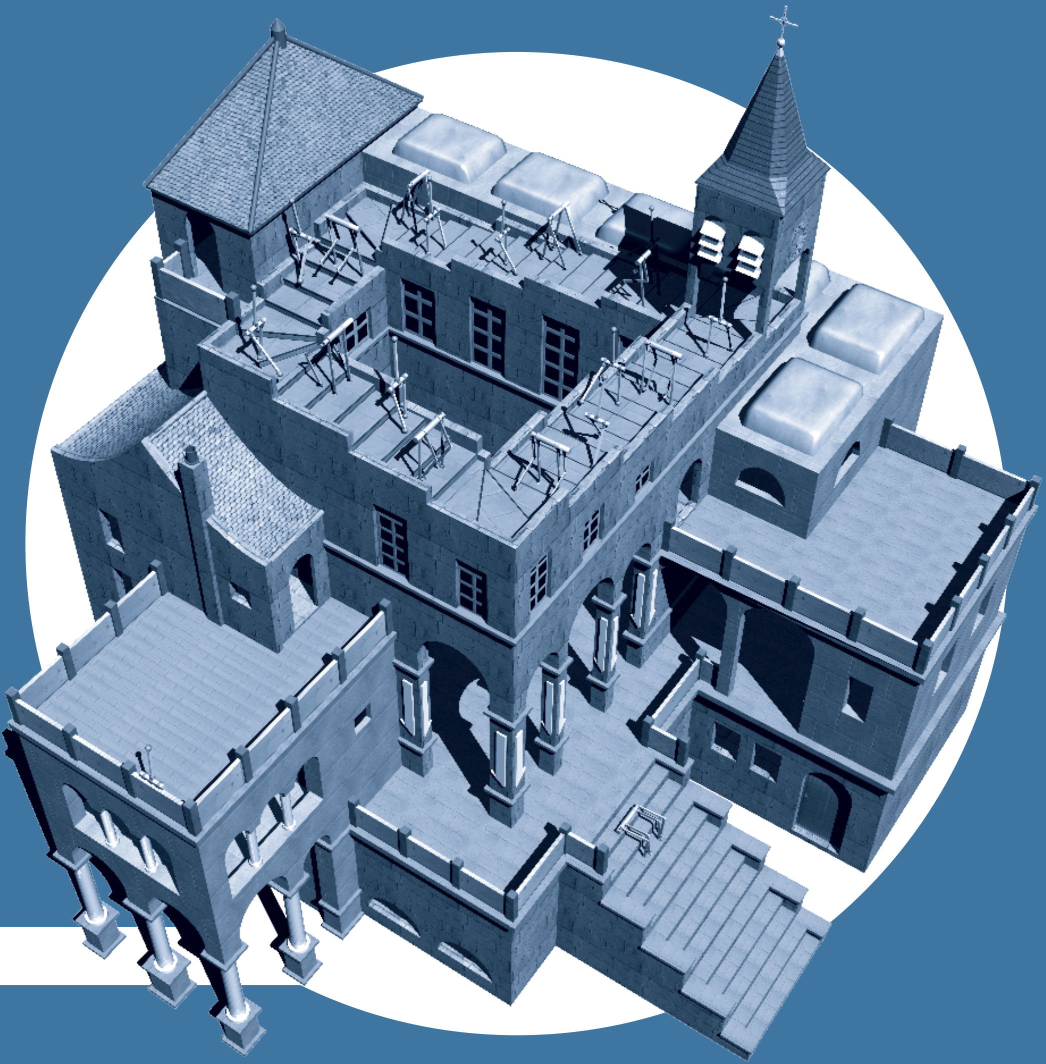
Port-Based Modeling and Control for Efficient Bipedal Walking Robots



Vincent Duindam

The research described in this thesis has been conducted at the Department of Electrical Engineering, Math, and Computer Science at the University of Twente, and has been financially supported by the European Commission through the project Geometric Network Modeling and Control of Complex Physical Systems (GeoPlex) with reference code IST-2001-34166.

The research reported in this thesis is part of the research program of the Dutch Institute of Systems and Control (DISC). The author has succesfully completed the educational program of the Graduate School DISC.

ISBN 90-365-2318-4

The cover picture of this thesis is based on the work 'Ascending and Descending' by Dutch graphic artist M.C. Escher (1898–1972), who, in turn, was inspired by an article by Penrose & Penrose (1958).

Printed by PrintPartners Ipskamp, Enschede, The Netherlands

PORT-BASED MODELING AND CONTROL FOR EFFICIENT BIPEDAL WALKING ROBOTS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. W.H.M. Zijm,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op vrijdag 3 maart 2006 om 13.15 uur

door

Vincent Duindam
geboren op 21 oktober 1977
te Leiderdorp, Nederland

Dit proefschrift is goedgekeurd door:

Prof. dr. ir. S. Stramigioli, promotor
Prof. dr. ir. J. van Amerongen, promotor

Samenvatting

Lopende robots zijn complexe systemen, vanwege hun niet-lineaire dynamica en interactiekrachten met de grond. Traditionele regelmethoden, gebaseerd op het volgen van een referentiesignaal, kunnen wel worden toegepast, maar gebruiken vaak een grote hoeveelheid energie. Lopen op zich vraagt weinig energie, en er bestaan veel energiezuinige lopende robots die dat bewijzen. Jammer genoeg is de regeling van veel van deze robots gebaseerd op ingenieursintuïtie en *ad hoc* oplossingen, en ontbreken algemene analyse- en regelmethoden.

Dit proefschrift presenteert een raamwerk voor het modeleren, analyseren, en efficiënt regelen van lopende robots. Een poort-Hamiltoniaanse systeembeschrijving maakt het mogelijk om de dynamica van algemene rigide mechanismen and hun interactie met de grond te beschrijven. De structuur van deze modellen vormt de basis om algemene analyse- en regelmethoden te ontwikkelen.

Het modeleringsraamwerk is een uitbreiding van bekende modeleringsmethoden naar een brede klasse van rigide mechanismen met configuratieruimte beschreven door een willekeurige combinatie van Euclidische delen, Lie groep delen (zoals balgewrichten) en niet-holonome delen (zoals niet-slippende wielen). Het afleiden van de modelvergelijkingen is een systematisch en modulair proces, en dus geschikt voor software-implementatie. Twee soorten 3D contactmodellen worden beschreven: een zacht contactmodel, beschreven door een veer en een demper – vooral geschikt voor simulatie, en een hard contactmodel, beschreven door een impulsprojectie tijdens de impact – vooral geschikt voor analyse. Alle resultaten zijn gebaseerd op coördinaat-vrije concepten en beschrijvingen.

Met behulp van de structuur van de modellen wordt het probleem om efficiënte loopritmes te vinden geformuleerd als een numeriek optimalisatieprobleem. Op deze manier kunnen niet alleen de gewrichtsbewegingen maar ook de mechanische structuur van een robot worden geoptimaliseerd. Ter illustratie worden de meest efficiënte loopritmes uitgerekend voor drie verschillende robots. Hieruit blijkt, dat de loopsnelheid van een simpele planaire passieve robot op een heuvel kan worden veranderd door de massaverdeling aan te passen, en dat een driedimensionale robot met bovenlichaam het meest efficiënt loopt als de massa van het bovenlichaam zo laag mogelijk wordt geplaatst.

Tenslotte worden drie efficiënte regelmethoden besproken. De eerste methode gebruikt de berekende optimale trajecten om nieuwe coördinaten te definiëren, die expliciet de prestatie uitdrukken. De uiteindelijke regelaar is vermogenscontinu, volgt de trajecten asymptotisch, en reageert alleen om verstoringen te compenseren, dus niet tijdens de nominale loopbeweging. De tweede regelmethode stabiliseert het loopgedrag van een experimentele robot met knieën door middel van een enkele PD regelaar in het heupgewricht van de robot. De derde regelmethode gebruikt voetplaatsing om de robuustheid van een driedimensionale lopende robot te vergroten, en om de robot een referentiepad te laten volgen.

Summary

Walking robots are complex systems because of their nonlinear dynamics and interaction with the ground. Although traditional control methods, based on the tracking of a reference signal, can be applied, they generally require a significant amount of energy. On the other hand, research has shown that walking in itself requires little energy, and many experimental robots have been built that walk with high efficiency. General analysis and control tools for such efficient walkers, however, are lacking, and many results are based on engineering intuition and *ad hoc* solutions.

This thesis aims to provide a framework for modeling, analysis, and efficient control of walking robots. The framework uses a port-Hamiltonian system description to express the dynamics of rigid mechanisms and their interaction with the ground. The structure of the resulting models forms the basis for the development of general analysis and control techniques.

The proposed framework extends well-known modeling methods to a broad class of rigid mechanisms with a configuration space described by any combination of Euclidean components, Lie group/algebra components (such as ball joints), and nonholonomic components (such as non-slipping wheels). The derivation of the corresponding model equations is a systematic, modular process, and hence suitable for software implementation. Two different 3D contact models are presented: a compliant contact model, described by a spatial spring/damper and mainly suitable for simulation, and a rigid contact model, characterized by a momentum projection on impact and mainly suitable for analysis. All results are based on coordinate-free concepts and descriptions.

Using the structure of the models, the problem of finding efficient walking gaits is cast as a numerical optimization problem. This setting allows one to optimize not only the joint trajectories but also the mechanical structure of a walking robot. The approach is illustrated by computing the most efficient gaits for three different walking robots. It is shown how the walking speed of a simple planar passive (*i.e.* unactuated) robot on a slope can be changed by adjusting its mass distribution, and how a three-dimensional robot with a trunk walks most efficiently if the mass on the trunk is located as low as possible.

Finally, three control techniques for efficient walking are presented. The first control technique uses the computed optimal trajectories to define new coordinates that explicitly reveal the tracking performance. The resulting controller is power-continuous, tracks the trajectory asymptotically, and acts only to compensate for disturbances – not during nominal, natural walking. The second control technique stabilizes the walking behavior of a kneed experimental robot by means of a single PD controller on the hip joint. The third control technique uses foot placement to increase the robustness of a three-dimensional walking robot, and to control it to follow a reference path.

Contents

Chapter 1

Introduction

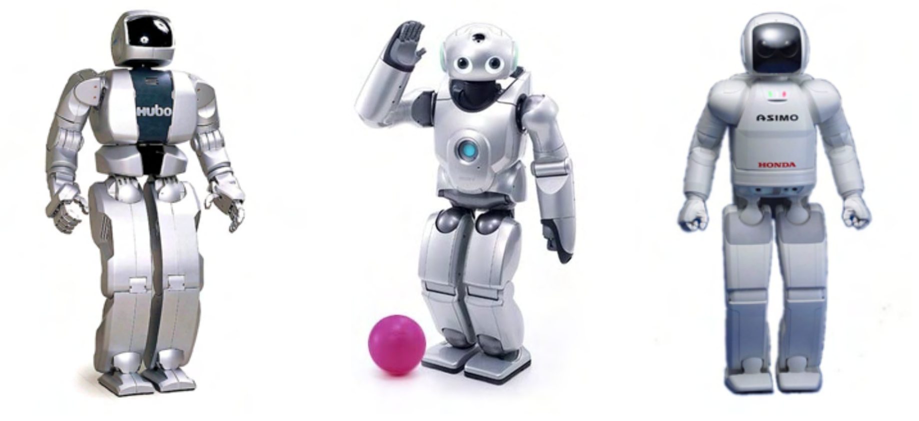
The main goal of engineering research is to find general methods to build real-world physical objects that solve real-world problems. The real-world problem studied in this thesis is the energy-efficient control of walking robots, and the general method that is used is based on the port-Hamiltonian framework for modeling and control.

1.1 Walking robots

1.1.1 Humanoid robots

Robots are cool. Whether it is their smooth controlled motion, the detailed precision mechanics, or the appearance of intelligence, there is something that attracts people. In the last decades, industry has realized this and has begun building and selling robots for personal and entertainment purposes. An important, but so far rather expensive market segment is being filled by humanoid robots: robots that are designed in the image of humans. Though still in their relatively infant stages, current humanoid robots already have a human-like appearance and behavior, see for example the robots in Figure 1.1: Asimo (Sakagami et al. 2002) by Honda, Qrio (Geppert 2004) by Sony, and Hubo (Kim et al. 2005) by the Korea Advanced Institute of Science and Technology (KAIST).

Reasons for building robots with human characteristics are manifold. From a practical point of view, if robots are ever to work inside a house that was built for humans, then obviously they need to be able to move and reach where humans can move and reach (walk up the stairs, grab something from a shelf), and these requirements are trivially met if the robots are shaped and articulated like humans. For similar reasons, NASA is developing Robonaut (Ambrose et al. 2000), an astronaut-size robot with arms, hands, fingers, a head, and a torso, to assist in space missions, both indoors and outdoors during space walks, without the need



(a) KAIST's Hubo. (b) Sony's QRIO. (c) Honda's Asimo.

Figure 1.1: Three examples of modern humanoid robots.

for the special grappling hooks used in more conventional robot systems.

As a second reason, various studies indicate that people are more likely to accept robots in their environment if they appear natural, *i.e.* human-like. See for example the interesting field experiment by Kanda et al. (2004) on introducing robotic playmates and tutors for children, or the research by Breazeal (2003) on developing interactive robots with facial expressions.

Thirdly, from an engineering point of view, humanoid robots can be designed not just for the purpose of having human-like features, but because human shapes have evolved to be the optimal solution for certain problems. Moving around in a rocky outdoor environment, for example, is much easier for humans than for cars, because humans have very articulate legs that can cross high obstacles, whereas the wheels on cars can not. Instead of trying to cook up solutions from scratch and reinvent the wheel, it is much easier to take a design that has been under development for millions of years.

Humanoid robots themselves can be useful research objects, but also the many parts and spin-off results can be fruitful in various areas. Taking this literally, robotic arms and legs with dimensions and properties similar to humans may be used, for example, as artificial arms and legs for disabled people (Harwin et al. 1995), or as an exoskeleton to increase human capabilities towards the super-human (Pratt et al. 2004). But more indirectly, the process of building robotic arms and legs can help in understanding how human arms and legs work. This knowledge can then help in the design of other robotic systems that assist human arms and legs, for example the robotic rehabilitation aid for people recovering from a stroke (Ekkelenkamp et al. 2005).

1.1.2 Research on walking robots

One of the most interesting aspects of a humanoid robot is its locomotion technique: walking. Contrary to most, if not all, traditional mobile robots, which are based on rolling wheels, a humanoid robot has legs. Although wheeled locomotion is easier to control and highly efficient for moving on hard, flat terrain, legged locomotion can be useful in rocky or soft terrain, and it can be efficient as well.

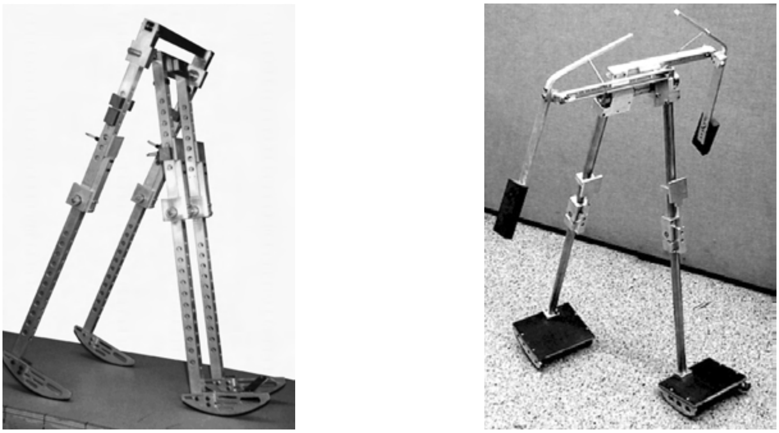
Walking can be defined as a locomotion gait of a biped, in which the feet are lifted alternately, while at least one foot is on the ground at all times. If both feet (temporarily) leave the ground, the gait is called running. From a mechanical point of view, walking is the process in which periodic internal shape changes of the mechanical structure, combined with reaction forces from the ground, result in an overall displacement.

The main way in which walking locomotion is implemented in robots is based on so-called *static walking*. From the definition of walking, we see that at least one foot is always on the ground. If we now construct a fully actuated robot, and ensure (by means of active control) that the center of mass is always located above the foot area, then if the robot is moving slowly enough (hence the name static), it is always stable. If we then command the joints to move periodically such that the rear foot is lifted, moved forward, and put down, then we obtain a stable walking motion: the robot walks without falling over. The control problem of walking is thus reduced to traditional joint tracking control of a rigid mechanism, and all standard control techniques from this field can be applied.

Starting from the basic idea of static walking, we can tune the joint trajectories such that a human-like motion appears. In addition, the static analysis of the center of mass can be extended to an analysis of the Zero Moment Point (ZMP), in which the center of pressure (resulting from both gravitational and inertial effects) is required to remain strictly inside the foot area. A more detailed description of ZMP-analysis can be found in Vukobratović (2004).

Many walking robots, such as the humanoids of Figure 1.1, are based on static walking, and these robots are generally highly versatile: they can walk up stairs, walk at various speeds, and use sensory information to plan the next steps. On top of that, the motion generally looks quite human-like. The one big remaining problem, however, is energy consumption. Even with huge battery packs, these robots can only operate for maybe an hour.

One of the reasons for this large energy consumption is that, although the walking motion of the robot may *look* natural and human-like, it may not *be* natural at all for the robot to walk in this way. Humans walk with a certain leg and joint motion because it is efficient for their specific mechanical structure to do so. This does not at all mean that the same leg and joint motion is also efficient for a robot with a different mechanical structure. Hence, simply using human joint trajectories as reference trajectories for walking robots may result in motion that



(a) Close copy of McGeer's passive dynamic walker, by Garcia et al. (1998).

(b) Three-dimensional passive dynamic walker, by Collins et al. (2001).

Figure 1.2: Two examples of unactuated passive dynamic walking robots.

looks good, but there is no reason to assume that it is efficient as well.

Fortunately, energy-efficient trajectories for walking robots do exist. McGeer (1989, 1990*a*) showed in his remarkable work on passive walking robots, that certain mechanical systems indeed have a natural tendency to walk. More precisely, these mechanisms, such as the ones in Figure 1.2, can walk down a shallow hill without actuation, only powered by a little bit of gravity. The center of mass of these robots does not always remain above the stance foot, and hence they are not statically stable. Nevertheless, it has been shown, both in theory and in practice, that the walking cycles of these robots are dynamically stable, hence the name *dynamic walking*.

Inspired by passive dynamic walkers, researchers have looked at various ways of adapting and augmenting the passive dynamic motions in order to obtain more robustness, walk on different slopes (including a level floor), and to attain other design goals. Many of these strategies are biologically inspired. For example, van der Linde (2000) used McKibben muscles in the design of a stepping robot, and Wisse & van Frankenhuyzen (2003) extended McGeer's original passive dynamic walking with McKibben muscles to obtain stable, robust, level-floor walking. Alexander (1990) showed how locomotion can be made more efficient by using springs at locations where animals have (elastic) tendons. Kuo (1999) gave several examples to control the unstable lateral motion in three-dimensional walking, one of which being the use of a trunk (upper body). Finally, Pratt & Pratt (1999) described how the use of compliant ankles and kneecaps can help to obtain better controllable and more efficient walking.

Others have focussed more on the energy conserving properties of passive walkers, and designed their controllers to adjust the energy balance. Spong & Bullo (2005) describe a control law that effectively rotates the apparent gravitational field, thus making the controlled robot move with the same gait on different slopes. Asano et al. (2004) define various control laws that control the energy level of a walking robot. Yamakita et al. (2000, 2001) apply the method of Passive Velocity Field Control (Li & Horowitz 1995) to the control of walking robots.

Unfortunately, no matter what control technique is used, the behavior of (almost) passive walkers remains difficult to understand, due to the highly nonlinear, coupled, and generally unstable dynamics, together with the hybrid aspects of switching between left and right foot, and between contact and no-contact. Controller design of almost-passive dynamic walkers has therefore been based mainly on engineering intuition, biological inspiration, and physical tinkering with experimental robots. Such methods have always been the basis of new design, and this author believes that they will always remain valuable in nonlinear control, mainly because the class of general nonlinear systems is so large and diverse: the optimal controller cannot just be computed from scratch. However, once a certain direction or type of solution has been determined by intuition or biological analogy, systematic and automated methods are valuable methods for specific quantitative design and fine-tuning. For example, although Alexander (1990) found useful locations for springs in walking robots based on observation of biological systems, the optimal stiffness value of such a spring cannot be easily deduced from biology, whereas it *can* be determined by a systematic optimization procedure, as shown in Section 5.1 of this thesis.

1.2 Port-Hamiltonian modeling and control

1.2.1 Port-Hamiltonian modeling

Engineering systems are becoming increasingly complex. Although this may be due partially to the yearning of researchers and engineers for new challenges and more difficult problems (Tanie 2005), the main reason is that society demands engineering systems to solve more complex problems. The complexity is caused by many aspects, for example the demand for better performance, smaller scale, faster response, lower cost, or lower weight. Such demands generally require the use of physical components that exhibit more complex behavior. For example, when a lighter metal bar is used to achieve less weight and higher speed, it can exhibit flexibility and oscillatory behavior that is not present with a heavier bar at a lower speed. These components can then result in nonlinear behavior and stronger dynamic coupling between the different parts of the system. If the higher-order dynamics occur on a much smaller time-scale than the leading-order dynamics, it may even lead to an apparent hybrid, or switching, behavior.

As the complexity of the problem (and its solution) increases, controlling the system and improving its behavior become more difficult. A detailed understanding of the behavior of the system is then required to be able to solve the problem. A common and sensible approach in engineering research is to 'divide and conquer', *i.e.* to think of the system as an interconnection of simpler subsystems. The interconnection is required since the subsystems are generally coupled in some way, but hopefully some of the coupling is weak or simple enough to allow the subsystems to be studied separately.

The process of deciding how to structure a system as an interconnection of basic elements is called modeling. As argued, for example, by Breedveld (2004), this decision process is highly dependent on the goal of the analysis, *i.e.* whether a basic qualitative understanding of the workings of the system is required, or a more detailed description that quantitatively predicts its behavior. It also depends on the operating conditions that are considered, and hence the decisions and the resulting model are only valid under those conditions; extrapolation to different conditions can easily lead to incorrect predictions.

Example 1.1 (bungee jump). Figure 1.3 shows an example of two models for a perfectly vertical bungee jump, in which a 91 kg person is attached to a 9 kg rope with measured stiffness $k = 30$ N/m and damping $d = 1$ Ns/m. In the first model (middle figure) the choice is made to concentrate the mass of the person in a point at the end of the rope and assume the rope to be a perfect linear spring plus damper, thus neglecting its mass. In the second, more complex, model (right figure), the rope is modeled as a more distributed interconnection of n (we choose $n = 10$) small masses, dampers, and springs, with still the person as a point mass at the end.

Simulation (bottom figure) shows that the behavior of the two models is more or less the same: when started from a height of 60 m with gravity pulling down, the vertical position of the person is a damped oscillation. For the purpose of getting a rough idea of how a bungee jump works and what velocities and accelerations are experienced by the jumper, the results of the two models are close enough, and hence the simplest model can be chosen. On the other hand, when the goal is to determine whether it is safe to jump from this height with this rope, the two models are clearly not equal: the lumped model shows a (barely) safe oscillation, while the distributed model shows the jumper crashing into the ground. The difference between these models is significant, and hence the simplest model does not suffice for the modeling goal. △

The result of the modeling process is an abstract model. In order to perform computer simulations of this model, it needs to be described as a set of equations, expressing the relations between variables, and, for dynamical systems, their time derivatives. Once the model has been determined, the relations between the variables are fixed, and all sets of equations that describe the model are essentially

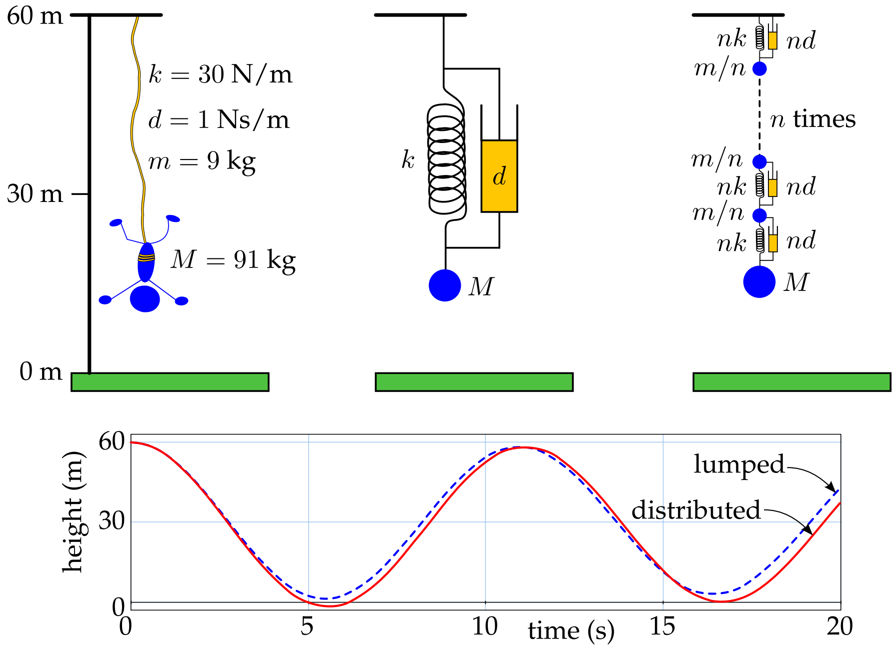


Figure 1.3: Two models of a vertical bungee jump (left figure): the middle figure shows a model where the mass, stiffness, and damping are assumed to be lumped, whereas the right figure shows a model with mass, stiffness, and damping distributed in n parts. Simulations of the two models for $n = 10$ (bottom figure) show that they behave similarly, but that the details are different.

equivalent. However, since the dynamic equations of complex systems are generally too complex to work with for analysis purposes, it is beneficial to structure them in a certain way, or even better, to represent them graphically. The models of Figure 1.3 already show examples of such a graphical representation. They are depicted as interconnections of standardized mechanical elements (mass, spring, damper), and these figures can be translated directly into equations.

Depicting nonlinear and multi-dimensional elements as standardized icons is less easy, however, and instead, several frameworks have been developed that allow such more general elements. The most well-known framework is that of block diagrams, which is based on causal information flow between subsystems. General systems are described as a set of subsystems (up to the desired level of detail) that are interconnected by signal lines. Each line is connected between the output of one subsystem and the input of another, and the subsystems themselves are described by functions that compute the output variables from the input variables.

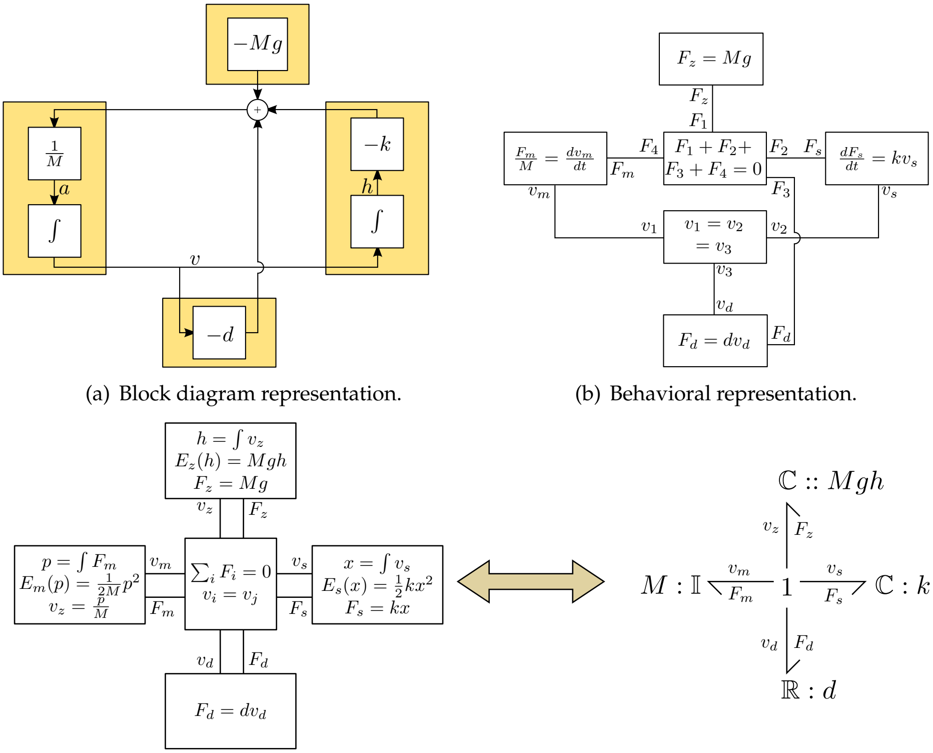
Block diagrams are very general and can describe all kinds of systems, but they are limited by causality: once a causal direction has been chosen, signals can only travel in one direction and not in the opposite direction. This gives a problem for re-usability, since the same element may allow several equivalent descriptions. For example, the damper of Figure 1.3 can be described equivalently as $F = dv$ and $v = F/d$ (with F the force and v the velocity). As a block diagram, only one of these formulations can be used, depending on what is the input signal and what is the output signal.

As an alternative to the block diagram framework, Willems (1991) introduced the behavioral framework (see Polderman & Willems (1998) for a complete introduction). In this framework, the signals between subsystems are not fixed to be uni-directional, but only their relation is fixed, *i.e.* that they are equal. Also the subsystems themselves are not described as functional assignments from input to output, but only as relations between the connected variables. This solves the problem of requiring multiple formulations of the same equation. Of course, for computer simulation, the equations need to be written as sequential assignments, but this can be done automatically by software tools, once the complete model has been described.

Both the block-diagram and the behavioral framework can be used to describe pretty much any type of system that exchanges any type of signal. This thesis, however, focusses mainly on physical (sub)systems that exchange energy. For example, the models of walking robots developed in Chapter 4 are described as an interconnection of two subsystems: one describing the kinetic and potential energy of the mechanical structure, and one describing the contact forces of the robot with the ground. These two subsystems interact by exchanging energy, *i.e.* some of the kinetic energy of the mechanism is transferred to the ground contact and then stored as deformation energy or dissipated through friction.

The framework of port-Hamiltonian systems is tailored to represent physical systems as an interconnection of energy-exchanging subsystems. External variables to a port-Hamiltonian subsystem only appear in pairs of collocated power-variables, *i.e.* variables that, when multiplied, give the physical power associated with these variables. Such an input structure is called a power-port. For example, in mechanical systems, the two power-variables could be the torque on a joint, and the collocated angular velocity of that joint. Physics demands that the computational directions of these variables are opposing, but which variable travels in which direction is not determined a priori. More detailed information on port-Hamiltonian systems can be found in Appendix B.1.

Example 1.2 (bungee jump model representations). Before reflecting more on the aspects of the different frameworks, let us first discuss an example of how the various model representations look for the bungee jump system of Example 1.1. Figure 1.4 shows the block diagram, behavioral, and port-Hamiltonian representations of the fully lumped parameter model of the bungee jump. An



(a) Block diagram representation.

(b) Behavioral representation.

(c) Port-Hamiltonian representation, together with the equivalent bond graph.

Figure 1.4: Three different model representations of the lumped bungee jump model of Figure 1.3: block diagram, behavioral, and port-Hamiltonian.

explanation of the symbols in the bond graph (bottom right figure) can be found in Appendix B.2.

All representations express the same model, so when simulating the three representations in a suitable simulation package, the results are exactly equal. However, for interpretation and analysis of the model structure by a designer or researcher, the models are clearly different. The block diagram (Figure 1.4(a), shown here in a non-standard way to allow better comparison with the other representations) is probably the most efficient implementation in terms of fewest variables and simplest blocks. However, the different parts of the model (spring, damper, mass, gravity, all marked by a dark box) have been tailor made and connected specifically for this model. For other models, the equations inside the parts may need to be rewritten with different causality or a change of sign.

The behavioral framework in Figure 1.4(b), on the other hand, is clearly built from separate components with separate variables (*e.g.* the force and velocity variables F_d and v_d of the damper). These components can be interconnected in different ways to other elements to model different physical interconnections (*e.g.* series or parallel connection for the damper), without the need to rewrite the equations inside the components themselves. The equality signs in the component expressions should be taken as real mathematical equality relations, not as assignment operators as used in programming languages.

Finally, the port-Hamiltonian representation of Figure 1.4(c) is similar to the behavioral representation, but the equations and interconnections are structured to represent power and energy flow. Each element is connected by two variables, which, when multiplied, give the power flow and hence form a power port. Furthermore, some elements have an internal energy E that is a function of a state variable, and it can be seen that the rate of change of internal energy for these elements precisely equals the power flow associated with the power port. $\triangle$

From the example, we can see that the different modeling frameworks each have their own positive and negative aspects. For a software engineer who needs to implement the model in a sequential programming language, probably the block diagram representation is most valuable, as it directly shows how the value of one variable can be computed from the values of the other variables. For a researcher trying to understand the physics of the system, however, the port-Hamiltonian implementation may provide most insight, as its structure directly shows energy flows inside the system.

1.2.2 Port-Hamiltonian control

A control system can often be described as an addition to a physical system, usually electronic, that measures certain signals (outputs of the physical system), processes them in some way, and then produces the appropriate control signals

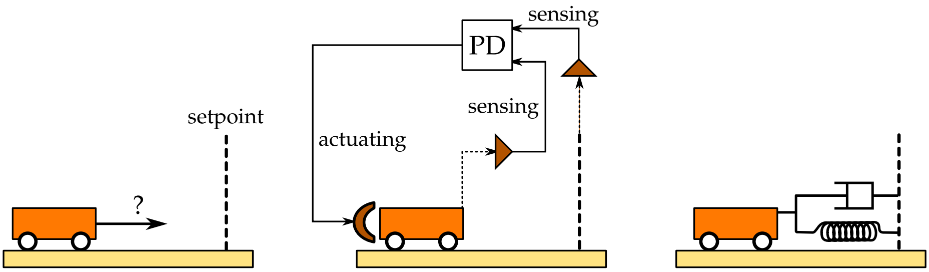


Figure 1.5: Stabilization of a cart at a certain setpoint (left): solution as a standard PD controller (middle), and its interpretation as a spring-damper (right).

(inputs of the physical system). Such a description is highly suitable for representation by a block diagram, as it clearly shows the signal flows.

When looking at a port-Hamiltonian representation, on the other hand, such an implementation is not very immediate: signals can only be accessed in pairs (one in, one out) of collocated power variables. To accommodate for this deficiency, port-Hamiltonian systems are usually augmented with elements from block diagrams, for example, to measure internal variables (such as state variables in the energy functions) and actuate other variables. As long as these additions only change the energy in the elements through the power ports (and not through the signal channels), they do not destroy the power-flow interpretation of the representation and actually provides a nice differentiation between information flow and energy flow.

Suppose, however, that we do *not* want to use these additional block diagram elements, but only act through the power ports. This suggests that the controller itself should be represented as a physical system, with associated power ports, internal energy, and dissipation. Such an interpretation was used, for example, by Takegaki & Arimoto (1981), who proposed a controller for mechanical systems that basically mimics a potential field. Hogan (1985) generalized these ideas to describe controllers as impedances, and even postulates that *any* controller will make a controlled system indistinguishable from a physical system (the postulate of physical equivalence). Stramigioli (2001), in turn, generalized Hogan's ideas to general spatial robotic systems.

Although these approaches are partially again just a matter of representation (any port-Hamiltonian system can be rewritten as a block diagram), thinking of control in terms of interacting physical systems has several advantages. First, physics may suggest a solution for control. For example, when a cart needs to be stabilized at a certain position (Figure 1.5), a physical solution would be to place a spring and damper between the cart and the desired position. This is fully equivalent to PD control, but provides a physical interpretation and motivation to use such a controller. Secondly, a port-interconnection of physical systems is again a physical system, and hence it inherits properties of physical systems, such as passivity (for lower-bounded energy functions). These properties can directly prove, for example, stability and boundedness of state trajectories and control inputs. Thirdly, the resulting controllers not only behave like physical systems, they may also be partially implemented as real physical systems. If this is possible, it can save considerably in efficiency, as well as sensing and computing requirements.

1.2.3 The European project GeoPlex

So far, we have mainly focussed on controlled mechanical systems, but the concept of energy is present in all physical domains. This raises the question of whether port-Hamiltonian model representations can be useful across the boundaries of different physical domains. Is it possible to formulate existing results and concepts in these domains in port-Hamiltonian terms? And if so, does this add anything to the understanding of the problem, by providing a different, fresh look? And suppose problems in different domains can be formulated in the same port-Hamiltonian framework, can specific solutions for problems in one domain be transformed to solutions for similar problems in a different domain?

Questions like these form the research basis of the European sponsored project called *Geometric Network Modeling and Control of Complex Physical Systems*, GeoPlex for short. The consortium for this project consists of researchers from different European universities working on problems in different physical domains, ranging from robotics through electrical power systems to chemical engineering. In addition, the software company Control Lab Products takes part in the consortium. The goals of GeoPlex are to address the aforementioned issues and questions about port-Hamiltonian systems, and, as the name of the project suggests, to attempt to use the port-Hamiltonian framework to model, analyze, and control complex physical systems, *i.e.* systems for which direct equation analysis is not feasible. Finally, the results are to be implemented in useable software tools. More information about the project is available on the website http://www.geoplex.cc.

1.3 Goals of this thesis

The research described in this thesis has been conducted as part of the GeoPlex project. As such, the goal is to attempt to apply the port-Hamiltonian framework to a specific complex physical system, in this case to walking robots. As argued in Section 1.1, walking robots, and especially passively walking robots, are complex systems to analyze, understand, and control efficiently. Especially two-legged walking robots are complex, since they often do not rely on static stability, contrary to robots with four or more legs.

The specific goals of this thesis are as follows.

- To develop a systematic way to construct port-Hamiltonian models for general rigid walking robots, *i.e.* mechanisms that can be approximated accurately as rigid bodies interconnected by ideal kinematic constraints, and that can interact with another mechanism through general contact forces and torques in three dimensions;

- To use this systematic method to construct port-Hamiltonian models of several walking robots, and analyze these models to obtain a better understanding of how and why these robots walk;

- To use the knowledge obtained from the model analysis to design energy-efficient controllers for walking robots.

Since this research is part of the GeoPlex project, we use the common inter-domain language of port-Hamiltonian systems to describe the models and main results. However, we do not stick to this framework dogmatically, and choose more appropriate representations when they are clearly better suited for the specific situation. The idea of thinking in terms of power-ports and interconnection by energy flows, though, is used throughout this thesis.

1.4 Thesis Outline

To describe the development of a systematic modeling technique for walking robots, and to apply these models for analysis and control, this thesis is structured as follows.

Chapters 2 and 3 describe a systematic way to model general rigid mechanical structures, possibly in contact with other mechanical structures. Chapter 2 generalizes earlier work on models of rigid mechanisms to a more general class of systems, including joints with configuration space unequal to $\mathbb{R}^n$ and nonholonomic joints. Chapter 3 describes a general way to model the contact kinematics and dynamics between rigid mechanisms, both for compliant and rigid contact.

Chapter 4 then applies these techniques to obtain port-Hamiltonian models of three examples of walking robots: a simple two-dimensional compass-gait robot, a planar experimental walking robot with knees, and a three-dimensional walking robot. Models at different levels of detail are derived, suitable for either simulation, analysis, or control. The models of these robots are analyzed in a systematic way to draw conclusions about the influence of their mechanical structure on their ability to walk, and efficient (or even passive) walking trajectories for these robots are derived.

Chapter 5 presents several techniques, based on the port-Hamiltonian models, to develop controlled energy-efficient walking. It discusses both control by passive adjustments of the structure, and control by active steering to increase robustness. It also shows preliminary results for the control of a planar experimental walker, and the use of foot-placement to control a three-dimensional walker.

▼ Several results in this thesis can be interpreted in terms of differential geometry or other more advanced mathematics. In order not to scare readers with limited knowledge of these topics, such interpretations have been carefully delimited by the triangle marks shown next to this paragraph. These paragraphs are not needed to understand the rest of the text, and can be safely skipped. ▲

Chapter 2

Modeling of Rigid Mechanisms

The goal of this chapter and the next is to obtain building blocks that can be used to construct mathematical models of the mechanical behavior of general walking robots. This chapter contains the necessary ingredients for the description of mechanisms comprised of a finite number of rigid links interconnected by ideal joints, possibly moving freely in space. Chapter 3 discusses models for collisions and contact of rigid mechanisms with their environment.

A rigid link (or rigid body) is defined as a finite volume of point masses, all of which have fixed relative distances. The most common example is a piece of solid material (*e.g.* aluminum) in which the point masses are the atoms of the material with constrained relative distances due to the structure of the solid.

An ideal joint is defined as a constraint between two rigid links that allows only certain relative velocities and prevents others, independently of the forces and torques applied to the links. An example is an (ideal) door hinge, which constrains the velocity of the door to be a rotation around the vertical hinge axis, relative to the building, even though gravitational forces may try to translate it vertically.

These definitions show that rigid links and ideal joints are idealizations of practical objects. Real links always have a certain stiffness, *i.e.* with a large enough force or a large enough operating frequency they will bend considerably or even break. Similarly, when real joints are subjected to large enough forces in the constrained directions, they will break. For the purpose of practical walking robots, however, the assumption of rigidity of links and idealness of joints is reasonable: robot links and joints are designed to be stiff (rigid) and strong (ideal) to increase their lifetime, and also to simplify modeling and control of these robots, which is much harder if the links are flexible. The frequencies of the motion and magnitudes of the forces are low enough to allow this design. However, as in all modeling tasks, models should be tested against the real practical realizations to check if the modeling assumptions are justified.

In this chapter, we develop a dynamic model of a general rigid mechanism in several steps. We first look at the description and analysis of the *kinematics* of rigid mechanisms (Sections 2.1 and 2.2), then at the description of the *dynamics* of the mechanisms in the framework of port-Hamiltonian systems (Section 2.3), and finally at methods to deal with kinematic loops and nonholonomic constraints (Section 2.4).

2.1 Kinematics of rigid bodies

The objective of a kinematic study is to obtain insight in the possible configurations and velocities that a mechanical system can have, without looking at the physical causes for these configurations and velocities. In other words, we study and try to describe the possible motions of a mechanical system without looking at the forces acting on it.

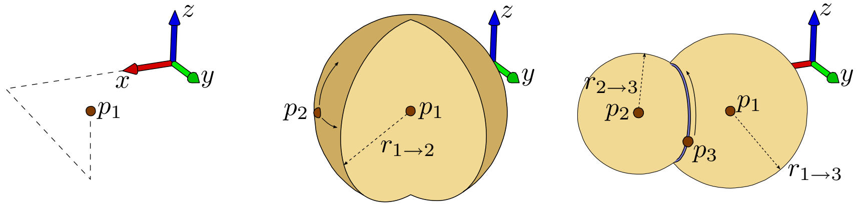
2.1.1 Configuration of a rigid body

As a first step, we consider a single rigid body, freely moving in space. The configuration of this rigid body could be described by the positions of all the point masses that are part of the body. If, for example, we consider a one kilogram slab of aluminum, we could describe its configuration (expressed in some coordinate frame) by the positions of all the atoms; this would amount to roughly 10^{26} variables.

Fortunately, the assumption of rigidity can reduce the dimension of the state space enormously. The point masses are constrained to have certain fixed relative distances, and so the positions of three fixed non-collinear points of the body are enough to determine the positions of all the other points. To be more precise, we prove the following lemma, using the definitions of the various spaces from Appendix A.3.

Lemma 2.1. *The space of all possible configurations of a rigid body in three-dimensional space, relative to some reference frame, is the six-dimensional space* $SE(3)$*, which is topologically equivalent to the set* $\mathbb{R}^1 \times \mathbb{R}^1 \times \mathbb{R}^1 \times \mathbb{S}^2 \times \mathbb{S}^1$.

Proof. We prove the lemma geometrically by considering Figure 2.1. Suppose we have a rigid object with three non-collinear reference points on it. How many degrees of freedom do we have to place this object in a three-dimensional world? Starting in Figure 2.1a, we see that the first reference point (labeled p_1) can be positioned freely in space, *i.e.* there are three degrees of freedom for this point. Then, following Figure 2.1b, there are only two degrees of freedom in positioning the second reference point p_2, since the rigidity constraint forces p_2 to be somewhere on a sphere with center p_1 and radius $r_{1\to2}$ ($r_{a\to b}$ denotes the distance between p_a



(a) Three degrees of freedom for the first point.

(b) Two degrees of freedom for the second point.

(c) One degree of freedom for the third point.

Figure 2.1: Three non-collinear points on a rigid body determine its configuration.

and p_b). Finally, Figure 2.1c shows that the last reference point p_3 must be both on a sphere centered around p_1 and on a sphere centered around p_2, *i.e.* , on the intersection of the two spheres: a circle. The position on this circle has one degree of freedom, making a total of six degrees of freedom. With all three reference points positioned, the object can be constructed using the other rigidity constraints.

The resulting space is thus indeed the six-dimensional space $\mathbb{R}^1 \times \mathbb{R}^1 \times \mathbb{R}^1 \times \mathbb{S}^2 \times \mathbb{S}^1$: an element of $\mathbb{R}^1 \times \mathbb{R}^1 \times \mathbb{R}^1$ to fix the position of p_1, subsequently an element of $\mathbb{S}^2$ to fix the position of p_2, and finally an element of $\mathbb{S}^1$ to fix the position of p_3 and hence the whole body. Grouping together the translation parts and the rotation parts, we obtain the equivalent group $T(3) \times SO(3)$. $T(3)$ is the group of translations in three dimensions, and $SO(3)$ is the special orthogonal group in three dimensions, *i.e.* the group of 3D rotations. The space $\mathbb{S}^2 \times \mathbb{S}^1$ is called the Poincare sphere and is a representation of $SO(3)$, as shown in Marsden & Ratiu (1999).

Combining the two resulting groups $T(3)$ and $SO(3)$, we obtain the group known as $SE(3)$, the special Euclidean group in three dimensions, which hence is the space of all possible 3D configurations of a rigid body, relative to a certain frame. □

In numerical computations and simulations, instead of describing the relative configuration of a rigid body as an abstract element of $SE(3)$, we would like to use coordinates, *i.e.* real numbers. Since $SE(3)$ is a six-dimensional space, we would like to use six real numbers to describe it, just like we would to describe the six-dimensional Euclidean space.

Unfortunately, $SE(3)$ is topologically different from the Euclidean space, and it is hence not possible to continuously and globally cover it using six coordinates. Instead, several representation methods exist to describe $SE(3)$ either only locally continuously using six numbers, or globally continuously using more than six numbers. Selig (2005) discusses representation theory in the context of robotics.

Examples of locally continuous descriptions are the often-used methods of Euler angles. In these methods, the rotation part is parameterized by three consecutive rotations about local axes (which axes depends on the precise variation of the method). The amounts of rotation provide three numbers (angles), which, together with three coordinates for translation, give a parameterization of $SE(3)$. For every element of $SE(3)$ there exist such coordinates, so the covering is global. However, whatever variation is chosen, there are always rotations that are described by only two coordinates with the third arbitrary (when using consecutive rotations around x, y, and z, for example, the coordinates $(a-\theta, \frac{\pi}{2}, \theta)$ describe the same rotation for all θ). The result of this is that, during a smooth, continuous change of rotation, the coordinates can have non-smooth discontinuities! Since we want to take time-derivatives of rotations, these artifacts are undesired.

An example of a globally continuous but redundant representation is by a vector of the form $q = \left[\cos(\frac{\theta}{2}), n_1 \sin(\frac{\theta}{2}), n_2 \sin(\frac{\theta}{2}), n_3 \sin(\frac{\theta}{2})\right]$ with $n_1^2+n_2^2+n_3^2 = 1$. The three numbers n_i define the unit axis of rotation in three-dimensional space, and the number θ the angle of rotation around this axis. The vector q is usually thought of as a unit quaternion (Selig 2005). Together with three numbers for translation, this quaternion can be used as a representation of $SE(3)$. However, the quaternion q is constrained to have unit norm, and there is a double covering of the space of rotations (a $360°$ rotation around a certain axis is the same as a $0°$ around that axis, but the quaternion representation for these two rotations is different).

In this thesis, we choose to use a representation method that uses so-called homogeneous matrices to describe the relative configuration (translation and rotation) of two coordinate frames in three-dimensional space. Homogeneous matrices have the advantage of being globally continuous and being easy to perform calculations with, *i.e.* using basic matrix multiplications. Homogeneous matrices are defined as follows.

Definition 2.2 (Homogeneous Matrices). *A homogeneous matrix H is a matrix of the form*

$$H := \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \left[\begin{array}{ccc:c} R_{xx} & R_{xy} & R_{xz} & p_x \\ R_{yx} & R_{yy} & R_{yz} & p_y \\ R_{zx} & R_{zy} & R_{zz} & p_z \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \in \mathbb{R}^{4\times 4} \tag{2.1}$$

with $R^{-1} = R^T \in \mathbb{R}^{3\times 3}$, $\det(R) = 1$, and $p \in \mathbb{R}^{3\times 1}$.

