MŰEGYETEM 1782

Budapest University of Technology and Economics
Department of Control Engineering and Information Technology
Budapest, Hungary

STRATIFIED MOTION AND MANIPULATION PLANNING ALGORITHMS IN ROBOTICS

RÉTEGEZETT MOZGÁS ÉS MANIPULÁCIÓ-TERVEZÉSI ALGORITMUSOK A ROBOTIKÁBAN

Thesis by

István Harmati

In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Supervisors:

Dr. Béla Lantos
Dept. of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest, Hungary

Dr. Shahram Payandeh
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia, Canada

March, 2003

(Defended January 20, 2004)

Declaration

Undersigned, István Harmati, hereby state that this Ph. D. Thesis is my own work wherein I have used only the sources listed in the Bibliography. All parts taken from other works, either in a word for word citation or rewritten keeping the original contents, have been unambiguously marked by a reference to the source.

Nyilatkozat

Alulírott Harmati István kijelentem, hogy ezt a doktori értekezést magam készítettem és abban csak a megadott forrásokat használtam fel. Minden olyan részt, amelyet szó szerint, vagy azonos tartalomban, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Budapest, 2003. 03. 16.

Harmati István

Az értekezésről készült bírálatok és a jegyzőkönyv a későbbiekben a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki Karának Dékáni Hivatalában elérhetőek.

The reviews of this Ph. D. Thesis and the record of defense will be available later in the Dean Office of the Faculty of Electrical Engineering and Informatics of the Budapest University of Technology and Economics.

Acknowledgements

I wish to thank all the people who positively influenced my work on this thesis, provided me with scientific advice and moral support.

I would like to thank Professor Dr. Béla Lantos, my supervisor at the Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Budapest, Hungary. Without his guidance and support, my work would not have been possible. In addition, it was his classes that introduced me to the wonderful world of control theory.

Also, I would like to express my sincere gratitude to Professor Dr. Shahram Payandeh, my supervisor at the School of Engineering Science, Simon Fraser University, Burnaby, British Columbia, Canada, for his valuable advices and inspiring ideas. I am thankful to him for turning my attention to stratified control and giving me the opportunity to do a part of my research in Beautiful British Columbia, the most fascinating place in the world I have ever seen.

I owe a special thank to Dr. Bálint Kiss, my Mentor, for the opportunity to learn so much from him. I appreciate very much his time and knowledge he shared with me to make my research successful. In particular, our discussions about differential geometry with his thought-provoking arguments remain unforgettable for me.

I would like to thank my collegues in the lab for their encouragements and working conditions provided during my work. Special thanks go to Gábor Vass, Zsolt Kemény and Tamás Falus for their help during my study. The friendship of Huba Lehel Csekő, Lőrinc Márton, Tamás Péni, Gábor Rödönyi, María del Carmen Carolina Amézquita Benítez, María Guadalupe Delgadillo Padierna, Reza Naseras, Kamran Kaveh Mariam, Tissaphern Mirfakhrai gave me also a lot of enthusiasm.

I would like to thank my family. To my parents, for their loving care and unconditional support. Their devotion allowed me to devise an own "motion planning algorithm" toward my desired goals in my life. I am also thankful for my brother, Zoltán, for his kindness and passionate attitude in our talks about academic professions.

Finally, I would like to acknowledge the Hungarian National Research Program (under grants OTKA T 029072 and OTKA T 042634), the Hungarian Research and Development Program for Higher Education (under grant FKFP 0417/1997), the Natural Sciences and Engineering Research Council (NSERC) of Canada (under Research Grant 611205), the project EU GIRT-CT-2002 (CLAWAR - CLimbing and WAlking Robots) of European Union and the Siemens Regional Company (granted Scholarship for PhD students) for their support.

Köszönetnyilvánítás

Szeretnék köszönetet mondani mindazoknak, akik a tézissel kapcsolatos munkámat tudományos tanáccsal és morális támogatással pozitívan befolyasolták.

Szeretnék köszönetet mondani Dr. Lantos Béla professzornak, a Budapesti Műszaki és Gazdaságtudományi Egyetem Irányítástechnika és Informatika Tanszékén dolgozó konzulensemnek. A szakmai vezetése és támogatása nélkül a munkám nem lett volna lehetséges. Ezenkívül az Ő órái voltak azok, amelyek bevezettek az irányításelmélet csodálatos világába.

Szintén szeretném kifejezni őszinte hálámat a kanadai Simon Fraser University, School of Engineering Science tanszékén dolgozó konzulensemnek, Dr. Shahram Payandeh professzornak is az értékes tanácsaiért és inspiráló ötleteiért. Hálás vagyok azért, hogy figyelmemet a rétegezett irányítás felé forditotta és lehetőséget adott arra, hogy kutatásom egy részét a gyönyörű Brit Kolumbiában, a világ általam látott leglenyűgözőbb részén végezzem.

Speciális köszönettel tartazom Dr. Kiss Bálintnak, Mentoromnak, a lehetőségért, hogy oly sokat tanulhattam Tőle. Nagyon sokra értékelem azt az időt és tudást amit megosztott velem annak érdekében, hogy a kutatásom sikeres legyen. A differencál-geometriai diszkusszióinkban megjelenő gondolatébresztő érvelései különösen felejthetetlenek maradnak számomra.

Szeretnék köszönetet mondani a laborbeli kollégáimnak a bátorításokért és a munkám folyamán nyújtott munkakörülményekért. Speciális köszönet illeti Vass Gábort, Kemény Zsoltot, és Falus Tamást a tanulmányaim alatt nyújtott segítségükért. Csekő Lehel Huba, Márton Lőrinc, Péni Tamás, Rödönyi Gábor, María del Carmen Carolina Amézquita Benítez, María Guadalupe Delgadillo Padierna, Reza Naseras, Kamran Kaveh Mariam, Tissaphern Mirfakhrai szintén sok lelkesedést adott számomra.

Szeretnék köszönetet mondani a családomnak. Szüleimnek, szerető gondoskodásukért és feltétlen támogatásukért. Az odaadásuk lehetővé tette számomra, hogy kidolgozzak egy saját "pályatervezési algoritmust" a kívánt céljaim felé az életben. Szintén hálás vagyok testvéremnek, Zoltánnak, kedvességért és azért a szenvedélyes hozzáállásért, amelyet a tudományos pályáról szóló beszélgetéseinkben tanusított.

Végül szeretnék köszönetet mondani az Országos Tudományos Kutatási Alapprogramok OTKA T 029072 és OTKA T 042634 számain, Felsőoktatási Kutatási és Fejlesztési Programok FKFP 0417/1997 számán, a Natural Sciences and Engineering Research Council (NSERC) of Canada 611205 számán, az Európai Unió EU GIRT-CT-2002 (CLAWAR - CLimbing and WAlking Robots) kutatási programja és a Siemens Nemzeti Vállalat (PhD. hallgatói ösztöndíja) által nyújtott támogatásokért.

Abstract

This Ph. D. Thesis deals with motion and manipulation planning of stratified systems. Stratified systems occur when configuration space can be divided into submanifolds such that different equations of motion are valid on different submanifolds. The change of equations of motion are related to kinematic constraints and originated in the most mechanical systems from establishing and breaking contact points. The most important applications from the point of view of control theory are the legged robots and object manipulations with robotic hand using finger relocations. An earlier method reported basic results for point to point motion planning of such systems.

This dissertation presents algorithms that improve the efficiency of basic method and extends the applicability of the concept.

First, we give a procedure that improves the convergency making possible to achieve desired final state more precisely and, observing the initial phase of divergency, it leads the system back to the near of reference configurations. In many cases, it may be necessary to steer the system not just into desired configuration, but to reach this configuration at desired time. We give a procedure for the required modifications in algorithms to achieve this goal.

Some important parts of manipulation tasks can be studied in stratified frame. The handle of intricate constraints, however, often implies challenges. We propose a technique for parametrization that makes easier handling such issues for stratified manipulation planning methods. We provide a method how to manipulate an object having smooth surfaces which also improves the computational features and increases the freedom in design.

We provide generalized motion planning algorithms for so-called holonomic and nonholonomic stratified systems that are able to steer the system along a specific class of trajectories. In contrast to earlier method, they are able to solve obstacle avoidance problem in the configuration space.

The characteristics of algorithms are demonstrated by simulations focusing especially on two examples. The first example is the motion plannning of six-legged robot while the second task which emerges often is the object manipulation with robotic hand equipped with 4 fingers.

Kivonat

Ez a Ph. D. értekezés rétegezett rendszerek mozgás- és manipulációtervezésével foglalkozik. Rétegezett rendszerek akkor fordulnak elő, amikor a konfigurációs tér részsokaságokra bonthatók oly módon, hogy minden részsokaságon különböző mozgásegyenletek érvényesek. A mozgásegyenletek megváltozása kinematikai kényszerekhez kötődik és a mechanikai rendszerek többségében a kontaktuspontok megszűnése és kialakulása hozza létre. Irányításelméleti szempontból a lépegető robotok és az ujjáthelyezésekkel operáló robotkezes tárgymanipulációk alkotják a legfontosabb alkalmazásokat. Korábbi módszer eredményeket mutatott be az ilyen rendszerek pont-pont mozgástervezésére.

Ez a disszertáció az alapmódszer hatékonyságát növelő algoritmusokat mutat be, amelyek kibővítik a koncepció alkalmazhatóságát is.

Először eljárást adunk az alapmódszer konvergenciájanak javítására, amely lehetővét teszi a kívánt állapot pontosabb elérést és a divergencia kezdeti fázisát megfigyelve visszavezeti a rendszert a referencia konfigurációk közelébe. Sok esetben szükség lehet arra, hogy ne csak a kívánt konfigurációba vezessük a rendszert, hanem, hogy ezt a konfigurációt meghatározott időben érjük el. Eljárást adunk arra, hogy egy ilyen cél megvalósítása milyen módosítást igényel az algoritmusokban.

A manipulációs feladatok egy fontos része rétegezett megközelítésben is tárgyalható. A fellépő bonyolult kényszerek kezelése azonban sok esetben kihívásokat jelentenek. Olyan parameterezési eljárást javasolunk, amely lehetővé teszi ilyen problémák könyebb kezelését a rétegezett manipuláció-tervezések számára. Módszert adunk sima felületekkel rendelkező tárgyak rétegezett manipuláció tervezésére, amelyek a korábbi technikák számítási tulajdonságáit is javítják és egyúttal növelik a tervezés szabadságfokát is.

Egy általánosított rétegezett mozgástervezési algoritmust adunk az ún. holonóm és anholonóm rétegezett rendszerekre, amelyek lehetővé teszik speciális trajektóriatípusok követését. Ez a tulajdonság, a korábbi módszerrel ellentétben, lehetővé teszi akadályok elkerülését a konfigurációs térben.

A bemutatott algoritmusok tulajdonságait szimulációkkal demonstráljuk, amelyek közül kiemelt szerepet játszik két példa. Az egyik példa a hatlábú robot mozgástervezése, míg a másik, gyakran előforduló feladat a négyujjas robotkézzel megvalósított tárgymanipuláció.

Contents

List of Figures

Chapter 1

Introduction

Mankind is in strong relation with systems from his environment. In order to make our life efficient, it is beneficial to get acquainted with them and use them responsibly in our service. It is especially valid for those systems, machines that are created by our own work. The two goals are achieved by analysis and synthesis.

The analysis involves abstraction, modeling of systems and concentrating only on few key features. The models presented in engineering sciences do not reflect comprehensive descriptions but they are sufficient to make the model relevant from a certain point of view. In general, different modeling levels may exist depending on which one can be handled within the frame of the actual methodology of synthesis. For example, the kinematic and dynamic models are typical representations in mechanics. Another important property of a system that attracts the attention of system scientists nowadays appears when the equations of motion are nonlinear instead of the simpler linear ones.

As an illustration, consider a complex mechanical system such as a six-legged robot in Figure 1.1 which will play important role in further investigations. In our discussion, the legs are divided into two groups (legs 1, 3, 5 and legs 2, 4, 6) and the legs in the same group move in the same manner. This simple modeling allows two degrees of freedom for legs, the first one is responsible for lifting legs up and down (breaking and establishing the contact with the terrain) while the second one moves legs forward and backward. It is easy to see that every foot in contact with the ground realizes a (kinematic) constraint. The feet of the robot are not allowed to penetrate into the ground and all the contacts are frictional. The six-legged robot has a peculiarity that makes interesting its exploration. It is a typical example of those systems whose equations of motion depend on the actual constraints i.e. on the collection of legs contacting the ground. The systems with this property are defined as *stratified systems* (see Goodwine (1998a)). The most common discontinuous systems of this kind are the walking robots and manipulation systems using finger relocations. In spite of smooth systems, the equations of motion of stratified systems may change discontinuously at certain states.

In general, the synthesis of system aims to put subsystems together to achieve desired behaviors and characteristics. Such a composition of subsystems forms a class of control system. Here, one of the subsystems is

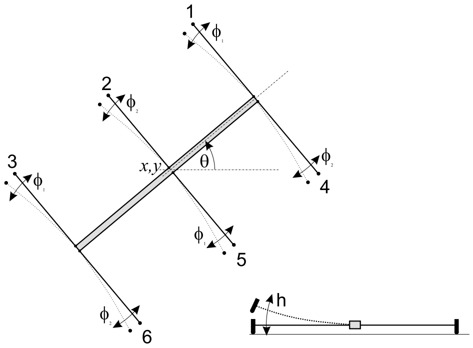

Figure 1.1: Simple model of six-legged robot

the plant. Its regulation and control are the main part of the whole synthesis. Its outputs represent the system variables that should evolve according to given specifications. In addition, controller design is also subjected to restrictions and realization issues. It is the controller that possesses primarily the responsibility to satisfy the requirements. In open-loop architecture, the control algorithm computes the eligible plant input without using output as seen in Figure 1.2. If there is a feedback from the output to the controller then the control system is implemented in closed loop (Figure 1.3). In this case, it is possible to take into consideration the error between output and desired reference signal for improving the system properties.

Consider again the example of six-legged robot (Figure 1.1). Its open-loop control illustrated on Figure 1.4 is to solve a motion planning problem. The outputs of the robot are usually its position and orientation while its inputs are the leg motion. An obvious question arises, how should one drive the legs in order to lead the robot from an initial position and orientation to a desired position and orientation, possibly along a desired path. One can state motion planning problem more generally if in addition to the positions and orientations one also considers the whole state vector including the inner variables of the robot. Such problems concerning stratified systems are addressed in this dissertation.

There exist alternative concepts to control a nonlinear system. Some of them (e.g. methods based on soft computing or linearization) prefer not to be directly involved in the nonlinear equations of the system. They rather try to apply approximations (even at the level of modeling) or trace back the problem to the control of linear system, if it is possible.

The results in this thesis are based on differential geometry which exploits that several achievements of nonlinear control can be associated with expressive geometric interpretation. As an advantage, geometric view is able to suggest new ideas how to control a nonlinear system. Thanks to this property, differential geometry and Lie algebra play increasing role in modern control theory. Their mathematical foundation can be found in some books, for example see Abraham, E., Marsden, and Ratiu (1988), Varadarajan (1994), Serre (1992), Lang (1999), Olver (1993), Lantos (2003). The fundamental issue is to solve vector differential equations by means of suitable operations on vector fields generated by the system. From control aspect, Isidori (1996), Nijmeijer and

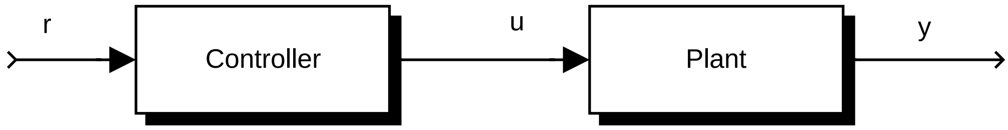

Figure 1.2: Open loop control architecture (y is the output, u is the input of the plant, r is the specifications or reference signal)

van der Schaft (1990) give a comprehensive overview of main results from recent time. The relation between some important nonlinear operations and linear systems is illustrated in Sontag (1998). For the more specific robotic applications Murray, Li, and Sastry (1994) studied a unified approach based on differential geometry.

From the general nonlinear results, many smooth motion planning algorithms have been derived in the past few years. Most of them exploit special properties. For example, there exist efficient methods for differentially flat systems (Fliess, Lévine, Martin, and Rouchon (1995), Kiss (2001)). They establish a special relationship between control input and output function that makes the motion planning and control easy in this case. Another class of systems having special structure are the chained systems. They are controlled by sinusoids inputs (Murray and Sastry (1993), Xu, Ma, and Tso (1999)). It may also be possible to exploit such special features in locomotion. These approaches are based on a special investigation of locomotion from point of view of geometric phases, successfully applied for many systems that have periodic motion in characteristics such as mobile robots, snakelike robots (see Kelly and Murray (1995)). A smooth motion planning algorithm for general (but nilpotent) smooth system was proposed in Lafferriere and Sussmann (1991), Sussmann (1992).

This technique plays a central role in this dissertation. First of all, this is a quite general approach for smooth system which can also be extended to more general class of systems (stratified system) introduced later on. On the other hand, the method has a main drawback since it is not able to steer the system along a desired reference trajectory, it assures only to reach or approximate a final configuration. Approximation occurs if the smooth system is not nilpotent. For this case, nilpotent approximation can be applied as was proposed for example in Vendittelli, Oriolo, and Laumond (1999). Digital version of the algorithm in Lafferriere and Sussmann (1991) was developed in Monaco and Normand-Cyrot (1992). To solve the trajectory tracking problem, Sussmann and Liu (1993), Sussmann and Liu (1991) offer a bit abstract solution using highly oscillatory inputs. This method claims also a significant role in some part of this work.

Motion planning problem for stratified systems are more complicated. Its major reason comes from the fact that the exact form of actual vector fields depends on the actual constraints and it defines different subsystems. For example, consider the motion of hexapod robot while it is swinging one of the leg groups backward on the ground. This results a specific robot motion if the actuated legs are on the ground and another motion if the actuated legs are in the air. If one wants to steer such a system into a final state, the control and motion

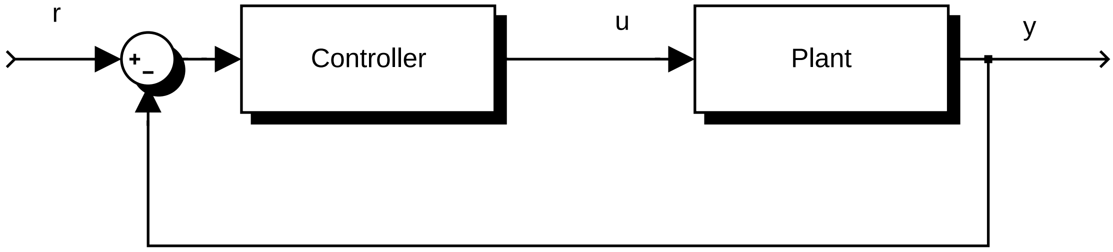

Figure 1.3: Closed loop control architecture (y is the output, u is the input of the plant, r is the specifications or reference signal)

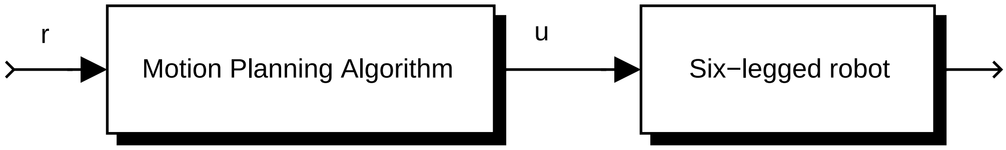

Figure 1.4: Motion planning problem of a (six-legged) robot is solved by an open-loop control.

planning algorithm require more sophisticated approaches than it occurred at smooth systems, because none of the subsystem is controllable in itself, however one may have a chance to steer the robot to desired configuration if connections and transitions are established between them. Some methods such as Ma, Tomiyama, and Wada (2002), Ostrowski and Burdick (1993), Collins and Stewart (1993) develop morphology for the robot gaiting and they intend to provide foot placement concepts.

The most important feature of stratified motion planning is that exploiting the geometry of the system it develops a general approaches without designing contact (foot or finger) placements. In order to handle discontinuous nature, it unifies the subsystems into one abstract system under some carefully chosen conditions. The fundamentals of stratified motion planning concept were established in Goodwine (1998a). The theory is based on the newly defined notion of stratified controllability and provides tests for this property. The controllability test and a basic stratified control were extended to walking robots (Goodwine (1998a), Goodwine and Burdick (2001)) and they were also proposed to manipulation problems with finger relocation (Goodwine (1998b), Goodwine (1999)). Independently from the proper physical system, stratified motion planning exploits the special structure of geometry and offers a general approach that steers the robotic system into a desired configuration.

Some related works for walking robots and manipulation systems

The most common stratified systems are the walking robots and those manipulations which use finger relocations. An important aspect in their control design is that a trade off appears between the simplicity of the controller and robotic system. If the robotic system possesses many degrees of freedom then it is easier to develop a control strategy for it. On contrary, if a robotic system is simple (such as the six-legged model discussed above), the control algorithm needs some sophistication. Stratified motion planning follows the second case. Simple robot model satisfies the conditions the stratified approach assumes, however, the algorithm uses extensive mathematical tool.

Consider first the walking robots. Its study began with Raibert's fundamental work (Raibert (1986)). In spite of stratified control, most of investigations involving legged locomotion concentrate on a particular morphology such as for example hopping robots in Koditschek and Buehler (1991), Ahmadi and Buehler (1997), Zeglin (1999), Hyon and Emura (2002), biped robots in McGeer (1990), Ono, Takahashi, and Shimada (2001), Pratt, Chew, Torres, Dilworth, and Pratt (2001), quadruped robots in Song and Waldron (1989), Ma, Tomiyama, and Wada (2002) or hexapod robots in Ostrowski and Burdick (1993), Collins and Stewart (1993), Schmitt and Holmes (2000a) and Schmitt and Holmes (2000b). Their strategy are mostly based on periodic motion patterns. There exist strategies for example Pfeiffer, Eltze, and Weidemann (1995) and Cruse, Bartling, Cymbalyuk, Dean, and Deifert (1995) that come from biological principles or ideas. Of course, there are concepts which are based on individual mechanical modeling (Berns, Albiez, Kepplin, and Schmalenbach (2001) and Spenneberg and Kirchner (2001)). However, if the robot model is simple (which is the case in this dissertation), the control strategy should exploit special properties such as the particular geometry of the robot.

Some motion planning philosophies relied on special but generally stated properties like stratified control are reported for example in Bullo and Lynch (2001) and Bullo and Zefran (2002a). Most of them are applied to a certain kind of hybrid systems where a switching strategy between subsystems is necessary to be elaborated. For hybrid systems, it is also beneficial to exploit the symmetries of the system (Bullo and Zefran (2002b)).

Another representative class of stratified applications are the manipulation systems. A possible arrangement is seen for example in Figure 1.5. There exist three categories of manipulation task for multi-fingered hand systems: a) the *object manipulation* tries to reach the desired object configuration without considering the contact configuration; b) the *grasp adjustment* attains the desired contact configuration by disregarding the object configuration; and c) *dextrous manipulation* leads the robot hand to its final state taking into account the desired object and contacts configuration, respectively.

All the three strategies mentioned above need a contact model to establish an unambiguous description between the object and the robotic hand. Several contact models are discussed for example in Murray, Li, and Sastry (1994). Most planning algorithms regard *point contact without friction*, the *point contact with friction* or the *soft finger* contact models. The planners also incorporate *fixed*, *sliding* and *rolling* contact points based on the friction model. A good overview about the manipulation philosophies can be found in Bicchi and Kumar (2000).

The fixed contact points offers a simple object manipulation philosophy where the location of the contact points does not change relatively to the object frame (Chevallier and Payandeh (1995)). To accomplish a grasp

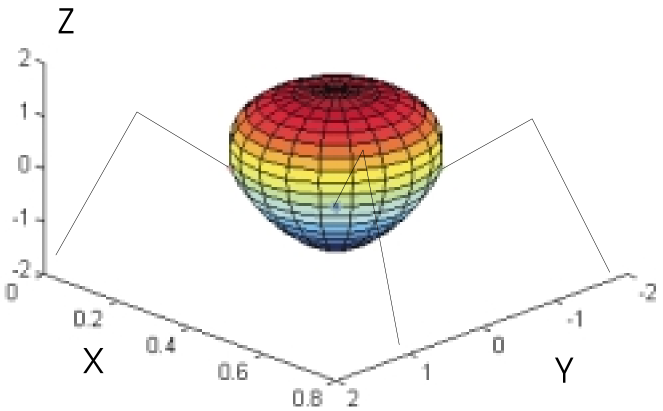

Figure 1.5: Schematic view of robotic manipulation system with four finger (the 4th finger is behind the object)

adjustment, Hong, Lafferriere, Mishra, and Tang (1990) suggested a finger gaiting technique where the contacts are temporary broken during the finger relocations. Restriction arises when the fingers reach the limit of their workspaces. This situation then requires a regrasping algorithm such as Han and Trinkle (1998a).

The rolling contact is able to complete the object manipulation, grasp adjustment and dextrous manipulation task, respectively, see for example Bicchi, Marigo, and Prattichizzo (1998), Han, Guan, Li, Shi, and Trinkle (1997), Kiss, Lévine, and Lantos (1999). Although this approach does not need extra regrasping strategies, it may need relatively complex kinematic description (Montana (1988)). The literature of the rolling contact manipulation is very extensive. Some recently proposed algorithms have been reported in Bicchi, Marigo, and Prattichizzo (1997), Marigo and Bicchi (2000). In general, the manipulation algorithm based on rolling contact expects less finger than the dextrous manipulation combining the fixed point and the finger gaiting. However, if the fingers reach the limit of their workspace, the rolling contact manipulation will also need a finger gaiting technique (Han and Trinkle (1998a), Han and Trinkle (1998b)).

The sliding contact points can be adopted mainly for the grasp adjustment tasks. Planning controlled slippage for improving robotic dexterity has been studied in Cole, Hsu, and Sastry (1992) and Payandeh (1997).

Dextrous manipulation involves many geometric, mechanical and control aspects which can offer different levels of coordination of robotic hand. At the higher planning level, the algorithms reported in Cherif and Gupta (2001), Vass, Payandeh, and Lantos (1999b) attempt to handle globally the manipulation issue decomposing the problem into primitives, high level functions. In this case, one may consider this strategies for convex polyhedra and smooth objects such as Rus (1999), Cherif and Gupta (1997), Cherif and Gupta (2001) where contact points between the object and fingers are always maintained. Stratified manipulation concept (Goodwine (1998b), Goodwine (1999)) and other similar approaches, Omata and Farooqi (1996), Leveroni and Salisbury (1995), Grupen, Huber, Jr., and Souccar (1995), Huber and Grupen (2002) allow repositioning of fingertips. Another trait of stratified manipulation planning is that it may be placed between the higher planning manipulation planning and the lower level manipulation strategies such as Yoshikawa (2000), Zefran, Desai, and Kumar (1996) where more complex dynamic models are regarded.

Outline and Contribution of this Dissertation

The main contribution of this thesis is the extension and modification of general results of basic stratified motion planning that improve the efficiency of algorithms, in particular consideration of legged robotic systems and manipulation systems using finger relocations.

Chapter 2 reviews the earlier smooth and stratified motion planning algorithms used in this thesis. To do so, the most important results, definitions from smooth and stratified control theories are summarized such as smooth and stratified controllability. Also, the chapter aims to transmit the philosophy of stratified approach.

Chapter 3 contains two improvements of basic stratified motion planning algorithm. The first contribution is an algorithm for adaptive insertion of reference points based on on-line experiences while the motion planning algorithm is running. This algorithm improves the convergence properties of basic stratified approach. The second contribution of this chapter is an algorithm which makes the reaching of reference points at desired time points possible. This is an extension of basic stratified motion planning which allowed to prescribe only the location of reference points without regarding time issues.

Chapter 4 contains three new manipulation planning algorithms based on stratified concept. The contributions of this chapter are based on a new description which fits well to stratified concept. The proposed methods reduce the sometimes very complicated symbolic computations to almost pure numerical ones. They provide easily interpretable path. The first method can be applied to dextrous manipulation in a restricted workspace while the second one performs an object manipulation but assuring force closure stability and collision avoidance. As a new extension of them, the third method is able to influence the path according to a desired object motion and finger tip motions above the surface in the configuration space.

Chapter 5 contains a holonomic stratified motion planning along decomposed trajectories. The contribution of this chapter is a motion planning algorithm that is able to steer a holonomic stratified system along special class of reference trajectories. It is an extension in comparison with previous stratified techniques that were not able to influence the shape of trajectory, they could only reach a desired final point. The method can also be used for solving obstacle avoidance problems. The chapter presents the theory of the new concept, the motion planning algorithm based on the theoretical results and it discusses how to plan reference trajectories which are optimal from certain point of view. It is also studied how to approximate complex symbolic computations.

Chapter 6 contains nonholonomic stratified motion planning along decomposed trajectories. The contribution of this chapter is an extension of the results of Chapter 5 to nonholonomic case. It is achieved by introducing new definitions, notions setting up a new foundation for nonholonomic stratified approach.

Finally, Chapter 7 summarizes the results and outlines the possible alternatives for future work.

Fundamental definitions and concepts of differential geometry and Lie algebras extensively used in the thesis are summarized in Appendix.

Chapter 2

Review of Nonlinear Motion Planning Methods

This chapter overviews some results of nonlinear control theory this dissertation is based on. Most of the concepts involve motion planning problem, as a specific area of the more general control theory.

The first section presents the description of smooth nonlinear systems from control aspect that is endowed with mathematical machinery. There are also given the most important definitions related to motion planning problem. Motion planning algorithms have a very extensive literature. Some representative samples were reported in Murray and Sastry (1993), Kelly and Murray (1995), Lafferriere and Sussmann (1991), Xu, Ma, and Tso (1999), Sussmann and Liu (1991), Vendittelli, Oriolo, and Laumond (1999). Here, two smooth motion planning algorithms are summarized, both play important role in the next chapters. The first one solves the problem along sequence of flows (SF). This method is a general motion planning method and also the core of basic stratified control introduced in Section 2.2. The second algorithm outlines a smooth motion planning using highly oscillatory sequence (HOS) of flows. This method is not a part of basic stratified motion planning but plays crucial role in Chapter 6.

The second section sets up the foundation of basic stratified motion planning focusing primarily on the geometry, controllability and realization of stratified systems introduced by Goodwine (1998a). The main goal of this thesis is to extend and improve the efficiency of basic stratified motion planning algorithm in Chapter 3 - Chapter 6.

2.1 Smooth Control Systems

This section reviews the issue of smooth controllability and, using the results, it summarizes two smooth motion planning methods. The reader finds the definition of basic notions related to control theory in Isidori (1996), Nijmeijer and van der Schaft (1990) and Sussmann (1992) while details of the two motion planning algorithms can be found in Lafferriere and Sussmann (1991), Sussmann and Liu (1993) and Sussmann and Liu (1991).

2.1.1 Controllability

A smooth nonlinear control system is generally given, in state space form, by means of set of equations of the following type

$$\begin{aligned} \dot{x} &= f(x) + \sum_{i=1}^{m} g_i(x) u_i \\ y_i &= h_i(x) \quad 1 \leq i \leq p \end{aligned} \tag{2.1}$$

where $x = (x_1, \ldots, x_n)^T$ is the state vector of the system, $f(x)$ and $g_i(x)$ are (practically real analytic) vector fields, u_i are the inputs, y_i are the outputs of the system. In equation (2.1), m is the number of inputs, p is the number of outputs and n denotes the dimension of the state space. In applications, the name configuration space is often used as a synonym of state space. The inputs do not have direct impact on outputs, they influence outputs only via x. For this reason, many discussions similar to the ones in this work omit simply the outputs, they rather express the problem immediately with state vector. The term $f(x)$ is called drift and it requires extra attention at the most time since it is not connected to any input. Drift generally complicates control algorithms and considering this argument, it is neglected in this discussion. In what follows, only nonlinear *kinematic* systems of the form

$$\Sigma \;: \dot{x} = u_1 g_1(x) + \ldots + u_m g_m(x) = G(x)u, \quad \dim(x) = n \tag{2.2}$$

are regarded. Kinematic systems are also called *driftless* or *control linear* systems.

The main issue this dissertation involves is the motion planning problem.

Definition 1. Consider a kinematic system given by (2.2). The *point to point motion planning problem* consists of finding an input trajectory $t \to (u_1(t), \ldots, u_m(t))$ that steers the system from a given initial state x_I to a desired final state x_F. The *motion planning problem along trajectory (or trajectory tracking problem)* consists of finding an input trajectory $t \to (u_1(t), \ldots, u_m(t))$ that steers the system (2.2) along a trajectory $x(t)$ that coincides with a desired reference trajectory $x_d(t)$ i.e. $x(t) = x_d(t)$ for all t.

Motion planning problem is a simplified control problem where one endeavors to obtain suitable inputs for reference trajectory (or final state) in ideal case disregarding disturbing effects. In order to decrease or eliminate errors, one should devise closed loop architecture which is, however, not subject of this work.

One may think that a trajectory tracking problem selecting representative points from reference trajectory turns into a sequence of point to point motion planning problems, if suitable refinement is applied. It means that representative points from reference trajectory are the initial and final states of the point to point motion planning problems. Note, however, that even in this case, the two problems are different. Trajectory tracking solves a continuous problem while the sequence of point to point motion planning problems is a discretized task, maybe with fine resolution, but the trajectory between representative states are still not determined. Obstacle avoidance problem will clarify the difference later.

Motion planning methods rely on some basic notions of control theory which are exposed now. The vector fields $g_1(x), \ldots, g_m(x)$ generate a distribution Δ i.e.

$$\Delta = span\{g_1, \ldots, g_m\}. \tag{2.3}$$

where Δ is typically nonsingular in control applications. It is seen that distribution generalizes vector fields by their bundling, hence, the question arises how to extend some notions derived from vector fields to distributions. It is known (Appendix, definition A-10) that integral curve is defined by integration of single vector field which is actually the solution of a differential equation. The integral curve in parameterized form defines the flow $\Phi(x, \epsilon)$ of a simple system $\dot{x} = g(x)$. The parameter ϵ is replaced by time in system theory and then $\Phi(x, \epsilon)$ defines a trajectory for every initial condition. The main difference in the case of distribution that it contains more than one vector field. The inputs u_i can be regarded as parameters. The key issue now how to obtain the possible states of (2.2) that can be reached from an initial state x_I. Similarly to the integration of vector fields, one should define the integration of (nonsingular) distributions.

Definition 2. A m-dimensional distribution Δ defined on an open set $U \in \mathbb{R}^n$ is said to be *completely integrable* if, for each $x_o \in U$ there exists a neighborhood U_o of x_o, and $n - m$ real-valued smooth functions $h_1, \ldots, h_{n-m}$, all defined on U_o such that

$$span\{dh_1, \ldots, dh_{n-m}\} = \Delta^{\perp}, \qquad dh_i = \frac{\partial h}{\partial x_i}, \quad i = 1, \ldots n - m \tag{2.4}$$

on U_o where dh_i is called the *differential* or the *gradient* of h_i.

It is expected that if the integration of a single vector field (i.e. flow) determines the points where the system $\dot{x} = g(x)$ goes through then the integration of a distribution determines the points where the system (2.2) can move from a fixed x_I. The next theorem asserts when a distribution is completely integrable.

Theorem 1. [Frobenius)]. A nonsingular distribution is completely integrable if and only if it is involutive.

Theorem 1 reveals why the Lie bracket in differential geometry and nonlinear control theory is so important. The system is not restricted to develop only in the directions assigned by control linear combinations

$$v = g_1 u_1 + \ldots + g_m u_m \tag{2.5}$$

of vector fields $g_i(x)$ but regarding flow sequence in (A-34) it may reach points in Lie bracket directions, as well. What is more, if this argument is repeated, the system may develop along recursive Lie bracket directions of Lie brackets. Successive Lie bracket operations define new and new vector fields and they span a Lie algebra generated by the control vector fields. The Lie algebra defined this way is called *control Lie algebra* and denoted by $\mathcal{L}(\bar{g})(x)$ where $\bar{g} = \{g_1, \ldots, g_m\}$. For the brevity, the notation $\mathcal{L}(\bar{g})$ is rather used instead of $\mathcal{L}(\bar{g})(x)$. If the extra vector fields mentioned previously are added to the original distribution, they make the distribution of control vector fields involutive in a finite dimensional state space. As a matter of fact, the extended (involutive) distribution is the involutive closure of Δ. In accordance with the convention in the

Appendix, notation $\bar{g} = \{g_1, \ldots, g_m\}$ is replaced now by the indetermines $\bar{\mathcal{X}}_m = \{\mathcal{X}_1, \ldots \mathcal{X}_m\}$ where one can simply put g_1 into $\mathcal{X}_1$, g_2 into $\mathcal{X}_2$ and so on. The intention of modification is to emphasize that vector fields are considered now as operators. Again, the system may reach a point along any vector field V if $V \in \mathcal{L}(\bar{\mathcal{X}}_m)$, even though the motion proceeds always only along a v defined in (2.5). It often occurs that more than one inputs are activated at a given time (achieving a control linear combination of control vector fields) which means that permitted directions are designated not only by $\mathcal{X}_i \in \mathcal{L}(\bar{\mathcal{X}}_m)$ but $V \in \hat{\mathcal{L}}(\bar{\mathcal{X}}_m)$. Recall that $\hat{\mathcal{L}}(\bar{\mathcal{X}}_m)$ is the Lie algebra of Lie series generated by $\bar{\mathcal{X}}_m$. Define now the exponential of an element $V \in \hat{\mathcal{L}}(\bar{\mathcal{X}}_m)$ analogously to common power series:

$$\hat{G}(\bar{\mathcal{X}}_m) = \{\exp(V) \ : \ V \in \hat{\mathcal{L}}(\bar{\mathcal{X}}_m)\}. \tag{2.6}$$

Example 1. Let $m = 2$, then some elements of $\hat{G}(\mathcal{X}_1, \mathcal{X}_2)$ are for example

$$\begin{aligned} \exp(\mathcal{X}_2) &= 1 + \mathcal{X}_2 + 1/2\mathcal{X}_2^2 + 1/6\mathcal{X}_2^3 + \cdots \qquad (2.7) \\ \exp(\mathcal{X}_1 + [\mathcal{X}_1, \mathcal{X}_2]) &= 1 + \mathcal{X}_1 + [\mathcal{X}_1, \mathcal{X}_2] + 1/2\mathcal{X}_1^2 + 1/2[\mathcal{X}_1, \mathcal{X}_2]^2 + 1/6\mathcal{X}_1^3 + 1/6[\mathcal{X}_1, \mathcal{X}_2]^3 + \cdots. \end{aligned}$$

Recall that a simple vector field generates an integral curve via its flow, which is in fact, a one dimensional Lie group (see the definition of $\Phi(g, \epsilon)$ in Appendix). In other words, vector field g is the infinitesimal generator of flow $\Phi(g, \epsilon)$ where $\Phi(g, \epsilon)$ is actually a Lie group action. Following the same argument, $\hat{G}(\bar{\mathcal{X}}_m)$ is a Lie group with m infinitesimal generators. (See that for any $\exp(V) \in \hat{G}(\bar{\mathcal{X}}_m)$, $\exp(V)\exp(-V) = 1$ and $\hat{G}(\bar{\mathcal{X}}_m)$ is closed under multiplication, hence $\hat{G}(\bar{\mathcal{X}}_m)$ is a group.) It follows that $\hat{G}(\bar{\mathcal{X}}_m)$ is also a manifold M and its dimension equals to the rank of control Lie algebra generated by $\bar{\mathcal{X}}_m$ (or more exactly $\bar{g} = \{g_1, \ldots, g_m\}$). The tangent space $T_x M$ is spanned by the Lie algebra $\mathcal{L}(\bar{g})$, i.e.

$$T_x M = \mathcal{L}(\bar{g})(x).$$

The integral submanifold M obtained this way contains the points that the system may reach under some control. How to find control from a given initial point x_0 to any point in M generated from x_0 is exactly the task of motion planning algorithms.

Some basic notions in control theory are summarized now. Let M be, in general, the manifold that includes the points where (2.2) may evolve. (Note, however, it does not necessarily contain only those points.) Given a system with the state vector x and input vector u. Consider an open set $W \subseteq M$. Let $R^W(x_0, T)$ be the set of states which can be reached up to time T i.e. the set of states x such that there exists a control $u(t)$, $0 \leq t \leq T$ that steers the system from the $x(0) = x_I$ to $x(T) = x_F$ meanwhile $x(t) \in W$ for $0 \leq t \leq T$.

Definition 3. Let

$$R^W(x_0, \leq T) = \bigcup_{0 \leq \tau \leq T} R^W(x_0, \tau) \tag{2.8}$$

be the *set of all states reachable from* x_0 up to time T.

Then local controllability is defined as follows.

Definition 4. A system is *small time locally controllable* (or briefly locally controllable or STLC) from x_0 if $R^W(x_0, \leq T)$ includes a neighborhood of x_0 for all neighborhoods W of x_0 and $T > 0$ small.

A straightforward extension of local controllability is the accessibility property.

Definition 5. The system has the *accessibility property* if Σ is STLC from every $x \in M$.

There exist a stronger but also important controllability property. Let $\mathcal{R}(\Sigma)$ be the set of all pairs (x_I, x_F) such that x_F is reachable from x_I, i.e. there exist a control $t \to (u_1(t), \ldots, u_m(t))$ that steers Σ from x_I to x_F. Then, Σ is *completely controllable* if $\mathcal{R}(\Sigma) = M \times M$. It is well known from Chow's Theorem that for real analytic systems, accessibility is equivalent to *Lie algebra rank condition (LARC)*, in other words with the condition that $\mathcal{L}(\bar{g})(x) = T_x M$. If the system has no drift (which actually holds in this work) then accessibility is equivalent to complete controllability, as well. Otherwise, it is difficult to give an algebraic condition to complete controllability condition. In this work, Σ has real analytic vector fields g_i, $i = 1, \ldots, m$ on $\mathbb{R}^n$ which means that complete controllability needs to suffice condition $\mathcal{L}(\bar{g})(x) = \mathbb{R}^n$. The condition is satisfied if and only if the dimension of involutive closure of Δ equals to the dimension of state vector x i.e. $\dim(\bar{\Delta}) = n$.

Beyond controllability, involutive distribution has a contribution whether a mechanical system is holonomic or nonholonomic (Marigo and Bicchi (2000)). To see this, assume that a vector x represent the configuration space of a mechanical system with n dimensional manifold $M' = \mathbb{R}^n$. Assume, the velocities $\dot{x} \in T_x M'$ are subjected to h locally independent constraints depicted in the form

$$A(x)\dot{x} = 0 \tag{2.9}$$

where A is a codistribution consisting of h covectors, $h < n$ and each covector is defined by n dimensional row vector of real valued analytic functions. So, the matrix A has h rows and n columns.

Definition 6. Constraints are said to be *holonomic* if their differential form (2.9) is integrable.

Holonomic constraints define $M \in M'$ invariant integral submanifolds of dimension $n - h$. (A submanifold S is invariant if the integral curve $x(t)$ of the system remain in S for any t and $x(0) \in S$).

Definition 7. If the constraints imply an integral submanifold of dimension $n - h + k$, $0 < k \leq h$ through x, then the constraints (and the corresponding system) are said to be *nonholonomic.*

The k denotes the degree of nonholonomy. The system (due to the constraints) is maximally nonholonomic if $k = h$. In the point of view of the control, one can put the constrained system description from (2.9) to (2.2). Namely, if $\Delta(x) = span(\bar{g}(x))$ is an $n \times (n-h)$ annihilator of $A(x)$ and $g_i(x)$, $i = 1, \ldots, n-h$ denote the columns of $G(x)$ then all the admissible velocities $\dot{x} \in A(x)^{\perp} \subset T_x M'$ can be obtained as

$$\dot{x} = G(x)u = \sum_{i=1}^{n-h} g_i(x)u_i \tag{2.10}$$

where u is the vector of quasivelocities from $\mathbb{R}^{n-h}$. The u represents a function in time, i.e. $u : \mathbb{R}^{+} \to \mathbb{R}^{n-h}$, $t \to u(t)$ which is the control or input of the system. The nonholonomy of the system, i.e. the nonintegrability of (2.9) is in accordance with the property that the distribution $\Delta = span(g_1(x), \ldots, g_{n-h}(x))$ in (2.10) is not involutive. For this reason, the terms holonomic system and involutive $\bar{g}(x) = \{g_1(x), \ldots, g_{n-h}(x)\}$ will be sometimes used as synonyms. Let $\bar{\Delta}$ be the involutive closure of Δ and let its dimension be d. In this case, if $d > n - h$ then the system is nonholonomic, if $d = n$, the system is maximally nonholonimic. In practice, nonholonomy generates an engineer goal how one can steer a controllable n-dimensional system with less than n actuator.

2.1.2 Smooth Motion Planning along Sequence of Flows

The following reviews the technique in Lafferriere and Sussmann (1991) discussed also in Sussmann (1992) and Murray and Sastry (1993). It generates a trajectory in the form of flow sequences for smooth, kinematic system (2.2) establishing a foundation for stratified motion planning in Section 2.2. According to (2.2), let the system be given by

$$\Sigma \ : \dot{x} = u_1 g_1(x) + \ldots + u_m g_m(x) \qquad \dim(x) = n$$

assuming that

1. $\bar{g} = \{g_1, \ldots, g_m\}$ are real analytic vector fields,
2. Σ is completely controllable which means specifically that it satisfies the LARC (see Subsection 2.1.1).

If control vector fields span the whole configuration space then it is easy to solve the motion planning problem since vector fields support the moving in any direction in the configuration space. This implies the need of n independent control vector fields driven by the inputs.

However, in general, the system has less input controls than n (and at the same time the system may still be controllable). The key point is that extra possible moving directions are acquired if system develops not only along control vector fields but along their Lie brackets. A base of Lie algebra generated by control vector fields manages to find a systematic system of appropriate Lie brackets. One possible choice, which is applied by the method to be introduced in this subsection, is the set of Philip Hall basis. Extending the system with these new directions and associating fictitious inputs to them, one can span the whole configuration space and solve easily the problem for this extended system. The distinct feature of the method in Lafferriere and Sussmann (1991) is that it provides a trajectory between x_I and x_F on the extended system as a sequence of flows. Flow sequence produced by activating only one input in a time interval assigns also a sequence of vector fields selected from extended system. This specification leads to restriction in the possible solutions for the motion planning problem because the task is not so easy comparing with the case where one could move in any direction determined by the combination of n independent vector fields. In spite of the restriction, the procedure possesses a benefit arising from the fact that transformation between the original and extended system becomes easily computable. The only remaining problem is to substitute extra flows and control inputs with original ones. Flow sequence philosophy reduces difficulty coming from substitution because, by definition, the extra directions (defined along Lie brackets) and their inputs can be replaced by suitable flow sequences of original control vector fields if the system is nilpotent (cf. Definition 9).

Since smooth motion planning algorithm involves nilpotent systems, one needs the following definitions.

Definition 8. *Nilpotent Lie algebra* with order k is defined by Lie algebra $\mathcal{L}$ of the vector fields g_i where all the Lie brackets $[g_{j_1}, [g_{j_2}, \dots, [g_{j_k}, g_{j_{k+1}}] \dots]]$, $j_i \in \{1, \dots, m\}$ equal to zero.

Definition 9. The system Σ is said to be *nilpotent* if its control Lie algebra $\mathcal{L}(\bar{g})$ is nilpotent.

The algorithm of the smooth motion planning outlined previously consists of the following steps.

Algorithm 1. [Smooth motion planning Lafferriere and Sussmann (1991)]
Step 1. System extension. Extend the system Σ to

$$\begin{aligned} \Sigma_e : \dot{x} &= v_1 g_1(x) + \ldots + v_m g_m(x) \\ &+ v_{m+1} g_{m+1}(x) + \ldots + v_r g_r(x) \\ &= G_e(x) v \end{aligned} \tag{2.11}$$

where $span\{g_1(x), \dots, g_r(x)\} = \mathbb{R}^n$ and vector fields $g_{m+1}, \dots, g_r$ are defined by the elements of the Philip Hall basis as higher order Lie brackets of the g_i, $i = 1, \dots m$. It is important to realize that the vector fields $\{g_1(x), \dots, g_r(x)\}$ span the whole tangent space of $\mathbb{R}^n$ since $\{g_1(x), \dots, g_r(x)\}$ are a base (namely, the Philip Hall basis) of Lie algebra $\mathcal{L}(\bar{g})(x)$ i.e. $\mathcal{L}(\bar{g})(x) = \mathbb{R}^n$. As a consequence, n independent vector fields of extended system are given, supporting moving in any direction. Moving along a vector field can be realized by activating the corresponding fictitious input v. In general, the first m vector fields in the original and extended system are the same, so their corresponding inputs will coincide, as well.

Step 2. Steering the extended system Σ_e along any arbitrary path. Attain a control v that steers the extended system Σ_e from x_I to x_F. Since vector fields of Σ_e span $\mathbb{R}^n$, it is easy to design the fictitious input v to the desired path. One regards usually the simplest path, the straight-line path between x_I and x_F. Based on this, one can obtain $v = (v_1, \dots, v_n)$ from the equation $\dot{x} = G_e(x)v$.

Step 3. Solving motion planning problem for extended system Σ_e as a sequence of flows. Fictitious control v solved previously the motion planning problem for extended system according to arbitrary desired trajectory between x_I and x_F. In this step, a special solution (from the fictitious control v) is obtained as a sequence of flows along control vector fields of extended system. (Solution of this type is very important for the stratified motion planning.) The step is broken into two parts:

I. Search the solution of the motion planning problem for Σ_e as a flow sequence. In this step, it is useful to introduce a convention in the notations. Let the vector field $g(x)$ be replaced by the notation xg, so the vector field is written on the right. Similarly, $t \to xe^{tg}$ is the integral curve of g passing through x at time $t = 0$, so the machinery introduced implies that a flow $\Phi(x)$ should be replaced by $x\Phi$. In general, if

$F: M \to N$ is a smooth map then $F(x)$ is substituted by xF and the tangent vector v at x mapped by $F_* v$ is denoted by vF. Based on this, in order to obtain the solution for motion planning problem , one has to solve the formal differential equation Sussmann (1992)

$$\begin{aligned} \Sigma_{fe}: \ \dot{S}(t) &= S(t)(v_1(t)g_1 + \ldots + v_m(t)g_m \\ &+ v_{m+1}(t)g_{m+1} + \ldots + v_r(t)g_r) \end{aligned}$$

on the Lie group

$$S = e^{\tilde{h}_1 B_1} \cdots e^{\tilde{h}_{z-1} B_{z-1}} e^{\tilde{h}_z B_z}, \tag{2.12}$$

where B_i are the Philip Hall basis, $\tilde{h}_i$ are the forward Philip Hall coordinates. Such a solution exists, since $S \in \hat{G}(\bar{\mathcal{X}}_m)$ in (2.6) can be uniquely expressed in the form (2.12). It is necessary that $z \geq n$, however the choice $z = n$ is sufficient and it is usually chosen. If $z = n$, S is a Lie group with n parameters $\tilde{h}_1, \ldots, \tilde{h}_n$. In fact, the solution $S(t)$ solves the motion planning problem between x_I and x_F by means of a flow sequence

$$x_F = \Phi(B_z, \tilde{h}_z) \circ \Phi(B_{z-1}, \tilde{h}_{z-1}) \circ \cdots \circ \Phi(B_1, \tilde{h}_1)(x_I) \tag{2.13}$$

however, one can not plan a desired path between the two points. At first, the system moves for a time $|\tilde{h}_1|$ along the vector fields which is symbolized by the Philip Hall base B_1 with $|v_1| = 1$ (the sign of v_1 is determined by the sign of $\tilde{h}_1$ and $v_i = 0$, $i \neq 1$). After this, the system moves along the next vector field represented by B_2 for a time $\tilde{h}_2$ while $|v_2| = 1$ and $v_i = 0$, $i \neq 2$ and so on up to the zth Philip Hall base and coordinate. As a result, one gains a sequence of flows that leads eventually the system from x_I to x_F. Note that the differential equation (2.12) is formal so the solution for the Philip Hall coordinates will be formal, too. To obtain an exact solution where the equation (2.12) has numerical Philip Hall coordinates, the next step should be performed.

