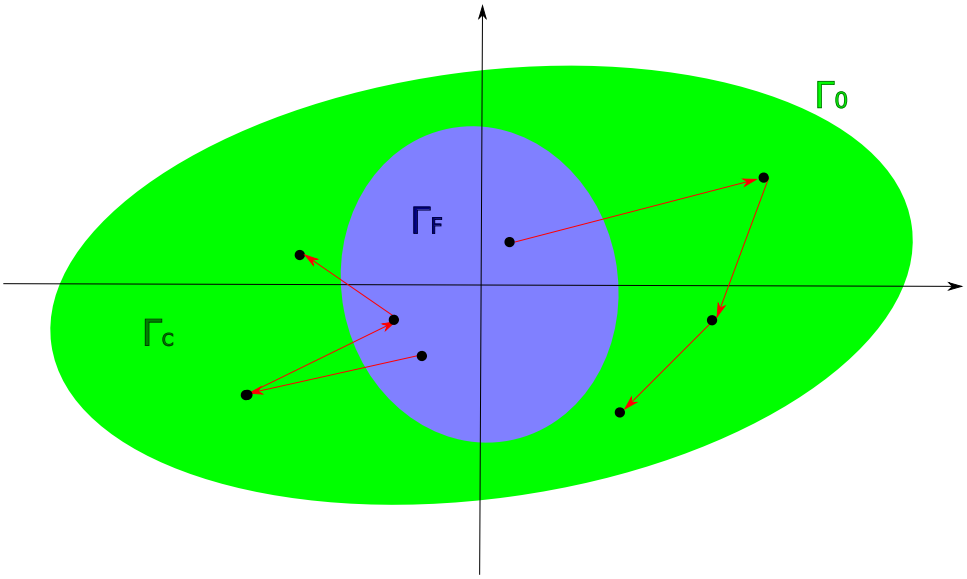


Fig. 4.1. A two-dimensional visual overlook of the property 4.1. The green area denotes a robustly d-invariant subset of Γ_0, while the blue area is the region Γ_F.

this leads to the LMI feasibility problem in τ and P_Z optimization variables 4.1. □

4.2 Trajectory feasibility

Let assume a 2D operational scenario $\Delta \subseteq \mathcal{R}^2$, discretized by finite dimensional grid C. An undirected weighted graph $\mathcal{G}$ is obtained using discrete points of Δ as nodes, further details about the process of mapping and graph construction are reported in 5.1.
Def. Δ-compatibility. Two points $A = (x_1, y_1)$ and $B = (x_2, y_2)$ closer than $\bar{L}$ are defined as Δ-compatible if $\forall \alpha \in [0, 1]$

$$P_\alpha = (x_\alpha, y_\alpha) \in \Delta \tag{4.25}$$

with $x_\alpha = (1 - \alpha)x_1 + \alpha x_2$ and $y_\alpha = \eta_\alpha x_\alpha + \rho_\alpha$ with

$$\eta_\alpha = \frac{y_1 - y_2}{x_1 - x_2} \tag{4.26}$$

and

$$\rho_\alpha = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2} \tag{4.27}$$

i.e. if the segment connecting them is completely included in Δ Let A and B be two Δ-compatible points and C_A and C_B be the two nodes associated with them. Thus, T^{AB} stands for the arc of graph $\mathcal{G}$ which connects C_A and C_B. It represents the trajectory segment $\overline{AB}$ of length l_{AB} to be crossed at the constant velocity V^{AB}.
Let define N_{AB} as the highest positive integer holding the inequality

$$N_{AB} \leq \frac{l_{AB}}{V^{AB} \cdot T_s} \tag{4.28}$$

In this thesis, the weight associated to T^{AB} is its length l_{AB}.

Be $C_A, C_B, C_C \in C$ three nodes representing, respectively, points $A, B, C \in \Delta$. Let suppose that the couples A, B and B, C are both Δ-compatible. Let assume a trajectory including the two adjacent arcs T^{AB} and T^{BC} as shown in Fig.4.2. In addition, be $[t_A, t_A + N_{AB} \cdot T_s]$ the timespan during which the mobile robot crosses the segment $\overline{AB}$. At time instant $t_{N_{AB}} = t_A + N_{AB} \cdot T_s$ a switch from arc T^{AB} and to arc T^{BC} is required. Switch is defined as admissible if the following condition is held:

$$(\xi_{N_{AB}} + \Pi) \in \Gamma_0 \tag{4.29}$$

where $\xi_{N_{AB}}$ is the trajectory tracking error at the switching time $t_{N_{AB}}$ according to (4.10) while

$$\Pi = \left[x_B - x^D(t_{N_{AB}}), y_B - y^D(t_{N_{AB}}), \delta_\theta, 0, \cdots 0\right]^T \tag{4.30}$$

is the variation of trajectory tracking error caused by the switch, see Fig. 4.2 for details.

If each switch between the segments of a trajectory is admissible, then the whole trajectory is feasible. Admissibility of switches can be asserted by considering the subsequent result.

Given (4.10), let $\xi_A \in S_A$ be the bounded but unknown initial tracking error at time t_A where

$$S_A = \{\xi \in \Gamma_0 : \xi^T P_A \xi \leq 1,\ P_A \geq 0\} \tag{4.31}$$

Let $S_{N_{AB}} \in \Gamma_0$ be the smallest ellipsoidal set

$$S_{N_{AB}} = \{\xi \in \Gamma_0 : \xi^T P_0 \xi \leq \tilde{\gamma},\ \ 0 \leq \tilde{\gamma} \leq 1\} \tag{4.32}$$

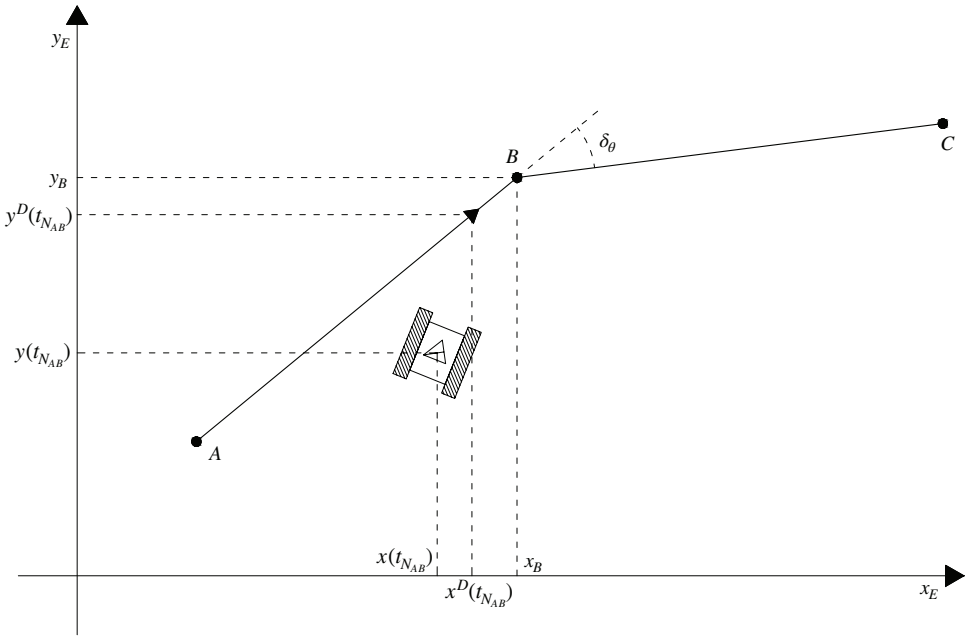


Fig. 4.2. Trajectory switch.

which contains all allowable tracking errors at the time instant $t_{N_{AB}}$,

$$\xi_{N_{AB}} = \tilde{\Phi}_n^{N_{AB}} \xi_A + \left[\tilde{\Phi}_n^{N_{AB}-1}\tilde{B}_D \;\; \tilde{\Phi}_n^{N_{AB}-2}\tilde{B}_D \cdots \tilde{B}_D\right] \hat{D} \tag{4.33}$$

with $n = 1, \cdots, N_v$ and $\hat{D} = \left[d_0^T \, d_1^T \cdots d_{N_{AB}}^T\right]^T$. S_{NAB} can be embedded in an ellipsoidal form obtained by considering the minimum volume ellipsoidal set which contains all S_{NAB}^n obtained by exploiting Theorem 1. S_{NAB} represents the minimum volume ellipsoidal set containing tracking error at the time instant $t_{N_{AB}}$ for the $n-th$ realization of system (4.10).

Theorem 4.2. *Let consider the $n-th$ vertex of polytopic linear time invariant system (4.10) with $n = 1 \cdots N_v$. The smallest ellipsoidal set*

$$S_{N_{AB}}^n = \{\xi \in \Gamma_0 : \xi^T P_0 \xi \leq \tilde{\gamma}_n, \;\; 0 \leq \tilde{\gamma}_n \leq 1\} \tag{4.34}$$

embedding all allowable tracking errors at the time instant $t_{N_{AB}}$ can be found by solving the subsequent SDP minimization problem

$$\min_{\tau_0, \cdots, \tau_{N_{AB}-1}, \lambda, \tilde{\gamma}_n} \tilde{\gamma}_n \tag{4.35}$$

s. t.

$$0 \leq \tilde{\gamma}_n \leq 1 \tag{4.36}$$

$$\tau_0, \cdots, \tau_{N_{AB}-1} \geq 0 \tag{4.37}$$

$$\lambda \geq 0 \tag{4.38}$$

$$\tilde{\gamma_n} - \sum_{h=0}^{N_{AB}-1} \tau_h d_{max}^2 - \lambda \geq 0 \tag{4.39}$$

$$\begin{bmatrix} -\bar{\Phi}_n^T P_0 \bar{\Phi}_n + \lambda_n P_A & -\bar{\Phi}_n^T P_0 \bar{\Phi}_n \\ * & -\bar{J}_n^T P_0 J_n + \sum_{h=0}^{N_{AB}-1} \tau_h \Lambda_h^T \Lambda_h \end{bmatrix} \geq 0 \tag{4.40}$$

where $\bar{\Phi}_n = \tilde{\Phi}_n^{N_{AB}}$,

$$\bar{J}_n = \left[\tilde{\Phi}_n^{N_{AB}-1} B_D \;\; \tilde{\Phi}_n^{N_{AB}-2} B_D \cdots B_D\right] \tag{4.41}$$

being Λ_h *a matrix of proper dimension such that* $\Lambda_h \hat{D} = d_h$

Proof - The condition

$$\xi_{N_{AB}}^T P_0 \xi_{N_{AB}} \leq \tilde{\gamma}_n \tag{4.42}$$

with $\xi_{N_{AB}} = \bar{\Phi}_n \xi_A + \bar{J}_n \hat{D}$ being $n = 1, \cdots, N_v$ can be expressed in the form

$$\left[1 \; \xi_A^T \; \hat{D}^T\right] \begin{bmatrix} \tilde{\gamma}_n & 0 & 0 \\ * & -\bar{\Phi_n}^T P_0 \bar{\Phi}_n & -\bar{\Phi_n}^T P_0 \bar{J}_n \\ * & * & -\bar{J}_n^T P_0 \bar{J}_n \end{bmatrix} \begin{bmatrix} 1 \\ \xi_A \\ \hat{D} \end{bmatrix} \geq 0 \tag{4.43}$$

.

Inclusion $\xi_A \in S_A$ can be promptly written as:

$$\left[1 \; \xi_A^T \; \hat{D}^T\right] \begin{bmatrix} 1 & 0 & 0 \\ * & -P_A & 0 \\ * & * & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \xi_A \\ \hat{D} \end{bmatrix} \geq 0 \tag{4.44}$$

while condition

$$(\Lambda_h \hat{D})^T (\Lambda_h \hat{D}) \leq d_{max}^2 \tag{4.45}$$

can be recast as:

$$\begin{bmatrix}1\ \xi_A^T\ \hat{D}^T\end{bmatrix}\begin{bmatrix}d_{max}^2 & 0 & 0\\ * & 0 & 0\\ * & * & -\Lambda_h^T\Lambda_h\end{bmatrix}\begin{bmatrix}1\\ \xi_A\\ \hat{D}\end{bmatrix}\geq 0 \quad h=0,\ldots,N_{AB}-1 \tag{4.46}$$

By exploiting the classical *S-procedure*, LMIs (4.35)-(4.40) can be promptly obtained. □

Eventually, be $\tilde{\gamma}$ the maximum value of $\tilde{\gamma}_n$. Following (4.32), $S_{N_{AB}}$ ellipsoidal set can be easily obtained. Following these premises, the feasibility condition can be introduced. Given (4.32), condition (4.29) is satisfied if direct (Minkowski) sum is included in the robust D-invariant region Γ_0:

$$S_{N_{AB}} \bigoplus \Pi \subseteq \Gamma_0 \tag{4.47}$$

This condition holds if there exists $\tau > 0$ such that the following LMI is feasible

$$\begin{bmatrix}1-\tau\tilde{\gamma}-\Pi^T P_0 \Pi & *\\ -P_0\Pi & -P_0+\tau P_0\end{bmatrix}\geq 0 \tag{4.48}$$

So, the existence of such strictly positive τ in (4.48) implies that T^{BC} can follow T^{AB} in a feasible trajectory.

Figure 4.3 gives a visual representation of the concept of feasibility as outlined so far. The figure proposes a 2-dimensional visual outline of a feasibility check. Since $S_{N_{AB}} \bigoplus \Pi_E \not\subseteq \Gamma_0$ the segment T^{BE} cannot follow $\overline{AB}$ in a feasible trajectory, while, given that $S_{N_{AB}} \bigoplus \Pi_C \not\subseteq \Gamma_0$ T^{BC} can follow T^{AB} in a feasible trajectory.

4.3 Feasible trajectory planning algorithm

Let consider two nodes C_S and C_F which represent starting S and ending F points, respectively. A^* algorithm [Dechter and Pearl, 1985; Helmert et al., 2008] is utilised to find a feasible shortest trajectory (if there is any) composed by a succession of segments to be cross at determined velocities from S to F.