Using the same notation as Stramigioli (2001), we can use homogeneous matrices in the following way to describe the configuration (position and orientation) of a rigid body.

1. Choose a right-handed coordinate frame Ψ_i as a reference frame, in which the configuration of the rigid body will be expressed;

2. Rigidly attach another right-handed coordinate-frame Ψ_j to the body, where 'rigidly attaching' means that the coordinates of all points of the body, when expressed in Ψ_j, are constant in time, independently of the motion of the body;
3. Define p^i_j as the 3×1 column vector of coordinates of the origin of Ψ_j expressed in Ψ_i;
4. Define R^i_j as the 3×3 matrix with the columns equal to the coordinates of the three unit vectors along the frame axes of Ψ_j expressed in Ψ_i;
5. Construct the matrix H^i_j using (2.1) with $R = R^i_j$ and $p = p^i_j$.

We first prove that this definition of H^i_j gives a proper homogeneous matrix, *i.e.* that it satisfies the required properties on R. Let us denote by $\hat{a}$ the unit vector in the direction of coordinate axis a and compute

$$(R^i_j)^T R^i_j = \begin{bmatrix} \hat{x}^T \\ \hat{y}^T \\ \hat{z}^T \end{bmatrix} \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix} = \begin{bmatrix} \hat{x}^T\hat{x} & \hat{x}^T\hat{y} & \hat{x}^T\hat{z} \\ \hat{y}^T\hat{x} & \hat{y}^T\hat{y} & \hat{y}^T\hat{z} \\ \hat{z}^T\hat{x} & \hat{z}^T\hat{y} & \hat{z}^T\hat{z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

where the last step follows since the axes of a coordinate frame are orthogonal to each other and the vectors $\hat{x}$, $\hat{y}$, and $\hat{z}$ have unit length. This proves that $(R^i_j)^T = (R^i_j)^{-1}$. Now, to prove $\det(R^i_j) = 1$, we compute

$$\det(R^i_j) = \det(\begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}) = \langle \hat{x} \times \hat{y}, \hat{z} \rangle = 1 \tag{2.3}$$

where the last step follows since the vectors have unit length and since the frame is right-handed, hence the cross product of $\hat{x}$ and $\hat{y}$ by definition equals $\hat{z}$. So indeed, the matrix H^i_j defined in this way is a proper homogeneous matrix of the form (2.1).

However, although the matrix may be homogeneous, it remains to be shown how it describes the configuration of the rigid body, which was its purpose. To this end, we need the following definitions of the augmented coordinates of a (free) vector and of a point.

Definition 2.3 (Coordinates of a Vector). *The (augmented) coordinate vector V^i corresponding to a (free) vector v in space is the vector*

$$V^i := \begin{bmatrix} v^i \\ 0 \end{bmatrix} := \begin{bmatrix} x^i_v \\ y^i_v \\ z^i_v \\ 0 \end{bmatrix} \in \mathbb{R}^4 \tag{2.4}$$

with x^i_v, y^i_v, z^i_v the orthogonal projections of v on the respective axes of a right-handed coordinate frame Ψ_i, i.e. *the conventional coordinates v^i of the vector v.*

Definition 2.4 (Coordinates of a Point). *The (augmented) coordinate vector Q^i corresponding to a point q in space is the vector*

$$Q^i := \begin{bmatrix} q^i \\ 1 \end{bmatrix} := \begin{bmatrix} x^i_q \\ y^i_q \\ z^i_q \\ 1 \end{bmatrix} \in \mathbb{R}^4 \tag{2.5}$$

with x^i_q, y^i_q, z^i_q the coordinates of the vector from the origin of the right-handed frame Ψ_i to the point q.

The coordinates of a single point q, expressed in different coordinate frames Ψ_i and Ψ_j, are related by the homogeneous matrix H^i_j as constructed before, namely in the following way.

$$Q^i = H^i_j Q^j \tag{2.6}$$

This can be proved as follows: for a point q with coordinates Q^j when expressed in Ψ_j, its coordinate vector q^i is the vector from the origin of Ψ_i to the origin of Ψ_j plus the vector from the origin of Ψ_j to the point q expressed in Ψ_i, or mathematically

$$\begin{aligned} q^i = p^i_j &+ \left(\hat{x}_i \hat{x}_i^T \hat{x}_j + \hat{y}_i \hat{y}_i^T \hat{x}_j + \hat{z}_i \hat{z}_i^T \hat{x}_j\right) q^j_x + \left(\hat{x}_i \hat{x}_i^T \hat{y}_j + \hat{y}_i \hat{y}_i^T \hat{y}_j + \hat{z}_i \hat{z}_i^T \hat{y}_j\right) q^j_y \\ &+ \left(\hat{x}_i \hat{x}_i^T \hat{z}_j + \hat{y}_i \hat{y}_i^T \hat{z}_j + \hat{z}_i \hat{z}_i^T \hat{z}_j\right) q^j_z = p^i_j + R^i_j q^j \end{aligned} \tag{2.7}$$

which can be written conveniently as the matrix multiplication (2.6).

When the coordinates Q^i of a point q are expressed in a frame Ψ_i that is rigidly attached to the body to which also q is rigidly attached, these coordinates are often called *body-coordinates*. Similarly, when the coordinates Q^0 are expressed in a frame Ψ_0 that is considered a reference frame, the coordinates are called *world-coordinates*.

The property of defining a coordinate transformation by a simple matrix multiplication can be used for various purposes. Knowing the relative homogeneous matrix between two coordinate frames means that quantities computed numerically in one of these frames can be transformed to the other frame by a matrix multiplication. Furthermore, when rigid bodies are moving around over time, we can describe this motion by a time-varying homogeneous matrix, and use it to describe the time-evolution of the position of any point attached to the body by a multiplication of the body-coordinates of that point. These properties show that indeed, the homogeneous matrix, constructed as before, describes the configuration (position and orientation) of a rigid body.

Homogeneous matrices have another nice property, which follows from the interpretation as a coordinate transformation. Since coordinate transformations

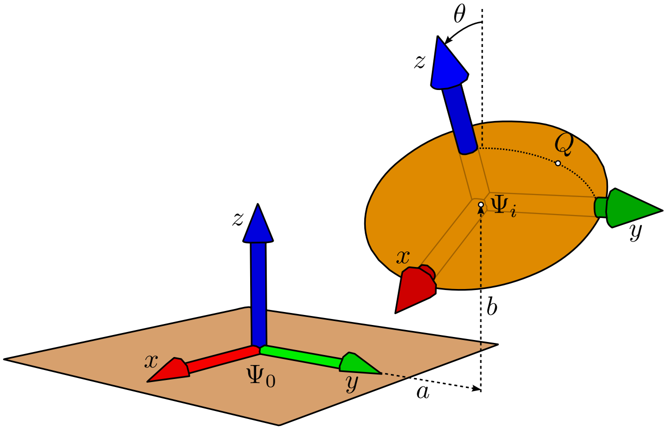


Figure 2.2: A body with frame Ψ_i translated and rotated relative to a frame Ψ_0.

must be invertible, we have

$$H_i^j = \left(H_j^i\right)^{-1} = \begin{bmatrix} (R_j^i)^{-1} & -(R_j^i)^{-1} p_j^i \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} (R_j^i)^T & -(R_j^i)^T p_j^i \\ 0 & 1 \end{bmatrix} \tag{2.8}$$

which defines the matrix H_i^j as a simple combination of the elements of H_j^i.

Remark. The representation using homogeneous matrices is highly redundant; the matrix contains sixteen numbers, whereas $SE(3)$ is only six-dimensional. Definition (2.1) constrains the structure of a homogeneous matrix: the bottom row must be equal to $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$, and R must be an orthogonal matrix with determinant 1. When homogeneous matrices are used in theoretical developments and proofs (as they are in this thesis), these constraints are automatically satisfied by the operations allowed on them. However, when they are used in numerical computations and simulations, in which roundoff errors and other approximations occur, it should be checked whether the matrices remain homogeneous and whether a numerical correction is necessary. △

Example 2.5. As an example of the described approach, consider Figure 2.2, which shows a rigid ellipsoid with coordinate frame Ψ_i and a plane with frame Ψ_0 (the symbol Ψ_0 is usually taken as the reference frame). To describe the relative configuration of the ellipsoid with respect to the plane, we can compute the matrix H_i^0 as

$$H_i^0 = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & a \\ -\sin(\theta) & 0 & \cos(\theta) & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

Suppose we are interested in the point Q on the ellipsoid which has body-fixed coordinates, say, $x^i = 0, y^i = 2, z^i = 1$ (these are independent of the configuration of the ellipsoid). Then to find the coordinates of the point Q relative to the plane, *i.e.* expressed in Ψ_0, we compute

$$Q^0 = H_i^0 Q^i = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & a \\ -\sin(\theta) & 0 & \cos(\theta) & b \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sin(\theta) \\ 2+a \\ \cos(\theta)+b \\ 1 \end{bmatrix} \tag{2.10}$$

△

2.1.2 Velocity of a rigid body

The next aspect in the kinematic analysis of a rigid body is the description of its possible velocities. As in the case of configuration, there is a mixture of linear velocities and angular velocities, and we seek to describe these two aspects as one mathematical object.

Since we want to use a time-varying homogeneous matrix $H(t)$ to describe the position and orientation of a rigid body, it would be straightforward to use a matrix $\dot{H}(t) := \frac{d}{dt}H(t)$ to describe its velocity. However, at all times the matrix $H(t)$ is constrained to be a homogeneous matrix, and so also the degrees of freedom of $\dot{H}(t)$ are constrained, depending on the current value of $H(t)$.

In order to find a better representation of velocity, we look more closely at the constraints on $\dot{H}$. From (2.1), we have

$$\dot{H}(t) = \begin{bmatrix} \dot{R}(t) & \dot{p}(t) \\ 0 & 0 \end{bmatrix} \quad \text{with} \quad R(t)R^T(t) = R^T(t)R(t) = I \tag{2.11}$$

If we compute the time-derivative of the constraint on R, we obtain

$$0 = \dot{R}^T R + R^T \dot{R} = \left(R^T \dot{R}\right)^T + R^T \dot{R} \tag{2.12}$$

which shows that the constraints on $\dot{H}$ can be translated into the constraint that $R^T\dot{R}$ must be a skew-symmetric matrix (plus the constraint that the bottom row of $\dot{H}$ must be all zeros). This leads us to the notion of a twist and its use as a representation for the velocity of a rigid body.

First, we define the tilde operator acting on elements of $\mathbb{R}^3$ as the bijective linear mapping from $\mathbb{R}^3$ to the space of 3×3 skew-symmetric matrices such that

$$\tilde{w} := \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \quad \forall\, w = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} \in \mathbb{R}^3 \tag{2.13}$$

With this definition and following the notation of Stramigioli (2001), we can formulate the definition[1] of a twist.

Definition 2.6 (Twists). *Let $H_j^i(t)$ be a time-varying homogeneous matrix representing the relative configuration of a rigid body j relative to a body i. We define the twist $T_j^{k,i}$ (twist expressed in coordinate frame Ψ_k) of the rigid body as the vector*

$$T_j^{k,i}(t) := \begin{bmatrix} \omega_j^{k,i}(t) \\ v_j^{k,i}(t) \end{bmatrix} \in \mathbb{R}^6 \tag{2.14}$$

such that at all times t we have

$$\tilde{T}_j^{k,i} := \begin{bmatrix} \tilde{\omega}_j^{k,i} & v_j^{k,i} \\ 0 & 0 \end{bmatrix} = H_i^k \dot{H}_j^i H_k^j = \begin{bmatrix} R_i^k(\dot{R}_j^i R_i^j) R_k^i & R_i^k \dot{p}_j^i - R_i^k \dot{R}_j^i p_k^j \\ 0 & 0 \end{bmatrix} \tag{2.15}$$

where we omitted the explicit time-dependencies for clarity.

Since the tilde operator is a bijective mapping and since homogeneous matrices are invertible, the space of allowed matrices $\dot{H}_j^i$ is bijectively parameterized by the twists. The matrix $\dot{H}_j^i$ can be simply recovered from the twist as

$$\dot{H}_j^i = H_k^i \tilde{T}_j^{k,i} H_j^k \tag{2.16}$$

The problem of constraints on the allowed velocities is thus avoided if velocities are represented by twists instead of time-derivatives of a homogeneous matrices.

▼ As demonstrated in Appendix A.3, the space $SE(3)$ is a Lie group, and homogeneous matrices can be used as matrix representation of the elements of $SE(3)$. The lie algebra $se(3)$ of the group is defined as the tangent space to the identity of the group. From Definition 2.6, we see that the twist is just the element of the algebra $se(3)$ corresponding to the translation of the tangent element $\dot{H}$ at H to the identity. The right and left translation correspond to the choice $\Psi_k = \Psi_i$ and ▲ $\Psi_k = \Psi_j$, respectively.

Apart from being a mathematically attractive representation of a constrained higher-dimensional space, twists also have a clear geometric interpretation. To see this, let us return to the use of a matrix H_i^0 to compute the world coordinates of a point q attached to a rigid body. We can compute the (linear) velocity $\dot{Q}^0$ of the point q as the rigid body moves in space by using the twist:

$$\dot{Q}^0 = \frac{d}{dt}\left(H_i^0 Q^i\right) = \dot{H}_i^0 Q^i = \tilde{T}_i^{0,0} H_i^0 Q^i = \tilde{T}_i^{0,0} Q^0 = \begin{bmatrix} \tilde{\omega}_i^{0,0} q^0 + v_i^{0,0} \\ 0 \end{bmatrix} \tag{2.17}$$

[1] Sometimes, such as in Murray et al. (1994), twists are defined as $\begin{bmatrix} v^T & \omega^T \end{bmatrix}^T$, so with ω and v ordered differently. This notation is not arbitrary, but depends on choosing either ray coordinates or axis coordinates, as defined in screw theory, see *e.g.* Lipkin (1985).

where the second equality follows from $\dot{Q}^i = 0$, since q is rigidly attached to the body and hence its coordinates relative to Ψ_i are constant. The twist $T_i^{0,0}$ is often called the *twist in world-coordinates*, since it directly relates the positions Q^0 and velocities $\dot{Q}^0$ of points, *i.e.* when expressed in the world-fixed coordinate frame Ψ_0. Similarly, the twist $T_i^{i,0}$ is called the *twist in body-coordinates* as it directly relates positions and velocities of points expressed in the body-fixed frame Ψ_i.

The previous discussion shows how the twist can be interpreted as a combination of a linear velocity and an angular velocity: $\omega_i^{0,0}$ defines a vector of rotation through the origin of Ψ_0, and $v_i^{0,0}$ defines a vector of linear velocity, such that a point q on the rigid body has a velocity relative to Ψ_0 equal to $\dot{q}^0 = \omega_i^{0,0} \wedge q^0 + v_i^{0,0}$.

However, the choice of splitting between rotation and translation depends on the choice of the reference frame Ψ_0; if we move the reference frame to a different location, both the rotation and translation part may change, even though the velocity of the rigid body is still the same. An interpretation of a twist that is independent of the choice of reference frame is given by Charles' Theorem, which is based on the notion that every rigid motion can be decomposed uniquely in a rotation around an axis plus a translation along the same axis.

Theorem 2.7 (Charles' Theorem). *Every twist T can be written as the sum of a rotation around an axis and a translation along the same axis.*

$$T = \alpha \begin{bmatrix} \hat{\omega} \\ r \wedge \hat{\omega} \end{bmatrix} + \beta \begin{bmatrix} 0 \\ \hat{\omega} \end{bmatrix} \tag{2.18}$$

where $\hat{\omega}$ is the unit vector in the direction of the axis, r is any vector from the origin of the reference frame to that axis, α is the magnitude of the rotation, and β is the magnitude of the translation.

Proof. Consider a general twist T. We can distinguish three situations:

- $\omega = v = 0$: choose $\alpha = \beta = 0$ and $\hat{\omega}$ and r arbitrary ($\hat{\omega}$ and r have no physical meaning here, since there is no motion).
- $\omega = 0, v \neq 0$: choose $\alpha = 0$, $\beta = \|v\|$, $\hat{\omega} = \frac{v}{\|v\|}$ and r arbitrary (r has no physical meaning here, since a translational motion is fully defined by its direction $\hat{\omega}$ and magnitude β).
- $\omega \neq 0$: choose $\alpha = \|\omega\|$, $\hat{\omega} = \frac{\omega}{\|\omega\|}$, $r = \frac{1}{\alpha}\hat{\omega} \wedge (v - \hat{\omega}^T v \hat{\omega})$, and $\beta = \hat{\omega}^T v$.

With these choices of α, β, $\hat{\omega}$, and r, any twist can be expressed in the form of the theorem. □

With a twist expressed in the form of Theorem 2.7, we can make a coordinate-independent distinction between a purely translational motion ($\alpha = 0$), a rotational motion ($\beta = 0$), or a general so-called screw-motion ($\alpha \neq 0$, $\beta \neq 0$). The ratio of the numbers α and β is called the pitch.

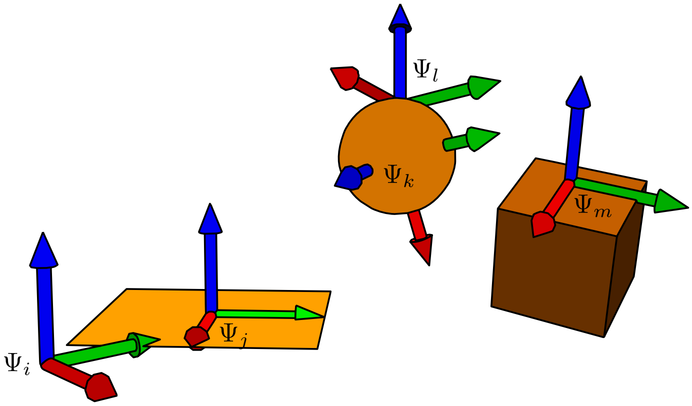


Figure 2.3: Three rigid bodies with several attached coordinate frames.

Finally, let us present some useful identities about twists and homogeneous matrices that are used later in this thesis.

Lemma 2.8. *Given the setup of Figure 2.3 with Ψ_i and Ψ_j rigidly attached to one rigid body (the plane), Ψ_k and Ψ_l rigidly attached to another rigid body (the sphere), and Ψ_m attached to the final rigid body (the cube). Then the following identities hold*

(a) $T_j^{i,i} = T_l^{k,k} = 0$

(b) $\tilde{T}_j^{m,k} = H_i^m \tilde{T}_j^{i,k} H_m^i$

(c) $T_j^{m,k} = \mathrm{Ad}_{H_i^m} T_j^{i,k}$ *with* $\mathrm{Ad}_H := \begin{bmatrix} R & 0 \\ \tilde{p}R & R \end{bmatrix}$ *for* $H = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$

(d) $T_k^{j,j} = -T_j^{j,k}$

(e) $T_m^{j,j} = T_k^{j,j} + T_m^{j,l}$

(f) $\frac{d}{dt} \mathrm{Ad}_{H_i^k} = \mathrm{Ad}_{H_i^k} \mathrm{ad}_{T_i^{i,k}}$ *with* $\mathrm{ad}_T := \begin{bmatrix} \tilde{\omega} & 0 \\ \tilde{v} & \tilde{\omega} \end{bmatrix}$ *for* $T = \begin{bmatrix} \omega \\ v \end{bmatrix}$

Proof. We prove each of the properties separately, using Definition 2.6 and the rigidity of the objects.

(a) The twist $T_j^{i,i}$ is defined in tilde form as $\tilde{T}_j^{i,i} = \dot{H}_j^i H_i^j$. Since the frames Ψ_i and Ψ_j are attached to the same body, we have $\dot{H}_j^i = 0$, and hence also $T_j^{i,i} = 0$. The same holds for $T_l^{k,k}$.

(b) We can directly write

$$\tilde{T}_j^{m,k} = H_k^m \dot{H}_j^k H_m^j = H_i^m H_k^i \dot{H}_j^k H_i^j H_m^i = H_i^m \tilde{T}_j^{i,k} H_m^i \tag{2.19}$$

which gives a coordinate transformation rule for twists in tilde form.

(c) Expanding the expression obtained in (b), we find

$$
\begin{aligned}
\tilde{T}_j^{m,k} &= H_i^m \tilde{T}_j^{i,k} H_m^i = \begin{bmatrix} R_i^m & p_i^m \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\omega}_j^{i,k} & v_j^{i,k} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R_m^i & -R_m^i p_i^m \\ 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} R_i^m \tilde{\omega}_j^{i,k} R_m^i & -R_i^m \tilde{\omega}_j^{i,k} R_m^i p_i^m + R_i^m v_j^{i,k} \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \widetilde{\left(R_i^m \omega_j^{i,k}\right)} & -\widetilde{\left(R_i^m \omega_j^{i,k}\right)} p_i^m + R_i^m v_j^{i,k} \\ 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} \widetilde{\left(R_i^m \omega_j^{i,k}\right)} & \tilde{p}_i^m R_i^m \omega_j^{i,k} + R_i^m v_j^{i,k} \\ 0 & 0 \end{bmatrix} \qquad (2.20)
\end{aligned}
$$

where the third line follows since for all $x \in \mathbb{R}^3$ we have

$$
\begin{aligned}
\widetilde{(R\omega)}x = (R\omega) \wedge x = RR^T\left((R\omega) \wedge x\right) &= R\left((R^T R\omega) \wedge (R^T x)\right) \\
= R\left(\omega \wedge (R^T x)\right) &= R\tilde{\omega} R^T x \qquad (2.21)
\end{aligned}
$$

and hence $\widetilde{(R\omega)} = R\tilde{\omega}R^T$. Comparing the expression for $\tilde{T}_j^{m,k}$ to the tilde representation of $\mathrm{Ad}_{H_i^m} T_j^{i,k}$, we see that they are equal.

(d) From the time-derivative of the identity $H_k^j H_j^k = I$, we immediately obtain

$$
0 = \dot{H}_k^j H_j^k + H_k^j \dot{H}_j^k = \tilde{T}_k^{j,j} + \tilde{T}_j^{j,k} \qquad (2.22)
$$

(e) Looking at the tilde-form of the equation, we see

$$
\begin{aligned}
\tilde{T}_m^{j,j} &= \dot{H}_m^{j,j} H_j^m = \frac{d}{dt}\left(H_k^j H_l^k H_m^l\right) H_j^m \\
&= \left(\dot{H}_k^j H_m^k + H_k^j \dot{H}_l^k H_m^l + H_l^j \dot{H}_m^l\right) H_j^m \\
&= \dot{H}_k^j H_j^k + H_k^j \dot{H}_l^k H_j^l + H_l^j \dot{H}_m^l H_l^m H_j^l \\
&= \tilde{T}_k^{j,j} + 0 + \tilde{T}_m^{j,l} \qquad (2.23)
\end{aligned}
$$

where the zero follows since Ψ_k and Ψ_l are attached to the same body.

(f) By definition of Ad_H, we have

$$
\frac{d}{dt}\mathrm{Ad}_{H_k^i} = \begin{bmatrix} \dot{R}_k^i & 0 \\ \dot{\tilde{p}}_i^k R_i^k + \tilde{p}_i^k \dot{R}_i^k & \dot{R}_i^k \end{bmatrix} = \begin{bmatrix} R_i^k \tilde{\omega}_i^{i,k} & 0 \\ R_i^k \tilde{v}_i^{i,k} + \tilde{p}_i^k R_i^k \tilde{\omega}_i^{i,k} & R_i^k \tilde{\omega}_i^{i,k} \end{bmatrix} \qquad (2.24)
$$

which can be written in the proposed form.

□

2.2 Kinematics of rigid mechanisms

We define a rigid mechanism as a (finite) number of rigid bodies, interconnected by ideal joints. Rigid bodies have been discussed in the previous section, and now we proceed to discuss the mathematical formulation of ideal joints using the same concepts of homogeneous matrices and twists. We then combine rigid bodies and ideal joints to obtain a kinematic description of general rigid mechanisms.

As described before, an ideal joint (sometimes also called a kinematic pair) is a purely kinematic relationship between the motion of two rigid bodies; some relative velocities are allowed and some are not, independently of the forces applied to the bodies. The following definition characterizes the class of joints considered in this thesis.

Definition 2.9 (Globally Parameterized Rigid Joint). *A globally parameterized rigid joint is a kinematic restriction of the allowed relative twist of two rigid bodies i and j to a linear subspace of dimension k, where the relative motion of the bodies is described by two sets of states, namely*

- *a matrix $Q \in \bar{\mathcal{Q}}$, parameterizing the relative configuration as $H^i_j = H^i_j(Q)$*
- *a vector $v \in \mathbb{R}^k$, parameterizing the relative twist as $T^{i,i}_j = X(Q)v$*

where $X(Q)$ depends smoothly on Q and $v = V_Q(\dot{Q})$ with V_Q invertible and linear in $\dot{Q}$. Furthermore, there exists a mapping $F_Q : \mathbb{R}^k \to \bar{\mathcal{Q}}$ satisfying

- *$F_Q(\phi)$ assigns local coordinates $\phi \in \mathbb{R}^k$ to a neighborhood of every $Q \in \bar{\mathcal{Q}}$.*
- *$F_Q(\phi)$ is twice continuously differentiable in ϕ*
- *$F_Q(0) = Q$*

The function $F_Q(\phi)$ defines a local coordinate patch with coordinates ϕ around every allowed configuration parameterized by Q. If the configurations around Q are described by coordinates ϕ, the velocity v can also be expressed as

$$v = V_Q(\dot{Q}) = V_{F_Q(\phi)}\left(\frac{\partial F_Q(\phi)}{\partial \phi}\dot{\phi}\right) \tag{2.25}$$

i.e. in terms of the time derivatives of the local coordinates. This property is used in the derivation of the dynamic equations in Theorem 2.17.

Note that the linearity of the subspace implies that no end-stops or speed-bounds are considered. Furthermore, since the dimension of the vector ϕ is equal to the dimension of v, we consider only joints for which the space of allowed relative configurations (described by ϕ) has the same dimension as the space of instantaneously allowed velocities (described by v). This type of joint is called a *holonomic* joint. Section 2.4 discusses an extension of the results to nonholonomic joints, *i.e.* joints for which the space of allowed instantaneous velocities has a smaller dimension than the space of allowed configurations.

The technical conditions on the mapping F are trivially satisfied if Q can be chosen as a k-vector (in which case we can take $F_Q(\phi) = Q + \phi$). Thus, the often encountered rotational and prismatic joints are part of the class of globally parameterized rigid joints, as is shown in Example 2.10 below.

▼ Another case for which the conditions on local coordinates are satisfied is when the space of allowed configurations of the joint is a Lie group, with Q the matrix-representation of the Lie group and v the vector representation of the corresponding Lie algebra. The global mapping F can then be chosen as the mapping of exponential coordinates ϕ around Q. More details about exponential coordinates can be found in Appendix A.3. ▲

Example 2.10. As examples, we show how the commonly encountered joints of Figure 2.4 can be described in the form required by the definition above, *i.e.* in terms of parameterized twists and homogeneous matrices. For each joint, we do not consider possible mechanical end-stops, even though the figures may suggest their existence.

(a) The rotational joint is frequently used in robotics modeling, as robots often contain several rotational joints. For the joint in the figure, the frames Ψ_1 and Ψ_2 differ by a consecutive fixed translation a along y, variable rotation of q about x, and fixed translation b along y, which can be written as

$$H_2^1(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q) & -\sin(q) & a + b\cos(q) \\ 0 & \sin(q) & \cos(q) & b\sin(q) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.26}$$

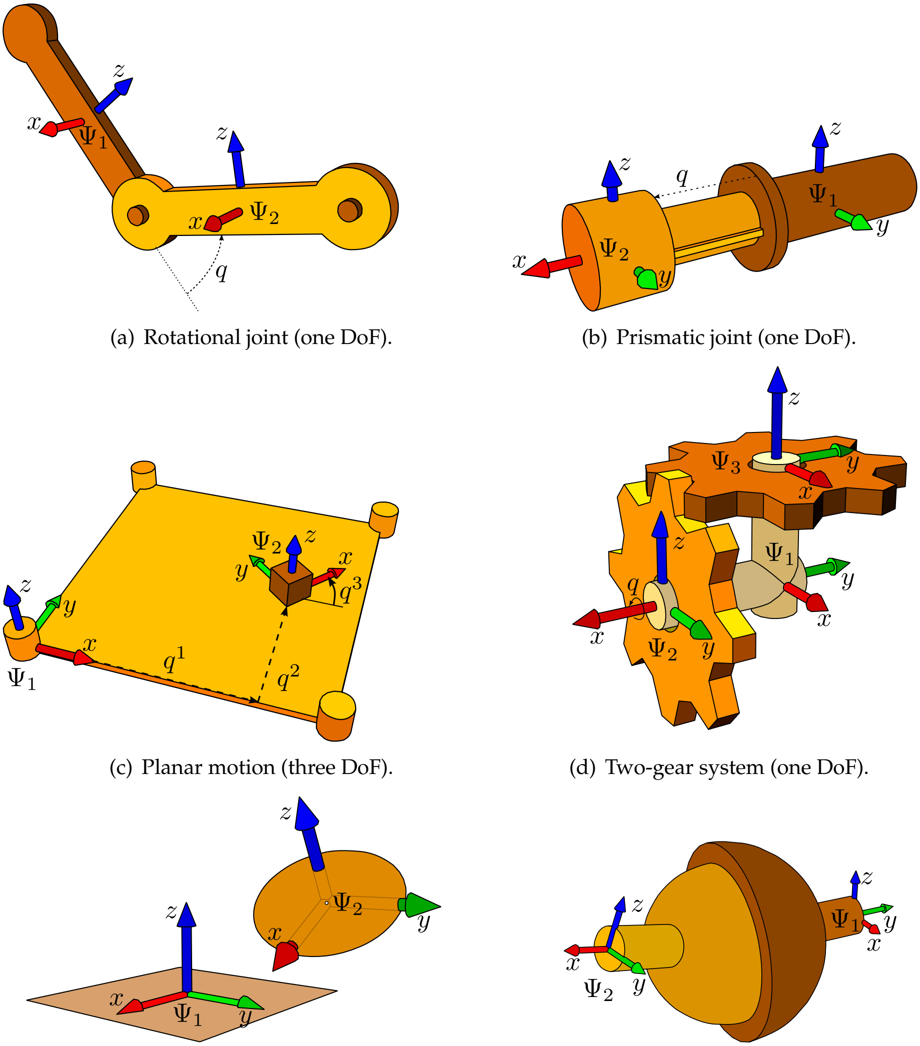
in which we chose simply $Q = q$ to parameterize the allowed configurations. The relative twist is a pure rotation around x displaced over a distance a along y, or in vector form

$$T_2^{1,1} = X(q)v = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -a \end{bmatrix}^T \dot{q} \tag{2.27}$$

which shows that we choose $v = \dot{q}$. Since the configuration parameter q is just an unconstrained scalar, the local coordinate mapping can be chosen $F_q(\phi) = q + \phi$.

(b) The prismatic joint is a simple 1 DoF joint that allows only translation along one axis, in this case x. If we denote by $q = 0$ the situation that Ψ_1 and Ψ_2 are coincident, then the relative configuration can be written as

$$H_2^1(q) = \begin{bmatrix} 1 & 0 & 0 & q \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$



(a) Rotational joint (one DoF).

(b) Prismatic joint (one DoF).

(c) Planar motion (three DoF).

(d) Two-gear system (one DoF).

(e) Free motion (six DoF).

(f) Ball joint (three DoF).

Figure 2.4: Examples of globally parameterized rigid joints.

The corresponding relative twist is a simple linear velocity, or

$$T_2^{1,1} = X(q)v = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \dot{q} \tag{2.29}$$

So also for this joint, we can choose $Q = q$ and $v = \dot{q}$, and hence again $F_q(\phi) = q + \phi$.

(c) A planar joint is used to describe free translation in a plane plus free rotation around the axis perpendicular to that plane. We can choose three coordinates as in the figure to describe the relative configuration as

$$H_1^2(Q) = \begin{bmatrix} \cos(q^3) & -\sin(q^3) & 0 & q^1 \\ \sin(q^3) & \cos(q^3) & 0 & q^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

such that in this case $Q = \begin{bmatrix} q^1 & q^2 & q^3 \end{bmatrix}^T$. The corresponding relative twist can be described as

$$T_2^{1,1} = X(Q)v \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & q^2 & -q^1 & 0 \end{bmatrix}^T \begin{bmatrix} \dot{q}^1 \\ \dot{q}^2 \\ \dot{q}^3 \end{bmatrix} \tag{2.31}$$

which shows again a choice $v = \dot{q}$. Thus, also for planar joints, we can use the coordinate mapping $F_Q(\phi) = Q + \phi$ with $\phi \in \mathbb{R}^3$.

(d) The two-gear system is an example of a system with a kinematic loop: the two gearwheels are connected by rotational joints to a common frame, but also to each other by the constraint that the linear velocity of the contact point is equal on both gearwheels (ideal gear systems have no flexibility or backlash). If we assume for conciseness that the gearwheels have unit radius, we obtain as the constraint of the contact point p that

$$\tilde{T}_2^{1,1} P^1 = \tilde{T}_3^{1,1} P^1 \tag{2.32}$$

$$\begin{bmatrix} 0 & 0 & -\dot{q}^2 & 0 \\ 0 & 0 & 0 & 0 \\ \dot{q}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{q}^3 & 0 & 0 \\ \dot{q}^3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \\ 1 \\ 1 \end{bmatrix} \tag{2.33}$$

where we expressed the velocity of the contact point using the twists as in (2.17). From this equation, we find $\dot{q}^3 = -\dot{q}^2$, *i.e.* that the angular speed of Ψ_3 must be equal but opposite to that of Ψ_2. Taking as parameter $q = q^2 = -q^3$ and as initial condition the configuration shown in the figure, we can describe the relative configurations as

$$H_2^1(q) = \begin{bmatrix} 0 & \cos(q) & -\sin(q) & 0 \\ -1 & 0 & 0 & -1 \\ 0 & \sin(q) & \cos(q) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.34}$$

$$H_3^1(q) = \begin{bmatrix} \cos(q) & \sin(q) & 0 & 0 \\ -\sin(q) & \cos(q) & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.35}$$

and the relative twists as

$$T_2^{1,1} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}^T \dot{q} \tag{2.36}$$

$$T_3^{1,1} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}^T \dot{q} \tag{2.37}$$

hence just two rotational joints as in (a), but now both depending on the same states q and $\dot{q}$. In a similar way, this approach can be used for systems of more than two gears.

(e) Free (6 DoF) relative motion of two rigid bodies can be seen as a degenerate joint (since in fact nothing is joined), but can still be described as a globally parameterized rigid joint. We can just take $Q = H_2^1$ and $v = T_2^{1,1}$, and hence $X(Q) = I$, and $V_Q(\dot{Q})$ as the vector representation of $\dot{Q}Q^{-1}$. As global mapping $F_Q(\phi)$, we can use the exponential coordinates described in Appendix A.3.

(f) The spherical joint allows two bodies to have a fixed relative displacement, but arbitrary rotation. If we choose a rotation matrix R to describe the free relative rotations of the two concentric spheres defining the joint, and if we assume for conciseness that Ψ_1 and Ψ_2 are displaced by vectors p_1 and p_2 from the centers of rotation, respectively, then their relative configuration can be described by

$$H_2^1(R) = \begin{bmatrix} I & p_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & p_2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R & Rp_2 + p_1 \\ 0 & 1 \end{bmatrix} \tag{2.38}$$

while the relative twist can be written as

$$\tilde{T}_2^{1,1} = \dot{H}_2^1 H_1^2 = \begin{bmatrix} \dot{R}R^T & -\dot{R}R^T p_1 \\ 0 & 0 \end{bmatrix} \quad \text{or} \quad T_2^{1,1} = \begin{bmatrix} I \\ \tilde{p}_1 \end{bmatrix} \omega \tag{2.39}$$

where $\tilde{\omega} := \dot{R}R^T$. For this joint, we choose $Q = R$, $v = \omega$, $V_Q(\dot{Q})$ the vector representation of $\dot{R}R^T$, and for the global mapping $F_Q(\phi)$ we choose again exponential coordinates.

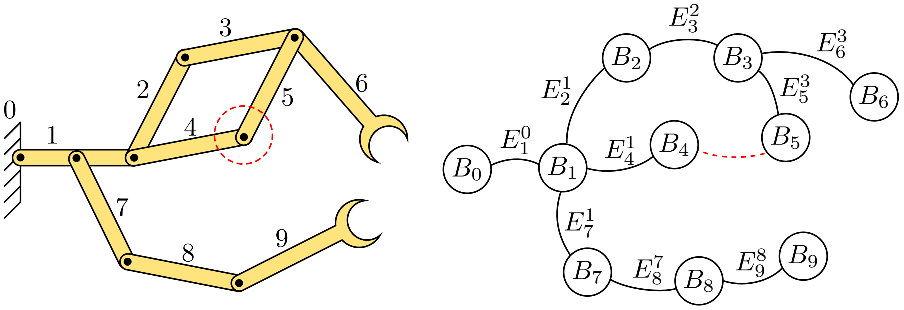


Figure 2.5: Example of a system and its kinematic graph. The graph can be made loop-free, *e.g.* by removing (in first instance) the joint between B_4 and B_5.

The examples show how general kinematic relations between rigid bodies can be represented as globally parameterized rigid joints. △

We can now describe the kinematics of a rigid mechanism as an interconnection of ideal joints attached to rigid bodies. Provided with a practical mechanism (or its preliminary design), it is up to the modeler to decide what parts of the mechanism will be considered rigid and what joints will be considered ideal.

The topology of the idealized mechanism can be represented by a so-called kinematic graph (more on graphs in Appendix B.2), in which each vertex represents a rigid body, and each edge represents an ideal joint; Figure 2.5 shows an example. If the graph contains loops (cyclic paths), this means that the mechanism contains kinematic loops. Unfortunately, the kinematics and dynamics descriptions of this section and the following are only suitable for mechanisms without kinematic loops. Therefore, the graph should be made loop-free, either (if possible) by formulating the mechanism in a different loop-free way, such as was done for the two-gear system in the example, or by initially removing a joint to break the loop, and modeling it in a later phase in terms of a constraint force (see Section 2.4).

For a loop-free kinematic graph, we use the following labeling conventions. First, the vertex labeled B_0 represents the reference frame. Second, a vertex B_i represents a rigid body with coordinate frame Ψ_i. Third, an edge E_j^i denotes an ideal joint that connects body j to body i, where B_i is an element of the path from B_j to B_0 (*i.e.* in the graph, B_i is closer to B_0 than B_j).

In order to obtain a mathematical description of the kinematics of the mechanism, we use the techniques described before to parameterize every matrix H_j^i corresponding to an edge E_j^i using joint coordinates Q^j. Furthermore, we write the relative twist as $T_j^{i,i} = X_j^i(Q^j)v^j$ with v^j the coordinates for the velocities. We combine the local descriptions of the individual joints to obtain an expression for the configuration and twist of each rigid body with respect to the reference frame. This is presented in the following lemma, which is a generalization of the geometric Jacobians as defined for example by Stramigioli (2001) and by Murray et al. (1994) — in the latter reference, it is called the spatial manipulator Jacobian.

Lemma 2.11 (Geometric Jacobian). *Given a loop-free rigid mechanism comprised of rigid bodies B_i with coordinate frames Ψ_i, and n globally parameterized rigid joints E_j^i with associated relative twists $T_j^{i,i} = X_j^i(Q^j)v^j$. The twist of a body B_i relative to the reference frame Ψ_0 can then be written as*

$$T_i^{0,0} = J_i(Q)v := \begin{bmatrix}\sigma_i^1 \operatorname{Ad}_{H_\alpha^0} X_1^\alpha(Q^1) & \cdots & \sigma_i^n \operatorname{Ad}_{H_\omega^0} X_n^\omega(Q^n)\end{bmatrix} \begin{bmatrix} v^1 \\ \vdots \\ v^n \end{bmatrix} \tag{2.40}$$

where J_i is called the (geometric) Jacobian of body i, and where σ_i^j is defined as

$$\sigma_i^j = \begin{cases} 1 & \text{if joint } E_j^\star \text{ is in the path from } B_i \text{ to } B_0, \text{for some } \star \in \{1, \dots, n\} \\ 0 & \text{otherwise} \end{cases}$$

Proof. The proof follows directly from Lemma 2.8, parts (c) and (e), since for a general body in the mechanism, we can write its twist as the sum of all the twists of the consecutive joints between that body and the reference frame, which can be expressed mathematically in the form of the lemma. The variables σ_i^j define the topological structure of the mechanism: only the twists of the joints in the path from B_i to B_0 influence the twist of B_i, not the joints further away from the reference body or joints in different branches of the mechanism. □

Example 2.12. As an example of the proposed techniques, consider the (planar) mechanism of Figure 2.6, containing a horizontal prismatic joint, a rotational joint, a gear system, and another prismatic joint. With the labels and coordinate frames as indicated in the figure, the kinematic graph can be drawn as shown (where the two gear wheels should be considered one joint). Assuming unit-radius gearwheels, the relative configurations can be described as

$$\begin{aligned}
H_1^0 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -q^1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} &
H_2^1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q^2) & -\sin(q^2) & 0 \\ 0 & \sin(q^2) & \cos(q^2) & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
H_3^1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(q^2) & \sin(q^2) & 0 \\ 0 & -\sin(q^2) & \cos(q^2) & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} &
H_4^3 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & q^3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned} \tag{2.41}$$

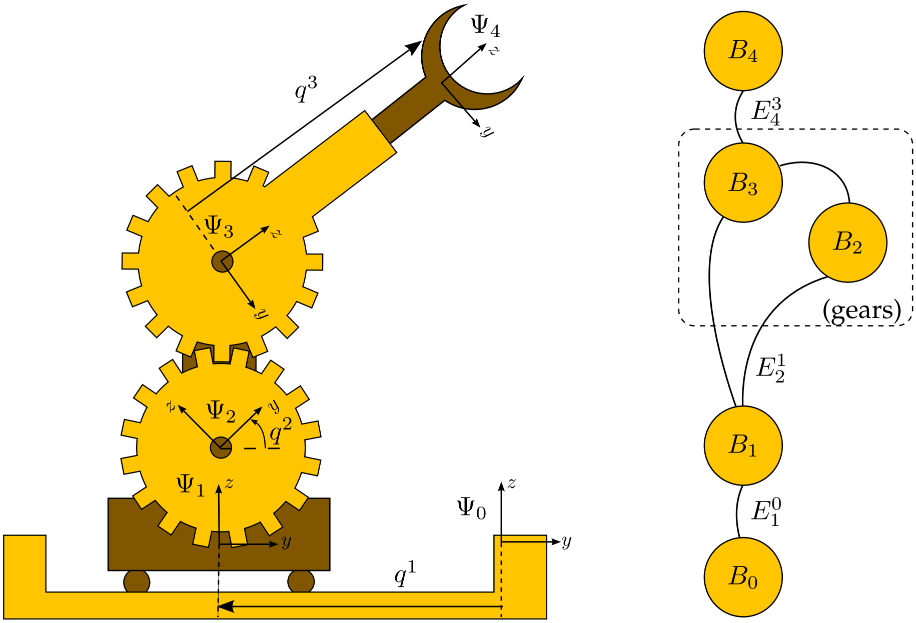


Figure 2.6: Example of a rigid mechanism and its kinematic graph.

and the relative twists can be described as

$$\begin{aligned} T_1^{0,0} &= \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}^T \dot{q}^1 \\ T_2^{1,1} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \dot{q}^2 \\ T_3^{1,1} &= \begin{bmatrix} -1 & 0 & 0 & 0 & -3 & 0 \end{bmatrix}^T \dot{q}^2 \\ T_4^{3,3} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \dot{q}^3 \end{aligned} \tag{2.42}$$

which results in the following Jacobian matrix for the end effector Ψ_4:

$$J_4 = \left[1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} \quad 0 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ q1 \end{pmatrix} + 1 \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ -3 \\ q^1 \end{pmatrix} \quad 1 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \sin(q^2) \\ \cos(q^2) \end{pmatrix} \right] = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & -3 & \sin(q^2) \\ 0 & q^1 & \cos(q^2) \end{bmatrix} \tag{2.43}$$

such that $T_4^{0,0} = J_4 v$, with $v = \begin{bmatrix} \dot{q}^1 & \dot{q}^2 & \dot{q}^3 \end{bmatrix}^T$. The Jacobians of the other links can be constructed in a similar way. △

2.3 Dynamics of open rigid mechanisms

Now that we have a suitable mathematical description language for the kinematics of a rigid mechanism, we can proceed to describing its dynamics, *i.e.* the relation between the forces acting on a mechanism and its motion.

2.3.1 Forces on rigid mechanisms

We first look at the effect of forces on rigid bodies. We want to describe these forces in such a way, that the power P associated with the action of forces on a rigid body with a twist T can be formulated as $P = W^T T$, just like we have $P = f^T v$ for point masses with velocity v and force f. Thus, we look for an expression W describing somehow the forces as a vector which, when paired with the twist, gives the power.

Consider a rigid body with attached coordinate frame Ψ_i and relative twist $T_i^{i,k}$ relative to some body k, and a single arbitrary force f acting on it. Let f_i denote the 3×1 column vector expressing the force magnitude in Ψ_i, and let r^i denote the 3×1 column vector with the coordinates of the point on the body at which the force acts, expressed in Ψ_i. Then we can compute the power associated with the force as the (dual) product of the force f_i and the velocity v^i (expressed in Ψ_i) of the point at which the force acts, or

$$
\begin{aligned}
P = f_i^T v^i &= \begin{bmatrix} f_i^T & 0 \end{bmatrix} \tilde{T}_i^{i,k} \begin{bmatrix} r^i \\ 1 \end{bmatrix} = f_i^T \begin{bmatrix} \tilde{\omega}_i^{i,k} & v_i^{i,k} \end{bmatrix} \begin{bmatrix} r^i \\ 1 \end{bmatrix} \\
&= f_i^T \begin{bmatrix} -\tilde{r}^i & I \end{bmatrix} \begin{bmatrix} \omega_i^{i,k} \\ v_i^{i,k} \end{bmatrix} = f_i^T \begin{bmatrix} -\tilde{r}^i & I \end{bmatrix} T_i^{i,k} = \begin{bmatrix} \tilde{r}^i f_i \\ f_i \end{bmatrix}^T T_i^{i,k}
\end{aligned}
\tag{2.44}
$$

The final step shows how we obtain the desired expression of power as $P = W^T T$ by taking W to be the wrench in the following definition. It is also the only possible solution for W, as the equation for power should hold for all possible twists.

Definition 2.13 (Wrenches). *Let f^j for $j = 1 \ldots k$ be a system of k forces acting on a rigid body i with attached coordinate frame Ψ_i. Let f_i^j be the coordinates in Ψ_i of the force f^j, and let r_j^i be the coordinates in Ψ_i of the point of attachment of the force on the body. The wrench $W^{i,i}$ associated with this system of forces is defined as the 6×1 column vector*

$$
W^{i,i} = \sum_{j=1}^{k} \begin{bmatrix} \tilde{r}_j^i f_i^j \\ f_i^j \end{bmatrix} = \sum_{j=1}^{k} \begin{bmatrix} r_j^i \wedge f_i^j \\ f_i^j \end{bmatrix} \tag{2.45}
$$

where the first superscript denotes the coordinate frame in which the wrench is expressed, and the second superscript denotes the body on which the wrench is applied.

The expression for power (2.44) is linear in the forces, and this allows to take linear combinations of wrenches and obtain again a valid wrench. A special example of this is when the term $\sum f^j$ is zero, while the term $\sum \tilde{r}_j f^j$ is nonzero. This corresponds to a system of forces where the total force is zero while the total moment is nonzero, *i.e.* to the case of a pure torque on a body.