II. Compute the Philip Hall coordinates by Chen-Fliess-Sussmann equation.

If the order of $B_i = 1$ for $1 \leq i \leq z$ then the corresponding forward Philip Hall coordinate is given by

$$\tilde{h}_i(t) = \int_0^t v_i(s)ds \tag{2.14}$$

For other case, B_i can be given in the recursive form $B_i = [B_j, B_k]$ determining the corresponding forward Philip Hall coordinate by evaluating the recursive formal integrals

$$\tilde{h}_i(t) = \int_0^t (-v_j(s)\tilde{h}_k(s) + v_i(s))ds \tag{2.15}$$

(For complete derivation, see Sussmann (1992) and Murray, Li, and Sastry (1994)).

Step 4. Obtain the control u from the Philip Hall coordinates. Actually, if the order of a Philip Hall base is more than one then one should substitute the corresponding system motion with extra flow sequences along the original control vector fields. Since the solution of motion planning problem for extended system Σ_e uses the vector fields $g_1, \ldots, g_r$ separately (i.e. there is only one active vector field at a time), one can substitute separately the flows along $g_{m+1}, \ldots, g_r$ with a sequence of flows consisting only of $g_1, \ldots, g_m$. For this substitution, one can employ (A-34) derived from Campbell-Baker-Hausdorff formula to each flow whose vector field is constructed by Lie brackets, i.e.

$$\Phi([g_{i_1}, g_{i_2}], \epsilon^2) \approx \Phi(-g_{i_2}, \epsilon) \circ \Phi(-g_{i_1}, \epsilon) \circ \Phi(g_{i_2}, \epsilon) \circ \Phi(g_{i_1}, \epsilon)(x) \tag{2.16}$$

where in general, $\Phi(g_i, \epsilon)$ denotes the flow along a vector field g_i for a small time ϵ. Here, g_{i_1} and g_{i_2} are represented by (any) two control vector fields of the original system Σ belonging to u_i's. Following this procedure, every flow in the sequence will be defined *along precisely one* of the control vector fields of the original system.

If the system is nilpotent, each exponential of Lie brackets can be developed exactly as a finite combination of the control vector fields. Such an operation can be done by using the Campbell-Baker-Hausdorff formula.

Remark 1. If he system is not nilpotent then motion planning outlined above steers the system only approximately. It comes from two facts:

1. The Philip Hall basis is not finite.
2. The Cambell-Baker-Hausdorff formula does not stop after finitely many terms.

Example 2. (Computing Philip Hall coordinates) Consider a system

$$\dot{x} = g_1(x)u_1 + g_2(x)u_2 \quad dim(x) = n \tag{2.17}$$

and assume that (2.17) is nilpotent of order $k = 4$ and $n = 6$.
The Philip Hall coordinates are to be obtained as follows. Since $n = 6$, it is necessary to find at least six Philip Hall basis. Vector fields $g_1(x)$ and $g_2(x)$ generate a nilpotent Lie algebra of order 4, hence Philip Hall basis to be inserted in flow sequence are written as

$$\begin{aligned} B_1 &= g_1 \\ B_2 &= g_2 \\ B_3 &= [g_1, g_2] \\ B_4 &= [g_1, [g_1, g_2]] \\ B_5 &= [g_2, [g_1, g_2]] \\ B_6 &= [g_1, [g_1, [g_1, g_2]]] \\ B_7 &= [g_2, [g_1, [g_1, g_2]]] \end{aligned} \tag{2.18}$$

By assumption, $n = 6$ and (2.17) is completely controllable, hence one can construct any trajectory between any $x_I \in \mathbb{R}^6$ and $x_F \in \mathbb{R}^6$ using the first six Philip Hall basis $B_1, \ldots, B_6$ and the fictitious input $v \in \mathbb{R}^6$ associated. Let $B = (B_1, \ldots B_6)$ and regard the straight-line trajectory between $x(0) = x_I$ and $x(T) = x_F$ where T may be prescribed. Then, extended system reads

$$\dot{x}(t) = B(x)v(t) \tag{2.19}$$

and exploiting that

$$\dot{x} = \frac{x_F - x_I}{T} = \lambda \tag{2.20}$$

it yields the fictitious input $v = B^{-1}(x)\lambda$. The Philip Hall coordinates are obtained by recursive integrals

$$\begin{aligned}
\tilde{h}_1(t) &= \int_0^t v_1(s)ds \\
\tilde{h}_2(t) &= \int_0^t v_2(s)ds \\
\tilde{h}_3(t) &= \int_0^t (-v_1(s)\tilde{h}_2(s) + v_3(s))ds \\
\tilde{h}_4(t) &= \int_0^t (-v_1(s)\tilde{h}_3(s) + v_4(s))ds \\
\tilde{h}_5(t) &= \int_0^t (-v_2(s)\tilde{h}_3(s) + v_5(s))ds \\
\tilde{h}_6(t) &= \int_0^t (-v_1(s)\tilde{h}_4(s) + v_6(s))ds \\
\tilde{h}_7(t) &= \int_0^t (-v_2(s)\tilde{h}_4(s)ds
\end{aligned} \tag{2.21}$$

Note, that $n = 6$ implies $v_7 \equiv 0$.

Example 3. Let be given a simple smooth nonlinear system

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} = \begin{pmatrix} \cos(x_1) \\ \sin(x_1 + x_3) \\ \sin^2(x_2) \end{pmatrix} u_1 + \begin{pmatrix} 1 \\ \cos(x_3) \\ \cos(x_1) \end{pmatrix} u_2 = g_1(x)u_1 + g_2(x)u_2 \tag{2.22}$$

Consider motion planning problemthat consists of finding input that steers the system (2.22) from $x_I = (0\ 0\ 0)^T$ to $x_F = (0.5\ 1\ 2)^T$ during $t = 2$ sec.

The two vector fields define Philip Hall basis, however, $n = 3$, hence one should find the third Philip Hall basis to attain the extended system. Define $g_3 = [g_1, g_2]$ that completes the extended system

$$\begin{aligned}
\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{pmatrix} &= \begin{pmatrix} \cos(x_1) \\ \sin(x_1 + x_3) \\ \sin^2(x_2) \end{pmatrix} v_1 + \begin{pmatrix} 1 \\ \cos(x_3) \\ \cos(x_1) \end{pmatrix} v_2 \\
&+ \begin{pmatrix} \sin(x_1) \\ -\sin(x_3)\sin^2(x_2) - \cos(x_1 + x_3) - \cos(x_1 + x_3)\cos(x_1) \\ -\sin(x_1)\cos(x_1) - 2\sin(x_2)\cos(x_2)\cos(x_3) \end{pmatrix} v_3 \\
&= g_1(x)v_1 + g_2(x)v_2 + g_3 v_3 = G_e v
\end{aligned} \tag{2.23}$$

Now, one can obtain an input v that steers extended system from x_I to x_F along straight-line trajectory:

$$v = G_e^{-1} \frac{x_F - x_I}{T} \tag{2.24}$$

Using the values of v, the integrals (2.14) and (2.15) can be evaluated resulting the forward Philip Hall coordinates

$$\tilde{h}_1 = -1.5 \qquad \tilde{h}_2 = 2 \qquad \tilde{h}_3 = 2 \tag{2.25}$$

The Philip Hall basis and Philip Hall coordinates define together a flow sequence that solve the motion planning problem

$$x_F \approx \Phi([g_1, g_2], 2) \circ \Phi(g_2, 2) \circ \Phi(-g_1, 1.5)(x_I) \tag{2.26}$$

The flow evolving along $[g_1, g_2]$ can be substituted by a new flow sequence using (2.16). The total flow sequence achieves the final form

$$x_F \approx \Phi(-g_2, \sqrt{2})\Phi(-g_1, \sqrt{2})\Phi(g_2, \sqrt{2})\Phi(g_1, \sqrt{2}) \circ \Phi(g_2, 2) \circ \Phi(-g_1, 1.5)(x_I) \tag{2.27}$$

The motion planning presented is a general approach and brings a foundation for basic stratified motion planning algorithm in Section 2.2. A study in Lafferriere and Sussmann (1991) reveals some drawbacks. First, the smooth system is not nilpotent generally, which concludes that smooth motion planning achieves only an approximation i.e. $x(T) \neq x_F$. There is given analytical formula in Lafferriere and Sussmann (1991) for critical distance. If the distance between x_I and x_F is less than the critical distance, the resulting trajectory converges to x_F in the sense that $x(T)$ is closer to x_F than to x_I. However, there is no theoretical result for critical step length, it should be obtained experimentally in simulation. To see this, let R be a bounded region where motion planning algorithm works and let x_I and x_F be the initial and final states, respectively. Denote the critical distance by D_{cr}. Let D_a be the admissible step length for R and let ϵ be the permissible error. Then the iterated algorithm for achieving a desirable error that is less than ϵ is the following.

Algorithm 2. [Reaching arbitrary small error Lafferriere and Sussmann (1991)]
Step 1. Let $x_0 = x_I$, $i = 0$.
Step 2. Perform Algorithm 1 to go from x_i to y_i where

$$y_i = x_i + \min(1, \frac{D_a}{\|y - x_i\|})(y - x_i). \tag{2.28}$$

Step 3. Let $x_{i+1} = y_i$.
Step 4. If $\|y - x_{i+1}\| \leq \epsilon$ stop, else $i = i + 1$ and go to Step 2.

Theorem 2. (Lafferriere and Sussmann.) If $0 \leq D_{cr}$ and $y - x = \nu D_{cr}$ then Algorithm 2 stops in at most $2\nu + \log_2 \frac{D_a}{\epsilon}$ steps.

A similar algorithm for critical distance, alternative to Lafferriere and Sussmann (1991) is proposed in Chapter 3. Another purpose of that chapter is to deliver an algorithm allowing to achieve final state x_F at a desired time t. Note, that all the Philip Hall coordinates depend on t however their sum do not necessarily equal to t. A simple scaling method proposed in Chapter 3 overcome this inconvenience. To disclose the main difficulty this algorithm displays, one should just realize it fails to follow any kind of trajectory. While basic stratified motion planning inherits this property in Subsection 2.2.3, Chapter 5 and Chapter 6 are involved in developing response to the challenge.

2.1.3 Smooth Motion Planning using HOS of Inputs

This section outlines the idea of smooth motion planning with highly oscillatory sequence (HOS) of inputs. The algorithm was elaborated by Sussmann and Liu (Sussmann and Liu (1991)), based on quite mathematical foundations not allowing the profound discussion here. For details, reader is referred to the original work. The paper Sussmann and Liu (1993) demonstrates the algorithm on an illustrative example and provides a better understanding of the technique. The results yield approximate tracking for controllable (smooth) systems without drift. Obviously, such a method offers benefits in obstacle avoidance problem, since it steers the system along a trajectory.

Highly oscillatory control starts, similarly to flow sequence philosophy in Subsection 2.1.2, from smooth controllable nonholonomic system without drift given in the form

$$\dot{x}(t) = \sum_{k=1}^{m} u_k(t) f_k(x(t)) \qquad \dim(x) = n \tag{2.29}$$

where $f_1, \ldots, f_m$ are smooth vector fields on a smooth manifold M. Assume, that (2.29) is controllable i.e. satisfies the *Lie Algebra Rank Condition (LARC)*. The "Lie bracket extension" of (2.29) is defined by

$$\dot{x}(t) = \sum_{k=1}^{r} v_k(t) f_k(x(t)) \qquad \dim(x) = n \tag{2.30}$$

where the first m vector fields coincide with $f_1, \ldots, f_m$ in (2.29) and the new admissible directions $f_{m+1}, \ldots, f_r$ are Lie brackets of the f_i, $i \in \{1, \ldots, m\}$. Obviously, (2.30) is much "richer" than (2.29) in the sense that (2.30) is allowed to follow any trajectory easily by means of its numerous vector fields while (2.29) meets difficulties because of having less than n vector fields. A reasonable goal is to find a sequence of trajectories of (2.29) generated by u that converge to a prescribed smooth trajectory of (2.30).

The idea is to convert the problem rather in terms of convergence of inputs than that of trajectories. Consider a sequence of inputs $u^j = (u_1^j, \ldots, u_m^j) \in L^1([0,T], \mathbb{R}^m)$ and let $x(0)$ be the initial condition. If the controls u^j converge to an input u^∞ in some sense then it involves that their generated trajectories x^j converge to x^∞ driven by u^∞. Definition of such a convergence needs u^j to remain bounded in L^1 norm as $j \to \infty$. If this does not hold, the convergence of x^j to x^∞ may fail, as well. An interesting and important fact, however, is that x^j may still converge to a trajectory x^∞ of (2.30) which may fail to be a trajectory of (2.29). To demonstrate how this approximation works on a simple smooth system, see the next example.

Example 4. Define the smooth nonlinear system satisfying LARC by

$$\dot{x}_1 = u_1 \quad \dot{x}_2 = u_2 \quad \dot{x}_3 = u_2 x_1 \tag{2.31}$$

with initial condition $x(0) = (0,\ 0,\ 0)^T$. Now, choose $u^j(t) = \sqrt{j}(\cos jt, \sin jt)$ converging to 0 in the sense that their indefinite integrals

$$U^j(t) = \int_0^t u^j(s)ds \tag{2.32}$$

converge to 0 uniformly. It turns out, however, that $x^j(t)$ converge uniformly to $x^\infty(t) = (0,\ 0,\ t/2)^T$ while the amplitudes of u^j converge to infinity. Indeed, if one defines $f_3 = [f_1, f_2]$, then it follows $f_3 = (0,\ 0,\ 1)^T$ in the example. As a consequence, $x^\infty(t)$ represents a trajectory of the system $\dot{x} = 0.5f_3(x)$. Keep in mind that f_3 is independent from f_1 and f_2 even though the trajectory along f_3 was produced by the inputs of f_1 and f_2. Exactly, that will be the main point in the method based on HOS. It allows the convergence to a trajectory which is defined on the integral submanifold of involutive closure of the distribution spanned by the control vector fields of the system.

To state the results of the theory, one should introduce some notations.

Definition 10. The u^j is said to be $ii(r)$ *convergent to* h i.e. it converges to $h = \{h_I\}_{|I|\leq r}$ in the r-th order iterated integral sense if all the iterated integrals

$$U_I^j := \int_0^t \cdots \int_0^{t_{k-1}} u_{i_1}^j(t_1)\, u_{i_2}^j(t_2) \cdots u_{i_k}^j(t_k)\, dt_k \ldots dt_1 \tag{2.33}$$

converge uniformly, as $j \to \infty$ to the indefinite integrals H_I of certain L^1 function h_I, for all multiindices $I = (i_1, \ldots, i_k) \in \{1, \ldots, m\}^k$, $k = 1, 2, \ldots$ such that $\mid I \mid := k \leq r$.

Definition 11. The u^j is said to satisfy the *r-th order boundedness condition or bc(r)* if certain sequences of functions $\tilde{u}_I^j$ related to multiindices I, can be chosen to be bounded in L^1 norm for $\mid I \mid = r$.

Remark 2. Both Definition 10 and Definition 11 are derived from the extension of classical ones (Sussmann and Liu (1991)).

It is useful now to recall and extend some notations from Appendix. Let $\mathcal{X}_I := \mathcal{X}_{i_1} \cdots \mathcal{X}_{i_k}$ for the multiindices I. Then, an extended input is an $\hat{\mathcal{L}}(\bar{\mathcal{X}}_m)$ valued function v on an interval $[0, T]$. Extended input v is integrable in the sense that all the functions $v_I(t)$ are integrable in $v(t) = \sum_I v_I(t)\mathcal{X}_{\bar{I}}$ ($\bar{I} = (i_k, \ldots, i_1)$ if $I = (i_1, \ldots, i_k)$ and the labeling $\bar{I}$ was introduced in the original work only for the sake of convenience in order to fit it to the order of vector fields in the Lie brackets). An ordinary input $u = (u_1, \ldots, u_m)$ can be considered now as an extended input, identified by the function $u(t) = u_1(t)\mathcal{X}_1 + \cdots u_m(t)\mathcal{X}_m$.

Now, the main results of Sussmann and Liu (1991) can be used which asserts when the convergence to a trajectory along an extended input is possible.

Theorem 3. (Sussmann and Liu (1991)). Let $\{u^j\}_{j=1}^\infty$ be a sequence of input functions belonging to $L^1([0, T], \mathbb{R}^m)$. Let r be a positive integer.

Assume that

C1. $\{u^j\}$ $ii(r)$-converges to an extended input v of order $\leq r$.

C2. the sequence $\{u^j\}$ satisfies the boundedness condition $bc(r)$ with respect to v.

Assume that $f_1, f_2, \ldots, f_m$ are vector fields of class C^{r-1} on manifold M of class C^r. Assume also that initial value problem

$$\dot{x}(t) = \sum_{|I|\leq r} \frac{(-1)^{|I|-1}}{|\, I \,|}\, v_I(t)\, [f_I]\,(x), \qquad x(0) = \bar{x} \tag{2.34}$$

has a unique solution $t \to x^\infty(t)$ which is defined on the interval $[0, T]$. (Here, $[f_I] := [\cdots[\,[f_{i_1}, f_{i_2}], f_{i_3}], \cdots, f_{i_k}].$)
Let $\{\bar{x}^j\}$ be a sequence of points of M that converges to $\bar{x}$.
Then

I. for sufficiently large j, there exists a solution $t \to x^j(t)$, defined on the whole interval $[0, T]$, of the initial value problem arising from (2.29) with inputs u^j and initial condition $x(0) = \bar{x}$,

II. the $x^j(t)$ converge to $x^\infty(t)$ as $j \to \infty$, uniformly with respect to t for $t \in [0, T]$.

In fact, the paper Sussmann and Liu (1991) provides theorems and algorithm how to construct an appropriate sequence of inputs which approximates the extended input

$$v(t) = \sum_{n=1}^{r} \sum_{B\in\mathcal{B}_n} v_B(t) B \tag{2.35}$$

where v_B are functions of class C^1 and $\mathcal{B}_n$ are the subset of Philip Hall basis $\mathcal{B}$ having degree n. Observe that every extended input can be described by (2.35). The sequence of ordinary inputs, with respect to $j \to \infty$, that converge to $v(t)$ is obtained in the form

$$u_k^j(t) = \eta_{k,0}(t) + j^{\frac{1}{2}} \sum_{\omega\in\Omega(2,k)} \eta_{\omega,k}(t) e^{ij\omega t} + \sum_{n=3}^{r} j^{\frac{n-1}{n}} \sum_{\omega\in\Omega(n,k)} \eta_\omega(t) e^{ij\omega t} \tag{2.36}$$

where

1. For $(n, k) \in \{2, \ldots, r\} \times \{1, \ldots, m\}$, $\Omega(n, k)$ are pairwise disjoint finite subsets of $\mathbb{R} - \{0\}$,
2. each $\Omega(n, k)$ is symmetric, in other words, $\omega \in \Omega(n, k)$ induces $-\omega \in \Omega(n, k)$,
3. the functions $\eta_{k,0}$, $\eta_{\omega,k}$, η_ω are complex valued with class C^1 on $[0, T]$,
4. The function $\eta_{k,0}$ is real valued,
5. For any $\omega \in \cup_{n=3}^{r} \cup_{k=1}^{m} \Omega(n, k)$, it holds that $\eta_{-\omega} = \bar{\eta}_\omega$ ($\bar{\eta}_\omega$ denotes the complex conjugate of η_ω),
6. For any $\omega \in \Omega(2, k)$, it holds that $\eta_{-\omega} = \bar{\eta}_\omega$.

The procedure how to compute the set $\Omega(n, k)$ and functions $\eta_{k,0}$, $\eta_{\omega,k}$, η_ω is omitted here (see details in Sussmann and Liu (1991)). It needs to study quite intricate mathematical machinery, hence only the basic idea is sketched.

The main challenge consists of ensuring the "high frequency superposition", possible to achieve even for nonlinear systems, hence, elements of $\Omega(n, k)$ requires to be chosen carefully, establishing independency in special sense. For, one needs to divide Lie brackets in (2.35) into independent equivalence classes whose members cannot be separated. Lie brackets in such a class should be managed together but a separated algorithm is needed to find the coefficients of an individual Lie bracket. The set of frequencies belonging to an equivalence class Ω_E should satisfy a special property called symmetrically minimally canceling property. A set F is symmetrically minimally canceling if

1. F is symmetric, i.e. if $\omega \in F$ then $-\omega \in F$, as well.
2. F contains a minimally canceling subset i.e. a subset where only those linear combinations $\sum_{\omega \in F} a_\omega$ equals to zero and have integer coefficients such that $\sum_{\omega \in F} |a_\omega| \leq l(F)$ holds if a_ω are all equal. Notation $l(F)$ represents the number of elements of F.

Finding suitable functions $\eta_{k,0}$, $\eta_{\omega,k}$, η_ω for every $\Omega(n,k)$ involves additional definitions and considerations. The reader finds details in the papers Sussmann and Liu (1991) and Sussmann and Liu (1993).

Remark 3. Smooth motion planning algorithm based on HOS for approximation of extended input consisting of a single Lie bracket deduces the input Example 4 exhibited.

Unfortunately, the method is mathematically complex and the highly oscillatory sequences induce some realization problems in real-time application. However, it accomplishes an approximation of trajectory tracking and in spite of flow sequence philosophy in Subsection 2.1.2 it promises a solution to obstacle avoidance problem. As some of the main contributions of this dissertation, the method in Chapter 6 will aim to fit this technique to basic stratified concept found in Section 2.2.

2.2 Stratified Control Systems

This section deals with stratified control systems. The conception and results involved throughout this section were proposed in Goodwine (1998a), Goodwine and Burdick (2001), Goodwine (1998b), Goodwine (1999). Subsection 2.2.1 draws the basic nature of such systems, mainly from physical point of view. Subsection 2.2.2 intends to enlighten difficulties arising from stratifaction, and comparing it especially with smooth techniques provides theoretical results how to control stratified systems in open loop. The results presented here rely on the extended notion of controllability called stratified controllability. A constructive manner for planning motion if the system is stratified is devised in Subsection 2.2.3.

2.2.1 Geometry of Stratified Systems

Stratified systems show peculiar features among mechanical systems. Their kinematic model cannot be described by a nonlinear equation of type (2.2). To see when this situation occurs, one needs only to realize, the equations of motion of a mechanical system may discontinuously change if circumstances in its environment change, as well. Typical impacts of this kind on the system discontinuity can be caused by constraints. If there exist several kinematic constraints, each of them implying distinct equations of motion, and the system moves from a state to another one while actual set of constraints are changing, the equations of motion change discontinuously, the global smoothness in (2.2) fails. Still, it is possible to describe the system by a set of smooth equations of motions, keeping in mind, however, that always only one of them is active. Since constraints are often originated from contact points and varying contact points (constraints) are related to robotic systems such as walking robots and manipulation systems where finger relocations are in use, these examples represent typical stratified systems. To enlighten the nature of stratified configuration space, consider the simple example of

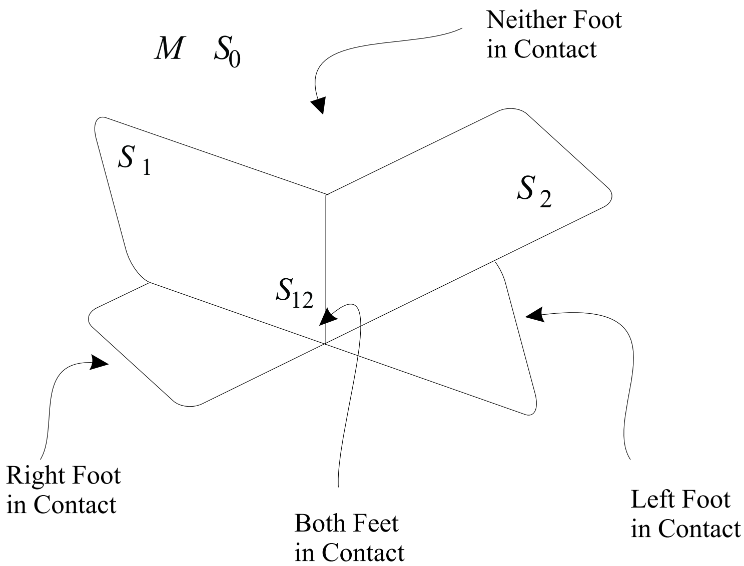

Figure 2.1: Contact structure of biped robot in configuration manifold.

biped robot.

Example 5. (Biped robot) The configuration of biped robot is expressed by state vector composing two groups of state variables. State variables in the first group consisting of the robot's position and orientation determine the robot motion in the space. The remaining variables such as joint angles belong to another group and describe the robot's internal state. The motion of robot is subjected to kinematic constraints due to contact points established by one or two feet. For instance, the robot moves different way if only the left leg is on the ground, and another way if only the second leg is on the ground although robot is driven by the same inputs. A third possible motion is generated if both feet contact the terrain. Based on this, it is possible to separate submanifolds in the configuration space where equations of motion at points belonging to the same submanifolds equal while they may differ from the ones in other regions. The structure of configuration manifolds is schematically illustrated in Figure 2.1. Observe, submanifolds defined this way have different number of dimensions and they usually detach allowed and forbidden regions for the robot. Submanifold defined by more constraints possesses usually less dimensions (more codimensions). Regions are related to distinct inner states as seen in Figure 2.2. It is reasonable to assume that none of the legs penetrates into the ground while one can easily imagine that both legs in the air is an acceptable situation if the robot is running or jumping.

To describe systematically the configuration space such as the one in Example 5, the extended notion of configuration space is required. It proceeds as follows.

Definition 12. A set $\aleph \subset \mathbb{R}^n$ defined by union of disjoint smooth manifolds (i.e. strata) is said to be *regularly stratified set.*

In strict sense, strata should satisfy the so called Whitney condition from Goresky and Macpherson (1980) requiring that the tangent space of two neighboring strata meet "appropriately". From now, it is assumed this

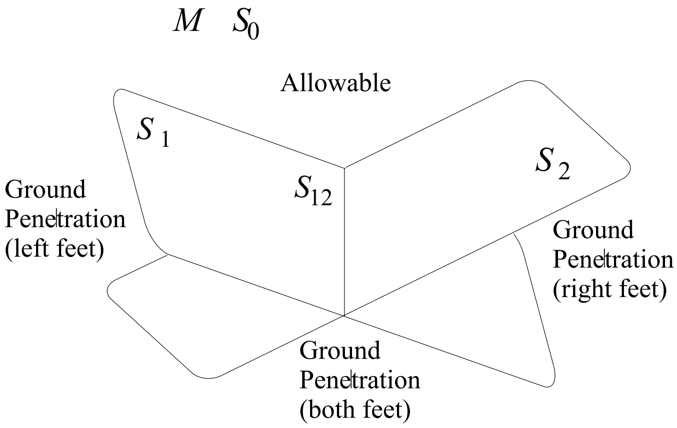

Figure 2.2: Allowed and configuration manifold.

condition holds which is actually true in robotic applications this dissertation concerns. The system is *stratified* if its configuration space is defined by regularly stratified sets as the following.

Definition 13. Let be given a manifold M and n so called level functions $H_i : M \to \mathbb{R}$, $i = 1, \ldots, n$ such that the level sets $S_i = H_i^{-1}(0) \subset M$, $i = 1, \ldots, n$ are (regular) submanifolds of M and any intersection $S_{i_1 i_2 \cdots i_m} = H_{i_1}^{-1}(0) \cap H_{i_2}^{-1}(0) \cap \cdots H_{i_m}^{-1}(0)$, $m \leq n$ is a regular submanifold of M, as well. Then, M and the level functions $H_1, \ldots, H_n$ define a *stratified configuration space*

Definition 13 assumes that equations of motion and vector fields must be smooth everywhere on any stratum excepting the points contained in intersection with other strata. The system is represented by different, smooth nonlinear systems in the strata. The main problem is to develop a motion planning algorithm that combines different systems from different strata into a unified approach. According to Definition 13, let $S_0 \equiv M$ be the whole configuration space where there is no constraint. Let the stratum $S_i \subset S_0$, $i > 0$ be a codimension one submanifold where the system is subjected to a kinematic constraint. In the discussion, this stratum corresponds to dimension $n-1$ manifold in the configuration space. Let $S_{ij} = S_i \cap S_j$ where system is subjected to the two constraints presented on S_i and S_j. In general, a stratum where a few constraints may emerge is denoted by $S_I = S_{i_1 i_2 \ldots i_k} = S_{i_1} \cap S_{i_2} \cap \cdots S_{i_k}$ where $I = i_1 i_2 \ldots i_k$ is a multiindex Isidori (1996).

Definition 14. The stratum with lowest dimension is said to be the *bottom stratum.* A stratum is called *lower stratum* if its dimension is lower than the dimension of the other one. The *higher stratum* is defined vice versa.

Evoking Figure 2.2 of Example 5 again, S_0 is the entire configuration space, while S_1 (S_2) corresponds to stratum where robot moves if the first (second) foot is on the ground. Bottom stratum S_{12} represents a submanifold with minimum dimension among strata, constraints of S_1 and S_2 are valid on this stratum at the same time. (Bottom stratum is a line, the intersection of two strata sketched by plane on Figure 2.2.) Next chapters pursuit two special examples from the class of stratified systems, hence their description assists a better understanding how stratified systems behave.

Example 6. (Six-legged robot) Consider the simple example of six-legged locomotion in Figure 1.1 which is also investigated in next chapters. As it was roughly mentioned in Chapter 1, the legs 1, 3, 5 and legs 2, 4, 6 form two groups, and the legs in a group move in the same manner. Such leg drives always establish a stable position if the leg angles do not leave an admissible range given by (ϕ_{min}, ϕ_{max}). Each leg is controlled by two inputs, one of them lifts the legs up and down (breaking and establishing the contact with the terrain) while the second one moves the leg forward and backward. Since two groups of legs are distinguished, one may think of six legged robot as biped robot with respect to their configuration space. It implies that Figure 2.2 sketches also the stratified configuration of six legged robot, with the only difference that instead of a foot, one should think of a leg group. Each combinations of contacting legs realizes specific (kinematic) constraint, and each of them achieves specific equations of motion. It is assumed that the feet of the robot do not penetrate into the ground and they move forward/backward on the ground with friction. The terrain with friction restricts the legs' motion but it is not useless, these constraints make the robot motion possible.

The variables x, y denote the position, θ denote the orientation of body. The distances of leg groups from the ground are represented by h_1 and h_2. The robot is able to perform different kind of motions. First, if only the set of legs $\{1,3,5\}$ (or $\{2,4,6\}$) touches the ground (active legs) while the other set of legs remains in the air (inactive legs) then the body may move by changing the angles ϕ_1 (resp. ϕ_2). In this case, only the active legs have effect on the planar motion of robot. The successive commutation of active and inactive legs (called gaiting) makes the walking possible.

On the other hand, if all legs touch the ground at the same time then all legs are active. The angles ϕ_1 and ϕ_2 change simultaneously. This motion will be referred to as the paddling motion. The equation of motion reads

$$
\begin{aligned}
\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \\ \dot{h}_2) \end{pmatrix} &= \begin{pmatrix} \alpha(h_1)\cos\theta, \\ \alpha(h_1)\sin\theta \\ l\alpha(h_1) \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} \alpha(h_2)\cos\theta, \\ \alpha(h_2)\sin\theta, \\ -l\alpha(h_2) \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \alpha(h_2) \\ 0 \end{pmatrix} u_3 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \alpha(h_1) \end{pmatrix} u_4 \\
&= g_{1,2}u_1 + g_{2,2}u_2 + g_{1,1}u_3 + g_{2,1}u_4
\end{aligned} \tag{2.37}
$$

where l denotes the length of the legs (for the simplicity, we assume $l = 1$ in the sequel) and

$$
\alpha(h_i) = 1 \quad \text{if} \quad h_i = 0 \text{ and } 0 \quad \text{if} \quad h_i > 0, \quad i = 1, 2. \tag{2.38}
$$

The discontinuity is represented by the function α, the values of h_1 and h_2 define the following strata:

- $S_0 : \mathbb{R}^5 \times [\phi_{min}, \phi_{max}]^2$,
- $S_1 : h_1 = 0$, i.e. legs $\{1,3,5\}$ on the ground,

- $S_2 : h_2 = 0,$ i.e. legs $\{2, 4, 6\}$ on the ground,
- $S_{12} = S_1 \bigcap S_2 : \; h_1 = h_2 = 0,$ i.e. all legs on the ground.

The system has $s = 4$ strata (including S_0) whose equations of motion can be separated as follows.

On the stratum S_{12}, all legs are on the ground, the equation of motion is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ 1 \\ 1 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ -1 \\ 0 \\ 1 \end{pmatrix} u_2 = [g_{12,1} \; g_{12,2}][u_1 \; u_2]^T. \tag{2.39}$$

The control u_3 and u_4 are constrained to be 0, otherwise robot would move to another stratum (cf. moving off vector fields in Definition 17 later).

On the stratum S_1, the leg group 1 is on the ground, the equation of motion is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_4 = [g_{1,1} \; g_{1,2} \; g_{1,3}][u_1 \; u_2 \; u_4]^T. \tag{2.40}$$

The control u_3 is constrained to be 0, i.e. it is not permitted to lift off the first leg group while the robot moves on S_1.

On the stratum S_2, the leg group 2 is on the ground, the equation of motion is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_3 = [g_{2,1} \; g_{2,2} \; g_{2,3}][u_1 \; u_2 \; u_3]^T. \tag{2.41}$$

The control u_4 is constrained to be 0, i.e. it is not permitted to lift off the second leg group while the robot moves on S_2.

On the stratum S_0, there is no leg in contact with the ground, the equation of motion is given by

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \\ \dot{h}_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} u_2 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} u_3 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_4 = [g_{0,1}\ g_{0,2}\ g_{0,3}\ g_{0,4}][u_1\ u_2\ u_3\ u_4]^T. \quad (2.42)$$

Observe that stratum S_{12} has 5 dimension (with 2 constraints) while S_1 and S_2 denote 6 dimensional manifold (with 1 constraint).

Some other chapters in this work study a simple model of four fingered manipulation system using finger relocation.

Example 7. (Four fingered robotic hand) Given a smooth object manipulation system equipped with 4 fingers where each finger has 3 degrees of freedom as sketched in Figure 1.5. Suppose that all the contact points are fixed to the object i.e. there is neither sliding nor rolling. Denote the linear and angular velocity vectors of the object frame (in a fixed inertial frame) by v_o and ω_o, respectively. Furthermore, Let v_{f_i} denote the ith fingertip velocity (in the fixed inertial frame). One can easily see that the tangent space of the whole configuration space S_0 is spanned by the vector $v_o \oplus \omega_o \oplus v_{f_1} \oplus \cdots \oplus v_{f_4}$ having dimension 18. Suppose all the Fingers $1, \ldots 4$ are in contact with the object (i.e. they are glued to the object) so the system is subjected to 4×3 velocity constraints (4 vector constraints). One can easily check that the degrees of freedom of system are provided by the orientation and position of object. It defines a subsystem that moves on the stratum S_{1234} having dimension 6. The stratum S_{1234} is the bottom stratum. The tangent space is spanned by $v_o \oplus \omega_o$ and equation of motion can be described as

$$S_{1234} \quad : \dot{x} = \begin{pmatrix} v_o \\ \omega_o \end{pmatrix} = g_1(x)u_1^{S_{1234}} + \ldots + g_6(x)u_6^{S_{1234}} = G_{1234}(x)u \quad (2.43)$$

Note, that one can choose only 6 independent inputs because the desired object motion demands all the fingertips to move together with the object. These fingertips provide six degrees of freedom in accordance to the tangent vector $v_o \oplus \omega_o$. Now, assume that Finger 1 breaks off the contact with the object and moves freely in the workspace. Meanwhile other three fingers keep the contact with the object. The new situation leads to a new strata S_{234} with equation of motion

$$S_{234} \ : \ \dot{x} = \begin{pmatrix} v_o \\ \omega_o \\ v_{f_1} \end{pmatrix} = g_1(x)u_1^{S_{234}} + \quad \ldots \quad + g_9(x)u_9^{S_{234}} = G_{234}(x)u \quad (2.44)$$

Note that we have 9 inputs because the desired object motion demands fingertip $2, 3$ and 4 to move together with the object. These fingertips provide six degrees of freedom. The Finger 1 can move in free space rendering additional three degrees of freedom. Scenario is similar when other finger brakes off the contact instead of the Finger 1.

Example 7 reveals that illustrating stratified configuration space is not elementary in every applications. Many strata can make the illustration complex.

2.2.2 Stratified Controllability

Smooth motion planning algorithms in Subsection 2.1.2 and Subsection 2.1.3 do not work on stratified systems presented in Subsection 2.2.1, it requires new conception invoking an extended definition of controllability called stratified controllability. This subsection asserts assumptions and sets up new notions Subsection 2.2.3 relies on providing a basic stratified motion planning. The main reason that smooth motion planning algorithms fail to work on stratified system arises from discontinuity. For, recall that equations of motion describing a smooth system

$$\Sigma\ : \dot{x} = u_1 g_1(x) + \ldots + u_m g_m(x) = g(x,u,t) \qquad \dim(x) = n \tag{2.45}$$

has a continuous solution $s(t)$ such that

$$\dot{s}(t) = g(s(t), u(t), t). \tag{2.46}$$

Stratified systems do not have usually continuous solution $s(t)$ (excepting for example smooth systems as special stratified systems), however, they carry fairly nice properties asserted in the following assumptions.

Assumption 1. Considering any stratum, assume that equations of motion imply a unique solution for all points in that stratum which have neighborhoods not containing any points from substrata.

Note that $M = S_0$, the whole configuration space also realizes a stratum. Assumption 1 says that discontinuities occurs only at transition between strata.

Assumption 2. If a flow of system comes to a stratum that define discontinuous motion then system leaves the stratum after a finite time.

Assumption 2 is necessary to avoid chattering problem i.e. the problem when the solution curve penetrates into the stratum, but on the stratum, vector fields direct back the flow into the region the solution arrived from. The aim of stratified motion planning is to exploit the variety of equations of motion originated from different strata, not to embarrass transitions between them, hence the assumptions above express acceptable conditions.

Beside assumptions, motion planning on stratified systems needs to extend the notion of controllability in Definition 4 and related topology from two main reasons.
1. It often occurs that reaching points in an open neighborhood of entire configuration space is not important, what is more it may not be possible. For example, six-legged robot will not slide on its torso while all the legs are in the air. Some legs should be in contact. Similarly, considering object manipulation with robotic hand, some fingers always must be in contact with the object, otherwise it falls down. Next definition gives a relative topology that assists to define the region stratified motion planning regards.

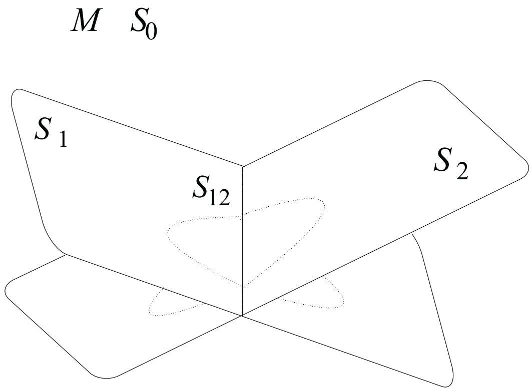

Figure 2.3: Stratified open neighborhood

Definition 15. Let Z be a subset of topological space S endowed by topology O (the collection of open set). Then, the *relative topology* on Z is defined by $O_Z = \{U \cap \mathrm{Z} : U \in O\}$

Considering the allowed and forbidden regions of configuration space of biped (or six-legged) robot in Figure 2.2 discussed in Example 5 (Example 6), a possible stratified open neighborhood defined by relative topology is seen in Figure 2.3. The stratified open set is defined on $S_1 \cup S_2$.

2. It may happen that strata have boundaries if there exist extra constraints subjected to physical boundaries. Such a case is illustrated in Figure 2.4. In this work, strata are regarded without boundaries except when opposite is emphasized. Note, that controllability in bounded regions needs more sophisticated treatment.

The arguments above support to extend the notion of controllability on stratified configuration manifold as follows.

Definition 16. Consider a stratified configuration space defined by a set of strata $\{S_{I_1} \dots, S_{I_m}\}$. The system is *small time locally stratified controllable* from x_0 if $R^W(x_0, \leq T)$ includes a neighborhood of x_0 in $\{S_{I_1} \cup S_{I_2} \cup \dots \cup S_{I_m}\}$ for all neighborhoods W of x_0 in $\{S_{I_1} \cup S_{I_2} \cup \dots \cup S_{I_m}\}$ and $T > 0$ small.

It is easy to see that Definition 16 of stratified controllability returns with smooth controllability due to Definition 4 if $\{S_{I_1} \cup S_{I_2} \cup \dots \cup S_{I_m}\} = M$ i.e. if strata cover the entire configuration manifold.

A main result from Goodwine (1998a) gives a sufficient condition for stratified controllability as stated in the next theorem.

Theorem 4. (Goodwine) Let $T_{x_0}M$ be the tangent space of M at x_0 and let $\bar{\Delta}_{S_j} |_{x_0}$ denote the involutive closures of a distribution spanned by the vector fields of a stratum S_j in x_0. If there exists a nested sequence of strata

$$x_0 \in S_p \subset S_{p-1} \subset \dots \subset S_1 \subset S_0, \tag{2.47}$$

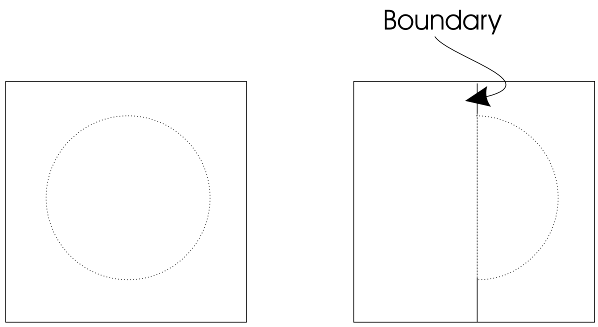

Figure 2.4: Open set without and with boundary

such that the involutive closures of distributions (of strata) fulfill

$$\sum_{j=0}^{p} \bar{\Delta}_{S_j} \mid_{x_0} = T_{x_0} M \tag{2.48}$$

then the system is locally stratified controllable from x_0.

Theorem 4 implies an important consequence. If Theorem 4 is applied repeatedly using several sequences of strata defined by (2.47), each of them starting from the same bottom stratum S_p, however arriving at distinct top strata, then controllability test is validated on the union of top strata. For example, let x_0 be a point in S_{12} in Example 5. If controllability test using Theorem 4 holds for the sequences $x_0 \in S_{12} \subset S_1$, and $x_0 \in S_{12} \subset S_2$, respectively, then system is (locally) stratified controllable on the union of S_1 and S_2 from x_0. This manner is suitable to determine stratified controllability on such open stratified sets as shown in Figure 2.3, often useful in robotic applications, while it does not need conceptionally extra mathematical machinery.

Remark 4. Theorem 4 is derived from the distribution approach of Chow's Theorem. Other stratified controllability tests, equivalent to distribution approach in contribution, were reported in Goodwine (1998a). They are based on exterior differentials and principal fiber bundles, however, not discussed here since the methods in next chapters keep track on distribution approach. Reader finds details in referred works.

2.2.3 Basic Stratified Motion Planning

This subsection intends to give a constructive way how to devise a basic stratified motion planning algorithm for stratified controllable system satisfying the assumptions and condition of Theorem 4 in Subsection 2.2.2. The method presented here extends the results of Subsection 2.1.2 from smooth system to stratified one. Major challenge in stratified motion planning is to integrate subsystems from different strata into uniform treatment. This description requires a common space which usually will be the bottom stratum. Note, however, vector fields of bottom stratum in themselves do not suffice to solve motion planning problem. In order to catch the idea behind the technique, recall Example 5 involving biped robot.

Example 8. (Biped robot, revisited) Let the initial and final state $x_0, x_F \in S_{12}$, respectively, in other words, both legs are on the ground at the beginning and at the end, though they represent distinct positions, orientations and leg angles for the robot. Stratified motion planning problem consists of finding appropriate drives for legs that steers robot from x_I to x_F. Vector field $g_{1,1}$ moves the system from S_{12} onto S_1 and it dues to the phenomena when biped robot lifts off Leg 2 from the ground. Similarly, vector field $g_{2,1}$ moves the system from S_{12} onto S_2 rising Leg 1 from the ground. Again, $g_{1,1}$ and $g_{2,1}$ switch between bottom stratum S_{12} and a higher stratum. Let $g_{1,2}$ be a vector field that move the robot entirely on S_1, while let $g_{2,2}$ be a vector field that move the robot entirely on S_2. Then the flow sequence

$$x_F \quad = \quad \underbrace{\Phi(-g_{2,1}, t_6)}_{S_{12} \leftarrow S_2} \circ \underbrace{\Phi(g_{2,2}, t_5)}_{\text{on } S_2} \circ \underbrace{\Phi(g_{2,1}, t_4)}_{S_2 \leftarrow S_{12}} \circ \underbrace{\Phi(-g_{1,1}, t_3)}_{S_{12} \leftarrow S_1} \circ \underbrace{\Phi(g_{1,2}, t_2)}_{\text{on } S_1} \circ \underbrace{\Phi(g_{1,1}, t_1)}_{S_1 \leftarrow S_{12}}(x_I) \tag{2.49}$$

can solve the motion planning problem. Note that (2.49) accomplishes a gaiting sequence, the robot rises one foot from the ground, moves it in the air parallel to the terrain and lifts it on back to the ground. In the next phase, the procedure is repeated with the other leg. Observe now, if

$$[g_{1,1}, g_{1,2}] = 0 \quad \text{and} \quad [g_{2,1}, g_{2,2}] = 0 \tag{2.50}$$

then Campbell-Baker-Hausdorff formula (A-33) ensures that flows corresponding to vector fields in a Lie bracket commute making rearrange in flow sequence possible. It means that $g_{1,1}$ and $g_{1,2}$ can be interchanged and so can $g_{2,1}$ and $g_{2,2}$. It modifies the flow sequence (2.49) to

$$x_F = \Phi(g_{2,2}, t_5) \circ \Phi(-g_{2,1}, t_6) \circ \Phi(g_{2,1}, t_4) \circ \Phi(g_{1,2}, t_2) \circ \Phi(-g_{1,1}, t_3) \circ \Phi(g_{1,1}, t_1)(x_I) \tag{2.51}$$

Assuming that $t_1 = t_3$ and $t_4 = t_6$, the sequence is simplified to

$$x_F = \underbrace{\Phi^{t_5}_{g_{2,2}} \circ \Phi^{t_2}_{g_{1,2}}}_{\text{on } S_{12}}(x_I), \tag{2.52}$$

where $g_{1,2}$ is defined on S_1 and $g_{2,2}$ is defined on S_2. However, one should evaluate them on the bottom stratum and then, the flow sequence (2.52) will lead to the same result as (2.49). In fact, if the $g_{1,2}$ and $g_{2,2}$ are in the tangent space of S_{12} then the flow sequence will always remain in the bottom stratum satisfying tangency requirement. If tangency requirement is not satisfied for a vector field, say $g_{1,2}$, in (2.52), one can help on it using vector field $g_{1,1}$ the vector field $g_{1,2}$ commutes with. For, let H_B be the level function of bottom stratum. By assumption $g_{1,2}$ is not tangent to bottom stratum, hence $\langle dH_B, g_{1,2}\rangle = h_1(x) \neq 0$. By definition, $g_{1,1}$ leaves the bottom stratum, hence $\langle dH_B, g_{1,1}\rangle = h_2(x) \neq 0$. Now, replacing $g_{1,2}$ by

$$\tilde{g}_{1,2} = g_{1,2} - \frac{h_1(x)}{h_2(x)} g_{1,1}, \tag{2.53}$$

the tangency requirement holds. This procedure can always be carried out without loss of generality when tangency in (2.52) does not hold, therefore, one can always assume that vector fields in (2.52) are tangent to bottom stratum.

Example 8 motivates to divide vector fields into two groups. Vector fields in one group switch between strata while vector fields belonging to the other group never leave their strata.

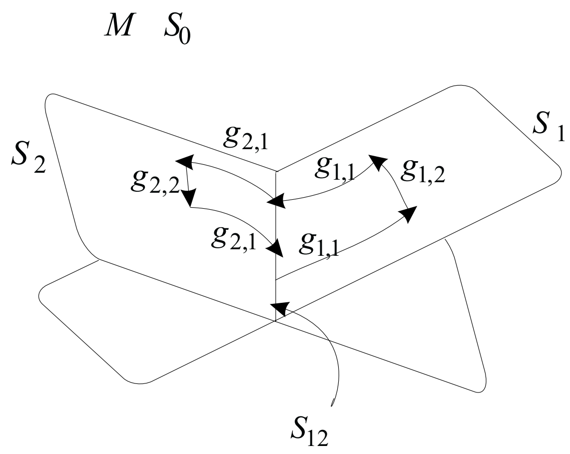

Figure 2.5: Gaiting by sequence of flows

Definition 17. Let be given a stratified configuration space M and let S_p be a stratum of M. Let $x_0 \in S_p$. A vector field $g(x)$ is called a *moving off vector field* if there exists a t, such that $t > 0$ and $x_F \equiv \Phi(g,t)(x_0) \notin S_p$.

Definition 18. Let be given a stratified configuration space M and let S_p be a stratum of M. Let $x_0 \in S_p$. A vector field $g(x)$ is said to be a *moving on vector field* if for every $t > 0$, it holds that $x_F \equiv \Phi(g,t)(x_0) \in S_p$.

Observe that a moving on vector field always keeps the system in the actual stratum while moving off vector field may leave it. It is not fortunate to require for a moving on vector field $g(x)$ only that $g(x_0) \in T_{x_0}S_p$ instead of $x_F \equiv \Phi(g,t)(x_0) \in S_p$ since system may reach a substrata along $g(x_0)$ where $g(x_0) \in T_{x_0}S_p$. Regarding Example 8, it is easy to see that $g_{1,1}$ and $g_{2,1}$ are moving off vector fields while $g_{1,2}$ and $g_{2,2}$ are moving on vector fields. Switchings between strata by means of moving off vector fields are illustrated in Figure 2.5 which draws a gaiting period of biped robot. Condition (2.50), necessary for switching between strata, can be generalized as follows.