To specify the heuristic function needed for A^* algorithm, let consider the trajectory connecting starting node C_S to a generic C_A node. c_{SA} represents the shortest succession of segments to traverse at assigned velocities from S to A. Length of c_{SA} is designated by $l(c_{SA})$. As

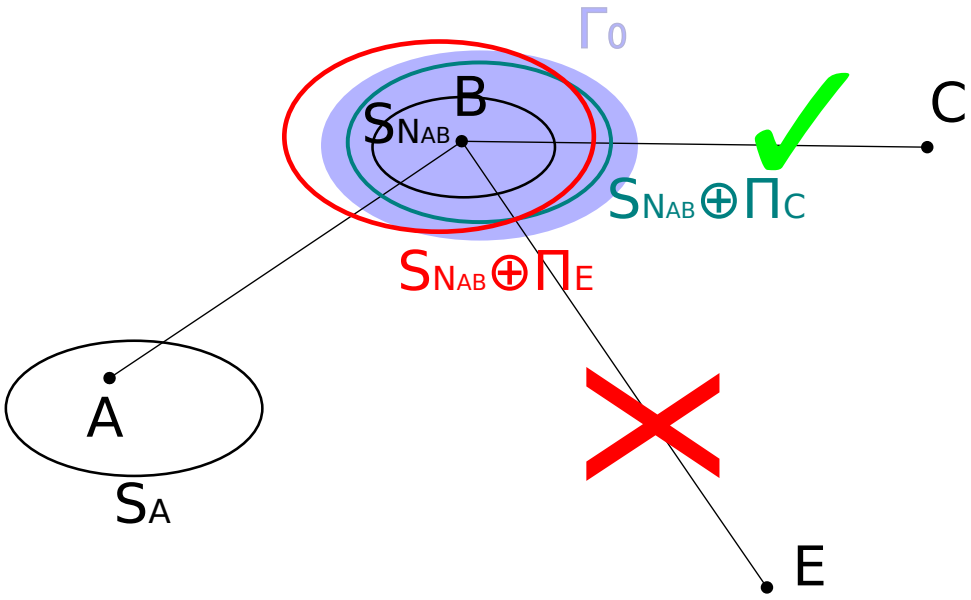


Fig. 4.3. A graphic representation of feasibility check. The 2D representation is merely explanatory. The subscripts of Π outline which segments they refer to.

written above, T^{AB} is the arc which connects C_A to C_B. The relevant heuristic function is expressed below

$$f(c_{SA}, T^{AB}, C_B, C_F) = l(c_{SA}) + l_{AB} + \|B - F\| \tag{4.49}$$

where $\|\cdot\|$ is the Euclidean norm. As required by $A*$, it is a lower bound in terms of length of trajectory linking C_S to C_F through c_{SA} and T^{AB}.

Given (4.49), the proposed approach for the feasible trajectory planning problem can be expressed in a recursive algorithm, as summarised in the following pseudo-code.

Algorithm 1

$c_{opt} = \varnothing$;
$c_{exp} = \varnothing$;
$C_A = C_S$;
function TRAJECTORYPLANNER($S_A, C_A, c_{opt}, c_{exp}$)
    $S_A \leftarrow$ initial tracking error as defined in (4.31);
    $C_A \leftarrow$ current node;
    $c_{opt} \leftarrow$ candidate optimal trajectory;
    $c_{exp} \leftarrow$ trajectory being explored;
    for all arcs T^{AB} starting from C_A **do**
        if $f(c_{exp}, T^{AB}, C_B, C_F)$ as defined in (4.49) $< l(c_{opt})$ **then**
            given S_A calculate $S_{N_{AB}}$ according to Theorem 4.2
            if $S_{N_{AB}} \bigoplus \Pi \subseteq \Gamma_0$ according to (4.48) **then**
                append T^{AB} to c_{exp}
                if C_B is the destination node **then**
                    $c_{opt} = c_{exp}$
                else
                    TRAJECTORYPLANNER($S_{N_{AB}}, C_B, c_{opt}, c_{exp}$)
                end if
            end if
        end if
    end forreturn c_{opt}
end function

5

Validation and results

The proposed approach has been validated through both numerical and experimental simulations. This chapter will offer an overview of the relevant methods along with the obtained results. For the sake of consistency, the physical, experimental, architecture has been considered in numerical simulations too.

5.1 Environment discretization

Let assume an occupation map of the operational environment. Such maps define the probability of the environment to be occupied by an obstacle, providing upper-level algorithms with information about mapping quality along with the presence of obstacles.
In the experimental setup used in this thesis, map is provided by AMCL ROS package as a 4096×4096 matrix of unsigned 8-bit integers, with a 2.5 cm sampling step. Each value represents the probability of a discretized point to be occupied by an obstacle, the value 205, as a convention, represents the lack of information.
In this thesis, for safety purpose, points closer to an obstacle than a given threshold are removed from the map, an example can be seen in figure 5.1, to be noted, these operations can be seamlessly carried out by the means of image processing technique e.g., border detection can be efficiently performed through Sobel convolution with a $3x3$ kernel. The map is then downsampled to achieve the desired sampling step. The graph shown in figure 5.2 has a 0.2 m resolution and includes 924 nodes.

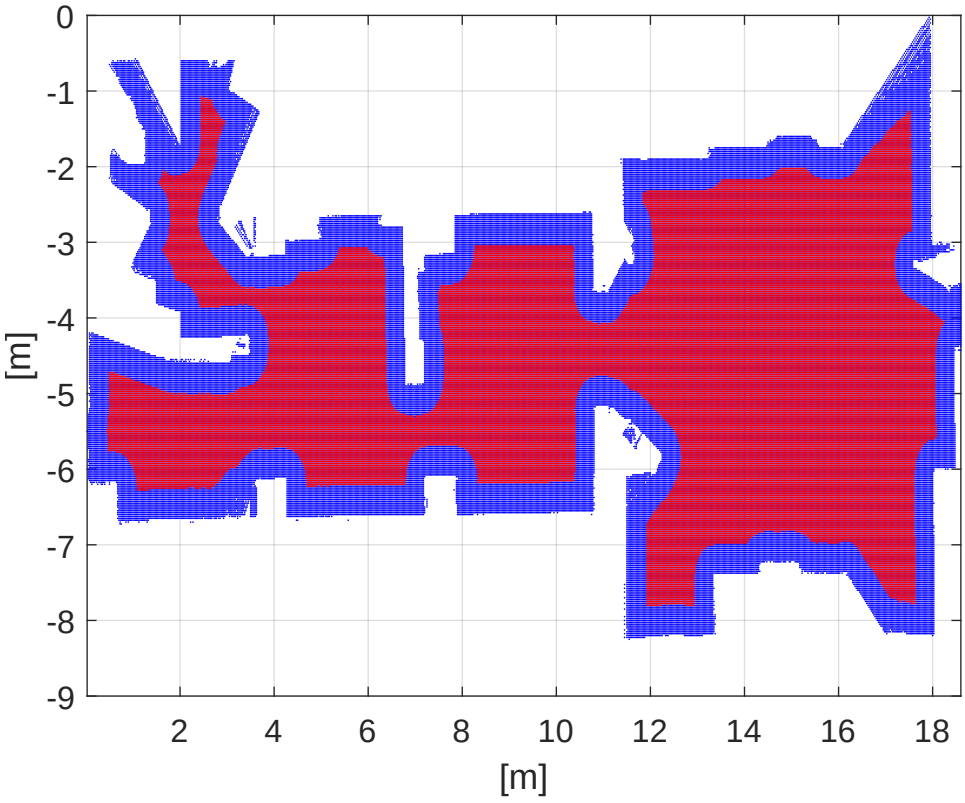


Fig. 5.1. An example of map as taken by the AMCL package, blue points are points which are discarded as too close to obstacles, while dark red area is safe area for robot operations.

5.2 Experimental architecture description

Dr. Robot Jaguar v4 is a skid-steer tracked mobile robot (see fig. 5.3) equipped with a manipulator arm, two Hokuyo UTM-30LN laser scanners, an 802.11b/g/n network, four 80 W-DC motors and a series of sensors. In order to perform an higher quality SLAM the robot has been equipped with a Raspberry Pi 3B unit, connected through USB to the laser scanners.
The vehicle is connected through a point-to-point Ubiquity networks Wi-Fi link to an Ethernet switch, which provides connectivity to an Ubuntu-powered laptop performing data logging e visualization through ROS and RViz environments, and a Windows PC running the MATLAB-powered control and planning routines.
The networking perspective of this architecture is depicted in figure 5.4.

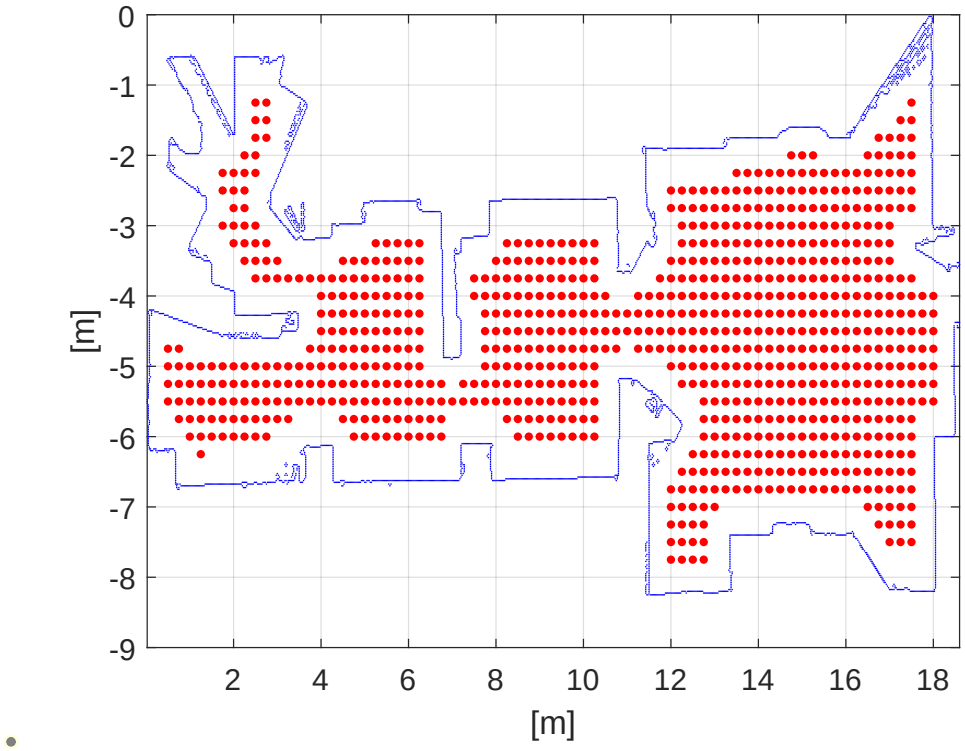


Fig. 5.2. The environment mapped by the SLAM procedure (blue borders) and the considered graph nodes of safe positions (red dots).

Fig. 5.3. Dr. Robot Jaguar v4 skid-steer tracked mobile robot.

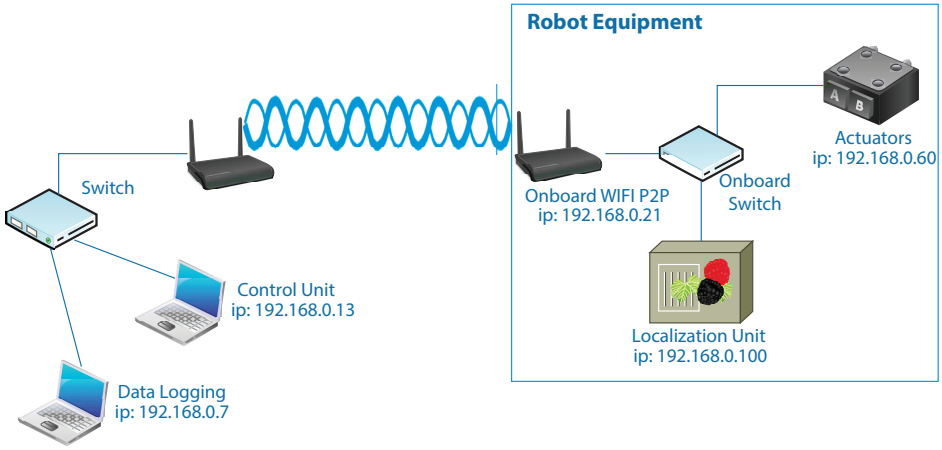


Fig. 5.4. Schematics of the experimental architecture.

The mobile robot has a radius of track sprocket R of 8 cm and a distance between left and right tracks D of 50 cm. The output of two (front and rear) laser-scanner sensors is used for localization and pose estimation through the widely used and robust Adaptive Monte Carlo Localization (AMCL) algorithm [Thrun et al., 2005] running on the onboard embedded computer Raspberry Pi 3 Model B, using the AMCL ROS.

A preliminary campaign for the evaluation of network delay has been carried out, by considering round-trip time of packets through the control loop in a TCP/IP socket through the application layer, thus taking into account delays induced by upper layers in network stack. Tests were carried out around the whole operational scenario, which is subject to heavy electro-magnetic interferences from concurrent 802.11 networks. An example of this analysis is given in figure 5.5.

Control action is computed on a remote ground-station, with a sampling time T_S of 0.2 s. The considered maximum and minimum network delay values are $\tau_{max} = 285.96\,ms$ $\tau_{min} = 104\,ms$ respectively. This takes into account not only round-trip time but also worst-case scenarios of SLAM delay (2.5 ms) and kernel and actuators latency.
Forward velocity to cross trajectory segments is equal to $0.25\,\mathrm{m\,s^{-1}}$, nominal rotation velocity is zero and the nominal values of friction coefficients μ_R and μ_L are both equal to 1. These values are summarized in table 5.1.

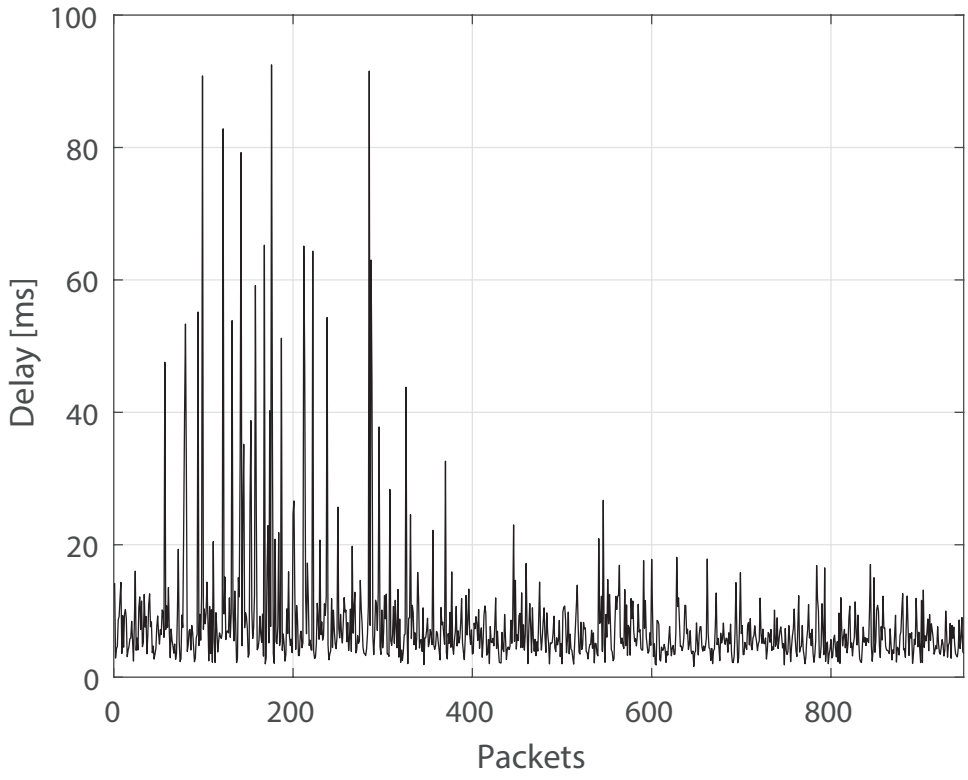


Fig. 5.5. Delays experienced by of a batch of TCP/IP packets in the experimental scenario.