To obtain a method for changing the coordinates of a wrench (*i.e.* expressing the same system of forces on the same body in a different coordinate frame Ψ_j), we again look at the expression for power. Since power is a physical quantity independent of the coordinates, we should have

$$\begin{aligned} P &= (W^{j,i})^T T_i^{j,k} = (W^{i,i})^T T_i^{i,k} \\ &= (W^{i,i})^T \left(\mathrm{Ad}_{H_j^i} T_i^{j,k} \right) = \left(\mathrm{Ad}_{H_j^i}^T W^{i,i} \right)^T T_i^{i,k} \end{aligned} \tag{2.46}$$

and since this should hold for all twists $T_i^{i,k}$, we find the following coordinate transformation rule for wrenches.

$$W^{j,i} = \mathrm{Ad}_{H_j^i}^T W^{i,i} \tag{2.47}$$

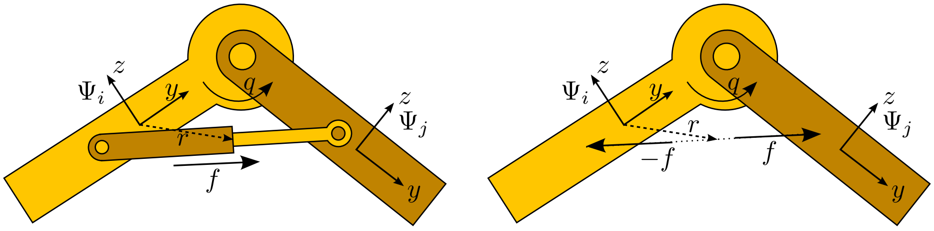
▼ Wrenches are linear operators on twists, as can be seen by the construction of a wrench using the expression for power. In Lie-group terms (Appendix A.3), since twists are elements of the lie algebra $se(3)$, wrenches are elements of the dual space of $se(3)$, denoted $se^*(3)$. When an element of the dual space (a wrench) is applied to an element of the space itself (a twist), the result is a coordinate-independent real number (the power). ▲

When considering rigid mechanisms, we have seen how the twist of a rigid link in the mechanism can be expressed more efficiently in terms of coordinates v together with the Jacobian (Lemma 2.11). We can also represent wrenches on rigid mechanisms efficiently by using a vector τ (commonly called the joint torques) which is collocated with v, *i.e.* such that $\tau^T v$ equals the power associated with the forces τ acting on the mechanism with velocity v. We can relate the external wrenches to the joint torques as in the following lemma.

Lemma 2.14. *For a mechanism with kinematics described by Lemma 2.11, an external wrench $W^{0,i}$ acting on body i has the same effect on the mechanism as a collection of joint torques τ equal to $\tau = J_i^T W^{0,i}$, where J_i is the Jacobian of body i and τ is collocated with v.*

Proof. The power associated with the wrench should be independent of whether it is expressed in joint variables (v and τ) or work space variables (T and W), and hence the following equality holds.

$$\tau^T v = (W^{0,i})^T T_i^{0,0} = (W^{0,i})^T J_i v = (J_i^T W^{0,i})^T v \tag{2.48}$$



(a) common practical implementation

(b) implementation for this example

Figure 2.7: A linear actuator attached to a rotational joint.

Since this should hold for all v, we have the expression for τ in the theorem. □

Note that the mapping of Lemma 2.14 is in general not injective, since the Jacobian can have a kernel, *i.e.* certain wrenches are projected to zero and have no effect on the mechanism. Physically, the wrenches in the kernel of J_i^T are precisely those cancelled out by the constraint forces in the joints.

As an example, consider the effect of gravity on a rigid body of mass m. If we assume gravity to act in the negative z direction of Ψ_0 and have magnitude g, the wrench $W_g^{i,i}$ associated with the gravitational force on body i equals

$$W_g^{i,i} = \begin{bmatrix} \tilde{r}_g f_g \\ f_g \end{bmatrix} = -mg \begin{bmatrix} \tilde{r}_g (R_0^i) e_z \\ R_0^i e_z \end{bmatrix} \quad \text{with} \quad e_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{2.49}$$

where R_0^i is the rotation part of H_0^i and r_g is the location of the center of mass of body i expressed in Ψ_i (often equal to zero). The equivalent joint torque τ_g is then given by $\tau_g = J_i^T \operatorname{Ad}_{H_0^i}^T W_g^{i,i}$. This expression can be used in the dynamics equation for a rigid mechanism, discussed later on in this thesis.

As another example, consider Figure 2.7, showing a rotational joint (with angle q) and linear actuator (force f) attached to it. We can compute the equivalent torques by computing the effect of the wrenches on both bodies i and j. The practical implementation of the actuator would be as in Figure 2.7(a), but to avoid having to deal with the resulting kinematic loop, we implement it as in Figure 2.7(b): we neglect the mass of the actuator and consider it a pure force source, with equal forces f acting in opposing directions on the two connected links.

In this case, the mechanism does not have kinematic loops, and since the two connected links are part of the same branch in the kinematic tree (they are still connected by the rotational joint), their Jacobians are almost equal, except for one extra nonzero column in one of the Jacobians, corresponding to the link that is furthest away from the reference body. Without loss of generality, we assume that J_j has an extra column, which by definition equals $\operatorname{Ad}_{H_i^0} X_j^{i,i}$. The Jacobian

J_j can then be expressed as

$$J_j = J_i + \begin{bmatrix} 0 & \cdots & 0 & \mathrm{Ad}_{H_i^0} X_j^{i,i} & 0 & \cdots & 0 \end{bmatrix} \tag{2.50}$$

Using this expression, we can compute the torques equivalent to the actuator forces as follows.

$$\begin{aligned}
\tau &= J_i^T W^{0,i} + J_j^T W^{0,j} \\
&= J_i^T \,\mathrm{Ad}^T_{H_0^i} \begin{bmatrix} -\tilde{r} f \\ -f \end{bmatrix} + \left(J_i^T + \begin{bmatrix} 0 \cdots 0 & \mathrm{Ad}_{H_i^0} X_j^{i,i} & 0 \cdots 0 \end{bmatrix}^T \right) \mathrm{Ad}^T_{H_0^i} \begin{bmatrix} \tilde{r} f \\ f \end{bmatrix} \\
&= \begin{bmatrix} 0 \cdots 0 & \mathrm{Ad}_{H_i^0} X_j^{i,i} & 0 \cdots 0 \end{bmatrix}^T \mathrm{Ad}^T_{H_0^i} \begin{bmatrix} \tilde{r} f \\ f \end{bmatrix} \\
&= \begin{bmatrix} 0 \cdots 0 & X_j^{i,i} & 0 \cdots 0 \end{bmatrix}^T \begin{bmatrix} \tilde{r} f \\ f \end{bmatrix}
\end{aligned} \tag{2.51}$$

This shows that for internal wrenches acting on two bodies connected by a single joint q^i, the equivalent torque is zero except for τ_i. Furthermore, the torque τ_i is equal to the inner product of the wrench and the matrix describing the relative motion allowed by the joint.

2.3.2 Kinetic co-energy of rigid mechanisms

Let us now proceed to define the kinetic co-energy of a rigid body. Again, we want to express this energy in terms of the twist of the body instead of the velocities of all its comprising points. Since we later want to apply Newton's law to describe the dynamics, we should express the kinetic co-energy in terms of velocities relative to an inertial frame, *i.e.* a coordinate frame (denoted $\Psi_\emptyset$) that is moving at a constant velocity relative to the distant stars. In this thesis, we make the common approximation that $\Psi_\emptyset$ is rigidly attached to the earth and we neglect the absolute motion of the earth.

▼ In the formal definition described in Appendix B.1, energy is a function of the state of a dynamic element, not a function of one of the port variables. However, it is often more convenient to describe the storage in terms of a port variable, in which case the storage function is properly called co-energy to distinguish it from the real energy. As the kinetic storage function above is expressed as a function of the twist, *i.e.* a port variable, it is called the kinetic co-energy. ▲

Given an inertial frame $\Psi_\emptyset$ and a rigid body i with attached frame Ψ_i, we can write the kinetic co-energy U_k^p of a point p on the body with body-coordinates P^i, velocity $v_p^\emptyset$ expressed in $\Psi_\emptyset$, and mass m_p as

$$\begin{aligned}
U_k^p &= \frac{1}{2}m_p(v_p^\emptyset)^T v_p^\emptyset = \frac{1}{2}m_p \left(H_i^\emptyset \tilde{T}_i^{i,\emptyset} \begin{bmatrix} p^i \\ 1 \end{bmatrix} \right)^T H_i^\emptyset \tilde{T}_i^{i,\emptyset} \begin{bmatrix} p^i \\ 1 \end{bmatrix} \\
&= \frac{1}{2}m_p \begin{bmatrix} (p^i)^T & 1 \end{bmatrix} \begin{bmatrix} -\tilde{\omega}_i^{i,\emptyset} & 0 \\ (v_i^{i,\emptyset})^T & 0 \end{bmatrix} \begin{bmatrix} (R_i^\emptyset)^T & 0 \\ (p_i^\emptyset)^T & 1 \end{bmatrix} \begin{bmatrix} R_i^\emptyset & p_i^\emptyset \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{\omega}_i^{i,\emptyset} & v_i^{i,\emptyset} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p^i \\ 1 \end{bmatrix} \\
&= \frac{1}{2}m_p \begin{bmatrix} (\omega_i^{i,\emptyset})^T & (v_i^{i,\emptyset})^T \end{bmatrix} \begin{bmatrix} \tilde{p}^i & 0 \\ I & 0 \end{bmatrix} \begin{bmatrix} (R_i^\emptyset)^T R_i^\emptyset & (R_i^\emptyset)^T p_i^\emptyset \\ (p_i^\emptyset)^T R_i^\emptyset & 1 + (p_i^\emptyset)^T p_i^\emptyset \end{bmatrix} \begin{bmatrix} -\tilde{p}^i & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \omega_i^{i,\emptyset} \\ v_i^{i,\emptyset} \end{bmatrix} \\
&= \frac{1}{2}m_p (T_i^{i,\emptyset})^T \begin{bmatrix} (\tilde{p}^i)^T \tilde{p}^i & \tilde{p}^i \\ (\tilde{p}^i)^T & I \end{bmatrix} T_i^{i,\emptyset} \qquad (2.52)
\end{aligned}$$

which shows that the kinetic co-energy of a point mass can be expressed as a quadratic function of the twist in body coordinates that is independent of the relative configuration $H_i^\emptyset$ of the body. If we sum (by means of an integral) the kinetic co-energy of all the point masses of the body, we obtain an expression for the total kinetic co-energy of a rigid body equal to

$$U_k = \frac{1}{2}(T_i^{i,\emptyset})^T I^{i,i} T_i^{i,\emptyset} \qquad (2.53)$$

with $I^{i,i}$ called the inertia matrix of the rigid body, defined as

$$I^{i,i} := \int_B m(x) \begin{bmatrix} (\tilde{p}^i(x))^T \tilde{p}^i(x) & \tilde{p}^i(x) \\ (\tilde{p}^i(x))^T & I \end{bmatrix}^T dx \qquad (2.54)$$

and $m(x)$ and $p^i(x)$ the mass density and body-fixed coordinates of the point x, respectively, and where the integral is taken over the volume B of the rigid body.

By construction, the inertia matrix is positive semi-definite. Furthermore, if a rigid body has a finite volume and mass distribution, then the inertia matrix will be strictly positive-definite, meaning that for every non-zero twist the body has nonzero kinetic co-energy.

If the coordinate frame on the body is chosen to be in the center of mass of the body with the axes of the frame aligned with the principle directions of the inertia ellipsoid of the rigid body, then the inertia matrix is a diagonal matrix of the form

$$I^{i,i} = \operatorname{diag} \begin{bmatrix} J_x & J_y & J_z & M & M & M \end{bmatrix} \qquad (2.55)$$

with M the total mass of the body, and J_x, J_y, and J_z the moments of inertia around the axes x, y, and z of Ψ_i. The parameters in this matrix are often easy to measure or compute for a given rigid body, and therefore the body-fixed frames are often chosen in this way.

If the inertia matrix is known in one coordinate frame, it can also be expressed in a different coordinate frame by using a suitable transformation rule. Since the kinetic co-energy of a body does not depend on the chosen coordinate frame, we have

$$U_k = \frac{1}{2}(T_i^{j,\emptyset})^T I^{j,i} T_i^{j,\emptyset} = \frac{1}{2}(T_i^{i,\emptyset})^T I^{i,i} T_i^{i,\emptyset} = \frac{1}{2}(T_i^{i,\emptyset})^T \, \mathrm{Ad}^T_{H^i_j} \, I^{i,i} \, \mathrm{Ad}_{H^i_j} \, T_i^{j,\emptyset} \tag{2.56}$$

and since this should hold for all twists, we obtain the following transformation rule for inertia matrices.

$$I^{j,i} = \mathrm{Ad}^T_{H^i_j} \, I^{i,i} \, \mathrm{Ad}_{H^i_j} \tag{2.57}$$

With the kinetic co-energy of one rigid body known, we can express the kinetic co-energy of a rigid mechanism as the sum of the kinetic co-energies of all the rigid bodies in the mechanism. This is expressed formally in the following lemma.

Lemma 2.15 (Kinetic Co-Energy of a General Rigid Mechanism). *Given a rigid mechanism consisting of m rigid links and n globally parameterized rigid joints as defined in Definition 2.9, and with kinematics described by a loop-free kinematic graph and corresponding Jacobians as in Lemma 2.11. The kinetic co-energy of the mechanism is equal to*

$$U_k = \frac{1}{2} v^T M(Q) v \tag{2.58}$$

where $v = \begin{bmatrix} v^1 & \cdots & v^n \end{bmatrix}^T$ with v^i the velocity state of link i, and where the mass matrix $M(Q)$ is defined as

$$M(Q) := \sum_{i=1}^{m} J_i^T(Q) \, \mathrm{Ad}^T_{H^i_\emptyset(Q)} \, I^{i,i} \, \mathrm{Ad}_{H^i_\emptyset(Q)} \, J_i(Q) \tag{2.59}$$

with $J_i(Q)$ the Jacobian of link i and $I^{i,i}$ the body-fixed inertia matrix of link i.

Proof. The kinetic co-energy of a mechanism of several rigid bodies is equal to the sum of the kinetic co-energies of the bodies. The kinetic co-energy of body i can be written as (2.53), which, using the Jacobian, can be expressed as

$$(U_k)_i = \frac{1}{2}(\mathrm{Ad}_{H^i_\emptyset} \, T_i^{\emptyset,\emptyset})^T I^{i,i} (\mathrm{Ad}_{H^i_\emptyset} \, T_i^{\emptyset,\emptyset}) = \frac{1}{2} v^T \left(J_i^T(Q) \, \mathrm{Ad}^T_{H^i_\emptyset} \, I^{i,i} \, \mathrm{Ad}_{H^i_\emptyset} \, J_i(Q) \right) v$$

Summing this expression for all rigid bodies gives the total kinetic co-energy of the mechanism. □

2.3.3 Dynamic equations of rigid mechanisms

For the derivation of the dynamics of a rigid mechanism, we start from the Euler-Lagrange equations (Goldstein 1980). For a mechanical system with generalized coordinates q and $\dot{q}$, kinetic co-energy $U_k(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ with M invertible, potential energy $U_p(q)$, and input forces u collocated with $y = B^T(q)\dot{q}$, the time-evolution of q is described by the differential equation

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \frac{\partial \mathcal{L}}{\partial q} = B(q)u \tag{2.60}$$

with the Lagrangian $\mathcal{L}(q,\dot{q}) = U_k(q,\dot{q}) - U_p(q)$. If we define $p := \frac{\partial \mathcal{L}}{\partial \dot{q}} = M(q)\dot{q}$, this can be written equivalently in port-Hamiltonian form as

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ B(q) \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & B^T(q) \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} \end{aligned} \tag{2.61}$$

with the Hamiltonian equal to $H(q,p) = p^T\dot{q} - \mathcal{L}(q,\dot{q}) = \frac{1}{2}p^T M^{-1}(q)p + U_p(q)$, and (u,y) defining a power port.

These equations can be generalized to what are known as the Boltzmann-Hamel equations, as described for example by Whittaker (1998). Instead of using $\dot{q}$ to describe the velocity, we use a general vector v of the form $v = S(q)\dot{q}$ with $S(q)$ a continuously-differentiable invertible matrix. Using these coordinates (sometimes called quasi-coordinates), the dynamics can be written as follows.

Lemma 2.16 (Boltzmann-Hamel Equations). *Given a mechanical system with generalized coordinates q, velocities $v = S(q)\dot{q}$ (with S invertible and continuously differentiable), inputs u collocated with v, potential energy $U_p(q)$ and kinetic co-energy expressed as $U_k(q,v) = \frac{1}{2}v^T M(q)v$. If we define $p := M(q)v$, we can write the dynamics as*

$$\frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} 0 & S^{-1} \\ -S^{-T} & S^{-T}\left(\frac{\partial^T(S^T p)}{\partial q} - \frac{\partial(S^T p)}{\partial q}\right)S^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \tag{2.62}$$

$$v = \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix}$$

where the energy equals $H(q,p) = \frac{1}{2}p^T M^{-1}(q)p + U_p(q)$.

Proof. The kinetic co-energy U_k can be written as a function of $\dot{q}$ instead of v simply by substituting the relation between v and $\dot{q}$:

$$U_k = \frac{1}{2} v^T M(q) v = \frac{1}{2} \dot{q}^T S^T(q) M(q) S(q) \dot{q} \tag{2.63}$$

Using this expression, we can write the Lagrangian as a function of q and $\dot{q}$ as

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} \dot{q}^T S^T(q) M(q) S(q) \dot{q} - U_p(q) \tag{2.64}$$

and its partial derivatives with respect to q and $\dot{q}$ can be computed as

$$\frac{\partial \mathcal{L}}{\partial q} = \frac{1}{2} \left(\frac{\partial(\dot{q}^T S^T(q) p)}{\partial q} + \frac{\partial(v^T M(q) v)}{\partial q} + \frac{\partial(p^T S(q) \dot{q})}{\partial q} \right) - \frac{\partial U_p(q)}{\partial q} \tag{2.65}$$

$$\frac{\partial \mathcal{L}}{\partial \dot{q}} = S^T(q) M(q) S(q) \dot{q} = S^T(q) p \tag{2.66}$$

where we used the definition of p in the theorem, and where we explicitly denoted the variables depending on q. Substituting these expressions in the Euler-Lagrange equations (2.60) (with $B = S^T$ since u is collocated with v), we obtain

$$\begin{aligned} S^T u &= \frac{d}{dt}\left(S^T p\right) - \frac{1}{2} \left(\frac{\partial(\dot{q}^T S^T(q) p)}{\partial q} + \frac{\partial(v^T M(q) v)}{\partial q} + \frac{\partial(p^T S(q) \dot{q})}{\partial q} \right) + \frac{\partial U_p(q)}{\partial q} \\ &= \frac{\partial(S^T(q) p)}{\partial q} \dot{q} + S^T \dot{p} - \frac{\partial^T(S^T(q) p)}{\partial q} \dot{q} - \frac{1}{2} \frac{\partial(v^T M(q) v)}{\partial q} + \frac{\partial U_p}{\partial q} \\ &= \frac{\partial(S^T(q) p)}{\partial q} \dot{q} + S^T \dot{p} - \frac{\partial^T(S^T(q) p)}{\partial q} \dot{q} + \frac{1}{2} \frac{\partial(p^T M^{-1}(q) p)}{\partial q} + \frac{\partial U_p}{\partial q} \\ &= \frac{\partial(S^T(q) p)}{\partial q} \dot{q} + S^T \dot{p} - \frac{\partial^T(S^T(q) p)}{\partial q} \dot{q} + \frac{\partial H}{\partial q} \end{aligned} \tag{2.67}$$

where the third equality follows from the identity

$$0 = \frac{\partial I}{\partial a} = \frac{\partial(X^{-1} X)}{\partial a} = \frac{\partial X^{-1}}{\partial a} X + X^{-1} \frac{\partial X}{\partial a} \tag{2.68}$$

which holds for any invertible matrix X and any variable a. The resulting dynamic equation can be written directly in the form of (2.62). □

The Boltzmann-Hamel equations serve as a basis for the derivation of the dynamic equations for general rigid mechanisms as presented in the following theorem. This theorem gives the dynamic equations for general rigid mechanisms without kinematic loops, comprised of rigid links and interconnected by globally parameterized rigid joints.

Theorem 2.17 (Dynamics of a General Rigid Mechanism). *Given a rigid mechanism as in Lemma 2.15 with total energy equal to kinetic plus potential energy, and with power-port (u, v). Its dynamics are described by the following equations.*

$$\dot{Q} = V_Q^{-1}(v) \tag{2.69}$$

$$\dot{p} = -S_0^{-T}\frac{\partial H}{\partial \phi} + S_0^{-T}\left(\frac{\partial^T(S^T p)}{\partial \phi} - \frac{\partial (S^T p)}{\partial \phi}\right)\Bigg|_{\phi=0} S_0^{-1}M^{-1}p + u \tag{2.70}$$

$$v = M^{-1}(Q)p \tag{2.71}$$

with the energy function $H(Q,p) = \frac{1}{2}p^T M^{-1}(Q)p + U_p(Q)$ and where we defined

$$S_i(Q^i,\phi^i)\dot{\phi}^i := V^i_{F_i(\phi^i)}\left(\frac{\partial}{\partial \phi^i}F_i(\phi^i)\dot{\phi}^i\right) \quad \forall\, \dot{\phi}^i \tag{2.72}$$

$$S(Q,\phi) := \operatorname{diag}\left[S_1(Q^1,\phi^1) \quad \cdots \quad S_n(Q^n,\phi^n)\right] \tag{2.73}$$

$$S_0 := S(Q,0) \tag{2.74}$$

$$\frac{\partial H}{\partial \phi^i} := \frac{\partial H}{\partial Q^i} \circ \frac{\partial F_i}{\partial \phi^i}\bigg|_{\phi=0} \quad \forall\, \dot{\phi}^i \tag{2.75}$$

with $F_i(\phi^i)$ the local coordinate function of the globally parameterized rigid joint i with local coordinates ϕ^i.

Proof. First, consider an arbitrary (fixed) state (Q,p) of the system, with $p = M(Q)v$. Then, by definition of a globally parameterized rigid joint, the functions $F_i(\phi^i)$ define local coordinates ϕ^i around Q^i for all joints i. With Q fixed and only ϕ changing, time derivatives of the configuration can locally be expressed as $\frac{\partial}{\partial \phi^i}F_i(\phi^i)\dot{\phi}^i$ (where, for joints with more than one DoF, summation over all components of ϕ^i should take place). Joint velocities can be expressed using the mappings V^i for each joint, but instead of expressing them as functions of Q^i and $\dot{Q}^i$, we express them as functions of ϕ^i and $\dot{\phi}^i$, as shown in (2.25). Since by definition, V^i is linear in $\dot{Q}$, the expression V^i can be written as a matrix-vector product $S_i(Q^i,\phi^i)\dot{\phi}^i$ as in (2.72). Collecting the matrices S_i in the matrix S as defined in (2.73), we obtain an expression for the velocities as $S(Q,\phi)\dot{\phi}$.

Second, we can write the dynamic equations for the system locally around Q in port-Hamiltonian form by using (2.62) with coordinates ϕ (and parameterized by Q). The momentum can locally be written as

$$p := M(F_Q(\phi))S(Q,\phi)\dot{\phi} \tag{2.76}$$

giving a Hamiltonian equal to $\frac{1}{2}p^T M^{-1}(F_Q(\phi))p + U_p(Q)$. The Boltzmann-Hamel equations in port-Hamiltonian form then become

$$\dot{\phi} = S^{-1}M^{-1}(F_Q(\phi))p \tag{2.77}$$

$$\dot{p} = -S^{-T}\frac{\partial H}{\partial \phi} + S^{-T}\left(\frac{\partial^T(S^Tp)}{\partial \phi} - \frac{\partial(S^Tp)}{\partial \phi}\right)S^{-1}M^{-1}(F_Q(\phi))p + u \tag{2.78}$$

where we abbreviated $S = S(Q,\phi)$ and where the partial derivative of H with respect to ϕ should be taken as

$$\frac{\partial H(Q,p)}{\partial \phi^i} := \left.\frac{\partial H(Q,p)}{\partial Q^i}\right|_{Q=F_Q(\phi)} \circ \frac{\partial F_i(\phi^i)}{\partial \phi^i} \tag{2.79}$$

with the concatenation operator ($\circ$) meaning a sum over all elements of the matrix Q^i times the partial derivative of $F_i(\phi^i)$ with respect to ϕ^i.

Finally, we obtain the dynamics *at* a certain point Q with pseudo-velocity v by evaluating the equations at $\phi = 0$. By definition, for $\phi = 0$ we have $F_i(0) = Q^i$ and $S(Q,0)\dot{\phi} = v$, which results in the dynamic equations (2.69)-(2.71) given in the theorem. The conditions on F_i given in Definition 2.9 ensure the existence and uniqueness of the partial derivatives of F_i and S^Tp with respect to ϕ. □

The equations in the theorem are basically just the Boltzmann-Hamel equations, but formulated in such a way that the local coordinates ϕ and $\dot{\phi}$ do not appear (except in the partial derivatives of S_i and F_i, evaluated at $\phi = 0$). Thus, the equations in the theorem are valid globally, for all Q, and they can be readily implemented in simulation, by means of two ordinary differential equations for the states Q and p plus an equation for the intermediate variable v.

Moreover, the structure of the mass matrix and the Jacobians can be used to write the partial derivative of the Hamiltonian to ϕ in a compact form. This can be seen from the definitions (2.40) and (2.59), which can be written as

$$M = \sum_{i=1}^{m}\begin{bmatrix}\sigma_i^1 (X_1^\alpha)^T(Q^1)\,\mathrm{Ad}^T_{H^i_\alpha} \\ \vdots \\ \sigma_i^n (X_n^\omega)^T(Q^n)\,\mathrm{Ad}^T_{H^i_\omega}\end{bmatrix} I^{i,i}\begin{bmatrix}\sigma_i^1\,\mathrm{Ad}_{H^i_\alpha}X_1^\alpha(Q^1) & \cdots & \sigma_i^n\,\mathrm{Ad}_{H^i_\omega}X_n^\omega(Q^n)\end{bmatrix} \tag{2.80}$$

This shows that the mass matrix (and hence the Hamiltonian) only depends on ϕ^i indirectly and locally, *i.e.* only through the matrix $X_i^\star(Q^i)$ defining the relative degrees of freedom of joint i and through the homogeneous matrix $H_i^\star$ defining the relative configuration of joint i. The partial derivatives of these terms to ϕ^i can be easily obtained (looking only locally at the joint description), and hence the partial derivative of the mass matrix follows by the chain rule for differentiation.

Example 2.18. As a first example, we consider the dynamics of a single rigid body with frame Ψ_0, freely floating in space (we ignore gravity in this example).

This system can be described as a (degenerate) mechanism with one rigid link and one free 6-DoF joint as in Figure 2.4(e), using $Q = H_0^\emptyset$ and $v = T_0^{0,\emptyset}$. From Appendix A.3, we find that the twist can be expressed locally around a certain Q with exponential coordinates ϕ as $T_0^{0,\emptyset} = S(Q,\phi)\dot{\phi}$ with

$$S(Q,\phi) = \sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} \operatorname{ad}_\phi^k = I - \frac{1}{2}\operatorname{ad}_\phi + \frac{1}{6}\operatorname{ad}_\phi^2 - \dots \tag{2.81}$$

where ad is as defined in Lemma 2.8. From the expression, we can see that $S(Q,0) = I$, and we can evaluate the partial derivative with respect to ϕ at $\phi = 0$ as follows.

$$\begin{aligned}
\left.\frac{\partial (S^T(Q,\phi)p)}{\partial \phi}\right|_{\phi=0} &= \frac{\partial}{\partial\phi}\left.\left(p - \frac{1}{2}\operatorname{ad}_\phi^T p + \frac{1}{6}(\operatorname{ad}_\phi^2)^T p - \dots\right)\right|_{\phi=0} \\
&= \frac{\partial}{\partial\phi}\left.\left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} p_a \\ p_b \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \tilde{\phi}_a & \tilde{\phi}_b \\ 0 & \tilde{\phi}_a \end{bmatrix}\begin{bmatrix} p_a \\ p_b \end{bmatrix} + \dots\right)\right|_{\phi=0} \\
&= \frac{\partial}{\partial\phi}\left.\left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} p_a \\ p_b \end{bmatrix} - \frac{1}{2}\begin{bmatrix} \tilde{p}_a & \tilde{p}_b \\ \tilde{p}_b & 0 \end{bmatrix}\begin{bmatrix} \phi_a \\ \phi_b \end{bmatrix} + \dots\right)\right|_{\phi=0} \\
&= -\frac{1}{2}\begin{bmatrix} \tilde{p}_a & \tilde{p}_b \\ \tilde{p}_b & 0 \end{bmatrix}
\end{aligned} \tag{2.82}$$

where the subscripts a and b denote the first three and second three components of a vector, respectively, and where the third equality follows from the identity

$$\tilde{x}y = x \wedge y = -y \wedge x = -\tilde{y}x \tag{2.83}$$

which holds for all vectors $x, y \in \mathbb{R}^3$. Since we used the twist in body coordinates and ignored gravity, the inertia tensor $I^{0,0}$ and the Hamiltonian do not depend on Q, and hence not on ϕ either. So if we substitute the expressions for S and its partial derivative into Theorem 2.17, we obtain

$$\dot{H}_0^\emptyset = H_0^\emptyset \tilde{T}_0^{0,\emptyset} \tag{2.84}$$

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix} p_a \\ p_b \end{bmatrix} &= \left(-\frac{1}{2}\begin{bmatrix} \tilde{p}_a^T & \tilde{p}_b^T \\ \tilde{p}_b^T & 0 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \tilde{p}_a & \tilde{p}_b \\ \tilde{p}_b & 0 \end{bmatrix}\right) T_0^{0,\emptyset} + u \\
&= \begin{bmatrix} \tilde{p}_a & \tilde{p}_b \\ \tilde{p}_b & 0 \end{bmatrix} T_0^{0,\emptyset} + W^{0,0}
\end{aligned} \tag{2.85}$$

as the equations describing the dynamics of a rigid body in body-coordinates.

Instead of using a momentum in body coordinates, we can also use the momentum $p_\emptyset$ in world coordinates, defined as

$$p_\emptyset := I^{\emptyset,\emptyset} T_0^{\emptyset,\emptyset} = \operatorname{Ad}_{H_\emptyset^0}^T I^{0,0} \operatorname{Ad}_{H_\emptyset^0} \operatorname{Ad}_{H_0^\emptyset} T_0^{0,\emptyset} = \operatorname{Ad}_{H_\emptyset^0}^T I^{0,0} T_0^{0,\emptyset} = \operatorname{Ad}_{H_\emptyset^0}^T p \tag{2.86}$$

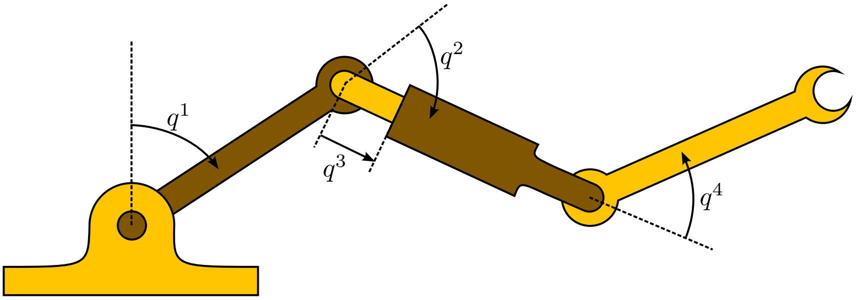


Figure 2.8: Example setup of a rigid mechanism for which Q is a vector.

Its time derivative can be obtained from the expression for $\dot{p}$ as follows

$$\begin{aligned}
\dot{p}_\emptyset &= \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset}\, p) = \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})p + \mathrm{Ad}^T_{H^0_\emptyset}\, \dot{p} = \mathrm{ad}^T_{T^{\emptyset,0}_\emptyset}\, \mathrm{Ad}^T_{H^0_\emptyset}\, p + \mathrm{Ad}^T_{H^0_\emptyset}\, \dot{p} \\
&= \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})p + \mathrm{Ad}^T_{H^0_\emptyset} \begin{bmatrix} \tilde{p}_a & \tilde{p}_b \\ \tilde{p}_b & 0 \end{bmatrix} \begin{bmatrix} \omega^{0,\emptyset}_0 \\ v^{0,\emptyset}_0 \end{bmatrix} + \mathrm{Ad}^T_{H^0_\emptyset}\, W^{0,0} \\
&= \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})p - \mathrm{Ad}^T_{H^0_\emptyset} \begin{bmatrix} \tilde{\omega}^{0,\emptyset}_0 & \tilde{v}^{0,\emptyset}_0 \\ 0 & \tilde{\omega}^{0,\emptyset}_0 \end{bmatrix} \begin{bmatrix} p_a \\ p_b \end{bmatrix} + W^{\emptyset,0} \\
&= \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})\, \mathrm{Ad}^T_{H^\emptyset_0}\, p_\emptyset + \mathrm{Ad}^T_{H^0_\emptyset}\, \mathrm{ad}^T_{T^{0,\emptyset}_0}\, \mathrm{Ad}^T_{H^\emptyset_0}\, p_\emptyset + W^{\emptyset,0} \\
&= \frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})\, \mathrm{Ad}^T_{H^\emptyset_0}\, p_\emptyset + \mathrm{Ad}^T_{H^0_\emptyset}\, \frac{d}{dt}(\mathrm{Ad}^T_{H^\emptyset_0})p_\emptyset + W^{\emptyset,0} \\
&= \left(\frac{d}{dt}(\mathrm{Ad}^T_{H^0_\emptyset})\, \mathrm{Ad}^T_{H^\emptyset_0} + \mathrm{Ad}^T_{H^0_\emptyset}\, \frac{d}{dt}(\mathrm{Ad}^T_{H^\emptyset_0})\right) p_\emptyset + W^{\emptyset,0} \\
&= W^{\emptyset,0} \qquad (2.87)
\end{aligned}$$

where we used property (f) of Lemma 2.8 and where the last equality follows from (2.68). The resulting equation is a generalization of Newton's second law to the case of a rigid body: the time-derivative of the momentum of a rigid body (when expressed in inertial coordinates) is equal to the forces (wrenches) applied to it, and hence the momentum is conserved if no external forces are applied. △

Example 2.19. As a second example of the general result of Theorem 2.17, we consider the dynamics of a class of mechanisms for which the allowed relative configuration Q^i of all joints can be described by a vector, such as the mechanism shown in Figure 2.8, with full direct actuation τ. In this case, we can choose $Q = q$ and $v = \dot{q}$, as well as $F_Q(\phi) = Q + \phi$, and hence we obtain $S(Q, \phi) = I$ and the corresponding dynamics

$$\frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \tau \tag{2.88}$$

with $H(q,p) = \frac{1}{2}p^T M^{-1}(q)p + U_p(q)$, *i.e.* the normal Euler-Lagrange equations in port-Hamiltonian form. The equation for $\dot{p}$ can be expanded as

$$\begin{aligned} \dot{p} &= \frac{d}{dt}(M(q)\dot{q}) = \frac{\partial(M(q)\dot{q})}{\partial q}\dot{q} + M(q)\ddot{q} \\ &= -\frac{\partial H}{\partial q} + \tau = \frac{1}{2}\frac{\partial(\dot{q}^T M(q)\dot{q})}{\partial q} - \frac{\partial U_p}{\partial q} + \tau \end{aligned} \tag{2.89}$$

which can be written as

$$\begin{aligned} &M(q)\ddot{q} + \left(\frac{\partial(M(q)\dot{q})}{\partial q^i} - \frac{1}{2}\frac{\partial^T(M(q)\dot{q})}{\partial q} \right)\dot{q} + \frac{\partial U_p}{\partial q} \\ &=: M(q)\ddot{q} + C(q,\dot{q})\dot{q} + V(q) = \tau \end{aligned} \tag{2.90}$$

which is the formulation of the dynamics of this class of mechanisms as commonly encountered in textbooks. The matrix $C(q,\dot{q})$ describes the centrifugal and Coriolis effects. △

2.4 Kinematic loops and nonholonomic constraints

The previous sections have shown how to obtain a set of explicit differential equations for general mechanisms with ideal holonomic joints and a loop-free kinematic graph, *i.e.* for systems without kinematic loops and for which the space of allowed configurations has the same dimension as the space of allowed instantaneous velocities. In this section, we look at more general mechanisms, possibly containing kinematic loops and nonholonomic constraints.

For conciseness reasons, we choose to start here from systems described by coordinates $q \in \mathbb{R}^n$ and $p \in \mathbb{R}^n$, with Hamiltonian equal to $H(q,p) = \frac{1}{2}p^T M^{-1}(q)p + U_p(q)$ and described by the differential equations (2.61). The results from this section can be extended to more complex descriptions such as in Theorem 2.17; the equations just become a little longer.

We consider constraints that can be expressed in the form

$$A^T(q)\dot{q} = 0 \tag{2.91}$$

with $A(q)$ a continuously differentiable $n \times m$ matrix. Furthermore, the associated constraint forces λ are collocated with $A^T\dot{q}$. This formulation means that we consider position-varying velocity constraints that are linear in the velocities with constraint forces that do not influence the energy of the system (since the power equals $\lambda^T(A^T\dot{q}) = 0$ if the constraints are satisfied).

Under these conditions, the mechanical system with constraints can be represented by the implicit set of differential equations

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ A(q) & B(q) \end{bmatrix}\begin{bmatrix} \lambda \\ u \end{bmatrix} \\ \begin{bmatrix} 0 \\ y \end{bmatrix} &= \begin{bmatrix} 0 & A^T(q) \\ 0 & B^T(q) \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} \end{aligned} \tag{2.92}$$

where λ is to be determined such that the output zero constraint is and remains satisfied.

Several approaches can be taken in the simulation of systems of the form (2.92). First, it could be left to a numerical solver to obtain the correct λ at each simulation step. We do not further pursue this direction, but refer to Ascher & Petzold (1998) for an introduction to numerical techniques for dealing with these sets of algebraic and differential equations.

A second approach is to obtain λ by differentiation of the constraint:

$$\begin{aligned} 0 = \frac{d}{dt}\left(A^T\frac{\partial H}{\partial p}\right) &= \frac{d}{dt}\left(A^TM^{-1}p\right) = \frac{\partial(A^TM^{-1}p)}{\partial q}\dot{q} + A^TM^{-1}\dot{p} \\ &= \frac{\partial(A^TM^{-1}p)}{\partial q}\frac{\partial H}{\partial p} + A^TM^{-1}\left(-\frac{\partial H}{\partial q} + A\lambda + Bu\right) \end{aligned} \tag{2.93}$$

from which λ can be solved as

$$\left(A^TM^{-1}A\right)\lambda = A^TM^{-1}\frac{\partial H}{\partial q} - A^TM^{-1}Bu - \frac{\partial(A^TM^{-1}p)}{\partial q}\frac{\partial H}{\partial p} \tag{2.94}$$

which has a unique solution for λ if the matrix A has full column rank. This solution gives the necessary force λ that keeps the velocities $A^T\dot{q}$ *constant* (since we looked at the derivative of the constraint). This means that if the constraint (2.91) is satisfied at some time, then applying this λ ensures that (2.91) *remains* satisfied. The approach can be extended to avoid numerical drift (which may cause $A^T\dot{q}$ to become nonzero), by changing the differential equation as follows

$$\frac{d}{dt}\left(A^T\frac{\partial H}{\partial p}\right) = 0 \qquad \rightarrow \qquad \frac{d}{dt}\left(A^T\frac{\partial H}{\partial p}\right) = -\beta\left(A^T\frac{\partial H}{\partial p}\right) \tag{2.95}$$

with $\beta > 0$ a damping coefficient. We apply this method in Section 3.3, where we discuss the computation of contact forces for rigid contact between two rigid bodies.

A third approach (presented by Duindam, Blankenstein & Stramigioli (2004), based on results by van der Schaft & Maschke (1994)) is to find new coordinates for the momentum such that satisfying the constraints is equivalent to some of these coordinates being zero. This approach can be taken if the rank of A is constant (say, m) for all q, by defining an $n \times (n-m)$ matrix $S(q)$ satisfying $A^T(q)M^{-1}(q)S(q) = 0$, *i.e.* such that the columns of S span the space of momenta compatible with the constraints. The dynamics can then be described by the explicit differential equations of the following theorem.

Theorem 2.20 (Dynamics of Nonholonomic Systems). *Given a system with nonholonomic constraints, described by (2.92) and let $A(q)$ have constant rank m for all q. If we can find a continuously differentiable $n \times (n-m)$ matrix $S(q)$ satisfying $A^T(q)M^{-1}(q)S(q) = 0$, then the dynamics of the constrained system can be described equivalently by the following port-Hamiltonian system*

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ \alpha \end{bmatrix} &= \bar{J}(q,\alpha)\begin{bmatrix} \frac{\partial \bar{H}}{\partial q} \\ \frac{\partial \bar{H}}{\partial \alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{M}S^TM^{-1} \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & M^{-1}S\bar{M} \end{bmatrix}\begin{bmatrix} \frac{\partial \bar{H}}{\partial q} \\ \frac{\partial \bar{H}}{\partial \alpha} \end{bmatrix} \end{aligned} \tag{2.96}$$

with α an $(n-m)$-vector, and $\bar{M}$, $\bar{H}$, and $\bar{J}$ defined as

$$\bar{M}(q) := \left(S^T(q)M^{-1}(q)S(q)\right)^{-1} \tag{2.97}$$

$$\bar{H}(q,\alpha) := H(q, S(q)\alpha) = \frac{1}{2}\alpha^T\bar{M}^{-1}\alpha + V(q) \tag{2.98}$$

$$\bar{J}(q,\alpha) := \begin{bmatrix} 0 & M^{-1}S\bar{M} \\ -\bar{M}S^TM^{-1} & \bar{M}S^TM^{-1}\left(\frac{\partial^T(S\alpha)}{\partial q} - \frac{\partial(S\alpha)}{\partial q}\right)M^{-1}S\bar{M} \end{bmatrix} \tag{2.99}$$

Proof. The vector α is the new reduced-order coordinate vector for the momentum satisfying $p = S(q)\alpha$. By definition of $S(q)$, the constraints $0 = A^TM^{-1}p = A^TM^{-1}S\alpha$ are automatically satisfied for all α. Furthermore, the definition of the new Hamiltonian $\bar{H}$ is just H with $S(q)\alpha$ substituted for p. From this definition we find

$$\frac{\partial \bar{H}}{\partial q} = \frac{\partial H}{\partial q} + \frac{\partial^T(S(q)\alpha)}{\partial q}\frac{\partial H}{\partial p} \tag{2.100}$$

$$\frac{\partial \bar{H}}{\partial \alpha} = S^T(q)\frac{\partial H}{\partial p} \tag{2.101}$$

$$\begin{tikzcd}
\dot{q} \arrow[r, shift left, "S_H"] \arrow[d, shift left, "S_H^T M S_H"] & v = \frac{\partial H}{\partial p} \arrow[l, shift left, "S_H^{-1}"] \arrow[r, "S_N^T"] \arrow[d, shift left, "M"] & \frac{\partial \bar{H}}{\partial \alpha} \arrow[d, shift left, "\bar{M}"] \\
S_H^T p \arrow[u, shift left, "S_H^{-1} M^{-1} S_H^{-T}"] \arrow[r, shift left, "S_H^{-T}"] & p \arrow[u, shift left, "M^{-1}"] \arrow[l, shift left, "S_H^T"] & \alpha \arrow[l, "S_N"] \arrow[u, shift left, "\bar{M}^{-1}"]
\end{tikzcd}$$

Figure 2.9: Commutation diagram for a system with position coordinates q, pseudo-velocities v, and momenta α allowed by the nonholonomic constraints.

Furthermore, we can write the differential equations for q and p as

$$\dot{q} = \frac{\partial H}{\partial p} = M^{-1}p = M^{-1}S\alpha = M^{-1}S\bar{M}\bar{M}^{-1}\alpha = M^{-1}S\bar{M}\frac{\partial \bar{H}}{\partial \alpha} \tag{2.102}$$

$$\dot{p} = \frac{d}{dt}\left(S(q)\alpha\right) = \frac{\partial (S\alpha)}{\partial q}\dot{q} + S\dot{\alpha} = -\frac{\partial H}{\partial q} + A\lambda + Bu \tag{2.103}$$

Pre-multiplying the last equation with $\bar{M}S^TM^{-1}$ gives

$$\begin{aligned}
&\bar{M}S^TM^{-1}\frac{\partial (S\alpha)}{\partial q}M^{-1}S\bar{M}\frac{\partial \bar{H}}{\partial \alpha} + \dot{\alpha} \\
&= -\bar{M}S^TM^{-1}\left(\frac{\partial \bar{H}}{\partial q} - \frac{\partial^T(S\alpha)}{\partial q}M^{-1}S\alpha\right) + \bar{M}\underbrace{S^TM^{-1}A}_{0}\lambda + \bar{M}S^TM^{-1}Bu \\
&= -\bar{M}S^TM^{-1}\left(\frac{\partial \bar{H}}{\partial q} - \frac{\partial^T(S\alpha)}{\partial q}M^{-1}S\alpha\right)
\end{aligned} \tag{2.104}$$

which, together with the equation for $\dot{q}$, can be formulated as in the theorem. □

The results of Theorem 2.20 are an extension of the Boltzmann-Hamel equations (2.62), with the S^{-1} replaced by a pseudo-inverse $M^{-1}S(S^TM^{-1}S)^{-1}$. However, the mass matrix M of the system without kinematic constraints is necessary for the definition of the new system in reduced coordinates, and hence it is useful to distinguish the holonomic joints (where S is invertible) from the nonholonomic joints (where S is not invertible).

To illustrate the precise difference between the Boltzmann-Hamel equations of Lemma 2.16 and the equations for nonholonomic systems of Theorem 2.20, consider a combination of both systems, *i.e.* a system with coordinates q, pseudo-velocities $v = S_H(q)\dot{q}$ (with S_H invertible), corresponding momentum variables

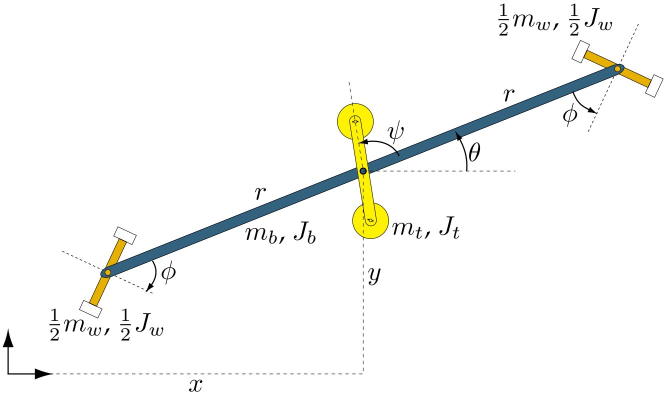

$\frac{1}{2}m_w$, $\frac{1}{2}J_w$
r
ϕ
ψ
θ
r
m_b, J_b
m_t, J_t
ϕ
y
$\frac{1}{2}m_w$, $\frac{1}{2}J_w$
x

Figure 2.10: Schematic (top) view of the snakeboard.

$p = Mv$, and nonholonomic constraints expressed by reduced-order coordinates α satisfying $p = S_N(q)\alpha$ (with S_N not invertible).

The relation between the various variables is illustrated in Figure 2.9. Apart from the relations mentioned above, the definitions of $H(q,p)$ and $\bar{H}(q,\alpha)$ allow to construct the other relations in the diagram. From the diagram, we can see that $\dot{q}$ can be obtained from $\frac{\partial H}{\partial p}$ just by multiplication with S_H^{-1}. However, since S_N^T is not invertible, v can be obtained from $\frac{\partial \bar{H}}{\partial \alpha}$ by going via the variables α and p, *i.e.* as $v = M^{-1} S_N \bar{M} \frac{\partial \bar{H}}{\partial \alpha}$. This illustrates the difference between Lemma 2.16 and Theorem 2.20.

Example 2.21. Figure 2.10 shows the snakeboard, an interesting mechanical system with nonholonomic constraints. It is a commercially available locomotion device (Snakeboard U.S.A. 2005), very similar to the skateboard, but with extra degrees of freedom that allow rotation of the front and rear wheel bases around the vertical axis, such that the direction of motion of the wheels can be changed. Using certain combinations of motion of the feet (to control the wheel base angles) and of the upper body, the person riding the snakeboard can increase the total forward momentum, without touching the ground or directly driving the wheels.

The snakeboard is a relatively simple system that still has many scientifically interesting properties, and it is hence used in numerous studies on modeling, analysis, and control of nonholonomic mechanical systems. Lewis et al. (1994) and Bullo & Zefran (2001) used Lie-bracket analysis, Vela (2003) applied averaging analysis, and Ostrowski & Burdick (1998), Bloch (2003), Bloch et al. (1996), Ostrowski (1999), and Blankenstein (2003) used principle bundles and momentum maps. These studies have revealed many interesting physical and geometrical properties and interpretations, and provide general tools to analyze nonholonomic systems.

In this example, we show how an explicit model of the snakeboard can be obtained by choosing new coordinates for the momentum, as described by Theorem 2.20. The advantage of this formulation is not only the possibility of fast simulation using straight-forward ODE algorithms, but also that the resulting equations turn out to be very simple and allow clear interpretation in terms of energy flows. More information about the use of this model for controller design can be found in (Duindam & Stramigioli 2004*a*) and (Duindam, Blankenstein & Stramigioli 2004).