Proposition 1. (Goodwine) If moving on vector fields commute with moving off vector fields (i.e. by definition, their Lie brackets are zero) then a flow sequence composed of them can be rearranged. The new order makes a simplification possible by eliminating some flows from the sequence.

Corollary 1. Switching between higher and lower strata is possible if moving off vector fields are decoupled from all the moving on vector fields defined on the substratum and higher level strata (in other words, their Lie brackets are zero).

Remark 5. Kinematic model of six-legged robot has special properties. There exist a (linear) combination of control inputs such that their activated vector fields point out a tangent vector

$$\frac{\partial}{\partial h_i} \tag{2.54}$$

where h_i is the height of a foot from the ground. Furthermore, the remaining part of equations of motion is decoupled from the foot height. Such properties guarantee that condition for commutation in Proposition 1 and Corollary 1 holds.

Example 6 illustrates well, how moving on and moving off vector fields are related to stratified view. Vector fields $g_{12,1}$, $g_{12,2}$, $g_{1,1}$, $g_{1,2}$, $g_{2,1}$, $g_{2,2}$ from (2.39)-(2.42) are moving on vector fields because they keep system on the stratum they are defined on. At the same time, however, $g_{0,3}$, $g_{0,4}$, may switch from S_0 to S_1 or S_2 under "suitable" u_4 or u_3, so they are moving off vector fields. Even their dimension change from 7 to 6 when they reach a lower stratum. Similar situation occurs in S_1 under certain u_4 which activates $g_{1,3}$. This vector field is capable to switch system from S_1 to bottom stratum S_{12} forcing dimension reduction from 6 to 5. Vector field $g_{2,3}$ has the same effect in S_2 concluding that $g_{1,3}$ and $g_{2,3}$ are also moving off vector fields. Another way to discover moving off vector fields consists in realizing that some inputs are constrained in strata. For instance, u_3 and u_4 are constrained to be 0 in S_{12}, otherwise they would activate moving off vector fields that switch the system from bottom stratum S_{12} to S_1 or S_2 which would contradict to the fact that (2.39) is the equation of motion of S_{12}. Same phenomena explains why inputs are constrained in higher strata. The results above make a foundation to the basic stratified motion planning algorithm summarized now.

Algorithm 3. [Stratified Motion Planning, Goodwine (1998a)]

Step 1. Determine the *multiple stratified system.* The equations of motion in the strata are

$$
\begin{aligned}
S_b \,:\, \dot{x} &= g_{0,1}u_{0,1} + \cdots + g_{0,n_0}u_{0,n_0} \\
S_1 \,:\, \dot{x} &= g_{1,1}u_{1,1} + \cdots + g_{1,n_1}u_{1,n_1} \\
&\vdots \\
S_I \,:\, \dot{x} &= g_{I,1}u_{I,1} + \cdots + g_{I,n_I}u_{I,n_I}
\end{aligned}
$$

Step 2. Create *bottom stratified system.* It is clear from Theorem 4 and Proposition 1 that stratified motion planning algorithm requires a special system on the bottom stratum (as a common space). Note, this system is also a fictitious system but without any connection to the fictitious system in smooth motion planning (see Subsection 2.1.2). In order to obtain this special (i.e. bottom stratified) system, one has to create a union set of vector fields in all strata and define with them a system

$$
\begin{aligned}
\dot{x} &= g_{0,1}u_{0,1} + \cdots + g_{0,n_0}u_{0,n_0} \\
&+ \; g_{1,1}\,|_{S_b}\, u_{1,1} + \cdots + g_{1,n_1}\,|_{S_b}\, u_{1,n_1} \\
&\vdots \\
&+ \; g_{I,1}\,|_{S_b}\, u_{I,1} + \cdots + g_{I,n_I}\,|_{S_b}\, u_{I,n_I}
\end{aligned}
\tag{2.55}
$$

where the notation $|_{S_b}$ refers to vector fields taking part in bottom stratified system, however, they are defined originally not in this stratum. It should consist of all the moving on vector fields that commute with the moving off vector fields. Note, that a vector field is S_b means that one "cuts" some dimensions from the original vector field.

Step 3. Create the *bottom stratified extended system*

$$\begin{aligned}\dot{x} &= g_{0,1}u_{0,1} + \cdots + g_{0,n_0}u_{0,n_0} \\ &+ g_{1,1}\,|_{S_b}\, u_{1,1} + \cdots + g_{1,n_1}\,|_{S_b}\, u_{1,n_1} \\ &\vdots \\ &+ g_{I,1}\,|_{S_b}\, u_{I,1} + \cdots + g_{I,n_I}\,|_{S_b}\, u_{I,n_I} \\ &+ \textit{Lie brackets} \qquad (2.56)\end{aligned}$$

including the vector fields of bottom stratified system and the Lie brackets among them. This system is the extension of bottom stratified system according to the way mentioned by smooth motion planning algorithm in Subsection 2.1.2.

Step 4. Solve the smooth motion planning problem on the bottom stratified extended system. Algorithm 1 applied on bottom stratified extended system also solves indirectly (indirectly because operating only with moving on vector fields) the stratified motion planning problem. It results a sequence of flows along the moving on vector fields specified by three sequences:

a. Sequence of moving on vector fields. A flow in the sequence is defined along the corresponding vector field in this sequence of moving on vector fields.

b. Sequence of time. The element of sequence defines how much time is needed to move along a vector field.

c. Sequence of inputs. Smooth motion planning algorithm sets the absolute value of inputs to 1, however, one needs to define a sequence that defines the sign of the inputs.

Step 5. Complete the solution for the stratified motion planning problem. If two neighboring flows in the sequence are defined in different strata, one has to insert moving off vector fields between them in order to switch between their strata. The extra flows along the moving off vector fields have to assure that constraints change in accordance with the change of strata. It is possible on the base of Proposition 1 and Corollary 1.

Chapter 3

Improvements for Basic Stratified Motion Planning

Stratified motion planning reviewed in previous chapter inherits some key properties from smooth motion planning that relies on flow sequence philosophy. Obviously, some difficulties coming from smooth motion planning are swelling on, too. This chapter focuses on two specific problems which were not regarded in Algorithm 3.

First problem concerns convergency. Basic stratified motion planning algorithm works out a point to point motion planning problem , which is only an approximation in general case and convergency is not guaranteed in every circumstances. Section 3.1 proposes an algorithm to handle the problem.

Second problem arises if one desires to steer the system to final configuration during given time interval. Motion planning algorithms discussed in Chapter 2 investigated the motion without respect to time. Section 3.2 presents a simple algorithm how the original methods can be extended to resolve time-scaling.

Both modifications are demonstrated on the motion planning of six-legged robot in Section 3.3. Reader finds details about contributions made in this section in Harmati, Lantos, and Payandeh (1999), Harmati, Lantos, and Payandeh (2000a), Harmati, Kiss, and Lantos (2000) and Harmati and Kiss (2001). If anything else is not said, smooth motion planning refers to smooth motion planning using flow sequence in this chapter.

3.1 Improvements of convergence property

This section addresses convergence problem of basic stratified motion planning reviewed in Subsection 2.2.3. The results presented here were reported also in Harmati, Lantos, and Payandeh (1999) and Harmati, Lantos, and Payandeh (2000a).

Convergence problem emerged first in Subsection 2.1.2 where motion planning endeavored to reach a desired final configuration x_F for smooth system using flow sequence. Considering this case first, Remark 1 revealed and Example 3 exhibited that achieving precisely desired final state is possible only if the smooth system is nilpotent. Otherwise, the solution is only an approximation. It is shown in Lafferriere and Sussmann (1991), however, that there exists a critical distance between the initial and final states in the sense that the reached state is closer to desired final state than to initial state. As a matter of fact, the work mentioned proposes an iterated algorithm to set up critical distance by inserting extra reference points. Applying iterated method is enforced by the fact that theoretical result is unfortunately not known for obtaining critical distance in general case. Similar idea for ensuring convergence is adopted in the algorithm to be presented here, however, it shows some differences in comparison with the conception of Lafferriere and Sussmann (1991). One of the main reasons to devise a new step length adaptation algorithm comes from a slightly different problem statement. Namely, smooth motion planning (and basic stratified motion planning considered later on) achieves or approximates only one desired final point and there is no freedom to design a path between initial state x_I and final states x_F. On contrary, the problem considered here takes into account several reference points, as follows.

Let a desired trajectory be given, though it is known it cannot be followed by the algorithms mentioned above. Note, in strict sense trajectory describes time relations while a path does not. However, path and trajectory cannot be separated clearly in this work because motion planning introduced is focusing on path planning, however it results a trajectory. Some of the next chapters intends to endow reference path with time relations that proceeds toward trajectory planning, however it will not define an exact trajectory if only the desired time for final point is prescribed (as occurs in next section). The primary goal now is to follow the path defined by trajectory disregarding time relation, therefore if it is not confusing trajectory and path are used as synonyms in this context. (To satisfy desired time relations at reference points will play central role only in the next section.) Taking extra reference points from reference trajectory, one attains subtrajectories. It is possible now to set up motion planning problem in every subsegment and carry out the smooth motion planning algorithm there. The initial point for smooth motion planning algorithm in a subsegment coincides with the point achieved in the preceding subsegment. The desired final state to be reached by smooth motion planning algorithm coincides with the point designated by the actual subsegment. The initial state in the first subsegment is x_I, the desired final state in the last subsegment is x_F. Such a treatment obviously manages to keep resulting path close to the desired one. Some points of resulting path, namely the points corresponding to final points of subsegments, lie close enough to reference path. Deviations depend mainly on the refinement realized by the number of extra reference points which determines the length of subsegments in equidistant case. It is indirectly expected that better refinement of representative points implies smaller deviations for the entire trajectory as well.

Algorithm 4. [Adaptive step length modification]

Step 1. Let $x(t)$ be the desired trajectory where $0 \leq t \leq T$ and let $x_I = x(0)$, $x(T) = x_F$. Let be given an integer r that measures the refinement along $x(t)$.

Step 2. Define $r + 1$ representative points by the sequence

$$x_i = x(\frac{i}{r}\,T) \quad i = 0, \ldots, r \tag{3.1}$$

where $x_0 = x_I$, $x_r = x_F$.

Step 3. Let $y_0 = x_0$ and set $i = 1$, $k = 3$, $F = 1/3$, $S = 1/2$.

Step 4. $D_1 = \|x_1 - y_0\|$, nextmod $= 1$.

Step 5. Perform Algorithm 1 to go from y_{i-1} to x_i. Denote the resulting end point in ith subsegment by y_i.

Step 6. If $\|x_i - y_i\| \geq F\,D_i$ and $i < r$ then

1. Insert k extra points uniformly along a straight-line between y_i and x_{i+1}:

$$c_j = y_i + \frac{j}{(k+1)} \frac{(x_{i+1} - y_i)}{\|x_{i+1} - y_i\|}, \quad j = 1, \ldots, k \tag{3.2}$$

2. $\{x_{i+1}, \ldots, x_{r+k}\} = \{c_1, \ldots, c_k, x_{i+1}, \ldots, x_r\}$

3. $r = r + k$

4. nextmod $= i + k$

elseif $i \geq$ nextmod and $i < r - 1$ and $\|x_i - y_i\| < S\,F\,D_i$ then

1. Delete the next representative point, i.e.: $\{x_{i+1}, \ldots, x_{r-1}\} = \{x_{i+2}, \ldots, x_r\}$

2. $r = r - 1$,

Step 7. If $i = r$ go to next step. Else $i = i + 1$, $D_i = \|x_i - y_{i-1}\|$ and go to Step 5

Step 8. Make a resultant sequence of flows by joining together sequences obtained in each subsegments.

Algorithm 4 starts with an initial distance between the first two representative points. The first representative point is the initial state. After performing smooth motion planning algorithm based on sequence of flows, the algorithm checks if final state y_i reached is close enough to the desired x_i in the actual subsegment. Point y_i is considered to be close enough to x_i if it is found inside a circle with origin x_i and with a radius that equals to the distance between x_{i-1} and x_i multiplied by F. If resulting point y_i is not close enough to x_i then k extra representative points are to be inserted between y_i and x_{i+1} along a straight-line that reduces distances between neighboring points in the next $k+1$ subsegments. It obviously improves accuracy and distance between initial and final points in the actual subsegment, the length of subsegments are decreasing toward critical distance. However, extra points generate extra subsegments and increase computational time which may risk real-time realization. Hence, instead of approaching actual desired final state x_i (as it happened in Lafferriere and Sussmann (1991)) if not acceptable error occurs, the algorithm considers the next representative point x_{i+1} and, extra points $c_1, \ldots c_k$ realizing a better refinement manage to drive the system there. Another way to reduce computational time is to decrease the number of representative points if accuracy exceeds the length of actual subsegment multiplied by $S\,F$ where S is a safety factor and usually $S < 1$. A cautious approach is realized in Algorithm 4 which does not delete a representative point if it was inserted as an extra point. This strategy aims to stabilize accuracy. On the other hand, one can see that step length, i.e. the distance between initial and final states in the actual subsegment always changes keeping the accuracy between $F\,D_i$ and $S\,F\,D_i$. Such a philosophy facilitates to adapt the length of subsegments to the critical distance, if $S\,F < 1/2$ and the number of representative points exceeds a critical value. There is no general result for obtaining critical value, since there is no theoretical results how to obtain critical distance either. Observe, using many representative points along desired trajectory x_d helps the system to reach many points along a trajectory, even though it was devised only to achieve a final point. Still, the problem remains discontinuous. Whatever small is the length of subsegments, one cannot effectively influence the system's motion between the two end points of subsegments.

Problem of convergency appears also in stratified motion planning. Its algorithm relies on smooth motion planning using flow sequences, therefore the question arises whether Algorithm 4 can be applied to stratified motion planning.

Proposition 2. Algorithm 4 can be inserted in the frame of stratified motion planning (Algorithm 3).
Proof: Algorithm 3 carries out a smooth motion planning (Algorithm 1) on bottom stratified extended system. For, Algorithm 4 can be applied on bottom stratified extended system. On the other hand, Algorithm 4 does not have any influence on the assumptions required for moving on and moving off vector fields. The solution of stratified motion planning problem contains extra flows along moving off vector fields, which do not have impact on accuracy, they only switch between strata. In addition, every switch is isolated from Algorithm 4, therefore, Algorithm 4 can be inserted in the frame of stratified motion planning based on flow sequences applying it to bottom stratified extended system in Algorithm 3.

The next algorithm combines Algorithm 3 and Algorithm 4 endowing stratified motion planning with adaptive step length modification.

Algorithm 5. [Stratified Motion Planning using adaptive step length modification]
Step 1. Determine multiple stratified system.
Step 2. Create bottom stratified system.
Step 3. Create bottom stratified extended system.
Step 4. Carry out Algorithm 4 on bottom stratified extended system.
Step 5. Insert necessary moving off vector fields into the sequence of flows that completes the solution for stratified motion planning problem.

3.2 Time-scaling

This section studies time-scaling problem for basic stratified motion planning reviewed in Subsection 2.2.3. The philosophy presented here was reported also in Harmati, Lantos, and Payandeh (2000a), Harmati, Kiss, and Lantos (2000) and Harmati and Kiss (2001).

First, recall that basic stratified motion planning incorporates a smooth motion planning operating with flow sequences. According to (2.13), Algorithm 1 results a sequence of flows

$$x_F \approx \Phi(B_z, t_z) \circ \Phi(B_{z-1}, t_{z-1}) \circ \cdots \circ \Phi(B_1, t_1)(x_I) \tag{3.3}$$

where x_F is exactly reached if the system is nilpotent, otherwise, flow sequence results an approximation. Vector fields $B_1, \ldots B_z$ symbolize the Philip Hall basis while $t_1, \ldots, t_z$ assign the length of time intervals where B_1 is activated by its corresponding input if $0 \leq t < t_1$, B_2 is activated by its corresponding input if $t_1 \leq t < t_2$ and so on. A vector field is made inactive by setting zero its corresponding input. Time intervals $t_1, \ldots, t_z$ define the (forward) Philip Hall coordinates. The system requires time

$$T_s = \sum_{i=1}^{z} t_i \tag{3.4}$$

to travel from x_I to x_F. Algorithm 1 and Algorithm 3 explored exclusively how to reach desired final state x_F without respect to time.

Many applications expect the system to arrive at desired final state x_F at a desired time T while the time of initial state is also given, i.e.

$$x(0) = x_I, \qquad x(T) = x_F \tag{3.5}$$

Note, initial time at initial point may differ from $t = 0$, for instance if one considers subsegments during motion planning as occured in Algorithm 4 and Algorithm 5. The contribution of this chapter is to propose treatments for this problem both in smooth and stratified motion planning if they use flow sequences.

Proposition 3. Consider a motion planning problem on smooth nonlinear system of type (2.11) with initial state x_I and desired final state x_d . Let the solution to motion planning problem be given in the form

$$x_d \approx x_F = \Phi(B_z, t_z, u^z) \circ \Phi(B_{z-1}, t_{z-1}, u^{z-1}) \circ \cdots \circ \Phi(B_1, t_1, u^1)(x_I) \tag{3.6}$$

where according to Algorithm 1, the active input in the ith time interval is $|u^i| = |u_j| = 1$ for some $j \in \{1, \ldots, m\}$ and for every $i = 1, \ldots, z$. If $x(0) = x_I$ is the initial condition and $x(T) = x_F$ is the end condition to be achieved with respect to time, then

$$x_F = x(T) = \Phi(B_z, \frac{T}{T_s} t_z, u_T^z) \circ \Phi(B_{z-1}, \frac{T}{T_s} t_{z-1}, u_T^{z-1}) \circ \cdots \circ \Phi(B_1, \frac{T}{T_s} t_1, u_T^1)(x(0)) \tag{3.7}$$

where

$$u_T^i = \frac{T_s}{T} \mathrm{sign}(u^i), \qquad T_s = \sum_{i=1}^{z} t_i \tag{3.8}$$

for every $i = 1, \ldots, z$.

Proof: Consider first a trajectory that satisfies the simple differential equation

$$\dot{x}(t) = g(x(t)) \tag{3.9}$$

with the initial condition $x(0) = x_I$. Let $x_F = x(T)$ be chosen for arbitrary time T such that $T > 0$. Since $g(x(t))$ determines the tangent vector of $x(t)$ (being a one dimensional manifold) for every $t > 0$, it holds that $g(x(t))u$ directs the integral curve into the same direction as $g(x(t))$ at any $x \in x(t)$ for any real valued $u(t)$. The term u determines however the absolute value of the "velocity", i.e. the length of tangent vector. Assume, $u(t) = U$, i.e. it is a constant. Then $x_u(t)$ is the solution of the differential equation

$$\dot{x}_u(t) = g(x_u(t))U. \tag{3.10}$$

Since U influences only the absolute value of tangent vector at any point of the integral curve $x(t)$, but it does not change the direction of the vector, one gets $x(TU) = x_u(T)$. It means that $x(t)$ and $x_u(t)$ travels through the same points, however at other times. Times differ in a constant term U. As a conclusion, if original velocity on a curve is multiplied by U, one needs to divide original time by U to reach the same final point.

Now, consider the solution (3.6) of the system (2.11) having m inputs. Only one input is active at a time t, which allows only one active vector field. Let the first subsequence of (3.6) be

$$x_1^b = \Phi(B_1, t_1, u^1)(x(0)). \tag{3.11}$$

and define the first subsequence of (3.7)

$$x_{1,u}^b = \Phi(B_1, \frac{t_1}{u_T^1}, u_T^1)(x(0)). \tag{3.12}$$

Using the arguments above (with $|u^1| = 1$ and $\text{sign}(u_T^1) = \text{sign}(u^1)$), one yields $x_1^b = x_{1,u}^b$. If

$$u_T^1 = \frac{T_s}{T} \tag{3.13}$$

then flow subsequence from (3.12) needs time

$$t_1^T = \frac{Tt_1}{T_s}$$

to reach the same point x_1^b while subsequence from (3.11) requires t_1. One can repeat the procedure for the second subsequence using initial point x_1^b

$$x_2^b = \Phi(B_2, t_2, u^2)(x_1^b(0)) \tag{3.14}$$

$$x_{2,u}^b = \Phi(B_2, \frac{t_2}{u_T^2}, u_T^2)(x_1^b(0)) \tag{3.15}$$

setting the initial time to zero. The point x_2^b to be reached, the subsequence of (3.15) needs time

$$t_2^T = \frac{Tt_2}{T_s}$$

instead of t_2 as required for (3.14). Repeating the procedure for every subsequences, composition created by the sequences (3.11), (3.14) and so on leads to (3.6). On the other hand, the trajectory $x_u(t)$ defined by (3.12), (3.15) and so on with points $x_{1,u}^b, \ldots, x_{z,u}^b$ needs

$$\sum_{i=1}^{z} t_i^T = \sum_{i=1}^{z} \frac{Tt_i}{T_s} = \frac{T}{T_s} \sum_{i=1}^{z} t_i = \frac{TT_s}{T_s} = T \tag{3.16}$$

to reach x_F which means that $x_u(T) = x_F$. By definition, $x_u(t)$ is exactly the trajectory of (3.7), hence the proof is completed.

Results of Proposition 3 can be applied if one desires to prescribe when the system should reach the desired final point.

Corollary 2. Proposition 3 remains true under initial condition $x_I = x(t_I)$, $t_I > 0$.
Proof: Straightforward from the proof of Proposition 3 by simple substitution.

As the proof of Proposition 3 revealed, the time intervals which separate active vector fields are changed if time-scaling is applied. If a Philip Hall coordinate is originally t_i, then after time-scaling it changes to

$$t_i^T = \frac{T t_i}{T_s}.$$

Time-scaling problem also concerns basic stratified motion planning, which is not surprising, since the method relies on Algorithm 1 and operates with sequence of flows. Switches between strata performed by flows along moving off vector fields appear in flow sequence as solution to motion planning problem , making slight difference between smooth and stratified conception. Moving off flows are inserted into sequence after carrying out Algorithm 1 on bottom stratified extended system. By using kinematic model, the length of time intervals along moving off vector fields do not play important role, the only condition expected for them to be greater than zero, which is sufficient to switch between two strata. Recalling for example biped robot, it is indifferent how far one leg arises from the ground, the important fact which affects robot's motion is that contact is broken. Therefore, active time for moving off vector fields can be chosen almost arbitrarily under the only condition that a switch from stratum S_i to stratum S_j requires same time as the switch from S_j to S_i. For simplicity, active time for all the moving off vector fields will be uniformly set in t^f. Assume, smooth motion planning algorithm on bottom stratified extended system (without time-scaling) yields a flow sequence

$$x_F = \Phi(B_z, t_z, u^z) \circ \Phi(B_{z-1}, t_{z-1}, u^{z-1}) \circ \cdots \circ \Phi(B_1, t_1, u^1)(x_I). \tag{3.17}$$

In order to complete stratified motion planning, one requires to insert moving off vector fields between B_i and B_{i+1}, $i \in \{1, \ldots, z-1\}$, if they are defined in different strata. Based on this, (3.17) is restated as

$$\begin{aligned} x_F = \quad & \Phi(B_z, t_z, u^z) \quad \circ \cdots \circ \Phi(B_p^f, t^f, u^{p,f}) \circ \cdots \circ \\ & \Phi(B_j, t_j, u^j) \quad \circ \cdots \circ \Phi(B_1^f, t^f, u^{1,f}) \circ \cdots \circ \Phi(B_1, t_1, u^1)(x_I). \end{aligned} \tag{3.18}$$

where $\left|u^{i,f}\right| = 1$, and B_i^f denotes a moving off vector field for every $i \in \{1, \ldots, p\}$.

Proposition 4. If stratified motion planning claims p moving off vector fields after smooth motion planning algorithm is completed on bottom stratified extended system, and T^{str} is the desired time for achieving x_F, i.e. $x(T) = x_F$ then Proposition 3 with $T = T^{str} - p\, t^f$ accomplishes a time-scaling on stratified motion planning.
Proof: Straightforwardly comes from the reasonings of proof followed in Proposition 3. Considering exclusively moving on vector fields from (3.18), one gets (3.17). On this flow, one can directly carry out a smooth time-scaling according to Proposition 3. Let the desired time $T = T^{str} - p\, t^f$. Now, inserting back moving off vector fields, the only change in time relations that the time taken by moving off vector fields should be accounted. Time spent for a single moving off vector field takes, by definition, t^f, so p moving off vector fields pt^f time altogether. It means that stratified motion planning needs time $T + p\, t^f = (T^{str} - p\, t^f) + p\, t^f = T^{str}$ which was originally desired.

Proposition 3 and Proposition 4 give constructive way for time scaling. Based on this, Algorithm 5 of stratified motion planning extended by time-scaling and adaptive step length modification can be summarized as follows.

Algorithm 6. [Improved Stratified Motion Planning. (Endowed with adaptive step length modification and time scaling)]

Step 1. Determine multiple stratified system.

Step 2. Create bottom stratified system.

Step 3. Create bottom stratified extended system.

Step 4. Carry out Algorithm 4 on bottom stratified extended system.

Step 5. Insert necessary moving off vector fields into the sequence of flows that completes the solution for stratified motion planning problem. All the moving off vector fields are uniformly activated for time $t^f > 0$.

Step 6. Perform time-scaling based on Proposition 4.

3.3 Example

This chapter intends to illustrate the results of Subsection 3.1 and Subsection 3.2 on the simple model of six-legged robot. Equations of motion that steer robot in the plane are adopted from Example 6. Consider a motion planning problem with the representative points

$$x_I = x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad x_{F1} = x(T_1) = \begin{pmatrix} 4 \\ 4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad x_{F2} = x(T_2) = \begin{pmatrix} 8 \\ 10 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.19}$$

where the robot starts from x_I and it should reach the configuration $x_{F1} = x(T_1)$ at $T_1 = 2$ sec, and the configuration $x_{F2} = x(T_2)$ at $T_2 = 4$ sec, respectively. Observe, robot should change only its position in final configurations, remaining state variables are expected to keep their values. It is permitted, however, the robot to change state variables between representative points. In fact it is necessary, for example orientation θ to be changed from initial value 0, otherwise the robot, since having coupled equations of motion, could not reach desired positions by means of control vector fields. As seen, desired time points are given at representative points, calling for time-scaling, therefore Algorithm 6 is to be applied. In order to highlight the benefits of proposed improvements related to convergency, simulation results of algorithm omitting step length adaptation are also presented.

Basic stratified motion planning starts with multiple stratified system. Expressions (2.39)-(2.42), the equations that define motions in strata, make up this system for six-legged robot. Subsystems in strata can delegate maximally 4 independent basis of tangent space into bottom stratum establishing bottom stratified system(BSS). Every basis should be fitted to the dimension of this stratum meaning that they are to be "cut off" to dimension 5. Note, the dimensions omitted in this way are related to leg heights, they do not influence the shape of path performed by the robot, but they play important role in selecting the set of active vector fields. Selected vector fields represent the basis and the equation of bottom stratified system reads

$$\Sigma^{BSS}: \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_2 \end{pmatrix} = [g_{1,1}\,|_{S_{12}} \quad g_{2,2}\,|_{S_{12}} \quad g_{1,2}\,|_{S_{12}} \quad g_{2,1}\,|_{S_{12}}] \begin{pmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} \\ u_{2,1} \end{pmatrix} \tag{3.20}$$

$$= \begin{bmatrix} cos(\theta) & cos(\theta) & 0 & 0 \\ sin(\theta) & sin(\theta) & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} u_{1,1} \\ u_{2,2} \\ u_{1,2} \\ u_{2,1} \end{pmatrix}$$

For the sake of simplicity, notation $|_{S_{12}}$ will be omitted if context uniquely points out that a vector field is considered in bottom stratum. Lie bracket $g_1^e = [g_{1,1}, g_{2,2}]$ produces a vector field which is independent of $g_{1,1}$, $g_{1,2}$, $g_{2,1}$ and $g_{2,2}$ concluding that Σ^{BSS} satisfies the LARC:

$$\text{rank}([g_{1,1}\ g_{2,2}\ g_{1,2}\ g_{2,1}\ g_1^e]) = 5.$$

These vector fields constitute the bottom stratified extended system Σ_{ext}^{BSS} and they give simultaneously the Philip Hall basis for smooth motion planning.

First, let the time moving off vector fields require be disregarded, i.e. $t^f = 0$, uniformly. (The modification to be carried out for stratified motion planning if $t^f > 0$ will be discussed later on.) This simplification allows the smooth motion planning on bottom stratified system to inherit time relations from stratified case, so prescribed times at representative points are still $T_1 = 2$ and $T_2 = 4$.

Performing smooth motion planning algorithm based on sequence of flows without taking into account any algorithm that improves convergency (such as Algorithm 4), but using time-scaling the solution in the first subsegment (between x_I and x_{F1}) yields

$$x_{F1}^r = \Phi(g_{2,1}, 0.5, -4) \circ \Phi(g_{1,2}, 0.5, -4) \circ \Phi(g_{2,2}, 0.5, 4) \circ \Phi(g_{1,1}, 0.5, 4)(x_I). \tag{3.21}$$

where $x_I = (0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0)^T$ and $x_{F1}^r = (1.82,\ 2.83,\ 0,\ 0,\ 0,\ 0,\ 0)^T$. Adding time of flows together, $x_{F1}^r = x(2)$, the system achieves x_{F1}^r at $t = T_1$ as desired. Deviation between the reached x_{F1}^r and desired x_{F1}, however, is considerable, which is seen in Figure 3.1. Resolving motion planning problem in second subsegment,

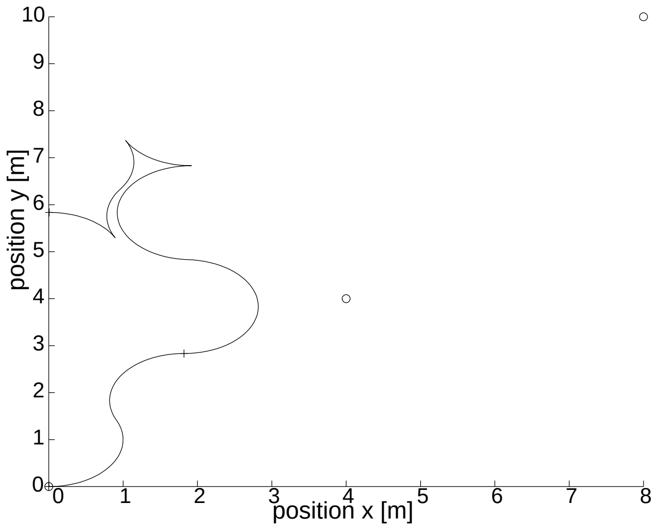

Figure 3.1: Stratified motion planning without improving convergency. Marks "o" denote the representative points x_I, x_{F1} and x_{F2} while marks "+" denote reached points the trajectory passing through: x_I, x^r_{F1} and x^r_{F2}.

with initial state x^r_{F1} and desired state x_{F1}, the solution proceeds as

$$\begin{aligned} x^r_{F2} &= \Phi(g_{2,2}, 0.1305, -8.37) \circ \Phi(g_{1,1}, 0.1305, 8.37) \circ \Phi(g_{2,2}, 0.1305, 8.37) \circ \Phi(g_{1,1}, 0.1305, -8.37) \circ \\ &\circ \Phi(g_{2,1}, 0.3695, -8.37) \circ \Phi(g_{1,2}, 0.3695, -8.37) \circ \Phi(g_{2,2}, 0.3695, 8.37) \circ \Phi(g_{1,1}, 0.3695, 8.37)(x^r_{F1}) \end{aligned} \tag{3.22}$$

where $x^r_{F2} = (0.01,\ 5.84,\ 0,\ 0,\ 0,\ 0,\ 0)^T$. Observe, the last 4 terms in flow sequence (i.e. the first 4 ones in writing order) approximates a flow along $g^e_1 = [-g_{1,1}, g_{2,2}]$ (see (A-34)). In order to obtain total flow sequence, one should replace x^r_{F1} in (3.22) with the flow sequence in (3.21). Adding time of flows in (3.22) to initial time of x^r_{F1}, $x^r_{F2} = x(4)$, i.e. desired time-scaling is accomplished. The weakness of algorithm in convergency is disclosed if one compares reached configuration x^r_{F2} with desired configuration x_{F2}. Error illustrated also in Figure 3.1 proves divergency in motion planning.

To improve convergency, apply Algorithm 6 that incorporates Algorithm 4. Results are illustrated in Figure 3.2 Algorithm in the first subsegment leads to the same result achieving x^r_{F1}. At this point, $\|x_{F1} - x^r_{F1}\| \geq F\,D_i$ ($\|x_{F1} - x^r_{F1}\| = 2.43$, $F = 1/3$, $D_1 = 5.66$), and Step 6 of Algorithm 4 recognizes developing divergency and intervenes by inserting 3 extra representative points along straight line between x^r_{F1} and x_{F2}: $c_1 = (3.36,\ 4.62,\ 0,\ 0,\ 0,\ 0,\ 0)^T$, $c_2 = (4.91,\ 6.41,\ 0,\ 0,\ 0,\ 0,\ 0)^T$, $c_3 = (6.45,\ 8.20,\ 0,\ 0,\ 0,\ 0,\ 0)^T$. Let the time is divided uniformly between x^r_{F1} and x_{F2}, i.e. let the time prescribed at c_1, c_2 and c_3 equal to $T^1_c = 2.5$, $T^2_c = 3$, $T^3_c = 3.5$, respectively. (Note, $x_{F2} = x(4)$ is desired.) Of course, it is possible to choose other k, F in Algorithm 4, as well. The idea behind choosing $F = 1/3$ is that if a deviation between desired and reached final state is permanently less than 33% of subsegment length then the necessary condition to reach critical distance (requiring maximum 50% deviation) holds with safety. Inserting 3 points comes from the philosophy as follows. Next subsegment length is assumingly close to the actual one, and considering the worst case, the

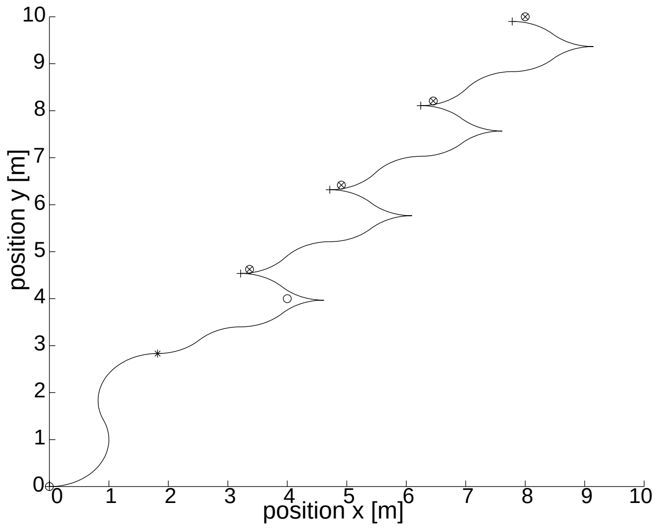

Figure 3.2: Stratified motion planning using improvements for convergency. Marks "o" denote the final representative points x_I, x_{F1}, c_1, c_2, c_3 and x_{F2} while marks "+" denote reached points the trajectory passing through: x_I, x^r_{F1}, c^r_1, c^r_2, c^r_3 and x^r_{F2}.

distance between initial and final state in the next subsegment is around $(F+1)D_i = 4/3D_i$. If one inserts 3 extra points, the length of next subsegment is changed, 4 times shorter than it was originally, i.e

$$D_{i+1} \approx \frac{1}{4}(F+1)\,D_i = \frac{1}{4}\frac{4}{3}D_i = \frac{1}{3}D_i = F\,D_i \tag{3.23}$$

which means that the length of next subsegment will be approximately equal to the threshold applied to error in divergent case.

Executing motion planning in second subsegment results the flow sequences

$$\begin{aligned} c^r_1 &= \Phi(g_{2,2}, 0.0625, -12.36) \circ \Phi(g_{1,1}, 0.0625, -12.36) \circ \Phi(g_{2,2}, 0.0625, 12.36) \circ \Phi(g_{1,1}, 0.0625, 12.36) \circ \\ &\circ \Phi(g_{2,1}, 0.0625, -12.36) \circ \Phi(g_{1,2}, 0.0625, -12.36) \circ \Phi(g_{2,2}, 0.0625, 12.36) \circ \Phi(g_{1,1}, 0.0625, 12.36)(x^r_{F1}) \end{aligned}$$

where $c^r_1 = x(2.5) = (3.21,\ 4.54,\ 0,\ 0,\ 0,\ 0,\ 0)^T$.

As before, the last 4 terms in flow sequence (i.e. the first 4 ones in writing order) approximates a flow along $g^e_1 = [g_{1,1}, g_{2,2}]$ (see (A-34)). Executing motion planning in 3rd subsegment results the flow sequences

$$\begin{aligned} c^r_2 &= \Phi(g_{2,2}, 0.0592, -12.87) \circ \Phi(g_{1,1}, 0.0592, -12.87) \circ \Phi(g_{2,2}, 0.0592, 12.87) \circ \Phi(g_{1,1}, 0.0592, 12.87) \circ \\ &\circ \Phi(g_{2,1}, 0.0658, -12.87) \circ \Phi(g_{1,2}, 0.0658, -12.87) \circ \Phi(g_{2,2}, 0.0658, 12.87) \circ \Phi(g_{1,1}, 0.0658, 12.87)(c^r_1) \end{aligned}$$

where $c^r_2 = x(3) = (4.71,\ 6.32,\ 0,\ 0,\ 0,\ 0,\ 0)^T$.

Executing motion planning in 4th subsegment results the flow sequences

$$\begin{aligned} c^r_3 &= \Phi(g_{2,2}, 0.0579, -12.98) \circ \Phi(g_{1,1}, 0.0579, -12.98) \circ \Phi(g_{2,2}, 0.0579, 12.98) \circ \Phi(g_{1,1}, 0.0579, 12.98) \circ \\ &\circ \Phi(g_{2,1}, 0.0671, -12.98) \circ \Phi(g_{1,2}, 0.0671, -12.98) \circ \Phi(g_{2,2}, 0.0671, 12.98) \circ \Phi(g_{1,1}, 0.0671, 12.98)(c^r_2) \end{aligned}$$

where $c_3^r = x(3.5) = (6.25,\ 8.10,\ 0,\ 0,\ 0,\ 0,\ 0)^T$.

Executing motion planning in the final subsegment results the flow sequences

$$\begin{aligned} x_{F2}^r &= \Phi(g_{2,2}, 0.0575, -13.01) \circ \Phi(g_{1,1}, -0.0575, 13.01) \circ \Phi(g_{2,2}, 0.0575, 13.01) \circ \Phi(g_{1,1}, 0.0575, 13.01) \circ \\ &\circ \Phi(g_{2,1}, 0.0675, -13.01) \circ \Phi(g_{1,2}, 0.0675, -13.01) \circ \Phi(g_{2,2}, 0.0675, 13.01) \circ \Phi(g_{1,1}, 0.0675, 13.01)(c_3^r) \end{aligned}$$

where $x_{F2}^r = x(4) = (7.78,\ 9.9,\ 0,\ 0,\ 0,\ 0,\ 0)^T$.

Comparing x_{F2}^r with x_{F2}, one can realize that distance between them is 0.24 which corresponds 10% error with respect to the newly created last subsegment defined by c_3 and x_{F2}, and only 3% error with respect to the original subsegment defined by x_{F1} and x_{F2}.

In order to obtain the total solution, one has to consider $t^f > 0$ which was omitted so far. For, the first step is to concatenate the flow sequences. Doing this, it is easy to realize that every flow is defined in other strata than its neighbors. Since total sequence consists of 32 flows, one need to insert 31 moving off vector fields which takes $31\,t^f$ additional time. Based on this, one can rescale the flow sequence. For example, if $t^f = 0.1$sec and $T_1 = 5.1$sec, $T_2 = 7.1$sec are the desired time at representative points then by using Proposition 4 motion planning problem on Σ^{BSS} results in $T_1^{new} = T_1 - 31\,t^f = 2$, $T_2^{new} = T_2 - 31\,t^f = 4$ (which is the task that have been solved above).

Finally, there is a phenomena which may be worth being discussed. Motion planning in general aims to steer the robot along a reference path in plane. Although, stratified methods presented here so far are able to approximate a final point including positions and orientation as state variables but unfortunately they cannot ensure the resulting path to be close enough to reference path between representative points. Figure 3.3 and Figure 3.4 illustrate how much the trajectory tracking error may depend on prescribed state variables if point to point motion planning algorithm is used. If desired orientation does not coincide with tangent vector of trajectory at representative points as illustrated in Figure 3.3 then robot moves along a path which has considerable deviation from reference path. On the other hand, if desired orientation does coincide with tangent vector of trajectory at representative points as illustrated in Figure 3.4 then robot approximates well the reference path. The phenomena is of great importance if obstacles lie on the ground and collision is not preferable.

3.4 Summary and Conclusions

This chapter proposed some improvements for basic stratified motion planning that increase robustness. Two problems were investigated, the convergency and time-scaling.

Algorithm 4 provided a technique how to insert extra representative points in order to improve convergency. The method starts from the idea of Lafferriere and Sussmann (1991) but it contains modifications that fits to motion planning problem consisting of numerous representative points to be passed through. Proposition 2 asserted that the algorithm can be integrated into stratified frame and Algorithm 5 provided a methodology

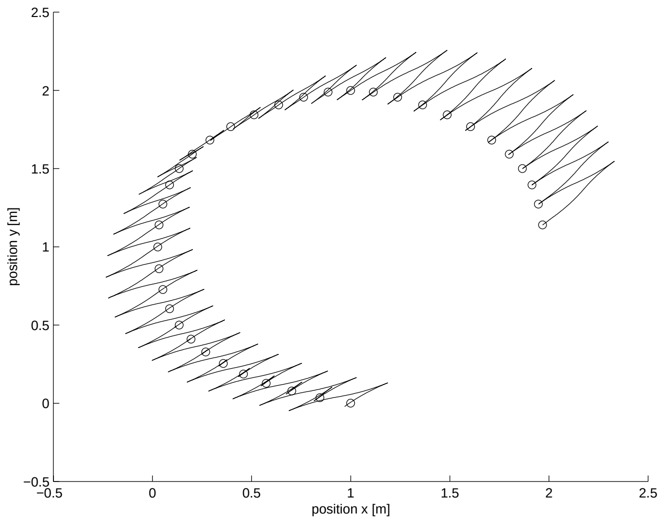

Figure 3.3: Basic stratified motion planning for six-legged robots. Marks "o" denote representative points (from a circle) to be reached. Desired orientation is constant at every representative points and differs from the tangent of trajectory.

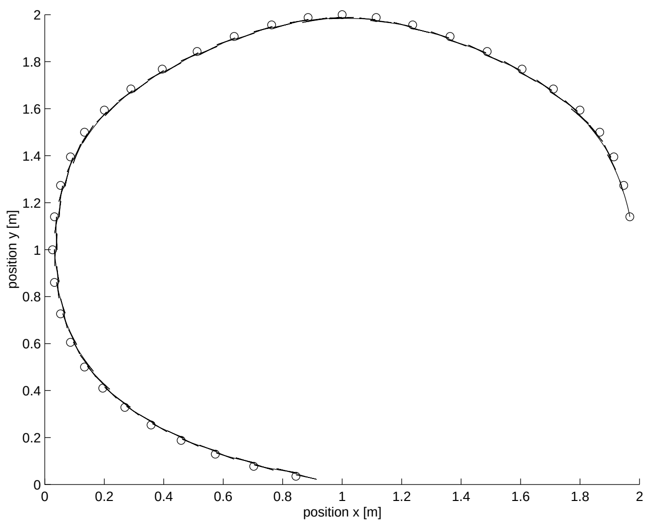

Figure 3.4: Basic stratified motion planning for six-legged robots. Marks "o" denote representative points (from a circle) to be reached. Desired orientation is constant at every representative points and coincides with the tangent of trajectory.

how to solve convergency for stratified systems.

Another problem this chapter focused on was the time-scaling. It aims to assure the freedom to relate desired time points to representative points. Proposition 3 solved the problem for smooth motion planning using sequence of flows, and Proposition 4 extended its result to stratified case. Algorithm 6 displayed how to combine algorithms together in order to realize time-scaling equipped with improved convergence property.

The results were illustrated on the motion planning problem of six-legged robot. This example revealed the main drawback of basic stratified motion planning, namely, it meets difficulties if obstacles can be found in the environment. Chapters 5 and Chapters 6 are devoted to overcome this problem.

Chapter 4

Stratified Manipulation Planning Algorithms

This chapter poses manipulation planning problem using fingertip relocations from stratified viewpoint. The idea of using stratified motion planning for manipulation was reported first in Goodwine (1998b) and Goodwine (1999). Methods investigated here carries fundamentally stratified motion planning strategies but they approach the problem in partly different way making techniques suitable for some extensions and improvements. Since manipulation of this kind concerns stratified motion planning, the results of Chapter 2 and Chapter 3 are intensively exploited.

Section 4.1 settles the manipulation setup the manipulation planning algorithms applied to. It presents the kinematic of object-hand arrangement and enlightens its stratified nature. Stratified manipulation planning, however, encounters some issues when stratified motion planning is to be applied.

In order to triumph over difficulties, Section 4.2 proposes three techniques, each of them employs specific, so called fitted parametrization. Algorithms devised here are built on the same foundation as stratified motion planning, however they follow different philosophies making profit from some particular features of manipulation structure.

Simulation results due to algorithms are discussed in Section 4.3 concerning the manipulation of egg-shaped object with robotic hand equipped by four fingers. Each finger is of three degrees of freedom.

Reader finds the results of this chapter in Harmati, Lantos, and Payandeh (2002f), Harmati, Lantos, and Payandeh (2000b), Harmati, Lantos, and Payandeh (2001), Harmati, Lantos, and Payandeh (2002g), Harmati and Lantos (2002).

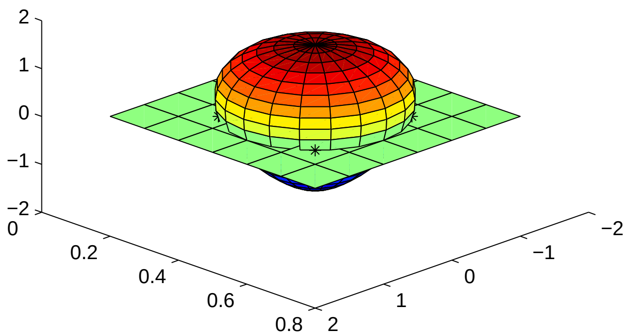

Figure 4.1: The manipulated object with the 4 initial contact points (denoted by $*$) in the plane $z = 0$. The task is a rotation around $[1\ 0\ 1]^T$. (Axis z is directed upwards)

4.1 Manipulation Concept with Fingertip Relocations: An Introductory Example

The results of this chapter will be demonstrated by using an example. This section introduces this manipulation system and presents some notations used in the following. The algorithms suggested in the next section are introduced via an egg-shaped object manipulation. The x, y and z coordinates of the surface points (Goodwine (1999)) shown in Figure 4.1 are parameterized by the equation

$$c(u,v) = \begin{pmatrix} (1+\frac{u}{\pi})\cos u \cos v \\ (1+\frac{u}{\pi})\cos u \sin v \\ \frac{3}{2}\sin u \end{pmatrix}, \qquad \begin{array}{c} u \in (\frac{-\pi}{2}, \frac{\pi}{2}) \\ v \in (-\pi, \pi) \end{array} \tag{4.1}$$

where u and v are the parameters of the surface.

Consider a robot hand equipped with four fingers where each finger has three degrees of freedom. The relation between a finger and the object is illustrated in the Figure 4.2. Frame K_p denotes the palm frame and it is the inertial frame in the manipulation system. The object frame K_o is fixed to the object. Without loss of generality, one may assume that the origin of the object frame K_o coincides with the origin of the palm frame K_p. Let the vector ω_o denote the angular velocity of the object frame relative to the palm frame, as seen from the palm frame. Similarly, let v_o denote the linear velocity of the object frame relative to the palm frame, as seen from the palm frame. The frames K_{f_i}, $i = 1 \ldots 4$ are attached to the fingertips. Let the homogeneous transformation between the palm frame and the base frames of fingers be given by

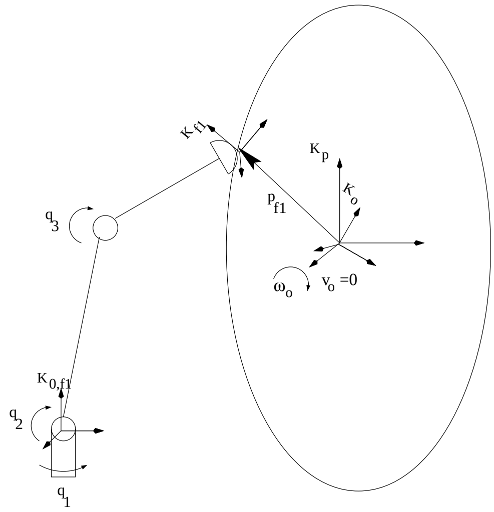

Figure 4.2: The connection between a finger and the object.

$$T_{p0_{f1}} = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p0_{f2}} = \begin{bmatrix} 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p0_{f3}} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_{p0_{f4}} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The fingers divide the workspace into quadrants. This arrangement is beneficial because the four fingers work in different quadrants of the workspace. The initial contact points are determined by the intersection of the object and the lines $x = \pm y$ in the plane $z = 0$ as shown in Figure 4.1 and Figure 4.3. Considering local manipulation planning where the object and fingertip motion are sufficiently small, the quadrants ensure separated workspaces avoiding collision of the fingertips. In the case of global manipulation planning, the collision avoidance problem requires additional strategies. They will be discussed in Subsection 4.2.1 and 4.2.2 in more detail. Their philosophy consists of inserting extra finger relocation back onto the lines $x = \pm y$ in the plane $z = 0$ if the fingertip is about to leave its own workspace (stratified manipulation) or one may carry out this finger relocation after a particular manipulation phase (see semi-stratified manipulation).