Table 5.1. Model parameters

Symbol	Value	Unit	
T_s	200	ms	Sample Time
τ_{max}	285.96	ms	Maximum network delay
τ_{min}	104	ms	Minimum network delay
$\bar{V}$	0.25	m/s	Robot nominal forward velocity
$\bar{\omega}$	0	rad/s	Robot nominal rotational velocity
$\bar{\mu}_R$	1	–	Nominal right friction coefficient
$\bar{\mu}_L$	1	–	Nominal left friction coefficient

Following the procedure highlighted in section 3.2, a trajectory tracking error model with $n_s = 9$ states and $N_v = 48$ vertices is found.

A series of constraints, compatible with robot mechanical capabilities, were imposed. The relevant values are reported in table 5.2.

Following the procedure highlighted in 5.1 the space domain Δ shown in Figs.5.11 and 5.15 was discretized with a regular grid of 0.2 m generating a 202-points mesh. Graph G was constructed by imposing

Table 5.2. Prescribed constraints

Symbol	Min Value	Max Value	Unit	
V	0	0.5	m/s	Forward Velocity
ω	-0.6	0.6	rad/s	Rotational Velocity
$\mu_R(\mu_L)$	0.8	1.2	$-$	Right (left) friction coefficient

a maximum length of $\bar{L} = 0.5\,m$ for the trajectory segment connecting two $\Delta-$compatible nodes.

5.3 Numerical simulations

5.3.1 Trajectory planning sensitivity analysis

Some preliminary numerical results are now introduced. Let assume different ranges for allowable skid and slip-induced disturbance. Different ranges result in different robust positively invariant set (as introduced in 4.1), whose influence on trajectory planning can be seen in fig. 5.6. Starting from the same pose, four different trajectories are obtained. Robot initial and final poses are denoted by arrows. Blue line represents the optimal feasible trajectory in absence of skid and slip phenomena while black, red, and green lines represent are the optimal trajectories obtained by expanding the range of friction coefficients.

Predictably, the algorithm counterbalances the effects of increased disturbance by choosing longer and smoother trajectories at expanding of allowable bounds of friction coefficients. In other words, bounds upon control actions are respected by choosing trajectories whose tracking error can be controlled to zero -by the given controller- without requiring an incompatible control action.

5.3.2 First simulated scenario

Figs. 5.7-5.10 pertains to a simulation scenario whose range for μ_R and μ_L coefficients is reported in Table 5.2. The optimal trajectory, shown in Fig. 5.7, consists of 24 segments with an overall length of $7.64\,m$. Red lines denote the optimal feasible trajectory found by the algorithm. This trajectory guarantees that, for any admissible value of friction coefficients, starting from its starting point (red square), the robot can reach the ending point (red circle), while respecting a safety distance with obstacles.

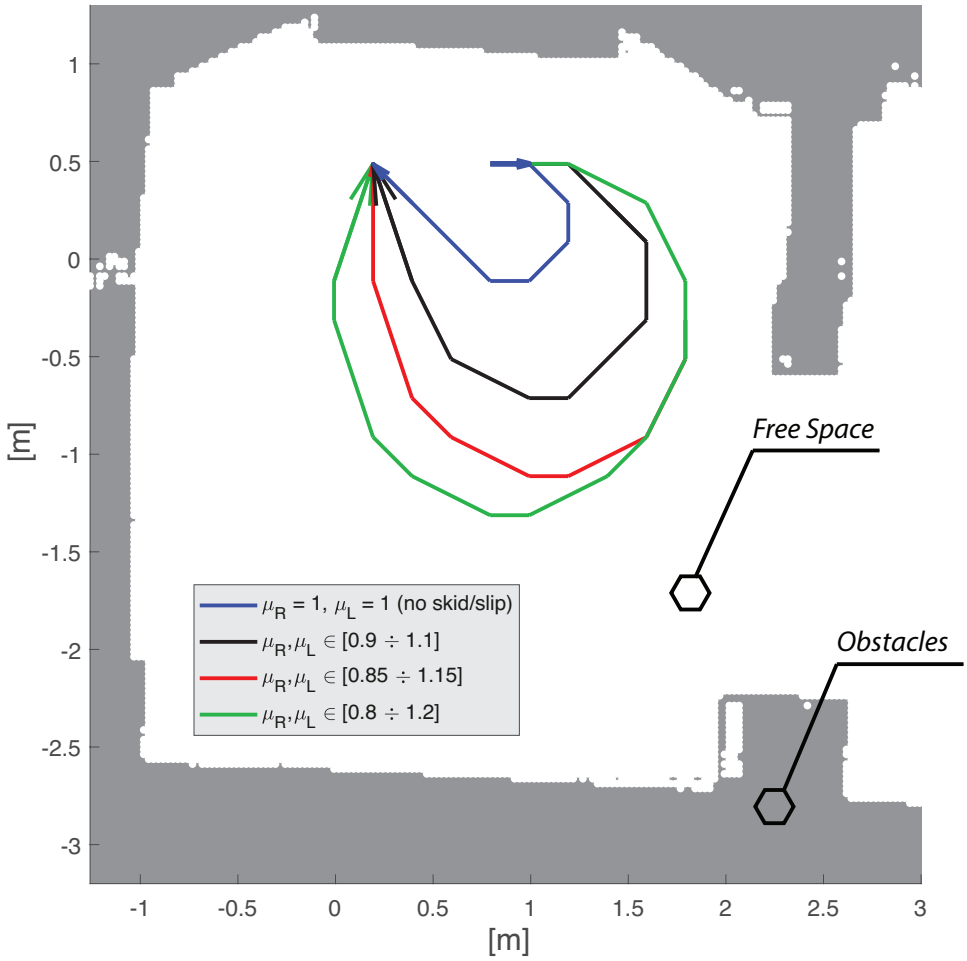


Fig. 5.6. Different optimal trajectories found by the algorithm in a sensitivity analysis at varying of allowable range of friction coefficients.

Several robot trajectories were simulated at varying of robot initial pose according to (4.1). Trajectories taken by the robot during different simulation runs are represented by black lines. Black triangles represent robot pose at $2\,s$ intervals. Figs. 5.8 and 5.9 show relevant robot forward and rotational speeds, respectively. Prescribed constraints are represented by red solid lines. Finally, Fig.5.10 illustrates the values of the Lyapunov function introduced in (4.1). As expected, the discrete-time Lyapunov function never exceeds one, proving that system state never leaves the robust positively invariant region associated with the control law.

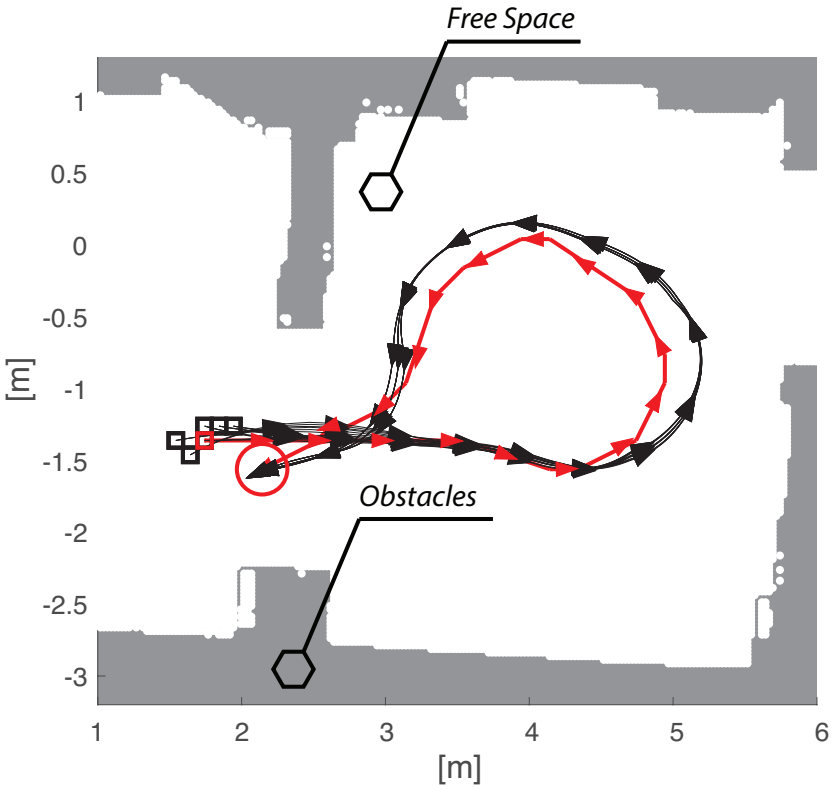


Fig. 5.7. Black solid lines represent mobile robot trajectories obtained at varying robot initial pose. Red line denotes the optimal feasible trajectory connecting starting (square) and ending (circle) points. Triangles represent robot simulated (black) and planned (red) poses taken each 2 *s*.

5.4 Experimental validation

Two experimental scenarios are here proposed, relevant figures span from Fig. 5.11 to Fig. 5.18. Scenarios try mimicking two kinds of manoeuvres which can be found in real-world contexts: moving in an obstacle-filled scenario to a non-line-of-sight destination and a *parking* manoeuvre, where the initial pose of the vehicle causes a 180Â°-degree turn.

5.4.1 First experimental scenario

Scenario I - Jaguar robot must track the planned trajectory shown in red in Fig. 5.11 at the nominal cruise speed of $0.25\,\mathrm{m\,s^{-1}}$. Opti-

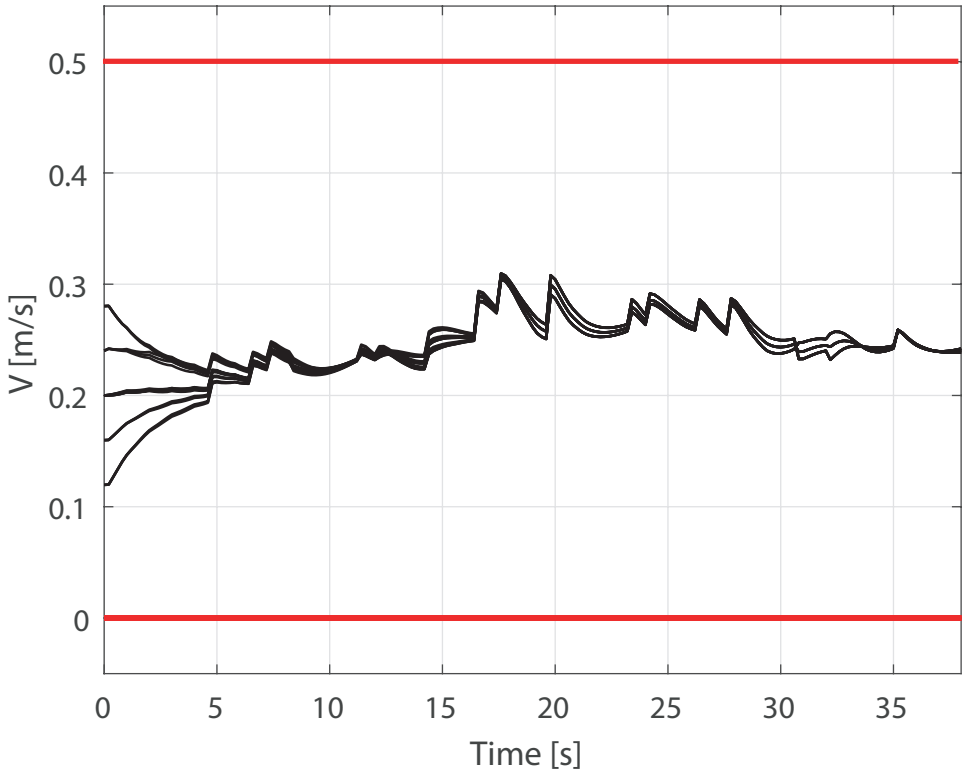


Fig. 5.8. Constraints are represented by red solid lines, while black lines represent robot forward velocities during different simulated runs.

mal planned trajectory is composed of 19 segments for a length of 10.7 m. Black lines represent experimental mobile robot trajectories obtained at varying of initial poses. Square indicates starting point of optimal feasible trajectory whereas circle the ending point. Control inputs are reported in figs.5.12 and 5.13 which show forward and angular robot control velocities, respectively. Red lines represent prescribed constraints. Fig. 5.14 shows the evolution of Lyapunov function (4.1) during each experimental simulation. As expected, since the optimal trajectory planned is feasible, it never exceeds the maximum allowable value (red line).

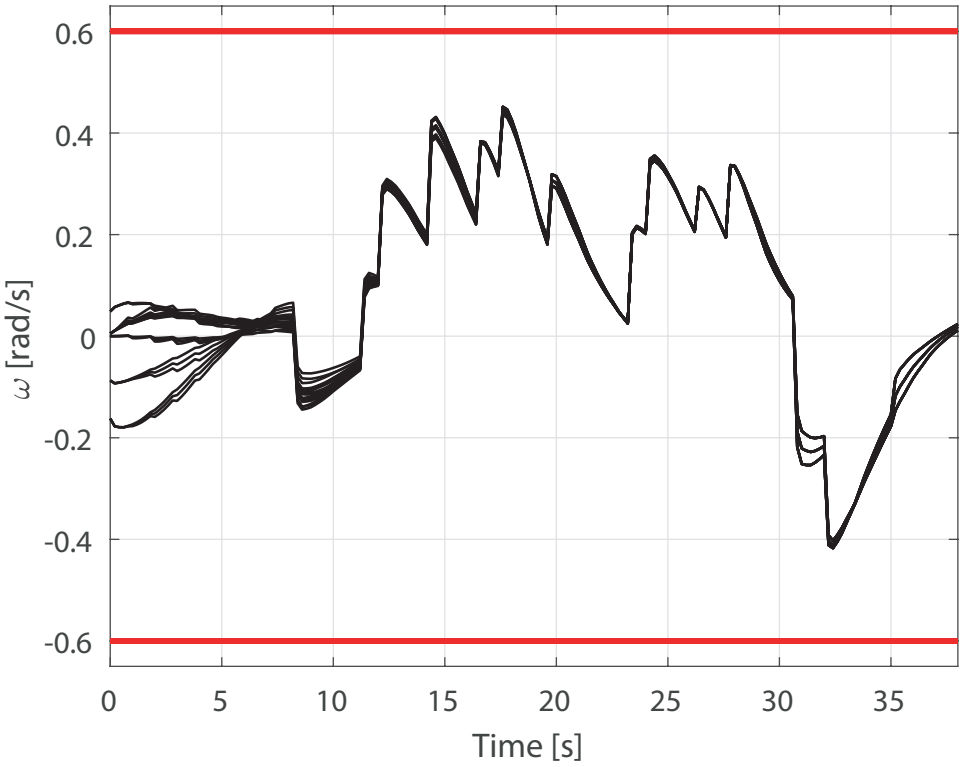


Fig. 5.9. Constraints are represented by red solid lines, while black lines represent robot angular velocities during different simulated runs.