To construct a model of the snakeboard, we assume that it consists of three rigid parts (body, torso and wheels), each with its own mass m_* and inertia J_*. We can write the total mass and inertia as follows

$$\begin{aligned} m &:= m_b + m_t + m_w && \text{total mass} \\ J &:= J_b + J_t + J_w + m_w r^2 && \text{total inertia around } (x, y) \end{aligned}$$

and we make the simplifying assumptions that (i) the wheels rotate at equal but opposite angles (only one coordinate ϕ is used) and (ii) the parameters are such that $J = mr^2$. These assumptions are standard in the analysis of the snakeboard and simplify the equations without affecting the essential geometry of the problem. We choose the states of the system as positions and momenta

$$\begin{aligned} q &= \begin{bmatrix} x & y & \theta & \psi & \phi \end{bmatrix}^T \\ p &= \begin{bmatrix} p_x & p_y & p_\theta & p_\psi & p_\phi \end{bmatrix}^T \end{aligned} \tag{2.105}$$

and the Hamiltonian of the system is just the kinetic energy $H = \frac{1}{2} p^T M^{-1} p$ with

$$M = \begin{bmatrix} m & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 \\ 0 & 0 & J & J_t & 0 \\ 0 & 0 & J_t & J_t & 0 \\ 0 & 0 & 0 & 0 & J_w \end{bmatrix} \tag{2.106}$$

We furthermore assume to have direct torque control on the torso and the wheels, which can be described by the input mapping

$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.107}$$

Finally, the (nonholonomic) kinematic constraints are that the wheels are not allowed to slip sideways. This can be represented by the matrix A as

$$A = \begin{bmatrix} \sin(\theta - \phi) & -\cos(\theta - \phi) & r\cos(\phi) & 0 & 0 \\ \sin(\theta + \phi) & -\cos(\theta + \phi) & -r\cos(\phi) & 0 & 0 \end{bmatrix}^T \tag{2.108}$$

which has rank 2 for all q, except for $\phi = \pm\frac{1}{2}\pi$, *i.e.* when the wheels are turned completely sideways. The two elements of $A^T\dot{q}$ describe the sideways velocities of the two wheel pairs (front and rear).

We can find an example of a suitable matrix $S(q)$ describing the allowed momenta as follows

$$S(q) = \begin{bmatrix} \frac{mr}{\beta}\cos(\theta)\cos(\phi) & \frac{mr}{\beta}\sin(\theta)\cos(\phi) & \frac{mr^2-J_t}{\beta}\sin(\phi) & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{2.109}$$

with $\beta(\phi) := \sqrt{mr^2 - J_t\sin^2(\phi)}$. Duindam, Blankenstein & Stramigioli (2004) show how different choices of $S(q)$ lead to the different models available in literature. If we use this matrix $S(q)$ in Theorem 2.20, we obtain the following explicit differential equations that describe the dynamics of the constrained snakeboard.

$$\frac{d}{dt}q = \begin{bmatrix} \frac{r}{\beta}\cos(\theta)\cos(\phi) & 0 & 0 \\ \frac{r}{\beta}\sin(\theta)\cos(\phi) & 0 & 0 \\ \frac{1}{\beta}\sin(\phi) & 0 & 0 \\ -\frac{1}{\beta}\sin(\phi) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \frac{\alpha_2}{J_t} \\ \frac{\alpha_3}{J_w} \end{bmatrix} \tag{2.110}$$

$$\frac{d}{dt}\alpha = \begin{bmatrix} \frac{1}{\beta}\sin(\phi) & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{2.111}$$

with $H = \frac{1}{2}\alpha^T\bar{M}^{-1}\alpha$ and $\bar{M} = \text{diag}\begin{bmatrix}1 & J_t & J_w\end{bmatrix}$. The mass matrix $\bar{M}$ is constant and diagonal, which is due to the choice of S: the matrix M^{-1} defines a metric on the space of momenta, and the columns of S are chosen to be mutually orthogonal and of constant norm in that metric.

The resulting dynamic equations turn out to be very simple, expressing the total kinetic energy of the system as the sum of three terms, each depending on only one momentum coordinate: α_3 represents the energy in the wheels, α_2 the energy stored in the torso, and α_1 the energy stored in the other motions of the snakeboard, in particular the forward momentum of the board (a quantity that may be important for control).

Since the energy function depends only on α and not on q, we can represent the energy flows in the snakeboard as in Figure 2.11 in terms of a bond graph (bond graphs are explained in detail in Appendix B.2). From the bond graph, we see that the wheels do not have an energy coupling to the rest of the snakeboard; they only influence the system indirectly through modulation of the $\mathbb{MTF}$. Furthermore, the input port (u_1,$\dot{\psi}$) can exchange energy with the torso ($\mathbb{I}$ element with parameter J_t) as well as the forward motions ($\mathbb{I}$ element with parameter 1), but only if $\sin(\phi) \neq 0$. All of this is clear from intuitive physical reasoning about

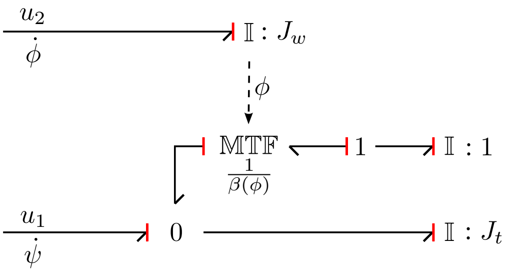


Figure 2.11: Bond graph of the snakeboard in reduced momentum coordinates.

the snakeboard (*e.g.* if the wheels are straight ($\phi = 0$), exerting a torque u_1 only makes the torso rotate, not the rest of the snakeboard); the reduced equations and the corresponding bond graph prove this reasoning and quantify it. △

Chapter 3

Modeling of Compliant and Rigid Contact

The second building block for dynamic models of walking robots is a model of the contact between the feet of the robot and the ground. Contacts are important in the analysis and control of walking motions for two main reasons. First, the dynamic behavior of a walking robot is strongly influenced by whether or not the feet have contact with the ground: making and breaking contact results in strong nonlinearities (or even switching behavior) in the dynamics. Secondly, impacts determine to some extent the efficiency of a walking cycle: large impacts (stamping) cause large energy losses during the walking cycle.

In the development of a contact model suitable for walking robots, two goals need to be balanced. First, the model should (obviously) give an accurate enough representation of physical contact behavior. But secondly, the model should not be too complex, in order to allow quick simulation, tractable analysis, and controller design. To balance between these conflicting modeling goals (which are present in some form in all modeling tasks), we discuss two contact models in this section: one compliant model, which is most suited for accurate simulation, and one rigid model, which is most suited for analysis and controller design.

The compliant model has the advantage of being more generally applicable as it can be represented as a standard finite port-Hamiltonian system that can be interconnected between bodies to model possible contact between these bodies. If more bodies can come into contact, more copies of the model can be added without any other special requirements or adjustments. Disadvantages of the compliant model are that it results in stiff differential equations when the contacting surfaces are relatively stiff, and that it is very hard to analyze a walking robot with a compliant contact model, due to the presence of both single-support and double-support phases during walking.

The rigid contact model has the advantage of being easy to simulate and analyze, since impacts can be represented by a simple step in the momentum, possibly completely eliminating the need for a double-support phase (this depends on the exact mechanical structure of the walking robot). The disadvantage is that this model needs to be implemented as an integrated part of the mechanism in contact, instead of a separate, more or less independent port-Hamiltonian system. Furthermore, it is much more difficult to generalize to multiple contact points; we only discuss an extension to two contacting points in this thesis.

In the rest of this chapter, we first discuss models of the kinematics of contact, *i.e.* the equations that describe when and where bodies come into contact, and then describe the two contact models, first the compliant model, and then the rigid model.

3.1 Contact kinematics

The purpose of contact kinematics is to detect collisions between rigid bodies, based on the kinematic information of these bodies. The kinematic contact analysis should detect both when (at what simulation time) and where (at which points of the two rigid bodies) a collision occurs. This means that it continuously needs to monitor the positions and velocities of the bodies in a mechanism and, using information about the surfaces of these bodies, check for collisions.

In the literature, especially in the field of computer graphics, we can find many ways to handle collision detection. The main objective of these approaches is to be able to detect, in real-time, possible contact between many objects in a virtual 3D environment. This objective is motivated by applications in interactive haptic virtual environments as well as computer games. A common way to speed up the detection process is to use rough approximations of objects when they are still far away from each other, and more detailed surfaces models as they get closer. Various shapes and volumes can be used as rough approximation, such as bounding boxes (Cohen et al. 1995, Suri et al. 1999), bounding spheres (Hubbard 1996), and convex polytopes (Klosowski et al. 1998). Jiménez et al. (2001) provide a survey of the various related problems and existing solutions.

When modeling walking robots, or multibody systems in general, the number of rigid bodies that may come in contact is usually limited. For a bipedal robot, contact normally only occurs between the feet and the ground; if other parts of the robot touch the ground, walking has clearly failed, and there is usually no need for accurate simulation of the robot falling down and breaking to pieces.

Therefore, it is not necessary for our purposes to have these demands of real-time detection of contact between many objects. Instead of judging a collision-detection mechanism on these aspects, focus can be put on accuracy of the detection, as well as ease of implementation in standard simulation software, without the need for extra iterative numerical techniques.

In this section, we consider collision detection and monitoring between two specific rigid bodies. These bodies may be part of a rigid mechanism or freely floating; this aspect does not influence the kinematic analysis. We use a description of the surfaces of the two bodies as well as their kinematic state (relative configuration and twist) to solve the generalized contact kinematics problem: to compute the positions and velocities of two points (called generalized contact points) on the two bodies that are closest to each other.

▼ We use the Euclidean metric to define closest distance of the points on the surface, as these distances are measured in the three-dimensional world in which we live. Technically, the metric is not important for collision detection, though, as a distance zero in the Euclidean metric implies distance zero in *any* positive-definite metric. ▲

We discuss two approaches that solve the generalized contact kinematics problem, and that are suitable for use in the modeling of multibody systems. We start with several examples of commonly encountered situations for which the contact kinematics can be derived directly, due to the elementary surface shapes or other simplifying assumptions. Then, we show a more general indirect (differential) approach, suitable for a broader class of surfaces.

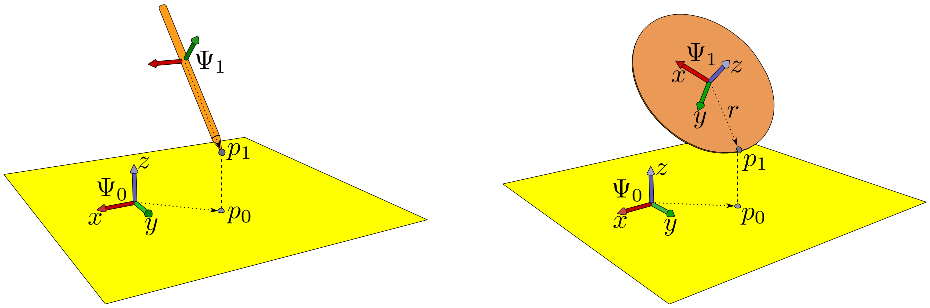
3.1.1 Direct derivation for simple cases

We give four examples of common situations where the contact kinematics can be obtained directly due to simplifying assumptions or simple contacting surfaces. The objective is to obtain the positions and velocities of the closest points (labeled p_0 and p_1) on two surfaces, given the shape of the surfaces and their relative configuration H_1^0 and twist $T_1^{0,0}$.

Example 3.1 (fixed point on a rigid body over a plane). Consider first the situation of Figure 3.1(a), with an object over a plane. In some situations, for example a walking robot with point feet, it can be assumed that the rigid body can only come in contact with the plane at one specific point on the body, such that the contact point p_1 has fixed coordinates P_1^1 (expressed in Ψ_1). In this case, the coordinates P_0^0 of the contact point p_0 follow easily as

$$P_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_1^0 P_1^1 \tag{3.1}$$

that is, we compute the coordinates of p_1 in Ψ_0 (by the multiplication $P_1^0 = H_1^0 P_1^1$) and then project the point onto the horizontal (x, y)-plane along the z-axis. From these expressions for P_1^1 and P_0^0, we can compute the velocities of the contact



(a) Fixed point on a rigid body.

(b) Flat disc over a plane.

Figure 3.1: Two simple examples for which the contact kinematics can be derived directly from the kinematics of the contacting bodies.

points as

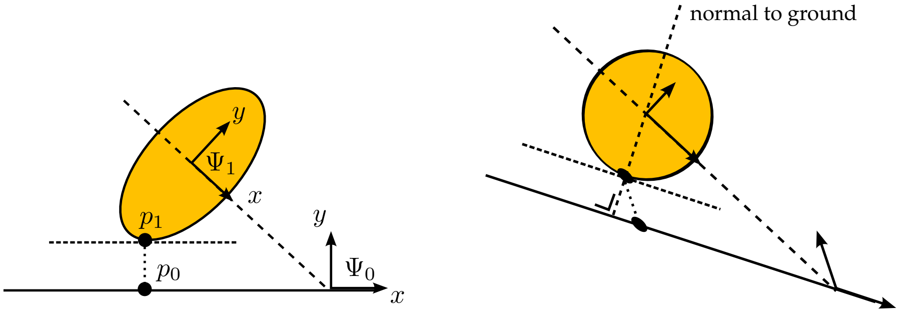
$$\dot{P}_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tilde{T}_1^{0,0} H_1^0 P_1^1 \qquad \dot{P}_1^1 = 0 \tag{3.2}$$

where $\dot{P}_1^1 = 0$ since p_1 is fixed to body 1. We thus obtain a direct expression of the contact kinematics in terms of the relative configuration H_1^0 and the relative twist $T_1^{0,0}$ of the two bodies. △

Example 3.2 (flat disc over a plane). Consider as a second example Figure 3.1(b), showing a flat disc of radius r over a plane. If we assume the disc to be above the plane at all times and choose Ψ_1 to have its origin in the center of the disc and its z-axis normal to the disc, then the coordinates of the contact points can be expressed as

$$P_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_1^0 P_1^1 \qquad P_1^1 = \begin{bmatrix} \frac{-r z_x}{\sqrt{z_x^2 + z_y^2 + \epsilon}} \\ \frac{-r z_y}{\sqrt{z_x^2 + z_y^2 + \epsilon}} \\ 0 \\ 1 \end{bmatrix} \tag{3.3}$$

where P_0^0 is as in (3.1), $z_x := H_0^1[1,3]$, $z_y := H_0^1[2,3]$ are the x and y coordinates in Ψ_1 of the z-axis of Ψ_0, and $\epsilon > 0$ is a small number added for numerical reasons, to avoid division by zero when the disc is exactly horizontal, which would lead to simulation problems. From these expressions we can find the velocities of the



(a) Ellipse over a line with closest points p_0 and p_1.

(b) Non-uniform scaling turns the ellipse into a circle.

Figure 3.2: Direct contact kinematics of an ellipse over a line using a non-uniform scaling operation.

contact point as follows.

$$\dot{P}_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left(\tilde{T}_1^{0,0} H_1^0 P_1^1 + H_1^0 \dot{P}_1^1 \right) \tag{3.4}$$

$$\dot{P}_1^1 = \frac{r}{\sqrt{z_x^2 + z_y^2 + \epsilon}} \begin{bmatrix} \dot{z}_x - \frac{z_x^2 \dot{z}_x + z_x z_y \dot{z}_y}{z_x^2 + z_y^2 + \epsilon} \\ \dot{z}_y - \frac{z_x z_y \dot{z}_x + z_y^2 \dot{z}_y}{z_x^2 + z_y^2 + \epsilon} \\ 0 \\ 0 \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} \dot{z}_x \\ \dot{z}_y \\ \dot{z}_z \\ 0 \end{bmatrix} = \tilde{T}_1^{0,0} H_0^1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.5}$$

which again expresses the contact kinematics directly in terms of the kinematics of the two rigid bodies. △

Example 3.3 (ellipse over a flat floor). Consider the case of an ellipse over a line as depicted in Figure 3.2(a). To obtain the coordinates of the contact point for this system, we apply the following procedure, illustrated in Figure 3.2(b).

First, note that the tangent line to the ellipse at the contact point p_1 is parallel to the ground (this is proved later on in Theorem 3.6). Second, we can assume without loss of generality that the coordinate frame Ψ_1 is chosen in the center of the ellipse with coordinate axes in the directions of its principle axes. Third, we apply a non-uniform scaling of the setup along the axes of Ψ_1 in such a way that the ellipse is transformed into the unit circle. As Figure 3.2(b) shows, this scaling affects the angles between lines, but parallel lines remain parallel, and hence the tangent line at p_1 is still parallel to the ground. Fourth, we compute

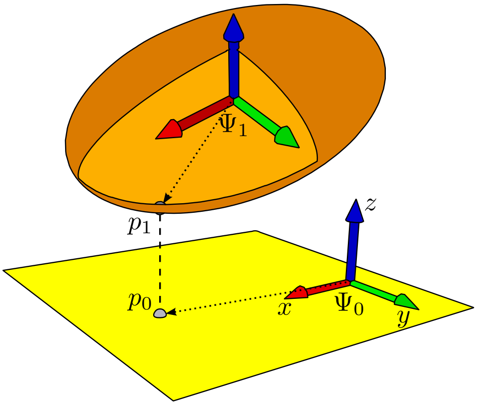


Figure 3.3: Ellipsoid over a plane and the two closest points p_0 and p_1.

the line normal (in the Euclidean sense in the distorted figure) to the ground, and hence also to the tangent line to the circle at the contact point: by definition of a circle, this line connects the contact point to the center of the circle. Finally, the results are transformed back to the original undistorted figure by reversing the scaling operation.

This approach can be described mathematically as the following equation, which gives the coordinates of contact points p_0 and p_1.

$$P_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_1^0 P_1^1 \qquad P_1^1 = \begin{bmatrix} S\frac{(S^{-1}x_0^1)\times(S^{-1}z_0^1)}{\|(S^{-1}x_0^1)\times(S^{-1}z_0^1)\|} \\ 1 \end{bmatrix} \tag{3.6}$$

where $S = \text{diag}\begin{bmatrix} r_x & r_y & 1 \end{bmatrix}$ is the scaling matrix containing the radii of the ellipse in the (x, y) coordinate directions, and x_0^1 and z_0^1 are the coordinates of the unit vectors along the x and z directions of Ψ_0, when expressed in Ψ_1 (the z-direction of both frames is outward perpendicular to the page, such that the frames are orthonormal and right-handed). The velocities of the contact points can then be computed as in the two examples before, by taking the time-derivatives of the expressions (3.6). △

Example 3.4 (ellipsoid over a flat floor). As a final example, we consider an ellipsoid over a plane, as depicted in Figure 3.3. To obtain the coordinates of the contact points for this system, we can directly extend the procedure described in Example 3.3 for the planar case. Again, we apply a non-uniform scaling, now by a matrix $S = \text{diag}\begin{bmatrix} r_x & r_y & r_z \end{bmatrix}$, *i.e.* with the radii of the ellipsoid in the coordinate directions of Ψ_1. The contact points follow immediately by generalizing (3.6) to

$$P_0^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} H_1^0 P_1^1 \qquad P_1^1 = \begin{bmatrix} S \frac{(S^{-1}y_0^1)\times(S^{-1}x_0^1)}{\|(S^{-1}x_0^1)\times(S^{-1}y_0^1)\|} \\ 1 \end{bmatrix} \tag{3.7}$$

Again, the velocities of the contact point can be found by computing the time-derivatives of the expressions (3.7). △

3.1.2 Indirect derivation for general case

In case the contacting surfaces are too complex to allow a direct contact kinematics description, we can take an indirect differential approach. This approach uses a differential equation to describe the evolution of the contact points over the surface, and requires several functions that describe the surfaces of the bodies. To be precise, we denote by $\mathcal{S}_i \subset \mathbb{R}^3$ the (oriented) surface of body i and define the following functions

- a twice continuously differentiable function

  $$f_i : D_i \to \mathcal{S}_i; \quad u_i \mapsto f_i(u_i)$$

  that maps local coordinates $u_i \in D \subset \mathbb{R}^2$ to a point on the surface.

- the tangent mapping of f_i

  $$f_{i*}(u_i) : \mathbb{R}^2 \to T\mathcal{S}_i; \quad \dot{u}_i \mapsto f_{i*}(u_i)\dot{u}_i$$

  mapping coordinate velocities to contact point velocities tangent to the surface, depending on the coordinates u_i. We assume f_i to be a well-defined coordinate mapping such that f_{i*} has full column rank at all points u_i.

- the Gauss map

  $$g_i : \mathcal{S}_i \to \mathbb{S}^2; \quad q \mapsto g_i(q)$$

  mapping a point on the surface to a point on the unit sphere representing the direction of the outward normal to the surface. One way to obtain this function is by taking the cross-product of the two columns of f_* and normalizing the resulting vector.

- the tangent mapping of g_i

  $$g_{i*}(q) : T\mathcal{S}_i \to T\mathbb{S}^2; \quad \dot{q} \mapsto g_{i*}(q)\dot{q}$$

  that relates velocities tangent to the surfaces to the change of the normal vector to the surface, depending on the surface point q.

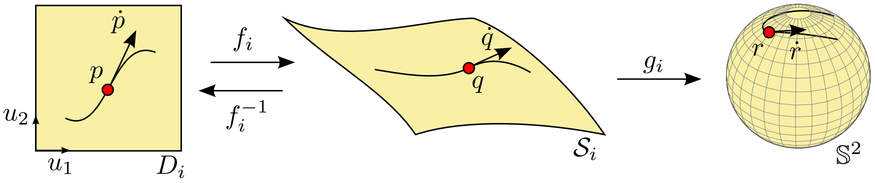


Figure 3.4: The mappings f_i and g_i describe coordinates and surface normals.

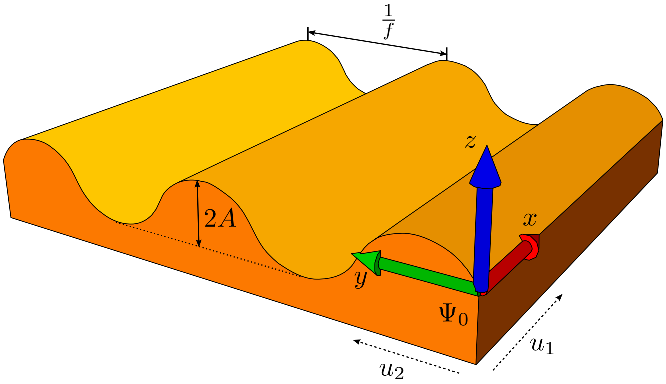


Figure 3.5: Sinus-shaped surface with frame Ψ_0 and local coordinates u.

For an interpretation of the mappings, consider Figure 3.4. It shows the coordinate space D_i, the surface $\mathcal{S}_i$, and the sphere $\mathbb{S}^2$. The invertible mapping f_i relates coordinates (two real numbers) in D_i to an element of the surface, and maps for example the point $p \in D_i$ to the point $q = f_i(p) \in \mathcal{S}_i$. The Gauss mapping g_i relates the point q on the surface to the point $r = g_i(q)$ on the sphere, in such a way that the (outward) normal vector to the surface at q touches the sphere at r, when placed at the center of the sphere. Note that the Gauss mapping is in general not invertible, since several points on the surface can have the same normal vector (an extreme case is the plane, for which *all* points of the surface have the same normal vector).

The tangent mappings f_{i*} and g_{i*} relate tangent elements in the various spaces. If, for example, the (differentiable) curves in the figure are point-wise related by the mappings f_i and g_i, then the velocity vectors are related as $\dot{q}_i = f_{i*}(p)\dot{p}$ and $\dot{r} = g_{i*}(p)\dot{p}$. The mapping g_{i*} also describes the curvature of the surface: it quantifies how the direction of the normal vector changes when moving along the surface. If the surface $\mathcal{S}_i$ is highly curved, then the normal vector will change direction quickly and g_{i*} will be large.

Example 3.5. Consider as an example surface the sinusoidal ground presented in Figure 3.5. If we use coordinates u_1 and u_2 as in the figure, we can represent points on the surface by the mapping

$$f_i(u_1, u_2) = \begin{bmatrix} u_1 \\ u_2 \\ A\sin(2\pi f u_2) \\ 1 \end{bmatrix} \tag{3.8}$$

with tangent mapping

$$f_{i*}(u_1, u_2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 2\pi A f\cos(2\pi f u_2) \\ 0 & 0 \end{bmatrix} \tag{3.9}$$

The Gauss mapping and its derivative can be chosen as

$$g_i(x, y, z) = \frac{1}{\sqrt{1 + 4\pi^2 A^2 f^2 \cos^2(2\pi f y)}} \begin{bmatrix} 0 \\ -2\pi A f \cos(2\pi f y) \\ 1 \\ 0 \end{bmatrix} \tag{3.10}$$

$$g_{i*}(x, y, z) = \frac{4\pi^2 A f^2}{(1 + 4\pi^2 A^2 f^2 \cos^2(2\pi f y))^{\frac{3}{2}}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -\sin(2\pi f y) & 0 & 0 \\ 0 & \pi A f \sin(4\pi f y) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.11}$$

which only have physical meaning for points $(x, y, z, 1)$ on the surface and velocities $(\dot{x}, \dot{y}, \dot{z}, 0)$ tangent to the surface. △

For simple analytically defined surfaces, the mappings f_i and g_i and their tangent mappings can be easily computed. However, this may not be so easy for more involved surfaces, such as those obtained from CAD software or measurement data. The shapes of such surfaces are usually available as point clouds or large sets of triangles. In this case, the surface can be approximated by software analysis and interpolation of the surface data. Ambrosius (2005) shows an algorithm that constructs the necessary surface functions by approximating the data obtained as the output of commonly used CAD packages.

Remark. The Gauss mapping g_i is defined here as mapping points on the surface to points on the sphere, independently of the coordinate mapping f_i. This is useful for the theoretical results in Theorem 3.6, which are stated in a general coordinate-independent way.

However, to perform calculations in practice, it is convenient to define a function $\hat{g}_i(u) := g_i(f_i(u))$ as mapping from the coordinates u directly to the sphere, in which case the mappings g_i and g_{i*} should be replaced by

$$g_i(q) \to \hat{g}_i(f_i^{-1}(q)) \qquad g_{i*}(q) \to \hat{g}_{i*} f_{i*}^{-1}(q) \tag{3.12}$$

where f_{i*}^{-1} is only valid for vectors tangent to the surface and can be implemented *e.g.* as pseudo-inverse $(f_{i*}^T f_{i*})^{-1} f_{i*}^T$ (since the matrix f_{i*} is not square, and hence not invertible in the usual sense). Since f_{i*} has full column rank, its pseudo-inverse exists and can be computed in this way. △

When the two surfaces under consideration have been described by the functions f_i, g_i, f_{i*}, and g_{i*} (for $i = 1, 2$), we can formulate the differential contact kinematics as in the theorem below. The results are based on Duindam & Stramigioli (2003*a*), and form a coordinate-independent extension to the work of Montana (1989*a*, 1989*b*), Pfeiffer & Glocker (1996), and Visser et al. (2002).

Theorem 3.6 (Differential contact kinematics). *Given two rigid bodies with coordinate frames Ψ_i and Ψ_j and relative twist $T_j^{i,i}$, with surfaces described by the invertible and continuously differentiable coordinate mappings f_i and f_j and the continuously differentiable Gauss mappings g_i and g_j. If the points p_i and p_j on the two surfaces that have smallest relative Euclidean distance are uniquely defined and move smoothly over the surfaces, then the time-evolution of their coordinates P_i^i and P_j^j is given by the equations*

$$\left(g_{i*} + H_j^i g_{j*} H_i^j (I + \Delta g_{i*})\right) \dot{P}_i^i = \tilde{T}_j^{i,i} g_i + H_j^i g_{j*} \left(\dot{\Delta} g_j - \tilde{T}_i^{j,j} P_j^j\right) \tag{3.13}$$

$$\left(g_{j*} + H_i^j g_{i*} H_j^i (I + \Delta g_{j*})\right) \dot{P}_j^j = \tilde{T}_i^{j,j} g_j + H_i^j g_{i*} \left(\dot{\Delta} g_i - \tilde{T}_j^{i,i} P_i^i\right) \tag{3.14}$$

where Δ is the distance between p_i and p_j given by

$$\Delta = \left\langle g_i, H_j^i P_j^j - P_i^i \right\rangle = \left\langle g_j, H_i^j P_i^i - P_j^j \right\rangle \tag{3.15}$$

Proof. We prove the theorem in two steps. First, we show that the line connecting the closest points is normal to both surfaces, as illustrated by Figure 3.6. Assume that the two points p_i and p_j have indeed the smallest relative distance (left figure). This means that if we look for points of body j in a sphere of increasing radius around p_i (middle figure), that p_j will be the first point of body j that is encountered. Furthermore, since by assumption p_j is the *only* point at this distance from p_i, the surface of body j is locally tangent to the sphere at p_j. By definition of a normal vector, this means that the normal vector g_j to the surface at p_j is also normal to the sphere, and by definition of a sphere, the line along this normal passes through the center of the sphere, *i.e.* the point p_i. In the same way it follows (right figure) that the line along the normal at p_i passes through p_j.

The result can be written mathematically, for example, as

$$\begin{cases} P_i + \Delta g_i = H_j^i P_j \\ \qquad\; g_i = -H_j^i g_j \end{cases} \tag{3.16}$$

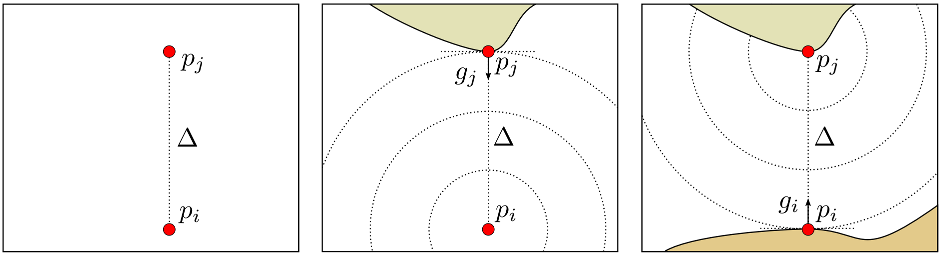


Figure 3.6: Geometric proof that the closest points on two surfaces are connected by a line collinear with the normals to the surfaces.

with $\Delta \in \mathbb{R}$ is as in (3.15). The first equation states that when starting from p_i, moving a distance Δ along the normal vector g_i results in arriving at p_j. The second equation states that the normal vectors g_i and g_j are equal but opposite (when expressed in the same coordinate frame).

The distance Δ becomes negative if the bodies penetrate each other, and hence the zero-crossing of Δ can be used to detect contact. We can compute the time-derivative of Δ as

$$\begin{aligned}\dot{\Delta} &= \left\langle \dot{g}_i, H^i_j P^j_j - P^i_i \right\rangle + \left\langle g_i, \dot{H}^i_j P^j_j + H^i_j \dot{P}^j_j - \dot{P}^i_i \right\rangle \\ &= 0 + \left\langle g_i, \dot{H}^i_j P^j_j \right\rangle + 0 + 0 = \left\langle g_i, \tilde{T}^{i,i}_j H^i_j P^j_j \right\rangle \end{aligned} \tag{3.17}$$

where we used the fact that the normal vector g_i is perpendicular to the velocities $\dot{P}^i_i$ and $H^i_j \dot{P}^j_j$ of the contact points over the surface, and that $\dot{g}_i$ is perpendicular to g_i (and hence to $H^i_j P^j_j + P^i_i$) since it is a unit vector.

As a second step, we compute the time-derivatives of (3.16) as follows.

$$\begin{gathered}\begin{cases} \dot{P}^i_i + \dot{\Delta} g_i + \Delta \dot{g}_i = \dot{H}^i_j P^j_j + H^i_j \dot{P}^j_j \\ \dot{g}_i = -\dot{H}^i_j g_j - H^i_j \dot{g}_j \end{cases} \\ \begin{cases} \dot{P}^i_i + \dot{\Delta} g_i + \Delta g_{i*} \dot{P}^i_i = \dot{H}^i_j P^j_j + H^i_j \dot{P}^j_j \\ g_{i*} \dot{P}^i_i = -\dot{H}^i_j g_j - H^i_j g_{j*} \dot{P}^j_j \end{cases} \\ \begin{cases} \dot{P}^j_j = H^j_i \left(\dot{P}^i_i + \dot{\Delta} g_i + \Delta g_{i*} \dot{P}^i_i - \dot{H}^i_j P^j_j \right) \\ g_{i*} \dot{P}^i_i = \tilde{T}^{i,i}_j g_i - H^i_j g_{j*} \dot{P}^j_j \end{cases}\end{gathered} \tag{3.18}$$

where we used the definition of a tangent mapping to write $\dot{g}_i = g_{i*} P^i_i$ as well as $\dot{g}_j = g_{j*} P^j_j$. If we substitute the first line of (3.18) into the second, we obtain

$$g_{i*} \dot{P}^i_i = \tilde{T}^{i,i}_j g_i - H^i_j g_{j*} H^j_i \left(\dot{P}^i_i + \dot{\Delta} g_i + \Delta g_{i*} \dot{P}^i_i - \dot{H}^i_j P^j_j \right) \tag{3.19}$$

which can be rewritten as (3.13). If we interchange the labeling of the bodies i and j and repeat the analysis, we obtain (3.14), thus completing the proof of the theorem. □

Theorem 3.6 gives a differential equation which the contact points satisfy if the contact points move smoothly over the surfaces (which, for example, is the case if both surfaces are convex). Still, two problems remain. First, the initial contact points are not given by the theorem and need to be determined, for example, by a numerical search at the start of the simulation or by choosing initial configurations for which the contact points are easily computed by hand. Second, as with all open-loop integrators, the solutions for the contact point coordinates are susceptible to numerical drift, which cause the integrated coordinates of the contact points to be less accurate over time. To avoid these problems, it may be possible to have a combined method of using global collision detection methods (such as used in computer graphics) for initialization and occasional re-calibration to avoid numerical drift, and the fast differential approach of Theorem 3.6 for the time-steps in between.

3.2 Compliant contact

This section describes an approach to model compliant contact between rigid bodies, or between rigid mechanisms in general. In compliant contact modeling, the interaction forces between objects in contact are assumed to be due to the compression and deformation of the surfaces of the objects. As the objects come in contact and push further into each other, the elasticity and damping of their surfaces result in increasing forces (the contact forces) that oppose the deforming motion and cause the objects to 'bounce back'. The amount of bounce depends on the relative elasticity and damping.

As discussed by Chatterjee & Ruina (1998), compliant contact of rigid bodies is an oxymoron: the bodies are assumed to be rigid, yet the contact forces are caused by the deformations of their surfaces. Fortunately, if the deformations are small relative to the size of the bodies, the dynamics can still be accurately described by the equations for rigid bodies in addition to an appropriate model of the compliant contact forces.

The most accurate way to model the forces resulting from surface deformation is by a (rather complex) finite element analysis of the objects in contact. Fortunately, if the deformation of the objects is small compared to the size of the objects, it is often accurate enough to model the contact forces as a lumped combination of an elastic element and a dissipative element, both possibly nonlinear. Finally, when the surface deformation is negligible (since the objects are relatively stiff) and when the 'bounce back' is also negligible, it can be more appropriate to consider the contact as rigid; this situation is discussed in Section 3.3.

As an example, consider modeling the contact of the feet of a walking robot

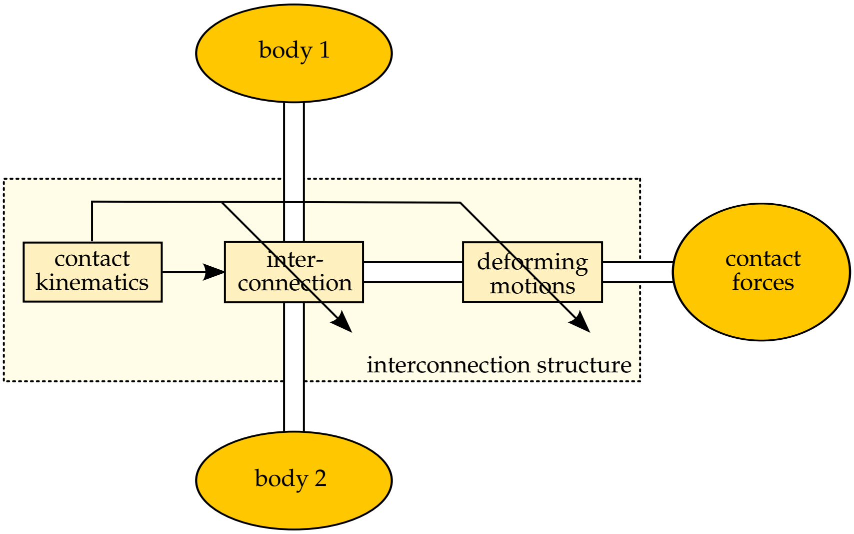
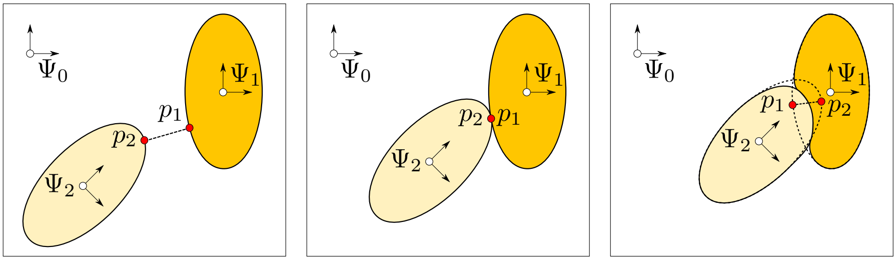


Figure 3.7: Modeling compliant contact as a port interconnection of contact forces between two bodies in contact, modulated by the contact kinematics.

with the ground. If the feet are padded with rubber and the robot is walking on a thick carpet, we can assume the deformations to be small but not negligible, and hence a compliant contact model can be used, such as discussed in this section. When harder materials are used, such as aluminum feet on a stone floor, the contact may better be modeled as rigid.

From a modeling point of view, the main advantage of compliant contact modeling over rigid contact modeling is in the implementation. Compliant contact can be implemented as a port-interconnection of the contact dynamics with the rigid bodies that are in contact. Rigid contact, on the other hand, needs to act directly on the state variables of the bodies in contact, due to the impulsive forces acting on impact (see Section 3.3), and it is not possible to implement using a port-interconnection, unless impulsive forces or time-varying causality is allowed.

Figure 3.7 shows the port-based implementation structure of compliant contact between two possibly contacting rigid bodies. It consists of two parts: the dark-colored part marked 'contact forces', in which the deformation energy is stored and/or dissipated and the corresponding contact forces are computed, and the light-colored parts which determine the interconnection structure, *i.e.* how the relative motion of the bodies is turned into deformation motions, and how and where the contact forces act on the bodies in contact. These two parts are discussed in the following sections.



(a) free motion, $\Delta > 0$ (b) impact, $\Delta = 0$ (c) contact, $\Delta < 0$

Figure 3.8: Evolution of the kinematic contact points during free motion, on impact, and during contact.

3.2.1 Interconnection structure of compliant contact

The first purpose of the interconnection structure in Figure 3.7 is to determine whether or not contact forces should act on the bodies, *i.e.* whether the bodies are in contact or not. We can use one of the contact kinematics approaches from Section 3.1 to determine this: using the relative configuration and twist of the bodies, together with the description of their surfaces, we monitor the contact points moving over the body, as well as their relative distance Δ. As Figure 3.8 shows, positive delta indicates that the bodies are not in contact, Δ equal to zero means that the bodies just come into contact (impact occurs), and negative Δ means that the bodies have penetrated (the contact points p_1 and p_2 are then the points with maximum penetration equal to $|\Delta|$).

Using these variables, a possible implementation of the interconnection block could be

$$\begin{bmatrix} W^{0,1} \\ W^{0,2} \\ T_2^{0,1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 1-\text{sign}(\Delta) \\ 0 & 0 & -1+\text{sign}(\Delta) \\ -1+\text{sign}(\Delta) & 1-\text{sign}(\Delta) & 0 \end{bmatrix} \begin{bmatrix} T_1^{0,0} \\ T_2^{0,0} \\ W^{0,c} \end{bmatrix} \tag{3.20}$$

where $(T_1^{0,0}, W^{0,1})$ is the power port on body 1, $(T_2^{0,0}, W^{0,2})$ the power port on body 2, and $(T_2^{0,1}, W^{0,c})$ the power port connected to the 'deforming motions' block. This is a power-continuous interconnection block, as shown by the skew-symmetry of the matrix in (3.20).

The second purpose of the interconnection structure is to transform the relative twist $T_2^{0,1}$ of the bodies into a twist that models the surface deformation occurring in the contact area. This is a necessary transformation, since not every relative motion of the bodies causes the surfaces to deform: penetrating, sliding and twisting motions do increase the deformation (until slip occurs), but pure rolling motions do not.

We can decompose $T_2^{0,1}$ into rolling, twisting, sliding, and penetrating using the tangent plane to the contacting surfaces; rolling and sliding are rotation around and translation along axes in this plane, while twisting and penetration are rotation around and translation along the axis perpendicular to this plane. More details on the group-theoretical aspects of this decomposition can be found in Stramigioli & Duindam (2004).

Unfortunately, the tangent plane to the contacting surfaces is not clearly defined, since we do not consider the precise deformed surfaces. As an approximation, we take the tangent contact plane to be a plane perpendicular to the line connecting the two kinematic contact points p_1 and p_2. Furthermore, the location of the plane is between p_1 and p_2 at a position that depends on the relative stiffnesses of the bodies, such that it is closest to the stiffest body (this is based on the idea that the stiffest body deforms the least and that hence the real contacting surface is closest to that body). We can compute the following homogeneous matrix H_1^c, which describes the transformation from body frame Ψ_1 to a frame Ψ_c on the tangent contact plane with z-axis perpendicular to the contact plane:

$$H_1^c(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{-k_2}{k_1+k_2}|\Delta(t)| \\ 0 & 0 & 0 & 1 \end{bmatrix} H_1^{1c}(t) \tag{3.21}$$

where k_1 and k_2 are the (normal) stiffnesses of bodies 1 and 2, H_1^{1c} is the transformation between Ψ_1 and Ψ_{1c}, and Ψ_{1c} is a frame at the contact point p_1 with z-axis equal to the surface normal vector g_1. The origin of Ψ_c is hence located between p_1 and p_2 at a distance $\frac{k_2}{k_1+k_2}|\Delta|$ from p_1, or equivalently, at a distance $\frac{k_1}{k_1+k_2}|\Delta|$ from p_2. So indeed, it is located closest to the stiffest body. For example, for $k_2 \to 0$ and $k_1 > 0$, Ψ_c is located at p_1.

With Ψ_c defined, we can express the relative twist in the contact plane as

$$T_2^{c,1} = \begin{bmatrix} \omega_2^{c,1} \\ v_2^{c,1} \end{bmatrix} = \mathrm{Ad}_{H_1^c}\,\mathrm{Ad}_{H_0^1}\,T_2^{0,1} \tag{3.22}$$

and we can use the coordinate components of this twist to describe the various motions as follows. The x and y components of $\omega_2^{c,1}$ are rolling, its z component is twisting, the x and y components of $v_2^{c,1}$ are sliding, and its z component is penetration.

From $T_2^{c,1}$, we can compute a deformation twist T_d^c (a twist describing only the deforming motions) in several ways, depending on the model assumptions we make. Probably the simplest one is to assume deformation for twisting, sliding, and penetrating motions, no deformation for rolling motions, and no slipping. In that case, the deformation twist can be written as

$$T_d^c = \begin{bmatrix} \omega_x^d \\ \omega_y^d \\ \omega_z^d \\ v_x^d \\ v_y^d \\ v_z^d \end{bmatrix} = X_d T_2^{c,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} (\omega_2^{c,1})_x \\ (\omega_2^{c,1})_y \\ (\omega_2^{c,1})_z \\ (v_2^{c,1})_x \\ (v_2^{c,1})_y \\ (v_2^{c,1})_z \end{bmatrix} \tag{3.23}$$

i.e. just by ignoring the rolling parts. We use the notation T_d^c to indicate that the coordinates of the twist are in frame Ψ_c and that the projected twist describes the deformation, not the relative velocity of bodies 1 and 2. This twist then becomes the input for the 'contact forces' block, where it can be integrated to a homogeneous matrix H_d that describes the deformation. We show in Section 3.2.2 that the resulting matrix H_d has the following form

$$H_d(t) = \begin{bmatrix} \cos(\theta_d) & -\sin(\theta_d) & 0 & x_d \\ \sin(\theta_d) & \cos(\theta_d) & 0 & y_d \\ 0 & 0 & 1 & z_d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.24}$$

The structure of the matrix shows that indeed only twisting (nonzero θ_d), sliding (nonzero x_d, y_d), and penetrating (nonzero z_d) deformations are possible.

We can extend the deformation twist to include slipping behavior, where for larger deformations in x_d, y_d, and θ_d, the deformation saturates to a certain maximum value. Furthermore, we can argue that rolling motion reduces the amount of sliding deformation, as the new contacting surface of the objects (the ones they roll onto) has not been slided and hence the effective surface sliding deformation reduces. An example of how these effects can be implemented is given by the matrix below, which should replace the diagonal matrix X_d in (3.23):

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & f_{\text{slip}}(\theta_d) & 0 & 0 & 0 \\ -\beta x_d \operatorname{sign}(\omega_2^{c,1})_x & 0 & 0 & f_{\text{slip}}(x_d) & 0 & 0 \\ 0 & -\beta y_d \operatorname{sign}(\omega_2^{c,1})_y & 0 & 0 & f_{\text{slip}}(y_d) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.25}$$

in which $\beta > 0$ is a parameter, and

$$f_{\text{slip}}(a) := 1 - (1 + \operatorname{sign}(v_a a)) \frac{|a|}{2a_{\text{max}}} \tag{3.26}$$

is an example of a function that describes the slipping behavior in the direction a, with maximum deformation $a_{\text{max}} > 0$ and v_a the element of $T_2^{c,1}$ in the direction

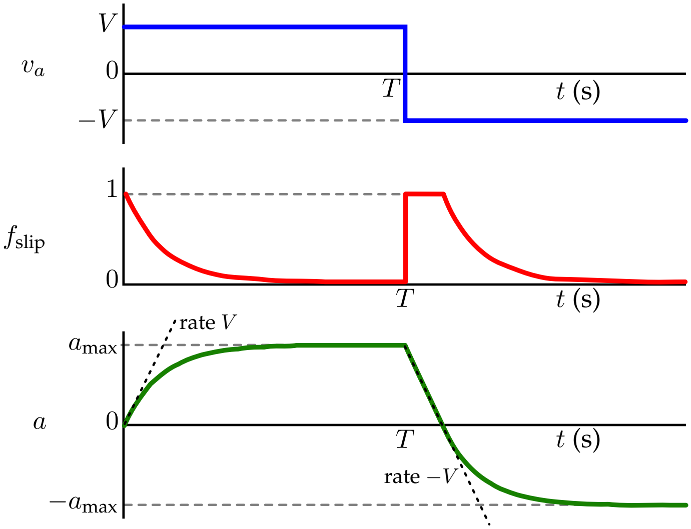


Figure 3.9: Effect of the slipping function (3.26) on the deformation. The figure shows f_{slip} and deformation a when the velocity v_a is a block signal.

of a. The effect of the slipping function is illustrated in Figure 3.9: it shows how, under constant velocity v_a, the deformation a increases as the integral of v_a for deformations much smaller than a_{max}, and then saturates as a approaches a_{max}.

Example 3.7. An example of the effect of this extended choice for the deformation twist is given in Figure 3.10. The figure shows three consecutive motions of a ball in contact with a plane (first penetrating, then sliding, then rolling), and its qualitative effect on the deformation matrix H_d, indicated by the frames in the lower figures (exaggerated). The dotted frame denotes the initial and undeformed configuration, and the frames in darkers shades of grey show the configuration H_d relative to the initial frame at increasing time steps along the motion.

The first motion shows that after remaining initially undeformed (when the ball is still in free motion), H_d shows increased vertical deformation in the direction of penetration. The second motion shows additional horizontal deformation, with saturation (slipping) as the deformation approaches its maximum. The third motion shows that rolling reduces the horizontal deformation until the surfaces are almost undeformed (the exact amount of reduction depends on the parameter β; larger β result in faster reduction). △

The previous discussion has shown possible choices for transformations and computations of the deformation twist, and of course different choices are possible. However, it is important to ensure that whatever transformation X_d for the

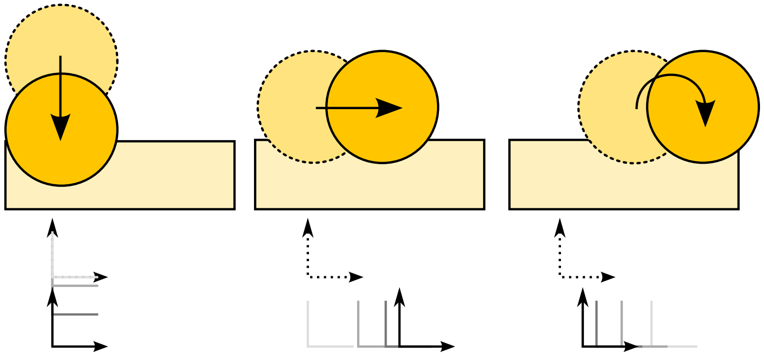


Figure 3.10: Penetrating, sliding, and rolling motions of two objects, and their effect on the deformation matrix H_d.