In this discussion, piecewise constant contact points are supposed where contact model does not allow the fingertip to slide or roll on the object but one can carry out a fingertip relocation. In other words, during the object motion, the fingertips in contact with the object keep their positions in relative to the object frame K_o. Contact points with the above properties can be realized by the contact point with friction model (Salisbury and Mason (1985)). Using this assumption, one describes the kinematics in the palm frame. Since the methods in this thesis concern the kinematic problem (Lantos (1997)), let J_{pf_i} denote the Jacobian matrix of the ith finger. The fingertip velocity of the ith finger is described by reduced Jacobian matrix containing only those rows from J_{pf_i} which are related to the linear velocities. For the ith finger, the reduced Jacobian matrix that establishes a connection between the joint variables and the fingertip velocity (the linear velocity of the origin

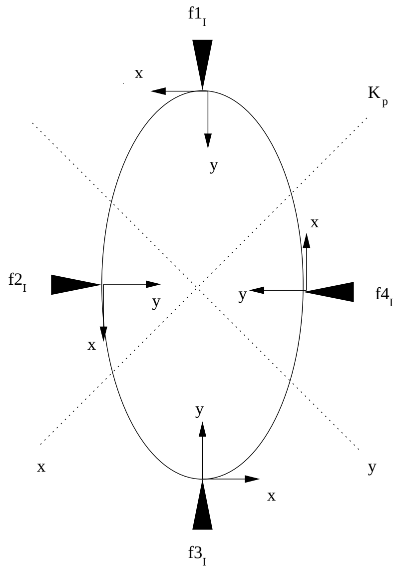

Figure 4.3: The quadrants of the workspace.

of K_{f_i}) in the palm frame K_p is given by

$$v_{pf_i} = J^v_{pf_i}\dot{\bar{q}}_i \tag{4.2}$$

where $J^v_{pf_i}$ has 3 rows and 3 columns. For example, a robotic hand equipped with the above homogeneous transformations claims that the Jacobian matrix of the first finger is given by

$$\begin{aligned}
J^v_{pf_1}(1,1) &= \sin(\bar{q}_1(1))(\sin(\bar{q}_1(2) - \bar{q}_1(3)) \\
&+ \sin(\bar{q}_1(2))) \\
J^v_{pf_1}(1,2) &= -\cos(\bar{q}_1(1))\cos(\bar{q}_1(2) \\
&- \bar{q}_1(3))\,(1 + \cos(\bar{q}_1(3))) \\
&+ \cos(\bar{q}_1(1))\sin(\bar{q}_1(2) - \bar{q}_1(3))\sin(\bar{q}_1(3)) \\
J^v_{pf_1}(1,3) &= -\cos(\bar{q}_1(1))\cos(\bar{q}_1(2) - \bar{q}_1(3)) \\
J^v_{pf_1}(2,1) &= -\cos(\bar{q}_1(1))(\sin(\bar{q}_1(2) - \bar{q}_1(3)) \\
&+ \sin(\bar{q}_1(2))) \\
J^v_{pf_1}(2,2) &= -\sin(\bar{q}_1(1))\cos(\bar{q}_1(2) \\
&- \bar{q}_1(3))\,(1 + \cos(\bar{q}_1(3))) \\
&+ \sin(\bar{q}_1(1))\sin(\bar{q}_1(2) - \bar{q}_1(3))\sin(\bar{q}_1(3)) \\
J^v_{pf_1}(2,3) &= -\sin(\bar{q}_1(1))\cos(\bar{q}_1(2) - \bar{q}_1(3)) \\
J^v_{pf_1}(3,1) &= 0 \\
J^v_{pf_1}(3,2) &= -\sin(\bar{q}_1(2) - \bar{q}_1(3))\,(1 + \cos(\bar{q}_1(3))) \\
&- \cos(\bar{q}_1(2) - \bar{q}_1(3))\sin(\bar{q}_1(3))
\end{aligned}$$

$$J^v_{pf_1}(3,3) \quad = \quad -\sin(\bar{q}_1(2) - \bar{q}_1(3)) \tag{4.3}$$

where $\bar{q}_1(j)$ is the jth joint variable of the 1st finger etc. Taking into account all the contact points it yields

$$\begin{aligned} v_{pf} &= \begin{pmatrix} v_{pf_1} \\ \vdots \\ v_{pf_4} \end{pmatrix} = \underbrace{\begin{bmatrix} J^v_{pf_1} & & \\ & \ddots & \\ & & J^v_{pf_4} \end{bmatrix}}_{J^v_{pf}} \begin{pmatrix} \dot{\bar{q}}_1 \\ \vdots \\ \dot{\bar{q}}_4 \end{pmatrix} \\ &= J^v_{pf}\dot{q} \end{aligned} \tag{4.4}$$

The discussion of singularities is not a subject of this investigation, hence one may insert the methods bellow into a global manipulation planning where singularities, collision avoidance and force closure are investigated in a higher level (Vass, Payandeh, and Lantos (1999a)). On the other hand, if v_o denotes the linear velocity and ω_o denotes the angular velocity of the object frame relative to the palm frame (as seen in the palm frame), then the fingertip velocities can be determined in an alternative way, namely,

$$\begin{pmatrix} v_{pf_i} \\ \omega_{pf_i} \end{pmatrix} = \begin{bmatrix} I & [-p_{pf_i}\times] \\ 0 & I \end{bmatrix} \begin{pmatrix} v_o \\ \omega_o \end{pmatrix} \tag{4.5}$$

Introducing the notation

$$J_{po_i}(p_{pf_i}) = \begin{bmatrix} I & [-p_{pf_i}\times] \\ 0 & I \end{bmatrix} \tag{4.6}$$

and assuming that all the fingers are in contact, one extends (4.5) to all linear and angular velocities as

$$\begin{pmatrix} v_{pf_1} \\ \omega_{pf_1} \\ \vdots \\ v_{pf_4} \\ \omega_{pf_4} \end{pmatrix} = \underbrace{\begin{bmatrix} J_{po_1}(p_{pf_1}) \\ \vdots \\ J_{po_4}(p_{pf_4}) \end{bmatrix}}_{J_{po}(p_{pf})} \begin{pmatrix} v_o \\ \omega_o \end{pmatrix} \tag{4.7}$$

where $p_{pf} = (p^T_{pf_1}, \ldots, p^T_{pf_4})^T$. For clarity, $J_{po_i}(p_{pf_i})$ is written as J_{po_i} and $J_{po}(p_{pf})$ is written as J_{po}, respectively. To express the fingertip velocities, one should eliminate the rows belonging to angular velocities ω_{pf_i} from J_{po_i}, $i = 1 \ldots 4$ from (4.7). It implies a reduced matrix J^v_{po} instead of J_{po} resulting fingertip velocities expressed in the palm frame as

$$\begin{pmatrix} v_{pf_1} \\ \vdots \\ v_{pf_4} \end{pmatrix} = \underbrace{\begin{bmatrix} I & [-p_{pf_1}\times] \\ & \vdots \\ I & [-p_{pf_4}\times] \end{bmatrix}}_{J^v_{po}} \begin{pmatrix} v_o \\ \omega_o \end{pmatrix} \tag{4.8}$$

From (4.4) and (4.8) one deduces

$$\begin{pmatrix} v_{pf_1} \\ \vdots \\ v_{pf_4} \end{pmatrix} = \underbrace{\begin{bmatrix} I & [-p_{pf_1}\times] \\ & \vdots \\ I & [-p_{pf_4}\times] \end{bmatrix}}_{J^v_{po}} \begin{pmatrix} v_o \\ \omega_o \end{pmatrix}$$

$$= \underbrace{\begin{bmatrix} J^v_{pf_1} & & \\ & \ddots & \\ & & J^v_{pf_4} \end{bmatrix}}_{J^v_{pf}} \begin{pmatrix} \dot{\bar{q}}_1 \\ \vdots \\ \dot{\bar{q}}_4 \end{pmatrix} \tag{4.9}$$

From here, the relation between joint variables and object motion can be obtained as:

$$\begin{pmatrix} \dot{\bar{q}}_1 \\ \vdots \\ \dot{\bar{q}}_4 \end{pmatrix} = \left(J^v_{pf}\right)^{-1} J^v_{po} \begin{pmatrix} v_o \\ \omega_o \end{pmatrix}. \tag{4.10}$$

Expression (4.10) is used below in this section and Section 4.2.1).

Now, it is to be shown that the manipulation system discussed above may be considered as a stratified system. Another goal is to describe the manipulation task in terms of stratifaction providing more exact equations for strata that reflect this nature. To study the problem in such a way, one should define the subsystems in the strata. Example 7 showed that the notion of finger gaiting fits well to stratified discussion. Pursuing this idea, strata can be defined with different subsystems where each subsystem is subjected to a set of constraints depending on the fingers being in contact with the object. For the sake of simpler formulation, a treatment in palm frame is adopted.

If all the fingers are in fixed contact with the object while the object is moving, the subsystem

$$\Sigma_{f,S_{1234}}: \quad \begin{pmatrix} v_o \\ \omega_o \\ \dot{p}_{pf_1} \\ \vdots \\ \dot{p}_{pf_4} \end{pmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \\ I & [-p_{pf_1}\times] \\ \vdots & \vdots \\ I & [-p_{pf_4}\times] \end{bmatrix} \begin{pmatrix} v_o^d \\ \omega_o^d \end{pmatrix} \tag{4.11}$$

realizes a possible manipulation phase. Here, $p_{pf_1}, \ldots, p_{pf_4}$ represent the fingertip positions. Indeed, $\Sigma_{f,S_{1234}}$ defines the system in the bottom stratum S_{1234}. Notations v_o^d and ω_o^d realize the desired linear and angular velocities of the object, however, they are also the (fictitious) inputs of $\Sigma_{f,S_{1234}}$. The physical inputs of the kinematic model are the derivatives of the joint variables $\bar{q}_i$. The transformation (4.10) makes a conversion between the fictitious and real input:

$$\begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_4 \end{pmatrix} = \left(J^v_{pf}\right)^{-1} J^v_{po} \begin{pmatrix} v^d_o \\ \omega^d_o \end{pmatrix} \tag{4.12}$$

The use of fictitious inputs has the benefit of leaving (4.11) in a form $\dot{x} = F(x)u$ instead of the more general $\dot{x} = F(x,u)$. Recall, that stratified motion planning in Section 2.2 was developed for a kinematic system given in the form $\dot{x} = F(x)u$.

Consider now a subsystem in a higher stratum. For instance, if all the fingers except Finger 1 are in contact with the object, it assigns a stratum S_{234}. The equations of motion on this stratum yields

$\Sigma_{f,S_{234}}$:

$$\begin{pmatrix} v_o \\ \omega_o \\ \dot{p}_{pf_1} \\ \vdots \\ \dot{p}_{pf_4} \end{pmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \\ I & [-p_{pf_2}\times] & 0 \\ I & [-p_{pf_3}\times] & 0 \\ I & [-p_{pf_4}\times] & 0 \end{bmatrix} \begin{pmatrix} v^d_o \\ \omega^d_o \\ \dot{p}^d_{pf_1} \end{pmatrix} \tag{4.13}$$

where $\dot{p}^d_{pf_1}$ is the desired (unconstrained) fingertip velocity of the 1st finger. One may derive the connection between the fictitious input $((v^d_o)^T,\ (\omega^d_o)^T,\ (\dot{p}^d_{pf1})^T)^T$ and joint variables similarly to (4.12). The only difference is that the first finger is not subjected to a constraint now, hence

$$\begin{pmatrix} \dot{q}_1 \\ \hline \dot{q}_2 \\ \vdots \\ \dot{q}_4 \end{pmatrix} = \left[\begin{array}{c|c} 0 & {J^v_{pf_1}}^{-1} \\ \hline {J^v_{pf_{\{234\}}}}^{-1} J^v_{po_{\{234\}}} & 0 \end{array}\right] \begin{pmatrix} v^d_o \\ \omega^d_o \\ \hline \dot{p}^d_{pf_1} \end{pmatrix} \tag{4.14}$$

where

$$J^v_{po_{\{234\}}} = \begin{bmatrix} I & [-p_{pf_2}\times] \\ & \vdots \\ I & [-p_{pf_4}\times] \end{bmatrix} \tag{4.15}$$

and

$$J^v_{pf_{\{234\}}} = \begin{bmatrix} J^v_{pf_2} & & \\ & \ddots & \\ & & J^v_{pf_4} \end{bmatrix} \tag{4.16}$$

The equations of motion and the conversions due to other higher strata have similar forms i.e. for instance,

$\Sigma_{f,S_{134}}$:

$$\begin{pmatrix} v_o \\ \omega_o \\ \dot{p}_{pf_1} \\ \dot{p}_{pf_2} \\ \dot{p}_{pf31} \\ \dot{p}_{pf_4} \end{pmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ I & [-p_{pf_1}\times] & 0 \\ 0 & 0 & I \\ I & [-p_{pf_3}\times] & 0 \\ I & [-p_{pf_4}\times] & 0 \end{bmatrix} \begin{pmatrix} v_o^d \\ \omega_o^d \\ \dot{p}_{pf_2}^d \end{pmatrix} \tag{4.17}$$

and so on. The stratified system in 4.17 will be denoted by Σ_f. Recall now, that one may accomplish the stratified motion planning algorithmif it satisfies the conditions of Proposition 1 and Corollary 1. As a contribution, it is shown in the following that this condition is not satisfied in Σ_f.

Proposition 5. The system Σ_f does not make possible switching between any two strata.
Proof: Consider the vector fields of the systems defined by (4.11) and (4.13). Define two groups of the vector fields by the last 3 vector fields of $\Sigma_{f,S_{1234}}$ and $\Sigma_{f,S_{234}}$

$$F_{on} = [f_1\ f_2\ f_3] = \begin{bmatrix} 0 \\ I \\ [-p_{pf_1}\times] \\ \vdots \\ [-p_{pf_4}\times] \end{bmatrix}, \qquad G_{off} = [g_1\ g_2\ g_3] = \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.18}$$

(the size of each block is 3 by 3). By the definition of the moving on and moving off vector fields, the vector fields $f_1, \ldots, f_3$ are moving on vector fields, the vector fields $g_1, \ldots, g_3$ are moving off vector fields. One can find moving on vector fields from F_{on} to each moving off vector fields from G_{off} such that their Lie brackets are not zero. For example:

$$\begin{aligned} [f_2, g_1] &= [f_3, g_2] = [f_1, g_3] \\ &= \begin{pmatrix} 0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0 \end{pmatrix}^T \neq 0 \end{aligned}$$

Furthermore, these equations are still valid if one defines G_{off} as the last 3 vector fields of any higher strata. It means that based on Corollary 1, the stratified motion planning algorithm cannot switch between arbitrary two strata in the system Σ_f.

The switching problem arising in Σ_f inspired to develop a new approach that fits well to stratified manipulation planning. This method is devised in Subsection 4.2.1. Additional motivation to elaborate a new technique is that the actual stratum identification requires to check the contacts in Σ_f due to a finger. It needs tedious numerical calculation because the distances between object and fingertips do not appear directly in the state vector.

4.2 Fitted Manipulation Planning Methods using Fingertip Relocations

This section proposes three algorithms that solve manipulation planning by means of finger relocations. Each of them is related to stratified motion planning, however they use specific parametrization that provides advantages and makes treatments more comfortable in comparison to earlier approach Goodwine (1999). Subsection 4.2.1 establishes the foundation of fitted stratified manipulation planning that makes switching between strata possible. Subsection 4.2.2 modifies this algorithm in order to reduce the quantity of symbolic computation. Subsection 4.2.3 enhances the robustness of fitted stratified manipulation by ensuring limited trajectory tracking.

4.2.1 Fitted Stratified Manipulation Planning

This subsection concerns one of the two other main contributions of this chapter. A new approach of stratified manipulation is proposed which is an extension of the earlier method Goodwine (1999). More precisely, our primarily goal is to provide a description and a manipulation technique for the system outlined in Section 4.1 that permits the use of stratified motion planning in accordance with Corollary 1.

The idea is that fingertip position is described by a vector $p^o_{pf_i} = (u_i \; v_i \; z_i)^T$ where u_i and v_i describe the projection of the fingertip position onto the object along the normal vector of the surface (Harmati, Lantos, and Payandeh (2001)). Variables u_i and v_i are given as explicit parameters of the object surface in Gaussian frame. Variable z_i denotes the distance between the ith fingertip and the object (Figure 4.4). The superscript "o" refers to the object frame whose coordinates are defined by explicit parameters of the object. This description clearly defines moving on and moving off vector fields in the equations of motion. Such a description fits well to stratified motion planning, hence motion planning problem is resolved using these representation. However, this representation is a fictitious (called fitted) system in the sense that real physical parameters (fingertip positions and joint variables) should be obtained via transformation. The method bellow includes a stratified motion planning algorithm on a fictitious system, and transformations between fictitious and real physical systems. Using the convention in notation above, the stratified manipulation system in Section 4.1 implies the following equations of motion on the fitted system.

In bottom stratum, all the fingers are in contact with object i.e. the degree of freedom comes only from the object motion:

$$\Sigma_{fitted,S_{1234}} \; : \quad \begin{pmatrix} v_o \\ \omega_o \end{pmatrix} = I_6 \begin{pmatrix} v_o^d \\ \omega_o^d \end{pmatrix} \tag{4.19}$$

$\dot{v}_i = \dot{u}_i = z_i = 0, \, i = 1, \ldots, 4$

where I_6 is the identity matrix with dimension 6. Similarly to (4.11), v_o^d and ω_o^d realize the desired linear

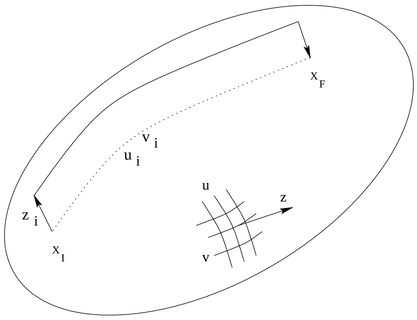

Figure 4.4: Trajectory by finger relocation and its orthogonal projection onto the object

and angular velocities of the object and they are also the (fictitious) inputs of $\Sigma_{f,S1234}$. Recall, that fingertips are given in object frame now with different coordinates, hence transformation (4.10) between fictitious and physical inputs (joint variables) should be modified. Let

$$p_{pf_i} = \text{object2palm}(Egg(u,v), p^o_{pf_i}) \tag{4.20}$$

be the function that transforms object coordinates $p^o_{pf_i}$ to palm coordinates p_{pf_i} if the object is of egg shape. Then, physical inputs represented by joint variables are obtained by

$$\begin{pmatrix} \dot{\bar{q}}_1 \\ \vdots \\ \dot{\bar{q}}_4 \end{pmatrix} = \left(J^v_{pf}\right)^{-1} J^v_{po}(\text{object2palm}(Egg(u,v), p^o_{pf})) \begin{pmatrix} v^d_o \\ \omega^d_o \end{pmatrix} \tag{4.21}$$

where $p^o_{pf} = ((p^o_{pf_1})^T, \ldots, (p^o_{pf_4})^T)^T$. As it can be seen, the above equation emphasizes that J^v_{po} depends on p^o_{pf} and not on p_{pf} as in (4.6).

Consider now the higher strata $S_{j_1 j_2 j_3}$ where multiindex $j_1 j_2 j_3$ assigns the fingers having contact with the object. The equations of motion on a higher stratum are given by

$\Sigma_{fitted,S_{j_1 j_2 j_3}}$:

$$\begin{pmatrix} v_o \\ \omega_o \\ \dot{u}_i \\ \dot{v}_i \\ \dot{z}_i \end{pmatrix} = \begin{bmatrix} I_6 & 0 \\ 0 & I_3 \end{bmatrix} \begin{pmatrix} v^d_o \\ \omega^d_o \\ \dot{u}^d_i \\ \dot{v}^d_i \\ \dot{z}^d_i \end{pmatrix} \tag{4.22}$$

$\dot{v}_{j_k} = \dot{u}_{j_k} = z_{j_k} = 0, \quad j_1 j_2 j_3 \in I_3$ (multiindex), $j_k \neq i$, $k = 1 \ldots 3$ for any $i = 1, \ldots, 4$

The transformation between inputs of (4.22) and joint variables is similar to (4.14). For instance, it leads on

S_{234} to the transformation:

$$\begin{pmatrix} \dot{\bar{q}}_1 \\ \hline \dot{\bar{q}}_2 \\ \vdots \\ \dot{\bar{q}}_4 \end{pmatrix} = \left[\begin{array}{c|c} 0 & {J^v_{pf_1}}^{-1} \\ \hline {J^v_{pf_{\{234\}}}}^{-1} J^v_{po_{\{234\}}} & 0 \end{array} \right] \begin{pmatrix} v_o^d \\ \omega_o^d \\ \hline \dot{p}_{pf1}^{o,d} \end{pmatrix} \tag{4.23}$$

where $\dot{p}_{pf1}^{o,d}$ is the desired velocity of the 1st fingertip described by $(\dot{u}_1, \dot{v}_1, \dot{z}_1)^T$. Note, that $J^v_{po_{\{234\}}}(p_{pf_{234}}) = J^v_{po_{\{234\}}}(\text{object2palm}(Egg(u,v), p^o_{pf_{234}}))$ in accordance with (4.20).

Again, the superscript d in inputs refers implicitly to the fact that the derivatives of the state can be influenced directly by the desired values of the velocities as the inputs of the fictitious system Σ_{fitted}. Variables u_i, v_i in (4.22) determine the projection of the unconstrained fingertip motion on the object surface. The next proposition highlights the main benefit of the suggested approach.

Proposition 6. The fitted stratified model Σ_{fitted} makes the switching between two strata possible.
Proof: In order to justify the statement, one has to show that every Lie bracket between a moving on and a moving off vector field is zero i.e. they commute. At first, it is to be shown that vector fields related to variables z_i^d (e.g. last column in the matrix in (4.22)) are moving off vector fields and all the other vector fields are moving on vector fields. For this, one can see that inputs z_i^d act only on the state variables z_i. However, the actual equations of motion (and the actual stratum) depend on the (not necessary only one) i for which $z_i = 0$. In other words, z_i^d determines the current stratum and at the same time, does not influence the state vector in the bottom stratum. In accordance with the definition of the moving off vector fields (see Definition 17), the vector fields related to the inputs z_i^d are moving off vector fields. Furthermore, let e_i denote the unit vector

$$e_i = (0, \dots, 0, \underbrace{1}_{i^{th}\ pos.}, 0, \dots 0)^T \in \mathbb{R}^{18}. \tag{4.24}$$

Each vector field of system Σ_{fitted} in the whole configuration space is equal to e_i. It is an elementary result from differential geometry that Lie brackets $[e_i, e_j] = 0$. It holds for arbitrary two vector fields and it holds also for Lie brackets defined between any moving on and any moving off vector fields. Based on Corollary 1, one can switch between arbitrary two strata in the system Σ_{fitted}.

Corollary 3. Stratified control algorithm can be applied to Σ_{fitted} by creating the bottom stratified (extended) system.

See now, how the fitted stratified manipulation planning works on Σ_{fitted}. The final goal in manipulation planning proposed is to obtain the joint variables of fingers that steer the object into a desired position and orientation. At the final stage, fingertips should reach desired contact points on the surface, as well. The stratified manipulation planning involves a stratified motion planning problem in the configuration space. Equations of strata are given by (4.19) and (4.22). These subsystems define a bottom stratified fitted system that satisfies the condition of Corollary 1 and forms

$$\begin{pmatrix} v_o \\ \omega_o \\ \dot{u}_1 \\ \dot{v}_1 \\ \vdots \\ \dot{u}_4 \\ \dot{v}_4 \end{pmatrix} = \begin{bmatrix} I_6 & 0 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 & 0 \\ 0 & 0 & I_2 & 0 & 0 \\ 0 & 0 & 0 & I_2 & 0 \\ 0 & 0 & 0 & 0 & I_2 \end{bmatrix} \begin{pmatrix} v_o^d \\ \omega_o^d \\ \dot{u}_1^d \\ \dot{v}_1^d \\ \vdots \\ \dot{u}_4^d \\ \dot{v}_4^d \end{pmatrix} \tag{4.25}$$

It is known from Section 2.2 that this bottom stratified (fitted) system is the foundation of basic stratified motion planning used in our manipulation planning. In fact, the (basic) stratified motion planning on Σ_{fitted} can be reduced into a smooth motion planning (see Subsection 2.1.2) on the bottom stratified fitted system (4.25). The vector fields of (4.25) are the moving on vector fields of Σ_{fitted}. Since the system is stratified, it is required to insert moving off vector fields where two flows of moving on vector fields from different strata meet at a common point. Smooth motion planning on (4.25) makes sense since the desired object position, orientations and fingertip positions are available. The only remaining problem is that the inputs of Σ_{fitted} are fictitious and not real. It means that one can attain the trajectory of stratified motion planning only if the fictitious inputs can be substituted with the joint variables. As mentioned before, this is possible by using the transformations (4.21), (4.23) and so on.

The main steps of the method are illustrated in Figure 4.5 and summarized as follows.

Algorithm 7. [Basic fitted stratified manipulation]

Step 1. Planning the desired motion of the object and the fingertips. The reference fingertips needs to be available in the object frame.

Step 2. Create the fitted system Σ_{fitted} (see (4.19) and (4.22)).

Step 3. Create the bottom stratified fitted system (4.25).

Step 4. Stratified motion planning (Algorithm 6) on fitted system. Performing the finger relocations according to the insertion of moving off vector fields.

Step 5. Transform the motion trajectory from the state space of the fitted system into the state space of the real system. One may use the geometric and kinematic transformations such as (4.21), (4.23). As a result, one obtains the joint variables (and their derivatives) due to desired manipulation.

Observe now some features generated by this approach. The main complication appeared in earlier stratified motion planning algorithm Goodwine (1999) when Philip Hall coordinates were computed (See subsection 2.1.2). It proceeds via evaluation of recursive integrations. If the vector fields of bottom stratified system has symbolically complex vector fields, the integrands become complicated. However, this is not the case for fitted stratified system because the vector fields do not contain symbolic variables. It makes computations very easy, almost entirely numerical.

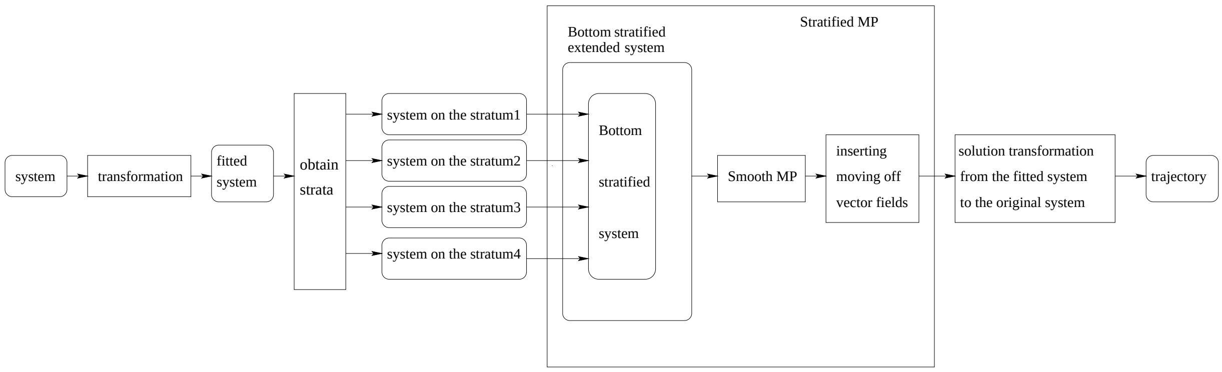

Figure 4.5: The main steps of the fitted stratified manipulation planning.

As a consequence of simple vector fields, one can easily show the property of (stratified) controllability on the proposed fitted system with a considerably fewer computations than in the earlier stratified approaches since all kind of computations with constant vector fields are elementary in differential geometry.

An additional advantage of the method in comparison with Goodwine (1999) that one may directly interpret the resulting trajectory of the system since the fictitious inputs in (4.19), (4.22) etc. are strongly related to the object parameters. One can also see that the creation of *extended* bottom stratified fitted system is unnecessary because the vector fields of bottom stratified fitted system span the entire tangent space in a given configuration. In our example, the dimension of system (4.25) equals to $3 + 3 + 4 \times 2 = 14$. Both linear and angular velocity of the object take 3 coordinates. Additionally, every fingertip is described by 3 parameters, of which 2 determine the projection of the fingertip onto the object surface expressed by the exact parameters of the object. The third parameter does not take place in bottom stratified fitted system but it is fixed to the moving off vector fields that lift the finger on and off the surface. Observe, that every input in (4.25) affects only one state variable i.e. the bottom stratified fitted system is decoupled. It implies that configuration space can be divided into subspaces where the strata assign disjoint subspaces (see the vector fields of (4.25). It poses the possibility of a stratified manipulation/motion planning where the trajectory may evolve locally on a submanifold whose dimension is greater than one. (This property is to be exploited in Subsection 4.2.3) This means that the projection of fingertips may follow a special trajectory on the object surface. Recall, that basic stratified motion planning relies on a smooth motion planning where the system trajectory evolves along sequence of flows i.e. it is locally restricted to one dimensional submanifolds. It means that arbitrary fingertip trajectory cannot be prescribed only a final state can be reached.

The technique introduced also possesses a main drawback. Namely, one needs to execute extra transformations between fictitious and real inputs. It requires to evaluate nonlinear transformations like (4.21) and (4.23) at each configuration i.e. joint variables are not obtained directly but they are derived.

Additionally, the method expects some reference data (fingertip positions) not in the palm frame but in the object (body) frame.

One more restriction appears when stratified manipulation is employed. Namely, the workspace were divided into four quadrants and it is assumed that all the fingers work in their own quadrant assuring the collision avoidance. Unfortunately, stratified manipulation does not necessarily keep fingers directly in their own workspace for arbitrary manipulation task. If the reference fingertips are prescribed in their own workspaces, fitted stratified manipulation implies a dextrous manipulation. It can be done by choosing a desired final point close enough to the initial configuration. However, the most important requirement is that the object should reach the desired position and orientation, hence the fingertip positions play relatively secondary role. In this special case, the problem should be overcome in the phase when the reference trajectory is generated. More exactly, if it turns out that a finger is about to leave its permitted workspace (quadrant) during the desired object motion, then one should insert an artifically generated finger relocation that pulls back the finger into its region, for example onto the lines $x = \pm y$ in the plane $z = 0$. Of course, it means that reference fingertip positions must be modified. The semi-stratified manipulation planning in the next subsection is devoted to handling this problem.

4.2.2 Fitted Semi-Stratified Manipulation Planning

In this subsection, a decomposed manipulation concept is proposed for object with smooth surfaces that combines fitted stratified motion planning with unconstrained motion planning. In this context, unconstrained motion means that a finger can move in the free space between two points independently of the object. The motivation to use unconstrained motion planning beside stratified manipulation may be useful from many reasons. One of them is to give a greater degree of freedom for finger relocations in manipulation planning. Second, it may be desired to reduce computations appearing for instance in computation of Philip Hall basis. Additionally, it is beneficial to keep the fingers in their own workspaces independent of the object orientation.

If one considers the fitted stratified manipulation in Subsection 4.2.1, then a final configuration can be obtained as a sequence of 14 connected flows

$$\begin{aligned}
x_F &= \Phi(t_{14}, v_{pf_4}^{o,d}(2)) \circ \Phi(t_{13}, v_{pf_4}^{o,d}(1)) \circ \cdots \\
&\circ \Phi(t_8, v_{pf_1}^{o,d}(2)) \circ \Phi(t_7, v_{pf_1}^{o,d}(1)) \\
&\circ \Phi(t_6, \omega_o^{o,d}(3)) \circ \cdots \circ \Phi(t_4, \omega_o^{o,d}(1)) \\
&\circ \Phi(t_3, v_o^{o,d}(3)) \circ \cdots \circ \Phi(t_1, v_o^{o,d}(1))(x_I)
\end{aligned} \tag{4.26}$$

where the second argument of flow Φ denotes the active input and the first one defines the length of its active time interval. In addition, $v_{pf1}^{o,d} \equiv \dot{p}_{pf1}^{o,d}$, $v_{pf1}^{o,d}(1) = \dot{u}_1^d$, $v_{pf1}^{o,d}(2) = \dot{v}_1^d$ and so on. The first 6 flows (i.e. the last 6 in writing order) manipulate the object and every 3 flows following it belong to fingers. Note, that active input uniquely assigns a moving on vector field from Σ_{fitted}. If two neighboring flows are defined in different strata, then insertion of appropriate moving off vector field is required. It is done by activating inputs $v_{pfi}^{o,d}(3)$. It may already occur in the object manipulation phase realized by flows $\Phi(t_6, \omega_o^{o,d}(3)), \ldots, \Phi(t_1, v_o^{o,d}(1))$ that the fingertip positions exceed their allocated workspace.

To avoid this phenomenon, semi-stratified manipulation planning is pursued through several steps. The main steps are outlined in the Figure 4.6.

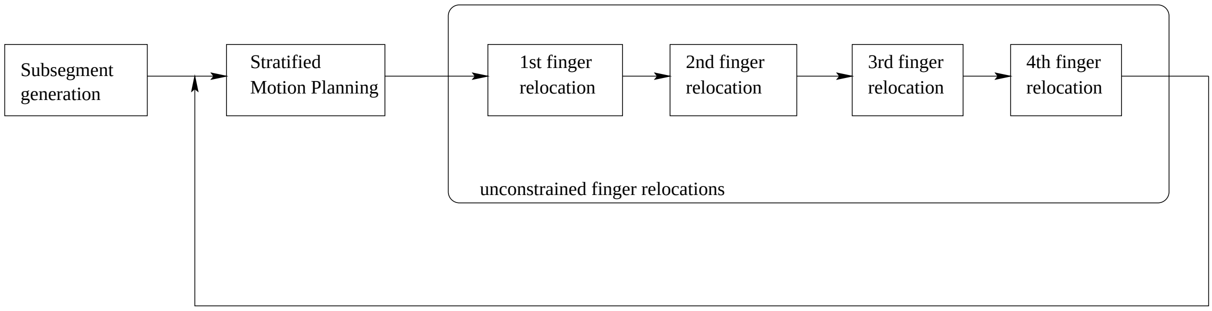

Figure 4.6: The main steps of the semi-stratified manipulation planning

In order to keep the fingertips in their own quadrants assuring their collision avoidance, the reference fingertip positions should be chosen in suitable way. The *subsegment generation* algorithm concerns this issue, as follows. It is assumed that the four initial fingertips on the object surface lie on the lines $x = y$ and $x = -y$ in the plane $z = 0$.

Follow a manipulation phase now. At first, one fixes contact points to the object and moves the object along the reference object trajectory. Before one of the fingertips is about to leave its quadrant (workspace), one records the object and fingertips configurations. This configuration will be the desired final configuration for the point to point fitted stratified manipulation. This configuration is reached by a sequence of flows (4.26).

At this point, the algorithms inserts four *unconstrained finger relocations* that "pull back" the fingertips into the plane $z = 0$ where $x = y$ and $x = -y$ (see Figure 4.1). The fingers are relocated in order i.e. only one of them moves back into the contact $z = 0$, $x = y$ or $x = -y$ at a time, the other three remain on the surface. After this, the following finger is relocated onto another contact point and the other three ones remain in contact with the object. While the fingertip is being relocated, it breaks the contact, moves in the free space and establishes a new contact in the above mentioned location. It can be shown Goodwine (1999) that this kind of motion of the fingertips are able to ensure force closure stability. As a matter of fact, the unconstrained finger relocations are new features in the fitted semi-stratified manipulation since they did not appear in fitted stratified manipulation. The unconstrained finger relocations allow to steer the free fingertip using $J^v_{pf_i}$. In spite of fitted stratified motion planning, it is allowed to be more than one active input, so that the relocated fingertip has 3 degrees of freedom in the space whereas fitted stratified motion planning had only one. This means that one can define strategies which keep the fingers in their workspaces avoiding fingertip collisions. Two simple ones are the followings:

- Finger relocation with straight-line subsegments. The trajectory is composed of three straight-line subsegments. Determine two points above the object starting from the initial and the desired final points, in the direction of normal vector of the surface. This can be a part of overall planner. The two points must be at a distance from the object such that the straight-line connecting the two points does not intersect the object. The two points with the initial and final points define three joined straight-line subsegments.
- Finger relocation with constant distance from the object. The difference from the previous approach is in the

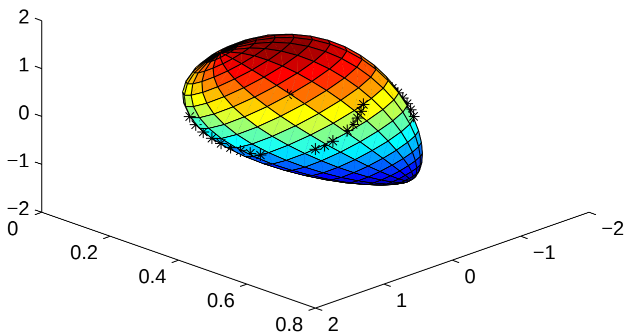

Figure 4.7: The evolution of the planned contact points during the manipulation. (Axis z is directed upwards.)

trajectory between the two points above the object. It is a curve evolving in parallel manner to the surface of the object.

The whole manipulation procedure consists of subsegments snapping the object in the state where the fingertips does not leave their workspaces, yet. The Figure 4.7- Figure 4.9 illustrate the steps of fitted semi-stratified algorithm if the manipulation task is a rotation around the axis $[1\ 0\ 1]^T$. Figure 4.7 shows the evaluation of reference contact points during the object motion. It means that the points marked by asterisk belonged to $x = y$ and $x = -y$ in the plane $z = 0$ at the times when finger relocations were needed. In fact, leaving a workspace is not the only reason to relocate a finger. It is better to choose relatively small subsegments because the convergence of smooth motion planning (using flow sequence) is guaranteed only under a critical distance between the initial and final configuration. Theoretical results are not known in general for critical distance. Consider now only a subsegment. Then, Figure 4.8 depicts the configuration where the phase of fitted stratified motion planning steers the system. After this, the fingers are relocated back into the plane $z = 0$ on the lines $x = y$ and $x = -y$. They assign four contact points. Finger 1 is relocated at first, then Finger 2 and so on. One configuration is snapped and illustrated in Figure 4.9. When all the fingers are relocated, the initial point of the new manipulation phase is arrives (marked also by asterisk in Figure 4.7). The algorithm is summarized as follows.

Algorithm 8. [Fitted semi-stratified MP]

Step 1. Move the object through the desired path.

Step 2. Snap the object from time to time with small enough time steps so that fingertips do not leave their quadrants.

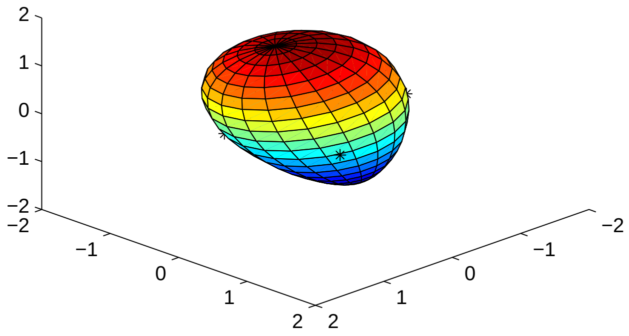

Figure 4.8: The phase of fitted stratified manipulation (object motion).

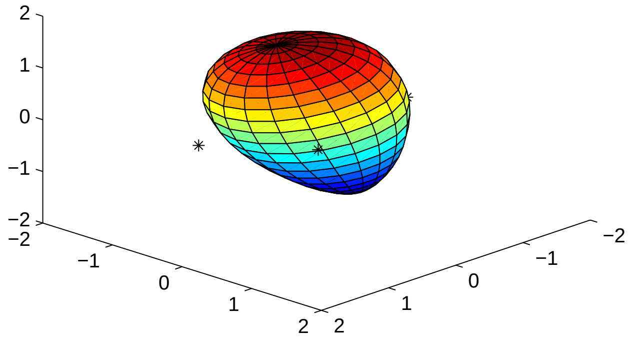

Figure 4.9: The relocation of the first fingertip. (The phase of systematic finger relocation where the object does not move.)

Step 3. Consider two neighboring configurations. The first state represents the initial state $x_{o,I}$ of the object, the second one describes the final state $x_{o,F}$ of the object.

Step 4. Intersect the object in its initial state $x_{o,I}$ with the lines $x \pm y$ in the plane $z = 0$. Let the four contact points be denoted by $C_{i,initial}$, $i = 1 \dots 4$ (expressed in the object frame). The initial state: $x_I = x_{o,I} \oplus C_{1,initial} \oplus \cdots \oplus C_{4,initial}$.

Step 5. Define $x_F = x_{o,F} \oplus C_{1,inter} \oplus \cdots \oplus C_{4,inter}$ as the final state of the object manipulation phase. The intermediate hand configuration $C_{i,inter}$, $i = 1, \dots 4$ is prescribed by the manipulation task but in the simplest case (that is followed also in this treatment), one can choose $C_{i,inter} = C_{i,initial}$, $i = 1, \dots 4$ where the positions of fingertips do not change relative to the object under the phase of object manipulation.

Step 6. Fitted stratified manipulation from x_I into x_F. If one choose $C_{i,inter} = C_{i,initial}$, then this phase is reduced only to an object manipulation and the fingertips do not change position on the object. (The phase of grasp adjustment is realized in the following steps.)

Step 7. Intersect the object in its final state $x_{o,F}$ with the lines $x \pm y$ in the plane $z = 0$. Let the four contact points be denoted by $C_{i,final}$, $i = 1 \dots 4$.

Step 8. Relocate Finger 1 from $C_{1,inter}$ into $C_{1,final}$. This motion planning is carried out in a higher stratum (in S_{234} where the motion of Finger 1 is free). The actual hand configuration: $x_{final1} = [C^T_{1,final}, C^T_{2,inter} \dots C^T_{4,inter}]^T$. In this and the following cases, only the free finger carries out a motion, the other ones stay in their state.

Step 9. Relocate Finger 2 from $C_{2,int}$ into $C_{2,final}$. This motion planning is carried out in a higher stratum (in S_{134} where the motion of Finger 2 is free). The actual hand configuration: $x_{final2} = [C^T_{1,final}, C^T_{2,final} \dots C^T_{4,inter}]^T$.

Step 10. Relocate the Finger 3 from $C_{3,int}$ into $C_{3,final}$. This motion planning is carried out in a higher stratum (in S_{124} where the motion of Finger 3 is free). The actual hand configuration: $x_{final3} = [C^T_{1,final} \dots C^T_{3,final}, C^T_{4,inter}]^T$.

Step 11. Relocate Finger 4 from $C_{4,int}$ into $C_{4,final}$. This motion planning is carried out in a higher stratum (in S_{123} where the motion of Finger 4 is free). The actual hand configuration: $x_{final4} = [C^T_{1,final} \dots C^T_{4,final}]^T$.

Step 12. Repeat the algorithm for the next subsegment.

It is worth concluding that fitted semi-stratified manipulation with its more global considerations (finger relocation strategy) belongs rather to a slightly higher level in manipulation than fitted stratified approach. It comes from the insertion of extra phase consisting of four unconstrained finger relocations. These unconstrained finger relocations were separated from the fitted stratified manipulation. They substitute systematically one part (namely the finger relocation part) of the fitted stratified approach. As a result, one may ensure force closure stability, collision avoidance. However, one should choose the reference contact points in a restricted way (finger relocation when fingers are about to leave their workspaces). It results that the fitted semi-stratified manipulation accomplishes rather an object manipulation than a dextrous manipulation.

4.2.3 Fitted Stratified Manipulation Planning along Decomposed Trajectories

This section presents a new method of exact trajectory tracking for fitted stratified manipulation system. Although the basic idea relies on the results of Subsection 4.2.1, the new method to be proposed can be enhanced holonomic stratified manipulation systems. In this case, not only decoupling condition of stratified motion planning between moving on and moving off vector fields Goodwine (1998a) should be satisfied but moving on vector fields should also span an involutive distribution in each stratum (which is always true, by structure, for fitted stratified manipulation).

Before discussing the method, recall, that manipulation task can be transformed into motion planning problem in the configuration space. The fitted stratified motion planning algorithm produced a path which can be described as a sequence of flows i.e.,

$$x_F = \Phi(g_{i_n}, t_n) \circ \Phi(g_{i_{n-1}}, t_{n-1}) \circ \cdots \circ \Phi(g_{i_2}, t_2) \circ \Phi(g_{i_1}, t_1)(x_I) \tag{4.27}$$

where x_F denotes the desired final state, x_I the initial point, n is the dimension of configuration space. As before, $\Phi(g, t)$ denotes a flow along a vector field g in time interval length t. Vector fields g_{i_j} are the control vector fields of the system with $i_j = 1, \ldots m$, $j = 1, \ldots n$.

Remark 6. Using stratified motion planning, expression (4.27) may contain more terms than n because some flows along Lie brackets expressed by sequence of "original" flows may appear. If bottom stratified *extended* system is not necessary to be constituted (as was the case in Subsection 4.2.1) then the sequence contains exactly n terms.

An argument for finding a trajectory in the form of flow sequence is that stratified motion planning forbids activating inputs at the same time if they belong to different strata . Obviously, if a trajectory is given by flow sequence then only one input is active at a time instant which assigns a stratum.

However, it is not necessary to have only one active input at a time. One can activate more than one input if the active vector fields belong to the same stratum. In this case, extra strata switching is not required. With an appropriate grouping of vector fields within the flow sequence, it is possible to modify resulting trajectory of fitted manipulation toward a desired way.

For this, consider the fitted stratified manipulation again and take the following notations. In order to distinguish and identify strata easily, replace multiindex I of strata (introduced in Section 2.2 and taken part also in (4.22)) by simple index i. Finite dimension of the configuration space enables this substitution. Denote k the number of strata giving vector fields in (4.25), so $i = 1, \ldots, k$. Then, let Δ_i be the distribution spanned by the vector fields which takes part in bottom stratified fitted system (4.25) from the strata S_i. Let d_i be the dimension of the stratum S_i. Then there exist d_i control vector fields in bottom stratified fitted system which are originally defined in stratum S_i. The idea is now to solve motion planning problem in the form

$$x_F = \Phi(g_{k,d_k}, t_{k,d_k}) \circ \cdots \circ \Phi(g_{k,1}, t_{k,1}) \circ \cdots \circ \Phi(g_{1,d_1}, t_{1,d_1}) \circ \cdots \circ \Phi(g_{1,1}, t_{1,1})(x_I). \tag{4.28}$$

Each flow is defined by $g_{i,j}$ and $t_{i,j}$ where $g_{i,j}$ denotes the jth vector field from the ith stratum in the bottom stratified system and $t_{i,j}$ designates the length of time interval while flow along $g_{i,j}$ is active. It is easy to see that

$\sum_{i=1}^{k} d_i = n$. The main feature of this description is that one takes all the possible vector fields from the same stratum at first, then one takes all the possible vector fields from another stratum and so on. Note that so far stratified motion planning has been carried out with the only difference that the vector fields are systematically regrouped. Doing this one easily observes that the configuration space has become split to submanifolds and they are in strong relation with strata. At first, the system moves on a submanifold which is defined by the vector fields of Δ_1, this is followed by the vector fields of Δ_2 and so on. Now define $k+1$ basic points in the configuration space in the following way:

$$\begin{aligned}
b_0 &= x_I \\
b_1 &= \Phi(g_{1,d_1}, t_{1,d_1}) \circ \cdots \circ \Phi(g_{1,1}, t_{1,1})(x_I) \\
&\vdots \\
b_{k-1} &= \Phi(g_{k-1,d_{k-1}}, t_{k-1,d_{k-1}}) \circ \cdots \circ \Phi(g_{1,d_1}, t_{1,d_1}) \circ \cdots \circ \Phi(g_{1,1}, t_{1,1})(x_I) \\
b_k &= \Phi(g_{k,d_k}, t_{k,d_k}) \circ \cdots \circ \Phi(g_{k,1}, (t_{k,1}) \circ \cdots \circ \\
&\quad \Phi(g_{1,d_1}, t_{1,d_1}) \circ \cdots \circ \Phi(g_{1,1}, t_{1,1})(x_I) = x_F
\end{aligned} \tag{4.29}$$

In the next step, one can change the original algorithm and define an arbitrary trajectory between b_0 and b_1 denoted by $x_1(t)$ where $x_1(t) \in S_1$ for all

$$0 < t \leq \sum_{i=1}^{d_1} t_{1,d_i}$$

and $x_1(0) = b_0$ and $x_1(\sum_{i=1}^{d_1} t_{1,d_i}) = b_1$. (The subscript refers to the stratum index.) For this the following proposition is used.

Proposition 7. Consider the manipulation system given by the Subsection 4.2.1 with the fitted bottom stratified system (4.25). Define the basic points $b_0, \ldots, b_k$ by (4.29). This manipulation system is able to move along an arbitrary path between b_{i-1} and b_i in the stratum S_i for all $i = 1, \ldots, k$.

Proof: According to (4.25), the vector fields of the fitted bottom stratified system can be defined by

$$g_{1,1}, \ldots, g_{1,d_1}, \ldots, g_{k,1}, \ldots, g_{k,d_k}.$$

In fact, they are the columns in the matrix (4.25). The rank of this matrix is n and it equals to the dimension of the whole configuration space. It means that the fitted bottom stratified system is holonomic. As a consequence, one does not need the Lie bracket extension to steer the system (4.25) from an initial point to a final point. It is sufficient to use the linear combination of the vector fields $g_{1,1}, \ldots, g_{1,d_1}, \ldots, g_{k,1}, \ldots, g_{k,d_k}$. The only thing one has to show that this holonomy inherits to the subsystem defined in the strata. To do that, cast these vector fields into different groups which are related to strata. Let the first group consist of the vector fields $g_{1,1}, \ldots, g_{1,d_1}$. All of them are defined in the stratum S_1. Additionally, these vector fields are the constant unit vectors (e_i) and define the basis vectors in the tangent space of submanifold S_1. One can easily check from (4.25) that d_1 is exactly the dimension of the stratum S_1.

Extending arguments to every stratum in (4.25) is possible. Doing this, all the tangent spaces of strata will be spanned by exactly as many vector fields as many is the dimension of the given stratum. In other words, the subsystems on submanifolds defined on strata are also holonomic, Lie bracket extension is not needed and as a straightforward conseqence, the subsystems $\Sigma_{fitted,S_1}, \ldots, \Sigma_{fitted,S_k}$ are able to move along an arbitrary path between b_{i-1} and b_i in the stratum S_i for all $i = 1, \ldots, k$ and the proof is completed.