5.4.2 Second experimental scenario

Scenario II - In this experimental simulation, trajectory planning algorithm constructs an optimal feasible trajectory, which is compliant with prescribed constraints, to reach the final pose by performing a 180−degree turn in a small space. The optimal feasible trajectory is shown in Fig. 5.15, see red line. Nominal robot advancing speed is $0.25\,\mathrm{m\,s^{-1}}$. Trajectory consists of 18 segments for an overall length of 6.3 m. The square mark denotes the starting point of optimal feasible trajectory whereas the ending point is marked by a circle. Black lines represent experimental mobile robot trajectories obtained at varying of initial pose. In Figs.5.15 and 5.17 are depicted forward and angular robot control velocities respectively. As shown in Fig. 5.18, in all experimental simulations, Lyapunov function (4.1) reaches almost the

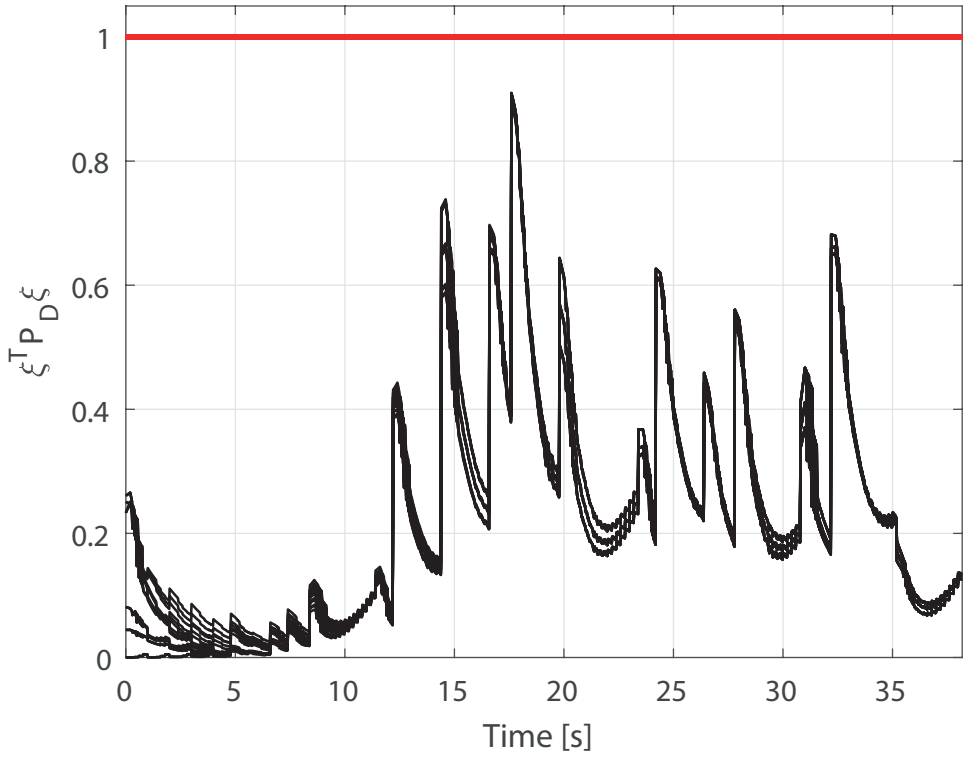


Fig. 5.10. Values of Lyapunov function (4.1) during different simulated runs.

maximum allowable value (red lines) when the robot experiences the maximum skidding phenomena during the 180−degree turn.

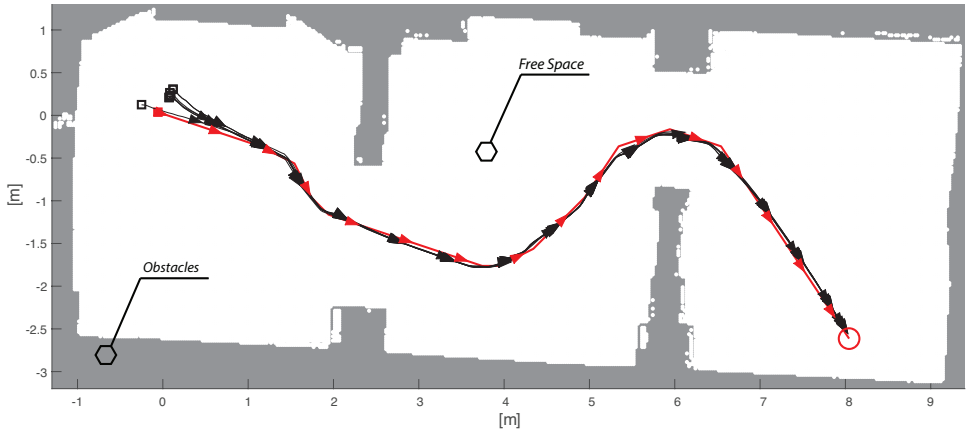


Fig. 5.11. Red line denotes the optimal feasible trajectory connecting starting (square) and ending (circle) points. Black solid lines represent mobile robot trajectories obtained at varying robot initial pose. Triangles represent robot planned (red) and experimental (black) poses taken each 3 s.

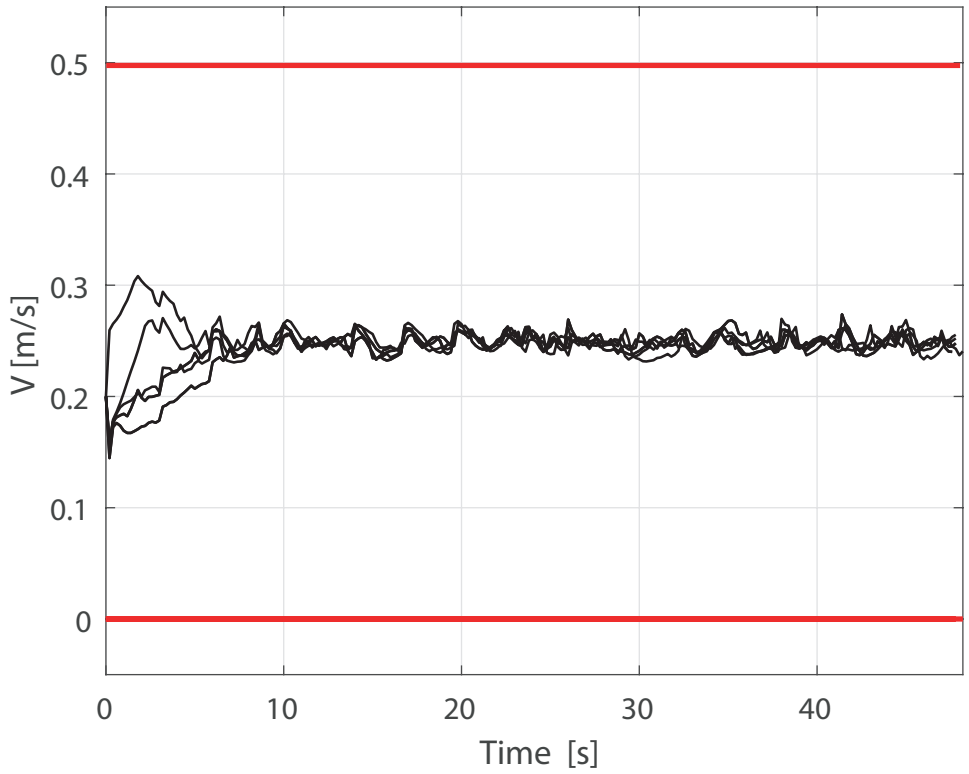


Fig. 5.12. Red lines represent considered actuation constraints. Black lines represent robot forward velocities during different runs in Scenario-I.

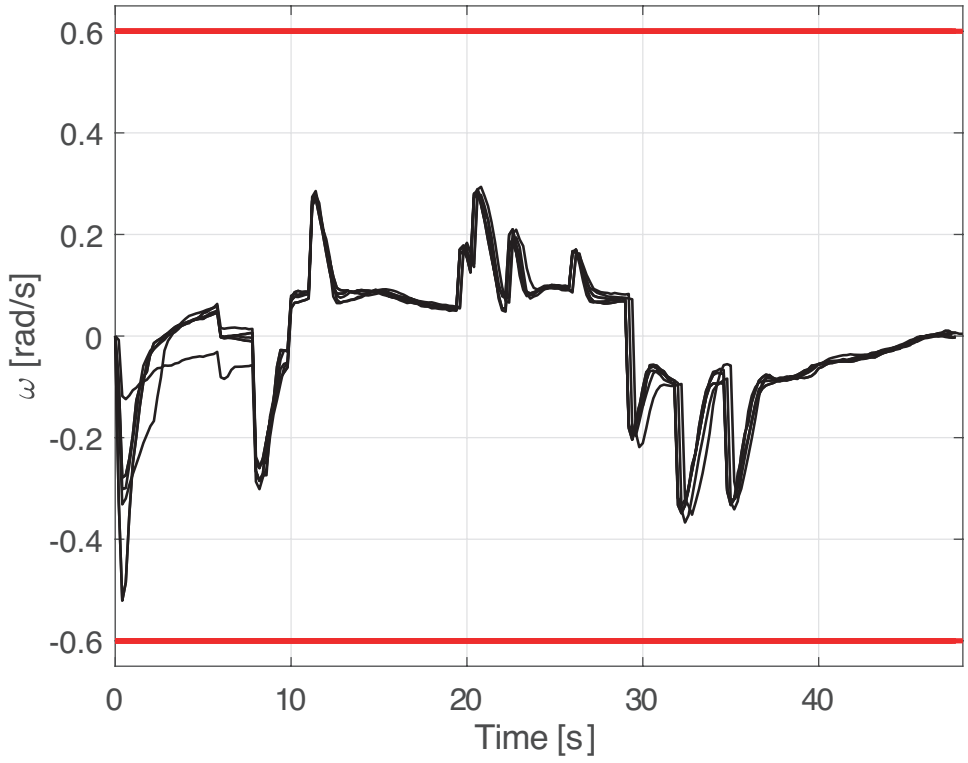


Fig. 5.13. Red lines represent considered actuation constraints. Black lines represent robot angular velocities during different runs in Scenario-I.

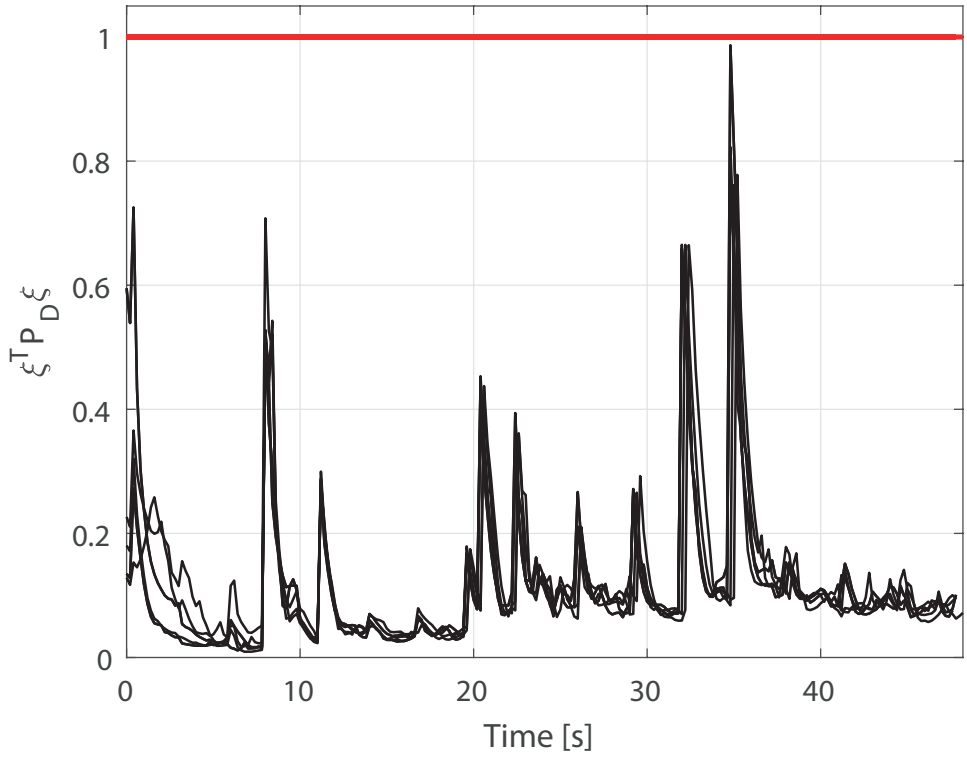


Fig. 5.14. Values of Lyapunov function (4.1) during different runs in Scenario I.

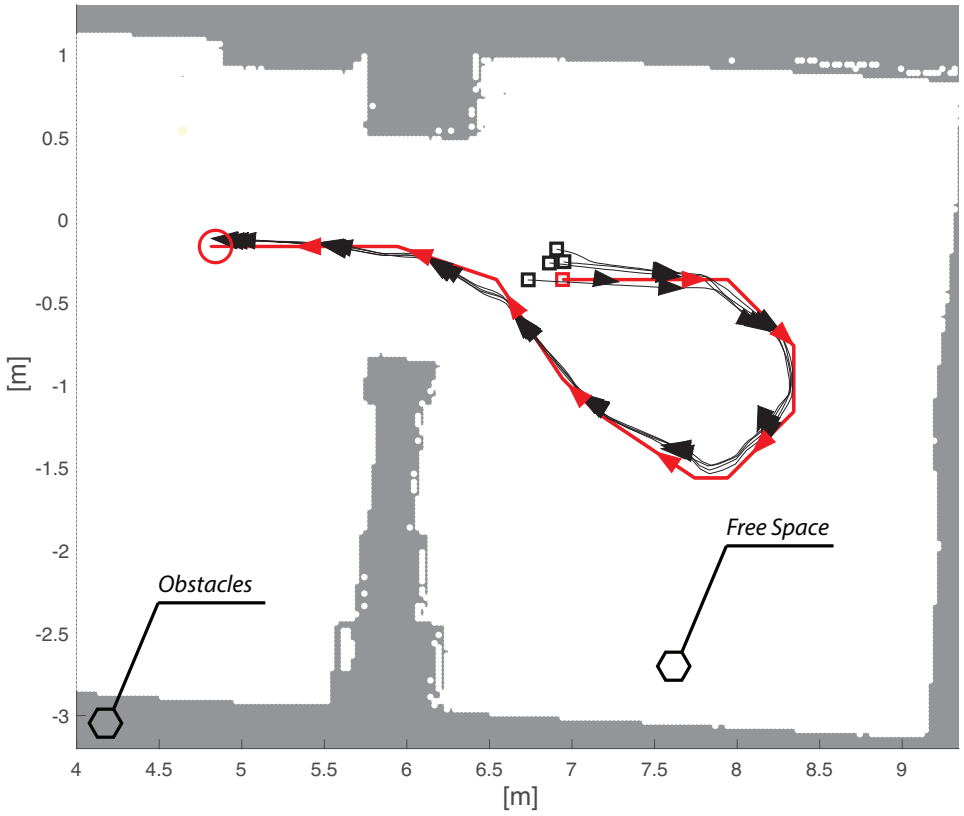


Fig. 5.15. Red line denotes the optimal feasible trajectory from the starting (square) to the ending (circle) point. Black solid lines represent mobile robot trajectories obtained at varying robot initial pose. Triangles represent robot planned (red) and experimental (black) poses taken each 3 s.