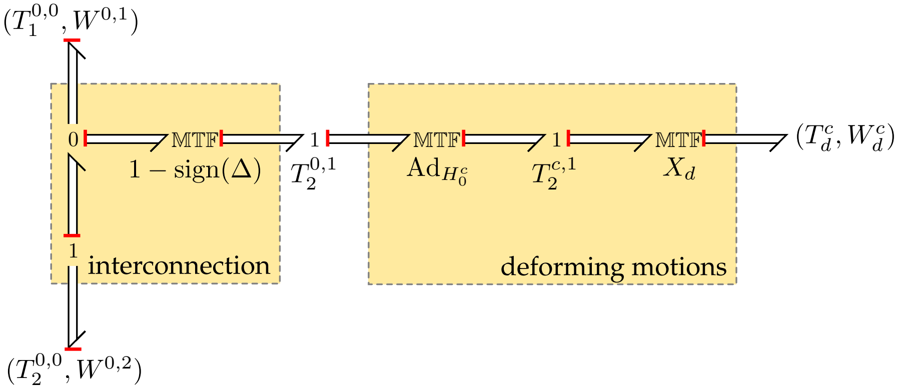


Figure 3.11: Bond graph of the interconnection structure of Figure 3.7 (the contact kinematics block has been left out for conciseness).

twists is chosen, the collocated wrenches should be transformed in the same but transposed way, *i.e.* as

$$\begin{bmatrix} T_d^c \\ W^{c,c} \end{bmatrix} = \begin{bmatrix} 0 & X_d \\ -X_d^T & 0 \end{bmatrix} \begin{bmatrix} W_d^c \\ T_2^{c,1} \end{bmatrix} \tag{3.27}$$

such that the transformation itself is power-continuous. After all, the energy properties of the compliant contact are to be modeled only by a passive spring and damper (to be described next), and hence its interconnection to the two bodies should not by allowed to generate or dissipate extra energy.

To emphasize the power-continuity of the parts discussed in this section, Figure 3.11 shows a bond graph representation of the interconnection structure in Figure 3.7. The bond graph is a port-interconnection of only power-continuous elements (junctions and modulated transformers) and hence it is itself power-continuous.

3.2.2 Compliant contact forces

The power-continuous parts described in the previous section transform the individual twists of the contacting bodies to a twist T_d^c that describes (approximately) the deforming motion of the contacting surfaces. The contact forces, described by the wrench W_d^c, are transformed back (in the dual way) to wrenches acting on the contacting bodies.

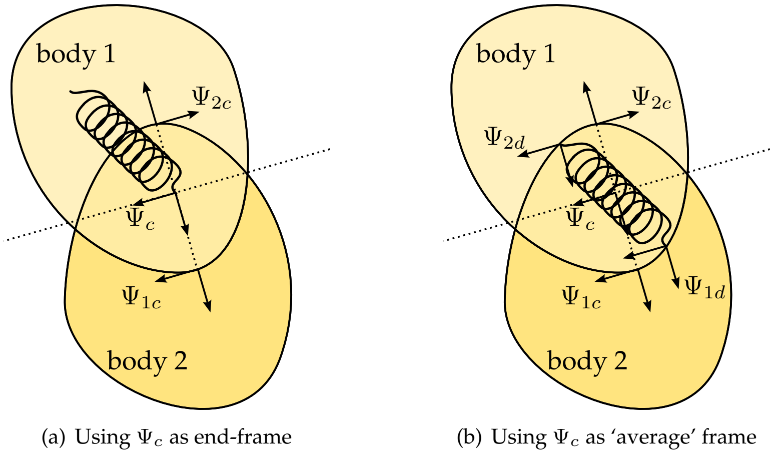
This section describes an approach to compute the wrench W_d^c from the deformation twist T_d^c. We decompose this wrench in an elastic part (*i.e.* the energy that is reversibly stored in the deformation of the bodies' surfaces) and a dissipative part (*i.e.* the energy that is irreversibly transformed into heat due to the friction in the deformation process).

We first describe the elastic part, which, like all elastic elements, is fully defined by a state and an energy function V. The state should describe the deformation of contacting surfaces, and V the energy that is stored for this deformation.

Since we have the deformation twist T_d^c available to describe the deformation velocities, the state should somehow be the integral of this twist. As we have seen in Definition 2.6, a twist can be integrated to a homogeneous matrix as $\dot{H}_j^i = H_k^i \tilde{T}_j^{k,i} H_j^k$, *i.e.* not by direct integration, but by pre- and post-multiplication with certain homogeneous matrices, depending on which coordinate frame the twist is expressed in. The problem with T_d^c is that it is not directly obvious what twist (*i.e.* between what frames i and j) it represents. Due to the projection operation, it does not just represent the relative motion of bodies 1 and 2 anymore, but only part of it (namely the motion without rolling). The question is hence: where do we need to attach the spring?

Figure 3.12 illustrates the effect of choosing different coordinate frames and integrating the twist in these frames. First, Figure 3.12(a) shows what happens if we choose to integrate T_d^c as $\dot{H}_d = \tilde{T}_d^c H_d$, so with Ψ_c being the 'world-frame'. The resulting H_d will then be the transformation matrix from H_d to the end of the spring, *i.e.* the spring will be placed at the location indicated in the figure. Similarly, if we choose Ψ_c to be the 'body-frame' and integrate $\dot{H}_d = H_d \tilde{T}_d^c$, the spring would be placed on the other side of Ψ_c.

Both situations are clearly undesired: the spring should be placed somehow around Ψ_c such that it is approximately attached between the two surfaces as shown in Figure 3.12(b). This can be accomplished by choosing Ψ_c as an 'average' location of the spring, in the following way: we split the deformation twist T_d^c in two parts, each describing (approximately) the deformation of one of the two



(a) Using Ψ_c as end-frame

(b) Using Ψ_c as 'average' frame

Figure 3.12: Effect on the spring location for different choices of integration of T_d.

surfaces relative to Ψ_c. Mathematically, we choose to decompose the twist as

$$T_d^c = T_{2d}^{c,1d} = T_c^{c,1d} + T_{2d}^{c,c} = \underbrace{\frac{k_2}{k_1+k_2}T_d^c}_{\text{deformation of surface 1}} + \underbrace{\frac{k_1}{k_1+k_2}T_d^c}_{\text{deformation of surface 2}} \tag{3.28}$$

where k_1 and k_2 are the relative stiffnesses of the two surfaces, as used when defining Ψ_c in (3.21). The two twists can then be integrated to obtain the homogeneous matrices describing the deformation of each surface, and these can be combined to give the total deformation matrix H_d, *i.e.*

$$\frac{d}{dt}H_c^{1d}(t) = H_c^{1d}(t)\tilde{T}_c^{c,1d}(t) = \frac{k_2}{k_1+k_2}H_c^{1d}(t)\tilde{T}_d^c(t) \tag{3.29}$$

$$\frac{d}{dt}H_{2d}^c(t) = \tilde{T}_{2d}^{c,c}(t)H_{2d}^c(t) = \frac{k_1}{k_1+k_2}\tilde{T}_d^c(t)H_{2d}^c(t) \tag{3.30}$$

$$H_d(t) = H_{2d}^{1d}(t) = H_c^{1d}(t)H_{2d}^c(t) \tag{3.31}$$

with initial conditions $H_c^{1d}(t_0) = H_{2d}^c(t_0) = I$ with t_0 the moment of impact (when there is no deformation and the frames $\Psi_{1d} = \Psi_{2d} = \Psi_c$ coincide). Since ω_x^d and ω_y^d (the rolling components of T_d^c in (3.23)) are zero, both H_c^{1d} and H_{2d}^c contain only a rotation component around z (and the z-axes of the two frames are parallel). Therefore, also their product contains only rotation around z, and is hence indeed of the form (3.24).

The previous discussion shows how to obtain a homogeneous matrix H_d that describes the deformation of the surfaces. Using this state[1], we can now define a spring by any suitable positive energy function $V(H_d)$ with minimum $V(I) = 0$ (no deformation). Such elastic elements acting on $SE(3)$ are called spatial springs, as introduced by Lončarić (1985) and further studied by Fasse (1997) and Zefran & Kumar (2002). Visser & Stramigioli (2006) describe a general approach to defining spatial springs, and how this approach can be used to obtain the various types of springs that are described in literature.

Spatial springs are used, for example, in spatial impedance control (Stramigioli 2001), 3D telemanipulation (Stramigioli et al. 2002, Secchi et al. 2001), and in the modeling of spatial visco-elastic couplings (Fasse 2000). Stramigioli & Duindam (2001) show extensions to port-controlled spatial springs with variable length and stiffness, and Stramigioli & Duindam (2004) discuss anisotropic spatial springs.

Defining a general spatial spring in a coordinate-independent way is not easy, since it should somehow be a suitable function of the homogeneous matrix H_d. Fortunately, in our case the homogeneous matrix H_d has the special structure (3.24), and it is hence usually easier to express the energy as a function of the variables $(\theta_d, x_d, y_d, z_d)$. The time-evolution of these coordinates can be derived from (3.29)-(3.31). A simple example of an energy function of the coordinates $(\theta_d, x_d, y_d, z_d)$ is the quadratic function

$$V(x_d, y_d, z_d, \theta_d) = \frac{1}{2} K_t \left(x_d^2 + y_d^2 + z_d^2 \right) + \frac{1}{2} K_r \theta_d^2 \tag{3.32}$$

with linear stiffness $K_t > 0$ and rotational stiffness $K_r > 0$. This means that the spring can be implemented as a port-Hamiltonian system with differential equation for the deformation parameters

$$\frac{d}{dt}\begin{bmatrix}\theta_d \\ x_d \\ y_d \\ z_d\end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -x_2\sin(\theta_1) - y_2\cos(\theta_1) & \cos(\theta_1) & -\sin(\theta_1) & 0 \\ 0 & 0 & x_2\cos(\theta_1) - y_2\sin(\theta_1) & \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} T_d^c \tag{3.33}$$

and output equation for the contact wrench

$$W_d^c = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & -x_2\sin(\theta_1) - y_2\cos(\theta_1) & x_2\cos(\theta_1) - y_2\sin(\theta_1) & 0 \\ 0 & \cos(\theta_1) & \sin(\theta_1) & 0 \\ 0 & -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} K_r\theta_d \\ K_t x_d \\ K_t y_d \\ K_t z_d \end{bmatrix} \tag{3.34}$$

where $(\theta_1, x_1, y_1, z_1)$ and $(\theta_2, x_2, y_2, z_2)$ are coordinates for H_c^{1d} and H_{2d}^c as in (3.24). This expression follows from expanding the time-derivative of (3.31), and can be directly coupled to the interconnection structure of Figure 3.11.

[1] We choose here to define one spatial spring as a function of the total deformation H_d. Another, more detailed, option would be to choose two springs with states H_c^{1d} and H_{2d}^c.

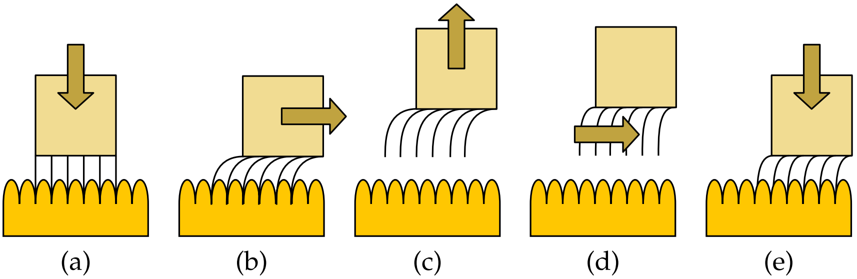


Figure 3.13: Illustration of remaining deformation on second impact after release of the first contact: after the first impact (a), the surface of the square deforms horizontally (b). Then, on release of the first contact (c), the deformation reduces (d), but some deformation still remains when the second impact occurs (e).

In addition to elastic contact forces, dissipative contact forces can be modeled by any relation between the twist T_d^c and wrench W_d^c such that the corresponding power is always non-negative, *i.e.* the resistive element can only dissipate energy, not generate or store it. A very simple dissipative relation would be $W_d^c = rT_d^c$ with $r > 0$ a parameter. The combination of this linear dissipative relation with the quadratic elastic energy function (3.32) (and both equal to zero if the bodies are not in contact) is known as the Kelvin-Voigt model (Flügge 1975).

Instead of the simple linear models for elastic and dissipative contact forces, existing contact models in the literature can be formulated as functions of H_d or the coordinates $(\theta_d, x_d, y_d, z_d)$ and then applied to the same interconnection structure. Examples of commonly used contact models are the Hertzian contact model (Johnson 1985), the micro-slip model (Johnson 1985), the Hunt-Crossley model (Hunt & Crossley 1985), and the LuGre model (Canudas de Wit et al. 1995).

Remark. Note that by construction of the interconnection structure, the first two elements of T_d^c are zero and hence the rolling motions are dissipation-free. This can be changed, if desired, by connecting a dissipative element to a different part of the interconnection structure, *e.g.* at the point marked $T_2^{c,1}$ in Figure 3.11. △

Remark. Apart from the dissipation of energy due to friction, energy is also 'lost' due to breaking of contact. When contact is made, the initial deformation is set to zero ($H_d = I$) with corresponding energy $V(I) = 0$. Then, during contact, the deformation and the corresponding stored energy V increase, partially due to penetration (described by z_d) and partially due to sliding and twisting (described by x_d, y_d, θ_d). When the contact is released, the penetration z_d necessarily crosses zero, but the sliding and twisting deformations may still be nonzero and (depending on the energy function) may have stored energy! When the next impact occurs, however, the deformations are reset to zero, and this stored energy is lost.

The physical motivation for this implementation is that for most practical surfaces and contact situations, any remaining surface deformations after detachment will indeed be quickly restored to the undeformed configuration, and the corresponding energy will be transformed into heat. However, for some materials or for fast contact switching, this remaining deformation energy may still be (partially) stored, in which case the deformation reset on impact should be adapted. Figure 3.13 shows an example how this situation could occur. After impact (a) and deformation (b) during a first contact phase, the deformed surface is released (c). During the subsequent brief no-contact phase, the surface starts to settle towards its undeformed shape (d), but it has not completely regained its undeformed shape, when already a second impact occurs (e). △

3.3 Rigid contact

The compliant contact model of the previous section allows for a wide range of surface properties and other situations. However, for the purpose of modeling walking robots, it has two main problems.

The first problem is that as the surfaces in contact become stiffer, the dynamics in the collision phase become faster. This leads to a dynamic model described by stiff differential equations; a model with relatively fast dynamics (the collision dynamics) as well as relatively slow dynamics (the other motions of the system). Simulation of such system requires special integration methods in order to ensure the accuracy of the results as well as acceptable simulation speeds, and simulation of the collision phase will be relatively slow.

The second problem relates to the analysis of the models. A walking cycle of a robot consists of a single-support phase, in which one foot is more or less fixed to the ground and one foot is above the ground, and a double-support phase, in which both feet are on the ground and support is transferred from one foot to the other. Of these two phases, the single support phase is generally the longest, and for some walking configurations, the double-support phase completely disappears as the surface stiffness increases. However, the compliance in the contact model prevents the stance foot from being exactly fixed to the ground, and the double-support phase from being exactly instantaneous, thus making the analysis of the overall cycle overly cumbersome.

For these two reasons, we present in this section a simpler contact model. This model is suitable for contact situations in which the stiffness and damping are large enough to permit approximating them by an instantaneous dissipation of energy on impact. The advantage of this model is that the analysis becomes simpler, especially for certain types of walking robots where the double support phase becomes a simple momentum reset between consecutive steps.

In general, it is not so easy to replace compliant contact by instantaneous impulsive contact. Aspects that are unimportant in compliant modeling (such as the exact order in which points of the bodies come in contact with each other) suddenly change the outcome of the simulation significantly in the case of impulsive contact modeling (Acary & Brogliato 2003). Furthermore, the presence of finite friction and slip requires various extra modeling assumptions to ensure a single deterministic solution of the dynamic equations (Glocker 2001, Glocker 2004).

To avoid such problems, we choose to restrict ourselves to the following class of contact situations: we assume instantaneous and fully plastic contact (zero restitution), no simultaneous collisions, and no sideways slipping. These assumptions are a reasonable approximation for most walking robots.

3.3.1 Model setup

As the starting point for the rigid contact model, we take a mechanical system of the form (2.92), repeated here.

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix}q\\p\end{bmatrix} &= \begin{bmatrix}0 & I\\-I & 0\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q}\\\frac{\partial H}{\partial p}\end{bmatrix} + \begin{bmatrix}0 & 0\\A(q) & B(q)\end{bmatrix}\begin{bmatrix}\lambda\\u\end{bmatrix}\\ \begin{bmatrix}0\\y\end{bmatrix} &= \begin{bmatrix}0 & A^T(q)\\0 & B^T(q)\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial q}\\\frac{\partial H}{\partial p}\end{bmatrix}\end{aligned} \tag{3.35}$$

In the model, the matrix $A(q)$ describes the directions of the (possibly) constrained velocities, or equivalently, the directions of the contact forces, and λ are the magnitudes of the contact forces. As in Section 2.4, we do not take the general form of Theorem 2.17 for the mechanism, in order for the equations to remain concise. Still, the results of this section extend directly to general mechanisms as well.

The columns of the matrix $A(q)$ describe the direction of the contact wrenches and contain information from the contact kinematics, in order to know when and at what position the contact forces act. The number of constraint forces should be determined by the modeler, for example whether only tangential forces should be used or also torsional constraints around the vertical axis.

To avoid confusion over plus and minus signs, we always take the positive direction on the vertical axis to be the direction of motion of the object away from the contacting surfaces, *e.g.* when modeling a robot walking on level ground, the positive vertical contact direction is upward, not downward.

Example 3.8. Consider as an example Figure 3.14, showing a mechanism with three degrees of freedom (q^1, q^2, q^3). We can express the coordinates of the tip of the end effector in Ψ_0 as

$$\begin{bmatrix}x_t\\y_t\\z_t\end{bmatrix} = \begin{bmatrix}x_{\text{base}}\\y_{\text{base}}\\l_1\end{bmatrix} + l_2\begin{bmatrix}\sin(q^1)\cos(q^2)\\-\cos(q^1)\cos(q^2)\\-\sin(q^2)\end{bmatrix} + l_3\begin{bmatrix}-\sin(q^1)\sin(q^2+q^3)\\\cos(q^1)\sin(q^2+q^3)\\-\cos(q^2+q^3)\end{bmatrix} \tag{3.36}$$

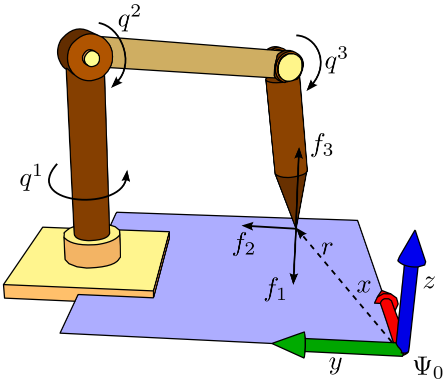


Figure 3.14: Example of a mechanism with the tip in contact with the ground plane.

where l_i is the length of link i, $(x_{\text{base}}, y_{\text{base}}, 0)$ are the fixed coordinates of the base plate, and the zero configuration is with link 2 horizontal and pointing towards the xz-plane and link 3 vertically downward.

If we consider the possible contact of the tip of the end effector, we can define for example the indicated force directions as

$$f_1 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad f_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad f_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.37}$$

which act on the tip at the point $r = \begin{bmatrix} x_t & y_t & 0 \end{bmatrix}^T$. Depending on the type of contact surface to be modeled, we can consider only the vertical force f_3 (while allowing possible sliding along the surface), or also the tangential forces f_1 and f_2 (in which case no sliding is allowed).

If we consider all three contact forces, and J_3 denotes the Jacobian of link 3 relative to Ψ_0, then we can compute the joint torques equivalent with the contact forces as follows, using the expressions of Definition 2.13 and Lemma 2.14.

$$A(q)\lambda = J_3^T(q) \begin{bmatrix} r(q) \wedge f_1 & r(q) \wedge f_2 & r(q) \wedge f_3 \\ f_1 & f_2 & f_3 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} \tag{3.38}$$

where λ_i indicates the magnitude of the contact force along f_i. The transpose of the matrix $A(q)$ can be used to compute the velocities $A^T(q)\frac{\partial H}{\partial p}$ collocated with the contact forces, *i.e.* the linear velocities of the tip of the end effector in the directions of f_i. If the tip is in contact, the contact forces λ should be such that these velocities are zero. If the tip is not in contact, λ (or A) can be taken zero, and the velocity constraints should be ignored. △

Due to the structure and assumptions of this contact model, the contact forces will contain an impulsive component on impact (to satisfy the constraints instantaneously) as well as a finite component during the contact phase. These two aspects are discussed in the next two sections. After that, the conditions on contact release are discussed, as well as extensions to multiple contact situations.

3.3.2 Momentum reset on impact

As a first step of a rigid contact model, we describe what happens on impact, *i.e.* when impulsive constraint forces λ set the velocities $A^T \frac{\partial H}{\partial p}$ of the contact point to zero.

We define t_i as the time of impact, and t_- and t_+ as the time instants just before and just after impact, *i.e.* mathematically as $t_- := t_i - \epsilon$ and $t_+ := t_i + \epsilon$ with $\epsilon \downarrow 0$. Using this notation, we can indicate the velocity constraint of (3.35) as

$$0 = A^T(q) \frac{\partial H}{\partial p}\bigg|_{t=t_+} = A^T(q) M^{-1}(q) p(t_+) \tag{3.39}$$

where we used t_+ (the time just after impact), since the momentum is discontinuous at impact $t = t_i$ and hence not well-defined.

We can integrate the dynamic equations (3.35) over the impact phase, *i.e.* from t_- to t_+, which results in the following.

$$\int_{t_-}^{t_+} \dot{p} = \int_{t_-}^{t_+} \left(-\frac{\partial H}{\partial q} + A(q)\lambda + B(q)u \right) dt \tag{3.40}$$

$$p(t_+) - p(t_-) = A(q) \int_{t_-}^{t_+} \lambda dt \tag{3.41}$$

where we assumed that the terms $\frac{\partial H}{\partial q}$ and $B(q)u$ have finitely large magnitude and hence zero integral between t_- and t_+. If we substitute the expression (3.41) for $p(t_+)$ into (3.39), we obtain an expression for λ as

$$0 = A^T(q) M^{-1}(q) \left(p(t_-) + A(q) \int_{t_-}^{t_+} \lambda dt \right) \tag{3.42}$$

$$\int_{t_-}^{t_+} \lambda dt = -\left(A^T(q) M^{-1}(q) A(q) \right)^{-1} A^T(q) M^{-1}(q) p(t_-) \tag{3.43}$$

The inverse of the matrix $A^T M^{-1} A$ exists if and only if the columns of A are linearly independent, *i.e.* if the constraint force magnitudes λ are uniquely determined. This aspect is discussed further in Section 3.3.5 when we discuss multiple contact points.

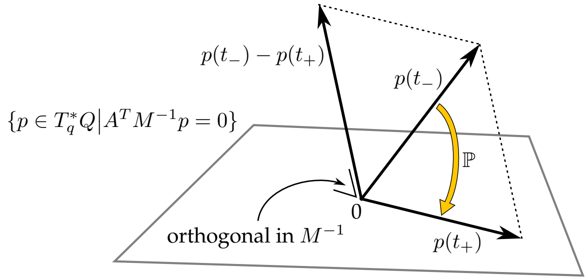


Figure 3.15: Illustration of the projection operator $\mathbb{P}$, projecting $p(t_-)$ onto the plane $A^T M^{-1} p = 0$.

Instead of explicitly computing and using λ, we can also substitute (3.43) into (3.41) to obtain an expression for the momentum after impact $p(t_+)$ as

$$p(t_+) = \mathbb{P}(q)p(t_-) = \left(I - A\left(A^T M^{-1} A\right)^{-1} A^T M^{-1}\right) p(t_-) \tag{3.44}$$

i.e. as a projection $\mathbb{P}$ along the columns of A^T of the momentum before impact p_- onto the plane defined by $A^T M^{-1} p = 0$, as illustrated in Figure 3.15.

Expression (3.44) is more suitable for simulations than (3.43), as it does not require dealing with the impulsive constraint forces λ directly. Unfortunately, it requires a state jump in the momentum and can hence not be implemented directly as a port interconnection.

Using (3.44), we can compute the energy loss on impact. Since the position variables are continuous, the potential energy does not change on impact, only the kinetic energy. We can compute ΔU_k as follows

$$\begin{aligned}
\Delta U_k &:= U_k(t_+) - U_k(t_-) \\
&= \frac{1}{2} p^T(t_+) M^{-1} p(t_+) - \frac{1}{2} p^T(t_-) M^{-1} p(t_-) \\
&= \frac{1}{2} p^T(t_-) \left(I - M^{-1} A (A^T M^{-1} A)^{-1} A^T\right) M^{-1} \\
&\qquad \left(I - A(A^T M^{-1} A)^{-1} A^T M^{-1}\right) p(t_-) - \frac{1}{2} p^T(t_-) M^{-1} p(t_-) \\
&= \frac{1}{2} p^T(t_-) M^{-1} p(t_-) - p^T(t_-) M^{-1} A (A^T M^{-1} A)^{-1} A^T M^{-1} p(t_-) \\
&\qquad + \frac{1}{2} p^T(t_-) M^{-1} A (A^T M^{-1} A)^{-1} A^T M^{-1} p(t_-) - \frac{1}{2} p^T(t_-) M^{-1} p(t_-) \\
&= -\frac{1}{2} p^T(t_-) M^{-1} A (A^T M^{-1} A)^{-1} A^T M^{-1} p(t_-)
\end{aligned} \tag{3.45}$$

This expression can be used, for example, to determine the efficiency of walking.

3.3.3 Constraint forces during contact

After impact, the velocity of the contact point is set to zero, such that the constraints are satisfied. During the contact phase that follows, the constraint magnitude λ should be such that these constraints *remain* satisfied until contact is broken (see Section 3.3.4). For finite external forces, the constraint forces themselves will also be finite.

In Section 2.4, we briefly discussed several methods to deal with the implicit set of the equations (3.35). Since the constraints are time-varying (contact can be on or off), they can be implemented in the easiest way using the second method discussed in Section 2.4, *i.e.* by computing the required constraint forces explicitly from the time derivative of the constraint equations. This results in an expression for λ given by (2.94), repeated here:

$$\left(A^T M^{-1} A\right) \lambda = A^T M^{-1} \frac{\partial H}{\partial q} - A^T M^{-1} B u - \frac{\partial (A^T M^{-1} p)}{\partial q} \frac{\partial H}{\partial p} \tag{3.46}$$

Simulations using reasonable integration step sizes have shown that the numerical drift on implementation of these equations leads to non-negligible simulation errors. Therefore, we include the extra term discussed in Section 2.4 in order to reduce these problems, leading to the following equation

$$\begin{aligned}\left(A^T M^{-1} A\right) \lambda = \\ - \beta A^T \frac{\partial H}{\partial p} + A^T M^{-1} \frac{\partial H}{\partial q} - A^T M^{-1} B u - \frac{\partial (A^T M^{-1} p)}{\partial q} \frac{\partial H}{\partial p}\end{aligned} \tag{3.47}$$

with $\beta > 0$ a damping factor, *i.e.* (3.46) with an extra term to drive numerical errors in the constraint velocity to zero. Simulations with this improved model show to be accurate enough for walking motions, *i.e.* when contacts do not last long enough to make drifting noticeable.

Remark. In both expressions for the contact forces λ, the partial derivative of $A(q)$ with respect to q appears. The matrix $A(q)$ contains information about the location of the contact point, and hence its partial derivative with respect to q^i describes how the contact point would move if the mechanism moved in the direction q^i.

In practice, this partial derivative is hard to express, and instead, it is often easier to use the equivalent formulation

$$\frac{\partial (A^T M^{-1} p)}{\partial q} \frac{\partial H}{\partial p} = \dot{A}^T M^{-1} p + A^T \frac{\partial (M^{-1} p)}{\partial q} \frac{\partial H}{\partial p} \tag{3.48}$$

The time-derivative of A can be expressed using the velocity of the contact point, which is available directly from the contact kinematics of Section 3.1. △

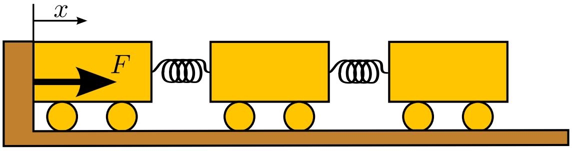


Figure 3.16: Example of a one-dimensional system, in which the left cart is constrained to have either positive x or positive F.

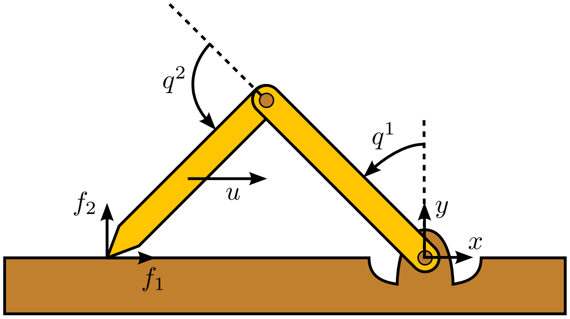


Figure 3.17: Example of a two-dimensional system in which contact release conditions should not be expressed in terms of the force directions.

3.3.4 Conditions for contact release

The contact forces λ in the previous section are computed such that the velocity of the contact point remains zero, irrespective of what these forces should be. Obviously, under certain conditions, the contact should be released, *i.e.* the contact forces set to zero, and the algebraic constraint equation discarded. The aim of this section is to determine these conditions for contact release.

For one-dimensional systems, such as the system shown in Figure 3.16, the conditions are often formulated in terms of the direction of the required constraint force. With the sign conventions of Figure 3.16, such a condition would be formulated as: contact is broken whenever the constraint force F (acting on the left cart) becomes negative. This condition for contact release is based on the intuitive idea that the wall can only push on the object, it cannot pull.

Based on this formulation for one-dimensional systems, it seems easy to extend this to multi-dimensional systems, namely just as a similar condition on the component of λ corresponding to the direction normal to the contacting surface (*e.g.* λ_3 in the example of Figure 3.14). This, however, leads to unphysical behavior, as proved by the following example.

Example 3.9. Figure 3.17 shows a two-link robot in contact with the ground. If we take (for simplicity) the links to have unit length and unit mass, we can find the mass matrix M and the Jacobian matrix J of the second link relative to the reference frame as

$$M(q) = \begin{bmatrix} \frac{3}{2} + \cos(q^2) & \frac{1}{4} + \frac{1}{2}\cos(q^2) \\ \frac{1}{4} + \frac{1}{2}\cos(q^2) & \frac{1}{4} \end{bmatrix} \tag{3.49}$$

$$J(q) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & \cos(q^1) & \sin(q^1) & 0 \end{bmatrix}^T \tag{3.50}$$

If we now use the two directions f_1 and f_2 indicated in the figure, we can write the constraint matrix $A(q)$ as

$$\begin{aligned} A &= J^T \begin{bmatrix} r \wedge f_1 & r \wedge f_2 \\ f_1 & f_2 \end{bmatrix} \\ &= \begin{bmatrix} -\cos(q^1) - \cos(q^1 + q^2) & -\sin(q^1) - \sin(q^1 + q^2) \\ -\cos(q^1 + q^2) & -\sin(q^1 + q^2) \end{bmatrix} \end{aligned} \tag{3.51}$$

with r the position vector to the tip of the second link. The constraints are satisfied for $A^T M^{-1} p = 0$ and the constraint forces λ act on $\dot{p}$ through the columns of A.

Consider now the static situation depicted in the figure, with $q^1 = \frac{\pi}{4}$, $q^2 = \frac{\pi}{2}$, and $p = 0$. We apply a constant force $u = 1N$ to the center of the second link in the positive x direction, with the equivalent joint torque τ given as

$$\tau = \left(J^T(q) \begin{bmatrix} r_u \wedge u \\ u \end{bmatrix} \right) \Bigg|_{q^1 = \frac{\pi}{4}, q^2 = \frac{\pi}{2}} = \begin{bmatrix} -\frac{1}{4}\sqrt{2} \\ \frac{1}{4}\sqrt{2} \end{bmatrix} \tag{3.52}$$

with r_u the coordinates of the actuation point expressed in the reference frame. Suppose we now compute the required constraint forces λ to keep the second link from slipping and penetrating. If we ignore other forces such as gravity, we can compute λ from (3.46) as

$$\lambda = -\left(A^T M^{-1} A\right)^{-1} A^T M^{-1} \tau = \begin{bmatrix} -\frac{3}{4} \\ -\frac{1}{4} \end{bmatrix} \tag{3.53}$$

where all other terms drop out since $p = 0$. The resulting λ has a negative second component, *i.e.* a negative component in the f_2 direction. If we used the contact-release condition based on the direction of forces, we would now conclude that for this λ, the ground would be 'pulling' the object down, and hence contact should be broken.

However, if we then compute the acceleration a of the tip of the second link (which is just the time derivative of the constraint equation), we find

$$a = \dot{A}^T M^{-1} p + A^T \dot{M}^{-1} p + A^T M^{-1} \dot{p} = 0 + 0 + A^T M^{-1} \tau = \begin{bmatrix} \frac{7}{5} \\ -\frac{3}{5} \end{bmatrix} \tag{3.54}$$

that is, the second link is accelerating to the right as well as into the ground! △

The example shows that judging contact release on the basis of the direction of the constraint force gives incorrect results, namely possible penetration of the ground. Although the use of the force direction *appears* to be physically motivated (the ground cannot pull on the object), it clearly leads to unphysical behavior.

Instead, we propose the following condition. Given a system of the form (3.35) with active unilateral contact constraints represented by the equation $A^T M^{-1} p = 0$. As shown in the example, the acceleration a of the contact point in the unconstrained case is equal to the time-derivative of the constraint equation:

$$a = \frac{d}{dt}\left(A^T M^{-1} p\right)\big|_{\lambda=0} = \frac{d}{dt}(A^T M^{-1})p + A^T M^{-1}\left(-\frac{\partial H}{\partial q} + Bu\right) \tag{3.55}$$

Contact should be released (*i.e.* the λ set to zero and the constraint equations discarded) as soon as the normal component of a (the component in the direction normal to the contacting surfaces) becomes larger than zero, that is, as soon as the states (q, p) and the input u are such that the acceleration a of the unconstrained system is directed away from the contacting surfaces.

This condition ensures (by definition) that no penetration of the objects in contact occurs. Furthermore, it provides minimum sticky behavior, since contact is released as soon as possible (as soon as no penetration will occur), even though the contact force may still be directed downward.

Remark. The unconstrained acceleration a and the constraint force λ are related as $(A^T M^{-1} A)\lambda = -a$, which can by seen by comparing (3.46) and (3.55). This shows why no similar problem occurs in one-dimensional systems (or systems with only one constraint force): for such systems, the matrix $A^T M^{-1} A$ reduces to a (positive) scalar, so $a > 0$ is exactly equivalent to $\lambda < 0$. △

3.3.5 Extension to two contact points

To be able to model a bipedal walking robot with rigid contacts with the ground, it is necessary to consider two possible contact locations (the two feet of the robot with the ground). During the motion of the robot, the two contacts can become both active (for the robot having two feet on the ground), both inactive (for the robot to be free in the air), and one active and one inactive (for the robot to have either the left or the right foot on the ground).

For compliant contact as discussed in Section 3.2, it is straightforward to extend the results to multiple contacts: just add extra spring-damper models acting between the appropriate parts of the robot and the ground. For rigid contact models, however, it is not so simple, since both the impact forces and the contact release conditions become more involved. Hence, in this section, we discuss only the extension to two contact points, which should be sufficient to model bipedal walking. A more general solution for larger numbers of contact points is the subject of ongoing research, *e.g.* in the field of complementarity problems (Pfeiffer & Glocker 1996).

We start again from a system of the form (3.35), but now with two possible sets of contact forces (λ_1 and λ_2) and contact directions (A_1 and A_2), corresponding to the contact forces acting at the first and the second contact point. Both can contain several components, both in the direction normal to the contacting surface as well as in tangential and rotational directions. We can write the implicit differential equations for these systems as follows

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ A_1(q) \end{bmatrix}\lambda_1 + \begin{bmatrix} 0 \\ A_2(q) \end{bmatrix}\lambda_2 + \begin{bmatrix} 0 \\ B(q) \end{bmatrix}u \\ \begin{bmatrix} 0 \\ 0 \\ y \end{bmatrix} &= \begin{bmatrix} 0 & A_1^T(q) \\ 0 & A_2^T(q) \\ 0 & B^T(q) \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} \end{aligned} \tag{3.56}$$

where the constraint equations and forces should be ignored if the corresponding contact is not active.

Generally speaking, if only one of the constraints is active, the computations of the constraint forces can be done as in the previous sections, with the matrix A replaced by either A_1 or A_2, depending on which contact is active. If both contacts are active, the matrix A should be taken as $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ such that both constraints are satisfied.

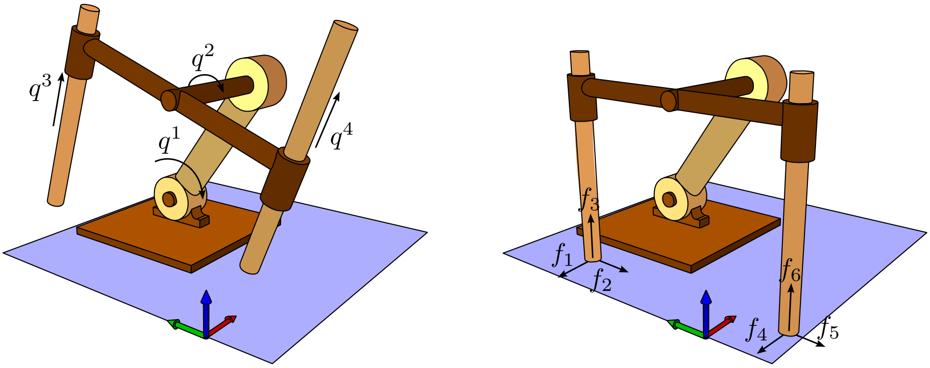
However, the combination of the contact states and especially the changing from contact to no contact (or partial contact) should be handled with care. Specifically, we discuss the following aspects of two-point contact modeling that make it different from the single-point contact models discussed in the previous sections: linear dependency of contact forces, two impulsive forces on single impact, and finally the contact-release conditions for two-point contact.

Linear dependency of contact forces

If both contacts are active, the directions of the constraint forces are described by the matrix $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$. This matrix is used in the expression $(A^T M^{-1} A)^{-1}$ in the computation of both the impulsive forces (3.43) on impact and the finite forces (3.46) during contact.

The matrix $A^T M^{-1} A$ is invertible if the columns of A are linearly independent, *i.e.* if A has full column rank. Physically, this means that the contact forces are statically determined: there is only one solution λ that satisfies the constraints. Dually and equivalently, it means that the velocity constraints (described by the rows of A^T) are independent.

The matrix A can be rank deficient when the two contacts are active at the same time, or when extra constraint directions are defined which are constrained



(a) Choice of coordinates q^1 — q^4.

(b) Constraint force directions f_1 — f_6.

Figure 3.18: Example of a system for which the six constraint forces f_1 through f_6 (when active in the double contact phase) can be represented equivalently by only three forces.

by the mechanism kinematics already. An example of both situations is shown in Figure 3.18: the forces f_1 and f_4 are both canceled out by the kinematics (mathematically speaking, the corresponding wrench is in the kernel of J^T), since the mechanism is essentially planar, and furthermore f_2 and f_5 are linearly dependent, since when the horizontal motion of the left pole is fixed (by f_2) as well as the vertical motions of both poles (by f_3 and f_6), then also the horizontal motion of the right pole is fixed.

If the matrix A is rank deficient, the inverse of $A^T M^{-1} A$ does not exist, and hence a different solution technique needs to be applied to solve λ from (3.43) and (3.46). Since we are not interested in a specific solution for λ, but just *any* solution that satisfies the constraints, we can use a pseudo-inverse of $A^T M^{-1} A$ instead of a regular inverse, such that a solution λ is obtained that is minimal in some metric. In fact, if we choose this metric to represent the relative compliances of the different contact points, then the resulting components of λ (the contact forces in the different directions) will be relatively large for contacts that are relatively stiff, which is a result that makes physical sense.

Two impulsive forces on single impact

A second aspect of two-point contact that needs attention, is when one point is in contact and the second point has an impact. The effect of this impact can cause the contact velocity of the first constraint to be violated, namely if

$$0 > A_1^T M^{-1} p(t_+) = A_1^T M^{-1} \left(I - A_2 (A_2^T M^{-1} A_2)^{-1} A_2^T M^{-1} \right) p(t_-) \qquad (3.57)$$

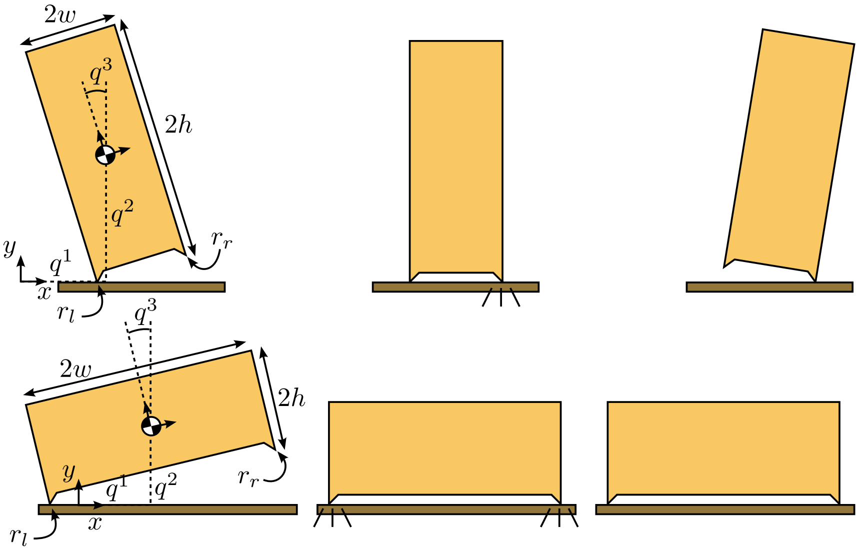


Figure 3.19: Example of a tall object that continues to move after an impact, and a wide object that comes to a standstill.

In this case, the impact on the second contact point causes a simultaneous impact on the first contact point, such that the momentum after impact should be computed as (3.44) with $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ (or possibly a full column rank equivalent).

If the constraint of the first joint is not violated after a single impact, *i.e.* (3.57) does not hold, then this means that, on impact, not only the second contact becomes active, but instantaneously also the first contact becomes inactive. In walking robots, this corresponds to the often-made assumption of instantaneous transfer of support from one foot to the next.

Example 3.10. To illustrate these ideas, consider Figure 3.19, showing a closet of a certain width $2w$ and height $2h$ as it rotates around one (left) contact point and then contacts the ground with a second (right) contact point. We know from practical experience that depending on the height and width of the closet, it will either continue to rock over the second contact point (for tall closets) or come to a complete stop (for wide closets). The aim of this example is to show for what parameters w, h these different behaviors occur.

First, let us write down the dynamic equations of the system. We use three coordinates as indicated in the figure to be able to handle all possible configurations (both free-floating, only left contact, only right contact, both contacts). If the closet has a certain mass m and inertia J around its center, as well as gravity acting in the negative y-direction, we can write the total energy of the system as

$$H(q,p) = mgq^2 + \frac{1}{2}\begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix}\begin{bmatrix} m^{-1} & 0 & 0 \\ 0 & m^{-1} & 0 \\ 0 & 0 & J^{-1} \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \tag{3.58}$$

with $p := M\dot{q}$ as usual. Possible contacts are assumed to occur at the left and right foot at the points r_l and r_r given by

$$r_l = \begin{bmatrix} q^1 - w\cos(q^3) + h\sin(q^3) \\ q^2 - w\sin(q^3) - h\cos(q^3) \\ 0 \end{bmatrix} \qquad r_r = \begin{bmatrix} q^1 + w\cos(q^3) + h\sin(q^3) \\ q^2 + w\sin(q^3) - h\cos(q^3) \\ 0 \end{bmatrix} \tag{3.59}$$

We assume both vertical constraint forces and horizontal constraint forces at the contact points, such that no sideways sliding of the closet is allowed. We can compute the constraint matrices A_l and A_r as in the example of Section 3.3.1, using the expression for the Jacobian of the planar joint given in the example of Figure 2.4(c).

$$\begin{aligned} A_l &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & q^2 & -q^1 & 0 \end{bmatrix} \\ &\qquad \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -q^2 + w\sin(q^3) + h\cos(q^3) & q^1 - w\cos(q^3) + h\sin(q^3) \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ w\sin(q^3) + h\cos(q^3) & -w\cos(q^3) + h\sin(q^3) \end{bmatrix} \end{aligned} \tag{3.60}$$

$$A_r = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -w\sin(q^3) + h\cos(q^3) & w\cos(q^3) + h\sin(q^3) \end{bmatrix} \tag{3.61}$$

The total dynamics can now be written as

$$\begin{aligned} \dot{q} &= M^{-1}p \\ \dot{p} &= \frac{\partial H}{\partial q} + A_l(q)\lambda_l + A_r(q)\lambda_r \\ 0 &= A_l^T(q)M^{-1}p \\ 0 &= A_r^T(q)M^{-1}p \end{aligned} \tag{3.62}$$

where λ_l and λ_r are the left and right constraint forces, we assumed no additional input ports, and where the constraints may be active or not, depending on the contact situation.

Let us now consider the situation that the left foot is in contact with the ground and the object is rotating clockwise around the contact point. This means that the momentum of the system can be written as

$$p = \begin{bmatrix} mw\sin(q^3) + mh\cos(q^3) \\ -mw\cos(q^3) + mh\sin(q^3) \\ -J \end{bmatrix} \omega \tag{3.63}$$

for some $\omega > 0$ (the angular velocity), since the two other degrees of freedom are forced to zero by the constraint forces λ_l.

Consider now the situation that the right foot hits the ground (*i.e.* $q^3 \to 0$). This results in an impact on the right foot, and if necessary also on the left foot. We can compute the momentum projection $\mathbb{P}(q)$ in (3.44), corresponding to the right impact, as follows.

$$\begin{aligned} \mathbb{P} &= \left(I - A_r(A_r^T M^{-1} A_r)^{-1} A_r^T M^{-1}\right)\big|_{q^3=0} \\ &= \frac{1}{J + m(h^2+w^2)} \begin{bmatrix} mh^2 & mhw & -mh \\ mhw & mw^2 & -mw \\ -Jh & -Jw & J \end{bmatrix} \end{aligned} \tag{3.64}$$

If only this impact occurs (and not one on the left foot), then the velocity of the left foot can be calculated as

$$\begin{bmatrix} \dot{x}_l \\ \dot{y}_l \end{bmatrix} = A_l M^{-1} \mathbb{P} p = \begin{bmatrix} 0 \\ \frac{2w(J+m(h^2-w^2))}{J+m(h^2+w^2)} \end{bmatrix} \omega \tag{3.65}$$

with p as in (3.63). The fact that $\dot{x}_l = 0$ is as expected, since the horizontal constraint on the right foot is the same as the horizontal constraint on the left foot.

Whether the resulting velocity is compatible with the constraints, *i.e.* whether $\dot{y}_l > 0$, depends on the parameter values. For a valid velocity (and hence rocking behavior of the closet), we should have

$$\begin{aligned} \dot{y}_l = \frac{2w(J+m(h^2-w^2))}{J+m(h^2+w^2)}\omega &> 0 \\ J + m(h^2 - w^2) &> 0 \\ h^2 - w^2 &> -\frac{J}{m} \end{aligned} \tag{3.66}$$

where we used the fact that all parameters as well as ω are strictly positive. This shows that rocking behavior will occur if the height h of the closet is large enough.

For smaller h, the single right impact results in violation of the left constraints, and hence an extra left impact is needed, which sets the total momentum to zero and brings the closet instantaneously to a standstill.

Practical limit values for h can be found by specifying the mass distribution of the closet. For example, if all mass is concentrated in the center ($J \to 0$), rocking will occur for $h > w$. Similarly, if the mass is homogeneously distributed over the closet ($J = \frac{1}{12}m(4h^2 + 4w^2)$), rocking will occur for $h > \frac{1}{2}\sqrt{2}w$. △

Contact-release conditions for two-point contact

The final aspect of two-point contact that needs attention is the condition for contact release. In case only one of the contact points is active, the single-contact conditions can be directly applied (with A replaced by A_1 or A_2, obviously). However, when both contacts are active, the situation is a little more involved.