Based on the Proposition 7, one can plan an arbitrary path between b_0 and b_1 on the submanifold spanned by the vector fields $g_{1,1}, \ldots, g_{1,d_1}$ and constrained only by the start points b_0 and the final points b_1. The procedure proceeds with the next submanifold in the same way where points b_1 and b_2 play important role involving the vector fields of Δ_2 and so on. Unfortunately, there is no choice to prescribe the basis point $b_1, \ldots, b_k$ in arbitrary way, however, there is a freedom to apply an optimal set of contact points in a sense. To show this, define the distance $D(x_p, x_d(t))$ between a basis point x_p and a desired trajectory $x_d(t)$ in the configuration space,

$$D(x_p, x_d(t)) = \text{distance}\left(x_p, x_d(\sum_{p=1}^{k} t_p)\right) \tag{4.30}$$

where $t_p = \sum_{j=1}^{p} \sum_{i=1}^{d_j} t_{j,d_i}$ is the time (after optional time scaling if it is necessary) when the basis point x_p is reached by motion planning along sequence of flows. The distance can be easily computed numerically. Let π denote a permutation of $(1, \ldots, k)$. It is possible to carry out a motion planning as flow sequences by different permutations of strata. Such a flow can be expressed in the form,

$$\begin{aligned} x_F &= \Phi(g_{\pi(k),d_{\pi(k)}}, t_{\pi(k),d_{\pi(k)}}) \circ \cdots \circ \Phi(g_{\pi(k),1}, t_{\pi(k),1}) \circ \cdots \\ &\circ \Phi(g_{\pi(1),d_{\pi(1)}}), t_{\pi(1),d_{\pi(1)}}) \circ \cdots \circ \Phi(g_{\pi(1),1}, t_{\pi(1),1})(x_I) \end{aligned} \tag{4.31}$$

Let b_π be the set of k basis points (not counting $b_0 = x_I$) due to the permutation π i.e. $b_\pi = (b_\pi(1), \ldots, b_\pi(k))$. Define

$$D(\pi, x_d(t)) = \sum_{i=1}^{k} D(b_\pi(i), x_d(t))$$

the aggregated distance between path and basis points belonging to permutation π. Considering the minimal error between the desired trajectory $x_d(t)$ and basis points, the optimal permutation of the strata is written as $\pi_o = \arg\min D(\pi, x_d(t))$. It means that one should search the actual trajectory in the form,

$$\begin{aligned} x_F &= \Phi(g_{\pi_o(k),d_{\pi_o(k)}}, t_{\pi_o(k),d_{\pi_o(k)}}) \circ \cdots \circ \Phi(g_{\pi_o(k),1}, t_{\pi_o(k),1}) \circ \cdots \circ \\ &\quad \Phi(g_{\pi_o(1),d_{\pi_o(1)}}, t_{\pi_o(1),d_{\pi_o(1)}}) \circ \cdots \circ \Phi(g_{\pi_o(1),1}, t_{\pi_o(1),1})(x_I) \end{aligned} \tag{4.32}$$

Using the example of fitted stratified manipulation, the algorithm is summarized as follows:

Algorithm 9. [Fitted manipulation with decomposed path planning]

1. Apply the Algorithm 6 using the form (4.31) for every $\pi \in P$ permutations to solve the motion planning problem of system given by (4.19)-(4.23).
2. Compute the basis points $(b_1, \ldots, b_k)$ to each permutation using (4.29).
3. Determine the $\pi_o = \arg\min D(\pi, x_d(t))$ which provide the optimal sequence of strata in the flow sequence.

4. Exact path planning for holonomic subsystems defined in the subspaces (strata) between b_{i-1} and b_i for all $i = 1, \ldots, k$.
5. Inserting to the moving off vector fields in accordance with Algorithm 6.

Comparing with the methods of Subsection 4.2.1, the advantage of method is easily revealed. The proposed method gives a greater freedom in choosing paths on submanifolds. The simple fitted system and the holonomy assure the efficiency of the algorithm. However, basis points still generate restrictions in trajectory planning since every path should travel through them.

The method introduced in this section inspired the generalization to holonomic stratified systems in Chapter 5.

4.3 Examples

Three algorithms for object manipulation problem were proposed in previous section whereas object is of smooth surface. This section demonstrates the simulation results in order to highlight typical features. As an example, consider a manipulation task consisting of object reorientation. Let the manipulation system be due to section 4.1. Let the desired object motion be a rotation around the axis $(\ 1\ 1\ 1\ )^T$ while the position p_x, p_y, p_z of the object frame does not change. Additionally, let desired the final fingertip contacts be found in the plane $z = 0$ where the lines $x = \pm y$ intersect the object. The task implies a point to point manipulation problem (considered as motion planning problem in configuration space) from the point of view of fingers and a trajectory tracking problem from the point of view of the object. Since stratified and semi-stratified manipulation planning methods involves only point to point motion planning problem, one should reduce the problem. To do so, object trajectory is to be divided into connecting subsegments where the ends of subsegments should lie on the reference trajectory generated by the prescribed rotation around axis $[\ 1\ 1\ 1\ ]^T$.

Fitted stratified manipulation is to be analyzed at first. As was discussed above, this method is not restricted in reference fingertip positions. Theoretically, any fingertips can be reached at the end of manipulation phase but because of this, one should check whether the fingers leave their allowed workspaces (the quadrants). Hence it is assumed that manipulation task does not require the fingertips to leave their quadrant while they travel to their desired final state.

Desired fingertip positions are obtained as follows. First, desired final object configuration is to be determined in each subsegment from the desired object trajectory by snapping. At this stage, the intersections of the object with the $x = \pm y$ lines in the plane $z = 0$ point out four fingertip positions. The object configuration and these 4 points give the desired final state for fitted manipulation in this subsegment. Now, one carries out the next part of the desired object rotation phase that determines the desired final object states and the desired final fingertip position in the next subsegment and so on. Leading the object along reference trajectory hypothetically, one can obtain the initial and final points to the fitted stratified manipulation in each subsegment. (The initial state is always the final state of the previous section). The reference final fingertips are defined similarly, by means of a trajectory designed between initial and final states. The simulation results are illustrated in the Figure 4.10 - Figure 4.16. Introducing the notations $\omega_o = [\dot{\phi}_x \ \dot{\phi}_y \ \dot{\phi}_z]^T$ for the components of the angular velocity of the object, Figure 4.10 displays the evolution of orientation of object during the manipulation.

In accordance with the philosophy of the stratified control, only one fictitious input is active at a given time instant as seen in the Figure 4.10. As a consequence, the object and fingers move along one vector field at any time. Since the fitted system is very simple, the nature of the motion is easily interpretable. More exactly, the object motion consists of a sequence of the rotation around the axis x, y and z which are performed one after the other. The initial and final stages of this part of process are illustrated in Figure 4.12 and Figure 4.13. As known from Section 4.2, one has to determine the real input via transformations (4.21) and (4.23). The real inputs are the joint variables and the ones due to the Finger 1 are depicted in Figure 4.11. Although only one fictitious input is active at a given time, it is not necessarily true for real inputs (i.e. for joint variables). One fictitious input is derived namely from more than one real input via a transformation. Comparing Figures 4.10 and 4.11, one observes that the most dynamic part in the joint variables belong to finger relocation. The reason is that moving off vector fields are inserted in this time intervals disturbing the smoothness of the characteristics and performing finger relocations. Finger relocations are carried out along distinct directions above the object. In the beginning, the finger arises from the object in the direction of normal vector to the surface. Maintaining distance, the fingertip moves above the object. This motion is performed along explicit parameters of the object. It is very important, that finger moves along only one parameter, at a given time instant. Furthermore, one can also easily see from the structure of fitted system and the algorithm of stratified control that finger relocations are decoupled. While a finger is being relocated, the joint variables of other fingers do not change (see constant intervals in Figure 4.11). The process of two finger relocations during manipulation is illustrated in the Figure 4.14 - Figure 4.15 (in fact, the other two fingers are also relocated in the same way). They snapped a state when one of the fingertips is being relocated i.e. it is not in contact with the object. After finger relocations, the manipulation is followed by new local manipulation task in the next subsegment as before (Figure 4.16 - Figure 4.17). The advantage of demonstrated fitted stratified manipulation is the simpler symbolic computation. As a matter of fact, the symbolic computation related mainly to Philip Hall coordinates is reduced to almost pure numerical computations due to the simple vector fields. The fitted stratified motion planning enables to interpret the fingertip motions directly from the result of its algorithm and it is not needed to decode them from the joint variables. It belongs to future work how to perform a finger relocation generally at arbitrary manipulation planning, when one finger is about to leave its own workspace. Although the quadrants as workspaces provide intuitively a reasonable structure to establish a grasp with force closure stability, the method does not ensure force closure stability automatically. It is required to choose reference contact points on the surface in appropriate way. It is one of the reasons that motivated to devise the fitted semi-stratified manipulation.

The simulation result of the manipulation using *fitted semi-stratified manipulation* leads to similar simulation

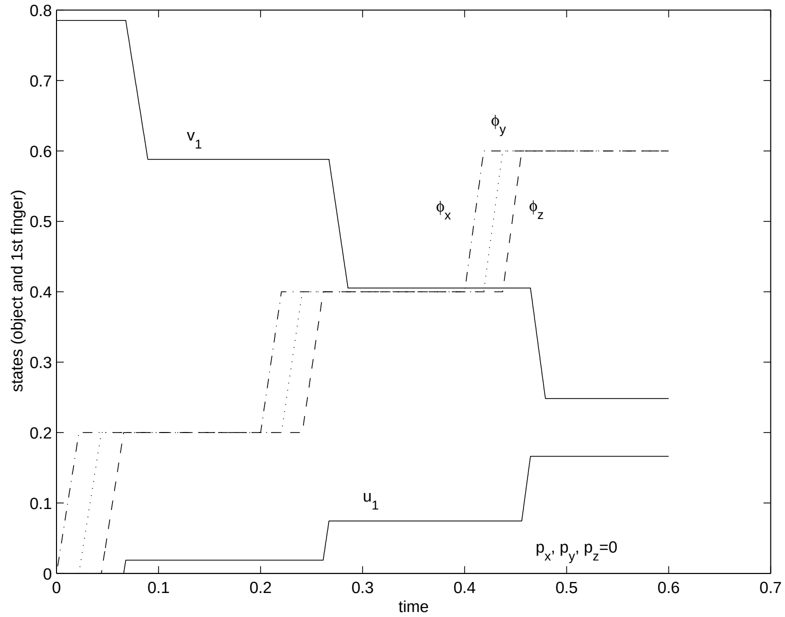

Figure 4.10: The states related to the object orientation and the 1st finger tip position ($p_x = 0,\ p_y = 0,\ p_z = 0$) by using Algorithm 7. The states are given in SI (rad and sec).

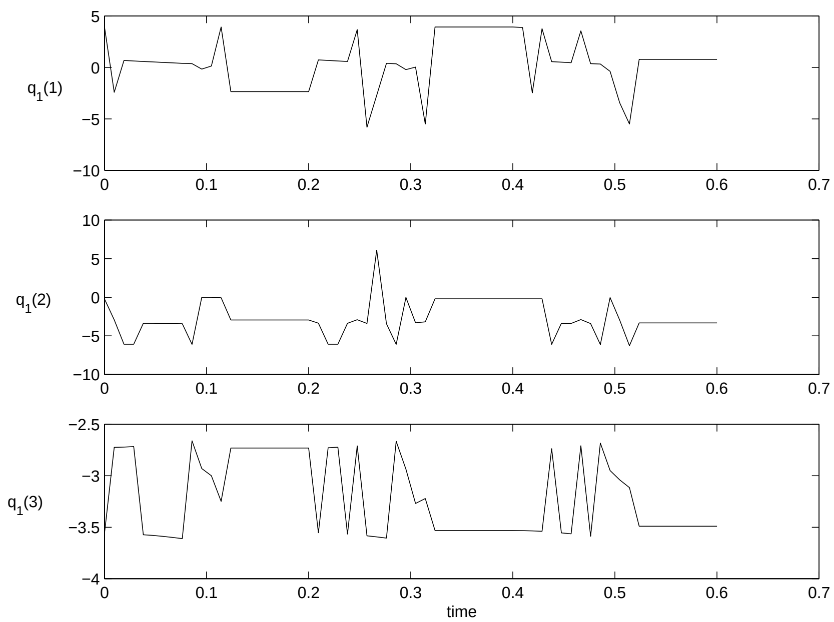

Figure 4.11: The real inputs belonging to the joint variables of the finger 1.

result as fitted stratified manipulation. The difference between them is found only in finger relocations and the way of choosing the reference points. Semi-stratified motion planning is devoted primarily only to object manipulation (and not dextrous manipulation) because fingertips are not chosen in an arbitrary way. For reference states, the procedure of fitted stratified approach is adopted excepting one difference. The method has two main parts. The desired final states for semi-stratified manipulation in a subsegment during the object manipulation phase are given by the object motion with fixed contact points to the surface. Then the actual fingertip configurations will be the initial state of a second phase, namely, of unconstrained finger relocations. The prescribed final states for the unconstrained finger relocations are the contact points in the lines $x = \pm y$, $z = 0$ as before at the fitted stratified manipulation. So fingertips are pulled back into $z = 0$. The semi-stratified approach sets apart a systematic finger relocation task from the object manipulation ensuring collision avoidance and force closure stability (similarly to Goodwine (1999)). At the same time, the fingertips can not be relocated to an arbitrary place on the surface. The object motion implied by semi-stratified approach coincides with the pure stratified approach (Figure 4.12- Figure 4.17). However, the motion of fingertips during relocations is not necessary the same. Semi-stratified control gives a greater freedom in choosing the path of fingertips when they are not in contact with the object. At the same time, this technique is not as unified as the (fitted) stratified manipulation. In addition, one needs to solve an extra computational task in "free" space for relocating fingers from a surface point to another one. It is also worth remarking that fitted semi-stratified manipulation can be applied specifically only to the manipulation problem posed in Section 4.1.

Consider now the manipulation concept *using decoupled trajectories* in Subsection 4.2.3 that exploited the advantages of the fitted description. Figure 4.10 illustrates the simulation result of fitted stratified manipulation (see Algorithm 7) and the Figure 4.18 demonstrates the results provided by Algorithm 9. Let the desired object motion be a rotation around the axis $[\ 1\ 1\ 1\ ]^T$ as before. (However, the prescribed fingertip positions in final states are different.) Figures show the states u_1, v_1 which determine the first fingertip motion. In fact, states $u_1(t)$ and $v_1(t)$ assign a trajectory on the object. Figures also show the evolution of the other part of the state vector represented by the angles around the frame axis x, y and z. The derivatives of the angles represent the components of angular velocity $\omega_o = [\dot{\phi}_x\ \dot{\phi}_y\ \dot{\phi}_z]^T$, similary to the examples above. The position of the object (which does not change in our desired manipulation task) plays similar role with the linear velocity $v_o = [\dot{p}_x\ \dot{p}_y\ \dot{p}_z]^T$. One can easily see the similarities and differences between the two methods (Figure 4.18 related to trajectory tracking due to Algorithm 9 and Figure 4.10 is related to basic fitted stratified manipulation due to Algorithm 7).

Note that time can be rescaled in a real application. Both algorithms decompose indirectly the manipulation task into an object manipulation phase and a finger relocation phase. The object manipulation occurs when u_i and v_i do not change for every i. On contrary, the finger relocation phase is carried out in the time interval when the object position and orientation do not vary. It is easy to recognize that the parts of the state vector related to the object motion ($\dot{\phi}$'s and $\dot{p}$'s) are always zero while the part of the state vector related to the

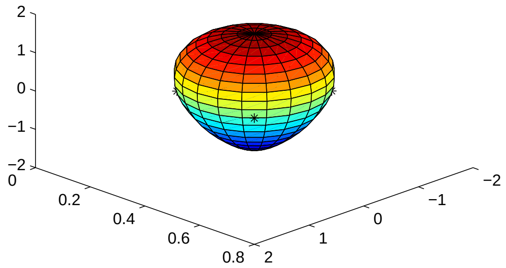

Figure 4.12: Phase 1. The initial configuration. (The vertical axis is the axis z.)

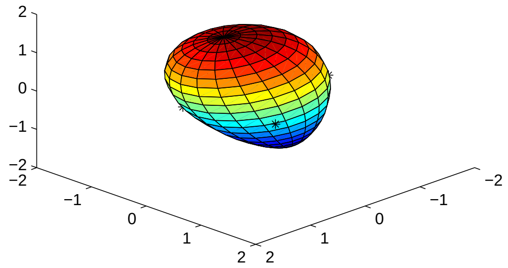

Figure 4.13: Phase 2. Object manipulation phase in the bottom stratum.

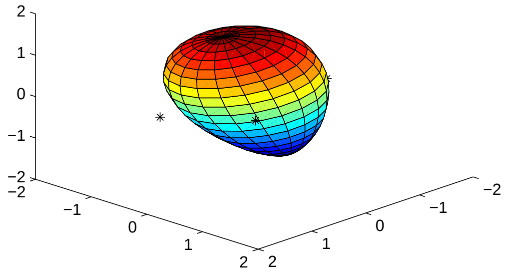

Figure 4.14: Phase 3. Finger relocation in higher stratum.

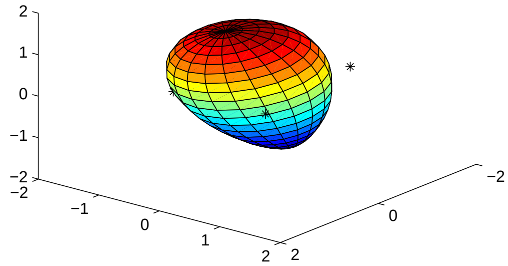

Figure 4.15: Phase 4. Another finger relocation in another stratum.

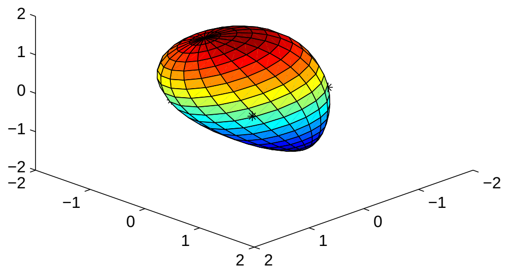

Figure 4.16: Phase 5. The final configuration of fitted stratified manipulation in the first subsegment.

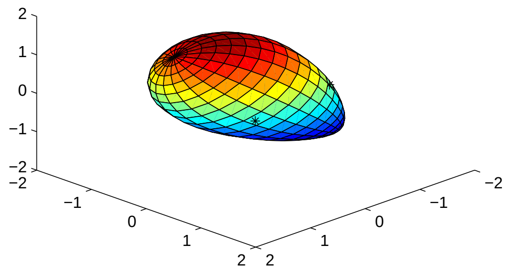

Figure 4.17: Phase 6. The final configuration in the last subsegment.

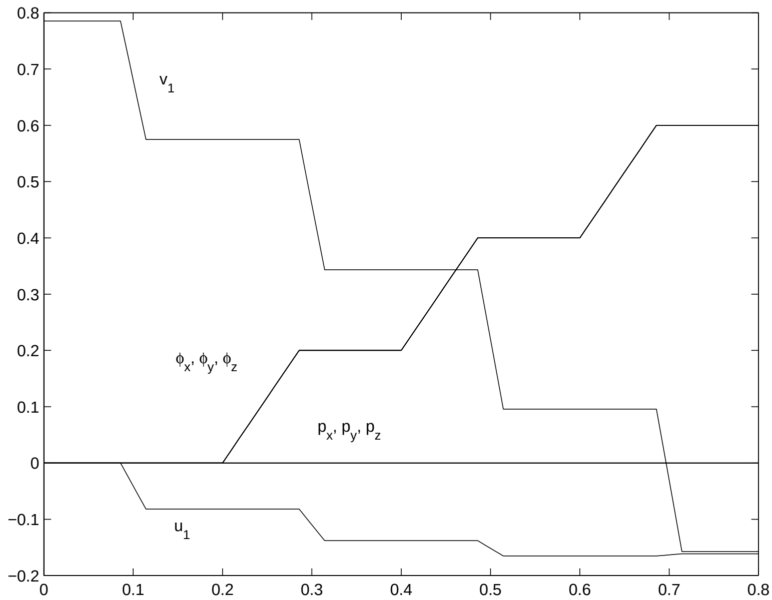

Figure 4.18: The states related to the object orientation and the 1st finger tip position ($p_x = 0,\ p_y = 0,\ p_z = 0$) by using Algorithm 9. The states are given in SI (rad and sec).

fingertip motion is changing and vice versa. It means that both parts of the state vector are never active at the same time. The basic method decomposes indirectly further the object manipulation phase, i. e. the algorithm moves the object with changing only one state (one of the angles or the position coordinates) at a given time. The extended algorithm works in a little different way than the basic one. It groups the states belonging to the same stratum and it changes these state at the same time. However, the initial and final object configurations should remain the same in every object manipulation phase. It becomes possible to combine their inputs and to follow a trajectory as shown on Figure 4.18. The object reaches the new object orientation via rotation around desired axis. Instead, the basic version accomplished this task via successive rotations around the axis of inertial frame. Additionally, while fingertips could move only along object parameters when basic method is applied, the extended method leads fingertips along an arbitrary trajectory above the object into the new contact point. The 3D object motion during the manipulation is illustrated in Figure 4.12 - Figure 4.16. The phases are the same in the case of both methods. The difference is hidden in the object manipulation phase while all the fingers are in contact with the object. In one of these phases, the object moves from the state of Figure 4.12 to the state of Figure 4.13 along a path which is different from the above one. Because of the philosophy of the stratified control, the extended method still keep the "sequence" approach. It means that at first only those inputs become active which are related to the object manipulation (with fixed contact points). After this one can carry out the finger relocations one after the other. The method can be used for obstacle avoidance since object and fingers are able to track a restricted class of trajectories.

The method also inherits the benefits of the fitted stratified approach like simpler symbolic computations and the less expensive computational cost. The symbolic computation appearing generally when one solves an equation like (2.52) is reduced to almost pure numerical computations due to the simple vector fields. It is also remarkable that introducing extended bottom stratified system is not needed. If one considers the time and velocity conditions, the extended method shows advantages again. The reason is that it reduces the length of the sequences from n (dimension of configuration space) to k (number of strata). Of course, one has to pay attention to the stable grasp during the finger relocations which is omitted in this discussion. The phenomenon has essential importance if only 3 fingers are in contact with the object.

4.4 Summary and Conclusions

This chapter proposed three manipulation methods based on fitted model. They rely on stratified motion planning and aim to reach a final configuration through the end points of connected subsegments. The fitted system provides simple control vector fields. It reduces the complexity of the differential geometric computations playing important role for motion planning algorithms based on symbolic treatments.

The method using stratified motion planning on a fitted system as proposed in Subsection 4.2.1 shows advantages in comparison to the earlier stratified motion planning, such as avoiding the hard symbolic computation and providing easily interpretable trajectory. It also assists to reduce the symbolic steps almost to pure numerical computations. The features are based on the special parametrization of the fitted system. Furthermore, it is easy to check a property of the fitted model like the stratified controllability. Basic fitted stratified manipulation cannot keep automatically the fingertips in their workspace since one does not know anything about the trajectory between the initial and final points in a subsegment. It results that the realization needs extra functions on higher level which on-line check the fingertip positions and modify the specifications in a subsegment, if it is necessary. It assists to avoid collisions. If the specifications of manipulation task guarantees the collision avoidance then the technique is able to perform a dextrous manipulation in a given region i.e. both the desired object position, orientation and the fingertips position become reachable. In return, one has to accomplish an extra transformation between the original and the fitted system. This transformation related to Jacobian matrices of the fingers provide the physical control signals represented by the joint variables. Although the transformations require additional steps, their usage become useful because they make possible to reduce the most complex parts of the symbolic computations. One drawback of the method appears in the issue of grasp stability. It is not guaranteed during the manipulation unless one takes into account higher level considerations such as generation of appropriate reference fingertip trajectories.

The elimination of this drawback is one of the main motivations to fitted semi-stratified manipulation proposed in Subsection 4.2.2. The method differs at two main points from the above mentioned stratified version. First, it requires suitably chosen reference contact points. Second, it extends basic fitted stratified approach with unconstrained finger relocations. The technique which provides appropriate contact points restricts also the manipulation. Namely, the dextrous manipulation becomes rather only object manipulation. However, it assures the collision avoidance and grasp stability at the same time. The unconstrained finger relocations in semi-stratified approach realize a second phase in manipulation after the fitted stratified manipulation is executed. It accomplishes a finger relocation algorithm with a greater freedom than basic fitted stratified manipulation. Finger relocations necessitate extra computations but they increase the degree of freedom in fingertip motions. Namely, a finger can be relocated along arbitrary path above the object while a fitted stratified approach should move the fingertip along only one certain direction above the object at a time.

The two methods resulted the same object motions during the simulation but the fingertips followed different trajectories. This chapter demonstrated the algorithms via a smooth object manipulation example using a four fingered robotic hand.

The basic fitted concept may allow to divide the configuration space of the bottom stratum into slices (subspace decompositions). Subsection 4.2.3 proposed an improved fitted stratified manipulation planning which provides exact trajectory tracking on the submanifolds defined by strata. The main advantage of the method that the algorithm is able to move the object along arbitrary path in a stratum and in the next sequence it can relocate the fingers along arbitrary trajectory above the object, as well. At the same time the algorithm inherits the benefits of the fitted stratified manipulation such as reduced symbolic computation and computational costs. One can also interpret easily the path of the finger tips and the object from the results of the motion planning algorithm. It induces that one may apply the method to obstacle avoidance since the object can locally move in desired way in the configuration space. The method works only on kinematic model and is generally restricted by fixed basis points since every reference path has to go through them.

Chapter 5

Holonomic Stratified Motion Planning with Decomposed Trajectory

This chapter is motivated by Subsection 4.2.3 and extends stratified motion planning to a special class of stratified system called holonomic stratified systems. Extension consists of finding decomposed trajectory between initial and final states that may be followed under suitable control. Note that this chapter considers trajectory tracking problem, under some restrictions (see below) and not a simple path planning (which, in strict sense, disregards time relations). Since restrictions arise, the motion planning as a general term is still used for the methodology. An obvious benefit to track any trajectory, even if it is restricted in some terms, appears in solving obstacle avoidance problem.

As a first step, Section 5.1 establishes the foundation of the new concept relying on the central notion of completely involutive representation (CIR) derived from bottom stratified system. Some results how to design decomposed trajectory for completely involutive representation are also presented in this section.

There may exist several CIRs of the bottom stratified system (BSS). Section 5.2 pursuits three techniques for choosing a representation with the best properties from different point of views.

Stratified motion planning methods in Chapter 2 and Chapter 3 concern a fixed order of flows. Section 5.3 reveals that the shape of trajectory depends on the order of the flows and one may influence the error between desired and resulting trajectories by changing the order of flows in appropriate manner.

To design a subtrajectory evolving on a slice is not elementary even if its dimension coincides with the number of available inputs. The difficulty arises from the fact that one seeks the solution of nonlinear differential equation occured in this discipline as a composition of maps. Section 5.4 offers a method which is based on symbolic approximation.

Section 5.5 displays the steps of the technique on a modified example of hexapod robot.

The contributions this chapter asserts can be found also in Harmati, Lantos, and Payandeh (2002c), Harmati, Lantos, and Payandeh (2002a), Harmati, Lantos, and Payandeh (2002b).

5.1 Fundamentals of Holonomic Stratified Motion Planning

Stratified motion planning presented in Chapter 2 and Chapter 3 provides algorithms that reaches a desired final point. However, it is not possible to plan a precise trajectory between the initial and final points. In fact, one cannot say in the design phase where the system is found at a given time. To overcome the problem, the next sections generalize the theory of stratified motion planning. The generalization renders a procedure that is able to lead the stratified system along a special type of trajectories called decomposed trajectories. This chapter often recalls smooth motion planning, which in basic understanding refers to smooth motion planning along sequence of flows discussed in Subsection 2.1.2.

After recalling the definition of holonomic smooth system in Subsection 2.1.1 the new concept begins with some definitions and conventions in notations. From now, the configuration space of bottom stratum will be denoted by $\mathcal{C}_0$.

Definition 19. A stratified controllable system is said to be *holonomic* if it is possible to find $n = dim(\mathcal{C}_0)$ independent moving on vector fields.

It is worth highlighting that the term holonomic in this definition intends to refer rather to consisting of holonomic (involutive) subsystems of bottom stratified system than the holonomic bottom stratified system in itself. Such a system arised for example in Subsection 4.2.3 and its specific properties exploited there is to be generalized now.

Assumption 3. This chapter pursuits the solution of motion planning problem if stratified system is endowed with the property concerned in Definition 19.

Motion planning would be very easy if the system was not stratified but a smooth one (in addition, with n dimension) and holonomic. However, the stratifaction does not allow to put into use all the vector fields at the same time. This uncomfortable phenomena inspires to devise new techniques that handle the difficulties.
Start the discussion from BSS (2.55) and simplify some notations. For, denote a vector field $g_{i,j} \mid_{S_b}$ simply by $g_{i,j}$ omitting $\mid_{S_b}$ where the index i assigns the stratum from which one originates the vector field and index j identifies the vector field in the ith stratum. Precisely, a stratum was previously defined by a multi-index I. However, since there are finite number of strata, one can change multi-index I to a simple integer i as done, indeed. Of course, one has to consider the jth vector field of the ith stratum in the tangent space of bottom stratum meaning that some dimensions of the original vector field will be cut off. Let Δ_i denote the distribution of the control vector fields in the ith stratum, i.e.

$$\Delta_i = span(g_{i,1}, \ldots, g_{i,d_i}) \tag{5.1}$$

where d_i denotes the number of vector fields in BSS from the ith stratum S_i. Special multiindex $J(k) = j_1 j_2 ... j_k$ will be also often considered which is in relation to a stratum S_i and defined by k regular indices. One refers to zth index of $J(k)$ as $J(k)(z)$, i.e. $J(k)(z) = j_z$ where $z \leq k$. It is always assumed that every index of multiindex

$J(k)$ have different values and none of them is greater than d_i. The treatment desires to introduce another multiindex, too. If s denotes the number of strata which takes part in BSS then the multiindex $I(l) = i_1 i_2 ... i_l$ points out a set of strata, or more specifically a sequence of strata. Unless anything else is said, the indices in $I(l)$ may have same values and $l > s$ is also permitted.

Definition 20. Consider a stratum S_i and let be given a multi-index $J(k) = j_1 j_2 ... j_k$, $k < d_i$. The distribution $\Delta_i^{J(k)}$ defines a *subdistribution of the stratum* Δ_i where $\Delta_i^{J(k)} = span(g_{i,j_1}, \ldots, g_{i,j_k})$.

Definition 21. Consider the multiindex $I(l)$ and the set of multiindices $J(k_1, \ldots, k_l) = (J_1(k_1), \ldots J_l(k_l))$. Then $\Delta_{I(l)}^{J(k_1,\ldots,k_l)}$ defines a set of subdistributions such that

$$\Delta_{I(l)}^{J(k_1,\ldots,k_l)} = span(g_{i_1,J_1(k_1)(1)}, \ldots, g_{i_1,J_1(k_1)(k_1)}, \cdots g_{i_l,J_l(k_l)(1)}, \cdots g_{i_l,J_l(k_l)(k_l)}) \tag{5.2}$$

For the easier notations, $\Delta(k_1, \ldots, k_l)$ will be often used instead of $\Delta_{I(l)}^{J(k_1,\ldots,k_l)}$ or $\Delta_{I(l)}(J(k_1, \ldots, k_l))$.

Remark 7. To avoid redundancy, it is claimed in the following that if two indices i_1 and i_2 in $I(l)$ have the same values then their corresponding multi-indices $J(k_1)$ and $J(k_2)$ are different in each element.

Remark 8. It is possible to consider $\Delta_{I(l)}(J(k_1, \ldots, k_l))$ not only as a set but also as a sequence of distributions. In this case, one employs rather the notation $\Delta_{I(l)}\{J(k_1, \ldots, k_l)\}$.

Definition 22. Suppose, a distribution Δ is defined as $\Delta = span(g_1, \ldots, g_n)$. Then, one can define a function $gen(\Delta) = (g_1, \ldots, g_n)$ which is called the *generator of the distribution* Δ.

Remark 9. In strict sense, only vector fields define distributions and a distribution does not define vector fields uniquely. However, in the context of this chapter, a distribution is always defined by explicit vector fields which enables to define its generators uniquely. As a matter of fact, producing generators from distribution can be considered as an inverse function of creating distributions from vector fields.

Definition 23. Consider a set of subdistributions $\Delta(k_1, \ldots, k_l)$. A subdistribution $\Delta_i^{J(k)} = \Delta_i(J(k))$ of Δ_i is said to be an *independent core of* Δ_i with respect to $\Delta(k_1, \ldots, k_l)$ if all the vector fields which define $\Delta_i(J(k))$ are independent from the vector fields which define $\Delta(k_1, \ldots, k_l)$ i.e. an independent core $\Delta_i(J(k))$ satisfies that $rank(\Delta_i(J(k)) + rank(\Delta(k_1, \ldots, k_l)) = rank(span(gen(\Delta_i(J(k)), gen(\Delta(k_1, \ldots, k_l))))$. The set of all independent cores of Δ_i with respect to $\Delta(k_1, \ldots, k_l)$ will be denoted by $core(\Delta_i, \Delta(k_1, \ldots, k_l))$.

Definition 24. Let $\Delta_{I(l)}(J(k_1, \ldots, k_l)) \setminus \Delta_{i_j}(J(k_j))$ define the operation

$$\begin{aligned} \tilde{\Delta}_{i_j}(J(k_j)) &\equiv \Delta_{I(l)}(J(k_1, \ldots, k_l) \setminus \Delta_{i_j}(J(k_j) \\ &= span(gen(\Delta_{i_1}(J(k_1))), \ldots, gen(\Delta_{i_{j-1}}(J(k_{j-1}))), gen(\Delta_{i_{j+1}}(J(k_{j+1}))), \ldots, gen(\Delta_{i_l}(J(k_l)))) \end{aligned} \tag{5.3}$$

which is a kind of subtraction between subdistributions of $\Delta_{I(l)}(J(k_1, \ldots, k_l))$.

Definition 25. A distribution $\Delta_i(J(k))$ which satisfies that $\bar{\Delta}_i(J(k)) = \Delta_i(J(k))$ and $\Delta_i(J(k)) \subset \Delta_i$ is called an involutive interior of Δ_i and will be marked as $\underline{\Delta}_i(J(k))$. The set of all subdistributions $\underline{\Delta}_i(J(k))$ will be called the *set of all involutive interiors of* Δ_i and denoted by $\underline{\Delta}_i$.

Remark 10. In Definition 25, $\underline{\Delta}_i(J(k))$ does not define a different distribution from $\Delta_i(J(k))$, it just intends to remind one in the further investigations that the subdistribution is an *involutive* interior of Δ_i.

Remark 11. It is easy to see that every nonempty Δ_i distribution containing d_i independent vector fields has at least d_i involutive interiors. It comes from the facts that every single vector field $g_{i,j}$ can be considered as a subdistribution of Δ_i and it is known from differential geometry that every vector fields is involutive to itself because $[g, g] = 0$.

Lemma 1. The smooth motion planning algorithm (Algorithm 6) induces a sequence of involutive interiors

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \{\underline{\Delta}_{i_1}(J_{k_1}), \ldots, \underline{\Delta}_{i_l}(J_{k_l})\} \quad i_j \leq s,\ 1 \leq j \leq l \tag{5.4}$$

Proof: Smooth motion planning algorithm generates a trajectory between the initial state $x(0)$ and the final state $x(T)$ in the form of sequence of flows. It means that trajectory evolves along one of the control vector fields of the system at each time point t. As a result, time interval $0 \leq t \leq T$ can be split into finite number of joining time intervals with the property that the same vector field is active at each time in a given time interval. Recording the active vector fields of the time intervals in time order while t is going from $t = 0$ to $t = T$, a sequence of active control fields is obtained. It is known from the differential geometry that each single vector field is also an involutive distribution, i.e. $[g, g] = 0$, hence the sequence of the active vector fields in smooth motion planning defines a sequence of involutive interiors.

Definition 26. Consider a BSS where $span(\Delta_1, \ldots, \Delta_s) = T_x\mathcal{C}_0$ and let $x(0) \in \mathcal{C}_0$ and $x(T) \in \mathcal{C}_0$, $T > 0$ i.e. two arbitrary points in the configuration space of BSS. Let s be the number of strata in the BSS and consider a sequence of involutive interiors

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \{\underline{\Delta}_{i_1}(J_{k_1}), \ldots, \underline{\Delta}_{i_l}(J_{k_l})\} \quad i_j \leq s,\ 1 \leq j \leq l \tag{5.5}$$

with the property that

(CIR1.) The distributions $\underline{\Delta}_{i_j}(J_{k_j}) \quad 1 \leq j \leq l$ are nonsingular.

(CIR2.) Each involutive interiors in (5.5) has independent core with respect to others, i.e.

$$core(\underline{\Delta}_{i_j}(J_{k_j}), \tilde{\underline{\Delta}}_{i_j}(J_{k_j})) \neq \{\emptyset\} \tag{5.6}$$

for every $1 \leq j \leq l$.

(CIR3.) there exists a trajectory $x(t)$ between $x(0)$ and $x(T)$ and a sequence of time intervals $t = \{t_1, \ldots t_l\}$ such that

$$\begin{aligned}
\sum_{z=1}^{l} t_z \quad &= \quad T \\
t < t_1 \quad &\Rightarrow \quad \dot{x}(t) \in \underline{\Delta}_{i_1}(J_{k_1})(x(t)) \\
\sum_{z=1}^{j-1} t_z \leq t < \sum_{z=1}^{j} t_z \quad &\Rightarrow \quad \dot{x}(t) \in \underline{\Delta}_{i_j}(J_{k_j})(x(t)) \quad 1 < j \leq l
\end{aligned} \tag{5.7}$$

If the properties (CIR1) - (CIR3) hold then (5.5) is called a *completely involutive representation (CIR) of BSS* and the trajectory satisfying (5.7) is said to be a *decomposed trajectory along CIR* .

Remark 12. The trajectory defined in Definition 5.5 reaches $x(T)$ through slices illustrated in Figure 5.1. More exactly, the trajectory evolves at first on an integral submanifold in $\mathcal{C}_0$ which is determined by some vector fields of S_{i_1}. Then, as a result of switching, the system turns onto a second slice assigned by some vector fields of S_{i_2}, when $t_1 \leq t < t_1 + t_2$ and so on. (The term slice is rather used instead of stratum to emphasize that all the vector fields are considered now in the bottom stratum and some vector fields of a stratum may be omitted.)

Lemma 2. If a sequence of involutive interiors

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \{\underline{\Delta}_{i_1}(J_{k_1}),\ldots,\underline{\Delta}_{i_l}(J_{k_l})\} \quad i_j \leq s,\ 1 \leq j \leq l$$

(see (5.5)) is a completely involutive representation of BSS then
(i.) $\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} \equiv (\underline{\Delta}_{i_1}(J(k_1)),\ldots,\underline{\Delta}_{i_l}(J(k_l)))$ satisfies the Lie Algebra Rank Condition (LARC).
(ii.) It holds that $\sum_{j=1}^{l} \dim\underline{\Delta}_{i_j}(J_{k_j})) \geq n$.

Proof: (i.) For real analytic systems, the accessibility is equivalent to the LARC where the accessibility property means that considering the vector fields of the BSS as a system (recall, it is a fictitious system), for every x, the set of reachable points from x has a nonempty interior. Additionally, one deals with system without drift which makes the completely controllability equivalent to the accessibility (Sussmann (1992)). So, one has to show only that the CIR as a subsystem of BSS created from subset of its vector fields is completely controllable or accessible. The complete controllability is obviously true because, by definition of CIR, there exists a trajectory that leads the BSS from an arbitrary $x(0) \in \mathcal{C}_0$ to an arbitrary $x(T) \in \mathcal{C}_0$, $T > 0$ in a way that the tangent vector of the trajectory $x(t)$ can be produced by the linear combination of the vector fields of CIR at any time. As a result, the CIR satisfies LARC which completes the proof.
(ii.) Since CIR satisfies LARC, $span(\underline{\Delta}_{i_1}(J(k_1)),\ldots,\underline{\Delta}_{i_l}(J(k_l))) = n$ which proves the statement.

Proposition 8. Consider a nonholonomic BSS $G^{BSS} \equiv (g_{1,1},\ldots,g_{1,d_1},\ldots,g_{1,s},\ldots,g_{1,d_s})$ which satisfies the LARC, i.e. $span(G^{BSS}) = T_x\mathcal{C}_0$. Then, there is no CIR of BSS which would be generated by smooth motion planning algorithm (Algorithm 6).
Proof: Lemma 1 displays that a smooth motion planning algorithm accomplished on BSS generates a sequence of involutive interiors

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1,\ldots,k_l)\} \tag{5.8}$$

where all $\underline{\Delta}_{i_j}(J_{k_j}) = g_p \quad 1 \leq j \leq l \quad g_p \in G^{BSS}$, i.e. every distribution in (5.8) is a single vector field. To reach a final point $x(T)$ from $x(0)$, a smooth motion planning based on flow sequence construction needs at least n flows generating a sequence of n involutive distributions ($n = dim(x)$). In fact, the sequence of n flows is sufficient only if the BSS is holonomic and nilpotent. In the case of nonholonomic system, smooth motion planning algorithm creates n independent vector fields from Philip Hall basis and generates the flows along

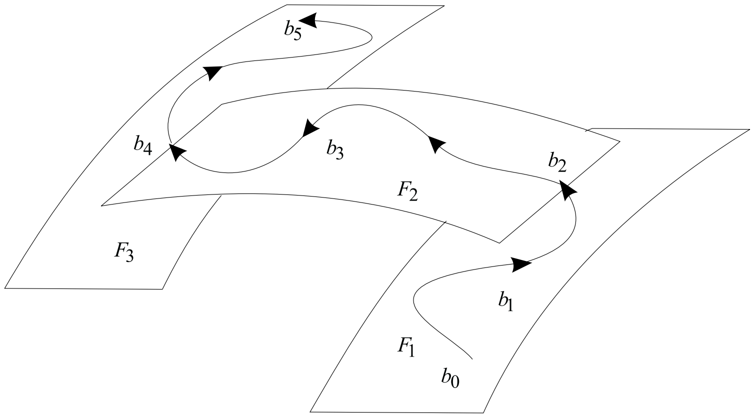

Figure 5.1: Reaching the final point via decomposed trajectory. The initial and final points are on different slices. The decomposed trajectory goes through a sequence of slices where the slices are joined. The initial point lays on the first slice, the final points lays on the last slice.

these vector fields. If the BSS is nonholonomic (which is the case) then some Philip Hall basis are composed by Lie bracket extensions. However, the flows along such a Lie bracket extension need a flow decomposition along single vector fields based on the approximation (A-33). This flow decomposition indicates flow extraction which should be built into the original flow sequence. Let this new flow sequence be associated with (5.8) from now. Based on the previous arguments, it should contain more than n flows. Obviously, such a sequence should define a sequence of involutive interiors (5.8) with more than n elements. Let the sequence of involutive interiors have l elements. Since, $l > n$ and $\underline{\Delta}_{i_j}(J_{k_j}) = g_p \quad 1 \leq j \leq l \quad g_p \in G^{BSS}$, the sequence in (5.8) should have two equal elements. Let $\underline{\Delta}_{i_p}(J_{k_p}) = g_p$ and $\underline{\Delta}_{i_q}(J_{k_q}) = g_q$ such that $g_p = g_q$. Then

$$core(\underline{\Delta}_{i_p}(J_{k_p}), \underline{\Delta}_{i_q}(J_{k_q})) = \{\emptyset\} \tag{5.9}$$

which contradicts to (CIR2.). Of course, there exists many combinations of flow sequences which solves the smooth motion planning problem for nonholonomic BSS, but using the previous arguments, each of them contains at least two equal elements causing the fail of (CIR2.) and proving the proposition.

Theorem 5. Consider a BSS which satisfies the LARC, i.e. $span(\Delta_1, \ldots, \Delta_s) = T_x\mathcal{C}_0$. A sequence of involutive interiors

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1,\ldots,k_l)\} \tag{5.10}$$

is a completely involutive representation of BSS if

(AS1.) Using the notation $\Delta_u = span(\underline{\Delta}_{i_1}(J_{k_1}), \ldots, \underline{\Delta}_{i_l}(J_{k_l}))$, it holds that $\dim(\Delta_u) = n$.

(AS2.) The distributions $\underline{\Delta}_{i_j}(J_{k_j}), \quad 1 \leq j \leq l$ are nonsingular.

(AS3.) There exist n independent vector fields g_p^{nil}, $p = 1 \ldots n$ such that

i. $g_p^{nil} \in gen(\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)}) \equiv \{gen(\underline{\Delta}_{i_1}(J_{k_1})), \ldots, gen(\underline{\Delta}_{i_l}(J_{k_l}))\}, \quad j = 1, \ldots, n$

ii. For every $j = 1, \ldots n$ there exists a p, $1 \leq p \leq l$ such that $g_p^{nil} \in core(\underline{\Delta}_{i_j}(J_{k_j}), \tilde{\underline{\Delta}}_{i_j}(J_{k_j}))$.

iii. $G^{nil} = (g_1^{nil}, \dots, g_n^{nil})$ generate a nilpotent Lie algebra.

Proof. First, it has to be shown that a sequence of involutive interiors satisfying (AS1.)-(AS3.) induces a completely involutive representation of BSS. One can directly see from the definition that (AS2.) satisfies explicitly (CIR1.) which is a necessary condition for completely involutive realization. Similarly, (AS3.)/(ii.) implies that (CIR2.) holds. As a consequence, one needs only to show that if (AS1.)-(AS3.) hold then there exists a trajectory between arbitrary $x(0) \in \mathcal{C}_0$ and $x(T) \in \mathcal{C}_0$, $T > 0$ which can be described by (5.7). To prove this constructively, one proceeds as follows.

Since (AS1.) holds, one can select exactly n independent vector fields from $gen(\underline{\Delta}_{I(l)}(J(k_1, \dots, k_l)))$. It would be able to provide arbitrary trajectory in $\mathcal{C}_0$ if each vector field had an own input. However this is not the case. It is required to look for a special trajectory between $x(0)$ and $x(T)$ whose tangent vector belongs to a prescribed involutive distribution as shown by (5.7). To construct such a trajectory, consider the involutive interiors of (5.5) in their defined order. Keeping the order and introducing the multiindices k_g one selects generators from $\underline{\Delta}_{i_j}(J_{k_j})$, $j = 1, \dots, l$ such that every distribution $\underline{\Delta}_{i_j}(J_{k_j})$ contains at least one generator. The assumption (AS2.)/(ii.) assures that choosing generators in such a way is possible. Let the candidates of n generators be selected as

$$(g_1^{nil}, \dots, g_n^{nil}) := (g_{i_1, J_1(k_1^g)(1)}, \dots, g_{i_1, J_1(k_1^g)(k_1^g)}, \dots, g_{i_l, J_l(k_l^g)(1)}, \dots, g_{i_l, J_l(k_l^g)(k_l^g)}) \tag{5.11}$$

Recall, that (AS1.) and (AS3.)/(i.) assure that there exist n generators chosen from generating vector fields of $\underline{\Delta}_{i_j}(J_{k_j})$, $j = 1, \dots, l$. It is possible now to execute the smooth motion planning algorithm (Algorithm 1). The resulting trajectory is obtained as a sequence of flows in the form

$$x(T) = \Phi(g_{i_l, J_l(k_l^g)(1)}, t_{l,k_l^g}) \circ \dots \circ \Phi(g_{i_l, J_l(k_l^g)(1)}, t_{l,1}) \circ \dots \circ \Phi(g_{i_1, J_1(k_1^g)(k_1^g)}, t_{1,k_1^g}) \circ \dots \circ \Phi(g_{i_1, J_1(k_1^g)(1)}, t_{1,1}) x(0) \tag{5.12}$$

The point $x(T)$ is precisely reached because (AS3.)/(iii.) claims that every vector fields in (5.12) is an element of nilpotent Lie algebra $\mathcal{L}(g_1^{nil}, \dots g_n^{nil})$. Moreover, it is easy to see that trajectory segment

$$x(t_{1,k_1^g}) = \Phi(g_{i_1, J_1(k_1^g)(k_1^g)}, t_{1,k_1^g}) \circ \dots \circ \Phi(g_{i_1, J_1(k_1^g)(1)}, t_{1,1}) x(0) \tag{5.13}$$

satisfies $\dot{x}(t) \in \underline{\Delta}_{i_1}(J_{k_1})$ when $0 \le t < t_{1,k_1^g}$ meaning if one chooses $t_1 = t_{1,k_1^g}$ then (5.7) holds in this time interval. Consider now $x(t_{1,k_1^g})$ the initial point in the next segment of the trajectory and take those flows which are related to the next involutive distribution, $\underline{\Delta}_{i_2}(J_{k_2})$. Obviously, these flows will keep the tangent vector of the trajectory on the integral submanifold of $\underline{\Delta}_{i_2}(J_{k_2})$. Eventually, if one defines $t_j = \sum_{k=1}^{j} \sum_{z=1}^{k_j^g} t_{z,k_j^g}$ as the ends of trajectory segments for all $j \le l$ then (5.7) holds for the whole time interval $0 \le t \le T$ and this proves the theorem.

Remark 13. Property (AS3)/(iii.) will be required to reach a final point precisely. However, it turns out that the method discussed here can be used to not nilpotent generators [1], as well. In this case, the solution to motion planning problem is an approximation.

[1]The nilpotency was required originally in smooth motion planning allowing a switching from $e^{\bar{h}_1 B_1 + \dots + \bar{h}_n B_n}$ to $e^{\bar{h}_1 B_1} \dots e^{\bar{h}_n B_n}$

Theorem 5 gives sufficient conditions to find CIRs. However, it does not assert that all the CIRs can be produced in this way. All the CIRs produced throughout this chapter rely on the conditions of Theorem 5. Therefore, any time when a CIR is constructed, one may assume that it has generators defined by (AS3). The only exception which is not always required is the nilpotency. Algorithm devised below works even if generated Lie algebra is not nilpotent but it provides only an approximation in this case (see also Section 5.5).

Definition 27. Assume a CIR generated by Theorem 5. The vector fields of g_p^{nil}, $p = 1 \ldots n$ in (AS3.) are said to be a generator set of the CIR (5.10).

Definition 28. Let us consider n generators g_p^{nil}, $p = 1 \ldots n$ of the CIR (5.10) obtained by Theorem 5. Let k_j, $j = 1, \ldots, n$ denote the number of generators in $\underline{\Delta}_{i_j}(J_{k_j})$. Then, the vector $(k_1^g, \ldots, k_l^g)$ is said to be the *structure vector* of CIR (5.10)

Proposition 9. There exists a completely involutive representation of BSS if it is possible to find $n = dim(x)$ independent, nonsingular generators from BSS which generate a nilpotent Lie algebra.

Proof: Let the n independent generators be denoted by g_p^{nil}, $\quad p = 1, \ldots, n$. Since, g_p^{nil} is a generator, i.e.

$$g_p^{nil} \in \{gen(\Delta_1), \ldots, gen(\Delta_s)\} \tag{5.14}$$

It means that any candidate for CIR which includes g_p^{nil} satisfies (AS3)/(i.). By assumption, g_p^{nil}, $\quad p = 1, \ldots, n$ are n independent vector fields generating a nilpotent Lie algebra, hence they fulfill (AS3.)/(iii.). Let every g_p^{nil}, $\quad p = 1, \ldots, n$ be considered now as a distribution $\underline{\Delta}_{i_p}(J_{k_p})$. Then, it leads to n nonsingular distribution complying with (AS2.) and (AS1.) in Theorem 5. Since $\underline{\Delta}_{i_p}(J_{k_p})$ is involutive,

$$\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} \equiv \underline{\Delta}_{I(l)}\{J(k_1, \ldots, k_l)\} = \{g_1^{nil}, \ldots, g_n^{nil}\} \tag{5.15}$$

points out a candidate for being a CIR of BSS where $l = n$. The only assertion remained to be shown is (AS3.)/(ii.) i.e. that for every $j = 1, \ldots, n$ there exists a p, $1 \leq p \leq l$ such that $g_p \in core(\underline{\Delta}_{i_j}(J_{k_j}), \tilde{\underline{\Delta}}_{i_j}(J_{k_j}))$. It is easy to show because, by assumption, g_p^{nil}, $p = 1, \ldots, l$ are independent. It means, by definition, that every $\underline{\Delta}_{i_p}(J_{k_p}) = g_p^{nil}$ is also a vector field which cannot be found in the other involutive interiors $\underline{\Delta}_{i_j}(J_{k_j})$, $j = 1, \ldots, l$, $j \neq p$. It deduces that $g_p^{nil} \in core(\underline{\Delta}_{i_p}(J_{k_p}), \tilde{\underline{\Delta}}_{i_p}(J_{k_p}))$ resulting that $g_p = g_p^{nil}$. Observe now, that all the necessary conditions in Theorem 5 hold i.e. the sequence of involutive interiors (5.15) realizes a CIR of BSS.