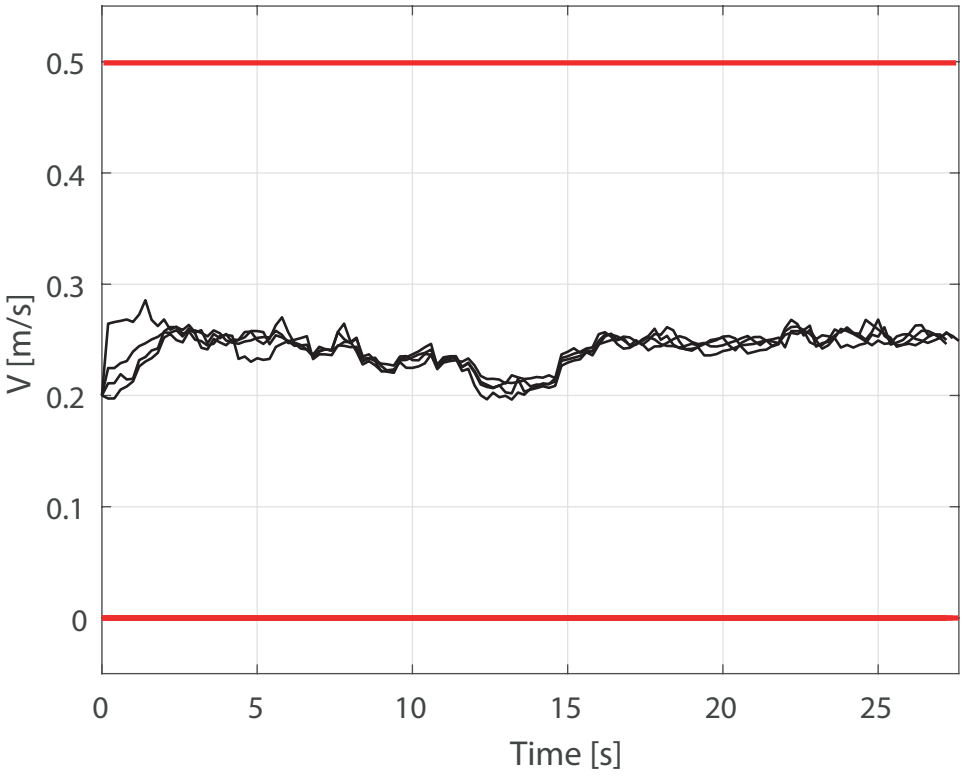


Fig. 5.16. Black lines represent robot forward velocities during different runs in Scenario-II. Red lines represent considered actuation constraints.

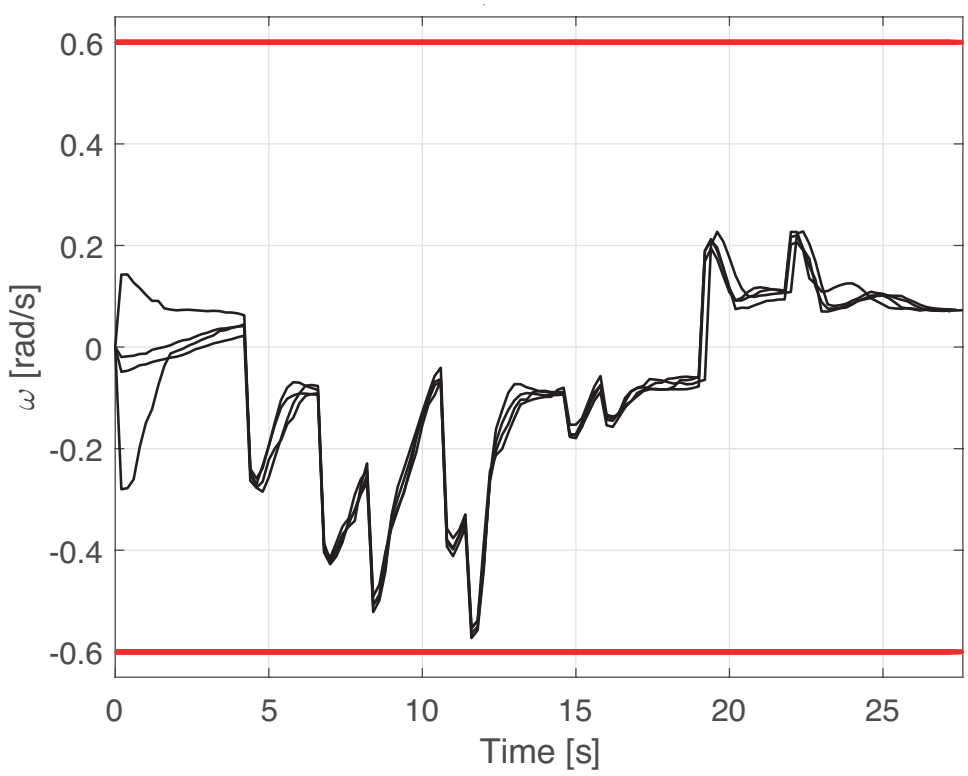


Fig. 5.17. Black lines represent robot angular velocities during different runs in Scenario-II. Red lines represent considered actuation constraints.

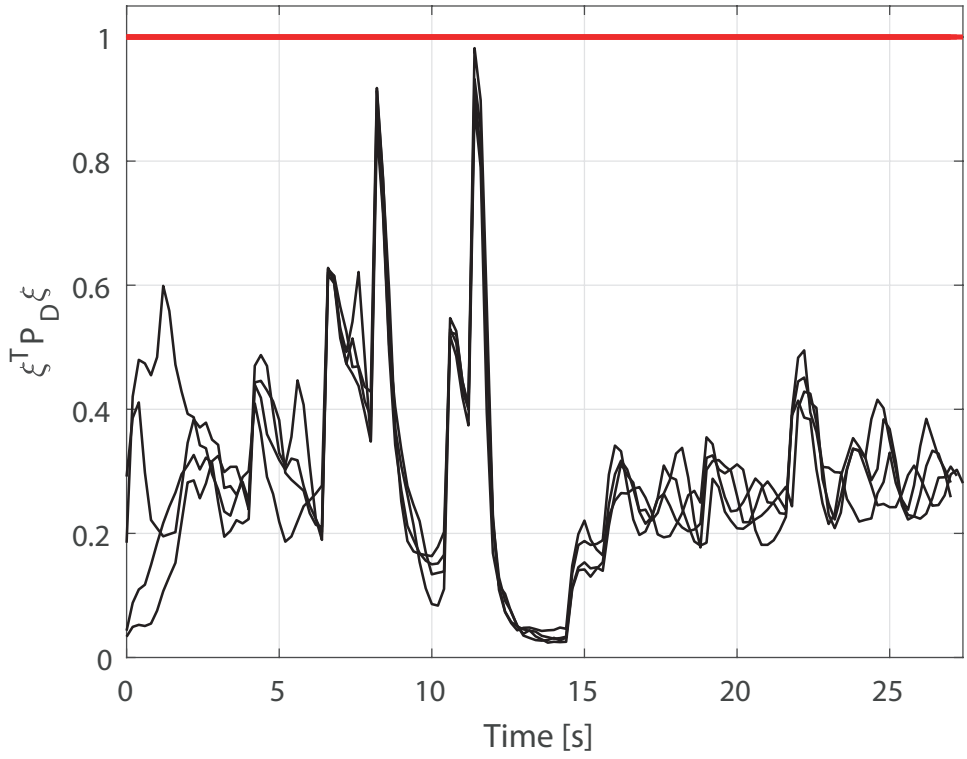


Fig. 5.18. Values of Lyapunov function (4.1) during different runs in Scenario II.

Conclusions

This thesis tackles the problem of trajectory planning for a vehicle taking into account its closed-loop dynamics. The proposed solution is applied to a skid-steered tracked mobile robot for its experimental validation and can handle both model uncertainty due to a control loop which includes a stochastic communication network and disturbance arising from skid and slip phenomena due to locomotion schema.
The proposed solution is a framework composed by a series of elements which can take into account closed-loop dynamics of the vehicle.
Namely, it includes the solution of a constrained control problem leading to a static-feedback control law which is associated with a positively invariant ellipsoidal region, embedding tracking errors which can be controlled to zero without violating actuation constraints.
Using set-based argument, the property of disturbance invariance of this region is tested. This allows to define a feasibility check for candidate trajectory segments, which are deemed feasible if the ellipsoidal embedding of the possible tracking error at their end is included within the robust positively disturbance-invariant region associated to control law. This procedure relies upon the solution to a series of semi-definite programming problems defined in terms of linear matrix inequalities.
A trajectory is thus defined as a series of connected segments which are associated to the arcs of a graph whose nodes represent discrete safe position of the operational environment.
The thesis includes the following contents: chapter 1 which offers an overview of mobile robotics. Chapter 2 which presents the problem of motion planning for a terrestrial mobile robot along with solutions featured in literature. Chapter 3 where a kinematic model of the skid-steer tracked mobile robot used in this work is derived and the problem of

remote control is tackled. Chapter 4 dealing with the synthesis of a control law, feasibility check and planning algorithm. Finally, chapter 5 which provides and discusses a series of numerical and experimental simulations, carried out on a Jaguar v4 robot platform, with the aim to prove the effectiveness of the proposed approach.
Results obtained proved that at varying of robot initial pose actuation constraints were never violated and robot state never left robust positively disturbance-invariant region.

References

Abate, A. and El Ghaoui, L. (2004). Robust model predictive control through adjustable variables: an application to path planning. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 3, pages 2485–2490. IEEE.

Afonso, R. J. M., Galvao, R. K. H., and Kienitz, K. H. (2013). Waypoint trajectory planning in the presence of obstacles with a tunnel-milp approach. In *2013 European Control Conference (ECC)*, pages 1390–1397. IEEE.

Agarwal, P. K., Biedl, T., Lazard, S., Robbins, S., Suri, S., and Whitesides, S. (2002). Curvature-constrained shortest paths in a convex polygon. *SIAM Journal on Computing*, 31(6):1814–1851.

Ahuactzin, J. M., Talbi, E.-G., Bessiere, P., and Mazer, E. (1991). Using genetic algorithms for robot motion planning. In *Workshop on geometric reasoning for perception and action*, pages 84–93. Springer.

Albore, A., Doose, D., Grand, C., Lesire, C., and Manecy, A. (2021). Skill-based architecture development for online mission reconfiguration and failure management. In *2021 IEEE/ACM 3rd International Workshop on Robotics Software Engineering (RoSE)*, pages 47–54.

Alwi, H. and Edwards, C. (2006). Robust sensor fault estimation for tolerant control of a civil aircraft using sliding modes. In *American Control Conference*, pages 5704–5709.

Antonelli, G. (2003). *A Survey of Fault Detection/Tolerance Strategies for AUVs and ROVs*, pages 109–127. Springer Berlin Heidelberg, Berlin, Heidelberg.

Antsaklis, P. and Baillieul, J. (2004). Guest editorial special issue on networked control systems. *IEEE Transactions on Automatic Control*, 49(9):1421–1423.

Arai, K., Sugawara, W., and Honda, T. (1995). Magnetic small flying machines. In *Proceedings of the International Solid-State Sensors and Actuators Conference-TRANSDUCERS'95*, volume 1, pages 316–319. IEEE.

Arcara, P. and Melchiorri, C. (2002). Control schemes for teleoperation with time delay: A comparative study. *Robotics and Autonomous systems*, 38(1):49–64.

Ariola, M., Corraro, F., Mattei, M., Notaro, I., and Sollazzo, A. (2013). An sfdi observer-based scheme for a general aviation aircraft. In *2013 Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 152–157. IEEE.

Arjomandi, M., Agostino, S., Mammone, M., Nelson, M., and Zhou, T. (2006). Classification of unmanned aerial vehicles. *Report for Mechanical Engineering class, University of Adelaide, Adelaide, Australia*.

Arrichiello, F. (2006). Coordination control of multiple mobile robots. *Dipartimento di Automazione, Elettromagnetismo, Ingegneria Dell'informazione e Matematica Industriale*.

Asama, H., Matsumoto, A., and Ishida, Y. (1989). Design of an autonomous and distributed robot system: Actress. In *IROS*, volume 89, pages 283–290.

Asensio, J. and Montano, L. (2002). A kinematic and dynamic model-based motion controller for mobile robots. *IFAC Proceedings Volumes*, 35(1):427–432. 15th IFAC World Congress.

Aulinas, J., Petillot, Y. R., Salvi, J., and Lladó, X. (2008). The slam problem: a survey. *CCIA*, 184(1):363–371.

Aydin, S. and Temeltas, H. (2002). A novel approach to smooth trajectory planning of a mobile robot. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623)*, pages 472–477.

Balch, T. and Arkin, R. C. (1998). Behavior-based formation control for multirobot teams. *IEEE transactions on robotics and automation*, 14(6):926–939.

Bemporad, A., Di Cairano, S., Henriksson, E., and Johansson, K. H. (2010a). Hybrid model predictive control based on wireless sensor feedback: An experimental study. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 20(2):209–225.

Bemporad, A., Heemels, M., and Vejdemo-Johansson, M. (2010b). *Networked Control Systems*. Lecture Notes in Control and Information

Sciences. Springer London.

Berdjag, D., Cieslak, J., and Zolghadri, A. (2012). Fault diagnosis and monitoring of oscillatory failure case in aircraft inertial system. *Control Engineering Practice*, 20(12):1410–1425.

Bernardini, D., Donkers, M., Bemporad, A., and Heemels, W. (2010). A model predictive control approach for stochastic networked control systems. *IFAC Proceedings Volumes*, 43(19):7–12.

Berntorp, K., Weiss, A., Danielson, C., Kolmanovsky, I. V., and Di Cairano, S. (2017). Automated driving: Safe motion planning using positively invariant sets. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6.