The decision process for two-point contacts is depicted in Figure 3.20. It is based on the same idea as in Section 3.3.4, namely that contact should be released as soon as the acceleration of the contact point for $\lambda = 0$ is away from the contacting surfaces. This means that first the accelerations are computed for the case $\lambda_1 = \lambda_2 = 0$. If these are both positive (away from the contacting surfaces), then both contacts are released.

If at least one of the accelerations is non-positive, then one of the contacts is activated, and the corresponding λ is computed, as well as the acceleration of the remaining free point. If only $a_1 \leq 0$ ($a_2 \leq 0$), it is most likely that λ_1 (λ_2) needs to be activated, and hence we compute this situation first. If both accelerations are non-positive, then either λ_1 or λ_2 can be chosen as active first (in the figure, λ_1 is chosen).

If the acceleration of the remaining free contact point is positive, then that contact is released and the first one remains active. If the remaining acceleration is non-positive, then the computations are repeated with the free and active contacts reversed.

If both choices result in non-positive acceleration, then one contact force is not enough but both contacts are needed, so the required λ_1 and λ_2 are computed and applied to the mechanism, and both contacts remain active.

Example 3.11. To illustrate the procedure of Figure 3.20, we discuss the systems shown in Figure 3.21. We only describe the results intuitively, without actually computing the accelerations and constraint forces. In each example, the gravitational force is equal to $f_z = 10N$ downward, and where applicable, the external force equals $u = 11N$. Furthermore, the '1' and '2' below each supporting triangle indicates the contact point number.

(a) The first system shows a bar resting on two triangular support blocks on each side of the bar. To show that the bar remains resting on the blocks,

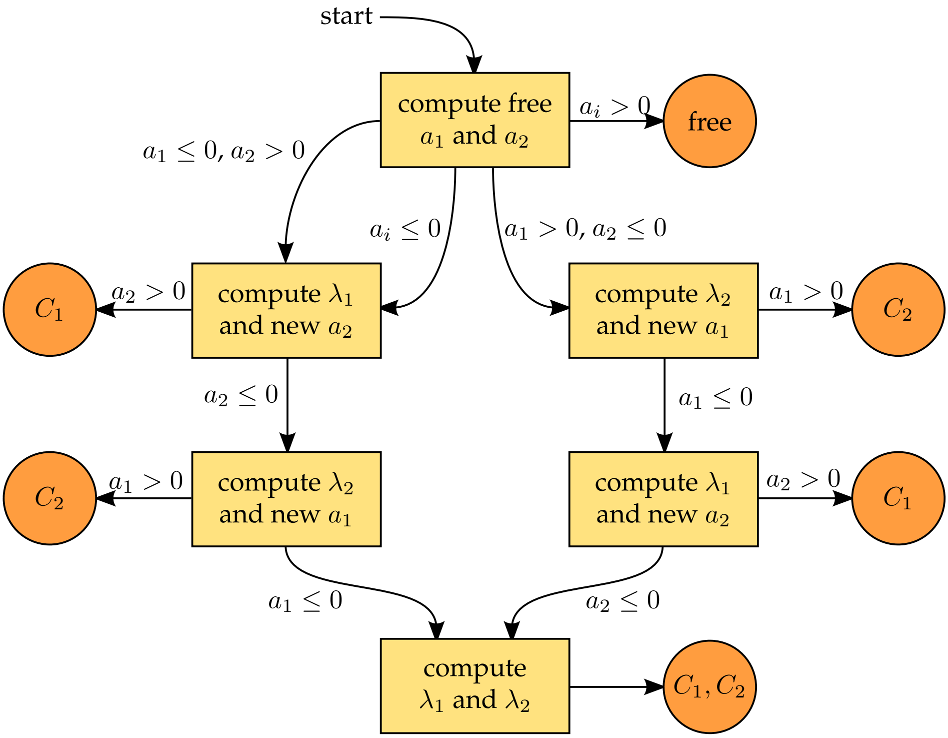

start
compute free a_1 and a_2
$a_i > 0$
free
$a_1 \leq 0, a_2 > 0$
$a_i \leq 0$
$a_1 > 0, a_2 \leq 0$
C_1
$a_2 > 0$
compute λ_1 and new a_2
compute λ_2 and new a_1
$a_1 > 0$
C_2
$a_2 \leq 0$
$a_1 \leq 0$
C_2
$a_1 > 0$
compute λ_2 and new a_1
compute λ_1 and new a_2
$a_2 > 0$
C_1
$a_1 \leq 0$
$a_2 \leq 0$
compute λ_1 and λ_2
C_1, C_2

Figure 3.20: Diagram to determine possible (partial) contact release for two-point contact. The symbols C_1 (C_2) indicate that contact 1 (contact 2) is active, whereas 'free' denotes that neither is active. The symbol a_1 (a_2) denotes the vertical acceleration of contact point 1 (contact point 2), possibly with $\lambda_2 \neq 0$ ($\lambda_1 \neq 0$).

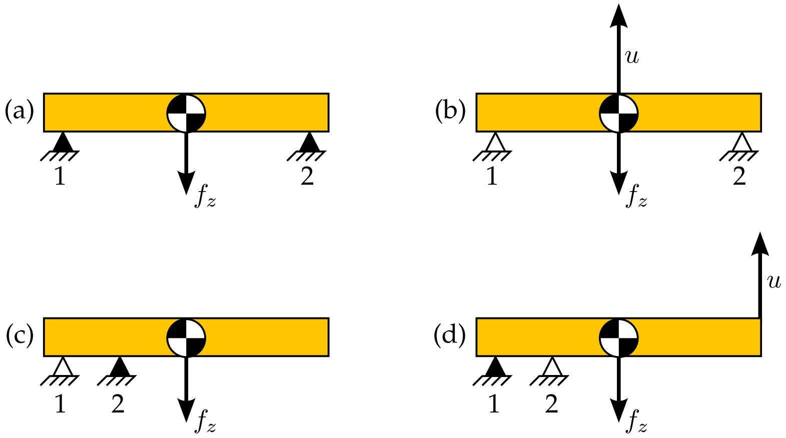


Figure 3.21: Examples of two-point contact situation with various contact releases. Black triangles mark active contact points.

we follow the diagram of Figure 3.20. If we determine the accelerations of the contact points in the absence of constraint forces, clearly they are both negative (the block would fall straight down through the triangles). Following the left side of the diagram, if contact 1 is assumed active and we compute the acceleration of point 2, we see that still it is downward. Similarly, if we assume contact 2 to be active, the acceleration of point 1 is downward. Hence, the only solution is that both constraints are active, as is clear intuitively.

(b) The second system is equivalent to the first, except that an extra external force $u = 11N$ acts on the system. If we compute the accelerations of the contact points in the absence of the constraint forces, we see that both are positive (since the net force on the center of mass is upward), and hence both contacts are released, as is clear intuitively.

(c) The third system shows a bar resting on two support blocks, both located on the left side of the center of mass. Following the diagram, we see that the contact point accelerations without contact forces are both negative, hence we follow the left path in the diagram. If we assume the first contact to be active and compute the acceleration of the second contact point, it turns out to be downward (since the block starts rotating clockwise around the first support block). If we then assume only the second contact to be active, we can see that the acceleration of the first contact point is positive, and hence this contact situation is chosen; only the second contact is active, and the block starts rotating clockwise around this point.

(d) Finally, the fourth system shows the same system as in example (c), except for the external force u. In this case, both free accelerations are again negative, but if we assume the first contact to be active, the acceleration of the second body is positive (since the body starts rotating counter-clockwise) and hence this contact situation is chosen.

△

Chapter 4

Modeling and Analysis of Walking Robots

Chapters 2 and 3 have developed the building blocks that are necessary to construct mathematical models of the dynamics of walking robots. This chapter discusses how to use these building blocks to obtain models for simulation and analysis, and the next chapter describes the use of models for controller design. These three goals (simulation, analysis, and control) generally require different mathematical models, with a different balance between accuracy and complexity. Models for analysis and controller design should be simple enough to show only the most prominent features, while simulation models should be detailed enough to represent the behavior in practice as accurately as possible.

For this reason, we construct walking robot models at several levels of detail. We start from a highly detailed level (suitable for simulation) and consider walking robots as freely floating (possibly planar) rigid mechanisms that can come in contact with a fixed ground with zero, one, or two feet. In the models in this chapter, we assume the ground to be a perfect plane, possibly tilted, and we consider only point feet. Generalizations to different ground and foot shapes are possible. Based on these detailed simulation models, we then derive simplified models for analysis under extra simplifying and restricting assumptions, for example by looking only at symmetric periodic walking cycles and assuming an instantaneous double-support phase. In the next chapter, further simplifications are made to obtain models suitable for controller design.

In order to study walking behavior, we have to find walking gaits of a mechanism: periodic motions of the links of the mechanism that, together with interactions with the ground, produce a net overall displacement along the ground. Section 4.1 describes a technique to find efficient walking gaits using numerical optimization. The following sections then describe the modeling and analysis of three examples of walking robots: a simple planar straight-legged walking robot (used as an illustration of the modeling and analysis approach), a more complex kneed planar walking robot (based on a real experimental robot), and finally a three-dimensional walking robot with a trunk.

As the complexity of the robots increases, the size of the corresponding equations increases as well, to the point that manual analysis of the equations is impossible, or at least not very insightful. For this reason, many of the model equations are not shown in the text of this thesis, but instead are available electronically (Duindam 2006) as equations in Mathematica (Wolfram Research 2005) and Matlab (The Mathworks 2005), and simulation models in 20-Sim (Control Lab Products 2005).

4.1 Gait search as an optimization problem

The simplified analysis models that are developed in the next three sections all have a similar structure: a set of differential equations that describe the continuous dynamics of the time from the start of a step to the end of a step, plus a projection and relabeling operation that link the positions and velocities at the end of a step to the positions and velocities at the beginning of the next step. The model of Section 4.3 contains an additional impact at some point during the step, but otherwise it has the same structure.

Figure 4.1 illustrates this structure for a planar bipedal walking robot with knees. The top figure shows a certain periodic walking cycle for the robot, consisting of two steps: one with the left foot and one with the right foot. This cycle is repeated in time (restarting from posture A after posture H), and is also symmetric with respect to the left and right leg. If we use certain coordinates $q(t)$ to describe a motion of the robot with the left foot on the ground, then we can use a mapping $\mathbb{S}(q(t))$ to describe the symmetric motion with the right foot on the ground (details of this mapping follow in the next sections). The total motion cycle A-E-A can then also be represented by two motions A-E, once with and once without the mapping $\mathbb{S}(q)$. This is illustrated in the bottom figure, showing a continuous curve in coordinate space that describes the motion from posture A (with coordinates $q(A)$) to posture E (with coordinates $q(E)$), as well as the motion from E (with coordinates $\mathbb{S}(q(E))$) to A (with coordinates $\mathbb{S}(q(A))$). The jump from $q(E)$ to $\mathbb{S}(q(E))$ and from $\mathbb{S}(q(A))$ to $q(A)$ is a combination of the momentum projection on impact (causing a change in the velocity) and the relabeling of the coordinates (affecting both position and velocity).

An important aspect of the research on passively walking robots is the search for *passive gaits*, that is, the search for natural, unactuated, periodic motions of a mechanism (also called passive limit cycles) that result in some net overall displacement. Depending on the configuration of the robot, its mass distribution, and the presence and direction of gravity, such limit cycles may exist or not, and they may be attractive (stable) or not (unstable).

Since only unactuated (passive) motions are considered, the dynamics of the

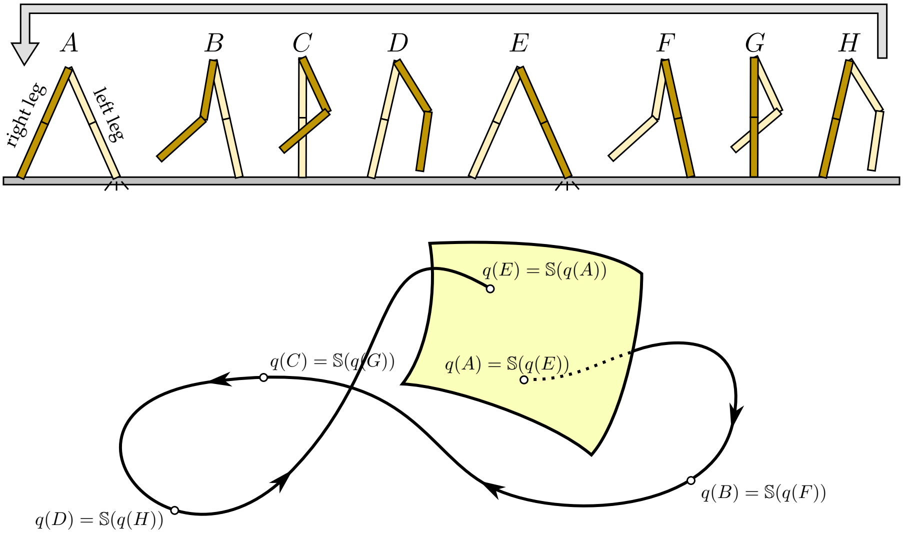

A
B
C
D
E
F
G
H
right leg
left leg
$q(E) = \mathbb{S}(q(A))$
$q(C) = \mathbb{S}(q(G))$
$q(A) = \mathbb{S}(q(E))$
$q(B) = \mathbb{S}(q(F))$
$q(D) = \mathbb{S}(q(H))$

Figure 4.1: Illustration of the idea of a Poincaré map for walking motions: a periodic symmetric walking cycle (top figure) can be represented by a closed curve in coordinate space (bottom figure) where left/right symmetry is expressed by the mapping $\mathbb{S}(q)$.

system are autonomous, and hence the initial conditions completely determine the motion. The deterministic function that maps the initial conditions at the beginning of one step to the initial conditions at the beginning of the next step (so the combination of the integrated continuous dynamics plus the impact momentum projection and the left/right relabeling operation) is called the *stride function* (McGeer 1990*b*) or *return map* (Koditschek & Bühler 1991), and it serves as a *Poincaré map* (Wiggins 2003) of the system. The problem of finding a passive walking gait can then be formulated as finding a fixed point of the Poincaré map, *i.e.* the initial conditions that map to themselves. Since the Poincaré map involves integration of nonlinear differential equations, it is usually not available symbolically, and searching for fixed points is hence a numerical problem. As an example, see Goswami et al. (1998) for a detailed study of the Poincaré map of the compass-gait walker.

The approach of searching for fixed points of the Poincaré map is very useful in passive dynamic walking, as it has few degrees of freedom (just the initial conditions of the system, which are often partially fixed due to the choice of the starting point of a cycle). However, it relies on the restriction to autonomous systems, *i.e.* systems with zero control input, or at least an input that is chosen a priori. When walking down a slope, zero-input limit cycles have been shown to exist for various walking mechanisms. However, for walking on level ground, purely passive walking cycles generally do not exist, since the kinetic energy lost during impact cannot be recovered from gravity. Some ideal walking mechanisms can be found that touch the ground with zero velocity (Gomes & Ruina 2005) and hence do not loose energy on impact, but these are exceptions.

For walking on general surfaces (whether level, downhill, or uphill), we do not search necessarily for purely passive motions, but only for natural efficient motions. If we consider the robot to be directly and fully actuated by ideal back-driveable motors (*i.e.* pure torque sources), then natural efficient motions of the robot are those for which little actuator torque is necessary. Purely passive motions are the limit case for which no torque is needed, but if such motions do not exist, then the best we can do is find the motions which require the least possible torque.

In order to properly define a measure for the amount of torque, we need to define a function on the space of torques that represents the cost of actuation in the various directions. For convenience, we consider functions of the form $\tau^T Q(t)\tau$, *i.e.* as a quadratic metric on the torque space, possibly time or configuration dependent. We then define the efficient gait search problem as the following minimization problem.

Definition 4.1 (Efficient Gait Search Problem). *The efficient gait search problem is the problem of finding the joint trajectories $q(t)$ that solve*

$$\min_{q(t)} \int_0^T \tau^T(t)Q(t)\tau(t)dt \tag{4.1}$$

with positive semi-definite cost metric $Q(t)$, subject to the constraints

$$\begin{aligned} \tau(t) &= \tau(q(t), \dot{q}(t), \ddot{q}(t)) \\ q_0 &= \mathbb{S}(q_T) \\ \dot{q}_0 &= \frac{\partial \mathbb{S}(q)}{\partial q}(q_T)M^{-1}(q_T)\mathbb{P}(q_T)M(q_T)\dot{q}_T \end{aligned}$$

where $\tau(q, \dot{q}, \ddot{q})$ are the dynamics of the system, $\mathbb{S}(q)$ describes the coordinate relabeling, and $\mathbb{P}(q)$ is the momentum projection due to impact.

The problem is hence to find the joint trajectories $q(t)$ for one step (between $t = 0$ and $t = T$) such that the total integrated torque τ is minimized in the metric $Q(t)$, and such that the initial conditions $(q_0, \dot{q}_0)$ are compatible with the projected and relabeled final conditions $(q_T, \dot{q}_T)$. The relation between $\dot{q}_0$ and $\dot{q}_T$ is a consecutive projection $\mathbb{P}$ of the momentum $M(q_T)\dot{q}_T$ due to impact, and a relabeling of the projected velocity by the differential of $\mathbb{S}(q)$. Additional constraints can be added, for example to force a certain desired walking speed, or to force $T > 0$.

Finding the optimal trajectory $q(t)$ that solves (4.1) is an infinite dimensional problem, and we use an approximation when searching for walking gaits. We parameterize the joint trajectories as polynomial functions of time, *i.e.* as

$$q^i(t) = \sum_{j=0}^{k} b_{ij} t^j = b_{i0} + b_{i1} t + b_{i2} t^2 + \ldots + b_{ik} t^k \tag{4.2}$$

for each coordinate $q^i(t)$ and for some constant integer $k > 0$. Different basis functions than polynomials can be taken, but polynomials have the advantage of being easy to differentiate (to obtain parameterizations of $\dot{q}(t)$ and $\ddot{q}(t)$), and they are suitable for approximating relatively slow-varying signals, which we expect $q(t)$ to be, since fast variations generally imply large torques. We also replace the integral over time by a finite Riemann sum. This leads to the following numerical approximation of Definition 4.1.

Definition 4.2 (Approximate Efficient Gait Search Problem). *The approximate efficient gait search problem is the problem of finding the parameters b_{ij} that solve*

$$\min_{b_{ij}} \frac{T}{N} \sum_{m=1}^{N} \tau^T(t_m) Q(t_m) \tau(t_m) \tag{4.3}$$

with positive semi-definite cost metric $Q(t)$ and positive integer N, and subject to the constraints

$$\begin{aligned} \tau(t_m) &= \tau(q(t_m), \dot{q}(t_m), \ddot{q}(t_m)) \\ q^i(t) &= \sum_{j=0}^{k} b_{ij} t^j \\ q_0 &= \mathbb{S}(q_T) \\ \dot{q}_0 &= \frac{\partial \mathbb{S}(q)}{\partial q}(q_T) M^{-1}(q_T) \mathbb{P}(q_T) M(q_T) \dot{q}_T \end{aligned}$$

where $\tau(q, \dot{q}, \ddot{q})$ are the dynamics of the system, $\mathbb{S}(q)$ describes the coordinate relabeling, and $\mathbb{P}(q)$ is the momentum projection due to impact.

Using this parameterization results in an optimization problem with $(k+1)n$ degrees of freedom (for a mechanism with n DoFs). This is generally a lot more than when searching for fixed points of the Poincaré map numerically, since then only $2n$ initial conditions need to be found. However, the problem formulated as (4.1) is much more general as its solution $q(t)$ is the *most natural* trajectory. This means it can also be used to find efficient (in the metric $Q(t)$), natural walking gaits on level floor, uphill, or other circumstances in which passive gaits do not

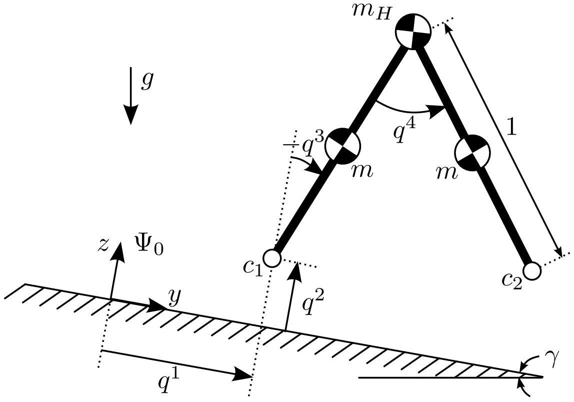


Figure 4.2: Setup and choice of coordinates and parameters for a planar straight-legged walking mechanism on an inclined flat floor.

exist. Furthermore, the optimization problem (4.1) can be extended, for example to include extra adjustable passive elements, which is exploited in Chapter 5.

The resulting gaits, obtained from either the Poincaré method or the optimization procedure, are periodic solutions to the dynamics equations. Whether these solution are stable or not, and how large the region of attraction of stable solutions is, is not determined by either gait search method. Especially for pure passive dynamic walking (in which case no control is available to stabilize the system), the region of attraction is very important: it determines whether a setup will actually be able to walk in practice, under the influence of disturbances and modeling errors.

4.2 A planar compass-gait walker

As a first example of the modeling and analysis techniques for walking robots, we consider the planar mechanism shown schematically in Figure 4.2. It consists of two legs with a point mass m at their centers, joined by a hip joint of mass m_H. The feet c_1 and c_2 can come in contact with the ground, which is tilted at an angle γ as shown in the figure. This robot is often called the compass-gait walker, because its mechanical structure is like that of a compass used for drawing circles.

The compass-gait walker has been studied by many different people in literature, mainly because it is the simplest possible mechanism that can still exhibit walking behavior. Its continuous dynamic equations are simple enough to be managed by hand, yet the total dynamics including impacts and contact switching possesses very interesting behavior involving stable passive limit cycles and bifurcations (see Goswami et al. (1998) for a presentation of these aspects). In this chapter, the mechanism serves as an example system to illustrate the basic modeling and analysis techniques that are used later on for more complex mechanisms.

4.2.1 Dynamic models of the compass-gait walker

Simulation model

We start by constructing a general dynamic model of the setup of Figure 4.2, suitable for all contact situations, for any number of feet on the floor (we assume that only the feet can contact the floor, no other parts of the mechanism). We use the coordinates shown in the figure to parameterize all possible configurations of the mechanism: two prismatic joints (q^1 and q^2) describe the position of the foot c_1, and two rotational joints (q^3 and q^4) describe the orientation of the legs. The coordinates in Ψ_0 of the foot c_2 can be expressed in terms of q as

$$\begin{bmatrix} c_{2x} \\ c_{2y} \\ c_{2z} \end{bmatrix} = \begin{bmatrix} 0 \\ q^1 - \sin(q^3) + \sin(q^3 + q^4) \\ q^2 + \cos(q^3) - \cos(q^3 + q^4) \end{bmatrix} \tag{4.4}$$

The geometric Jacobians J_3 and J_4 of the two legs relative to Ψ_0 can be constructed as in (2.40) to obtain

$$J_3(q) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & q^2 & 0 \\ 0 & 1 & -q^1 & 0 \end{bmatrix} \qquad J_4(q) = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & q^2 & q^2 + \cos(q^3) \\ 0 & 1 & -q^1 & -q^1 + \sin(q^3) \end{bmatrix} \tag{4.5}$$

Using these Jacobians, the mass matrix (2.59) of the mechanism can be constructed as the sum of three components due to the three point masses. The resulting mass matrix is of the form

$$M(q) = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & m_H + \frac{3}{2}m - m\cos(q^4) & \frac{1}{4}m(1 - 2\cos(q^4)) \\ * & * & \frac{1}{4}m(1 - 2\cos(q^4)) & \frac{1}{4}m \end{bmatrix} \tag{4.6}$$

where $*$ indicates components not represented here for conciseness. The kinetic co-energy of the system is $U_k(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q}$, and the potential energy $V(q)$ equals the sum of the three gravitational energies of the point masses, that is,

$$\begin{aligned} V(q) &= mg(-q^1 \sin(\gamma) + q^2 \cos(\gamma) + \frac{1}{2}\cos(q^3 - \gamma)) \\ &\quad + m_H g(-q^1 \sin(\gamma) + q^2 \cos(\gamma) + \cos(q^3 - \gamma)) \\ &\quad + mg(-q^1 \sin(\gamma) + q^2 \cos(\gamma) + \cos(q^3 - \gamma) - \frac{1}{2}\cos(q^3 + q^4 - \gamma)) \end{aligned} \tag{4.7}$$

The next aspect of the model is to include the possible contact forces by using the methods of Section 3.3 for rigid contact. The contact forces act on the feet c_1 and c_2, and are assumed to contain both vertical and horizontal components, in the sense that both the normal and tangential velocity components of c_1 and c_2 are set and kept to zero by the contact forces (if the corresponding contact is active). These velocity components can be expressed as a linear combination of the joint velocities $\dot{q}$, namely as

$$\begin{bmatrix} \dot{c}_{1y} \\ \dot{c}_{1z} \\ \dot{c}_{2y} \\ \dot{c}_{2z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & \cos(q^3+q^4)-\cos(q^3) & \cos(q^3+q^4) \\ 0 & 1 & \sin(q^3+q^4)-\sin(q^3) & \sin(q^3+q^4) \end{bmatrix} \begin{bmatrix} \dot{q}^1 \\ \dot{q}^2 \\ \dot{q}^3 \\ \dot{q}^4 \end{bmatrix} =: \begin{bmatrix} A_1^T(q) \\ A_2^T(q) \end{bmatrix} \dot{q} \quad (4.8)$$

where the last two rows are obtained by differentiation of (4.4). Depending on which foot is in contact with the ground, none, either, or both of the constraints $A_1^T\dot{q} = 0$ and $A_2^T\dot{q} = 0$ should be satisfied. The complete model can now be constructed in port-Hamiltonian form as (2.92), repeated here.

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ A(q) & B(q) \end{bmatrix} \begin{bmatrix} \lambda \\ \tau \end{bmatrix} \\ \begin{bmatrix} 0 \\ y \end{bmatrix} &= \begin{bmatrix} 0 & A^T(q) \\ 0 & B^T(q) \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} \end{aligned} \quad (4.9)$$

with $H(q,p) = \frac{1}{2}p^T M^{-1}(q)p + V(q)$, mass matrix (4.6), potential energy (4.7), momentum variables $p := M(q)\dot{q}$, input power port (τ, y), input matrix $B = I$, constraint matrix $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ as in (4.8), and constraint forces λ such that the active constraints are satisfied.

Analysis model

The model described in this way is suitable for simulation, since it can handle all contact situations. For analysis of a walking gait, however, it is still quite complex, and not yet in the form of Figure 4.1.

We therefore restrict our analysis to the situation of having exactly one foot on the ground, such that either c_1 or c_2 is and remains in contact with the ground. In the case that c_1 remains in contact, the mechanism is constrained to have q^1 constant and q^2 zero, and hence this situation can be modeled as an unconstrained dynamical system with coordinates q^3 and q^4, and extra parameter q^1. If, instead, only c_2 is in contact with the ground, we can use the symmetry of the mechanism with respect to the two legs to relabel the coordinates as

$$\begin{bmatrix} q^1 \\ q^2 \\ q^3 \\ q^4 \end{bmatrix} = q \quad \to \quad \mathbb{S}(q) := \begin{bmatrix} q^1 - \sin(q^3) + \sin(q^3 + q^4) \\ q^2 + \cos(q^3) - \cos(q^3 + q^4) \\ q^3 + q^4 \\ -q^4 \end{bmatrix} \tag{4.10}$$

and use the relabeled coordinates $\mathbb{S}(q)$ to express the constrained system as the same unconstrained dynamical system with new relabeled coordinates q^3 and q^4 and extra parameter q^1. For both situations, the continuous dynamics can be represented for example by the Euler-Lagrange equation (2.90) as

$$\bar{M}(q^3, q^4) \begin{bmatrix} \ddot{q}^3 \\ \ddot{q}^4 \end{bmatrix} + \bar{C}(q^3, q^4, \dot{q}^3, \dot{q}^4) \begin{bmatrix} \dot{q}^3 \\ \dot{q}^4 \end{bmatrix} + \begin{bmatrix} \frac{\partial V}{\partial q^3} \\ \frac{\partial V}{\partial q^4} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{4.11}$$

with $\bar{M}$ the bottom right 2×2 block of the mass matrix defined in (4.6), and $\bar{C}$ the corresponding Coriolis and centrifugal terms. We use the Euler-Lagrange equation here since it states the dynamics in terms of velocities $\dot{q}$ and accelerations $\ddot{q}$, rather than p and $\dot{p}$: this is useful since we use relabeling operations acting on the velocities, and since polynomial approximations for the joint trajectories (as used in the optimization routine described in Section 4.1) can be easily differentiated to obtain expressions for the velocities and accelerations. The port-Hamiltonian formulation (2.88) is, mathematically speaking, strictly equivalent, but for the purpose discussed here, it is not as convenient as the Euler-Lagrange formulation.

The reduced model is valid as long as the constrained foot remains constrained (this should be checked in simulation), and until the free foot hits the ground, which is the case when $q^4 = -2q^3$ or $q^4 = 0$ (this last condition is usually ignored, see the remark at the end of this section). If the free foot hits the ground, the full model with configuration state q can be used to compute the contact situation after impact; if the constrained foot releases contact instantaneously on impact of the free foot, then the situation after impact can again be described by the reduced model (in relabeled coordinates).

The relation between the momenta before and after impact is given by the projection operator $\mathbb{P}(q)$ defined in (3.44). The equivalent relation between the velocities before and after impact can be computed immediately as

$$\dot{q}(t_+) = M^{-1}\mathbb{P}M\dot{q}(t_-) = \left(I - M^{-1}A(A^TM^{-1}A)^{-1}A^T\right)\dot{q}(t_-) \tag{4.12}$$

Since c_1 is assumed to be fixed to the ground just before impact, we have $\dot{q}^1(t_-) = \dot{q}^2(t_-) = 0$, and hence only the last two columns of $M^{-1}\mathbb{P}M$ are of interest. For the compass-gait walker, (4.12) can be computed as

$$\dot{q}(t_+) = M^{-1}(q)\mathbb{P}(q)M(q)\dot{q}(t_-) =$$
$$\begin{bmatrix} \frac{2(m+m_H)(\cos(5q^3)-\cos(q^3))}{3m+4m_H-2m\cos(4q^3)} & \frac{-m\cos(3q^3)}{3m+4m_H-2m\cos(4q^3)} \\ \frac{2(2m_H\cos(2q^3)-m+2(m+m_H)\cos(4q^3))\sin(q^3)}{3m+4m_H-2m\cos(4q^3)} & \frac{-m\sin(3q^3)}{3m+4m_H-2m\cos(4q^3)} \\ \frac{-m-2m\cos(2q^3)+4(m+m_H)\cos(4q^3)}{3m+4m_H-2m\cos(4q^3)} & \frac{-2m\cos(2q^3)}{3m+4m_H-2m\cos(4q^3)} \\ \frac{8(m+m_H)(1+2\cos(2q^3))\sin^2(q^3)}{3m+4m_H-2m\cos(4q^3)} & \frac{-m+2m\cos(2q^3)}{3m+4m_H-2m\cos(4q^3)} \end{bmatrix} \begin{bmatrix} \dot{q}^3(t_-) \\ \dot{q}^4(t_-) \end{bmatrix} \tag{4.13}$$

where we used the fact that $q^4 = -2q^3$ on impact. It can be checked that $\dot{q}(t_+)$ is such that $\dot{c}_{2y} = \dot{c}_{2z} = 0$ in (4.8) as required by the fully inelastic impact of c_2. Whether c_1 instantaneously releases on impact of c_2 depends on $\dot{q}^2(t_+) = \dot{c}_{1z}$. If it is positive ($c_1$ starts moving upward), then indeed c_1 looses contact instantaneously when c_2 makes contact, and the double support phase is instantaneous. Expanding the expression for $\dot{q}^2(t_+)$ from (4.13), we find that this is the case if

$$\left(2(\sin(5q^3) - \sin(q^3))\dot{q}^3 - \frac{m}{m+m_H}\sin(3q^3)(2\dot{q}^3 + \dot{q}^4)\right)\Bigg|_{t=t_-} > 0 \tag{4.14}$$

It hence depends on the model parameters, as well as the configuration and velocity on impact, whether the assumption of an instantaneous double support phase is valid. For example, for $m \to 0$ and negative $\dot{q}^3$, the angle of impact q^3 must satisfy $-q^3 < \frac{1}{6}\pi$ for this assumption to be true.

If indeed the double stance phase is instantaneous, then we can use the reduced model for the dynamics, both before and after impact. The coordinates q^3 and q^4 before and after impact are related by the relabeling $\mathbb{S}(q)$ defined in (4.10). Similarly, the velocities before and after impact are related by consecutive projection (4.12) and relabeling (4.10), which together can be represented by the equation

$$\begin{bmatrix} \dot{q}^3 \\ \dot{q}^4 \end{bmatrix}_+ = \begin{bmatrix} \frac{m-2(m+2m_H)\cos(2q^3)}{-3m-4m_H+2m\cos(4q^3)} & \frac{m}{-3m-4m_H+2m\cos(4q^3)} \\ \frac{8(m+m_H)(1+2\cos(2q^3))\sin^2(q^3)}{-3m-4m_H+2m\cos(4q^3)} & \frac{2m\cos(2q^3)-m}{-3m-4m_H+2m\cos(4q^3)} \end{bmatrix}_- \begin{bmatrix} \dot{q}^3 \\ \dot{q}^4 \end{bmatrix}_- \tag{4.15}$$

where the subscript minus (plus) indicates the value of the variables before (after) momentum projection and coordinate relabeling. This reset equation, together with the relabeling operation (4.10) and the reduced dynamic equations (4.11) together form a complete model for the behavior of the compass-gait walker for motions with one foot in contact with the ground and only instantaneous double-stance phases.

Remark. The reset map should be applied when contact occurs, which is technically whenever c_{2z} becomes zero, or, whenever $q^4 = -2q^3$ or $q^4 = 0$. This last condition is due to the kinematics of the mechanism (the fact that it has two straight

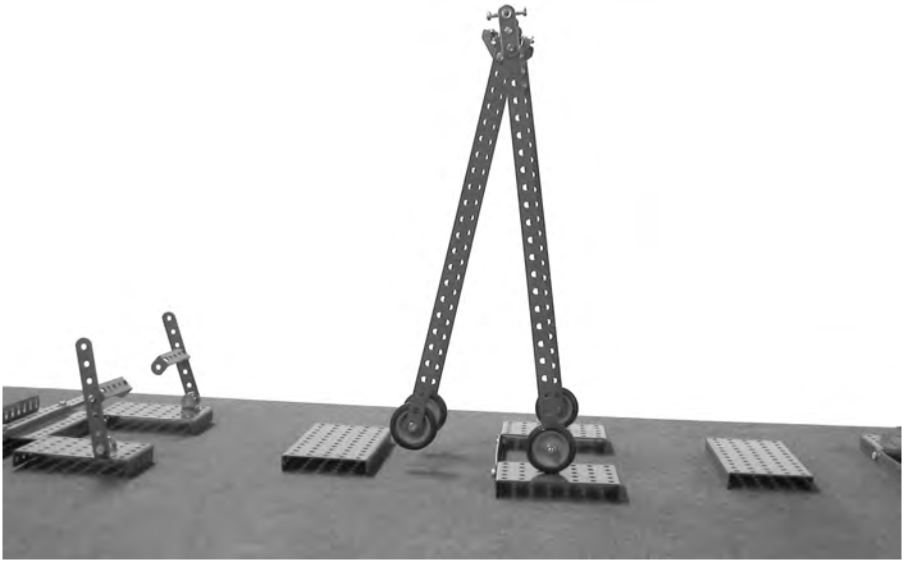


Figure 4.3: Experimental setup of a compass-gait walker that uses stepping stones to avoid toe stubbing of the swing leg.

legs), and results in foot scuffing whenever the swing leg passes the stance leg. In practice, this would prevent the mechanism from walking; it would never be able get one leg in front of the other. In simulations, this problem can be circumvented by relaxing the impact detection mechanism, *e.g.* by only detecting impact if c_{2z} becomes zero and q^4 is larger than some positive minimum angle (when walking left to right in Figure 4.2). In practical setups, the problem can be solved *e.g.* by using a retractable leg or by placing stepping stones at the expected foot placement locations, thus allowing the swing leg to pass below ground level (this is used in the experimental setup of Figure 4.3). △

4.2.2 Analysis of impact energy loss

The efficiency of a walking cycle is partly determined by the mechanical energy loss during the cycle, and partly by the non-idealness of the actuators in the system, *i.e.* the fact that actuators generally cannot absorb energy, and dissipate electrical power even if they do not supply mechanical power. The first aspect is analyzed in this section, the second aspect in Section 4.2.3. More precisely, we focus here on the energy loss due to the impact of the feet with the ground at the end of each step. Other mechanical losses, such as friction, are ignored.

As was shown in Section 3.3, the kinetic energy lost on impact is given by

(3.45), which can be written in terms of velocities as

$$\begin{aligned}\Delta U_k &= -\frac{1}{2}p^T(t_-)M^{-1}A(A^TM^{-1}A)^{-1}A^TM^{-1}p(t_-)\\ &= -\frac{1}{2}\dot{q}^T(t_-)A(A^TM^{-1}A)^{-1}A^T\dot{q}(t_-)\\ &= -\frac{1}{2}\dot{q}^T(t_-)\mathcal{D}\dot{q}(t_-) \qquad (4.16)\end{aligned}$$

where we defined the symmetric positive semi-definite matrix $\mathcal{D}(q)$ as

$$0 \leq \mathcal{D}(q) := A(q)(A^T(q)M^{-1}(q)A(q))^{-1}A^T(q) \qquad (4.17)$$

The kinetic energy loss on impact is hence a quadratic function of the impact velocities $\dot{q}(T)$ depending generally on the inertial properties of the system as well as the configuration $q(T)$ on impact.

If we consider the compass-gait walker and assume a symmetric walking cycle with instantaneous double-stance phase (as in the previous section), we can write the energy loss as a quadratic function of only $\dot{q}^3$ and $\dot{q}^4$, with $\mathcal{D}(q)$ a positive semi-definite 2×2 matrix. For general m and m_H, the symbolic representation of $\mathcal{D}(q)$ is too large to fit here, but for $m \rightarrow 0$ it is simply

$$\mathcal{D}(q)|_{m\rightarrow 0} = \begin{bmatrix} m_H \sin^2(q^4) & 0 \\ 0 & 0 \end{bmatrix} \qquad (4.18)$$

This makes sense intuitively: for $m = 0$, the velocity $\dot{q}^4$ does not influence the energy loss, since the inertia of the swing leg around the hip is zero and hence no energy is stored in the swing leg. Furthermore, as illustrated in Figure 4.4, the velocity of the hip mass changes from being tangent to a circle around one leg to tangent to a circle around the other leg, and the impulsive force on impact removes the part of the pre-impact velocity that is not along the post-impact circle. Hence, if the circles are close to each other, little is removed and hence little energy is lost, whereas if the circles are orthogonal[1] to each other, the remaining velocity is zero and hence all energy is lost. This also follows from (4.18), where for $q^4 = \frac{\pi}{2}$ we have $\Delta U_k = \frac{1}{2}m_H(\dot{q}^3)^2$, which is equal to the kinetic energy U_k for $m = 0$, and hence indeed all kinetic energy is lost on impact.

For general nonzero m, the matrix $\mathcal{D}(q)$ is too complex to study directly. Instead, we study its generalized principle directions, *i.e.* the velocity directions that result in minimal and maximal energy loss on impact, as given by (4.16).

The principle directions of a real symmetric matrix X are given by the singular

[1] Orthogonality of the velocities is well-defined here, since the system is essentially just a point mass, with a Euclidean state space and metric mI.

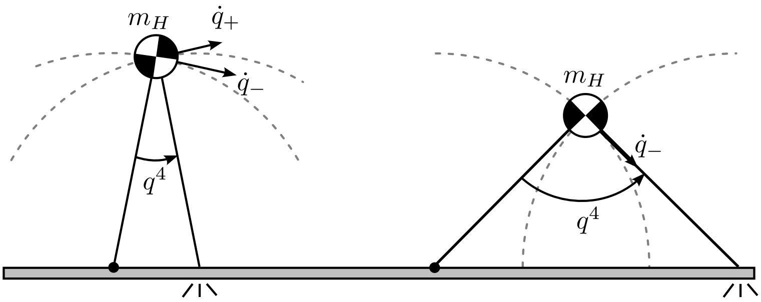


Figure 4.4: Comparison of the hip velocity before ($\dot{q}_-$) and after ($\dot{q}_+$) impact for two impact angles q^4. For $q^4 = \frac{\pi}{2}$ (right figure), the velocity after impact is zero.

value decomposition (Trefethen & Bau 1997), *i.e.* the decomposition of X as

$$X = U\Sigma V^T = U\Sigma U^T = U \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix} U^T \quad \text{with} \quad \sigma_1 \geq \ldots \geq \sigma_n \geq 0 \quad (4.19)$$

with U an orthogonal matrix, and $U = V$ since X is symmetric. The numbers σ_i are called the singular values of X. The matrix X can act as a quadratic form on vectors x, *i.e.* it can map a vector x to a number $x^T Xx$. When this quadratic form is applied to the unit sphere (all vectors satisfying $x^T x = 1$), the resulting set of vectors $(x^T Xx)x$ forms an ellipsoid, and the radii of this ellipsoid are precisely equal to the singular values σ_i. Furthermore, the principle axes of the ellipsoid are given by the columns of the matrix U^T, *e.g.* the first column of U^T is the vector that is enlarged most by the quadratic form, and the last column of U^T is the vector that is enlarged least.

We cannot use this singular value decomposition directly to study the singular values of $\mathcal{D}$, since the unit sphere $\dot{q}^T\dot{q} = 1$ has no physical meaning and would give coordinate-dependent results. Instead, we study the effect of $\mathcal{D}$ on vectors satisfying $\dot{q}^T M(q)\dot{q} = 1$, *i.e.* directions with constant kinetic energy. To adapt the singular value decomposition to this situation, we use the Cholesky factorization $M(q) = G^T(q)G(q)$ (which exists since M is positive definite and symmetric), and determine the singular value decomposition of $X = G^{-T}\mathcal{D}G^{-1}$. This decomposition again provides the principle directions of the quadratic form when applied to the unit sphere $x^T x = 1$. In addition, if we parameterize the vectors x as $x = G\dot{q}$, we see that this decomposition gives the principle directions of the quadratic form

$$x^T Xx = (G\dot{q})^T(G^{-T}\mathcal{D}G^{-1})(G\dot{q}) = \dot{q}^T\mathcal{D}\dot{q} \quad (4.20)$$

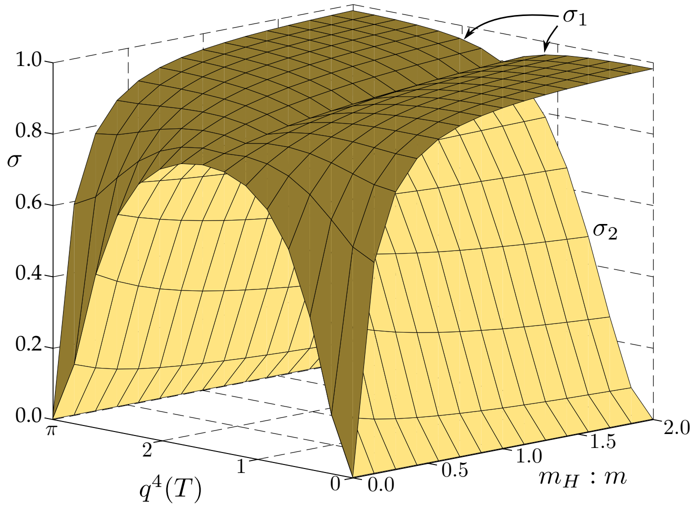


Figure 4.5: Plot of the singular values describing the energy loss on impact for various mass ratios $m_H : m$ and impact angles $q^4(T)$.

when applied to the space

$$x^T x = (G\dot{q})^T (G\dot{q}) = \dot{q}^T M \dot{q} \tag{4.21}$$

and hence the singular value decomposition, constructed in this way, describes physically meaningful principle values σ_i and corresponding principle directions, given by the columns of $G^{-1}U^T$. Note that the singular values must be between zero and one, since no more than 100% and no less than 0% of the kinetic energy can be lost on impact.

Figure 4.5 shows a plot of the singular values for the compass-gait walker with varying mass ratio $m_H : m$ and impact angle $q^4(T)$. Figure 4.6 shows the singular values and lists several principle directions for the impact velocity $\dot{q}$, with the parameters fixed at $m = 1$ kg and $m_H = 5$ kg, just as they are used in Section 4.2.3.

The figures show the singular values describing the energy loss, *i.e.* the extreme cases of maximum possible loss and minimum possible loss. For general velocities not aligned with any of the principle directions, the energy loss will be somewhere between the singular values, *i.e.* between the two surfaces in Figure 4.5 and in the darker area between the two curves in Figure 4.5. For a given angle of impact, the velocity could then be chosen closer to the 'efficient' direction to minimize energy loss. However, for some configurations (namely $q^4 = \pi/2$ or $m_H = 0$) the two singular values are the same, and hence the energy loss is constant for all velocity directions.

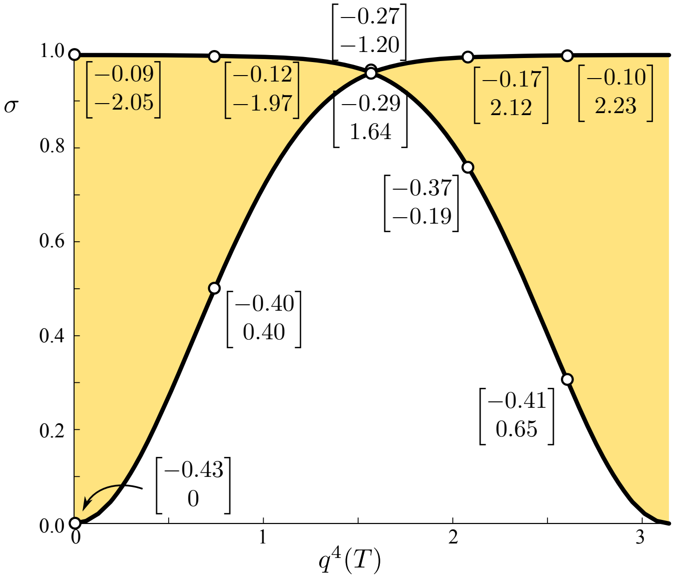
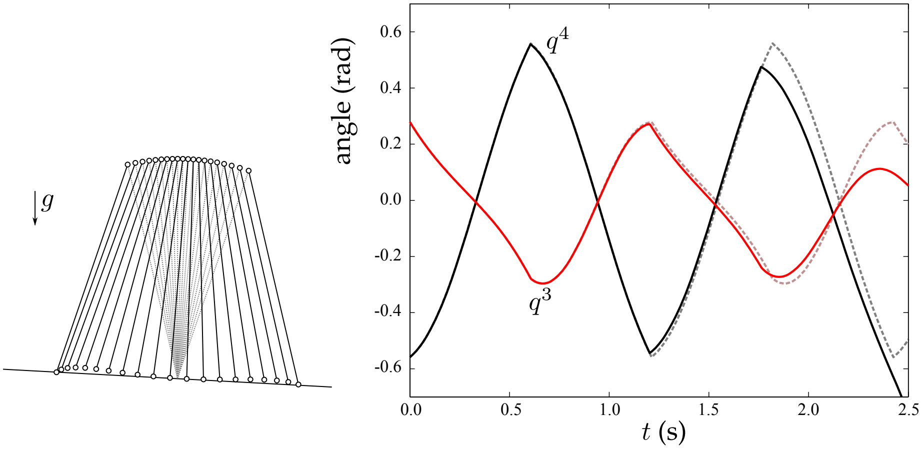


Figure 4.6: Plot of the singular values describing the energy loss on impact for $m = 1$ kg, $m_H = 5$ kg, and varying $q^4(T)$, with several principle directions $(\dot{q}^3, \dot{q}^4)$ listed.

The singular value analysis provides bounds on the energy loss during impact, and indicates efficient velocity directions for the end of a step. In this way, it can help suggest efficient walking strategies. For example, looking at Figure 4.6, it is efficient for small impact angles $q^4(T)$ to have an impact velocity that has large $\dot{q}^3$ and small $\dot{q}^4$ (*i.e.* moving the stance leg more than the swing leg), since then, the energy loss will be close to the lowest singular value. Note that the efficiency of the continuous dynamics is not taken into account yet (this is discussed in the next section), so the gaits that are most efficient overall may still have less efficient impact angles and velocities.