Remark 14. The CIR defined by Proposition 9 provides actually the base for the smooth motion planning operating with sequence of flows. The aim of new approach is to extend it to a motion on integral submanifolds with higher dimension.

Corollary 4. Consider a CIR of BSS described by $\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1, \ldots, k_l)\}$. Let $n = dim(\mathcal{C}_0)$. Then, it holds that $l \leq n$.

Proof Assume, there exists a sequence of involutive interiors according to (5.10) where $l > n$. Then, the (CIR2.) of Definition 26 entails that for every $j = 1, \ldots l$, there exists a p, $1 \leq p \leq l$ such that $g_p \in$

$core(\underline{\Delta}_{i_j}(J_{k_j}), \tilde{\underline{\Delta}}_{i_j}(J_{k_j}))$. In other words, every g_p should be independent from all the generators of $\underline{\Delta}_{i_j}(J_{k_j})$, $j = 1, \ldots, l$, $j \neq p$. But it is not possible because there are only n independent vector fields in a BSS having n dimension. This contradicts to the assumption.

Remark 15. It is easy too see that there may exist a CIR of BSS where $l < n$. In this case, the n generators are divided into fewer than n involutive interiors involving that one can find at least two g_p and g_q which are the part of the generators of CIR and $g_p, g_q \in \underline{\Delta}_{i_j}(J_{k_j})$, $1 \leq j \leq l$ i.e. they are in the same involutive interior. In fact this phenomena provides a benefit in comparison with the earlier stratified motion planning algorithm where the system can move only on an integral submanifold with one dimension.

Remark 16. Based on Proposition 9, if one can find $n = dim(x)$ independent, nonsingular generators from BSS which generate a nilpotent Lie algebra, then it always exists a completely involutive representation of BSS. However, Remark 15 shows that another CIR of the same BSS may exist by redefining involutive subdistributions $\underline{\Delta}_{i_j}(J_{k_j})$. Obviously, it may be possible if it is possible to find an involutive interior $\Delta_{i_j}(J_{k_j})$ which contains at least two generators.

Definition 29. The configurations

$$\begin{aligned}
b_1 &= \Phi(g_{i_1, J_1(k_1^g)(k_1^g)}, t_{1,k_1^g}) \circ \cdots \circ \Phi(g_{i_1, J_1(k_1^g)(1)}, t_{1,1}) x(0) \\
b_2 &= \Phi(g_{i_2, J_2(k_2^g)(k_2^g)}, t_{2,k_2^g}) \circ \cdots \circ \Phi(g_{i_2, J_2(k_2^g)(1)}, t_{2,1}) b_1 \\
&\vdots \\
b_l &= x(T) = \Phi(g_{i_l, J_l(k_l^g)(k_l^g)}, t_{l,k_l^g}) \circ \cdots \circ \Phi(g_{i_l, J_l(k_l^g)(1)}, t_{l,1}) b_{l-1}
\end{aligned} \tag{5.16}$$

established in the proof of Theorem 5 are said to be *the basis points of involutive representation*. Between any two neighboring basis points the system moves on an integral submanifold of $\mathcal{C}_0$. The integral submanifold determined by the actual involutive distribution $\underline{\Delta}_{i_j}(J_{k_j})$,$1 \leq j \leq l$ will be denoted by F_{i_j}.

Remark 17. The definition of basis points intends to reveal a posterior idea of trajectory planning saying that all the regarded trajectories will contain these points for a given CIR.

Theorem 6. A completely involutive representation of BSS can be steered along arbitrary *decomposed* trajectory between $x(0)$ and $x(T)$ if the decomposition (CIR3.) is defined by the involutive interiors of the CIR.

Proof: Every completely involutive representation (CIR) $\underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1, \ldots, k_l)\}$ induces l basis points, $b_1, \ldots, b_l$, as it was shown in Theorem 5. Actually, these basis points fit a sequence $\{F_{i_1}, \ldots, F_{i_l}\}$ of slices in $\mathcal{C}_0$ to each other such that $x(0), b_1 \in F_{i_1}$; $b_1, b_2 \in F_{i_2}; \ldots; b_{i_{l-1}}, b_l \in F_{i_l}$. It is easy to see now that keeping the basis points, the CIR is able to follow any trajectory on the actual integral submanifold F_{i_j} because this integral submanifold is defined exactly by the control vector fields $gen(\underline{\Delta}_{i_j}(J_{k_j}))$,$1 \leq j \leq l$ and, by definition of CIR, only these vector fields are allowed on this slice from BSS. Naturally, if trajectory reaches a new bases point then the control vector fields should be changed according to the new slice and involutive distribution. Changing does not cause difficulties because the system is stratified and every vector field of the actual involutive subdistribution on a slice F_{i_j} is derived from the same stratum (S_i). A trajectory pieced together such a way is obviously able to achieve arbitrary decomposed trajectory constrained only by basis points.

Remark 18. Now, it is easy to reveal the benefits of the concept. If the distribution of BSS is involutive, one may find a sequence of involutive interiors from strata, as well. The involutive interiors define integral submanifolds in $\mathcal{C}_0$. In fortunate case, the dimensions of the integral submanifolds are greater than 1 which increases the freedom for prescribing reference trajectory to the system. Observe, the worst case occurs at the conventional smooth motion planning (using sequence of flows) where the BSS moves along only single vector fields. Here, one could prescribe only the initial and final points for motion planning algorithm . On contrary, if the BSS would be represented by a single stratum and would be holonomic, then all the n vector fields are available for motion planning at the same time. In fact, this situation would turn the stratified system into a smooth system. The concept discussed in this section represents the case between the two extreme situations. The extended dimensions allow to prescribe trajectories on the integral submanifolds but the stratifaction prevents to prescribe arbitrary trajectories in the whole configuration space. The method is able to lead the system along decomposed trajectories.

5.2 Optimal Completely Involutive Representations

Remark 16 in Section 5.1 disclosed that more than one CIR may exist. The question arises, which of them yields the best solution to motion planning. As known from the previous section, decomposed trajectory has to satisfy some conditions which constraints the trajectory planning. A good representation should provide greater degrees of freedom in planning. This section is devoted to investigate optimal representations in this sense. A CIR implies two restrictions in trajectory planning:

1. Each bases point represents a constraint since trajectory has to go through it.
2. Each slice (integral submanifold) F_{i_j} narrows the dimension of $\mathcal{C}_0$.

Different CIRs influence the restrictions in different ways. This section is devoted to answer the question how to find an optimal CIR from these two points of view.

Definition 30. A CIR of BSS is said to be an optimal basis representation (OBR) if there is no another CIR having fewer basis points.

Proposition 10. Let $\underline{\Delta}(k_1,\ldots,k_l) = \underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1,\ldots,k_l)\}$ describe a CIR in general. Then, a CIR

$$\underline{\Delta}^{obr}(k_1,\ldots,k_L) = \underline{\Delta}_{I_{obr}(L)}\{J_{obr}(k_1,\ldots,k_L)\} = \{\underline{\Delta}_{i_1}^{obr}(J_{k_1}),\ldots,\underline{\Delta}_{i_L}^{obr}(J_{k_L})\} \tag{5.17}$$

is an OBR if there exists a sequence of subdistributions $\{\Delta_{i_1}^{core}(J_{k_1}),\ldots,\Delta_{i_L}^{core}(J_{k_L})\}$ such that

(OBR1.) $\Delta_{i_j}^{core}(J_{k_j}) \subset \underline{\Delta}_{i_j}^{obr}(J_{k_j})$ for all $j = 1,\ldots,L$.

(OBR2.) $rank(\Delta_{i_j}^{core}(J_{k_j})) + rank(\Delta_{i_p}^{core}(J_{k_p})) = rank([\Delta_{i_j}^{core}(J_{k_j}), \Delta_{i_p}^{core}(J_{k_p})])$ for all (j,p), $j = 1,\ldots,L$, $p = 1,\ldots,L$, $j \neq p$.

(OBR3.) $(I_{obr}(L), J_{obr}(k_1,\ldots,k_L)) \in \arg\min_{(I(l),J(k_1,\ldots,k_l))}(l)$.

(OBR4.) $\sum_{j=1}^{L} dim(\Delta_{i_j}^{core}(J_{k_j})) = n$, $n = dim(\mathcal{C}_0)$.

Proof: Consider a candidate $\underline{\Delta}^{obr}(k_1,\ldots,k_L)$. It will be shown if this CIR satisfies (OBR1.)-(OBR4.) then it is an optimal basis representation. To prove this, one proceeds as follows. Based on (OBR1.) it is possible to select subdistributions $\Delta_{i_j}^{core}(J_{k_j})$ from every $\underline{\Delta}_{i_j}^{obr}(J_{k_j})$ which represent cores below. Of course, $\Delta_{i_j}^{core}(J_{k_j})$ does not need to be involutive, however according to (OBR2.) they have to be independent from each other. Moreover, since (OBR4.) holds, they have to span the tangent space of BSS disjointedly. It is possible to select the cores in this way because Lemma 2 assures that involutive interiors of CIR span the whole tangent space. Obviously, independent cores provide a minimal number of vector fields needed to span the entire tangent space and none of them can be omitted. Since exactly one independent core is selected from each involutive interiors from (5.17) and the number of involutive interiors equals to the number of the basis points, it can can concluded that one achieves the minimal number of basis points if $\underline{\Delta}^{obr}(k_1,\ldots,k_L)$ contains minimal number of involutive interiors. Assertion (OBR3.) holds this condition and guarantees that there is no $\underline{\Delta}(k_1,\ldots,k_l)$ that would produce $l < L$ i.e. $\underline{\Delta}^{obr}(k_1,\ldots,k_L)$ is an OBR with minimal number of basis points L.

Definition 31. A CIR $\underline{\Delta}_{I_{odr}(l)}\{J_{odr}(k_1,\ldots,k_l)\}$ is said to be an optimal dimension representation (ODR) at length l if there is no CIR $\underline{\Delta}_{I(l)}\{J(k_1,\ldots,k_l)\}$ which would put into use more vector fields than

$$\underline{\Delta}_{I_{odr}(l)}\{J_{odr}(k_1,\ldots,k_l)\}$$

(with the same length l).

Remark 19. One can intuitively see that such a representation "maximize" the dimension of the integral submanifolds F_{i_j}.

Proposition 11. Let $\underline{\Delta}(k_1,\ldots,k_l) = \underline{\Delta}_{I(l)}^{J(k_1,\ldots,k_l)} = \underline{\Delta}_{I(l)}\{J(k_1,\ldots,k_l)\}$ describe a CIR in general. Then, a CIR $\underline{\Delta}^{odr}(k_1,\ldots,k_l) = \underline{\Delta}_{I_{odr}(l)}\{J_{odr}(k_1,\ldots,k_l)\} = \{\underline{\Delta}_{i_1}^{odr}(J_{k_1}),\ldots,\underline{\Delta}_{i_l}^{odr}(J_{k_l})\}$ is an ODR at length l if

ODR1. $(I_{odr}(l), J_{odr}(k_1,\ldots,k_l)) \in \arg\max_{(I(l),J(k_1,\ldots,k_l))}(\sum_{j=1}^{l} dim(\underline{\Delta}_{i_j}(J_{k_j}))$

Proof: Show at first that one can not omit any $\underline{\Delta}_{i_j}^{odr}(J_{k_j})$, $j = 1\ldots l$ meaning that the representation is not redundant. It can be seen if one recalls from the definition of CIR that every involutive interiors $\underline{\Delta}_{i_j}^{odr}(J_{k_j})$, $j = 1\ldots l$ has to have an independent core in respect to the others. It deduces that $dim(\tilde{\underline{\Delta}}_{i_j}(J_{odr}(k_j))) < n$, $n = dim(\mathcal{C}_0)$ for all $j = 1,\ldots l$. As a result, $\tilde{\underline{\Delta}}_{i_j}(J_{odr}(k_j)) = \underline{\Delta}^{odr}(k_1,\ldots,k_l) \setminus \underline{\Delta}_{i_j}(J_{odr}(k_j))$ would not satisfy LARC which according to Lemma 2 is a property of CIR i.e. the representation is not redundant.

One can also observe that every vector field in $gen(\underline{\Delta}_{i_j}^{odr}(J_{k_j}))$ has a contribution to the dimension of $\underline{\Delta}_{i_j}^{odr}(J_{k_j})$. It results that keeping all the involutive interiors, the CIR contains the most vector fields if the sum of the dimension of the involutive interiors $\underline{\Delta}_{i_j}^{odr}(J_{k_j})$ provides the highest possible number. The (ODR1.) guarantees this fact proving the proposition.

Remark 20. OBRs and ODRs can be obtained by testing all the CIR of BSS. It can be done because the discussion deals with finite dimensions implying finite number of CIRs.

Proposition 10 facilitates finding an OBR with the minimal basis points and Proposition 11 assigns an ODR at each possible length l where obviously $l \leq n$. There exist different optimal CIRs from different points of views. A question arises, which representation yields the best solution. Of course, it depends on the actual task specification. A heuristic technique is proposed now which endeavors to consider both viewpoints.

Algorithm 10. (Heuristic selection for optimal CIR)

Step 1. Create involutive interiors in strata.

Step 2. Create sequence of involutive interiors.

Step 3. Collect all those CIRs $\underline{\Delta}(k_1, \dots, k_l)$ of BSS which satisfy (AS1-AS3) of Theorem 5. If there is no sequence of involutive interiors that satisfy nilpotency, then "assume" that sequence of involutive interiors satisfy this property and they are CIRs. This assumption results, the motion planning will be only an approximation.

Step 4. Obtain the set of OBRs $\underline{\Delta}_{I_{obr}(L)}\{J_{obr}(k_1, \dots, k_L)\}$. Denote this set by S_{obr}.

Step 5. It may happen that a CIR does not exist for some l and then ODR does not exist at those l either. Then, let $l_{odr} = \{l :\ni \underline{\Delta}_{I_{odr}(l)}\{J_{odr}(k_1, \dots, k_l)\}\}$.

Step 6. Obtain the set of ODRs $\underline{\Delta}_{I_{odr}(l)}\{J_{odr}(k_1, \dots, k_l)\}$ for all $l \in l_{odr}$. Denote this set by S_{odr}

Step 7. Let R_o be the set of all ODRs and OBRs, i.e

$$R_o = S_{obr} \cup S_{odr} \tag{5.18}$$

Step 8. Let m denote the number of vector fields in a CIR and let b denote the number of involutive interiors in the same CIR. Then compute the m_i/l_i for every $\underline{\Delta}_i(k_1, \dots, k_l) \in R_o$.

Step 9. The heuristic candidate for optimal CIR is

$$\underline{\Delta}^{opt}(k_1, \dots, k_l) = \arg \max_{\underline{\Delta}_i(k_1, \dots, k_l) \in R_o} \frac{m_i}{l_i}. \tag{5.19}$$

To generate a CIR on the foundation of Theorem 5, one should obtain the n generators $g_1^{nil}, \dots g_n^{nil}$ at first. This n generators define a structure vector $(k_1^g, \dots, k_l^g)$ depending on how many generators each involutive interior $\Delta_{i_j}^g(J_{k_j})$ contains. (Notice that $\Delta_{i_j}^g(J_{k_j})$ is spanned by k_j^g vector fields from $g_1^{nil}, \dots g_n^{nil}$). Obviously, $\sum_{j=1}^{l} k_j^g = n$. At this stage every $\Delta_{i_j}^g(J_{k_j})$ consists of generators. Now, the question arises how one can extend the n generators to a CIR having the maximal number of vector fields. For this, recall that by every extension, the added vector fields $G^a = (g_{i_j}^{a,1}, \dots, g_{i_j}^{a,r})$, $g_{i_j}^{a,p} \in \Delta_{i_j}$, $p = 1, \dots, r$ have to be invariant under the composition of the original distribution $\Delta_{i_j}^g(J_{k_j})$ and the distribution $\underline{\Delta}_{i_j}^a(J_{k_j}^a) = span(G^a)$. In other words, the new distribution $\underline{\Delta}_{i_j}(J_{k_j}) := \{\underline{\Delta}_{i_j}^a(J_{k_j}^a), \underline{\Delta}_{i_j}(J_{k_j})\}$ is involutive. Observe, that maximally $n - (l-1) - k_j^g$ vector fields can be added from Δ_{i_j} to the initial distribution $\Delta_{i_j}^g(J_{k_j})$. To see this, start from the fact that a distribution can not include more than n independent vector fields. However, the $\Delta_{i_j}^g(J_{k_j})$ contained already k_j^g vector fields so the number of possible extra vector fields which can be added to $\Delta_{i_j}^g(J_{k_j})$ is to be modified to $n - k_j^g$. One should still decrease this number by $l-1$ because the existence of independent core of $\underline{\Delta}_{i_j}(J_{k_j})$ have to be ensured with respect to the other distributions $\underline{\Delta}_{i_p}(J_{k_p})$, $p \neq j$. It is possible only if all the l involutive interiors contain at least one vector field which is independent from all the vector fields in all the other involutive interiors. To determine the maximal number of added vector fields to generators, a sum needs to be evaluated as

$$\begin{aligned} n_{max}^{add}(k_1^g, \dots, k_l^g) &= \sum_{j=1}^{l}(n-(l-1)-k_j^g = ln - l(l-1) - (k_1^g + \cdots + k_l^g) = ln - l(l-1) - n \\ &= n(l-1) - l(l-1) = (n-l)(l-1) \end{aligned} \tag{5.20}$$

Now, the maximal possible number of vector fields in a CIR realized by l involutive interiors in n dimension configuration space of BSS is obtained if one adds n (number of generator) to the maximal (possible) number of extra vector fields, i.e.

$$n_{max}(k_1^g, \dots, k_l^g) = n + (n-l)(l-1) = n + ln - l^2 + l - n = ln - l^2 + l = l(n-l+1) \tag{5.21}$$

Observe now, that m_i in (5.19) is less than or equal to $n_{max}(k_1^g, \dots, k_l^g)$ for each CIR of BSS. It deduces that the maximal values of (5.19)

$$\max \frac{m_i}{l_i} \leq \max_l \frac{l(n-l+1)}{l} \equiv \max_l (n-l+1) \tag{5.22}$$

which occurs in ideal case at the minimal number of $l = f(k_1^g, \dots, k_l^g)$. Similarly, one can obtain the least possible value of l for a given n and m number of vector fields from the equation $m = l(n-l+1)$ which leads to $l = ceil(0.5(n+1+\sqrt{(n+1)^2 - 4(n+1)m}))$. (The function *ceil* symbolizes the round to the nearest integer towards infinity.)

Remark 21. It is easy to see that (5.22) provides the best values if $l = 1$ when n control vector fields can be inserted into the CIR. Of course, the problem is that the existence of such a CIR is not guaranteed. The case of $l = 1$ occurs if the stratified system consists of only one stratum degenerating the original system into a holonomic, smooth (not stratified) one. For this, the motion planning problem is not a big challenge and would make the whole stratified approach unnecessary.

5.3 Optimal Set of Basis Points

In general, one desires to follow an arbitrary trajectory in $\mathcal{C}_0$. However, as it was shown, to carry out this task is typically not possible. If one has a desired trajectory $x_d(t)$ in the configuration space, the resulting trajectory is desired to follow it as precisely as possible. In general, exact tracking is not possible for stratified systems but there exists a method presented previously that increases the dimension of the slices on which the system moves. At this point, taking into account some added considerations due to the basis points, there is unfortunately no choice to prescribe the basis point $b_1, \dots, b_k$ in arbitrary way. However, a freedom to apply an optimal set of basis points connecting subtrajectories on slices arises in a sense, since the actual set of basis points can be modified by the permutation of the involutive interiors. Consequently, the permutation of basis points will be understood as the set of the basis points at a given permutation of the distribution in the sequel. To begin the discussion, define the integrated distance

$$D(x(t), x_d(t)) = \int_0^t \sqrt{(x(s) - x_d(s))^T (x(s) - x_d(s))} ds \tag{5.23}$$

between two trajectories. The distance is easily computed numerically.

Definition 32. Let $\underline{\Delta} = span\{g_1, \dots g_k\}$ be an involutive distribution with the integral submanifold $S_{\underline{\Delta}}$ i.e. $g_i \in T_x S_{\underline{\Delta}} \subset T_x \mathcal{C}_0$. Then, $\Phi(\underline{\Delta}, t)(x(0))$ denotes the set of all possible flows on $S_{\underline{\Delta}}$ from $x(0)$ to $x(T)$ where $\dot{x}(t) \in \underline{\Delta}$ for every $0 < t \leq T$.

In fact, $\Phi(\underline{\Delta}, t)(x(0))$ defines all the integral curve in $S_{\underline{\Delta}}$. Let π denote a permutation of $(1, \dots, l)$ and let a permutation of CIR be

$$\underline{\Delta}_{\pi(I(l))}^{J(k_{\pi(1)}, \dots, k_{\pi(l)})} = \underline{\Delta}_{\pi(I(l))}(J(k_{\pi(1)}, \dots, k_{\pi(l)})) = \{\underline{\Delta}_{i_{\pi(1)}}(J_{k_{\pi(1)}}), \dots, \underline{\Delta}_{i_{\pi(l)}}(J_{k_{\pi(l)}})\}. \tag{5.24}$$

Based on the previous discussions (Algorithm 1 and Algorithm 6), solving motion planning as flow sequences by different permutations of strata results the expression

$$x_{\pi,F} = \Phi(\underline{\Delta}_{\pi(I(l))}^{J(k_{\pi(1)}, \dots, k_{\pi(l)}}, t) x_I = \Phi(\underline{\Delta}_{i_{\pi(l)}}(J_{k_{\pi(l)}}), t_l) \circ \cdots \circ \Phi(\underline{\Delta}_{i_{\pi(1)}}(J_{k_{\pi(1)}}), t_1)(x_I) \tag{5.25}$$

Let b_π be the set of l basis points obtained from (5.25) for π i.e. $b_\pi = (b_\pi(1), \dots, b_\pi(l))$. Let $x_d(t)$ be a desired trajectory. Then, $D(x_\pi(t), x_d(t))$, where

$$x_\pi(t) = \Phi(\underline{\Delta}_{\pi(I(l))}^{J(k_{\pi(1)}, \dots, k_{\pi(l)}}, t) x(0) \tag{5.26}$$

defines the integrated distance between the desired and real trajectories. Notation $x_\pi(t)$ represents the decomposed trajectories at the strata permutation π, and considering the minimal error between two trajectories, the optimal permutation of the strata is $\pi(o) = \arg\min_\pi D(x_\pi(t), x_d(t))$. It means that one should search the actual trajectory in the form

$$x_{\pi(o),F} = \Phi(\underline{\Delta}_{\pi(o)(I(l))}^{J(k_{\pi(o)(1)}, \dots, k_{\pi(o)(l)}}, t) x_I = \Phi(\underline{\Delta}_{i_{\pi(o)(l)}}(J_{k_{\pi(o)(l)}}), t_l) \circ \cdots \circ \Phi(\underline{\Delta}_{i_{\pi(o)(1)}}(J_{k_{\pi(o)(1)}}), t_1)(x_I) \tag{5.27}$$

It is not so easy to compute the optimal permutation of strata in (5.27) because $x_\pi(t)$ denotes all the decomposed trajectories at π. To simplify the computations, the distance definition

$$D(x_p, x_q) = \sqrt{(x_p - x_q)^T (x_p - x_q)} \tag{5.28}$$

provides benefits instead of $D(x(t), x_d(t))$. In this case, the summarized distance is defined by

$$D(b_\pi, x_d(t)) = \sum_{i=1}^{l} D(b_\pi(i), x_d(\sum_{j=1}^{i} t_j)) \tag{5.29}$$

replacing $D(x_\pi(t), x_d(t))$, providing the advantage that it points out only one distance to each permutation independently from the many possible decomposed trajectories at this permutation π. Finding the minimum among these distances is much easier.

Remark 22. To make a heuristically most advantageous optimization, $\underline{\Delta}_{\pi(I(l))}^{J(k_{\pi(1)},\ldots,k_{\pi(l)})}$ should be the permutation of the heuristic optimal CIR $\underline{\Delta}^{opt}(k_1, \ldots, k_l)$.

5.4 Reference Trajectory Generation

It is clear from the above discussion that steering a stratified controllable system from $x(0)$ to $x(T)$ via decomposed trajectory is possible by using an optimal CIR. The representation sets apart the bottom stratum into slices, each of them defined by involutive interiors. However, the trajectory planning on an integral submanifold was not discussed. This section endeavors to answer this question by using the proof of Frobenius theorem in Isidori (1996).

Let an involutive distribution $\Delta = span(g_1, \ldots g_d)$ be considered in the n dimensional configuration space with the property that $d < n$. (Every vector field is defined in a neighborhood U of a point x_0.) Let this distribution be extended by a set of vector fields $g_{d+1}, \ldots, g_n$ such that

$$span(g_1, \ldots g_d, g_{d+1}, \ldots, g_n) = \mathbb{R}^n. \tag{5.30}$$

As before, let $\Phi(g, t)x$ denote a flow along the vector field g during a time interval t. The proof of Frobenius theorem (Theorem 1) obtains functions $\lambda_1, \ldots, \lambda_{n-d}$, the solution of the partial differential equation

$$\frac{\partial \lambda_j}{\partial x}(g_1(x) \ldots g_d(x)) = 0 \tag{5.31}$$

that can be produced by means the flow sequence

$$\Phi(g_1, t_1) \circ \ldots \circ \Phi(g_n, t_n)x. \tag{5.32}$$

The solution relies on a mapping

$$\begin{aligned} \Psi \; : \quad U_\epsilon \quad &\to \quad \mathbb{R}^n \\ (z_1, \ldots z_n) \quad &\to \quad \Phi(g_1, z_1) \circ \ldots \circ \Phi(g_n, z_n)x_0 \end{aligned} \tag{5.33}$$

where $U_\epsilon = \{\in \mathbb{R}^n : |z_i| < \epsilon\}$ for small ϵ. The transformation Ψ provides benefits for trajectory planning. The Ψ can be obtained by the symbolic approximation of all the $\Phi(g, t)$ in the form[2]

$$\Phi(g, t)x = e^{g(x)t} = \sum_{k=1}^{\infty} \frac{L_g^k(x)t^k}{k!}. \tag{5.34}$$

Recall, that based on the discussion above, joining slices were obtained whose intersections define basis points. The basis points can be acquired by smooth motion planning algorithm, and it was also proved that a BSS can

[2]There exists a much more general form of (5.34) allowing to take the exponentials of complete Lie series instead of the "identity" x as was taken here. For this, the reader may be referred to Lafferriere and Sussmann (1991) and Sussmann and Liu (1993)

be steered between any two points through a decomposed trajectory. However, it is still an open question how a trajectory can be prescribed in a slice. Now, suppose a slice F_{i_j} of an integral submanifold of $\underline{\Delta}_{i_j}(J_{k_j})$ on which the initial point $x(t_{j-1})$ and final point $x(t_j)$ are lying, respectively. A trajectory between them in F_{i_j} is to be prescribed. The next algorithm is proposed for this.

Algorithm 11. (Trajectory generation in a slice)

Step 1. Transform $x(t_{j-1})$ and $x(t_j)$ into $z(t_{j-1}) = \Phi(x(t_{j-1}))$ and $z(t_j) = \Phi(x(t_j))$. (The last $n-d$ coordinates will become the same from both the vectors).

Step 2. Transform all the vector fields $g_k \in G = gen(\underline{\Delta}_{i_j}(J_{k_j}))$ into

$$\bar{g_k} = \left[\frac{\partial \Psi}{\partial x} g_k(x)\right]_{x=\Phi^{-1}(z)} = (\bar{g}_{k,1}, \ldots, \bar{g}_{k,d}, 0 \ldots, 0)^T. \tag{5.35}$$

(Transformation Ψ converts a slice into a hyperplane.)

Step 3. Omit the last $n-d$ coordinates of all the transformed vectors, i.e.

$$\begin{aligned} \hat{z}(t_j) &= (z_1(t_j), \ldots, z_d(t_j))^T \\ \hat{z}(t_{j-1}) &= (z_1(t_{j-1}), \ldots, z_d(t_{j-1}))^T \\ \hat{g}_k &= (g_{k,1}, \ldots, g_{k,d})^T \\ \hat{G} &= [\hat{g}_1, \ldots, \hat{g}_d] \end{aligned} \tag{5.36}$$

Step 4. Observe, that $\hat{G}$ composed by d independent vector fields, each of them having d dimension. It deduces that there exists inverse of $\hat{G}$ for all $\hat{z} \in \mathbb{R}^d$. Define the system

$$\hat{\Sigma}: \ \hat{G}(\hat{z})u_z = \hat{g}_1(\hat{z})u_1^z + \cdots, \hat{g}_d(\hat{z})u_d^z \tag{5.37}$$

which is a holonomic one.

Step 5. For arbitrary trajectory in the state space of $\hat{\Sigma}$, the input u_z can be found that steers $\hat{\Sigma}$ from $\hat{z}(t_{j-1})$ to $\hat{z}(t_j)$ on the reduced d dimensional integral submanifold F_{i_j}. For, supposing the desired trajectory in $\hat{z}$ is given by $\hat{z}(t)$, then $u_z = \hat{G}(\hat{z})^{-1}\dot{\hat{z}}(t)$.

Step 6. Since (5.35) is a linear transformation for every vector field at each x, the principle of superposition is valid implying that $u = u_z$ steers also

$$\Sigma: \ G(z)u = g_1(z)u_1 + \cdots + g_d(z)u_d \tag{5.38}$$

from $x(t_{j-1})$ to $x(t_j)$ on the slice F_j.

Step 7. Transform trajectory $z(t)$ back using the inverse of (5.35).

Remark 23. Algorithm 11 is able to avoid obstacles. For, one just should transform the points of obstacles into z and plan a trajectory which simply avoids these points. To compute appropriate u is easy since only a multiplication of inverse matrix is needed (at every trajectory point).

Finally, combining algorithms one may form a unique frame of holonomic stratified motion planning.

Algorithm 12. (decomposed holonomic stratified motion planning algorithm)

Step 1. Create maximally redundant BSS (inserting all the possible moving on vector fields from subsystems).

Step 2. Create optimal CIR of BSS using Algorithm 10.

Step 3. Obtain basis points for optimal CIR with smooth motion planning algorithm (Algorithm 1) and optimize the basis points by permutation in accordance with Section 5.3.

Step 4. Design a decomposed trajectory and obtain the input for it by means of Algorithm 11 between two basis points x_j, x_{j+1} on every slice F_{j+1}, $j = 1, \ldots, n-1$.

Step 5. Inserting moving off vector fields by adopting the corresponding step from stratified motion planning algorithm (Algorithm 6)

5.5 Example

This section intends to demonstrate the method on the modified example of hexapod robot. As before, x, y denote the position of the robot in the plane, θ is the orientation, ϕ_1 and ϕ_2 are the angles of the leg groups. The heights of the leg groups from the ground are denoted by h_1 and h_2. The equations of motion in strata are given as follows.

S_{12}, all legs on the ground:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{bmatrix} -2sin(\theta) & 2cos(\theta) \\ 2cos(\theta) & 2sin(\theta) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{bmatrix} g_{12,1} & g_{12,2} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \tag{5.39}$$

In fact, a modification is placed here because $g_{12,1}$ and $g_{12,2}$ are substituted by the Lie brackets of original vector fields from Example 6 in Subsection 2.2.1. Denote the vector fields $g_{12,1}$ and $g_{12,2}$ in Example 6 by $g^o_{12,1}$ and $g^o_{12,2}$, respectively. Then, the new vector fields are defined as $g_{12,1} = [g^o_{12,1}, g^o_{12,2}]$ and $g_{12,2} = [g^o_{12,2}, [g^o_{12,1}, g^o_{12,2}]]$.

On stratum S_1, the leg group 1 is on the ground.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_2 \end{pmatrix} = \begin{bmatrix} cos(\theta) & 0 & 0 \\ sin(\theta) & 0 & 0 \\ l & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_4 \end{pmatrix} = \begin{bmatrix} g_{1,1} & g_{1,2} & g_{1,3} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_4 \end{pmatrix} \tag{5.40}$$

On stratum S_2, the leg group 2 is on the ground.

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{h}_1 \end{pmatrix} = \begin{bmatrix} cos(\theta) & 0 & 0 \\ sin(\theta) & 0 & 0 \\ -l & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{bmatrix} g_{2,1} & g_{2,2} & g_{2,3} \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \tag{5.41}$$

Follow now the steps of the Algorithm 12 to perform motion planning.

Step 1. (Maximal redundant BSS) The distributions of the (maximally redundant) bottom stratified system (BSS) which contains all the possible moving on vector fields satisfying the decoupling condition and the strata with new indices are

$$\Delta^{BSS} = span\{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}\} \tag{5.42}$$

$$\Delta_1 = span\{g_{12,1}, g_{12,2}\} \tag{5.43}$$

$$\Delta_2 = span\{g_{1,1}, g_{1,2}\} \tag{5.44}$$

$$\Delta_3 = span\{g_{2,1}, g_{2,2}\} \tag{5.45}$$

where Δ_1 represents S_{12}, Δ_2 corresponds to S_1 and Δ_3 is related to S_2. From now, every (moving on) vector field is considered in the BSS i.e. $g_{1,1}\ |_{S_{12}}$ would precisely be the correct notation instead of $g_{1,1}$, however, for the sake of better understanding, $|_{S_{12}}$ will be omitted in the future.

Step 2. (Create optimal CIR) It is not checked now whether the involutive sequences satisfy CIR. A sequence of involutive interiors is considered a CIR even if its vector fields do not generate nilpotent Lie Algebra. It makes sense to do so since Lie brackets are approximated only up to order two with Cambell-Baker-Hausdorff formula, thus the solution is an approximation, anyway.

$\underline{\Delta}_1$	$\underline{\Delta}_2$	$\underline{\Delta}_3$
$\underline{\Delta}_1(1) = g_{12,1}$	$\underline{\Delta}_2(1) = g_{1,1}$	$\underline{\Delta}_3(1) = g_{2,1}$
$\underline{\Delta}_1(2) = g_{12,2}$	$\underline{\Delta}_2(2) = g_{1,2}$	$\underline{\Delta}_3(2) = g_{2,2}$
$\underline{\Delta}_1(1,2) = \{g_{12,1}g_{12,2}\}$	$\underline{\Delta}_2(1,2) = \{g_{1,1}g_{1,2}\}$	$\underline{\Delta}_3(1,2) = \{g_{2,1}g_{2,2}\}$

Six CIR can be formed on the foundation of the involutive interiors:

$$\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1)) = span\{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{2,1}\}$$

$$\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(2)) = span\{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{2,2}\}$$

$$\begin{aligned}
\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1,2)) &= span\{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}\} \\
\underline{\Delta}_{(1,2,3)}((1),(1,2),(1,2)) &= span\{g_{12,1}, g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}\} \\
\underline{\Delta}_{(1,2,3)}((1,2),(1),(1,2)) &= span\{g_{12,1}, g_{12,2}, g_{1,1}, g_{2,1}, g_{2,2}\} \\
\underline{\Delta}_{(1,2,3)}((2),(1,2),(1,2)) &= span\{g_{12,1}, g_{12,2}, g_{1,2}, g_{2,1}, g_{2,2}\}
\end{aligned} \tag{5.46}$$

but for example, $\underline{\Delta}_{(1,2,3)}((2),(1,2),(1,2)) = span\{g_{12,2}, g_{1,1}, g_{1,2}, g_{2,1}, g_{2,2}\}$ is not a CIR. It is possible to derive other CIRs from every single CIR having $n = dim(\mathcal{C}_0) = 5$ vector fields in (5.46), if the involutive interiors are split into single vector fields as involutive interiors. For instance,

$$\begin{aligned}
\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1)) &\Rightarrow \underline{\Delta}_{(1,1,2,3)}((1),(2),(1,2),(1)) \\
&\Rightarrow \underline{\Delta}_{(1,2,2,3)}((1,2),(1),(2),(1)) \\
&\Rightarrow \underline{\Delta}_{(1,1,2,2,3)}((1),(2),(1),(2),(1)).
\end{aligned}$$

However, these CIRs are not handled from now because they are degenerated, have more basis points, less dimensions and same vector fields in comparison with their original one. In other words, they are always worse representation than their parents in (5.46). Turn back now to CIRs in (5.46). All representations include all the strata which means that using Proposition 10, every CIR needs three basis points i.e. there is no distinguished optimal OBR among them. Considering ODRs, $\underline{\Delta}^{odr}_{(1,2,3)}((1,2),(1,2),(1,2)) = \underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1,2))$ [3] presents the optimal solution. Observe that it has the highest aggregated dimension number. Since every CIR involves three strata and $\underline{\Delta}^{odr}_{(1,2,3)}((1,2),(1,2),(1,2))$ possesses the most vector fields, the optimal CIR is also the ODR, i.e $\underline{\Delta}^{opt}_{(1,2,3)}((1,2),(1,2),(1,2)) = \underline{\Delta}^{odr}_{(1,2,3)}((1,2),(1,2),(1,2))$.

Step 3. (Obtain basis points) Take $\underline{\Delta}^{opt}_{(1,2,3)}((1,2),(1,2),(1,2))$ to solve motion planning problem. For, let $x(0) = [0\ 0\ 0.5\ 0\ 0]^T$ and $x(T) = [0.4\ 0.5\ 0.2\ 0.3\ 0.1]^T$. To obtain basis points, smooth motion planning algorithm takes only $n = 5$ vector fields into account from the six ones. Let them be selected by the generators of $\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1))$ i.e $gen(\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1))) = \{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{2,1}\}$. After performing smooth motion planning algorithm, the motion of modified robot in the plane is depicted in Figure 5.2. Its breakpoints that is the points which separate flows are

$$b^p_1 = \begin{pmatrix} -0.1184 \\ 0.2168 \\ 0.5 \\ 0 \\ 0 \end{pmatrix},\ b^p_2 = \begin{pmatrix} -0.3898 \\ 0.0685 \\ 0.5 \\ 0 \\ 0 \end{pmatrix},\ b^p_3 = \begin{pmatrix} -0.1519 \\ 0.2494 \\ 0.8 \\ 0.3 \\ 0 \end{pmatrix},\ b^p_4 = \begin{pmatrix} -0.1519 \\ 0.2494 \\ 0.8 \\ 0.3 \\ -0.5 \end{pmatrix},\ b^p_5 = \begin{pmatrix} 0.3668 \\ 0.5328 \\ 0.2 \\ 0.3 \\ 0.1 \end{pmatrix}$$

The breakpoints can be easily identified also in Figure 5.4. It shows the evolution of the configuration and position in the plane while the robot is moving from the initial point x_0 to the final point $b^p_5 \approx x(T)$. From the structure of CIR, one can realize that the basis points are $b_1 = b^p_2$, $b_2 = b^p_4$, $b_3 = b^p_5$.

[3] The superscript of all the optimal CIRs are kept for easier identification as a notation of a property and not for a determination or selection.

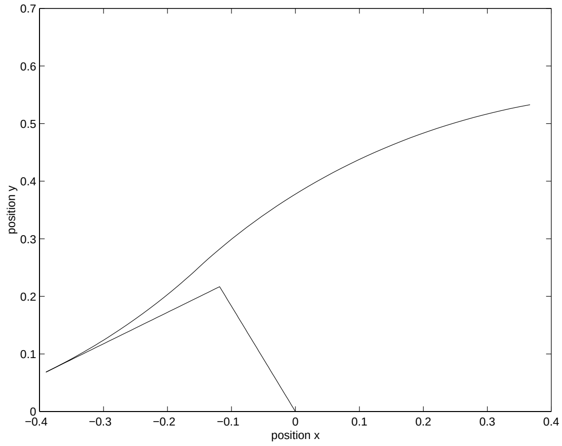

Figure 5.2: The path of modified hexapod robot in plane using basic stratified motion planning.

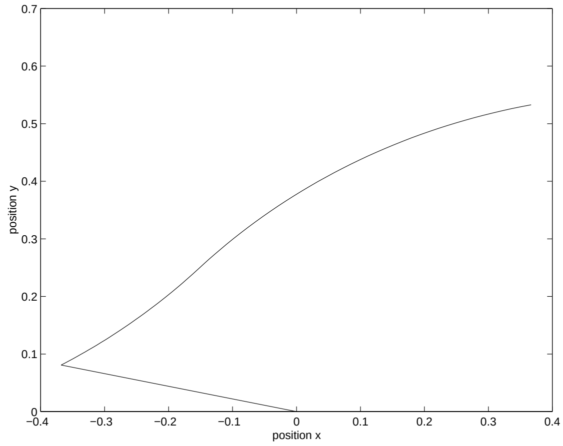

Figure 5.3: A possible path of modified hexapod robot in plane using holonomic stratified motion planning.

Step 4. (Decomposed trajectory planning.) In ideal case, b_5 should be the same as $x(T)$. The first two coordinates, however, are different because the vector fields of $gen(\underline{\Delta}_{(1,2,3)}((1,2),(1,2),(1)))$ do not generate a nilpotent Lie algebra and some numerical errors also appears by solving the differential equations. However it is guaranteed that all the basis points lie on the right slices. The discussion of optimal basis point permutation is omitted now, it depends on the actual prescribed trajectory and would need to solve smooth motion planning problem more than once. Now, the basis points which lie on the same slice and do not join two integral submanifold can be dropped out. It occurs with b_1^p and b_3^p because they lie exclusively on F_2 and F_3, respectively. (F_2 is the integral manifold determined by $\underline{\Delta}_2$ passing through $x(0)$ and F_3 is the slice of Δ_2 passing through b_1.) Arbitrary trajectories may be achieved now between any two neighboring basis points by using the trajectory generation Algorithm (Algorithm 11). It is possible for example to optimize the path of robot with straight-line trajectory between $x(0)$ and b_1 (Figure 5.3). As a second example, consider a trajectory planning between b_1 and b_2 on F_3 defined by $\underline{\Delta}_3$, where the solution of partial differential equation (5.31) is needed. As a result, the procedure in Section 5.4 yields a transformation (5.33), its inverse and Jacobian. In this particular case, one gets

$$\Psi_2 = \begin{pmatrix} x(1) \\ x(5) \\ x(1) - 2/5 - x(4)C(x(3) - x(4) + 1/2x(4)^2 S(x(3) - x(4)) + 1/6x(4)^3 C(x(3) - x(4)) \\ x(1) - 2/5 - x(4)S(x(3) - x(4) + 1/2x(4)^2 C(x(3) - x(4)) + 1/6x(4)^3 S(x(3) - x(4)) \\ x(3) - x(4) \end{pmatrix} \quad (5.47)$$

$(C(x) = cos(x),\ S(x) = sin(x))$. Keep in mind that Frobenius theorem may be applied only to nonsingular distribution. It may imply that obtaining Ψ is not possible if the distribution is singular at a point. However it is not the case now. During the procedure, a Taylor series approximation was used for the exponential map. It causes drawbacks in finding Ψ's inverse because it may spoil the property of diffeomorphism of Ψ.

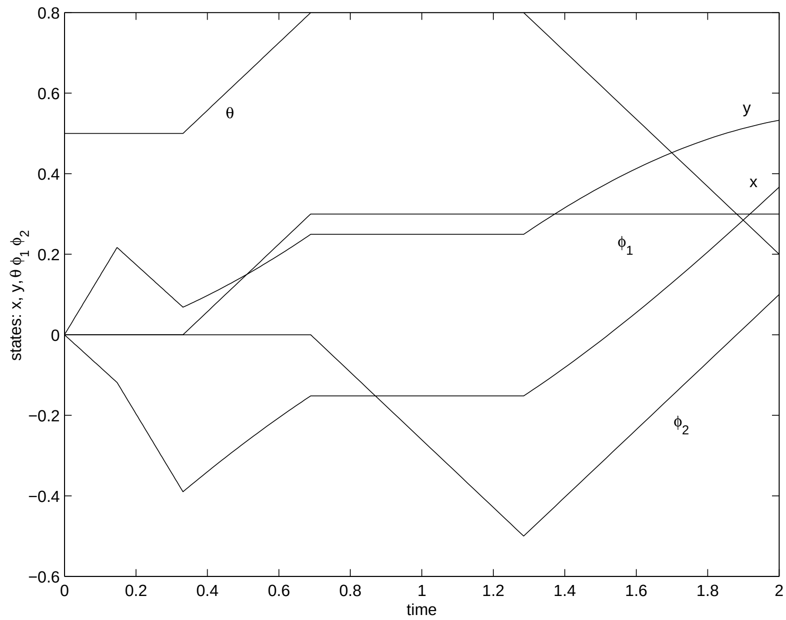

Figure 5.4: Configuration evolution. The evolution of robot position, orientation and leg angles during the motion of the robot from x_0 to $b_5^p \approx x(T)$.

However, in this example, the inverse and the Jacobian is computable and renders the vector fields of $\underline{\Delta}_3$ in the form

$$\frac{\partial \Psi_2}{\partial x}[g_{1,1} \quad g_{1,2}] = \begin{bmatrix} cos(x(3)) & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{5.48}$$

After transforming the basis points as $b_1^z = \Psi_2(b_1)$ and $b_2^z = \Psi_2(b_2)$, (5.48) allows to design a trajectory $z(t)$ in F_3 between b_1^z and b_2^z because its upper 2×2 part is a nonsingular matrix. (The last 3 coordinates of b_1^z and b_2^z are the same, since

$$b_1^z = \begin{pmatrix} -0.38 \\ 0 \\ -0.78 \\ -0.33 \\ 0.5 \end{pmatrix}, \qquad b_2^z = \begin{pmatrix} -0.15 \\ -0.5 \\ -0.78 \\ -0.33 \\ 0.5 \end{pmatrix} \tag{5.49}$$

If there are obstacles in the configuration space, it may be possible to design a trajectory which is able to avoid them. To do so, transform the obstacle points p^o laying in F_3 with $p_z^o = \Psi_2(p^o)$ and consider them during the trajectory planning in F_3. Of course, it is needed to transform back the trajectory into the original coordinates. It can be done by the inverse transformation $x(t) = \Psi_2^{-1}(z(t)$. The technique demonstrated

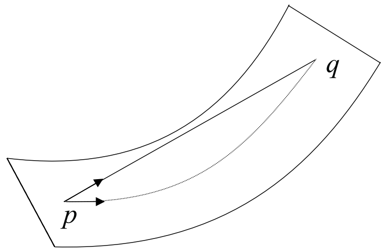

Figure 5.5: Heuristic method for local trajectory planning on an integral submanifold. The tangent vector of the desired trajectory is projected onto the submanifold. The projection determines the actual trajectory on the submanifold.

here is to be repeated for every slice F_i in the same way.

A much more heuristic method may be also used for trajectory planning. The idea is that one plans a trajectory between b_1 and b_2 regarding the entire configuration space $\mathcal{C}_0$. While the nominal trajectory is running along, its tangent vector can be projected into the tangent space spanned by the actual involutive distribution $\underline{\Delta}_3$ (see Figure 5.5). Moving along the projected direction omitting the original tangent vector, a trajectory evolves on an integral submanifold defined by $\underline{\Delta}_3$. Of course, this step requires switching between the original trajectory and its projection for every time t. The straight line in Figure 5.5 shows desired trajectory between the initial and final points both of them lying on the same integral submanifold. Straight line motion is prohibited, however, the final point may be reached through the points of the integral submanifolds using the projected tangent vectors in every points.

Step 5. (Inserting Moving off vector fields). In order to complete stratified motion planning, the insertion of moving off vector fields is needed. Obviously it occurs at basis points b_1 and b_2. Here, one can apply $g_{1,3}$ and $g_{2,3}$, respectively.

5.6 Summary and Conclusions

A new stratified motion planning algorithm has been proposed in this chapter. The new method improves the efficiency of the previous stratified approaches and assures a greater degree of freedom in the trajectory planning of holonomic stratified systems. To achieve this goal, this chapter has set up a central notion called completely involutive representation (CIR) fitting well to stratifaction and implying a class of trajectories called decomposed trajectories. Theorem 5 asserts statements that can be used for finding CIRs. In many cases nilpotency is not satisfied which fails a sequence of involutive interiors to be a CIR. However in real applications one can omit this condition considering this construction as it was a CIR. The only effect of this "cheating" is that solution is an approximation. It was shown by Theorem 6 that stratified system, under suitable control derived from CIR, is able to follow corresponding decomposed trajectories. Existence of CIR may be not unique. How to choose an optimal one was the subject in Section 5.2. Taking some considerations into account depending on the type of restrictions, Algorithm 10 proposed a heuristic way to choose a CIR for motion planning. It occurs that the order of involutive distributions in CIR is not indifferent either, influencing tracking error. Section 5.3 studied this problem and proposed a concept for finding an optimal order. The essential goal of holonomic stratified motion planning to divide motion planning problem into decomposed smooth ones. Section 5.4 proposed a way for solving motion planning problem on a slice based on symbolic approximation. The results were illustrated on a modified example of six-legged robot.

The technique generalizes the stratified motion planning accomplished along sequence of flows. It makes possible to prescribe a special kind of trajectory, a decomposed trajectory for the system. This decomposed trajectory is defined by basis points which can be selected from the results of a motion planning algorithm. The basis points are connected by slices defined by the vector fields of the strata. The drawback of the actual version of the approach is that it works only for systems which can be decomposed into holonomic subsystems along strata. It can be always done if the BSS has as many independent vector fields as many its dimension is. At some points of the method, symbolic approximation were used to avoid the hard (symbolic) computation task. The extension of the method which does not need the assumption of holonomy will be discussed in Chapter 6.

Chapter 6

Nonholonomic Stratified Motion Planning with Decomposed Trajectory

This chapter extends the results of holonomic stratified motion planning investigated in Chapter 5 to nonholonomic case. Unlike basic stratified approach reviewed in Subsection 2.2 where only desired final configuration could be achieved, the method to be proposed here follows decomposed trajectory, intensively relying on HOS inputs and trajectories summarized in Subsection 2.1.3, reported in Sussmann and Liu (1991) and Sussmann and Liu (1993). Note that similarly to previous chapter, this chapter considers also trajectory tracking problem, under some restrictions and not a simple path planning (which, in strict sense, disregards time relations). Since the method is subjected to restrictions, the motion planning as a general term is still used for the methodology. Decomposed trajectory is still restricted in the sense that it should go through a sequence of connected integral submanifolds. The sequence of inputs generates a sequence of trajectories that may converge to a reference trajectory. Since this technique is able to follow a special class of reference trajectories, one can apply it for avoiding obstacles in the stratified configuration space.

Section 6.1 establishes a foundation for algorithm and, structuralizing moving on vector fields taking part in bottom stratified system, defines the central notion of Decoupled Nonholonomic Representation (DNR), crucial to exploit stratifaction and HOS at the same time.