Blanchini, F. (1992). Constrained control for systems with unknown disturbances. In LEONDES, C., editor, *Robust Control System Techniques and Applications, Part 2 of 2*, volume 51 of *Control and Dynamic Systems*, pages 129–182. Academic Press.

Blanchini, F. (1999). Set invariance in control. *Automatica*, 35(11):1747–1767.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. SIAM.

Breivik, M. (2010). *Topics in Guided Motion Control of Marine Vehicles*. PhD thesis.

Brezak, M. and Petrović, I. (2011). Time-optimal trajectory planning along predefined path for mobile robots with velocity and acceleration constraints. In *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 942–947.

Brockman, M. L. and Corless, M. (1998). Quadratic boundedness of nominally linear systems. *International Journal of Control*, 71(6):1105–1117.

Bruyninckx, H. and Reynaerts, D. (1997). Path planning for mobile and hyper-redundant robots using pythagorean hodograph curves. In *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*, pages 595–600.

Buonanno, A., D'Urso, M., Mattei, M., Meliadó, E. F., and Scordamaglia, V. (2011). Robotic sensor networks for security. In *2011 IEEE International Workshop on Measurements and Networking Proceedings (M N)*, pages 12–15.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. E. (2005). Coordinated multi-robot exploration. *IEEE Transactions on robotics*, 21(3):376–386.

Caccia, M., Bono, R., Bruzzone, G., Spirandelli, E., Veruggio, G., Stortini, A., and Capodaglio, G. (2005). Sampling sea surfaces with sesamo: An autonomous craft for the study of sea-air interactions. *Robotics & Automation Magazine, IEEE*, 12:95 – 105.

Calisti, M., Picardi, G., and Laschi, C. (2017). Fundamentals of soft robot locomotion. *Journal of The Royal Society Interface*, 14(130):20170101.

Cariou, C., Lenain, R., Thuilot, B., and Berducat, M. (2009). Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. *Journal of Field Robotics*, 26(6-7):504–518.

Carmi, A. and Oshman, Y. (2009). Adaptive particle filtering for spacecraft attitude estimation from vector observations. *Journal of Guidance Control and Dynamics*, 32:232–241.

Carts-Powell, Y. (2000). Tiny camera in a pill extends limits of endoscopy. *SPIE: OE-Reports*, (200).

Caruntu, C. F., Lazar, M., Gielen, R. H., Van den Bosch, P., and Di Cairano, S. (2013). Lyapunov based predictive control of vehicle drivetrains over can. *Control Engineering Practice*, 21(12):1884–1898.

Casalino, G., Turetta, A., and Simetti, E. (2009). A three-layered architecture for real time path planning and obstacle avoidance for surveillance usvs operating in harbour fields. pages 1 – 8.

Chagas, R. A. J. and Waldmann, J. (2012). Rao-Blackwellized particle filter with vector observations for satellite three-axis attitude estimation and control in a simulated testbed. *Sba: Controle & Automação Sociedade Brasileira de Automatica*, 23:277 – 293.

Chaimowicz, L., Grocholsky, B., Keller, J. F., Kumar, V., and Taylor, C. J. (2004). Experiments in multirobot air-ground coordination. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 4053–4058. IEEE.

Chang, Y.-C. and Yamamoto, Y. (2009). Path planning of wheeled mobile robot with simultaneous free space locating capability. *Intelligent Service Robotics*, 2(1):9–22.

Chen, C., Li, H.-X., and Dong, D. (2008). Hybrid control for robot navigation - a hierarchical q-learning algorithm. *IEEE Robotics Automation Magazine*, 15(2):37–47.

Cheng, Y. and Crassidis, J. (2004). *Particle Filtering for Sequential Spacecraft Attitude Estimation*. AIAA.

Choset, H. and Nagatani, K. (2001). Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on robotics and automation*, 17(2):125–137.

Choset, H., Walker, S., Eiamsa-Ard, K., and Burdick, J. (2000). Sensor-based exploration: Incremental construction of the hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19(2):126–148.

Christ, R. and Wernli, R. (2008). Chapter 10 - underwater vehicles. In Molland, A. F., editor, *The Maritime Engineering Reference Book*, pages 728–783. Butterworth-Heinemann, Oxford.

Clark, R. N. (1978). Instrument fault detection. *IEEE Transactions on Aerospace and Electronic Systems*, (3):456–465.

Clark, R. N., Fosth, D. C., and Walton, V. M. (1975). Detecting instrument malfunctions in control systems. *IEEE Trans. Aerosp. Electron. Syst.*, 11:465–473.

Cloosterman, M., Hetel, L., van de Wouw, N., Heemels, W., Daafouz, J., and Nijmeijer, H. (2010a). Controller synthesis for networked control systems. *Automatica*, 46(10):1584 – 1594.

Cloosterman, M. B., Hetel, L., Van de Wouw, N., Heemels, W., Daafouz, J., and Nijmeijer, H. (2010b). Controller synthesis for networked control systems. *Automatica*, 46(10):1584–1594.

Corradini, M., Leo, T., and Orlando, G. (1999). obust stabilization of a mobile robot violating the nonholonomic constraint via quasi-sliding modes. In *Proceedings of the Amer- ican Control Conference*.

Crassidis, J., Markley, L., and Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance Control and Dynamics*, 30:12–28.

Curcio, J., Leonard, J., and Patrikalakis, A. (2005). Scout - a low cost autonomous surface platform for research in cooperative autonomy. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 725–729 Vol. 1.

Dahl, O. and Nielsen, L. (1990). Torque-limited path following by online trajectory time scaling. *IEEE Transactions on Robotics and Automation*, 6(5):554–561.

D'Amato, E., De Capua, C., Filianoti, P. G., Gurnari, L., Nardi, V. A., Notaro, I., and Scordamaglia, V. (2021a). Ukf-based fault detection and isolation algorithm for imu sensors of unmanned underwater vehicles. In *2021 IEEE Workshop on metrology for the sea (METROSEA)*. IEEE.

D'Amato, E., Nardi, V. A., Notaro, I., and Scordamaglia, V. (2021b). A visibility graph approach for path planning and real-time collision avoidance on maritime unmanned systems. In *2021 IEEE Workshop on metrology for the sea (METROSEA)*. IEEE.

Danielson, C., Berntorp, K., Di Cairano, S., and Weiss, A. (2020a). Motion-planning for unicycles using the invariant-set motion-planner. In *Proceedings of the American Control Conference*.

Danielson, C., Berntorp, K., Weiss, A., and Di Cairano, S. (2020b). Robust motion planning for uncertain systems with disturbances using the invariant-set motion planner. *IEEE Transactions on Automatic Control*, PP:1–1.

Dearden, R. and Clancy, D. (2002). Particle filters for real-time fault detection in planetary rovers. Technical report, National Aeronautics and Space Administration.

Dechter, R. and Pearl, J. (1985). Generalized best-first search strategies and the optimality of a*. *J. ACM*, 32(3):505–536.

Depraetere, B., Liu, M., Pinte, G., Grondman, I., and BabuÅ¡ka, R. (2014). Comparison of model-free and model-based methods for time optimal hit control of a badminton robot. *Mechatronics*, 24(8):1021 – 1030.

Di Cairano, S., Kalabić, U., and Berntorp, K. (2016). Vehicle tracking control on piecewise-clothoidal trajectories by mpc with guaranteed error bounds. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 709–714. IEEE.

Dogru, S. and Marques, L. (2017). Estimation of rotational speeds of skid-steered wheeled mobile robots using an improved kinematic model. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 73–78.

Duan, Z., Cai, Z., and Yu, J. (2006). Adaptive particle filter for unknown fault detection of wheeled mobile robots. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1312–1315.

Dudek, G., Jenkin, M. R., Milios, E., and Wilkes, D. (1996). A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4):375–397.

Eldershaw, C. and Cameron, S. (1999). Real-world applications: Motion planning using gas. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, pages 1776–1776.

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3):249–265.

Endo, D., Okada, Y., Nagatani, K., and Yoshida, K. (2007). Path following control for tracked vehicles based on slip-compensating odometry. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2871–2876.

Espinosa, F., García, R., Mazo, M., López, E., and Mateos, R. (1998). Modelling and simulation of the kinematic and dynamic behavior of a fork-lift-truck. *IFAC Proceedings Volumes*, 31(3):51–55.

Fahimi, F., Ashrafiuon, H., and Nataraj, C. (2003). Obstacle avoidance for spatial hyper-redundant manipulators using harmonic potential functions and the mode shape technique. *Journal of Robotic Systems*, 20(1):23–33.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *IEEE Transactions on control systems technology*, 15(3):566–580.

Ferrera, M., Moras, J., Trouvé-Peloux, P., and Creuze, V. (2019). Real-time monocular visual odometry for turbid and dynamic underwater environments. *Sensors*, 19(3):687.

Feynman, R. P. (2018). *Thereâ€™s plenty of room at the bottom: An invitation to enter a new field of physics*. CRC Press.

Fierro, R. and Lewis, F. L. (1997). Control of a nonholomic mobile robot: Backstepping kinematics into dynamics. *Journal of robotic systems*, 14(3):149–163.

Fliess, M. and Join, C. (2013). Model-free control. *International Journal of Control*, 86(12):2228–2252.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Fox, D., Burgard, W., Thrun, S., and Cremers, A. B. (1998). A hybrid collision avoidance method for mobile robots. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1238–1243. IEEE.

Fraichard, T. and Ahuactzin, J.-M. (2001). Smooth path planning for cars. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3722–3727. IEEE.

Fujimura, K. (2012). *Motion planning in dynamic environments*. Springer Science & Business Media.

Fukuda, T., Kawamoto, A., Arai, F., and Matsuura, H. (1994). Mechanism and swimming experiment of micro mobile robot in water. In

Proceedings of the 1994 IEEE International Conference on Robotics and Automation, pages 814–819. IEEE.

Fukuda, T., Nakagawa, S., Kawauchi, Y., and Buss, M. (1989). Structure decision method for self organising robots based on cell structures-cebot. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 695–700. IEEE.

Gao, Y. (2016). *Contemporary Planetary Robotics: An Approach Toward Autonomous Systems*. Wiley-VCH, 1st edition.

Gao, Y. and Bando, Y. (2002). Carbon nanothermometer containing gallium. *Nature*, 415(6872):599–599.

Gao, Y., Gray, A., Tseng, H. E., and Borrelli, F. (2014). A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823.

Gavrilova, M., Cortes, J., and Jarvis, R. (2008). Computational geometry in navigation and path planning [from the guest editors]. *IEEE Robotics & Automation Magazine*, 15(2):6–7.

Geiger, B. R., Horn, J. F., Sinsley, G. L., Ross, J. A., Long, L. N., and Niessner, A. F. (2008). Flight testing a real-time direct collocation path planner. *Journal of guidance, control, and dynamics*, 31(6):1575–1586.

Gerkey, B. P. and Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954.

Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York.

Ghosh, S. K., Burdick, J. W., Bhattacharya, A., and Sarkar, S. (2008). Online algorithms with discrete visibility - exploring unknown polygonal environments. *IEEE Robotics Automation Magazine*, 15(2):67–76.

González-García, J., Gómez-Espinosa, A., Cuan-Urquizo, E., García-Valdovinos, L. G., Salgado-Jiménez, T., and Cabello, J. A. E. (2020). Autonomous underwater vehicles: Localization, navigation, and communication for collaborative missions. *Applied Sciences*, 10(4).

Goupil, P. (2011). AIRBUS state of the art and practices on FDI and FTC in flight control system. *Control Engineering Practice*, 19:524–539.

Graver, J., Bachmayer, R., Leonard, N., and Fratantoni, D. (2003). Underwater glider model parameter identification.

Griswold, N. C. and Eem, J. (1990). Control for mobile robots in the presence of moving objects. *IEEE Transactions on Robotics and Automation*, 6(2):263–268.

Gutmann, J.-S. and Konolige, K. (1999). Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, 1999. CIRA'99. Proceedings. 1999 IEEE International Symposium on*, pages 318–325. IEEE.

Haga, Y., Tanahashi, Y., and Esashi, M. (1998). Small diameter active catheter using shape memory alloy. In *Proceedings MEMS 98. IEEE. Eleventh Annual International Workshop on Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Systems (Cat. No. 98CH36176*, pages 419–424. IEEE.

Hahnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211. IEEE.

Heemels, W. M., van de Wouw, N., Gielen, R., Donkers, M., Hetel, L., Olaru, S., Lazar, M., Daafouz, J., and Niculescu, S.-I. (2010). Comparison of overapproximation methods for stability analysis of networked control systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 181–190.

Hegedüs, F., Bécsi, T., Aradi, S., and Gápár, P. (2017). Model based trajectory planning for highly automated road vehicles. *IFAC-PapersOnLine*, 50(1):6958 – 6964. 20th IFAC World Congress.

Helmert, M., Röger, G., et al. (2008). How good is almost perfect?. In *AAAI*, volume 8, pages 944–949.

Hespanha, J. P., Naghshtabrizi, P., and Xu, Y. (2007). A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1):138–162.

Horn, S. and Janschek, K. (2010). A set-based global dynamic window algorithm for robust and safe mobile robot path planning. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–7.

Howard, T. M., Green, C. J., and Kelly, A. (2010). Receding horizon model- predictive control for mobile robot navigation of intricate paths. *Field and Service Robotics*, pages 69–78.

Howard, T. M., Green, C. J., Kelly, A., and Ferguson, D. (2008). State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *Journal of Field Robotics*, 25(6-7):325–345.

Hsu, D., Kindel, R., Latombe, J.-C., and Rock, S. (2002). Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 21(3):233–255.

Huang, H.-M. (2007). Autonomy levels for unmanned systems (alfus) framework: safety and application issues. In *Proceedings of the 2007 Workshop on Performance Metrics for Intelligent Systems*, pages 48–53.

Huang, Z., Liu, W., Wang, X., Song, X., Xu, X., Chen, X., Ma, L., and Tang, L. (2016). Design of an Intelligent Trimaran USV for Maritime Rescue. volume All Days of *International Ocean and Polar Engineering Conference*. ISOPE-I-16-136.

Hughes, W. (2015). Musings on the history of the rov industry. *Marine Technology Society Journal*, 49(6):79–84.

J. Wünnenberg and P. M. Frank (1987). Sensor fault detection via robust observers. In S. Tzafestas and M. Singh and G. Schmidt, editor, *System Fault Diagnostics, Reliability & Related Knowledge-based Approaches – Volume 1*, pages 147–160. D. Reidel Publishing Company, Dordrecht, Holland.

Jager, E. W., Inganäs, O., and Lundström, I. (2000a). Microrobots for micrometer-size objects in aqueous media: potential tools for single-cell manipulation. *Science*, 288(5475):2335–2338.

Jager, E. W., Smela, E., and Inganäs, O. (2000b). Microfabricating conjugated polymer actuators. *Science*, 290(5496):1540–1545.