Besides for efficiency studies, the energy loss computed in (4.16) can be used in the search for purely passive dynamic walking motions. For such motions, the gravitational energy converted to kinetic energy *during* a step must be equal to the kinetic energy lost on impact at the *end* of a step. For the compass-gait walker, this gives the following equation

$$\begin{aligned}\Delta U_p &= \Delta U_k \\ U_p(t_-) - U_p(0) &= U_k(t_+) - U_k(t_-) \\ -2g(2m + m_H)\sin(\gamma)\sin(\frac{1}{2}q^4(t_-)) &= -\frac{1}{2}\dot{q}^T(t_-)\mathcal{D}(q)\dot{q}(t_-) \qquad (4.22)\end{aligned}$$



(a) Snapshots of a single step.

(b) Simulation of the gait; solid lines indicate the simulated unstable motion, dashed lines the optimized gait.

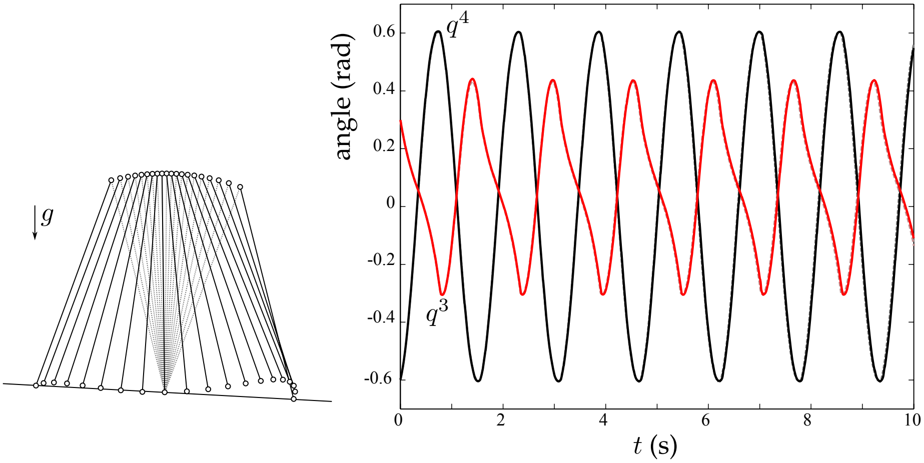
Figure 4.7: An unstable passive gait for the compass-gait walker.

which any passive (or otherwise energy-continuous) gait must satisfy. This equation also gives bounds on the minimally and maximally achievable speeds of the robots. If we choose for example $m = 1$ kg, $m_H = 5$ kg, $g = 9.81$ m/s^2, $\gamma = 3°$ and $q^4(t_-) = 0.5$ rad, we obtain that $\Delta U_p = -1.78$ J, such that also, by (4.22), the energy loss on impact must be 1.78 J per step. Since the singular values of $\mathcal{D}$ are 1.0 and 0.22, it means that the kinetic energy on impact must be at least 1.78 J and at most $1.78/0.22 = 8.08$ J. No passive limit cycles can exist for velocities outside this range (at least for this choice of parameters).

4.2.3 Analysis and simulation of passive dynamic walking

The energy balance (4.22) gives a necessary condition that the initial and final conditions of a walking cycle must satisfy in order for it to be a passive motion, but this condition is by no means sufficient. First, it is only one equation whereas two initial velocity variables need to be specified (assuming the initial configuration is fixed). But more importantly, it may not be true that the initial and final conditions that are compatible with the energy equation are connected by a natural passive motion of the system, *i.e.* that when started from the specified initial conditions, the system will end up at final conditions that are mapped to the same initial conditions again. To take these aspects into account, an analysis of the continuous part of the motion is necessary.

Given the reduced model (4.11) of the compass-gait walker for a single step, together with the projection and relabeling mappings (4.10) and (4.15), we can



(a) Snapshots of a single step.

(b) Simulation of the gait; solid lines indicate the simulated, dashed lines (barely visible) the optimized gait.

Figure 4.8: A stable passive gait for the compass-gait walker.

search for efficient trajectories using the approach outlined in Section 4.1. As a numerical example, we choose the parameters $m = 1$ kg, $m_H = 5$ kg, $g = 9.8$ m/s^2, $\gamma = 3°$ for the dynamic model, $Q(t) = I$ for the cost function, and $k = 10$ and $N = 20$ for the numerical approximation in Definition 4.2 of the minimization problem. As constraints, we specify the constraints in Definition 4.2, *i.e.* the compatibility conditions between the end and the beginning of a step, as well as the constraint $q^4(0) = -2q^3(0)$, *i.e.* the constraint that both feet are on the ground at the start of a step (and hence also at the end of a step by the compatibility conditions). As a numerical optimization routine, we use the sequential quadratic programming (SQP) method implemented in Matlab (The Mathworks 2005), with initial guesses for the trajectories and step time equal to $q(t) \equiv 0$ and $T = 1$ s.

Running the optimization routine produces a solution with cost 0.0004. Figure 4.7(a) shows snapshots of the resulting walking gait, which looks quite natural. However, when we simulate the passive system using the extended model equations (4.9), the robot falls down after three steps. Figure 4.7(b) shows a plot of the angles q^3 and q^4: although the simulation (solid lines) starts exactly like the optimized polynomial trajectories (dashed lines), the curves diverge after a few seconds, until the swing leg is not swung high enough anymore and the robot falls. Apparently, although the computed gait is natural (indeed, the uncontrolled robot initially follows the computed trajectory perfectly), it is not stable.

The results suggest that stable walking may be obtained by raising the swing leg higher above the ground, in order to ensure that the foot makes contact and the next step can start. We can add an additional constraint to the optimization problem to enforce this, for example, a constraint that forces the final downward velocity of the swing foot (just before impact) to be larger than some positive number v. The downward velocity (towards the ground) is given by the time-derivative of (4.4), from which we find a suitable extra inequality constraint

$$\dot{c}_{2z} = -\sin(q^3)\dot{q}^3 + \sin(q^3 + q^4)(\dot{q}^3 + \dot{q}^4) \leq -v \tag{4.23}$$

for some large enough $v > 0$ (we choose $v = 0.5$ m/s). Running the optimization routine (starting with the solution of Figure 4.7 as initial guess) gives a result with cost 0.0010, so higher than before (as expected, since we have added a constraint), and the trajectory shown in Figure 4.8(a). Feeding the initial conditions into the passive simulation gives the stable behavior shown in Figure 4.8(b), where the computed optimal gait is practically on top of the simulated behavior.

The results in this section show that, at least for this simple example, passive walking gaits can be found. The fact that there is still a nonzero cost is due to the approximation of the trajectories by polynomials. This approximation does not impede the search for stable passive limit cycles, since if the limit cycle is stable with some practical region of attraction, it should attract the solution obtained by this polynomial approximation.

For control purposes, the question whether an efficient limit cycle is stable or not is not so important; the controller can stabilize the cycle in case of disturbances anyway (provided that the system has enough control inputs). The important aspect of the optimized cycles is that *nominally*, little or no energy is required to follow them, and hence that nominal walking is efficient.

4.3 A planar walking robot with knees

The compass-gait walker from the previous section served as a simple example of how to construct simulation and analysis models of walking robots using the tools from Chapters 2 and 3. As a practical walking mechanism, though, it serves very little purpose: the straight unbendable legs require a special floor with stepping stones for it to be able to walk without stubbing its toes.

From nature, we now that a different way to prevent toe stubbing is to use bendable legs, *i.e.* using a knee joint that raises the foot some distance from the ground during the swing phase, but remains approximately straight during the stance phase. As argued for example by Pratt & Pratt (1999), adding a mechanical kneecap (an end stop) simplifies the control problem of keeping the leg straight during the stance phase, as it is only required to push the knee against the kneecap, not to position it exactly in the center.

In this section, we discuss modeling and analysis of an experimental robot which behaves like a planar robot with knees and kneecaps. Although it is designed to be used as a controlled robot walking on level ground, here we consider the problem of passive dynamic walking down a slope.

4.3.1 The walking robot Dribbel

Figure 4.9 shows 'Dribbel', the kneed walking robot developed by Dertien (2005), Beekman (2004), and van Oort (2005) at the University of Twente. It is a mechanism consisting of four legs, which are connected in pairs, such that both the two outer legs and the two inner legs move together. This construction was first used by McGeer (1991), and provides an easy way to construct a physical three-dimensional robot that behaves essentially two-dimensional; the leg pairs prevent it from falling sideways.

The leg pairs are joined in two concentric aluminum tubes, which contain most of the electronics as well as a motor and a torque sensor. The knees of the robot have mechanical kneecaps that prevent them from hyper-extending. They also contain electromagnets that can be actuated in order to hold the leg straight. The feet of the robot are small U-shaped metal plates that are kept aligned with the leg by simple elastic bands. They contain switches that are activated when the feet touch the ground. Finally, all joints are equipped with rotational encoders that measure the joint angles.

Inspired by human locomotion, we choose to activate the magnets on the knees during the whole stance phase, such that knee buckling is prevented. When the stance foot releases the ground (as detected by the contact switch), the magnet is deactivated and the knee is free to flex during the swing phase. Then, before the lower leg hits the kneecap on its forward swing, the magnet is reactivated in order to catch and hold the lower leg straight for the subsequent stance phase.

Dribbel is similar to planar kneed robots developed before by *e.g.* McGeer (1990*b*), Garcia et al. (2000), and Wisse & van Frankenhuyzen (2003). One of the main differences with this robot is, however, the use of an electric hip motor (McGeer and Garcia did not use actuators at all, and Wisse used McKibben muscles). Furthermore, Dribbel has (approximately) point feet, whereas other robots have some type of curved feet. Although curved feet have been proved to allow more efficient walking (McGeer 1990*a*), the use of point feet simplifies analysis, position sensing, and control.

4.3.2 Dynamic models of Dribbel

Simulation model

To develop a simulation model of Dribbel, we make the assumption that the links of the robot are rigid and the joints ideal. We also assume that the legs move in pairs, meaning that the inner legs move together and the outer legs move together, in particular the lower legs. These assumptions imply that the mechanism essentially consists of four rigid bodies: the outer upper legs (joined by the inner hip tube), the inner upper legs (joined by the outer hip tube), the outer lower legs (not joined, but assumed to move together), and the inner lower legs (not joined,

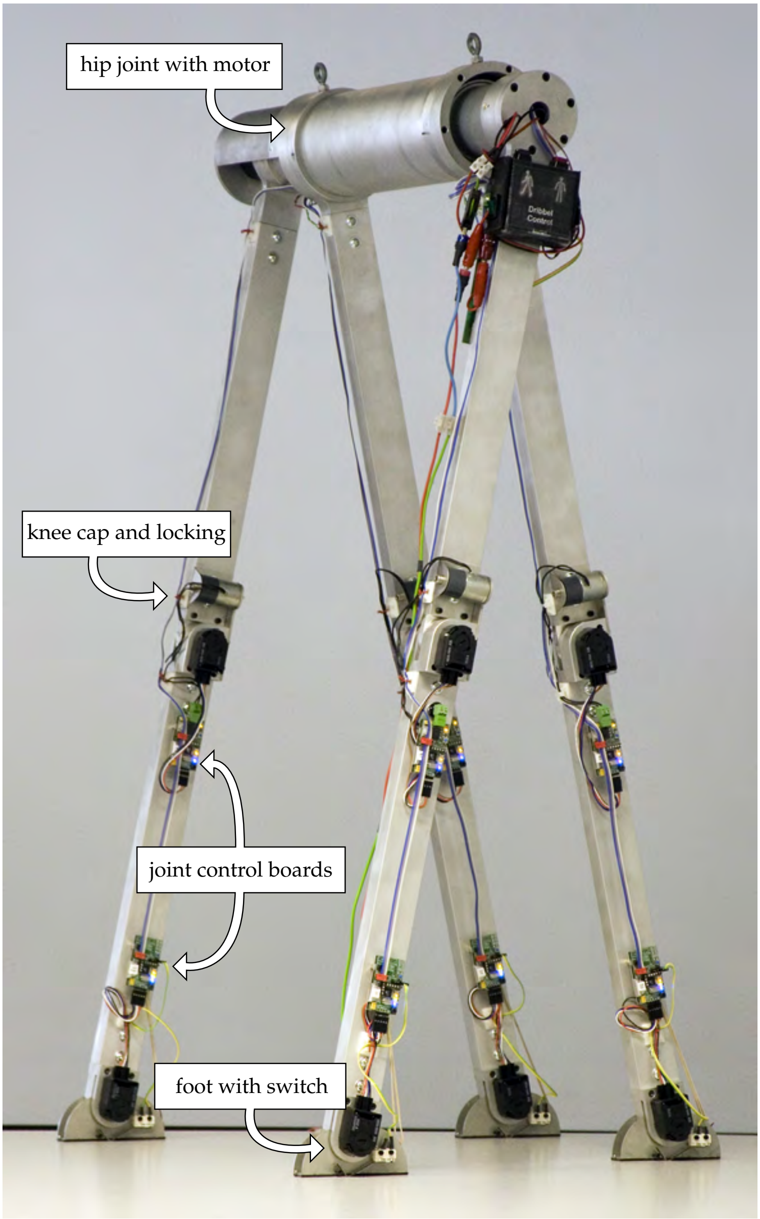


Figure 4.9: Experimental kneed walking robot ‘Dribbel’.

Table 4.1: Measured parameters of Dribbel.

part	amount	unit
upper leg length (l_u)	47	cm
lower leg length (l_l)	43	cm
hip mass (excl. batteries)	4.0	kg
upper leg mass	240	g
knee mass	540	g
lower leg mass	240	g
foot mass	280	g

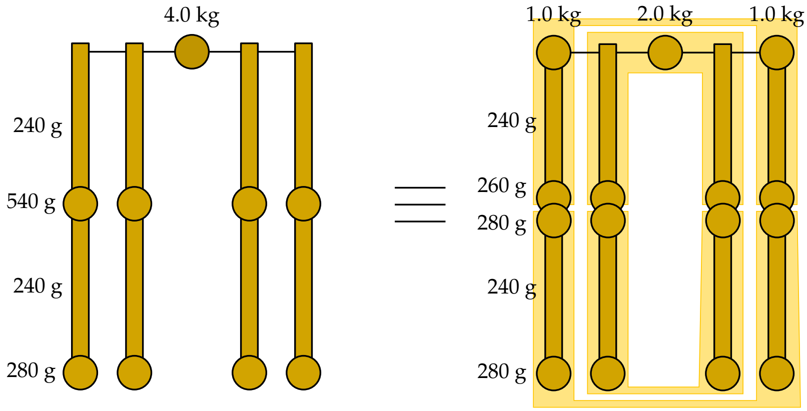


Figure 4.10: Schematic frontal view of Dribbel, showing the reassignment of mass to transform measured data (left) to equivalent model parameters (right).

but assumed to move together).

We estimate the mass, center of mass, and inertia of these four rigid bodies as follows. We first measure the mass of the *mechanical* parts of the robot, and we assume the mass of the links to be distributed uniformly, and the mass of the joints to be concentrated in a point. The results are shown in Table 4.1. Then, we reassign the masses to an equivalent mass distribution on the *model* parts as shown in Figure 4.10: the point masses at the knees are divided between the upper and lower legs, and the point mass at the hip is divided between the outer and inner legs. With this redistribution, the mass and inertia are equally distributed for the inner and outer legs, and we can compute the parameters as

$$\text{upper legs:} \quad m_u = 3.0 \text{ kg} \quad J_u = 0.108 \text{ kgm}^2 \tag{4.24}$$

$$\text{lower legs:} \quad m_l = 1.6 \text{ kg} \quad J_l = 0.059 \text{ kgm}^2 \tag{4.25}$$

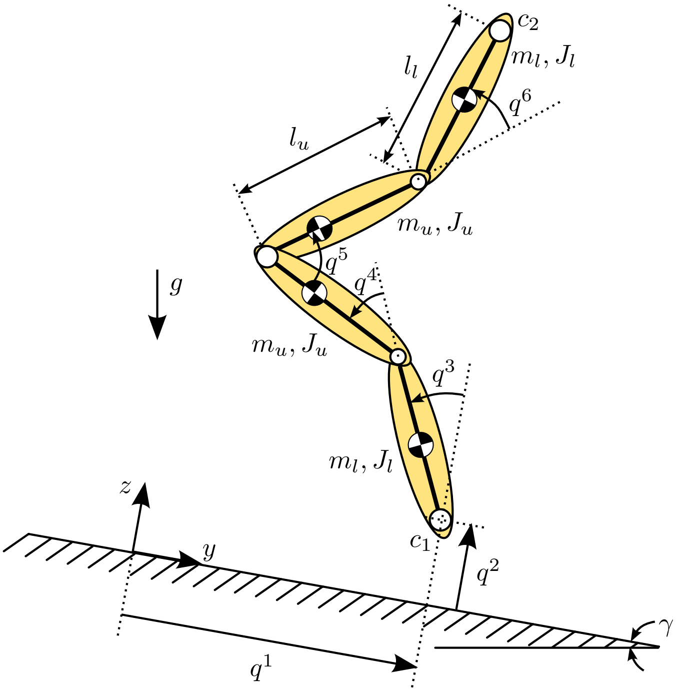


Figure 4.11: Setup and choice of coordinates and parameters for a planar walking mechanism with knees on an inclined flat floor.

where the inertias J_u and J_l are around the lateral axis through the centers of mass. The center of mass of the lower leg is (by construction) in the middle of the leg, and the center of mass of the upper leg is at a point

$$z_{\text{com,u}} = \frac{2 \cdot 0.240 \cdot 0.47/2 + 2.0 \cdot 0.47 + 2 \cdot 0.260 \cdot 0}{2 + 2 \cdot 0.240 + 2 \cdot 0.260} = 0.35 \text{ m} \qquad (4.26)$$

from the knee upward, so some distance above the center of the upper leg.

Since the robot was constructed to behave like a planar walker, we construct a simulation model that only represents the lateral behavior of the robot. Furthermore, we model the feet as point feet, and possible contact with the ground as rigid contact. Finally, we model the knees as rotational joints with stiff spring-damper combinations for hyper-extending angles, combined with tight control loops that fix the knee angles if the knee-locking electromagnets are active.

Figure 4.11 shows the resulting setup, with coordinates q^1 through q^6 as indicated. The springs and actuators in the knees are implemented as external elements and are connected to the mechanism through power-ports. The resulting model can be used for simulation of the planar behavior of Dribbel.

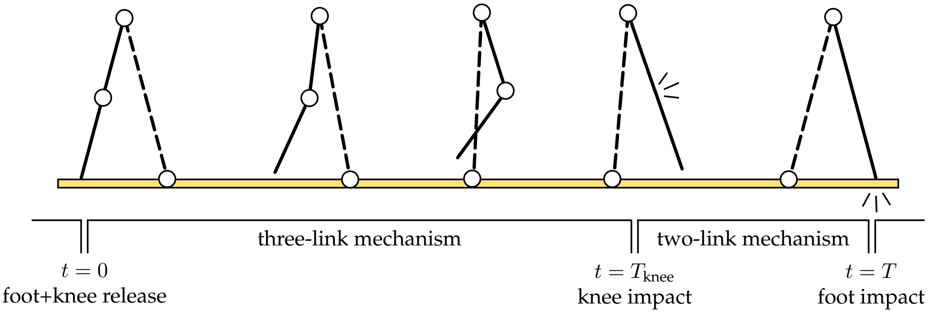


Figure 4.12: A single step of Dribbel is split into two phases: the first phase from foot lift-off until knee strike (modeled as a three-link mechanism), and the second phase from knee strike until foot strike (modeled as a two-link mechanism).

Analysis model

To obtain a model of Dribbel that is suitable for analysis, we make the same assumptions as for the compass-gait walker of Section 4.2, namely that only one foot is on the ground at the same time, that the gait is symmetric, and that the motion of the mechanism is such that that foot remains on the ground throughout the whole step. However, additional assumptions and model modifications are needed, since the fast dynamics resulting from the stiff spring and damper modeling the kneecap can not be adequately approximated by a smooth polynomial function of reasonably low order.

Therefore, we replace the stiff spring and damper by another impact equation, which sets the velocity of the knee joint instantaneously to zero. Furthermore, we assume the walking cycle to be of the form shown in Figure 4.12: a step starts at $t = 0$ with the release of the new swing leg (due to the impact at the end of the last step), followed immediately by a release of the knee-lock. This implies that the initial conditions for the swing knee-joint are: position and velocity equal to zero. Then, the walking motion continues smoothly until the swing leg passes the stance leg and, at some time $t = T_{\text{knee}} > 0$, the knee hits the kneecap and the swing knee is locked for the rest of the step. Finally, the mechanism continues to move (with locked knees) until the swing leg hits the ground at $t = T > T_{\text{knee}}$, marking the start of the next step.

Under these assumptions, the analysis model of a step of Dribbel can be split into two continuous phases: between $t = 0$ and $t = T_{\text{knee}}$, it behaves like an unconstrained three-link serial mechanism (with coordinates q^3, q^5 and q^6), and between $t = T_{\text{knee}}$ and $t = T$, it behaves like an unconstrained two-link serial mechanism (with coordinates q^3 and q^5). The transitions between the phases occur when the lower leg hits the kneecap (at $t = T_{\text{knee}}$) and when the swing foot hits the ground (at $t = T$). The first impact is assumed not to affect the velocities $\dot{q}^1$, $\dot{q}^2$, and $\dot{q}^4$ (this is discussed in Section 4.3.3), and the second impact is assumed not to affect $\dot{q}^4$ and $\dot{q}^6$, which requires the electromagnets on the knees to be strong enough to resist the impact forces.

The impact at $t = T$ is modeled by a ground contact force equal to the force computed in Section 4.2.1 for the compass-gait walker; in fact the model is exactly the same, since the knees are assumed to be fixed both before and (just) after impact. The impact of the knee at $t = T_{\text{knee}}$ is implemented as the momentum projection defined in (3.44), or the equivalent velocity projection (4.12). We use the equations of the three-link mechanism in coordinates q^3, q^5, and q^6, compute the corresponding mass matrix and impact direction $A^T = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, and take the model parameters as obtained before, to finally obtain the relation between the velocities before and after knee impact as

$$\begin{bmatrix} \dot{q}^3 \\ \dot{q}^5 \\ \dot{q}^6 \end{bmatrix}_+ = \begin{bmatrix} 1 & 0 & \frac{0.0516\cos(q^5)}{3.254+\cos(q^5)} \\ 0 & 1 & \frac{-0.999+0.0516\cos(q^5)+0.237\cos(2q^5)}{(-1.803+\cos(q^5))(1.804+\cos(q^5))} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^3 \\ \dot{q}^5 \\ \dot{q}^6 \end{bmatrix}_- \tag{4.27}$$

The structure of this matrix is as expected: the knee velocity $\dot{q}^6_+$ after impact equals zero, independently of the pre-impact velocities. Furthermore, if the knee hits the kneecap with zero velocity ($\dot{q}^6_- = 0$), there is no impact and the velocities before and after $t = T_{\text{knee}}$ are the same.

4.3.3 Impact analysis and efficient walking

The impact of the knee at T_{knee} was assumed not to affect $\dot{q}^1$, $\dot{q}^2$, and $\dot{q}^4$. This means that the configurations and velocities on impact should be such that the ground constraint remains satisfied and that the stance knee remains locked.

The first condition can be characterized by considering the vertical velocity $\dot{q}^2_+$ of the stance leg in the case that ground contact forces are *not* applied, *i.e.* by considering the projection operation (4.12) for a mechanism with free coordinates q^1, q^2, q^3, q^5, and q^6, and with impact force in the direction $A^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}$. The post-impact velocity $\dot{q}^2_+$ (with $\dot{q}^2_- = 0$) can be computed from this projection operator as

$$\dot{q}^2_+ = \frac{0.106\sin(q^3 - q^5) - 0.0378\sin(q^3 + q^5)}{9.391 - 0.488\cos(2q^3)}\dot{q}^6_- \tag{4.28}$$

The assumption that ground contact is maintained on knee impact is correct if $\dot{q}^2_+ < 0$ in the absence of contact forces, *i.e.* for $\dot{q}^2_+$ as in (4.28). Figure 4.13 shows a plot of the region in which $\dot{q}^2_+ < 0$ as a function of the configuration q^3 and q^5 on impact (where we took $\dot{q}^6 > 0$ since the swing knee is being locked). From the figure, we see that for most configurations, and certainly for practical configurations $q^3 < 0$ and $q^5 > 0$, the first condition is satisfied.

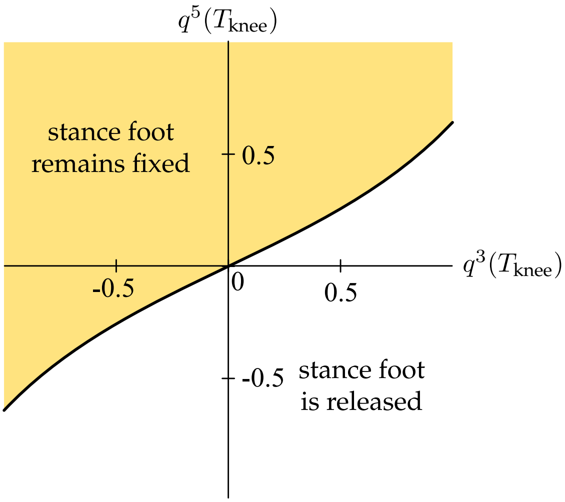


Figure 4.13: Postures for which the stance foot remains fixed on knee impact.

Assuming that ground contact is maintained, we can check the second condition (*i.e.* whether the stance knee remains locked) by computing the post-impact velocity $\dot{q}^4_+$ of the stance knee in absence of locking forces. We now compute the projection operator for the mechanism with coordinates q^3, q^4, q^5, and q^6 and swing knee constraint defined by $A^T = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}$, giving

$$\dot{q}^4_+ = \frac{0.111 \cos(q^5)}{3.071 - \cos^2(q^5)} \dot{q}^6_-$$

which results in $\dot{q}^4 > 0$ for $\dot{q}^6 > 0$ and for hip angles satisfying $|q^5| < \pi/2$. Hence, for practical hip angles, the kneecap on the stance leg does not prevent knee buckling on impact of the swing knee, and the additional knee locking mechanism (using the electromagnets) needs to be strong enough to prevent buckling during the stance phase.

The previous discussion shows that the assumptions for the analysis model are valid for practical walking motions and if the knee locking mechanism is strong enough. We can hence use the analysis model to search for efficient walking gaits. We again follow the procedure outlined in Section 4.1, but now one step of the robot is split into the two phases shown in Figure 4.12. Since the transition between the two phases is marked by a jump in the velocities due to the knee impact, the joint trajectories cannot be approximated accurately by single polynomials for all $0 \leq t \leq T$ of reasonably low order. Instead, we use separate polynomial approximations for the two phases, with three polynomials (for q^3, q^5, and q^6) for the first phase and two polynomials (for q^3 and q^5) for the second phase, and then concatenate them by specifying suitable compatibility conditions for the trajectories before and after impact.

More precisely, the compatibility conditions for two polynomials are described by the two impacts at $t = T_{\text{knee}}$ and $t = T$ in the following way. The configurations q^3, q^5, and q^6 are continuous during the knee impact, and the velocities $\dot{q}^3$, $\dot{q}^5$, and $\dot{q}^6$ before and after impact are related by (4.27). Furthermore, the configurations q before and after the foot impact are related by a mapping $\mathbb{S}(q)$, in this case described by

$$\begin{bmatrix} q^1 \\ q^2 \\ q^3 \\ q^5 \end{bmatrix} = q \quad \rightarrow \quad \mathbb{S}(q) := \begin{bmatrix} q^1 - 0.9\sin(q^3) + 0.9\sin(q^3+q^5) \\ q^2 + 0.9\cos(q^3) - 0.9\cos(q^3+q^5) \\ q^3 + q^5 \\ -q^5 \end{bmatrix} \tag{4.29}$$

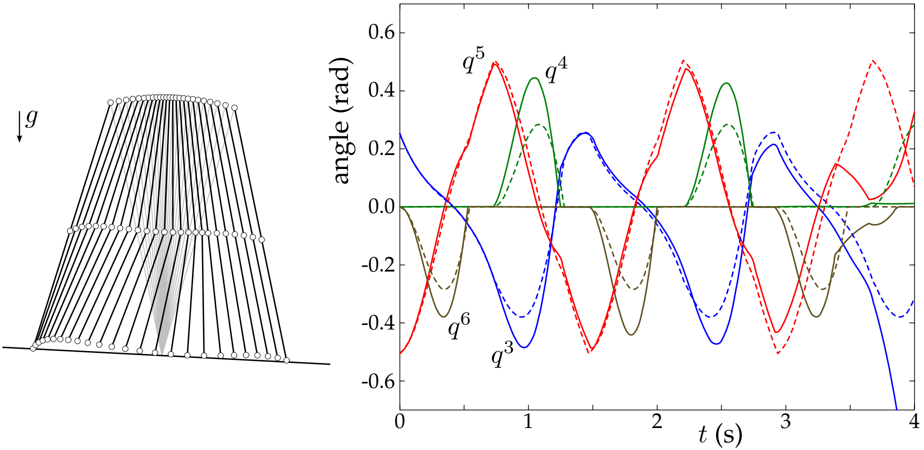
and $q^4 = q^6 = 0$ throughout the impact, and the relation between the velocities $\dot{q}$ before and after impact is again described by a consecutive projection $\mathbb{P}$ due to impact plus a relabeling according to the derivative of $\mathbb{S}(q)$ with respect to q. The symbolic expression for $\mathbb{P}$ is too large to include here, but it is computed as in (4.12), with $M(q)$ the mass matrix of the mechanism with free coordinates q^1, q^2, q^3, and q^5 (and q^4 and q^6 fixed and equal to zero).

In addition to the compatibility constraints, extra constraints should be added to ensure that no foot-scuffing occurs, which was the reason for including knees in the first place. The allowed motion of the swing foot can be described by the condition

$$l_l\cos(q^3) + l_u\cos(q^3) - l_u\cos(q^3+q^5) - l_l\cos(q^3+q^5+q^6) > \epsilon_f \tag{4.30}$$

with parameters l_l and l_u as in Table 4.1 and foot clearance $\epsilon_f > 0$, and where we used the fact that $q^4 \equiv 0$ throughout the step. Although this equation describes exactly the desired behavior (*i.e.* foot clearance), it depends in a highly nonlinear way on the coordinates q and is hence not suitable for use in a global optimization search. Solutions can be found more easily by using a simpler but approximate constraint, such as $q^6(T_{\text{knee}}/2) < -\epsilon_k$ to force knee bending. Then, when the optimal solution under this constraint has been found, the more exact constraint (4.30) can be used locally to further optimize the trajectories.

When the trajectories and constraints have been parameterized, we can solve the optimization problem (4.3) to obtain efficient walking trajectories for Dribbel. Since the parameterization as well as the dynamic equations for Dribbel are much more complex than for the compass-gait walker, the optimization does not just converge for all initial guesses for the solution. Instead, it is better to impose some extra artificial constraints (such as fixed initial angles and fixed T), optimize under these constraints, and then remove the extra constraints and further optimize the solution.



(a) Snapshots of a single step.

(b) Simulation of the gait; solid lines indicate the simulated unstable motion, dashed lines the optimized gait.

Figure 4.14: An efficient but passively unstable gait for the kneed robot Dribbel.

As an optimization goal, we choose to search for passive walking trajectories for Dribbel on a 3° slope (controlled walking on level ground is discussed in Section 5.3). As parameters in the numerical approximation (4.3), we choose $k = 10$, $N = 20$, and $Q(t) = I$ for both phases of the walking cycle. The total cost is then the sum of the cost of the individual two phases. The optimization produces the gait shown in Figure 4.14(a), with an associated cost $J = 0.0122$. This cost is considerably larger than the cost for the passive gait of the compass-gait walker (which was $J = 0.0010$), and hence it is unlikely that the computed gait for Dribbel can be approximated by a completely passive motion.

Indeed, simulation of the robot with the optimal initial conditions and zero control effort (except knee-locking) shows that the computed gait is not fully passive, see Figure 4.14(b): while the ankle and hip joints passively follow the computed gait quite closely, at least in the initial phase, the knee joint deviates severely. Still, despite this difference, the computed gait is a fair approximation of a passive (or at least efficient) motion of the walker, and indeed the uncontrolled robot takes a few steps before falling over.

4.4 A three-dimensional walking robot

As a final example of a walking robot, we consider the mechanism shown in Figure 4.15: a three-dimensional robot with two unit-length straight legs and a unit-length trunk, interconnected by three 2-DoF joints. The mechanism has six

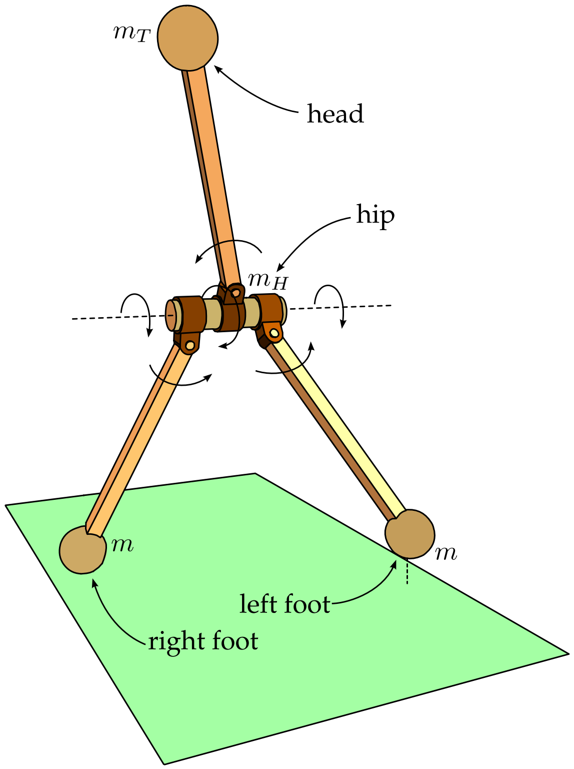


Figure 4.15: Mechanical setup of a 3D walker with trunk.

internal degrees of freedom and can move in three-dimensional space. The freedom of moving in 3D allows the walker to avoid foot-scuffing by swinging the swing-leg out and upward. Furthermore, the trunk provides additional degrees of freedom that, for example, may help in improving efficiency of the walking motions.

To simplify modeling and analysis of the robot, we assume that all joints are concentrated in one point (the hip) and that all mass is concentrated in points in the feet (mass m), the hip (mass m_H), and the head (mass m_T). This results in the simplified setup of Figure 4.16, which also shows the precise definition of the coordinates q.

The goals of the analysis in this section are to find efficient walking gaits for this mechanism that avoid foot-scuffing, and to investigate whether the trunk can help in increasing efficiency of the gait. A similar study, but of a *planar* robot with straight legs and a trunk, has been described in earlier research (Duindam & Stramigioli, 2005*a*, 2005*b*).

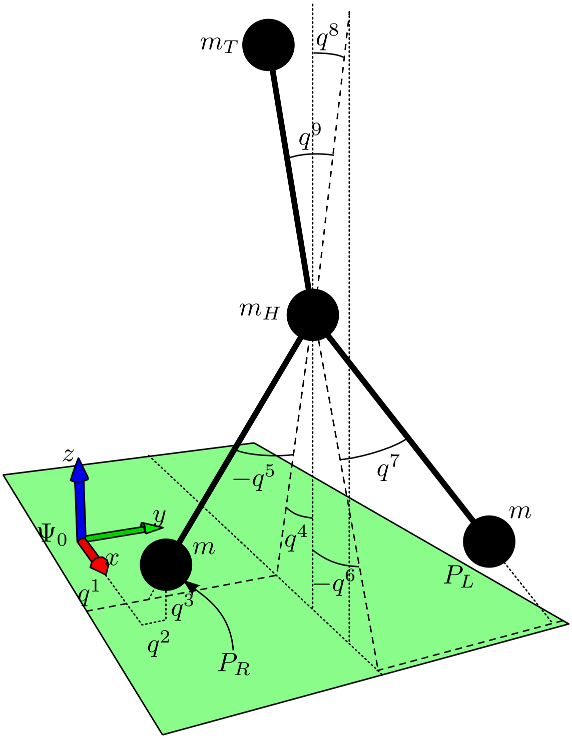


Figure 4.16: Choice of coordinates for the walker of Figure 4.15.

4.4.1 Dynamic models of the three-dimensional robot

The coordinates for the robot in Figure 4.16 have been chosen in a slightly different way than for the planar robots in Sections 4.2 and 4.3. The position of one of the feet is still specified directly (in this case in terms of the coordinates q^1, q^2, and q^3), but the orientation of the other limbs is specified relative to the reference frame Ψ_0, as opposed to relative to the first coordinates, as was done for the other examples.

The choice for these coordinates was made to allow easier intuitive interpretation of the coordinates; the definition of angles in 3D can quickly lead to ambiguities and mistakes. Another option would be to take q^1 through q^3 to be the coordinates of the hip point and leave the other coordinates as in the figure. This leads to a completely symmetrical formulation, but then the constraint of one foot being fixed to the ground is not easily translated into a constraint on the coordinates. Whatever coordinates are chosen, the modeling approach of Chapters 2 and 3 can be used in all cases to describe the dynamics of the system.

Simulation model

The dynamics of the mechanism can be described again in terms of the joint coordinates q. The legs are rigid bodies with mass m concentrated at the foot, so the center of mass is at the foot and the moments of inertia around all axes through the foot are zero. The point masses of the hip and the head can be combined into one rigid body, with mass $M = m_H + m_T$, center of mass at a position z_{com} (from the hip along the trunk) equal to

$$z_{\text{com}} = \frac{m_T}{m_T + m_H} \tag{4.31}$$

and moments of inertia around the center of mass equal to

$$J_x = J_y = m_H z_{\text{com}}^2 + m_T(1 - z_{\text{com}})^2 = \frac{m_H m_T}{m_H + m_T} \qquad J_z = 0 \tag{4.32}$$

where the local z-axis is along the trunk.

These three rigid bodies (two legs and one trunk) can be interconnected by the joints to obtain a model of the dynamics of the mechanism. Note that although this model can freely move in three dimensions, we did not use a homogeneous matrix to describe its absolute position and orientation. Instead, we used coordinates q^1 through q^3 to describe the position of a point on the mechanism (*i.e.* a foot) plus the other coordinates to describe the relative orientation of the legs relative to that point. The disadvantage of this approach is the coordinate singularities (for example, $q^5 = \pi/2$ makes q^4 a redundant coordinate), but the advantage is that we obtain a set of dynamic equations in terms of a simple vector q of unconstrained coordinates. Since in practical walking motions, the configurations are expressed by coordinates that are not close to singularity, this approach is indeed advantageous.

The next part of the simulation model describes the contact between the (point) feet and the ground. The generalized contact points P_R and P_L on the right and left foot can be expressed directly in terms of the coordinates q as

$$P_R^0 = \begin{bmatrix} q^1 \\ q^2 \\ q^3 \\ 1 \end{bmatrix} \qquad P_L^0 = \begin{bmatrix} q^1 + \sin(q^4)\cos(q^5) - \sin(q^6)\cos(q^7) \\ q^2 - \sin(q^5) + \sin(q^7) \\ q^3 + \cos(q^4)\cos(q^5) - \cos(q^6)\cos(q^7) \\ 1 \end{bmatrix} \tag{4.33}$$

and the corresponding generalized contact points on the ground are equal but with zero z-component.

To simplify the simulation model, we choose to model the contact as rigid. We choose the contact forces and torques to be linear forces both along the normal direction (z-axis in frame Ψ_0) and along the tangential directions (x and y-axes), acting at the contact points. We do not model possible contact torques around the vertical axis, since the feet are in point contact with the ground. The matrices A_R and A_L that define the directions of the constrained velocities of the right and left foot, respectively, (and hence the directions of the collocated contact forces) are given by

$$A_R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad A_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \cos(q^4)\cos(q^5) & 0 & -\sin(q^4)\cos(q^5) \\ -\sin(q^4)\sin(q^5) & -\cos(q^5) & -\cos(q^4)\sin(q^5) \\ -\cos(q^6)\cos(q^7) & 0 & \sin(q^6)\cos(q^7) \\ \sin(q^6)\sin(q^7) & \cos(q^7) & \cos(q^6)\sin(q^7) \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

which are obtained directly by taking the transpose of the partial derivative of (4.33) with respect to q. With the force directions defined by these matrices, we can implement the velocity projection (4.12) on impact as well as the finite contact forces (3.46) during contact, as described in Section 3.3.

Analysis model

For analysis purposes, we can simplify the model equations under some additional assumptions. If only the right foot is in contact with the ground ($q^3 = 0$), then we can describe the system by the free coordinates q^4 through q^9 and with parameters q^1 and q^2. By symmetry of the mechanism, we can use the same dynamic equations when, instead, only the left foot is in contact with the ground, if we relabel the coordinates as $\mathbb{S}(q)$, with

$$\mathbb{S}^T(q) := \begin{bmatrix} q^6 & -q^7 & q^4 & -q^5 & q^8 & -q^9 \end{bmatrix} \tag{4.34}$$

where the relabeling of q^1 through q^3 has been left out.

For the analysis model, we assume that only one foot is on the ground at the same time, and that hence the double support phase on impact is instantaneous. We also consider only walking gaits that are symmetric with respect to the left and right leg, such that only half a walking cycle needs to be found.

These assumptions imply that we only consider walking in a straight line here, since walking along a curve necessarily implies an asymmetry in the gait (the inner leg moves less than the outer leg). They also imply that this analysis model is only valid if the slope of the ground is directed in the direction of walking, *i.e.* only for walking straight up or down a slope. To study more elaborate walking motions, the analysis model should be a concatenation of two steps, one with each foot, just like the analysis model of the kneed biped of Section 4.3 is the concatenation of two phases, one before knee lock and one after knee lock.

Under these assumptions, the dynamics of the mechanism can be described by the (possibly relabeled) coordinates q^4 through q^9, together with a velocity projection when the swing leg hits the ground, *i.e.* when

$$\cos(q^4)\cos(q^5) - \cos(q^6)\cos(q^7) = 0 \tag{4.35}$$

When the double support phase is instantaneous, the system after impact is described in relabeled coordinates $\mathbb{S}(q)$, and hence the (projected) velocity variables need again be relabeled by the partial derivative of $\mathbb{S}(q)$ with respect to q.

4.4.2 Efficient walking on level ground

With the analysis model as presented before, we can use the procedure of Section 4.1 to look for efficient walking gaits. This time, we look for efficient gaits on level floor, and take again as the cost function the sum of all the torques squared, integrated (by means of a Riemann sum) over a step. Although most of the joint angles in Figure 4.16 are relative to the fixed world, and hence do not define suitable locations for actuators, it is not really clear from the figure how the actuators should be attached (collocated with what angles). For real control of this robot, the extended system of Figure 4.15 should be used, where the possible locations of the actuators are well-defined. For the simplified setup used for the analysis here, we choose the cost metric equal to the identity, and hence assume the actuation torques to be collocated with the velocities $\dot{q}$. We conjecture that the resulting optimal motion is also efficient (though perhaps not optimal) for a slightly different choice of cost metric.

As constraint equations in the optimization routine, we take the standard compatibility equations for the configurations and velocities at the beginning and end of a step, the condition (4.35) for $t = T$ to force the foot to be on the ground at the end of a step, as well as the constraint

$$\cos(q^4(t))\cos(q^5(t)) - \cos(q^6(t))\cos(q^7(t)) > \epsilon \qquad 0 < t_1 < t < t_2 < T \tag{4.36}$$

to obtain a ground clearance $\epsilon > 0$ of the swing foot for $t_1 < t < t_2$ and suitable $t_1 > 0$ and $t_2 < T$. Since we look for gaits on level ground, we also need to add a speed constraint of the form

$$\frac{\sin(q^6(0))\cos(q^7(0)) - \sin(q^4(0))\cos(q^5(0))}{T} = v_{\text{des}} \tag{4.37}$$

where v_{des} is the desired average forward walking speed. The expression on the left hand side of (4.37) equals the step length divided by the step time, and is hence indeed the average walking velocity. This extra constraint is necessary, since otherwise the optimal zero-cost motion would be to stand perfectly still with all limbs vertical. The parameters are fixed at $v_{\text{des}} = 0.5$ m/s, $\epsilon = 0.01$ m,

Table 4.2: Effects of the variation of mass distribution on the optimal walking gaits for the three-dimensional walker of Figure 4.16.

m_H (kg)	m_T (kg)	T (s)	x_{step} (m)	y_{step} (m)	J	P_{mech} (W)
1.0	5.0	0.52	0.26	0.20	4.18	1.82
2.0	4.0	0.63	0.31	0.18	3.12	1.35
3.0	3.0	0.68	0.34	0.15	2.56	1.11
4.0	2.0	0.73	0.37	0.15	2.18	0.99
5.0	1.0	0.77	0.39	0.14	1.91	0.90

$t_1 = T - t_2 = 0.1T$, $m = 1$ kg, and $m_H + m_T = 6$ kg, and the mass distribution between m_T and m_H is varied.

Due to the complexity of the robot and the constraints, the Matlab algorithms cannot find a solution if we run the optimization routine straight from a zero initial estimate with all constraints enabled. Instead, we start with the initial estimate $q \equiv 0$, $T = 1$ and add a constraint that fixes the initial configuration at $t = 0$ as

$$q(0) = \left[-0.25, -0.1, 0.25, 0.1, 0, 0\right]^T \tag{4.38}$$

We then optimize the cost function using only 6th-order polynomials for the joints. After this first optimization, the resulting optimized trajectories are used as initial estimate for the second optimization run, now with 10th-order polynomials and without the fixed initial configuration (4.38).

The results of this procedure for varying mass distribution m_T to m_H are shown in Table 4.2 and Figures 4.17 and 4.18. Table 4.2 shows for various mass distributions the values of the optimal step time T, step length x_{step}, step width y_{step}, cost J, and average mechanical power supply

$$P_{\text{mech}} = \frac{1}{T}\int_0^T \tau^T(t)\dot{q}(t)dt \tag{4.39}$$

The step length x_{step} and step width y_{step} are defined as the distance on impact between the feet in the x and y direction, respectively. The step length can be computed as $x_{\text{step}} = v_{\text{des}}T = 0.5T$. Figure 4.17 shows a top view of the trajectories traced out by the point masses at the feet, the hip, and the head for a number of steps. Finally, Figure 4.18 shows stick figure snapshots of three steps of the optimal gait for $m_T = m_H = 3$ kg.

From the table and figures, we can see that concentrating the mass at the hip instead of the head results in the following:

- The strides become slower and longer, *i.e.* both T and x_{step} increase.

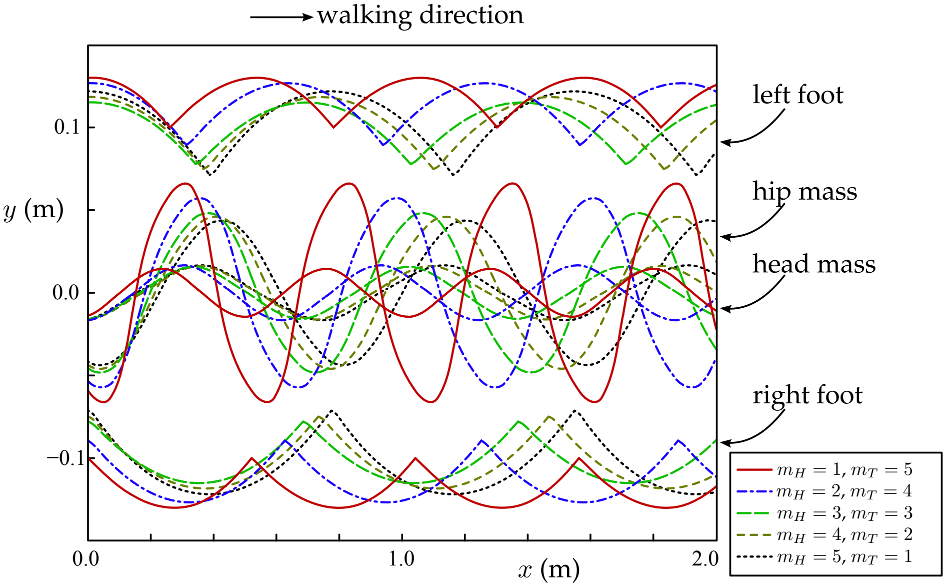


Figure 4.17: Top view of the traces of the feet, hip, and head masses for varying mass distribution.

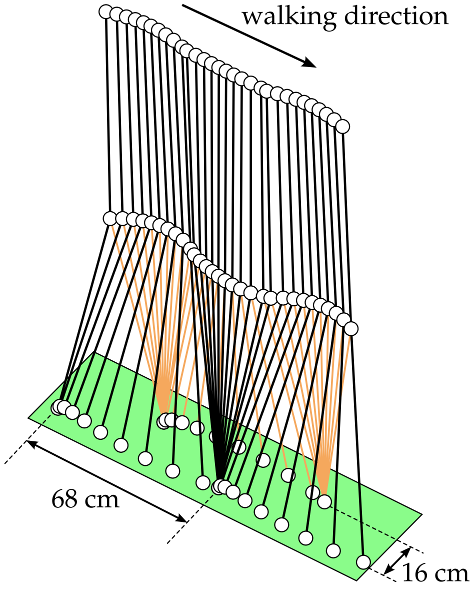


Figure 4.18: Snapshots of three steps of the optimal gait for $m_H = m_T = 3$ kg.