Section 6.2 asserts the main contributions of this chapter, all of them related to DNR, that assist to form a nonholonomic stratified motion planning algorithm. The algorithm, using a specific version of controllability conditions, carries out an approximation in general case from which, however, one may expect arbitrary precision.

The method is demonstrated on an illustrative example of six-legged robot where tracking a straight-line subsegment is to be accomplished.

Contributions of this chapter are covered by Harmati, Lantos, and Payandeh (2002d), Harmati, Lantos, and Payandeh (2002e), Harmati, Lantos, and Payandeh (2002b) and Harmati, Kiss, and Lantos (2002).

6.1 Decoupled Nonholonomic Representations (DNRs)

Stratified motion planning explored in Section 2.2 providing exact reaching of final point if control vector fields of BSS generates a nilpotent Lie algebra, cannot steer the system along desired trajectory. To overcome the problem, this section is devoted to devise a technique that extending the theory of Chapter 5 established originally for holonomic stratified systems is able to approximate a special class of trajectories to stratified systems. Initiate the discussion with the definition of system to be regarded in this chapter. From now, the configuration space of bottom stratum will be denoted by $\mathcal{C}_0$.

Definition 33. Consider a stratified controllable system. The system is said to be *nonholonomic stratified (controllable) system (NHSS)* if it is not possible to find $n = dim(\mathcal{C}_0)$ independent moving on vector fields from its BSS.

Stratifaction and nonholonomy bring major difficulties encumbering the use of all the vector fields at the same time. Nonholonomy viewed from control aspect requires extra constructions such as Lie brackets to solve motion planning problem. On the other hand, stratifaction does not allow arbitrary combination of vector fields at the same time, since some vector fields may be defined in different subsystems on different strata i.e. establishing connection between them requires switching.

Assumption 4. This chapter addresses motion planning problem to nonholonomic stratified systems which is stratified controllable and given by Definition 33.

This section extends the results of holonomic stratified motion planning (Chapter 5) and establishes a more general technique for nonholonomic stratified systems. Considering now the starting point, the BSS (2.55), it is useful to simplify its notations, similarly as occured in Chapter 5. Since considering the jth vector field of the ith stratum in the tangent space of the bottom stratum is necessary, some dimensions of the original vector field will be cut off. Denote a vector field $g_{i,j}\mid_{S_b}$ simply by $g_{i,j}$ omitting $\mid_{S_b}$ where index i assigns the stratum from which the vector field originates and index j identifies the vector field in the ith stratum. Precisely, a stratum was previously defined by multiindex I. Since there are finite number of strata, one can change the multiindex I to a simple integer i as actually done.
Let Δ_i denote the distribution of control vector fields in ith stratum, i.e. $\Delta_i = span(g_{i,1}, \ldots, g_{i,h_i})$ where h_i denotes the number of vector fields in BSS from the ith stratum S_i.

Definition 34. Consider the involutive closures of Δ_i which is denoted by $\bar{\Delta}_i$. Then Δ_i can be defined by

$$\bar{\Delta}_i = span(g_{i,1}, \ldots, g_{i,d_i}), \quad dim(\bar{\Delta}_i) = d_i \tag{6.1}$$

where $g_{i,1}, \ldots, g_{i,d_i}$ are said to be the Philip Hall generators of $\bar{\Delta}_i$ if they are the Philip Hall basis of the Lie algebra $\mathcal{L}(gen(\Delta_i)) := \mathcal{L}(g_{i,1}, \ldots, g_{i,h_i})$. Philip Hall generators can be associated to a function $pgen$ where $(g_{i,1}, \ldots, g_{i,d_i}) = pgen(\bar{\Delta}_i)$

If s denotes the number of strata taking part in BSS then multiindex $I(l) = i_1 i_2 ... i_l$ assigns a set of strata, or more specifically a sequence of strata. Unless anything else is said, the indices in $I(l)$ may have same values and $l \equiv \mid I \mid > s$ is not allowed. Notation $I(l)(z) = i_z$ is used to point out a single element of the multiindex I. Now, one can define the central notion of nonholonomic stratified motion planning.

Definition 35. Consider a BSS and let s be the number of strata in BSS. A sequence of involutive closures

$$\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}, \quad i_j \leq s, \ 1 \leq j \leq l, \ l \leq s \tag{6.2}$$

is said to be a *Decoupled Nonholonomic Representation (DNR)* of the BSS.

DNR plays a central role in this chapter. It is easy to see that vector fields in it are arranged by specific grouping establishing a structure which becomes beneficial later on. The key feature of structure that considering DNR as a sequence of vector fields, neighboring vector fields belong possibly to the same stratum. The idea is that using flow sequence philosophy in such order, the effect of vector fields belonging the same stratum may be modified in desired manner.

Definition 36. Consider a DNR $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}, \ i_j \leq s, \ 1 \leq j \leq l, \ l \leq s$ where $\bar{\Delta}_{i_j}$ is the involutive closure of Δ_{i_j} and $\Delta_{i_j} = span(g_{i_j,1}, \ldots, g_{i_j,h_{i_j}})$ i.e. $gen(\Delta_{i_j}) = (g_{i_j,1}, \ldots, g_{i_j,h_{i_j}})$. Then the set of generators

$$fgen(\bar{\Delta}_{I(l)}) = (g_{i_1,1}, \ldots, g_{i_1,h_{i_1}}, \ldots g_{i_j,1}, \ldots, g_{i_l,h_{i_l}}) \tag{6.3}$$

is said to be the *fundamental generators of DNR.*

The same situation occurs now as in Chapter 5, namely, only vector fields define distributions and a distribution does not define vector fields uniquely. However, in the context of this chapter, a distribution is always defined by explicit vector fields which enables to define its fundamental generators uniquely. As a matter of fact, producing fundamental generators from distribution can be considered as an inverse function of creating distributions from vector fields. Similarly, definition of extended generators proceeds as follows.

Definition 37. Consider a DNR $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}, \ i_j \leq s, \ 1 \leq j \leq l, \ l \leq s, \ \bar{\Delta}_{i_j} = span(g_{i_j,1}, \ldots, g_{i_j,d_{i_j}})$ i.e. $gen(\bar{\Delta}_{i_j}) = (g_{i_j,1}, \ldots, g_{i_j,d_{i_j}})$. Then the set of generators

$$egen(\bar{\Delta}_{I(l)}) = (g_{i_1,1}, \ldots, g_{i_1,d_{i_1}}, \ldots g_{i_l,1}, \ldots, g_{i_l,d_{i_l}}) \tag{6.4}$$

is said to be the *extended generators of DNR.*

Remark 24. Observe, that the set of extended generators of a DNR does not allow coupled Lie brackets, i.e. Lie brackets defined by vector fields of different strata.

Definition 38. The DNR $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}, \ i_j \leq s, \ 1 \leq j \leq l, \ l \leq s$ is said to be *Complete Decoupled Nonholonomic Realization (CDNR)* if $l = s$ and $i_p \neq i_q$ for every $p \neq q, \ 1 \leq p \leq s, \ 1 \leq q \leq s$.

Remark 25. Obviously, a BSS always has DNRs and CDNRs. For this, one should only produce a sequence of distributions $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}, \ i_j \leq s, \ 1 \leq j \leq l, \ l \leq s$ from the distribution of strata. If one establishes a sequence of distributions of *all* the strata playing role in BSS then a CDNR can be obtained. It is important to realize that different permutations of strata result different DNRs and CDNRs.

CDNR contains a maximal number of distributions allowed in a DNR, hence constructing a DNR with all the possible moving on vector fields requires usually a CDNR. Obviously, a CDNR is the richest in the number of vector fields, however it contains also the maximal number of strata which restricts a decomposed trajectory in greater extent.

Definition 39. Consider a DNR $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}$, $i_j \leq s$, $1 \leq j \leq l$, $l \leq s$. The DNR satisfies the *Decomposed Lie Algebra Rank Condition (DLARC) if*

$$\sum_{j=1}^{l} \bar{\Delta}_{i_j} = T_x\mathcal{C}_0. \tag{6.5}$$

Remark 26. DLARC resembles the condition (2.48) of stratified controllability in Theorem 4, however, the condition $\sum_{j=1}^{l} \bar{\Delta}_{i_j} = T_x\mathcal{C}_0$ is associated to the tangent space of bottom stratum while (2.48) takes into account the whole tangent space of the configuration space. The condition of Theorem 4, sufficient but not necessary for stratified controllability in testing NHSS, and the condition $\sum_{j=1}^{l} \bar{\Delta}_{i_j} = T_x\mathcal{C}_0$ of DLARC for a DNR of BSS of NHSS are equivalent if moving on vector fields are independent and decoupled from moving off vector fields. Indeed, this condition holds for the kinematic model of hexapod robots from its nature.

DLARC plays an important role in Section 6.2. As a matter of fact, the motion planning algorithm devised there works only for those stratified systems which satisfy this condition, however, DLARC covers a predominant range of stratified systems.

6.2 Nonholonomic Stratified Motion Planning Algorithm

This section deals with NHSS whose BSS has a DNR which satisfies DLARC. Unless anything else is said, this property is assumed. The following statement gives a strict but sufficient condition for satisfying DLARC.

Proposition 12. Consider the set of involutive closures $(\bar{\Delta}_1, \ldots, \bar{\Delta}_s)$. If it is possible to find n independent Philip Hall generators from it then there exists DNR of BSS that satisfies DLARC.
Proof: The proof is constructive and very simple. Let $(g_1, \ldots g_n)$ be n independent vector fields chosen from Philip Hall generators of involutive closures. Every g_j, $j = 1, \ldots, n$ points out a distribution from $(\bar{\Delta}_1, \ldots, \bar{\Delta}_s)$. Let $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\}$, $i_j \leq s$, $1 \leq j \leq l$, $l \leq s$ be the distributions selected in such way. Vector fields $(g_1, \ldots g_n)$ span $T_x\mathcal{C}_0$, hence, $\sum_{j=1}^{l} \bar{\Delta}_{i_j} \mid_x = T_x\mathcal{C}_0$. However, this is exactly the condition in the definition of DLARC.

Remark 27. Obviously, if a DNR satisfies DLARC then any CDNR does it, as well. In fact, in the proof of Proposition 12 a reduced DNR was obtained where $g_j \in egen(\bar{\Delta}_{I(l)})$, $j \leq n$ not necessarily using the vector fields of all the strata. Indeed, $(g_1, \ldots g_n)$ in Proposition 12 belong to the extended generators of the DNR. It is also possible to define a DNR with DLARC property by means of less than n vector fields at beginning. For this, one just should seek for instance suitable fundamental generators instead of extended generators.

Definition 40. Let a DNR of BSS be given in the form of $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$. Moreover, let be given the initial and the final points $x_I = x(0)$ and $x_F = x(T)$. Then, a trajectory $x(t)$, $0 \leq t \leq T$ is said to be *decomposed trajectory along* $\bar{\Delta}_{I(l)}$ between x_I and x_F if there exists a sequence of time intervals $t = \{t_1, \ldots t_l\}$ such that

$$
\begin{aligned}
\sum_{z=1}^{l} t_z \quad &= \quad T \\
t < t_1 \quad &\Rightarrow \quad \dot{x}(t) \in \bar{\Delta}_{i_1}(x(t)) \\
\sum_{z=1}^{j-1} t_z \leq t < \sum_{z=1}^{j} t_z \quad &\Rightarrow \quad \dot{x}(t) \in \bar{\Delta}_{i_j}(x(t)) \quad 1 < j \leq l
\end{aligned}
\tag{6.6}
$$

Definition 41. A DNR of BSS in the form of $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$ is said to be *decomposed controllable* if there exists decomposed trajectory along $\bar{\Delta}_{I(l)}$ between any $x_I, x_F \in \mathcal{C}_0$.

Definition 42. A DNR of BSS in the form of $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$ is said to be *weakly decomposed controllable* if for any x_I, x_F satisfying $\| x_F - x_I \| < d_{cr}$, there exists a sequence of inputs u^j,$j \to \infty$ involving a sequence of trajectories $x^j(t)$ that converges to a decomposed trajectory $x_{dec}(t) := x^{\infty}(t)$ along $\bar{\Delta}_{I(l)}$ such that $x_{dec}(0) = x_I$ and

$$
\| x_F - x^{\infty}(T) \| < \frac{1}{2} \| x_F - x_I \| \tag{6.7}
$$

The positive number d_{cr} is the critical distance to DNR. Trajectory $x_{dec}(t)$ is said to be an *approximated decomposed trajectory* between x_I and x_F.

Note that trajectory $x_{dec}(t)$ is a decomposed trajectory between x_I and $x^{\infty}(T)$, however it is an approximated trajectory between x_I and x_F. Critical distance plays similar role as in Subsection 2.1.2 and a theoretical way to obtain it is unknown. As a matter of fact, the need of using critical distance is originated from Subsection 2.1.2, as will be seen below.

Theorem 7. If a DNR $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$ is decomposed controllable then it satisfies the DLARC.

Proof: By assumption, let $x_{dec}(t)$ denote a decomposed trajectory between $x_I = x_{dec}(0)$ and $x_F = x_{dec}(T)$ where $x_I, x_F \in \mathcal{C}_0$. Such a trajectory defines a sequence of time intervals which satisfies (6.6), resulting the existence of an $i,\ i \leq s$ such that the tangent vector $\dot{x}$ in a time interval can be constructed in the form $\dot{x}(t) = g_{i,1}u_1 + \cdots g_{i,d_i}u_{d_i}$ where $g_{i,k} \in pgen(\bar{\Delta}_i)$. Let G be the set of all vector fields whose associated u_k, $k \leq d_i$ are not zero while the trajectory is traveling from x_i to x_F. It is known that $\dim(\mathcal{C}_0) = n$ and x_I, x_F can be arbitrary two points in $\mathcal{C}_0$ and there exists a decomposed trajectory between them, hence G contains n extended generators of DNR. Indeed, extended generators are defined by Philip Hall generators of the single involutive closures $(\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l})$. Observe now, that it coincides with condition of Proposition 12 asserting that $\bar{\Delta}_{I(l)}$ satisfies DLARC.

Theorem 7 is a result for analysis. In order to reveal controllability properties of the system, one needs its reversed version. Next theorem is not completely reverses the theorem, but represents its weakened version.

Theorem 8. If $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \dots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$ satisfies DLARC then $\bar{\Delta}_{I(l)}$ is weakly decomposed controllable.

Proof: The proof is constructive and provides an approximated decomposed trajectory along $\bar{\Delta}_{I(l)}$ between any $x_I, x_F \in \mathcal{C}_0$, $\| x_F - x_I \| < d_{cr}$. The proof proceeds as follows. Since DLARC holds, the set of involutive closures $(\bar{\Delta}_{i_1}, \dots, \bar{\Delta}_{i_l})$ spans the tangent space of $\mathcal{C}_0$. As a consequence, it is possible to select n independent vector fields $g_1, \dots, g_n$ from the extended generators $egen(\{\bar{\Delta}_{i_1}, \dots, \bar{\Delta}_{i_l}\})$. Let $g_1, \dots, g_n$ be ordered due to the sequence of $\{\bar{\Delta}_{i_1}, \dots, \bar{\Delta}_{i_l}\}$. More precisely, for all indices p and q, $1 \leq p < l$, $1 < q \leq l$, if $p < q$ then $g_p \in pgen(\bar{\Delta}_{i_{\bar{p}}})$, $g_q \in pgen(\bar{\Delta}_{i_{\bar{q}}})$ where $\bar{p} \leq \bar{q}$. Based on this, if k_j denotes the number of Philip Hall generators from $\bar{\Delta}_{i_j}$ taking part in $(g_1, \dots, g_n)$, then $(g_{i_1,1}, \dots, g_{i_1,k_1}, \dots, g_{i_l,1}, \dots, g_{i_l,k_l}) := (g_1, \dots, g_n)$. (It is easy to see that $\sum_{j=1}^{l} k_j = n$.)

Now, suppose for a moment that $(g_{i_1,1}, \dots, g_{i_1,k_1}, \dots, g_{i_l,1}, \dots, g_{i_l,k_l})$ generate a nilpotent Lie Algebra. Then, Algorithm 1 is able to reach any $x_F = x(T) \in \mathcal{C}_0$ from any $x_I = x(0) \in \mathcal{C}_0$ as a sequence of flows. The resulting trajectory can be described in the form

$$x(T) = \Phi(g_{i_l,k_l}, t_l) \circ \dots \circ \Phi(g_{i_1,1}, t_1) x(0) \tag{6.8}$$

Let us define now $l+1$ basis points in the following way:

$$\begin{aligned}
b_0 &= x(0) \\
b_1 &= \Phi(g_{i_1,k_1}, t_{1,k_1}) \circ \dots \circ \Phi(g_{i_1,1}, t_{1,1}) b_0 \\
b_2 &= \Phi(g_{i_2,k_2}, t_{2,k_2}) \circ \dots \circ \Phi(g_{i_2,1}, t_{2,1}) b_1 \\
&\vdots \\
b_l &= x(T) = \Phi(g_{i_l,k_l}, t_{l,k_l}) \circ \dots \circ \Phi(g_{i_l,1}, t_{l,1}) b_{l-1}
\end{aligned} \tag{6.9}$$

If $(g_{1,1}, \dots, g_{1,k_1}, \dots, g_{l,1}, \dots, g_{l,k_l})$ do not generate a nilpotent Lie Algebra then $x_F \neq x(T)$. If $\| x_F - x_I \| < d_{cr}$ then (6.7) holds (since smooth motion planning is carried out on bottom stratified system due to Subsection 2.1.2), which does not injure the weak decomposed controllability. It is easy to see that trajectory section

$$x(t) = \Phi(g_{i_1,k_1}, t_{1,k_1}) \circ \dots \circ \Phi(g_{i_1,1}, t_{1,1}) x(0) \quad 0 < t \leq \sum_{j=1}^{k_1} t_j \tag{6.10}$$

satisfies $\dot{x}(t) \in \bar{\Delta}_{i_1}$ since the flows in (6.10) develop always along vector fields which belong to $\bar{\Delta}_{i_1}$. As a result, the trajectory evolves on the integral submanifold of $\bar{\Delta}_{i_1}$ passed through $x(0)$. Similarly, one can see that $\dot{x}(t) \in \bar{\Delta}_{i_2}$, if $\sum_{j=1}^{k_1} t_j < t \leq \sum_{j=1}^{k_1+k_2} t_j$. This argument holds also for every subsegment between two neighboring basis points concluding that

$$x(T) = \Phi(g_{i_l,k_l}, t_{l,k_l}) \circ \dots \circ \Phi(g_{i_l,1}, t_{l,1}) \circ \dots \circ \Phi(g_{i_1,k_1}, t_{1,k_1}) \circ \dots \circ \Phi(g_{i_1,1}, t_{1,1}) x(0) \tag{6.11}$$

approximates a decomposed trajectory and leads the system from $x(0)$ to $x(T)$ where

$$\| x_F - x(T) \| < \frac{1}{2} \| x_F - x_I \|$$

due to (6.7). As a consequence, the $\bar{\Delta}_{I(l)}$ is weakly decomposed controllable and it proves the theorem.

Theorem 8 produces a trajectory along sequence of flows. Indeed, the trajectory was obtained by that smooth motion planning algorithm which fits to the stratified motion planning concept. However, keep in mind that smooth motion planning can not be applied to the original stratified system but only to the bottom BSS. The sequence of flows were selected in a special way which was defined by the DNR created from BSS. The advantage of such a structured sequence turns out bellow (cf. Theorem 9).

Theorem 9. Consider $\bar{\Delta}_{I(l)} = \{\bar{\Delta}_{i_1}, \ldots, \bar{\Delta}_{i_l}\},\ i_j \leq s,\ 1 \leq j \leq l,\ l \leq s$ satisfying DLARC. Let a motion planning problem be given between x_I and x_F such that $\| x_F - x_I \| \leq d_{cr}$. Moreover, let the reference trajectory be a decomposed trajectory $x_d(t)$ along $\bar{\Delta}_{I(l)}$ equipped with the basis points (6.9) (which is an approximated decomposed trajectory between x_I and x_F). Then, there exists a sequence of inputs $u^j(t)$, $j \to \infty$ that generates a sequence of trajectories $x^j(t)$ which converges to $x_d(t)$.

Proof: The proof is constructive. Since the $\bar{\Delta}_{I(l)}$ satisfies the DLARC, Theorem 8 can be applied. It generates $l+1$ basis points that restricts the set of possible reference decomposed trajectories. Keep in mind, b_0 and b_l are derived from x_I and x_F which can be prescribed. All the possible $x_d(t)$ have to go through $b_0, b_1, \ldots, b_l$. Additionally, it is known from definition that

$$\begin{array}{rcll} \dot{x}_d(t) & \in & \bar{\Delta}_{i_1} & \text{if} \quad 0 < t \sum_{j=1}^{k_1} \leq t_j \\ \dot{x}_d(t) & \in & \bar{\Delta}_{i_2} & \text{if} \quad \sum_{j=1}^{k_1} t_j < t \sum_{j=1}^{k_1+k_2} t_j \\ & \vdots & & \\ \dot{x}_d(t) & \in & \bar{\Delta}_{i_2} & \text{if} \quad \sum_{j=1}^{k_1+\cdots+k_{l-1}} t_j < t \leq \sum_{j=1}^{k_1+\cdots+k_l} t_j \end{array} \tag{6.12}$$

should be held. The question arises how to find an input which follows such a trajectory if the system is stratified. One may exploit the structure of DNR $\bar{\Delta}_{I(l)}$ derived from the bottom stratified system. Turn the attention now to the first subsegment of trajectory. The two end points of this subsegment are b_0 and b_1, both of them lying on the integral submanifold of $\bar{\Delta}_{i_1}$. Additionally, $\dot{x}_d(t) \in \bar{\Delta}_{i_1}$ if $0 < t \leq \sum_{j=1}^{k_1} t_j$, hence $x_d(t)$ is allowed to evolve on the integral submanifold of $\bar{\Delta}_{i_1}$. Such a trajectory satisfies all the restrictions prescribed for $x_d(t)$. Denote the integral submanifold of $\bar{\Delta}_{i_1}$ by F_1. Now, the motion planning problem is to be solved on F_1 between b_0 and b_1. The main problem arises when one realizes that it is desired to follow an arbitrary trajectory by means of less than d_i inputs (d_i is the dimension of the F_1). Obviously, it is not an elementary task to solve the problem, however invoking the algorithm which was discussed in Subsection 2.1.3, one obtains a highly oscillatory sequence (HOS) of inputs u^j which generates a sequence of trajectory converging to $x_d(t)$ in the subsegment between b_0 and b_1. In the next step, the procedure is accomplished for second subsegment between b_1 and b_2, similarly. Obviously, the only change in the algorithm that one should use the Philip Hall generators of $\bar{\Delta}_{i_2}$ instead of $\bar{\Delta}_{i_1}$. Proceeding the algorithm, the procedure is finished with a motion planning between b_{l-1} and b_l. Observe, in every subsegment only the vector fields of the associated distribution are used. Indeed, they are those moving on vector fields of the associated stratum that take part in

DNR $\bar{\Delta}_{I(l)}$ of BSS. It does not injure the requirements of stratified motion planning algorithm, namely, coupled Lie brackets i.e. the (recursive) Lie brackets defined by vector fields of different strata are not applied in the algorithm.

Theorem 9 provides a concept how one can plan a desired path between initial and final points that can be approximated using stratified motion planning and HOS sequence of inputs at the same time. Their combination allows the system to avoid obstacles in configuration space.

This chapter studies motion planning for stratified systems having a DNR with DLARC property. In fact, it is a restriction. It may exist a stratified controllable system that has no DNR which satisfies DLARC. Such a system emerges if the tangent space of BSS can be spanned only by applying coupled Lie brackets i.e. Lie brackets which are defined by the vector fields of different strata. Basic version of stratified motion planning algorithm can be still employed for such systems. The technique above omitted the question of time scaling which defines the motion of the robot along the trajectory in respect with time. The time scaling is needed if one defines a decomposed trajectory. The problem is solvable by time scaling algorithms which modify the values of inputs by constant multiplier in each subsegment. Such a treatment was used in Chapter 3. Finally, let the new algorithm be summarized

Algorithm 13. (Nonholonomic stratified motion planning with decomposed trajectory approximation)

Step 1. Create the BSS of stratified system (2.55).

Step 2. Obtain $\Delta_1, \ldots \Delta_s$.

Step 3. Determine the involutive closures $\bar{\Delta}_j$,$1 \leq j \leq s$ and their Philip Hall generators (see Definition 34).

Step 4. Create a DNR $\bar{\Delta}_{I(l)}$ that satisfies DLARC. Use Proposition 12.

Step 5. Obtain the fundamental and extended generators of $\bar{\Delta}_{I(l)}$ (see Definition 36 and Definition 37).

Step 6. Obtain the basis points $b_0, \ldots b_l$ by using Theorem 8.

Step 7. Plan the reference decomposed trajectory along $\bar{\Delta}_{I(l)}$ which goes through $b_0, \ldots b_l$.

Step 8. Produce a highly oscillatory sequence (HOS) of inputs u^j that generates a sequence of trajectories $x^j(t)$. Use the procedure of Theorem 9 so that the $x^j(t)$ converge to the prescribed decomposed trajectory.

Step 9. Insert moving off vector fields at the basis points according to stratified motion planning algorithm (Algorithm 3).

6.3 Example

In order to illustrate the method in previous section, consider the motion planning problem of six legged robot. The equations of motion in strata were presented in Example 6. Let the motion planning problem consist of finding input that steers the robot from $x_I = x(0) = (0\ 0\ 0\ 0\ 0)^T$ to $x_F = x(2) = (0.4\ 0.4\ 0\ 0\ 0)^T$ in the bottom stratum. It is seen that 2 sec is desired for final state. For, the motion planning is accomplished by the steps of Algorithm 13.

Step 1. Create BSS. The differential equation of BSS is cast into the form

$$\Sigma^{BSS} \quad \dot{x} \quad = \quad g_{12,1}u_{12,1} + g_{12,2}u_{12,2} + g_{1,2}\mid_{S_{12}} u_{1,2} + g_{2,2}\mid_{S_{12}} u_{2,2}$$
$$= \begin{bmatrix} cos(\theta) & cos(\theta) & 0 & 0 \\ sin(\theta) & sin(\theta) & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} u_{12,1} \\ u_{12,2} \\ u_{1,2} \\ u_{2,2} \end{pmatrix} \tag{6.13}$$

It is easy to see from (6.13) that the system is stratified controllable and nonholonomic. Keep in mind, that from now, according to convention suggested previously, every vector field and x are considered in the bottom stratum and the notation $|_{S_{12}}$ will be omitted.

Step 2. Obtain $\Delta_1, \ldots \Delta_s$. Based on earlier conventions, the distributions get new indices, as well. Distribution Δ_1 is related to the bottom stratum S_{12}, Δ_2 corresponds to S_1 and Δ_3 is composed of some vector fields of S_2. It is also useful to modify the first indices of vector fields according to their strata indices implying for example that $g_{2,1}$ is replaced by $g_{3,1}$, $g_{12,1}$ is replaced by $g_{1,1}$ and $g_{12,2}$ changes to $g_{1,2}$. The generators of the distributions are determined as

$$gen(\Delta_1) \quad = \quad (g_{1,1}, g_{1,2}) = \begin{bmatrix} cos(\theta) & cos(\theta) \\ sin(\theta) & sin(\theta) \\ 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad gen(\Delta_2) = (g_{2,1}, g_{2,2}) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$
$$gen(\Delta_3) \quad = \quad (g_{3,1}, g_{3,2}) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{6.14}$$

i.e. the distributions are $\Delta_1 = span(g_{1,1}, g_{1,2})$, $\Delta_2 = span(g_{2,1}, g_{2,2})$, $\Delta_3 = span(g_{3,1}, g_{3,2})$.

Step 3. Determine the involutive closures $\bar{\Delta}_j$, $j = 1, \ldots, s$ and their Philip Hall generators. First of all, some Lie brackets need to be evaluated such as

$$g_{1,3} = [g_{1,1}, g_{1,2}] = \begin{pmatrix} -2sin(\theta) \\ 2cos(\theta) \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad g_{1,4} = [g_{1,2}, g_{1,3}] = \begin{pmatrix} 2cos(\theta) \\ 2sin(\theta) \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{6.15}$$

It induces the Philip Hall generators

$$pgen(\bar{\Delta}_1) = (g_{1,1}, g_{1,2}, g_{1,3}, g_{1,4}) = \begin{bmatrix} cos(\theta) & cos(\theta) & -2sin(\theta) & 2cos(\theta) \\ sin(\theta) & sin(\theta) & 2cos(\theta) & 2sin(\theta) \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

$$pgen(\bar{\Delta}_2) = (g_{2,1}, g_{2,2}) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad pgen(\bar{\Delta}_3) = (g_{3,1}, g_{3,2}) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (6.16)$$

Step 4. Create a DNR $\bar{\Delta}_{I(l)}$ that satisfies DLARC. One may use Proposition 12 for this purpose. Let a DNR be constructed by

$$\bar{\Delta}_{(1,2,3)} = (\bar{\Delta}_1, \bar{\Delta}_2, \bar{\Delta}_3). \quad (6.17)$$

In fact, this is a CDNR since it includes all the involutive distributions. If one intended to optimize the number of basis points, it would be sufficient to compose a DNR with two distributions, for instance by $\bar{\Delta}_1$ and $\bar{\Delta}_2$. However, CDNR leads to an easier motion planning which may be better for illustration.

Step 5. Obtain the fundamental and extended generators of $\bar{\Delta}_{(1,2,3)}$. By definition, the set of fundamental generators of $\bar{\Delta}_{(1,2,3)}$ turns into

$$fgen(\bar{\Delta}_{(1,2,3)}) = (g_{1,1}, g_{1,2}, g_{2,1}, g_{1,2}, g_{3,1}, g_{3,2}) \quad (6.18)$$

and the extended generators of $\bar{\Delta}_{(1,2,3)}$ yields

$$egen(\bar{\Delta}_{(1,2,3)}) = (g_{1,1}, g_{1,2}, g_{1,3}, g_{1,4}, g_{2,1}, g_{1,2}, g_{3,1}, g_{3,2}). \quad (6.19)$$

Step 6. Obtain the basis points $b_0, \dots b_3$ by using Theorem 7. To do this, one carries out smooth motion planning steering the system from $x_I = (0\ 0\ 0\ 0\ 0)^T$ to $x_F = (0.4\ 0.4\ 0\ 0\ 0)^T$. The only thing necessary for it is to select vector fields from $\bar{\Delta}_{(1,2,3)}$ that generate a Lie algebra with maximal rank. Let them be $g_{1,1}$, $g_{1,2}$, $g_{2,2}$, $g_{3,2}$. The solution is obtained in the form of sequence of flows

$$\begin{aligned} x(T) &= \Phi(-1.2485 g_{3,2}, 0.1602) \circ \Phi(-1.2485 g_{2,2}, 0.1602) \circ \Phi(-1.2485 g_{1,2}, 0.3398) \circ \\ &\circ \Phi(-1.2485 g_{1,1}, 0.3398) \circ \Phi(1.2485 g_{1,2}, 0.3398) \circ \Phi(1.2485 g_{1,1}, 0.3398) \circ \\ &\circ \Phi(1.2485 g_{1,2}, 0.1602) \circ \Phi(1.2485 g_{1,1}, 0.1602) x_I \end{aligned}$$

The generated Lie algebra is not nilpotent, hence the $x(T)$ does not coincide exactly with x_F, namely, $x(T) = (0.3973\ 0.3945\ 0\ 0\ 0)^T$. From this, the basis points of smooth motion planning using flow sequences

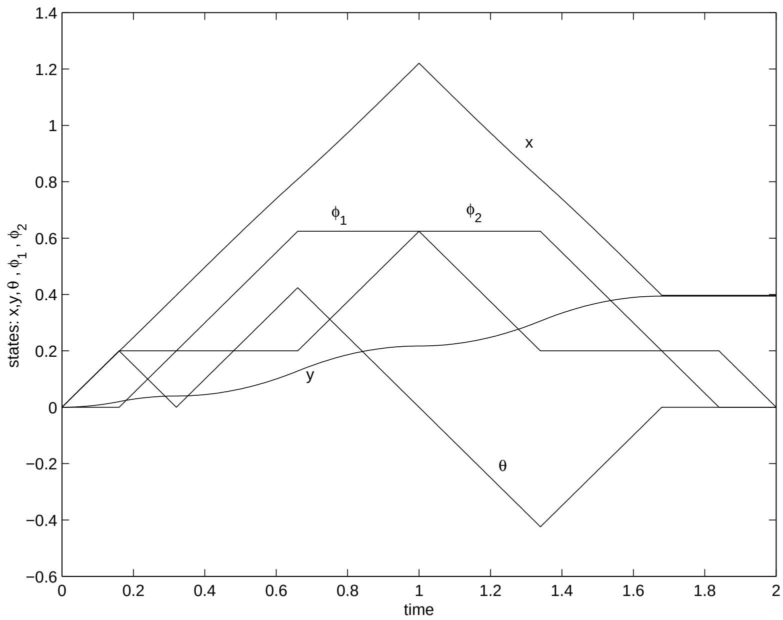

Figure 6.1: States of BSS evolving as sequence of flows and assigning basis points.

are

$$
\begin{aligned}
b_0 &= x_I = (0\;0\;0\;0\;0)^T \\
b_1 &= \Phi(-1.2485 g_{1,2}, 0.3398) \circ \Phi(-1.2485 g_{1,1}, 0.3398) \circ \Phi(1.2485 g_{1,2}, 0.3398) \\
&\circ\; \Phi(1.2485 g_{1,1}, 0.3398) \circ \Phi(1.2485 g_{1,2}, 0.1602) \circ \Phi(1.2485 g_{1,1}, 0.1602) b_0 \qquad (6.20) \\
&= (0.3973\;0.3945\;0\;0.2\;0.2)^T \\
b_2 &= \Phi(-1.2485 g_{2,2}, 0.1602) b_1 = (0.3973\;0.3945\;0\;0.2\;0)^T \\
b_3 &= \Phi(-1.2485 g_{3,2}, 0.1602) b_2 = (0.3973\;0.3945\;0\;0\;0)^T
\end{aligned}
$$

The entire trajectory generated by sequence of flows is depicted in Figure 6.1.

Step 7. Plan a decomposed trajectory along $\bar{\Delta}_{(1,2,3)}$ that travels through $b_0, \ldots b_3$. Let the trajectory between b_1 and b_3 be the same as determined by (6.20) and concentrate only to the trajectory segment between b_0 and b_1. In fact this can be considered the most important part because this subsegment corresponds to the path of hexapod robot in the plane. Let a straight-line trajectory be desired between b_0 and b_3. Figure 6.2 shows the resulting trajectory with the earlier stratified motion planning where the trajectory between b_0 and b_1 can not be modified (and cannot be predefined).

Step 8. Producing a highly oscillatory sequence (HOS) of inputs u^j that generates a sequence of trajectories $x^j(t)$. Use the procedure of Subsection 2.1.3 so that the $x^j(t)$ converge to the reference straight-line trajectory between b_0 and b_1. One can see, that a straight-line trajectory can be produced by $g_{1,1}$, $g_{1,2}$ and $g_{1,3}$ without using $g_{1,4}$ which is important because the approximation of a simple Lie bracket like $g_{1,3} = [g_{1,1}, g_{1,2}]$ is much easier than a nested one like $g_{1,4} = [g_{1,2}, [g_{1,1}, g_{1,2}]]$. In fact, Subsection 2.1.3 showed that $u^j(t) =$

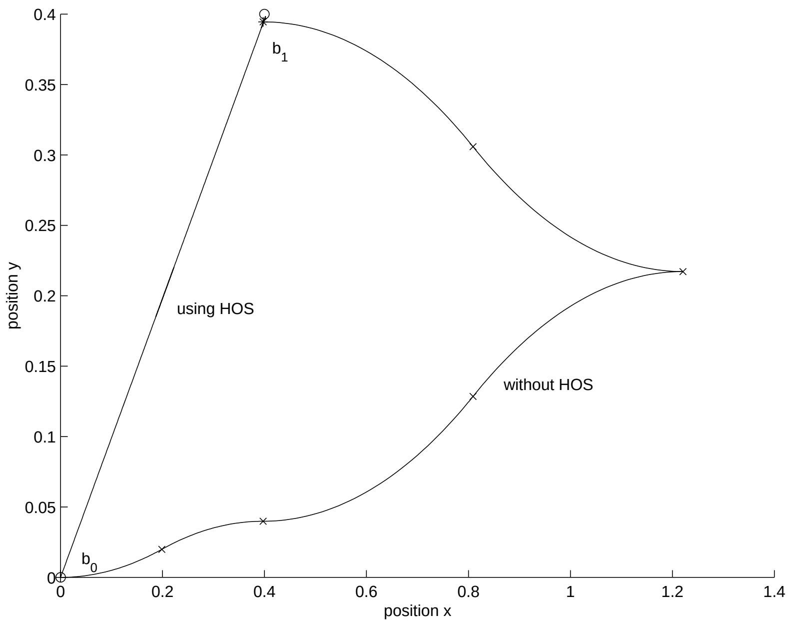

Figure 6.2: The path of the hexapod robot in the plane using the earlier stratified approach (without HOS) and the new version (with HOS) where the reference trajectory is a straight-line.

$\sqrt(j)\cos jtg_{1,1} + \sqrt(j)\sin jtg_{1,2}$ converge to $g_{1,3}/2$ which can be easily performed. The straight-line trajectory is depicted in Figure 6.2. The feature of the input is illustrated in a smaller time interval on Figure 6.3. In the algorithm, $j = 500$ was applied. Of course, j is far from infinity but still provided good approximation. The configuration b_1 after evaluating HOS inputs was reached at $t = 0.8$ in the simulation. The suitable time scaling fitted to the results of previous steps (where b_1 is desired to be reached at $t = 1.66$) is needed but it can easily be done by simple changing the values of the inputs (see Chapter 3).

Step 9. Insert moving off vector fields at basis points according to stratified motion planning algorithm (Algorithm 3). Since the system is stratified, one should insert moving off vector fields to switch between strata. It occurs at the base point b_1 and b_2 in the example. The moving off vector fields of higher strata are used for this purpose which, of course, needs extra time. The extra time intervals do not change the robot motion in the plane, however, a time rescaling may be required again (See Section 3.2).

6.4 Summary and Conclusions

This chapter presented a motion planning algorithm which can be used for trajectory approximation for stratified controllable systems extending the results of Chapter 5 to stratified nonholonomic case. The key notion the method relies on is called Decoupled Nonholonomic Representation, derived from BSS. Such a structure is strongly related to the so called decomposed trajectories and decomposed controllability, a property that allows to reach final point along decomposed trajectory. A special property, the DLARC was defined in this chapter facilitating motion planning by exploring the structure of DNR and covering a class of stratified systems this

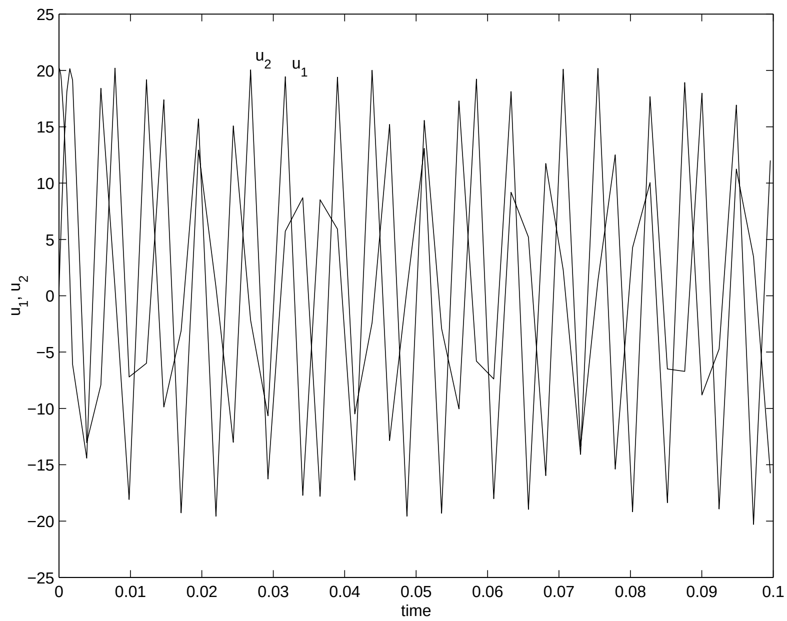

Figure 6.3: The highly oscillatory inputs that generates a straight-line path for hexapod robot in the plane. (Figure illustrates only a part of the whole time interval.)

chapter is involved in. Theorem 7 asserted that stratified system satisfies DLARC if it is decomposed controllable which can be used for analysis of stratified system. Its weakened inverse version, Theorem 8 stated that a stratified system with DLARC property is weakly decomposed controllable, i.e. one can find an approximated decomposed trajectory between initial and final configurations. The proof of theorem revealed the strong relation with basic stratified motion planning. Theorem 9 proposed a way how to approximate decomposed reference trajectories, suitable for obstacle avoidance. Using the results above, Algorithm 13 draws the steps of nonholonomic stratified motion planning.

The technique illustrated trajectory tracking problem on the simple example of hexapod robot. The tracking especially has benefits in obstacle avoidance problems even though it meets restrictions for reference trajectory. Based on the stratified concept, several symbolic computations are needed which are hard to be performed in some cases. Hence, they may cause difficulties in the realization. Such a problem is for example, if the distributions in DNR include recursive Lie brackets. The approximation of a trajectory using directions of the recursive Lie brackets may become complicated. Additionally, since resulting trajectory is generated by highly oscillatory sequence of inputs, the convergence to reference trajectory depends on the number of trajectories considered in the sequence.

Chapter 7

Conclusions

This dissertation proposed stratified motion and manipulation planning algorithms extending the results of basic approach in the Ph. D. Thesis of Goodwine (1998a). Stratified systems cover a wide class of physical systems where a locomotion is completed by a sequence of making and breaking contact points. Typical examples are the legged robots and manipulations with robotic hand using finger relocations. Controlling such systems implies difficulties because equations of motion may change when actual constraints change during a motion. The basic algorithm solves the point to point motion planning problem for stratified systems.

Chapter 3 proposed some improvements for basic stratified motion planning that increase robustness. Two problems were investigated, the convergency and time-scaling. Both of them are originated from the motion planning using flow sequences. One usually desires the robot to move along a reference path. This task cannot be completed by means of point to point motion planning such as the one the basic stratified motion planning relies on. Taking reference points along reference path facilitates the problem, however, the path between them remains still uncontrollable and the convergence is not guaranteed. A possible way to improve convergency which was proposed in this chapter consists of systematic insertion of extra reference points that leads robot back to reference path when divergence is observed. Another requirement for robot in many applications is to reach final state at a given time. The task was resolved by input scaling. Note however, that prescribing time point for every point of path was not solved. A possible extension in future could extend time scaling problem for other points of the path. Such a technique needs to give up the philosophy of constant inputs in subsegments.

Manipulation using finger relocations represents a typical applications where stratified motion planning may be invoked. Chapter 4 addressed manipulation planning algorithms. Since constraints are strongly dependent from the object shape, manipulation tasks are much more complex to be resolved by stratified approach than it would be for walking robots. Three manipulation strategies were proposed to solve this problem, each of them is based on a special parametrization allowing a simple stratified description. The fitted system obtained this way provides simple control vector fields and reduces the complexity of underlying differential geometric computations based on symbolic treatments. Manipulation planning using fitted system has benefits such

as avoiding the hard symbolic computation, easily interpretable trajectory and assists to reduce the symbolic steps almost to pure numerical computations. However, the trajectory between the initial and final points in a subsegment is not under control, therefore basic fitted stratified manipulation cannot keep fingertips in their assigned operating space. Fitted system requires to apply transformation related to Jacobian matrices of the fingers. The control signals are represented by the joint variables.
The fitted semi-stratified manipulation differs at two main points from the previous approach. It requires suitably chosen reference contact points and it extends basic fitted stratified approach with unconstrained finger relocations. The technique needs appropriate contact points that restrict manipulation. In this case, dextrous manipulation becomes rather only an object manipulation ensuring collision avoidance and grasp stability at the same time. Unconstrained finger relocations in semi-stratified approach realize a second phase in manipulation after the fitted stratified manipulation is completed. This task is separable from stratified core and provides a greater freedom in alternative paths than basic fitted stratified manipulation when a finger is relocated back into other contact points. A finger can be relocated along arbitrary path above the object while a fitted stratified approach should move the fingertip only along certain direction above the object at a time.
The basic fitted concept allows to divide the configuration space of bottom stratum into slices (subspace decompositions) which makes exact trajectory tracking on slices possible. As a result, it is able to move the object along arbitrary path and in the next sequence it relocates fingers along arbitrary path above the object, as well. Since the object can locally move in desired way, the strategy can be successfully applied for obstacle avoidance.

The idea of decomposing bottom stratum into slices, assuming that stratified system shows holonomic property, was generalized in Chapter 5. First one needs to carry out basic stratified motion planning. If there exists a sufficient number of moving on vector fields and they are grouped into a sequence, required for basic approach, where neighboring vector fields belong to the same stratum then subsequences created by the vector fields of a stratum can be replaced by any path generated by the same vector fields. In fact, this is a sequence of holonomic motion planning methods on strata, which is, however, not elementary. To solve the problem, a symbolic approximation was proposed. The sequence of systematically grouped vector fields is well described by CIRs realizing the central notion of the method that allows the system to follow decomposed trajectories. Existence of CIR, however, may not unique. There exist alternative possible manners to select an optimal CIR, but optimality also depends on the actual task. Two methods were proposed, one of them minimizes the number of basis points and the second one intends to maximize the dimensions of strata where trajectory evolves. There is a trade-off between the two approaches, therefore a heuristically optimal procedure was devised which establishes a good compromise.

Chapter 6 followed the more general approach when the system is stratified controllable, however, the holonomic property does not hold in every stratum. Using the key notion of DNR one can define decomposed controllability which enables the stratified system to approximate decomposed trajectories. A motion planning was developed in this chapter for those stratified systems that satisfy the so called Decomposed Lie Algebra Rank Condition, a weakened version of the well known Lie algebra Rank Condition. The technique is able to follow an approximated decomposed reference trajectory. The main advantage of holonomic and nonholonomic stratified motion planning that they assist to solve obstacle avoidance problem, since they are able to modify the trajectory of basic approach. Problems arise, however, if basis points are found in a forbidden region occupied by obstacles. The technique illustrated trajectory tracking problem on the simple example of hexapod robot. Based on the stratified concept, several symbolic computations are needed, hard to be performed in some cases, hence they may cause difficulties in the realization. Such a problem is for example, if distributions in DNR include recursive Lie brackets. The approximation of a trajectory using directions of recursive Lie brackets may easily become complicated. Additionally, the procedure intensively uses HOS inputs and trajectories, hence the convergency of trajectory sequence depends on the highest approximation index considered.

The methods introduced and elaborated in this thesis are subjected to some restrictions facing up to important challenges in future works. One of the main limitations is that kinematic model was considered throughout the thesis. It comes from two main reasons. First of all, the whole concept relies on the commutation of moving on and moving off vector fields. While kinematic model satisfies this property naturally in many cases (see six-legged robot model), it does not hold for dynamic model. On the other hand, all the known smooth motion planning methods the stratified approach relies on work on kinematic model, as well. Algorithm solving the above problems should probably involve new mathematical tools that also generalize the whole modeling. A potential avenue of this work would apply the results of Lewis (1995) Bullo and Zefran (2002a) and Bullo (1999). Fortunately, many hierarchical control concepts exist where kinematic design on middle level control hierarchy play a control role.

This dissertation considered motion planning methods which realize an open loop control. In applications, the disturbances, simplified modeling, and inaccurate computations do not allow to achieve exactly the desired goal of task. Hence, a meaningful extension in the future would be a closed loop architecture. Developing such methods encounters difficulty for the present, since algorithms invoked use the tool of differential geometry and some part of it requires extensive symbolic computations. The software packages supporting such computations are not efficient enough for complex applications, yet.

A key epiphenomenon of stratified control is the switching. Obviously, the number of switchings and number of moving on vector fields in a sequence may play important role in many application. The minimization of number of switchings and the number of vector fields are also a potential task in the future. To solve this task, one should consider the order of Lie brackets taking part in bottom stratified system. Order of Lie brackets have another impact on efficiency. HOS trajectories along higher order Lie brackets are much more complex than trajectories along single Lie brackets.

An interesting extension of stratified control would be applied to formation control where different agents move in configuration space to accomplish a common task. This area of control is being intensively explored nowadays and stratified control may be able to add some contribution to it, as well. The agents which are usually mobile robots or other machines (e.g. aircrafts) resemble the fingers of robotic hand which manipulate an object together.

Many issues to be solved in future arise from stratified control in stair-like environment. Such a situation occurs when a legged robot climbs a stair or a robotic hand manipulates an object with nonsmooth surface. Its stratified modeling requires to attach bottom stratum to every smooth surface. This however spoils smoothness and a possible concept should be involved in grasp or gait planning, a high level algorithm that defines extra (supporting) contact points. Thus the motion and manipulation planning consist of finding input that steers the system from a supporting point to the next one. Choosing appropriate supporting points one may avoid the legs to kick into stair. Holonomic and nonholonomic motion planning methods along decomposed trajectories play important role in this phase because they can be applied to follow or approximate a trajectory to avoid forbidden regions realized by obstacles. One of the main problems with stratified control in stair-like environment is that defining supporting points is not elementary because stratifaction becomes intricate easily. The graph of strata structure facilitates finding a suitable strata transition between initial and final configuration. A possible future work may elaborate a complete algorithm that solves the stratified motion planning problem in stair-like environment.

Appendix

This appendix intends to summarize the most important definitions and results of differential geometry the dissertation relies on. More extended discussions can be found in several books. The concept introduced here follows mainly the treatment of Isidori (1996), Olver (1993), Sastry (1999) and Murray, Li, and Sastry (1994).

Manifolds

Definition A-1. Let A be an open subset of $\mathbb{R}^n$ and let $f: A \to \mathbb{R}$ be a function. The function $f(x) = f(x_1, \ldots, x_n)$ is said to be a *smooth* (or C^∞) function if its partial derivatives of any order with respect to $(x_1, \ldots, x_n)$ exist and are continuous.

Definition A-2. Let A be an open subset of $\mathbb{R}^n$. A *smooth mapping* $F: A \to \mathbb{R}^m$ is a collection $(f_1, \ldots, f_m)$ of smooth functions $f_i: A \to \mathbb{R}$.

Smoothness in the above definitions simplifies the mathematical treatments in differential geometry, hence they are very useful in the more complex object and system descriptions. It immediately appears in one of the most important basic notions of differential geometry called manifold which essentially set up a coordinate-free description of an object.