Janabi-Sharifi, F. and Vinke, D. (1993). Integration of the artificial potential field approach with simulated annealing for robot path planning. In *Proceedings of 8th IEEE International Symposium on Intelligent Control*, pages 536–541. IEEE.

Jensfelt, P., Kragic, D., Folkesson, J., and Björkman, M. (2006). A framework for vision based bearing only 3d slam. In *IEEE International Conference on Robotics and Automation (ICRA) Location: Orlando, FL Date: MAY 15-19, 2006*, pages 1944–1950. IEEE.

Kadirkamanathan, V., Li, P., Jaward, M. H., and Fabri, S. G. (2000). A sequential monte carlo filtering approach to fault detection and isolation in nonlinear systems. In *Proceedings of the 39th IEEE Con-*

ference on Decision and Control (Cat. No.00CH37187), volume 5, pages 4341–4346 vol.5.

Kadirkamanathan, V., Li, P., Jaward, M. H., and Fabri, S. G. (2002). Particle filtering-based fault detection in non-linear stochastic systems. *International Journal of Systems Science*, 33(4):259–265.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45.

Kamyar, R. and Taheri, E. (2014). Aircraft optimal terrain/threat-based trajectory planning and control. *Journal of Guidance, Control, and Dynamics*, 37(2):466–483.

Kant, K. and Zucker, S. W. (1986). Toward efficient trajectory planning: The path-velocity decomposition. *The International Journal of Robotics Research*, 5(3):72–89.

Kavraki, L., Svestka, P., Latombe, J.-C., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.

Kayacan, E., Kayacan, E., Chen, I.-M., Ramon, H., and Saeys, W. (2018). *On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles*, pages 49–73. Springer International Publishing.

Kayacan, E., Ramon, H., and Saeys, W. (2016). Robust trajectory tracking error model-based predictive control for unmanned ground vehicles. *IEEE/ASME Transactions on Mechatronics*, 21(2):806–814.

Kelly, A. (2004). Linearized error propagation in odometry. *The International Journal of Robotics Research*, 23(2):179–218.

Khatib, M., Jaouni, H., Chatila, R., and Laumond, J.-P. (1997). Dynamic path modification for car-like nonholonomic mobile robots. In *Proceedings of international conference on robotics and automation*, volume 4, pages 2920–2925. IEEE.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer.

Kim, C.-J., Pisano, A. P., and Muller, R. S. (1992). Silicon-processed overhanging microgripper. *Journal of Microelectromechanical Systems*, 1(1):31–36.

Kim, J.-O. and Khosla, P. K. (1992). Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349.

Kolmanovsky, I. and Gilbert, E. G. (1998). Theory and computation of disturbance invariant sets for discrete time linear systems. *Mathematical Problems in Engineering*, 4(4):317–367.

Kothare, V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. 32.

Krishnamurthy, D. A. (2008). Modeling and simulation of skid steered robot pioneer 3at. Master's thesis, Florida State University, College of Engineering.

Krishnaswamy, V. and Newman, W. S. (1992). Online motion planning using critical point graphs in two-dimensional configuration space. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 2334–2339. IEEE.

Kuffner, J. J. and LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

Kumar, V., Å½efran, M., and Ostrowski, J. P. (1999). *Motion Planning and Control of Robots*, chapter 15, pages 295–315. John Wiley & Sons, Ltd.

Kumar, V., Źefran, M., and Ostrowski, J. P. (2007). *Motion Planning and Control of Robots*, chapter 15, pages 295–315. John Wiley & Sons, Ltd.

Kuwata, Y., Schouwenaars, T., Richards, A., and How, J. (2005). Robust constrained receding horizon control for trajectory planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 6079.

Laschi, C., Mazzolai, B., and Cianchetti, M. (2016). Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science robotics*, 1(1):eaah3690.

Latif, R. and Saddik, A. (2019). Slam algorithms implementation in a uav, based on a heterogeneous system: A survey. In *2019 4th World Conference on Complex Systems (WCCS)*, pages 1–6.

LaValle, S. M. et al. (1998). Rapidly-exploring random trees: A new tool for path planning.

Lenain, R., Thuilot, B., Cariou, C., and Martinet, P. (2006). High accuracy path tracking for vehicles in presence od sliding: Application to farm vehicle automatic guidance for agri- cultural tasks. *Autonomous Robots*, 21(1):79–97.

Lhomme-Desages, D., Grand, C., and Guinot, J. (2007). Trajectory control of a four-wheel skid-steering vehicle over soft terrain using physical interaction model. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Liu, C., Li, C.-Y., Wang, Y., and Tomizuka, M. (2017). Convex feasible set algorithm for constrained trajectory smoothing. In *Proceedings of the American Control Conference*.

Liu, Y. and Thrun, S. (2003). Results for outdoor-slam using sparse extended information filters. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 1227–1233. IEEE.

Lombardi, W., Olaru, S., and Niculescu, S. I. (2009). Invariant sets for a class of linear systems with variable time-delay. In *2009 European Control Conference (ECC)*, pages 4757–4762.

Lumelsky, V. and Stepanov, A. (1986). Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE transactions on Automatic control*, 31(11):1058–1063.

Madhevan, B., Sakkaravarthi, R., Singh, G. M., Diya, R., and Jha, D. K. (2017). Modelling, simulation and mechatronics design of a wireless automatic fire fighting surveillance robot. *Defence Science Journal*, 67(5):572–580.

Majumdar, A. and Tedrake, R. (2013). Robust online motion planning with regions of finite time invariance. In Frazzoli, E., Lozano-Perez, T., Roy, N., and Rus, D., editors, *Algorithmic Foundations of Robotics X*, pages 543–558, Berlin, Heidelberg. Springer Berlin Heidelberg.

Mandow, A., Martinez, J. L., Morales, J., Blanco, J. L., Garcia-Cerezo, A., and Gonzalez, J. (2007). Experimental kinematics for wheeled skid-steer mobile robots. In *Proceedings of the International Conference on Intelligent Robots and Systems*.

Manley, J. E. (2008). Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, pages 1–4. Ieee.

Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., and García-Cerezo, A. (2005). Approximating kinematics for tracked mobile robots. *The International Journal of Robotics Research*, 24(10):867–878.

Masehian, E., Amin-Naseri, M., and Khadem, S. E. (2003). Online motion planning using incremental construction of medial axis. In

2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422), volume 3, pages 2928–2933. IEEE.

Masehian, E. and Amin-Naseri, M. R. (2008). Sensor-based robot motion planning - a tabu search approach. *IEEE Robotics Automation Magazine*, 15(2):48–57.

Mattei, M. and Blasi, L. (2010). Smooth flight trajectory planning in the presence of no-fly zones and obstacles. *Journal of guidance, control, and dynamics*, 33(2):454–462.

Mattei, M., Ollio, L., and Scordamaglia, V. (2016). A set based approach for sfdi on small commercial aircraft. In *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pages 678–683. IEEE.

Mattei, M., Paviglianiti, G., and Scordamaglia, V. (2004). Nonlinear h¥ identity observers for sensor fdi: an application example. In *ICM'04 Proceedings of the IEEE International Conference on Mechatronics.*

Mattei, M., Paviglianiti, G., and Scordamaglia, V. (2005). Nonlinear observers with H_∞ performance for sensor fault detection and isolation: a linear matrix inequality design procedure. *Control Engineering Practice*, 13(10):1271–1281.

Mattei, M. and Scordamaglia, V. (2010a). Coordinated control of a team of helicopters. In *AIAA Infotech@ Aerospace 2010*, page 3473.

Mattei, M. and Scordamaglia, V. (2010b). Path planning for wheeled mobile robots using core paths graphs. *IFAC Proceedings Volumes*, 43(16):360–365.

Mattei, M. and Scordamaglia, V. (2010c). Path planning for wheeled mobile robots using core paths graphs. In *Proceedings of the 7th IFAC Symposium on Intelligent Autonomous Vehicles.*

Michaelson, D. G. and Jiang, J. (2000). Modelling of redundancy in multiple mobile robots. In *American Control Conference, 2000. Proceedings of the 2000*, volume 2, pages 1083–1087. IEEE.

Michalek, M., Dutkiewicz, P., Kielczewski, M., and Pazderski, D. (2009). Trajectory tracking for a mobile robot with skid-slip compensation in the vector-field orientation control system. *Int. J. Appl. Math. Comput. Sci.*, 19(4):547–559.

Michałek, M., Dutkiewicz, P., Kiełczewski, M., and Pazderski, D. (2009). Trajectory tracking for a mobile robot with skid-slip compensation in the vector-field-orientation control system. *International Journal of Applied Mathematics and Computer Science*, 19(4):547–559.

Miki, N. and Shimoyama, I. (1999). Flight performance of micro-wings rotating in an alternating magnetic field. In *Technical Digest. IEEE International MEMS 99 Conference. Twelfth IEEE International Conference on Micro Electro Mechanical Systems (Cat. No. 99CH36291)*, pages 153–158. IEEE.

Moosavian, S. (2008). Experimental slip estimation for exact kinematics modeling and control of a tracked mobile robot. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 95 – 100. IEEE.

Motte, I. and Campion, G. (2000). A slow manifold approach for the control of mobile robots not satisfying the kinematic constraints. *IEEE Transactions on Robotics and Automation*, 16(6):875–880.

Motwani, A. (2012). A survey of uninhabited surface vehicles. *Marine and Industrial Dynamic Analysis, School of Marine Science and Engineering, Plymouth University, Tech. Rep.*

Naeem, W., Xu, T., Sutton, R., and Tiano, A. (2008). The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 222(2):67–79.

Nagatani, K., Tokunaga, N., Okada, Y., and Yoshida, K. (2008). Continuous acquisition of three-dimensional environment information for tracked vehicles on uneven terrain. In *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 25–30.

Nardi, V., Scordamaglia, V., De Capua, C., and Morello, R. (2021). A particle filter approach for fault detection and isolation of rover imu sensors.

Nardi, V. A., Ferraro, A., and Scordamaglia, V. (2018). Feasible trajectory planning algorithm for a skid-steered tracked mobile robot subject to skid and slip phenomena. In *23rd MMAR, Miedzyzdroje*.

Newman, P., Leonard, J., Tardós, J. D., and Neira, J. (2002). Explore and return: Experimental validation of real-time concurrent mapping and localization. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1802–1809. IEEE.

Nieto, J., Guivant, J., Nebot, E., and Thrun, S. (2003). Real time data association for fastslam. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 412–418. IEEE.

Notaro, I., Ariola, M., D'Amato, E., and Mattei, M. (2014). HW VS SW sensor redundancy: Fault detection and isolation observer based approaches for inertial measurement units. In *29th Congress of International Council of Aeronautical Sciences*.

Oftadeh, R., Aref, M. M., Ghabcheloo, R., and Mattila, J. (2013). Mechatronic design of a four wheel steering mobile robot with fault-tolerant odometry feedback. *IFAC Proceedings Volumes*, 46(5):663–669. 6th IFAC Symposium on Mechatronic Systems.

Oh, H. N. H., Tsourdos, A., and Savvaris, A. (2014). Development of collision avoidance algorithms for the c-enduro usv. *IFAC Proceedings Volumes*, 47(3):12174–12181. 19th IFAC World Congress.

Okamura, A. M. (2004). Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot: An International Journal*, 31(6):499–508.

Oommen, B., Iyengar, S., Rao, N., and Kashyap, R. (1987). Robot navigation in unknown terrains using learned visibility graphs. part i: The disjoint convex obstacle case. *IEEE Journal on Robotics and Automation*, 3(6):672–681.

Open Source Robotics Foundation (2013). ROS/TCPROS - ROS wiki. `https://web.archive.org/web/20191228024833/http://wiki.ros.org/ROS/TCPROS`. [Online; accessed 03-July-2020].

Orebäck, A. and Christensen, H. I. (2003). Evaluation of architectures for mobile robotics. *Autonomous robots*, 14(1):33–49.

Oshman, Y. and Carmi, A. (2006). Attitude estimation from vector observations using a genetic-algorithm-embedded quaternion particle filter. *Journal of Guidance Control and Dynamics*, 29:879–891.

Pacheco, L., Luo, N., Cufi, X., and Cobos, J. (2009). Local wmr navigation with monocular data. In *Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on*, pages 584–589. IEEE.

Park, K.-T. and Esashi, M. (1999). A multilink active catheter with polyimide-based integrated cmos interface circuits. *Journal of microelectromechanical systems*, 8(4):349–357.

Pek, C. and Althoff, M. (2018). Efficient computation of invariably safe states for motion planning of self-driving vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3523–3530.

Peng, S. T., Sheu, J., and Chang, C. C. (2004). On one approach to constraining wheel slip for the autonomus control of a 4ws/4wd. In *Proceedings of the International Conference on Control Applications*.

Peng, Y., Han, J., and Huang, Q. (2009). Adaptive ukf based tracking control for unmanned trimaran vehicles. *International Journal of Innovative Computing, Information and Control*, 5:3505–3515.

Pentzer, J., Brennan, S., and Reichard, K. (2014). Model-based prediction of skid-steer robot kinematics using online estimation of track instantaneous centers of rotation. *Journal of Field Robotics*, 31(3):455–476.

Perez, E., Poncela, A., Bandera, A., Urdiales, C., and Sandoval, F. (2000). A curvature-based path tracking method for autonomous mobile robot. In *Proc. of the 8th Int. Symposium on Intelligent Robotics Systems (SIRS 00)*, pages 335–340.

Peter van Blyenburgh - UVS International (2008). Unmanned aircraft systems - the current situation. `https://www.easa.europa.eu/sites/default/files/dfu/ws_prod-g-doc-Events-2008-February-1-Overview-of-the-UAV-Industry-(UVS).pdf`.

Petrov, P., de Lafontaine, J., Bigras, P., and Tetreault, M. (2000). Lateral control of a skid-steering mining vehicle. In *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, volume 3, pages 1804–1809 vol.3.

Pivtoraiko, M., Howard, T. M., Nesnas, I., and Kelly, A. (2008). Field experiments in rover navigation via model-based trajectory generation and nonholonomic motion planning in state lattices. In *Proceedings of the 9th International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, pages 25–29.

Polack, P., Altché, F., d'Andréa-Novel, B., and de La Fortelle, A. (2017). The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 812–818.

Poncharal, P., Wang, Z., Ugarte, D., and De Heer, W. A. (1999). Electrostatic deflections and electromechanical resonances of carbon nanotubes. *science*, 283(5407):1513–1516.

Posthumus - Cloosterman, M. (2008). *Control over communication networks : modeling, analysis, and synthesis*. PhD thesis, Department of Mechanical Engineering.

Qi, J., Peng, Y., Wang, H., and Han, J. (2007). Design and implement of a trimaran unmanned surface vehicle system. In *2007 International Conference on Information Acquisition*, pages 361–365.

Quinlan, S. and Khatib, O. (1993). Elastic bands connecting path planning and control. In *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, pages 802–807. IEEE.

Ramirez, G. and Zeghloul, S. (2000). A new local path planner for nonholonomic mobile robot navigation in cluttered environments. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 3, pages 2058–2063. IEEE.

Rao, N. S., Kareti, S., Shi, W., and Iyengar, S. S. (1993). Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. Technical report, Oak Ridge National Lab., TN (United States).

Rao, N. S., Stoltzfus, N., and Iyengar, S. S. (1991). A'retraction'method for learned navigation in unknown terrains for a circular robot. *IEEE Transactions on Robotics and Automation*, 7(5):699–707.

Reinstein, M., Kubelka, V., and Zimmermann, K. (2013). Terrain adaptive odometry for mobile skid-steer robots. In *2013 IEEE International Conference on Robotics and Automation*, pages 4706–4711.

Rekleitis, I., Dudek, G., and Milios, E. (2001). Multi-robot collaboration for robust exploration. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):7–40.

Rodriguez-Losada, D., Matia, F., Jimenez, A., Galan, R., and Lacey, G. (2005). Implementing map based navigation in guido, the robotic smartwalker. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3390–3395. IEEE.

Rosa, P. and Silvestre, C. (2013). Fault detection and isolation of LPV systems using set-valued observers: An application to a fixed-wing aircraft. *Control Engineering Practice*, 21(3):242–252.

Rudin, K., Ducard, G. J., and Siegwart, R. Y. (2014). A sensor fault detection for aircraft using a single Kalman filter and hidden markov models. In *2014 IEEE Conference on Control Applications (CCA)*, pages 991–996. IEEE.

Samad, T., McLaughlin, P., and Lu, J. (2007). System architecture for process automation: Review and trends. *Journal of Process Control*, 17(3):191–201.

Scheding, S., Nebot, E., and Durrant-Whyte, H. (1998). The detection of faults in navigation systems: A frequency domain approach.

In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 3, pages 2217–2222. IEEE.

Schouwenaars, T., How, J., and Feron, E. (2004). Receding horizon path planning with implicit safety guarantees. In *Proceedings of the 2004 American control conference*, volume 6, pages 5576–5581. IEEE.

Schweppe, F. (1973). *Uncertain Dynamic Systems*. Prentice-Hall electrical engineering series. Prentice-Hall.

Scordamaglia, V., Nardi, V. A., and Ferraro, A. (2019a). A feasible trajectory planning algorithm for a network controlled robot subject to skid and slip phenomena. In *2019 24th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE.

Scordamaglia, V., Nardi, V. A., and Ferraro, A. (2019b). A feasible trajectory planning algorithm for a network controlled robot subject to skid and slip phenomena. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 933–940. IEEE.

Sekhavat, S. and Chyba, M. (1999). Nonholonomic deformation of a potential field for motion planning. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 1, pages 817–822. IEEE.

Sevil, H. E., Desai, P., Dogan, A., and Huff, B. (2012). Modeling of an unmanned ground vehicle for autonomous navigation and obstacle avoidance simulations. In *ASME 2012 5th Annual Dynamic Systems and Control Conference joint with the JSME 2012 11th Motion and Vibration Conference*, volume 1, pages 529–534. American Society of Mechanical Engineers Digital Collection.

Sherman, J., Davis, R., Owens, W., and Valdes, J. (2001). The autonomous underwater glider "spray". *IEEE Journal of Oceanic Engineering*, 26(4):437–446.

Siciliano, B., Khatib, O., and Kröger, T. (2008). *Springer handbook of robotics*, volume 200. Springer.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). Modelling, planning and control. *Advanced Textbooks in Control and Signal Processing. Springer,*.

Silva, I. S., Campopiano, F., Lopes, G. S., Uenojo, A. K., Silva, H. T., Pellini, E. L., Alvarez, A. A., and Barros, E. A. (2018). Development of a trimaran asv. *IFAC-PapersOnLine*, 51(29):8–13. 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018.

Simmons, R. (1996). The curvature-velocity method for local obstacle avoidance. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3375–3382. IEEE.

Sisbot, E. A., Marin-Urias, L. F., Broquere, X., Sidobre, D., and Alami, R. (2010). Synthesizing robot motions adapted to human presence. *International Journal of Social Robotics*, 2(3):329–343.

Steer, B. (1989). Trajectory planning for a mobile robot. *The International Journal of Robotics Research*, 8(5):3–14.

Tanner, H. G., Pappas, G. J., and Kumar, V. (2004). Leader-to-formation stability. *IEEE Transactions on robotics and automation*, 20(3):443–455.

Tao, W., Ou, Y., and Feng, H. (2012). Research on dynamics and stability in the stairs-climbing of a tracked mobile robot. *International Journal of Advanced Robotic Systems*, 9:1.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., et al. (1999). Minerva: A second-generation museum tour-guide robot. In *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*, volume 3. IEEE.

Thrun, S., Burgard, W., and Fox, D. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5(3-4):253–271.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.

Thrun, S. et al. (2002). Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1(1-35):1.

Tian, L. and Collins, C. (2004). An effective robot trajectory planning method using a genetic algorithm. *Mechatronics*, 14(5):455–470.

Topalov, A. V., Tsankova, D. D., Petrov, M. G., and Proychev, T. P. (1998). Intelligent sensor-based navigation and control of mobile robot in a partially known environment. *IFAC Proceedings Volumes*, 31(3):267–272.

Tsach, S., Yaniv, A., Avni, H., and Penn, D. (1996). High altitude long endurance (hale) uav for intelligence missions. In *ICAS PROCEEDINGS*, volume 20, pages 368–379.

Uchiyama, A. (1995). Endoradiosonde needs micro machine technology. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, pages 31–37.

Ulrich, I. and Borenstein, J. (2000). Vfhˆ*: Local obstacle avoidance with look-ahead verification. In *ICRA*, pages 2505–2511.

Valero, F., Mata, V., and Besa, A. (2006). Trajectory planning in workspaces with obstacles taking into account the dynamic robot behaviour. *Mechanism and machine theory*, 41(5):525–536.

Varga, A., Ossmann, D., and Joos, H. (2014). A fault diagnosis based reconfigurable longitudinal control system for managing loss of air data sensors for a civil aircraft. In *Proc. of 19th IFAC World Congress 2014*, volume 47, pages 3489–3496.

Vicen-Bueno, R., Borrione, I., and Wittwer, S. (2020). Conference proceedings of the sixth workshop on military applications of underwater glider technology (6wmaugt).

Villagra, J., Milanés, V., Pérez, J., and Godoy, J. (2012). Smooth path and speed planning for an automated public transport vehicle. *Robotics and Autonomous Systems*, 60(2):252–265.

Viquez, O. A., Fischell, E. M., Rypkema, N. R., and Schmidt, H. (2016). Design of a general autonomy payload for low-cost auv r d. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pages 151–155.

Wang, D. and Low, C. B. (2011). Modeling and analysis of skidding and slipping in wheeled mobile robots: Control design perspective. *IEEE Transactions on Robotics*, 24(3):676–687.

Wang, E., Jia, C., Tong, G., Qu, P., Lan, X., and Pang, T. (2018). Fault detection and isolation in GPS receiver autonomous integrity monitoring based on chaos particle swarm optimization-particle filter algorithm. *Advances in Space Research*, 61(5):1260 – 1272.

Washington, R. (2000). On-board real-time state and fault identification for rovers. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 1175–1181. IEEE.

Wei, T., Huang, Y., and Chen, C. L. P. (2009). Adaptive sensor fault detection and identification using particle filter algorithms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(2):201–213.

Wolek, A. and Woolsey, C. A. (2017). Model-based path planning. In *Sensing and Control for Autonomous Vehicles*, pages 183–206. Springer.

Wong, J. (2008). *Theory of Ground Vehicles*. Wiley.

Wu, W., Chen, H., and Woo, P.-Y. (2000). Time optimal path planning for a wheeled mobile robot. *Journal of Robotic Systems*, 17(11):585–591.

Wu, Y., Wang, T., Liang, J., Chen, J., Zhao, Q., Yang, X., and Han, C. (2013). Experimental kinematics modeling estimation for wheeled skid-steering mobile robots. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 268–273.

Xue-Yuan Jiang and Guang-Fu Ma (2005). Spacecraft attitude estimation from vector measurements using particle filter. In *2005 International Conference on Machine Learning and Cybernetics*, volume 2, pages 682–687 Vol. 2.

Yang, H., Liu, C., Shi, J., and Zheng, G. (2019). Development and control of four-wheel independent driving and modular steering electric vehicles for improved maneuverability limits. In *SAE Technical Paper*. SAE International.

Yokokohji, Y. (2021). The use of robots to respond to nuclear accidents: Applying the lessons of the past to the fukushima daiichi nuclear power station. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:681–710.

Yokoyama, N. and Suzuki, S. (2005). Modified genetic algorithm for constrained trajectory optimization. *Journal of guidance, control, and dynamics*, 28(1):139–144.

Yu, J. (2012). A particle filter driven dynamic gaussian mixture model approach for complex process monitoring and fault diagnosis. *Journal of Process Control*, 22(4):778 – 788.

Yun, X. and Yamamoto, Y. (1997). Stability analysis of the internal dynamics of a wheeled mobile robot. *Journal of Robotic Systems*, 14(10):697–709.

Zhang, J. and Knoll, A. (1995). An enhanced optimization approach for generating smooth robot trajectories in the presence of obstacles. In *Proc. 1995 European-Chinese Automation Conference*.

Zhang, M. (2019). Finite-time model-free trajectory tracking control for overhead cranes subject to model uncertainties, parameter variations and external disturbances. *Transactions of the Institute of Measurement and Control*, 41(12):3516–3525.

Zhang, W., Branicky, M. S., and Phillips, S. M. (2001). Stability of networked control systems. *IEEE control systems magazine*, 21(1):84–99.

Zhong, Y., Zhang, W., Zhang, Y., Zuo, J., and Zhan, H. (2019). Sensor fault detection and diagnosis for an unmanned quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 96(3-4):555–572.

Zhou, G.-X. (1989). Swallowable or implantable body temperature telemeter-body temperature radio pill. In *Proceedings of the Fifteenth Annual Northeast Bioengineering Conference*, pages 165–166.

Zohar, I., Ailon, A., and Rabinovici, R. (2011). Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application. *Robotics and Autonomous Systems*, 59(6):343–353.

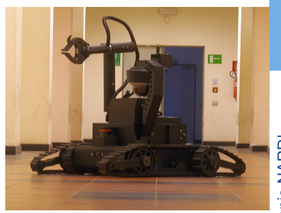
Zong, Z., Zweiri, Y. H., and Seneviratne, L. D. (2006). Non-linear observer for slip estimation of skid-steering vehicles. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

copertina_tesi_3.pdf 1 20/01/2022 17:43:50

Vito Antonio Nardi

recevied his BSc degree in telecommunication engineering in 2016, and his MSc degree (summa cum laude) in 2018 from Mediterranea University of Reggio Calabria. His research interests during the XXXIV cycle of information engineering PhD course are focused upon control systems. Namely, the include the control and motion planning of vehicles subject to uncertainties and disturbance, computer vision, fault detection and isolation of attitude sensors and robust control of heterogeneous vehicles. He won the “best paper presented by a young researcher” award during the 2021 IEEE workshop “metrology for the sea”.

DOTTORATO di RICERCA in INGEGNERIA dell’INFORMAZIONE
SCUOLA di DOTTORATO
Università degli Studi *Mediterranea* di Reggio Calabria



This thesis aims to discuss a set-based approach, and its experimental validation, to the problem of trajectory planning of a skid-steered tracked mobile robot, taking into account actuation constraints, model uncertainty arising from the use of a networked control system, and disturbances caused by skid and slip phenomena.
The problem of trajectory planning consists in providing a vehicle with a path and a timing law suitable for the accomplishment of a certain mission while abiding by a series of constraints. Such problem requires the solution of a wide series of problems of scientific relevance in information engineering and control systems research.
The proposed framework consists of two main elements: a constrained control problem, whose solution is checked in order to guarantee its disturbance invariance and a series of set-based problems, based upon semi-definite programming, aimed at checking whether a manoeuvre can be part of a trajectory, which are defined in terms of series of connected segments, to be crossed at a given velocity.
More in detail, a disturbance-invariant set, associated to a static feedback control law, which embeds the region of tracking error state-space where actuation constraints are satisfied is considered. Candidate trajectory segments are checked for their feasibility before being added to a trajectory. Such check is performed through an ellipsoidal embedding of possible states, reached by the vehicle at the end of a candidate trajectory segment. If the embedding is included in the d-invariant region associated with the controller, the trajectory segment is deemed feasible. Experimental validation is carried out on a Jaguar v4 mobile robotic platform.

ISBN 978-88-99352-44-8

Progetto grafico
Centro Stampa d'Ateneo

COLLANA DELLA SCUOLA DI DOTTORATO DELL'UNIVERSITA' DEGLI STUDI *MEDITERRANEA* DI REGGIO CALABRIA

Vito Antonio NARDI

Feasible trajectory planning problem for mobile robot subject to uncertainties and disturbances: a set based approach

DRINF

DOTTORATO di RICERCA in INGEGNERIA dell’INFORMAZIONE

DOCTORAL SCHOOL OF MEDITERRANEA UNIVERSITY OF REGGIO CALABRIA

Centro Stampa d’Ateneo

DIPARTIMENTO DI INGEGNERIA DELL’INFORMAZIONE, DELLE INFRASTRUTTURE E DELL’ENERGIA SOSTENIBILE

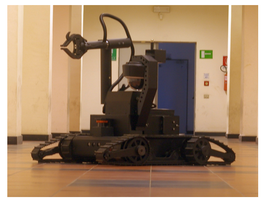
SCIENTIFIC BOARD MEMBERS:

Tommaso ISERNIA (coordinator)
Pier Luigi ANTONUCCI
Giuseppe ARANITI
Francesco BUCCAFURRI
Salvatore COCO
Giuseppe COPPOLA
Lorenzo CROCCO
Claudio DE CAPUA
Francesco DELLA CORTE
Dominique DALLET
Aimè LAY EKUAKILLE
Giuliana FAGGIO
Fabio FILIANOTI
Patrizia FRONTERA
Sofia GIUFFRE'
Giorgio GRADITI
Voicu GROZA
Antonio IERA
Tommaso ISERNIA
Gianluca LAX
Giacomo MESSINA
Antonella MOLINARO
Andrea Francesco MORABITO
Giacomo MORABITO
Rosario MORELLO
Domenico ROSACI
Giuseppe RUGGERI
Domenico URSINO

Vito Antonio NARDI

Feasible trajectory planning problem for mobile robot subject to uncertainties and disturbances: a set based approach

Supervisor: Prof. Tommaso ISERNIA
Coordinator: Prof. Valerio SCORDAMAGLIA
S.S.D. ING-INF/04
XXXIV Ciclo



Collana
Quaderni del Dottorato di Ricerca in
Ingegneria dell’Informazione
Quaderno n°53