Definition A-3. An *m-dimensional manifold* is a set M, together with countable collection of subsets $U_\alpha \subset M$, called *coordinate charts* , and one-to-one functions $\phi_\alpha: U_\alpha \to V_\alpha$ onto connected open subsets $V_\alpha \subset \mathbb{R}^m$ called *local coordinate map* , which satisfy the following three properties:

1. The coordinate charts cover M, i.e.
$$\bigcup_\alpha U_\alpha = M.$$
2. On the overlap of any two coordinate charts $U_\alpha \cap U_\beta$ the composite map
$$\phi_\beta \circ \phi_\alpha^{-1}: \phi_\alpha(U_\alpha \cap U_\beta) \to \phi_\beta(U_\alpha \cap U_\beta) \tag{A-1}$$
is smooth.
3. If $x \in U_\alpha$, $\tilde{x} \in U_\beta$ are distinct points of M, then there exist open subsets $W \subset V_\alpha$, $\tilde{W} \subset V_\beta$, with $\phi_\alpha(x) \in W$, $\phi_\beta(\tilde{x}) \in \tilde{W}$ such that
$$\phi_\alpha^{-1}(W) \cap \phi_\beta^{-1}(\tilde{W}) = \emptyset.$$

It is easy to see that U_α and ϕ_α are strongly related to each other, hence, if it is not confusing, a pair (U_α, ϕ_α) is also regarded as a coordinate chart in a wider sense. The coordinate charts $\phi_\alpha : U_\alpha \to V_\alpha$ endow the manifold M with the structure of topological space which means that for each open subset $W \subset V_\alpha \subset \mathbb{R}^m$, $\phi_\alpha^{-1}(W)$ assigns an open subset of M. These sets compose a *basis* for the topology. In terms of this topology, the third condition in definition A-3 expresses the Hausdorff separation axiom , namely, if x and $\tilde{x}$ are two distinct points in M then there exist open sets U and $\tilde{U}$ such that $x \in U$, $\tilde{x} \in \tilde{U}$ and $U \cap \tilde{U} = \emptyset$. Manifold plays a crucial role in system theory because it is the scene where the state of system develops in time.

Example A-1. The most evident m-dimensional manifold is the Euclidean space $\mathbb{R}^m$ itself.

Example A-2. The unit circle $S^1 = \{(x, y) : x^2 + y^2 = 1\}$ is a one-dimensional manifold (with two coordinate charts). A point on S^1 can be identified with the angular coordinate θ, where $(x, y) = (\cos\theta, \sin\theta)$.

Example A-3. The Cartesian product $T^2 = S^1 \times S^1$ is a two-dimensional manifold known as torus.

It is useful to consider sometimes only a part of a manifold which is also a manifold itself. For, it is necessary to introduce the notion of maximal rank.

Definition A-4. Let $F : M \to N$ be a smooth mapping from an m-dimensional manifold M to an n-dimensional manifold N. The *rank* of F at a point $x \in M$ is the rank of the $n \times m$ Jacobian matrix $(\partial F_i/\partial x_j)$ at x, where $y = F(x)$ can be given in any convenient local coordinates near x. The mapping F has maximal rank on a subset $S \subset M$ if for each $x \in S$ the rank of F is the minimum of m and n.

Definition A-5. Consider a smooth manifold M. A subset $N \subset M$ is said to be a *submanifold* of M if there exists a smooth one-to-one map (immersion) $\phi : \tilde{N} \to N$ that satisfies the maximal rank condition everywhere, where the parameter space $\tilde{N}$ is some other manifold and $N = \phi(\tilde{N})$ is the image of ϕ.

The notion of submanifold becomes beneficial especially for those system descriptions where additional restrictions emerge.

Lie groups

Roughly speaking, Lie groups are the combination of the algebraic notion of a group and the differential-geometric notion of a manifold. This concept provide an efficient tool in nonlinear system theory. The review starts with the basic definition of group.

Definition A-6. A *group* is a set G together with a group operation, called multiplication, such that for any two elements g and h of G, the product gh is also an element of G. The group operation must satisfy three axioms:

1. *Associativity.* If g, h and j are elements of G, then $g(hj) = (gh)j$.
2. *Identity element.* There exist a special element of G denoted by e and called the identity element such that $eg = g = ge$ for all $g \in G$.
3. *Inverses.* For each $g \in G$, there is an inverse $g^{-1} \in G$ such that $gg^{-1} = e = g^{-1}g$.

There are some commonly used mathematical structures that carries the feature of groups. The next examples give an essence from them.

Example A-4. The set of integers $G = \mathbb{Z}$ together with the addition as a group operation, with the identity element 0 and with the inverse $-x$ of an integer x is a group.

Example A-5. Let $GL(n, \mathbb{R})$ be the set of all invertible $n \times n$ matrices where all the elements are real numbers. Then, the matrix multiplication as group operation, the identity matrix as identity element, and the inverse of matrix as inverse element define the *general linear group* .

What makes the Lie group different from the group is that Lie group possesses also the features of smooth manifold. In this case, one may also apply the tools of calculus such as differentiation.

Definition A-7. A group G is said to be an *r-parameter Lie group* if it is also an r-dimensional smooth manifold and the group operation

$$m : \; G \times G \to G, \quad m(g,h) = L_g h = gh, \quad g, h \in G, \tag{A-2}$$

and the inversion

$$i : \; G \to G, \quad i(g) = g^{-1}, \quad g \in G, \tag{A-3}$$

are smooth maps between manifolds.

Lie groups have a profound impact on all areas of mathematics as well as engineering. They are especially helpful on the field of differential equations, in strong relation with modern nonlinear control theory. Some examples of Lie groups are presented now as simple illustrations.

Example A-6. The $G = \mathbb{R}^r$ carries an evident smooth manifold structure, and it is equipped with vector addition $(x, y) \to x + y$ which satisfies the conditions of group operation with the identity element 0 vector and with inverse $-x$ of an element x.

Example A-7. Let $G = SO(2)$ define the the group of rotation in the plane as

$$G = \left\{ \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} : \; 0 \leq \theta < 2\pi \right\} \tag{A-4}$$

where θ is the angle of rotation. On the one hand, G can be identified with unit circle S^1 (see example A-2) since both of them plot the same points. In other words, G carries the manifold structure. On the other hand, G is an orthogonal group defined by

$$O(2) = \{X \in GL(2) : X^T X = I\} \tag{A-5}$$

Obviously, G is a Lie group since it is a manifold and a group as well.

One may define many Lie groups that come from distinct definition, however, it may easily turn out that they are actually in strong connection. Hence, its is worth introducing two notions. A Lie group *homomorphism* is a smooth map $\phi : G \to H$ between two Lie groups where the group operations satisfy

$$\phi(gh) = \phi(g)\phi(h) \quad g, h \in G. \tag{A-6}$$

If there exists a smooth inverse of ϕ, it is considered as an *isomorphism* between G and H. The isomorphic Lie groups carry the same structure so there is no specific reason to distinguish them while they are studied. For instance, the positive real numbers $\mathbb{R}^+$, together with everyday multiplication define a Lie group. Indeed, this Lie group is isomorphic to the additive Lie group R where the exponential function $\phi : \mathbb{R} \to \mathbb{R}^+$, $\phi(t) = e^t$ produces the isomorphism. An interesting fact is that there exist only two (connected) one-parameter Lie groups, namely $\mathbb{R}$ and $SO(2)$, any other one-parameter Lie group is isomorphic to one of them. Another way to simplify discussion is to consider group elements close to the identity element. In this case, the Lie group is local. Lie groups are mostly regarded in applications as groups of transformations on a manifold and not as self contained entities. In fact, this gives the power to the tools based on Lie groups. The first step is to define transformations.

Definition A-8. Given a smooth manifold M. A *local group of transformations* acting on M is defined by a (local) Lie group G, an open subset U, with

$$\{e\} \times M \subset U \subset G \times M \tag{A-7}$$

which is, by definition, the domain of group action, and a smooth map $\Psi : U \to M$ which is required to satisfy:

1. If $(h, x) \in U, (g, \Psi(h, x)) \in U$ and $(gh, x) \in U$, then $\Psi(g, \Psi(h, x)) = \Psi(gh, x))$.
2. $\Psi(e, x) = x$ for all $x \in M$.
3. If $(g, x) \in U$, then $(g^{-1}, \Psi(g, x)) \in U$ and $\Psi(g^{-1}, \Psi(g, x)) = x$.

For the easier notations, $\Psi(g, x)$ can be simply replaced by gx. The notion of (local) group of transformation will reveal interesting structures. To do so, the introduction of vector fields is necessary.

Vector fields

Consider a manifold M and a smooth curve C defined on it such that the $\phi : I \to M$ parameterizes the curve where I is a subinterval of $\mathbb{R}$. Then, curve C can be given by m smooth functions $\phi(\epsilon) = (\phi_1(\epsilon), \ldots, \phi_m(\epsilon))$ in local coordinates $(x_1, \ldots, x_m)$ with respect to the real variable ϵ. The curve C provide a so called *tangent vector* at each $x = \phi(\epsilon)$ which is defined by $\dot{\phi}(\epsilon) = d\phi/d\epsilon = (\dot{\phi}_1(\epsilon), \ldots, \dot{\phi}_m(\epsilon))$. The notation

$$v_x = \dot{\phi}(\epsilon) = \dot{\phi}_1(\epsilon)\frac{\partial}{\partial x_1} + \cdots + \dot{\phi}_m(\epsilon)\frac{\partial}{\partial x_m} \tag{A-8}$$

with $\partial/\partial x_i$ assists to make a difference between tangent vector and any points given in local coordinates. The set of tangent vectors generated by all the curves passing through a given point x in M is the *tangent space* to M at x and is denoted by $T_x M$. The m-dimensional manifold M renders an m-dimensional tangent space

T_xM which is actually an m-dimensional vector space equipped with the basis $(\partial/\partial x_1, \ldots, \partial/\partial x_m)$ in local coordinates. The notation

$$TM = \bigcup_{x \in M} T_xM \tag{A-9}$$

used for the collection of all tangent spaces belonging to all points x in M defines the *tangent bundle* of M. Note, that tangent bundle is a smooth manifold having $2m$ dimensions. Everything is available now to define a central notion of this work.

Definition A-9. A *vector field* $f(x)$ on manifold M defines a tangent vector $v_x \in T_xM$ in each point $x \in M$ such that the tangent vector is smoothly varying from point to point. A vector field can be described in the form

$$f(x) = f^1(x)\partial/\partial x_1 + \cdots + f^m(x)\partial/\partial x_m \tag{A-10}$$

(To avoid the confusion, note that $f^1(x)$ here does not correspond to a whole vector field f_1 that emerges often in this dissertation but $f^1(x)$ is only a component of a vector field.) Every vector field generates a curve on its manifold.

Definition A-10. A smooth parametrized curve $x = \phi(\epsilon)$ is called *integral curve* of a vector field $f(x)$ if its tangent vector at any point equals to the value of $f(x)$ at the same point, namely $\dot{\phi}(\epsilon) = f(\phi(\epsilon))$.

Note, that $x = \phi(\epsilon) = (\phi_1(\epsilon), \ldots, \phi_m(\epsilon))$ (given in local coordinates) solves the ordinary differential equations

$$\frac{dx_i}{d\epsilon} = f^i(x), \quad i = 1, \ldots, m, \tag{A-11}$$

where $f^i(x)$ are the coefficients of $f(x)$ as mentioned above. An integral curve is *maximal* if it is not contained by any longer integral curve. Integral curve leads to the notion of flow.

Definition A-11. A $\Phi(x, \epsilon)$ flow generated by a vector field $f(x)$ is a parametrized maximal integral curve of $f(x)$ that passes through x in M.

Every pair (x, ϵ) assigns a point on an integral curve. The integral curve is assigned by x since it should pass through x and the point on integral curve is assigned by its parameter, ϵ. The flow manifests three fundamental properties:

$$\begin{aligned} \Phi(\Phi(x,\epsilon),\delta)) &= \Phi(x, \epsilon + \delta), \quad x \in M \text{ for all } \epsilon, \delta \in \mathbb{R} \\ \Phi(x, 0) &= x \\ \frac{d}{d\epsilon}\Phi(x,\epsilon) &= f(\Phi(x,\epsilon)) \quad \text{ for all } \epsilon \text{ where it is defined.} \end{aligned} \tag{A-12}$$

Observe now, that the three flow properties in (A-12) comply with the conditions of definition A-8 associating the flow along a vector field to a local group action of Lie group $\mathbb{R}$ on manifold M. One should only pay attention to the parameter orders. In the notation $\Psi(\epsilon, x)$, the parameters ϵ and x appear in reverse order in comparison with $\Phi(x,\epsilon)$. The constructions like $\Psi(\epsilon,x)$ where $\epsilon \in \mathbb{R}$ ($\mathbb{R}$ is a Lie group) are generally referred to as one-parameter group of transformations. The vector field $f(x)$ is also said to be the *infinitesimal generator* of the action which is, in fact, the group of translations in infinitesimal extent:

$$\Phi(x,\epsilon) = x + \epsilon f(x) + O(\epsilon^2) \tag{A-13}$$

The flow is also called *exponentiation of the vector field* promoting the notation

$$\exp(\epsilon f)x = \Phi(x,\epsilon) \tag{A-14}$$

On first encounter, it may seem somewhat strange but using this convention to flow properties (A-12)

$$\begin{aligned} \exp((\delta+\epsilon)f)x &= \exp(\delta f)\exp(\epsilon f)x \\ \exp(0f)x &= x \\ \frac{d}{d\epsilon}(\exp(\epsilon f)x) &= f(\exp(\epsilon f)x) \end{aligned} \tag{A-15}$$

for all $x \in M$ which reflect exactly the properties of exponential function.

Differential operators

After introducing vector fields, it is possible to define operators related to them. As a matter of fact, the most applications using differential-geometric concept can not evade to invoke their service. Vector fields will be also considered as differential operators on smooth functions in this part of discussion, hence in this case, vector fields are denoted by capital letter in order to clearly distinguish them from smooth functions and group elements. However, note, that this is not the common situation in the most part of the dissertation. In this case, vector fields are denoted by lower case, again.

Definition A-12. Let $s \in C^\infty(M)$ be a smooth, $\mathbb{R}$ valued function on M and let X be a (smooth) vector field. The *Lie derivative* of s along X at $p \in M$ is a new function

$$Xs : M \to \mathbb{R}, \quad Xs(p) = X_p s \tag{A-16}$$

which is expressed in a local coordinates, with local coordinate chart (V,ϕ) as

$$Xs(x) = \sum_{i=1}^{n} \frac{\partial s}{\partial x_i} X^i(x) \tag{A-17}$$

where

$$X = \sum_{i=1}^{n} X^i(x)\frac{\partial}{\partial x_i} \quad \text{and } x = \phi(p) \tag{A-18}$$

Lie derivative has several notation in literature: $Xs(p) = X_p s = L_X s(p) = X[s](p) = ds(p)X(p)$. Lie derivative is obtained by inner product and, roughly speaking, it measures how close the gradient vector of function s is to the vector field X at a given point x. The definition of Lie derivative can be extended to $\mathbb{R}^m$ vector valued function s which implies that its Lie derivative becomes also vector valued with the same number of components. Each component of Lie derivative is computed from the corresponding component of s on the base of definition A-12.

$$Xs(x) = \begin{pmatrix} \sum_{i=1}^n \frac{\partial s_1}{\partial x_i} X^i(x) \\ \vdots \\ \sum_{i=1}^n \frac{\partial s_m}{\partial x_i} X^i(x) \end{pmatrix} \tag{A-19}$$

Let Y is also a smooth vector field. Then, $XYs(p) = X(Ys(p))$ is a meaningful expression which can be evaluated by applying X after Y. The short version XY of $XYs(p)$ is often used throughout this work.

Another vector valued and maybe the most important operator is the Lie bracket.

Definition A-13. Let $s \in C^\infty(M)$ be a smooth, $\mathbb{R}$ valued function on M and let be given two smooth vector fields, X and Y. The *Lie bracket* of X and Y is a new vector field which is denoted by $[X, Y]$ and defined by

$$[X, Y]s = X(Ys) - Y(Xs). \tag{A-20}$$

The definition in local coordinates follows the vector field

$$[X, Y] = \sum_{j=1}^n \left(\sum_{i=1}^n \left(\frac{\partial Y^j(x)}{\partial x_i} X^i - \frac{\partial X^j(x)}{\partial x_i} Y^i \right) \right) \frac{\partial}{\partial x_j} \tag{A-21}$$

where

$$X = \sum_{i=1}^n X^i(x) \frac{\partial}{\partial x_i}, \qquad Y = \sum_{i=1}^n Y^i(x) \frac{\partial}{\partial x_i}. \tag{A-22}$$

Vector fields are often transformed onto another manifold or onto other place of the same manifold. To do so, two special mappings are to be defined.

Definition A-14. Let $F : M \to N$ be a diffeomorphism i.e. a one-to-one invertible smooth mapping. Let X be a vector field defined on manifold M and let Y be a vector field defined on manifold N (i.e. they lie always in the tangent space of the actual point of the manifold). Let $p \in M$ and $q \in N$ and $q = F(p)$. Then, the *push forward map* $F_* : TM \to TN$ is a vector field in TN defined by

$$(F_* X)_q = F_{*F^{-1}(q)} X_{F^{-1}(q)}. \tag{A-23}$$

Similarly, the *pull back map* $F^* : TN \to TM$ defines a new vector field

$$(F^* Y)_p = (F_{*p})^{-1} Y_{F(p)}. \tag{A-24}$$

Notation $*$ says that F_* and F^* are not mappings between manifolds but mappings between tangent spaces. The relation between pull back and push forward is obtained by $F^* = (F^{-1})_*$. One can easily see that push forward F_* maps the vector fields into the same direction, in other words, if F maps p to q then F_* maps the tangent space of p into the tangent space of q. On contrary, the pull back F^* works in reverse direction, if F maps p to q then F^* maps the tangent space of q into the tangent space of p. Another important property of Lie bracket in relation to mappings are stated in the next proposition.

Proposition A-1. Consider two smooth vector fields X and Y and let $F: M \to N$ be a smooth map. Then $F_*[X, Y] = [F_*X, F_*Y]$.

The proposition says that push forward mapping keeps the Lie bracket between two vector fields.

Lie algebra

Lie groups and vector fields are suitable to carry algebraic structure leading to the basic concept of Lie algebra. This concept evinced also great profit in development of control theory. Now some basic notions are highlighted.

Definition A-15. A *Lie algebra* is a vector space V (over $\mathbb{R}$) if it has a bilinear operator $V \times V \to V$ denoted by $[\cdot, \cdot]$ that satisfies:

1. Skew symmetry: $[f_1, f_2] = -[f_1, f_2]$ for all $f_1, f_2 \in V$.
2. Jacoby identity: $[[f_1, f_2], f_3] + [[f_3, f_1], f_2] + [[f_2, f_3], f_1] = 0$ for all $f_1, f_2, f_3 \in V$.

It is possible that a subspace of V carries also a Lie algebra structure. A subspace $W \subset V$ is a *Lie subalgebra* if $[f_1, f_2] \in W$ for all $f_1, f_2 \in W$. The vector space of all smooth vector fields on manifold M under the Lie bracket operation on vector fields defines an infinite dimensional Lie algebra.

Lie algebra has a strong relation with Lie groups. If X is a vector field on a Lie group G, then X is said to be *left invariant* if $(L_g)_*X = X$. Note, that $L_g h = gh$, $g, h \in G$ was defined in definition A-7 as group operation. On the other hand, $L_g h$ is a mapping from a point h of manifold G (which is in fact more general, a Lie group) to another point $k = gh$. In this term, $(L_g)_*h$ maps the tangent spaces of h into the tangent space of k via push forward map. The expression $(L_g)_*X = X$ can be restated by $T_h L_g X(h) = X(gh)$ for all $g, h \in G$ where $T_h L_g$ corresponds to $(L_g)_*$ that maps the tangent space of h into the tangent space of gh. Let $\mathbb{X}(G)$ be the set of all vector fields on G. If $\mathbb{X}_L(G)$ is the set of left invariant vector fields on G and $X, Y \in \mathbb{X}_L(G)$ then using proposition A-1, one writes

$$L_{g*}[X, Y] = [L_{g*}X, L_{g*}Y] = [X, Y] \tag{A-25}$$

which means that $\mathbb{X}_L(G)$ is a Lie subalgebra of the Lie algebra $\mathbb{X}(G)$. If e is the identity element of G, then $\mathbb{X}_L(G)$ and T_eG is isomorphic as vector spaces (Murray, Li, and Sastry (1994)). It implies that one defines Lie bracket in T_eG with this isomorphism as

$$[\xi_1, \xi_2] = [X_{\xi_1}, X_{\xi_2}](e), \quad \xi_1, \xi_2 \in T_eG, \quad X_{\xi_i} = T_e L_g \xi_i, \quad i = 1, 2. \tag{A-26}$$

This Lie bracket endows T_eG with Lie algebra structure, hence T_eG is the *Lie algebra of* G and is denoted by $\mathcal{L}(G)$. Note, if $x \in G$ and $\bar{\xi} = \{\xi_1, \dots \xi_n\}$, $\xi_i \in T_xG$, $i = 1, \dots, n$ then $\mathcal{L}(G)$ can be replaced also by the notation $\mathcal{L}(\bar{\xi})(x)$. This result allows to study Lie algebra locally at a point since any other point on the Lie group carries the same structure (via isomorphism).

Now, it is useful to obtain the basis of a Lie algebra generated by vector fields. For exact statements, an algebraic machinery is established. Let $\bar{\mathcal{X}}_m = \{\mathcal{X}_1, \dots \mathcal{X}_m\}$ be a sequence of m objects called *indeterminates* .

The name indeterminate reveals that $\mathcal{X}_i$ are not predefined but one puts distinct objects into their place. (The objects will be, not surprisingly, vector fields in the discussion of Lie algebra.) A (formal) *monomial* in $\bar{\mathcal{X}}_m$ is a formal product whose elements are from $\{\mathcal{X}_1, \ldots \mathcal{X}_m\}$. For example if, $m = 2$, $\mathcal{X}_1\mathcal{X}_2\mathcal{X}_1$, $\mathcal{X}_1\mathcal{X}_1\mathcal{X}_2\mathcal{X}_1\mathcal{X}_2 = \mathcal{X}_1^2\mathcal{X}_2\mathcal{X}_1\mathcal{X}_2$ are monomials in $\{\mathcal{X}_1, \mathcal{X}_2\}$. If M_1,M_2 are monomials, then M_1M_2 is defined by concatenation of their elements in order. For example, if $M_1 = \mathcal{X}_1\mathcal{X}_2$, $M_2 = \mathcal{X}_2\mathcal{X}_1$ then $M_1M_2 = \mathcal{X}_1\mathcal{X}_2\mathcal{X}_2\mathcal{X}_1$. Let $\mathcal{A}(\bar{\mathcal{X}}_m)$ denote the algebra generated by $\bar{\mathcal{X}}_m$. Its elements are the (formal) *noncommutative polynomials* in $\bar{\mathcal{X}}_m$ which are finite linear combination of formal monomials. For example, $\mathcal{X}_1 + \mathcal{X}_1\mathcal{X}_2^2 + 4\mathcal{X}_2^2\mathcal{X}_1$ is a noncommutative polynomial. It is easy to see that $1, \mathcal{X}_1, \mathcal{X}_2, \ldots \mathcal{X}_m, \mathcal{X}_1\mathcal{X}_1, \mathcal{X}_1\mathcal{X}_2, \ldots \mathcal{X}_m\mathcal{X}_m, \mathcal{X}_1^3, \ldots \mathcal{X}_2\mathcal{X}_1\mathcal{X}_m, \ldots \mathcal{X}_m^3, \ldots$ are the basis of $\mathcal{A}(\bar{\mathcal{X}}_m)$. The associative algebra above is also a Lie algebra . Lie algebra is denoted by $\mathcal{L}(\bar{\mathcal{X}}_m)$ and its elements are the *Lie polynomials* in $\bar{\mathcal{X}}_m$, i.e. those $\mathcal{A}_i \in \mathcal{A}(\bar{\mathcal{X}}_m)$ that can be written as linear combinations of Lie brackets of the elements of $\bar{\mathcal{X}}_m = \{\mathcal{X}_1, \ldots \mathcal{X}_m\}$. Note, that Lie bracket is defined as $[\mathcal{X}_1, \mathcal{X}_2] = \mathcal{X}_1\mathcal{X}_2 - \mathcal{X}_2\mathcal{X}_1$. It may seem somewhat strange to apply multiplication to vector fields but note that (A-20) defines exactly such an operation (on $\mathbb{R}$ valued functions which can be the identity map in simplest case). As examples, $\mathcal{X}_1\mathcal{X}_2 \notin \mathcal{L}(\bar{\mathcal{X}}_2)$, $\mathcal{X}_1\mathcal{X}_2 - \mathcal{X}_2\mathcal{X}_1 \in \mathcal{L}(\bar{\mathcal{X}}_2)$, $2\mathcal{X}_1 + 4\mathcal{X}_2 - [[\mathcal{X}_2, \mathcal{X}_1], [\mathcal{X}_2, [\mathcal{X}_2, \mathcal{X}_1]] \in \mathcal{L}(\bar{\mathcal{X}}_2)$.

The algebra of *formal noncommutative power series* denoted by $\hat{\mathcal{A}}(\bar{\mathcal{X}}_m)$ are the not necessarily finite, linear combination of monomials. Similarly, the Lie algebra of *Lie series* denoted by $\hat{\mathcal{L}}(\bar{\mathcal{X}}_m)$ are the not necessarily finite, linear combination of Lie monomials. In contrast to Lie polynomials, Lie series can be infinite. For instance,

$$\mathcal{X}_1 + [\mathcal{X}_1, \mathcal{X}_2] + 2\mathcal{X}_1^2 + 2[\mathcal{X}_1, \mathcal{X}_2]^2 + 3\mathcal{X}_1^3 + 3[\mathcal{X}_1, \mathcal{X}_2]^3 + \cdots$$

is an element of $\hat{\mathcal{L}}(\bar{\mathcal{X}}_2)$.

The nilpotent associative and nilpotent Lie algebra of order k are denoted by $\mathcal{A}_k(\bar{\mathcal{X}}_m)$ (or $\mathcal{A}_k^m$) and $\mathcal{L}_k(\bar{\mathcal{X}}_m)$ (or $\mathcal{L}_k^m$), respectively. They are the same as $\mathcal{A}(\bar{\mathcal{X}}_m)$ and $\mathcal{L}(\bar{\mathcal{X}}_m)$ except that all the monomials are killed at degree $k+1$. For example, a basis of $\mathcal{L}_2(\bar{\mathcal{X}}_m)$ is defined by $\mathcal{X}_1, \mathcal{X}_2$ and $[\mathcal{X}_1, \mathcal{X}_2]$. (Note, that $[\mathcal{X}_2, \mathcal{X}_1] = -[\mathcal{X}_1, \mathcal{X}_2]$ so it can be expressed in the given basis.) Similarly, $\hat{\mathcal{A}}_k(\bar{\mathcal{X}}_m)$ and $\hat{\mathcal{L}}_k(\bar{\mathcal{X}}_m)$ denote the power and Lie series which are built from the linear combinations of the elements of $\mathcal{A}_k(\bar{\mathcal{X}}_m)$ and $\mathcal{L}_k(\bar{\mathcal{X}}_m)$, respectively.

It is harder to find a basis of a Lie algebra $\mathcal{L}_k^m$ than the basis of associative algebra. The main reason is that a basis of Lie algebra does not include all the Lie monomials since some of them can be dropped using the properties of skew symmetry and Jacoby identity of Lie bracket.

Definition A-16. Let $\{g_1, \ldots, g_m\}$ be a set of vector fields and define the *degree of Lie product* (bracket) as

$$\begin{aligned} l(g_i) &= 1 \quad i = 1, \ldots, m \\ l([X,Y]) &= l(X) + l(Y), \end{aligned} \tag{A-27}$$

where X and Y may be Lie products, as well. A *Philip Hall (or P. Hall) basis* is an ordered set of Lie Products $H = \{B_i\}$ satisfying:

1. $g_i \in H$, $i = 1, \ldots, m$
2. If $l(B_i) < l(B_j)$, then $B_i < B_j$

3. $[B_i, B_j] \in H$ if and only if
   a) $B_i, B_j \in H$ and $B_i < B_j$, and
   b) either $B_j = g_k$ for some k or $B_j = [B_l, B_r]$ with $B_l, B_r \in H$ and $B_l \leq B_i$

As an example, the P. Hall basis of $\mathcal{L}_2^4$ (i.e. the basis of $\mathcal{L}(\bar{\mathcal{X}}_2)$ up to degree 4) is: $\mathcal{X}_1, \mathcal{X}_2$, $[\mathcal{X}_1, \mathcal{X}_2]$, $[\mathcal{X}_1, [\mathcal{X}_1, \mathcal{X}_2]]$, $[\mathcal{X}_2, [\mathcal{X}_1, \mathcal{X}_2]]$, $[\mathcal{X}_1, [\mathcal{X}_1, [\mathcal{X}_1, \mathcal{X}_2]]]$, $[\mathcal{X}_2, [\mathcal{X}_1, [\mathcal{X}_1, \mathcal{X}_2]]]$, $[\mathcal{X}_2, [\mathcal{X}_2, [\mathcal{X}_1, \mathcal{X}_2]]]$.

Why the Lie brackets are so important

One cannot overemphasize the importance of Lie bracket $[X, Y]$ in differential-geometry and control theory. The fact in the background is twofold. First, the vector field produced by Lie bracket has a special property, namely, that it may be independent from both X and Y. Secondly, the Lie bracket $[X, Y]$ in equation (A-20) measures the commutation between X and Y. A possible way to show it proceeds as follows.

Definition A-17. Let G be a Lie group and $g, h \in G$. The *conjugation map associated with* g is an action of G, denoted by $C_g(h)$, $C_g(h) : G \to G$:

$$C_g(h) = ghg^{-1} = R_{g^{-1}} L_g h. \tag{A-28}$$

The derivative of the conjugation map at the identity I is the *Adjoint map*, denoted by Ad_g where $Ad_g : \mathcal{L}(G) \to \mathcal{L}(G)$

$$Ad_g(\xi) = d(C_g)(\xi) = d(R_g^{-1} L_g)(\xi) \tag{A-29}$$

for any $\xi \in \mathcal{L}(G)$, $g \in G$. In frequent use, $G \subset GL(n, C)$ which involves $Ad_g(\xi) = g\xi g^{-1}$.

The Lie algebra $\mathcal{L}(G)$ is a vector space and Ad_g is an element of the Lie group of bijective linear maps from $\mathcal{L}(G)$ into itself, denoted by $Aut(\mathcal{L}(G))$ and called *automorphisms*. The Lie algebra (or differential) of the Lie group $Aut(\mathcal{L}(G))$ is the space $End(\mathcal{L}(G))$ called *endomorphisms* which is the space of *all* linear maps from $\mathcal{L}$ into itself. For any $\xi_1, \xi_2 \in \mathcal{L}(G)$, the map $ad_{\xi_1} : \mathcal{L}(G) \to \mathcal{L}(G)$,

$$ad_{\xi_1}(\xi_2) = [\xi_1, \xi_2], \qquad ad_{\xi_1} \in End(\mathcal{L}(G)) \tag{A-30}$$

is called *adjoint map* . The following reveals that adjoint map ad can be considered as the differential of Adjoint map Ad. For, let $\gamma(t) \in G$ is a curve in G with $\gamma(0) = I$ and $\gamma'(0) = \xi_1$. Then,

$$\frac{d}{dt} Ad_{\gamma(t)}(\xi_2) = ad_{\xi_1}(\xi_2). \tag{A-31}$$

To see this, it is known that any $g \in G$ can usually be produced as the exponential $exp(X) = \Psi(X, 1)$ for some $X \in \mathcal{L}(G)$ establishing a connection between $Ad_g \in Aut(\mathcal{L}(G))$ and $ad_X \in End(\mathcal{L}(G))$. If $X, Y \in \mathcal{L}(G)$, then

$$\begin{aligned} Ad_{exp(X)} Y &= \exp(X) Y \exp(-X) \qquad \text{(A-32)} \\ &= Y + (XY - YX) + \frac{1}{2!}\{X(XY - YX) - (XY - YX)X\} \end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{1}{3!}\{(X(X(XY-YX)-(XY-YX)X)-(X(XY-YX)-(XY-YX)X)Y\}+\cdots \\
&= \quad Y+[X,Y]+\frac{1}{2!}[X,[X,Y]]+\frac{1}{3!}[X,[X,[X,Y]]]+\cdots \\
&= \quad Y+ad_XY+\frac{1}{2!}ad_X^2Y+\frac{1}{3!}ad_X^3Y+\cdots \\
&= \quad exp(ad_X)Y.
\end{aligned}
$$

Both $Ad_{exp(t\,X)}$ and $\exp(t\,ad_X)$ provide one-parameter subgroups of the Lie group $Aut(\mathcal{L}(G))$ having the tangent vector ad_X for $t=0$ and therefore for all t. The expression (A-32) says how much X and Y fails to be commutative over the exponential. If they commute, $[X,Y]=0$ and $Ad_{exp(X)}Y=Y$.

The commutation can be measured also by *Cambell-Baker-Hausdorff formula*

$$
\begin{aligned}
\Phi(X,\epsilon)\circ\Phi(Y,\epsilon) &= \Phi(X+Y+1/2[X,Y]+1/12([X,[X,Y]]-[Y,[X,Y]]+\cdots,\epsilon) \qquad \text{(A-33)}\\
&= \Phi(\sum_{m=1}^{\infty}\sum\frac{(-1)^{m-1}ad_Y^{q_m}ad_X^{p_m}\cdots ad_Y^{q_1}ad_X^{p_1}}{m(\sum_{i=1}^{m})(p_i+q_i)\prod_{i=1}^{m}(p!q!)}).
\end{aligned}
$$

where the inner sum contains all the m-tuples of pairs of nonnegative integers (p_i,q_i) satisfying $p_i+q_i>0$ and adopting the convention $ad_x=X$ or $ad_y=Y$ for the final terms in the multiplication. This form can be used for approximation of a flow along Lie bracket up to second order:

$$
\Phi([X,Y],\epsilon^2)\approx\Phi(-Y,\epsilon)\circ\Phi(-X,\epsilon)\circ\Phi(Y,\epsilon)\circ\Phi(X,\epsilon)(x) \qquad \text{(A-34)}
$$

In the most cases, one adopts only the second order approximation. This feature is intensively explored in control theory as well as in this dissertation.

Distributions

Let $f_1(x),\ldots,f_m(x)$ be m vector fields. They span a vector space $span(f_1(x),\ldots f_m(x))$ at each point x on an open set $U\subset\mathbb{R}^n$ which is called *distribution* and denoted by

$$
\Delta = span(f_1,\ldots f_m). \qquad \text{(A-35)}
$$

The distribution assigns a vector space to each point. The dimension of a distribution at a point x is the dimension of the subspace Δ. A distribution Δ is *nonsingular* on an open set U if its dimension is constant at each $x\in U$. A distribution is *involutive* if $f_1\in\Delta$, $f_2\in\Delta$ involves that $[f_1,f_2]\in\Delta$. The *annihilator of* Δ is a codistribution spanned by set of all covectors w^* which annihilates all vector in Δ

$$
\Delta^{\perp} = span(\{w^*(\mathbb{R}^n)^* : \langle w^*,v\rangle = 0 \text{ for all } v\in\Delta(x)\}) \qquad \text{(A-36)}
$$

where covectors w^*, $(\mathbb{R}^n)^*$ and codistributon are the dual objects of vectors, $\mathbb{R}^n$ and distribution, respectively. The *involutive closure of a distribution* $\Delta=span(f_1,\ldots f_m)$ is denoted by $\bar{\Delta}$ and defined by the closure of Δ under Lie bracketing. Note, that the involutive closure $\bar{\Delta}$ is the smallest Lie subalgebra of $\mathbb{X}(M)$ which contains $f_1,\ldots f_m$.

Index

Bibliography

Abraham, R., J. E., Marsden, and T. Ratiu (1988). *Manifolds, tensor analysis and application.* Springer Verlag New York Inc.

Ahmadi, M. and M. Buehler (1997). Stable control of a simulated one-legged running robot with hip and leg compliance. *IEEE Transactions on Robotics and Automation*, 96–104.

Berns, K., J. Albiez, V. Kepplin, and S. Schmalenbach (2001). Control of six-legged robot fluiic muscles. In *Proceeding of 10th IEEE International Conference on Advanced Robotics*, pp. 467–472.

Bicchi, A. and V. Kumar (2000). Robotic grasping and contact: A review. In *Proceedings of the Conference on Robotics and Automation*, pp. 348–353.

Bicchi, A., A. Marigo, and D. Prattichizzo (1997). Robotic dexterity via nonholonomy. In B. Siciliano and K. Valavanis (Eds.), *Control Problems in Robotics and Automation*, Berlin, pp. 35 – 49. Springer-Verlag.

Bicchi, A., A. Marigo, and D. Prattichizzo (1998). Dexterity through rolling: Manipulation of unknown objects. In *Proceedings of the Conference on Robotics and Automation*, pp. 1583–1568.

Bullo, F. (1999). *Nonlinear control of mechanical systems: A Riemannian geometry approach.* Ph. D. thesis, California Institute of Technology.

Bullo, F. and K. M. Lynch (2001). Kinematic controllability for decoupled trajectory planning in underactuated mechanical systems. *IEEE Transactions on Robotics and Automation 17*(4), 402–412.

Bullo, F. and M. Zefran (2002a). Modeling and controllability for a class of hybrid mechanical systems. *To appear in IEEE Transactions on Robotics and Automation, http://motion.csl.uiuc.edu/ bullo/papers/.*

Bullo, F. and M. Zefran (2002b). On mechanical control systems with nonholonomic constraints and symmetries. *Systems and Control Letters 45*(2), 133–143.

Cherif, M. and K. K. Gupta (1997). Planning quasi-static motions for re-configuring objects with a multi-fingered robotic hand. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 986–991.

Cherif, M. and K. K. Gupta (2001). Global planning for dexterous reorientation of rigid objects; finger tracking with rolling and sliding. *The International Journal of Robotics Research 20*(1), 57–84.

Chevallier, D. and S. Payandeh (1995). On manipulation of object in three-fingered grasp. *Transaction of ASME, Journal of Mechanical Design 117*, 561–565.

Cole, A. A., P. Hsu, and S. S. Sastry (1992). Dynamic control of sliding by robot hands for regrasping. *IEEE Trans. Robotics and Automation 8*(2), 151–165.

Collins, J. J. and I. Stewart (1993). Hexapodal gaits and coupled nonlinear oscillator models. *Biological Cybernetics* (68), 287–298.

Cruse, H. C., C. Bartling, G. Cymbalyuk, J. Dean, and M. Deifert (1995). A modular neural net for controlling a six legged walking system. *Biological Cybernetics* (72), 421–430.

Fliess, M., J. Lévine, P. Martin, and P. Rouchon (1995). Flatness and defect of nonlinear systems: introductory examples. *The International Journal of Control 61*(6), 1327–1361.

Goodwine, B. (1998a). *Control of stratified systems with robotic applications.* Ph. D. thesis, California Institute of Technology.

Goodwine, B. (1998b). Stratified motion planning with application to robotic finger gaiting. *IFAC World Congress*.

Goodwine, B. (1999). Stratified nonlinear motion planning with application to robotic finger gaiting. 10^{th} *congress on the theory of machines and mechanism.*

Goodwine, B. and J. W. Burdick (2001). Controllability of kinematic control systems on stratified configuration spaces. *IEEE Transaction on Automatic Control 46*(3), 358–368.

Goresky and Macpherson (1980). *Stratified morse theory.* Springer-Verlag.

Grupen, R. A., M. Huber, J. A. C. Jr., and K. Souccar (1995). A basis for distributed control of manipulation tasks. *Special Track on Intelligent Robotic Systems, IEEE Expert 10*(2), 9–14.

Han, L., Y. S. Guan, Z. Li, Q. Shi, and J. C. Trinkle (1997). Dextrous manipulation with rolling contacts. In *Proceedings of the Conference on Robotics and Automation*, pp. 992–997.

Han, L. and J. C. Trinkle (1998a). Dextrous manipulation by rolling and finger gaiting. In *Proceedings of the Conference on Robotics and Automation*, pp. 730–735.

Han, L. and J. C. Trinkle (1998b). Object reorientation with finger gaiting. In *Proceedings of 2nd IMACS International Multiconference: Computational Engineering in Systems Applications (CESA'98).*

Harmati, I. and B. Kiss (2001). Motion planning algorithms for stratified kinematic systems with application to hexapod robot. *Acta Cybernetica, Szeged, Hungary 15*(2), 225–240.

Harmati, I., B. Kiss, and B. Lantos (2000). Motion planning for legged robots. In *Proceedings of the INT-COM'00 Tempus Symposium on Intelligent Systems in Control and Measurement*, Veszprém, Hungary, pp. 34–39.

Harmati, I., B. Kiss, and B. Lantos (2002). Two motion planning approaches for six-legged robots. In *Proceedings of the 3rd International Workshop on Robot Motion and Control (RoMoCo '02)*, Bukowy Dworek, Poland, pp. 87–92. ISBN 1 86058 380 6.

Harmati, I. and B. Lantos (2002). Rétegezett irányítás s manipuláció (in hungarian). In *Proceedings of conference Microcad 2002*, Miskolc, Hungary, pp. 95–100.

Harmati, I., B. Lantos, and S. Payandeh (1999). Improved stratified control for hexapod robots and object manipulation with finger relocation. *Periodica Polytechnica, Budapest, Hungary 43*(3), 163–175.

Harmati, I., B. Lantos, and S. Payandeh (2000a). Extensions of nonlinear control concept to object manipulation with finger relocation. In *Proceedings of the 6th Int. IFAC Symp. on Robot Control*, pp. 223–228.

Harmati, I., B. Lantos, and S. Payandeh (2000b). Stratified control of multi-agent manipulation for objects with nonsmooth surfaces. In *Proceedings of the 4th IEEE international conference on Intelligent Engineering Systems (INES)*, pp. 59–62.

Harmati, I., B. Lantos, and S. Payandeh (2001). On object manipulation methods using finger relocation. In *Proceedings of the 10th IEEE International Conference on Advanced Robotics (ICAR'01)*, pp. 613–618.

Harmati, I., B. Lantos, and S. Payandeh (2002a). Completely involutive representations for holonomic stratified motion planning. In *Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems (INES)*, pp. 35–40.

Harmati, I., B. Lantos, and S. Payandeh (2002b). Holonomic and nonholonomic stratified motion planning along decomposed trajectory for walking robots. In P. Bideud and F. B. Amar (Eds.), *Proceedings of 5th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines - CLAWAR 2002*, Paris, France, pp. 579–586. ISBN 1 86058 380 6.

Harmati, I., B. Lantos, and S. Payandeh (2002c). Motion planning along decomposed trajectory for holonomic stratified systems. In *Proceedings of the 6th IEEE International Conference on Intelligent Engineering Systems (INES)*, pp. 81–86.

Harmati, I., B. Lantos, and S. Payandeh (2002e). Nonholonomic stratified motion planning along decomposed reference trajectories. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, Lausanne, Switzerland, pp. 2341–2346.

Harmati, I., B. Lantos, and S. Payandeh (2002d). Nonholonomic stratified motion planning using decomposed trajectories. In *Proceedings of 8th IEEE International Conference on Methods and Models in Automation and Robotics (MMAR)*, Szczecin, Poland, pp. 953–958.

Harmati, I., B. Lantos, and S. Payandeh (2002f). On fitted stratified and semi-stratified geometric manipulation planning with fingertips relocations. *The International Journal of Robotics Research 21*(5-6), 489–510.

Harmati, I., B. Lantos, and S. Payandeh (2002g). Semi-straified motion planning of multi-agent manipulation. *Journal of Advanced Computational Intelligence, Fuji Press Ltd, Tokyo 5*(5), 248–256.

Hong, J., G. Lafferriere, B. Mishra, and X. Tang (1990). Fine manipulation with multifinger hand. In *Proceedings of the Conference on Robotics and Automation*, pp. 1568–1573.

Huber, M. and R. A. Grupen (2002). Robust finger gaits from closed-loop controllers. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, Lausanne, Switzerland, pp. 1578–1584.

Hyon, S. and T. Emura (2002). Quasi-periodic gaits of passive one-legged hopper. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, Lausanne, Switzerland, pp. 2625–2630.

Isidori, A. (1996). *Nonlinear control systems.* Springer Verlag London Limited.

Kelly, S. and R. Murray (1995). Geometric phases and robotic locomotion. *J. Robotic Systems 12*(6), 417–431.

Kiss, B. (2001). *Motion planning and control of a class of flat and Liouvillian mechanical systems.* Ph. D. thesis, Budapest University of Technology and Economics.

Kiss, B., J. Lévine, and B. Lantos (1999). Trajectory planning for dextrous manipulation with rolling contacts. In *Proceedings of the 1999 IEEE Conference on Decision and Control*, pp. 2118–2119.

Koditschek, D. and M. Buehler (1991). Analysis of a simplified hopping robot. *The International Journal of Robotics Research 10*(6), 587–605.

Lafferriere, G. and H. Sussmann (1991). Motion planning for controllable systems without drift. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1148–1153.

Lang, S. (1999). *Fundamentals of differential geometry.* Springer-Verlag.

Lantos, B. (1997). *Robotok irányítása (In Hungarian).* Akadémiai kiadó, Budapest.

Lantos, B. (2003). *Irányítási rendszerek elmélete és tervezése II. (In Hungarian).* Akadémiai kiadó, Budapest.

Leveroni, S. and K. Salisbury (1995). Re-orienting object with a robot hand using grasp gaits. *International Symposium on Robotics Research*, 2–15.

Lewis, A. D. (1995). *Aspects of geometric mechanics and control of mechanical systems.* Ph. D. thesis, California Institute of Technology.

Ma, S., T. Tomiyama, and H. Wada (2002). Omni-directional walking of a quadruped robot. In *Proceedings of 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, Lausanne, Switzerland, pp. 2605–2612.

Marigo, A. and A. Bicchi (2000). Rolling bodies with regular surface: Controllability theory and applications. *Trans. Autom. Control. 45*(9), 1586–1599.

McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research 9*(2), 62–82.

Monaco, S. and D. Normand-Cyrot (1992). An introduction to motion planning under multirate digital control. In *Proceedings of the 31st Conference on Decision and Control*, pp. 1780–1785.

Montana, D. (1988). The kinematics of contact and grasp. *The International Journal of Robotics Research 7*(3), 17–25.

Murray, R. and S. Sastry (1993). Nonholonomic motion planning: steering using sinusoids. *IEEE Transactions on Automatic Control 38*, 700–716.

Murray, R. M., Z. Li, and S. S. Sastry (1994). *Mathematical Introduction to Robotic Manipulation.* CRC Press.

Nijmeijer, H. and A. J. van der Schaft (1990). *Nonlinear dynamical control systems.* Springer Verlag New York Inc.

Olver, P. (1993). *Application of Lie groups to differential equations.* Springer-Verlag.

Omata, T. and M. A. Farooqi (1996). Regrasps by a multifingered hand based on primitives. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2774–2780.

Ono, K., R. Takahashi, and T. Shimada (2001). Self-excited walking of a biped mechanism. *The International Journal of Robotics Research 20*(12), 953–966.

Ostrowski, J. and J. Burdick (1993). Designing feedback algorithms for controlling the periodic motion of legged robots. In *Proccedings of the IEEE International Conference on Robotics and Automation*, Volume 2, pp. 260–266.

Payandeh, S. (1997). Planning controlled slips in dexterous manipulation. In *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 2248–2252.

Pfeiffer, F., J. Eltze, and H. J. Weidemann (1995). Six-legged technical walking considering biological principles. *Robotics and Autonomous Systems* (14), 223–232.

Pratt, J., C. Chew, A. Torres, P. Dilworth, and G. Pratt (2001). Virtual model control: An intiutive approach for bipedal locomotion. *The International Journal of Robotics Research 20*(2), 129–143.

Raibert, M. H. (1986). *Legged robots that balance.* The MIT Press, Cambridge, Massachusetts.

Rus, D. (1999). In-hand dexterous manipulation of piecewise-smooth 3-d objects. *The International Journal of Robotics Research 18*(4), 355–381.

Salisbury, J. K. and M. T. Mason (1985). *Robot Hands and the Mechanics of Manipulation.* MIT Press.

Sastry, S. (1999). *Nonlinear systems: analysis, stability and control.* Springer-Verlag.

Schmitt, J. and P. Holmes (2000a). Mechanical models for insect locomotion: Dynamics and stability in the horizontal plane i: Theory. *Biological Cybernetics 83*(6), 501–515.

Schmitt, J. and P. Holmes (2000b). Mechanical models for insect locomotion: Dynamics and stability in the horizontal plane ii: Application. *Biological Cybernetics 83*(6), 517–527.

Serre, J. P. (1992). *Lie Algebras and Lie groups*. Springer-Verlag.

Song, S. and K. Waldron (1989). *Machines that walk: The Adaptive Suspension Vechicle*. MIT Press.

Sontag, E. D. (1998). *Mathematical control theory: deterministic finite dimensional systems*. Springer Verlag New York Inc.

Spenneberg, D. and F. Kirchner (2001). An approach towards autonomous outdor walking robots. In *Proceedings of 10th IEEE International Conference on Advanced Robotics*, pp. 485–491.

Sussmann, H. J. (1992). New differential geometric methods in nonholonomic path finding. In A. Isidori and T. J. Tarn (Eds.), *Systems, Models, and Feedback: Theory and Applications*, pp. 365–384. Birkhäuser, Boston.

Sussmann, H. J. and W. Liu (1991). Limits of highly oscillatory controls and the approximation of general paths by admissible trajectories. In *Proceedings of the 30th IEEE Conference on Decision and Control*, pp. 437–442.

Sussmann, H. J. and W. Liu (1993). Lie bracket extensions and averaging: the single-bracket case. In Z. X. Li and J. F. Canny (Eds.), *Nonholonomic Motion Planning*, pp. 109–148. Kluwer Academic Publishers.

Varadarajan, V. (1994). *Lie Groups, Lie Algebras, and Their Representations*. Springer-Verlag.

Vass, G., S. Payandeh, and B. Lantos (1999a). Application of simulated annealing for object manipulation with multiple agents. In *Proceedings of the Second IASTED International Conference, Control and Applications*, pp. 566–571.

Vass, G., S. Payandeh, and B. Lantos (1999b). On controlled manipulation of objects within multiple dexterous agents. 10^{th} *congress on the theory of machines and mechanism 3*, 1103–1108.

Vendittelli, M., G. Oriolo, and J. P. Laumond (1999). Steering nonholonomic systems via nilpotent approximations: The general two trailer system. In *Proceedings of the Conference on Robotics and Automation*, pp. 823–829.

Xu, W. L., B. L. Ma, and S. K. Tso (1999). Curve fitting approach to motion planning of nonholonomic chained systems. In *Proceedings of the Conference on Robotics and Automation*, pp. 811–816.

Yoshikawa, T. (2000). Control algorithm for grasping and manipulation by multifingered robot hands using virtual truss model represantation of internal force. In *Proceedings of the Conference on Robotics and Automation*, pp. 369–376.

Zefran, M., J. P. Desai, and V. Kumar (1996). Continuous motion plans for robotic systems with changing dynamic behavior. *Workshop on Algorithmic Foundations of Robotics (WAFR).*

Zeglin, G. (1999). *The bow leg hopping robot*. Ph. D. thesis, Carnegie Mellon University.