- Sideways hip swinging decreases, which is to be expected, since more mass is concentrated at the hip. The amount of sideways swinging of the head, on the other hand, is constant for all mass distributions.

- The step width decreases, but the sideways motion of the swing foot increases. This can be explained by looking at the hip swinging: for larger hip mass, hip swinging is decreased, and hence the foot needs to swing out more to obtain the required ground clearance.

- Both the mechanical power demand P_{mech} and the cost are J reduced.

We discuss the cost and energy aspects in more detail. From Table 4.2, we clearly see that increasing the hip mass while decreasing the head mass results in decreased cost and decreased mechanical energy supply. Duindam & Stramigioli (2005*a*) showed that also for a planar straight-legged walker with trunk, the lowest cost is achieved when the mass is located at the hip instead of the head. In that paper, the cost was computed as the squared required *internal* torques between the links, as opposed to the torque relative to the fixed world, as was done here. These results suggest that, although energy loss on impact may be reduced by locating the mass at the head and striking with low swing foot velocity, the optimal joint trajectories that connect these initial and final conditions are unnatural and require more actuation, even though some mechanical energy may be saved on impact. Also note that we assumed here that the trunk was connected to the legs through two freely moving rotational joints. If the trunk is connected through a constrained mechanism, the results may very well be different: Wisse (2004) showed that walking with a trunk connected through a constrained bisecting mechanism actually improves the energy efficiency.

Instead of looking at the mechanical power to judge energy efficiency, researchers often use a quantity called *specific resistance* (McGeer 1993) or *specific cost of transport*, which takes into account both mechanical power and walking speed. It is defined as

$$\eta := \frac{\text{energy dissipated}}{\text{weight} \times \text{distance travelled}} = \frac{\text{average dissipated power}}{\text{weight} \times \text{average walking speed}} \tag{4.40}$$

Sometimes the dissipated power only contains the mechnical power lost on impact (in which case η describes the specific mechanical cost of transport, denoted c_{mt}), and sometimes it contains the total energy dissipated in the system, *i.e.* including the electronics and actuator losses. If we only look at mechanical power loss, we can compute c_{mt} for the three-dimensional walker here as

$$c_{mt} = \frac{P_{\text{mech}}}{(2m + m_T + m_H) g v_{\text{des}}} = \frac{P_{\text{mech}}}{8 \cdot 9.81 \cdot 0.5} = 0.0255 P_{\text{mech}} \tag{4.41}$$

For the mass distributions of Table 4.2, this leads to a specific mechanical cost between 0.023 and 0.046. This is lower than for humans (0.05), and hence the computed motions can be really said to be efficient. Whether implementation results in a system with high overall efficiency depends on the location and efficiency of the actuators, and hence cannot be determined at this point.

The question remains whether it is useful to have a trunk or whether just a heavy hip is better. From the results in this section, it seems that for efficiency of walking it is better to have a mechanism without a trunk. However, these results are based on the assumption of full actuation (including ankle joints), which may not be true in practice. In the case of underactuation, it might be useful to have extra degrees of freedom in a trunk to obtain better or more efficient control possibilities. Furthermore, the cost function was chosen quite arbitrarily to be the summed square of the torques on the joints defined in Figure 4.16. Optimization of a comparable planar robot with a different choice of coordinates (Duindam & Stramigioli 2005*b*) gives the same conclusion and hence support the idea, but it still depends on the actual actuator configuration and efficiency whether the overall cost is reduced when no trunk is used. Finally, although the trunk by itself may not increase efficiency, it may be beneficial to have a trunk together with freely movable arms. This should be checked by analysis of an extended system including arms.

Chapter 5

Control of Walking Robots

The previous chapters described modeling and analysis techniques for walking robots, and showed how to find energy-efficient, natural gaits for several walking robots. By construction, these trajectories require little torque, and hence they could be implemented quite efficiently by putting local controllers (for example PID controllers) on the joints that force the joint angles to track the computed reference trajectories as time-varying setpoints. The global control problem is thus split into local one-dimensional tracking problems.

Unfortunately, this direct approach has several problems. First, the control tasks are specified by desired joint angles as a function of time, although the time aspect is not directly important in walking. Second, it is implicitly assumed that disturbances can best be compensated for by driving each joint individually back to its time-varying setpoint, whereas this may not be the case at all. Thirdly, the energy balance in the system is completely neglected, meaning *e.g.* that energy injected during disturbances may be dissipated by the controllers, whereas it could also be used to compensate for friction or other energy losses.

The idea behind the control techniques described in this chapter is to take into account these problems, and to approach control not as a forced exact trajectory tracking problem, but as an interactive process. Instead of trying to torture the mechanical structure into doing something externally specified, we try to follow its natural motions as much as possible, while *suggesting* (not forcing) a certain trajectory, and correcting only if necessary. As a first part of this approach, we describe in Section 5.1 how the natural dynamics of the mechanism can be shaped to attain a certain control goal, using an extension of the optimization techniques of Section 4.1. Then, in Section 5.2, we show a 'minimally invasive' approach to trajectory tracking, taking into account the aspects discussed before. Finally, Sections 5.3 and 5.4 discuss preliminary results in the control of Dribbel, the experimental robot introduced in Section 4.3, as well as a foot-placement control strategy for a simple three-dimensional walking robot.

5.1 Passive Mechanical Control

The use of actuators (whether electrical, hydraulic, pneumatic, or other) costs energy, even if no actual mechanical energy is supplied to the system: it costs energy due to dissipation in the actuators themselves, in the controlling circuitry, or because the actuators generally cannot absorb energy retrieved from the mechanism. This extra energy directly reduces the efficiency of the system, and it is hence beneficial to reduce the need for actuators as much as possible.

One way of reducing actuators requirements has been discussed in Chapter 4, where the reference trajectories of walking robots were chosen to be the 'most natural' motions of the system, *i.e.* the motions that require the least actuator torque (in some metric). Another way to reduce the actuator needs is to adapt the mechanical structure of the robot to simplify reaching certain control goals. The natural walking motion of a robot depends, for example, on its mass distribution and the presence and strength of other passive elements, such as springs. Changing the value of these parts of the mechanical structure changes the natural motions.

In the true spirit of the mechatronic design philosophy (van Amerongen & Breedveld 2003), we can thus change the mechanical structure in order to simplify actuator requirements and control action. This can be done during the construction phase of a robot, say, to design the mechanics most suitable for walking at a certain speed. But in addition, the mechanical structure can be adapted *during walking*, in order to optimize it for the desired walking speed at a certain time. Figure 5.1 shows an example of how this could be implemented in practice: the mass distribution m_0 on the lower legs can be adjusted by rotating the spindles x_m, and the (effective) stiffness of the spring at the hip joint can be changed by rotating the spindles x_s. An actual practical actuator with mechanically adjustable stiffness has been designed by Hurst et al. (2004). Obviously, we can treat these dynamically adjustable structures as structures with adjustable *parameters* only if the adjustments are made quasi-statically (*i.e.* very slowly), since otherwise the adjustment process itself introduces extra dynamics.

Van den Bogert (2003) showed how a system of exotendons, attached to a mechanical system, can be optimized to reduce torque requirements for a certain fixed walking gait of that mechanism. Here, we optimize both the mechanical structure and the walking gait at the same time. We include the adjustable mechanical structure in the optimization procedure of Section 4.1, by extending the set of degrees of freedom for the optimization routine to include the parameters describing how the mechanical structure can be adjusted. For example, the robot in Figure 5.1 has the extra free parameters x_m and x_s, and possibly m_0 and k_0, although these last two can only be changed off-line. The designer still has to choose what mechanical parameters to optimize: he/she has to decide what parts of the mechanism should be adjustable, for example, what degrees of free-

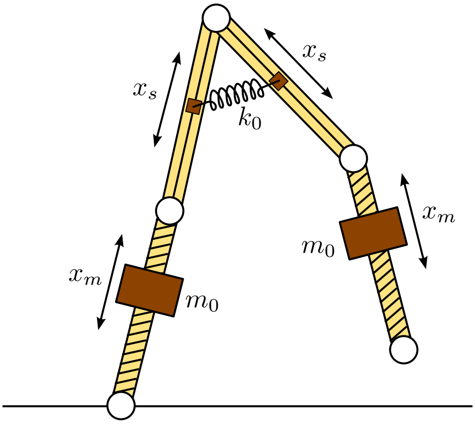


Figure 5.1: Example of the implementation of a robot with adaptable mass distribution on the lower legs and adaptable spring at the hip joint.

dom the force-displacement relationship of the spring should have. Some parts may be unsuitable for adjustment because of practical reasons, and some because adjusting them has little or no effect on the natural motions of the system.

The extra degrees of freedom in the optimization routine can result in achieving control goals at a lower cost. As an example, we show how the mechanical structure of the compass-gait walker can be adjusted to obtain highly efficient downhill walking at a range of speeds.

Example 5.1. We consider again the compass-gait walker of Section 4.2, in particular Figure 4.2, but now we consider the mechanical structure to be adjustable in the following three different ways:

1. an adjustable linear spring (stiffness k) and damper (damping factor d) are attached between the legs of the robot;

2. the mass distribution between the point masses m_H and m is adjustable, while the total weight of the robot $m_H + 2m$ equals 7 kg;

3. both the mass distribution and the spring-damper combination between the legs can be adjusted.

Note that, for simplicity, we vary the mass distribution by directly changing the parameters m and m_H, *i.e.* not using a pulley system such as in Figure 5.1. For the three situations outlined above, plus for the nominal case in which the structure is fixed at $m = 1$ kg and $m_H = 5$ kg (and no spring or damper), we search for natural walking gaits down a 3° slope at various average walking speeds

Table 5.1: Optimized cost at several walking speeds for the compass-gait walker with different degrees of freedom in the mechanical structure.

v_{des} (m/s)	nominal cost	spring-damper cost	mass cost	mass-spring-damper cost
0.5	1.4	1.4	0.062	0.037
0.6	0.44	0.44	0.011	0.0029
0.7	0.032	0.032	0.0023	0.0017
0.8	0.0059	0.0038	0.0032	0.0014
0.9	0.025	0.0030	0.0071	0.0016
1.0	0.28	0.0065	0.012	0.0027
1.1	0.82	0.015	0.016	0.0042
1.2	1.7	0.032	0.011	0.0054
1.3	2.8	0.061	0.011	0.0099

v_{des}. The required average walking speed constrains the relationship between the initial conditions and the step time as

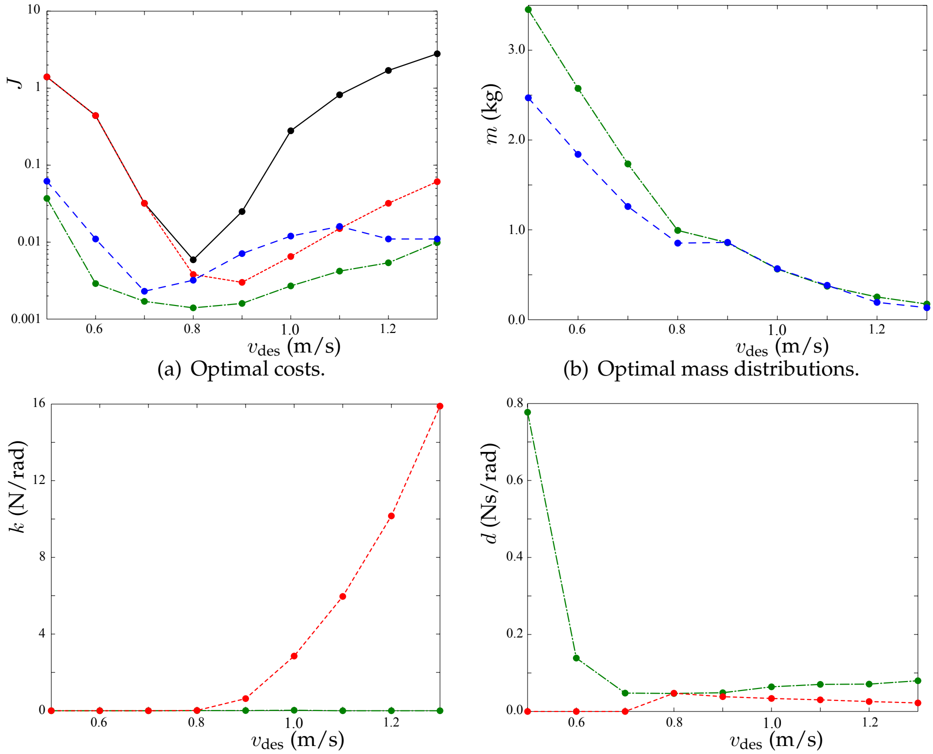
$$v_{\text{des}} = 2\frac{\sin(q^3(0))}{T} = -\frac{\sin(q^4(0))}{T} \tag{5.1}$$

We add this constraint to the optimization routine of Section 4.2.3, as well as the practical constraints that the stiffness, damping, and both m and m_H must be positive. Then, we search for the most efficient trajectories and the corresponding optimal parameters of the mechanical structure. The results of these optimizations are shown in Tables 5.1 and 5.2, as well as Figure 5.2.

From the tables and the figure, we see that the nominal robot with fixed mechanical structure has a small cost only at a speed around 0.8 m/s (indeed, the average walking speed of the passive motion in Figure 4.8 is 0.76 m/s); walking motions at lower and higher speeds require significantly more control effort.

We also see that adding an adjustable spring and damper at the hip joint can help reduce the cost for higher walking speeds, but not for lower speeds. This makes sense intuitively: if we add a spring between the legs, it increases the natural oscillation frequency between the kinetic energy of the legs and the potential energy of gravity plus spring, which in turn increases the natural walking speed. It can, however, never decrease the natural frequency, since only positive spring values are allowed.

By adjusting (only) the mass distribution on the legs, it is possible to obtain natural low-cost walking in the full range of speeds. For low speeds, more mass should be located on the legs (up to 3.5 kg per leg, leaving no mass at the hip joint), while for high speeds the mass should be transferred to the hip joint. For



(a) Optimal costs.

(b) Optimal mass distributions.

(c) Optimal spring constants.

(d) Optimal damping factors.

Figure 5.2: Optimal costs and structural parameters for the compass-gait walker at several walking speeds, depicted for the fully constrained structure (solid lines), variable spring and damper (dotted lines), variable mass distribution (dashed lines), and variable spring, damper, and mass distribution (dash-dotted lines).

Table 5.2: Optimized parameters at several walking speeds for the compass-gait walker with different degrees of freedom in the mechanical structure.

	spring-damper		mass	mass-spring-damper		
v_{des}	k	d	m	m	k	d
(m/s)	(Nm/rad)	(Ns/rad)	(kg)	(kg)	(Nm/rad)	(Ns/rad)
0.5	0.0	0.0	2.5	3.5	0.0	0.78
0.6	0.0	0.0	1.8	2.6	0.0	0.14
0.7	0.0	0.0	1.3	1.7	0.0	0.048
0.8	0.012	0.047	0.85	0.99	0.0051	0.047
0.9	0.63	0.038	0.86	0.86	0.0083	0.49
1.0	2.9	0.034	0.57	0.56	0.023	0.064
1.1	6.0	0.030	0.38	0.37	0.0	0.070
1.2	10	0.026	0.19	0.25	0.0	0.071
1.3	16	0.022	0.13	0.17	0.0	0.080

most speeds, the cost for walking using the optimal mass distribution is lower than when using the optimal spring/damper configuration.

Finally, if both the mass distribution and the spring/damper combination are adjustable, we obtain the lowest cost for walking. This is clearly to be expected, since optimizing only the mass distribution or only the spring/damper combination are special cases. From the values of the parameters, we see that in this general case, the spring at the hip joint is really not necessary; it has a very low stiffness value. Using only the mass distribution and the damper results in motions at more or less zero cost; the remaining cost is due to the approximation of the trajectories by polynomials. △

5.2 Port-based curve tracking

As discussed before, the joint angle trajectories computed in Chapter 4 as well as the extension in Section 5.1 are formulated as explicit functions of time. The time aspect in itself, however, is artificial and does not influence the dynamics or the efficiency of walking; only the relation between the joint angles is important.

One approach to remove the explicit time-dependency in the joint trajectories is to use what is sometimes called 'virtual time', as used for example by Chevallereau (2003). This virtual time is an extra variable that indexes what point (time instant) of the trajectory the current set point for the joint angles is. Depending on the current state of the robot, the virtual time may accelerate or decelerate in order to let the robot 'catch up' with the trajectory. This approach solves the problem of time-dependency to a certain extend, but still requires approximately proper initial conditions (the initial virtual time) to start off smoothly.

Instead of using an extra virtual-time variable, we propose to use fewer variables by describing the desired trajectory as a subspace of the configuration space, namely as the subspace Q_d of all configurations that are on the trajectory for some time t. For a single curve, this subspace is one-dimensional (since it can be parameterized by one variable), but the approach can be generalized to higher-dimensional subspaces, for example if the goal is to obtain convergence to a surface instead of a curve. The control objective can thus be better described in a time-independent way as 'convergence to Q_d'. Another, equivalent, formulation is to require certain outputs to become zero, as used in the (hybrid) zero dynamics approach of Westervelt et al. (2003).

A second problem with control of the individual joint angles is the implicit assumption on how to converge in the case of disturbance. Instead of returning straight (with 'straight' requiring the definition of a suitable metric) to Q_d, it may be more reasonable to converge in a different way, for example to avoid toe stubbing on the way, or to minimize energy loss.

To encode these aspects in the control task, we extend the desired behavior from a subspace to a vector field, *i.e.* a vector $w(q)$ at each point q, such that the integral curves of this vector field describe the path in coordinate space that the system should follow. This means that Q_d should be one of the integral curves, and hence that $w(q)$ should be tangent to Q_d for all $q \in Q_d$. Finally, the control laws defined in the following section require $w(q) \neq 0$ (which makes sense, since otherwise the desired direction is undefined), hence the vector field should be designed such that it is nonzero for as large a region as possible. It is generally not possible to have nonzero $w(q)$ for all q, due to either the shape of Q_d or the shape of the configuration space. For example, on the sphere $\mathbb{S}^2$, this is the famous 'hairy ball theorem' (Milnor 1978). Therefore, from now on, we only consider the system in the open subspace $\mathcal{D} \subset Q$, defined as

$$\mathcal{D} := \{q \in Q \mid w(q) \neq 0\} \tag{5.2}$$

i.e. the space of all points $q \in Q$ where the vector field is nonzero.

When the control objective has been encoded as a vector field $w(q)$, we can construct a controller that realizes convergence to this vector field. The next section describes an approach based on the representation of the system as an interconnection of port-Hamiltonian subsystems, with one representing the desired behavior (with associated energy), one the undesired behavior (with associated energy), and one the power-continuous interconnection structure between the two and the input port. The results can also be found in (Duindam & Stramigioli, 2003*b*, 2004*b*), and are based on earlier work (Duindam, Stramigioli & Scherpen 2004). The idea of describing the control objective as a vector field is also used in the method of Passive Velocity Field Control, introduced by Li & Horowitz (1995, 1999) and used in the control of walking machines by Yamakita et al. (2000, 2001).

5.2.1 System representation encoding desired behavior

We start the derivation of the controller from a system representation in port-Hamiltonian form, and, for simplicity, we consider a system of the form (2.61) with direct and full actuation, such that it can be represented as

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} &= \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & I \end{bmatrix}\begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} \end{aligned} \tag{5.3}$$

with Hamiltonian $H(q,p) = \frac{1}{2}p^T M^{-1}(q)p + V(q)$ and control port (u,y). The desired behavior is assumed to be encoded by a suitable vector field $w(q)$ on the configuration space $\mathcal{D}$. This desired behavior can be described equivalently by the co-vector field $M(q)w(q)$, since $M(q)$ is invertible for all $q \in Q$.

We now proceed to write the dynamics equations in coordinates that explicitly encode the desired behavior, *i.e.* movement along the vector field w. For this purpose, we choose new coordinates α for the momentum as $p = S^T(q)\alpha$, with $S^T(q)$ an invertible $n \times n$ matrix, smoothly varying in q, such that

- $Sw = \begin{bmatrix} * & 0 & \dots & 0 \end{bmatrix}^T$ for all $q \in \mathcal{D}$;
- $\bar{M} := SM^{-1}S^T$ is diagonal and independent of q.

These conditions imply that the first column of $S^T(q)$ is a nonzero scalar multiple of $Mw(q)$, and that the columns of $S^T(q)$ are orthogonal to each other (in the metric M^{-1}) and have constant length (in the metric M^{-1}). We also define the Hamiltonian $\bar{H}(q,\alpha)$ in new coordinates as

$$\begin{aligned} \bar{H}(q,\alpha) :&= H(q, S^T(q)\alpha) = \frac{1}{2}\alpha^T S(q)M^{-1}(q)S^T(q)\alpha + V(q) \\ &= \frac{1}{2}\alpha^T \bar{M}^{-1}\alpha + V(q) = \sum_{i=1}^{n} \frac{1}{2}\bar{M}_{ii}^{-1}\alpha_i^2 + V(q) \end{aligned} \tag{5.4}$$

where $\bar{M}_{ii}^{-1}$ is the inverse of the (i,i)th element of the diagonal matrix $\bar{M}$. Hence, the total energy can be written in coordinates (q,α) as the sum of the potential energy (a function of q) and the energies in the directions defined by the columns of S^T, each of which is a function only of the corresponding coordinate α_i. In particular, the kinetic energy associated with the velocity in the direction of the vector field is fully determined by the coordinate α_1. The other coordinates α_i indicate the kinetic energy in the directions perpendicular (in the metric M) to the vector field. Hence, the control objective 'follow the vector field' can now be expressed as 'keep all coordinates α_2 through α_n equal to zero'.

▼ Readers familiar with the concepts of Riemannian geometry may wonder whether it is always possible to find a matrix S^T that induces a constant diagonal metric $\bar{M}$. Indeed, in Riemannian geometry it is shown how coordinate transformations can give such an induced metric *only if* the original metric is differentially flat, which is in general not the case. However, in this case we use a transformation S^T only on the momentum variables, that is, it is *not* induced by a transformation on the q variables as is the case in the aforementioned Riemannian context. Here, we just want to find a transformation S^T (smoothly varying in q) that transforms a symmetric positive matrix M (smoothly varying in q) to a ▲ constant diagonal matrix, which is indeed always possible.

As was explained in the commutation diagram of Figure 2.9, choosing coordinates $p = S^T(q)\alpha$ is equivalent to choosing coordinates $v = S(q)\dot{q}$. Hence, following Lemma 2.16, the dynamic equations of the system (5.3) can be expressed in coordinates (q, α) instead of (q, p) as

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ \alpha \end{bmatrix} &= \begin{bmatrix} 0 & S^{-1} \\ -S^{-T} & S^{-T}\left(\frac{\partial^T(S^T\alpha)}{\partial q} - \frac{\partial(S^T\alpha)}{\partial q}\right)S^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial\bar{H}}{\partial q} \\ \frac{\partial\bar{H}}{\partial\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ S^{-T} \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & S^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial\bar{H}}{\partial q} \\ \frac{\partial\bar{H}}{\partial\alpha} \end{bmatrix} \end{aligned} \tag{5.5}$$

which are the same equations as in Lemma 2.16, modulo the renaming of the variables α, $\bar{M}$, and $\bar{H}$, plus that the input u is collocated with $\dot{q}$, not with $S(q)\dot{q}$. For conciseness, we can structure these equations in the following form

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q \\ \alpha_1 \\ \alpha_2 \end{bmatrix} &= \begin{bmatrix} 0 & S_1^{-1} & S_2^{-1} \\ -S_1^{-T} & 0 & X \\ -S_2^{-T} & -X^T & Y \end{bmatrix} \begin{bmatrix} \frac{\partial\bar{H}}{\partial q} \\ \frac{\partial\bar{H}}{\partial\alpha_1} \\ \frac{\partial\bar{H}}{\partial\alpha_2} \end{bmatrix} + \begin{bmatrix} 0 \\ S_1^{-T} \\ S_2^{-T} \end{bmatrix} u \\ y &= \begin{bmatrix} 0 & S_1^{-1} & S_2^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial\bar{H}}{\partial q} \\ \frac{\partial\bar{H}}{\partial\alpha_1} \\ \frac{\partial\bar{H}}{\partial\alpha_2} \end{bmatrix} \end{aligned} \tag{5.6}$$

where Y is skew-symmetric, subscripts 1 and 2 denote the first and other components, respectively, and where the Hamiltonian can be written as

$$\bar{H}(q, \alpha_1, \alpha_2) = \frac{1}{2}\alpha_1^T\bar{M}_1^{-1}\alpha_1 + \frac{1}{2}\alpha_2^T\bar{M}_2^{-1}\alpha_2 + V(q) \tag{5.7}$$

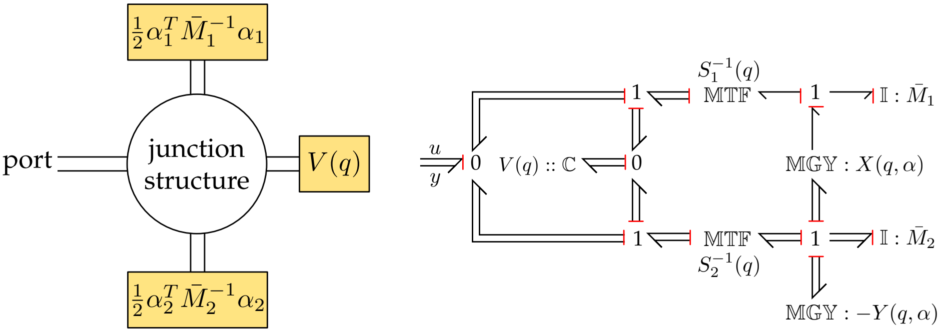

$\frac{1}{2}\alpha_1^T\bar{M}_1^{-1}\alpha_1$
port
junction structure
$V(q)$
$\frac{1}{2}\alpha_2^T\bar{M}_2^{-1}\alpha_2$
$S_1^{-1}(q)$
1
MTF
1
$\mathbb{I} : \bar{M}_1$
u
y
0
$V(q) :: \mathbb{C}$
0
$\mathbb{MGY} : X(q,\alpha)$
1
MTF
1
$\mathbb{I} : \bar{M}_2$
$S_2^{-1}(q)$
$\mathbb{MGY} : -Y(q,\alpha)$

Figure 5.3: Graphical representation of the system (5.5) with Hamiltonian (5.4) as an abstract system (left) and a concrete bond graph (right).

The real benefit of representing the dynamics in the form (5.6) reveals itself when we look at its representation as a port-interconnection of energy-storing subsystems, as shown in Figure 5.3. Since the energy is the sum of three terms (5.7), each having its own state variables, we can think of the energy being stored in three separate buffers: one for the potential energy (with state q), one for the kinetic energy in the direction of w (with state α_1), and one for the other kinetic energy (with state α_2). The internal energy flow between the three buffers, and the energy flow from the input port (u, y) is described by the power-continuous junction structure described by (5.6), which can be directly translated to the bond graph of Figure 5.3.

The bond graph and the equations (5.6) show how the energy flows between the different parts of the network. In particular, it explicitly shows how well the system is performing its control task: if there is perfect tracking of the vector field w, then α_2 (and hence the energy in the corresponding $\mathbb{I}$-element) is zero. In this way, the control task (and the control performance) is explicitly encoded into the system description. It is almost trivial to derive a tracking controller from this system description, as will be shown in the next section.

Example 5.2. As a simple example, we consider the motion of a point mass m in the plane, shown in Figure 5.4. The dynamic equations of the point mass can be written as (5.3) with the mass matrix being the 2×2 diagonal matrix $M = mI$.

As the desired vector field $w(q)$, we choose a vector field describing all the circles around the origin, *i.e.*

$$w(q) = \begin{bmatrix} -q^2 \\ q^1 \end{bmatrix} \tag{5.8}$$

This vector field is zero for $q = 0$, and hence we only consider points in $\mathcal{D} = Q \setminus 0$.

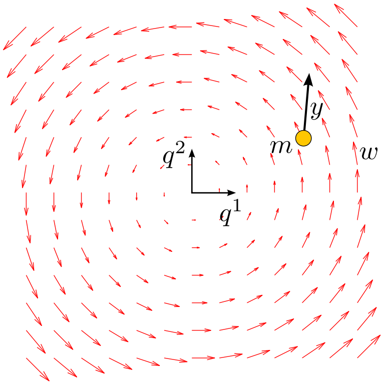


Figure 5.4: Two dimensional motion of a point mass with velocity v, and a vector field w describing the desired motion.

From this vector field, we can find a suitable coordinate transformation $S^T(q)$ as

$$S^T(q) = \frac{1}{|q|}\begin{bmatrix} -q^2 & q^1 \\ q^1 & q^2 \end{bmatrix} \tag{5.9}$$

in which the first column describes the direction along the vector field, and the second column describes the direction perpendicular (in the metric mI) to the vector field. The second column, in fact, describes motion along lines through the origin. The mass matrix in new coordinates can be computed as

$$\bar{M} = \left(SM^{-1}S^T\right)^{-1} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \tag{5.10}$$

which is diagonal and independent of q, so indeed, the two columns of S^T are orthogonal and have constant norm, as required. With the transformation (5.9), we can compute the dynamic equations (5.5) in new coordinates (q, α) as

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} q^1 \\ q^2 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & -\frac{\alpha_1}{|q|} \\ 0 & 1 & \frac{\alpha_1}{|q|} & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \frac{\alpha_1}{m} \\ \frac{\alpha_2}{m} \end{bmatrix} + \frac{1}{|q|}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ -q^2 & q^1 \\ q^1 & q^2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} y^1 \\ y^2 \end{bmatrix} &= \frac{1}{|q|}\begin{bmatrix} 0 & 0 & -q^2 & q^1 \\ 0 & 0 & q^1 & q^2 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ \frac{\alpha_1}{m} \\ \frac{\alpha_2}{m} \end{bmatrix} \end{aligned} \tag{5.11}$$

These equations describe the same system with input forces u_1 and u_2 acting in the x and y direction, but just expressed in different coordinates that show explicitly (through the coordinate α_1) how much the point mass is moving along the vector field w. In particular, when no forces are acting on the system, the mass moves in a straight line with constant kinetic energy. For example, when $u = 0$ and $\alpha_1 = 0$ (the mass is moving along a line through the origin), then $\dot{\alpha} = 0$, and hence α_1 remains zero: the systems keeps moving along the same line through the origin. △

5.2.2 Port-based asymptotic control

With the system represented in new coordinates, we now derive a controller that asymptotically stabilizes it to motion along the vector field $w(q)$. We initially assume the potential energy to be zero and deal with kinetic energy only. Afterwards, we discuss how to handle potential energy.

The controller we derive is a cascade port-interconnection of sub-controllers, each with a specific goal in terms of energy flows. The controllers are represented in port-Hamiltonian form and graphically as a bond graph.

Input transformation and decoupling

As the first part of the controller, we propose a power-continuous controller with the goal to decouple the two kinetic energy storage elements (with states α_1 and α_2) such that no energy can flow between them. Furthermore, since we want to build the controller as a cascade port-interconnection, it provides two new input ports, one for each storage element.

Theorem 5.3 (Decoupling control). *For the mechanical system described by (5.3) or in transformed coordinates by (5.5) with $V(q) = 0$, the following controller is power-continuous and, for $\bar{u} = 0$, keeps the kinetic energy separated in the two storage elements as defined by S^T.*

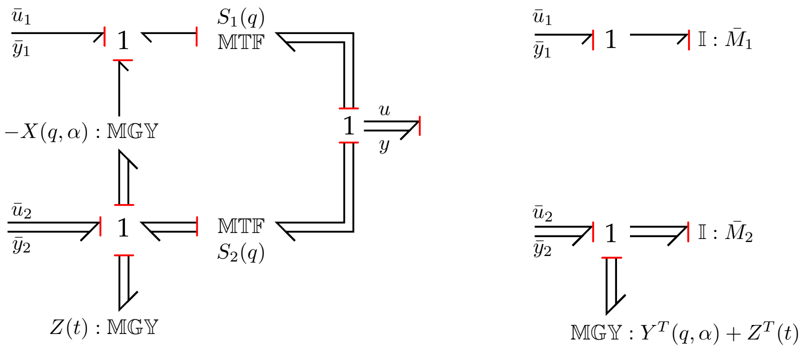
$$\begin{bmatrix} u \\ \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \begin{bmatrix} S_1^T X S_2 - S_2^T X^T S_1 + S_2^T Z S_2 & S_1^T & S_2^T \\ -S_1 & 0 & 0 \\ -S_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} -y \\ \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} \tag{5.12}$$

In this expression, Z is any skew-symmetric matrix, and $(\bar{u}_1, \bar{y}_1)$ and $(\bar{u}_2, \bar{y}_2)$ are new control ports, one connected directly to each energy storage element.

Proof. To prove power-continuity, we compute the power P_{in} going into the controller as well as the power P_{out} coming out:

$$P_{\text{in}} = \bar{u}_1^T \bar{y}_1 + \bar{u}_2^T \bar{y}_2 = \bar{u}_1^T S_1 y + \bar{u}_2 S_2 y \tag{5.13}$$

$$P_{\text{out}} = u^T y = y^T \left(S_2^T X S_1 - S_1^T X^T S_2 - S_2^T Z S_2 \right) y + \bar{u}_1^T S_1 y + \bar{u}_2^T S_2 y \tag{5.14}$$



(a) Bond graph of the controller (5.12). (b) Interconnected system.

Figure 5.5: Bond graphs of the decoupling controller (5.12) and its port-interconnection with the plant of Figure 5.3 with $V(q) = 0$.

which are equal, since the first term of (5.14) is zero by skew-symmetry, proving power-continuity.

To prove the energy separation property, we can compute the interconnected system as

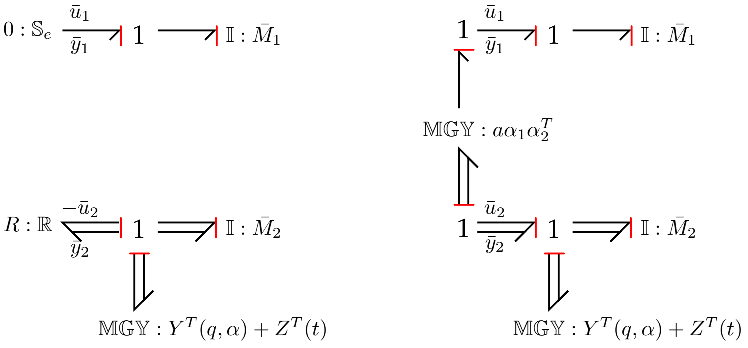
$$\frac{d}{dt}\begin{bmatrix} q \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0 & S_1^{-1} & S_2^{-1} \\ -S_1^{-T} & 0 & 0 \\ -S_2^{-T} & 0 & Y+Z \end{bmatrix} \begin{bmatrix} \frac{\partial \bar{H}}{\partial q} \\ \frac{\partial \bar{H}}{\partial \alpha_1} \\ \frac{\partial \bar{H}}{\partial \alpha_2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix}$$
$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \frac{\partial \bar{H}}{\partial q} \\ \frac{\partial \bar{H}}{\partial \alpha_1} \\ \frac{\partial \bar{H}}{\partial \alpha_2} \end{bmatrix} \tag{5.15}$$

Since $V(q) = 0$, we have $\partial \bar{H}/\partial q = 0$, so the equations for $\dot{\alpha}_{1,2}$ and $\bar{y}_{1,2}$ reduce to

$$\begin{aligned} \dot{\alpha}_1 &= \bar{u}_1 \\ \dot{\alpha}_2 &= (Y+Z)\bar{M}_2^{-1}\alpha_2 + \bar{u}_2 \\ \bar{y}_1 &= \bar{M}_1^{-1}\alpha_1 \\ \bar{y}_2 &= \bar{M}_2^{-1}\alpha_2 \end{aligned} \tag{5.16}$$

which shows that indeed the two storage elements with states α_1 and α_2 are decoupled, and the two ports $(\bar{u}_1, \bar{y}_1)$ and $(\bar{u}_2, \bar{y}_2)$ act separately and directly on the two storage elements. The matrix Z can be any skew-symmetric time-varying matrix and can be used, for example, to minimize actuator torques. □

Figure 5.5 shows the bond graph of the controller (5.12) as well as the port-interconnection of this controller to the plant (5.5) for $V(q) = 0$. The bond graph



(a) Damping injection. (b) Power-continuous transfer.

Figure 5.6: Two methods to achieve asymptotic tracking: by dissipating the energy in α_2, and by irreversibly transferring the energy from α_2 to α_1.

of the closed loop system can be seen to have no power-coupling (*i.e.* no bonds) between the two storage elements, hence the energy between the storage elements is really decoupled. This means that if there are no disturbances, and if the system is following the vector field ($\alpha_1 \neq 0$, $\alpha_2 = 0$), then the system will keep following the vector field. This shows that the controller of Theorem 5.3 provides nominal curve tracking.

Asymptotic convergence

The next part of the controller has the purpose of making the tracking behavior asymptotic, *i.e.* extending the nominal (marginal) tracking behavior to a behavior that attracts motions towards the curve. From the bond graph of Figure 5.5 and the interpretation of tracking in terms of two energy storages, this clearly means that α_2 (and its associated kinetic energy) should be driven asymptotically to zero.

A first naive option would be to simply dissipate this energy to drive it to zero, *i.e.* by so-called damping injection. In bond graph terms, this means attaching an $\mathbb{R}$ element to the port $(\bar{u}_2, \bar{y}_2)$ and zero force to the port $(\bar{u}_1, \bar{y}_1)$, as shown in Figure 5.6(a). This controller can then be written in port-Hamiltonian form as

$$\begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} -\bar{y}_1 \\ -\bar{y}_2 \end{bmatrix} \qquad (5.17)$$

with $R > 0$. It can be shown trivially that this controller is passive (because of the dissipative element), and that whenever $\alpha_2 \neq 0$, it decreases monotonically to zero since R is strictly positive definite. Hence, indeed this results in asymptotic tracking.

Unfortunately, all kinetic energy associated with α_2 is dissipated by the controller, which is not a very efficient way to deal with this energy. Instead, we can construct a (power-continuous) controller that transfers the energy from the α_2 storage to the α_1 storage in an irreversible way, for example as

$$\begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} 0 & -a\alpha_1\alpha_2^T \\ a\alpha_2\alpha_1^T & 0 \end{bmatrix} \begin{bmatrix} -\bar{y}_1 \\ -\bar{y}_2 \end{bmatrix} \tag{5.18}$$

for some parameter $a > 0$. The power balance for this controller can be shown to give $P_{\text{in}} = P_{\text{out}}$, proving that this controller is power-continuous. More interestingly, we can compute the change of the kinetic energy in the two storage elements when this controller is connected:

$$\frac{d}{dt}\left(\frac{1}{2}\alpha_1^T \bar{M}_1 \alpha_1\right) = a\left(\alpha_1^T \bar{M}_1 \alpha_1\right)\left(\alpha_2^T \bar{M}_2 \alpha_2\right) \tag{5.19}$$

$$\frac{d}{dt}\left(\frac{1}{2}\alpha_2^T \bar{M}_2 \alpha_1\right) = -a\left(\alpha_1^T \bar{M}_1 \alpha_1\right)\left(\alpha_2^T \bar{M}_2 \alpha_2\right) \tag{5.20}$$

which shows that whenever both α_1 and α_2 are nonzero, the energy in the α_2 storage will decrease, and the energy in the α_1 storage will increase. So, if the initial α_1 is nonzero (*i.e.* if the system is moving at least a little bit in the desired direction), then it will converge asymptotically to motion along $w(q)$ while keeping the total kinetic energy in the system constant. Figure 5.6(b) shows the bond graph of the controller connected in cascade to the system of Figure 5.5(b).

Remark. The nonlinear transformer described by (5.18) implements irreversible transformation of energy from one side to the other. It is irreversible, since no matter what port variables are applied to the two sides, power can only flow from one side to the other, and never back. In this respect, it is very similar to the $\mathbb{RS}$ element (Thoma 1975) in bond graphs, which describes the irreversible transduction of energy to the thermal domain, for example to describe the (incorrectly named) dissipation in an electrical resistor due to current flow. △

Remark. The particular controller parameterization (5.18) gives slow convergence, because it is quadratic in α_2: as α_2 approaches zero, the control force approaches zero even faster. This can be improved, for example, by replacing the parameter a by the expression

$$a \rightarrow \frac{a_0}{\sqrt{\alpha_2^T \bar{M}_2 \alpha_2} + a_1} \tag{5.21}$$

for $a_0 > 0$ and some small $a_1 > 0$. △

Example 5.4. We return to the planar point mass of Example 5.2 and compute the nominal and asymptotic controllers (5.12) and (5.18). With the choice for $S(q)$

as before, we obtain as the nominal controller

$$\begin{bmatrix} u_1 \\ u_2 \\ \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \frac{1}{|q|} \begin{bmatrix} 0 & \alpha_1 & -q^2 & q^1 \\ -\alpha_1 & 0 & q^1 & q^2 \\ q^2 & -q^1 & 0 & 0 \\ -q^1 & -q^2 & 0 & 0 \end{bmatrix} \begin{bmatrix} -y_1 \\ -y_2 \\ \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} \tag{5.22}$$

which ensures that the energy remains separated in the two energy reservoirs. In particular, when the point mass is moving along the vector field $w(q)$ ($\alpha_1 \neq 0$, $\alpha_2 = 0$), it can be shown that the control effort u for $\bar{u} = 0$ equals

$$u = -m \frac{|\dot{q}|^2}{|q|^2} \begin{bmatrix} q^1 \\ q^2 \end{bmatrix} \tag{5.23}$$

i.e. the well known expression for the centripetal force necessary for making a point mass m move along a circle at distance $|q|$ with velocity $|\dot{q}|$.

To obtain asymptotic tracking of the vector field, we add the asymptotic controller (5.18), which results in the cascaded controller that can be described as

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\alpha_1}{|q|} - \alpha_1\alpha_2 \\ \frac{\alpha_1}{|q|} + \alpha_1\alpha_2 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{5.24}$$

which is power-continuous by skew-symmetry of the matrix. This controller drives the point mass asymptotically to move along the circles defined by the vector field $w(q)$, always with constant kinetic energy. △

Dealing with potential energy

The combination of the nominal and asymptotic controllers discussed so far results in asymptotic convergence to the vector field w for the case of $V(q) = 0$. We now discuss how to deal with potential energy.

In general, the potential function V has no specific structure, unlike the kinetic energy, which is quadratic in the momenta. This is the main reason that in most research on passive controllers, the total potential energy is simply compensated for by simulating the inverse potential field in the controller and applying the corresponding potential forces to the system, thus effectively cancelling out the potential energy. Under certain conditions on the potential field and the measurements, this approach is passive (Ortega et al. 2001).

Although such an approach may be passive, it is generally not efficient, in the sense that the potential forces (*e.g.* gravity) can be quite large, and that hence large control forces are necessary to compensate for them. Furthermore, for the application of efficient walking, the trajectories computed in Chapter 4 are only efficient and natural in the presence of potential energy. After all, the natural

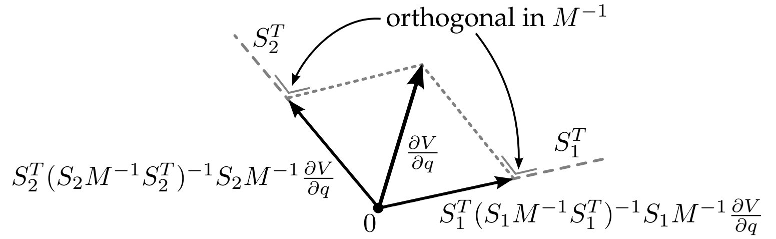

orthogonal in M^{-1}
S_2^T
S_1^T
$\frac{\partial V}{\partial q}$
$S_2^T(S_2M^{-1}S_2^T)^{-1}S_2M^{-1}\frac{\partial V}{\partial q}$
$S_1^T(S_1M^{-1}S_1^T)^{-1}S_1M^{-1}\frac{\partial V}{\partial q}$
0

Figure 5.7: Decomposition of the differential of V in the desired and undesired directions defined by S_1^T and S_2^T.

oscillatory motion of the legs is due to the power flow between kinetic energy and gravitational energy. If the gravitational energy is compensated for (*i.e.* the robot is effectively walking in zero g), then the computed trajectories are certainly not natural anymore.

Instead, we propose to compensate for only part of the potential field, namely the part that changes the undesired momentum α_2. We can compute this part by a projection, illustrated in Figure 5.7, namely the orthogonal projection (in the metric M^{-1}) of the differential of V onto the space of undesired momenta, described by the columns of S_2^T. This projection is described by the equation

$$u_{\text{pot}} = S_2^T(S_2M^{-1}S_2^T)^{-1}S_2M^{-1}\frac{\partial V}{\partial q} = S_2^T\bar{M}_2S_2M^{-1}\frac{\partial V}{\partial q} \tag{5.25}$$

where we used the definition of $\bar{M}$ from Section 5.2.1. The effect of this torque, together with the torque from the potential field, on the components α can be computed from (5.5) as follows.

$$\begin{aligned}
\frac{d}{dt}\alpha = \frac{d}{dt}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix} &= (\cdots) - S^{-T}\frac{\partial V}{\partial q} + S^{-T}u_{\text{pot}}\\
&= (\cdots) - S^{-T}(S^T\bar{M}SM^{-1})\frac{\partial V}{\partial q} + S^{-1}S_2^T\bar{M}_2S_2M^{-1}\frac{\partial V}{\partial q}\\
&= (\cdots) - S^{-T}\left(S_1^T\bar{M}_1S_1 + S_2^T\bar{M}_2S_2\right)M^{-1}\frac{\partial V}{\partial q} + \begin{bmatrix}0\\ \bar{M}_2S_2M^{-1}\frac{\partial V}{\partial q}\end{bmatrix}\\
&= (\cdots) - \begin{bmatrix}\bar{M}_1S_1M^{-1}\frac{\partial V}{\partial q}\\ \bar{M}_2S_2M^{-1}\frac{\partial V}{\partial q}\end{bmatrix} + \begin{bmatrix}0\\ \bar{M}_2S_2M^{-1}\frac{\partial V}{\partial q}\end{bmatrix}\\
&= (\cdots) - \begin{bmatrix}\bar{M}_1S_1M^{-1}\frac{\partial V}{\partial q}\\ 0\end{bmatrix}
\end{aligned} \tag{5.26}$$

So indeed, with the additional control law (5.25), the effect of the potential energy on α_1 is not changed, but the effect on α_2 is cancelled out. Furthermore, if the vector field describes a natural efficient trajectory for the system with potential energy, then the sum of the nominal control action (5.12) and the partial potential energy compensation (5.25) should be (close to) zero, and hence the total actual torque that is computed and applied by the motors is (close to) zero as well. This will be shown in the application to walking robots in Section 5.2.3.

One problem in this approach remains: the control law (5.25) is generally not passive, *i.e.* it is possible for a finite energy disturbance to extract infinite energy from the controller. This aspect needs to be taken into account in the higher-level energy management, discussed next.

High-level energy management

In addition to the direct low-level controllers of the previous sections, a higher-level controller is necessary to ensure that the energy balance is held at a desired level. This approach is similar to the Intrinsically Passive Control (IPC) plus supervisory control discussed by Stramigioli (2001). With the exception of the partial potential energy compensation, the controllers discussed so far are power-continuous and hence intrinsically passive. The energy management controller discussed here is a form of supervisory control, in that it is not intrinsically passive but can add energy to the system.

Energy management is required for several reasons. A first reason is to compensate for any energy introduced by the non-passive potential energy controller, hence to ensure stability of the closed-loop system. A second reason is to compensate for friction and other losses, which would (in the case of only power-continuous controllers) reduce the energy in the system to zero and make it stop. Finally, a third and very important reason, especially in the context of following efficient trajectories, is to ensure that the system is moving along the trajectories at the speed prescribed in the optimization routine. Not only simply because the control goal is to move at this speed, but also because the trajectories are only efficient (or optimal) when they are being followed at a specific speed.

When the desired, possibly time varying, energy level E_{des} is known, an extra torque u_{energy} can be added to the system to regulate the mechanical energy towards E_{des}. This torque should be applied in the direction of the desired vector field w, in order not to increase the undesired momentum α_2. The magnitude of the correcting torque can be computed for example as a simple linear proportional controller

$$u_{\text{energy}} = K_p \left(E_{\text{des}} - \frac{1}{2} p^T M^{-1}(q) p - V(q) \right) w(q) \tag{5.27}$$

for some $K_p > 0$. This simple controller works fine for some applications, such as steady walking down a certain slope, as discussed in the next section. For more difficult tasks, such as walking uphill, in which case the desired energy level is constantly increasing, extra control action such as integral control may be needed to stabilize to this energy level.

5.2.3 Application to walking robots

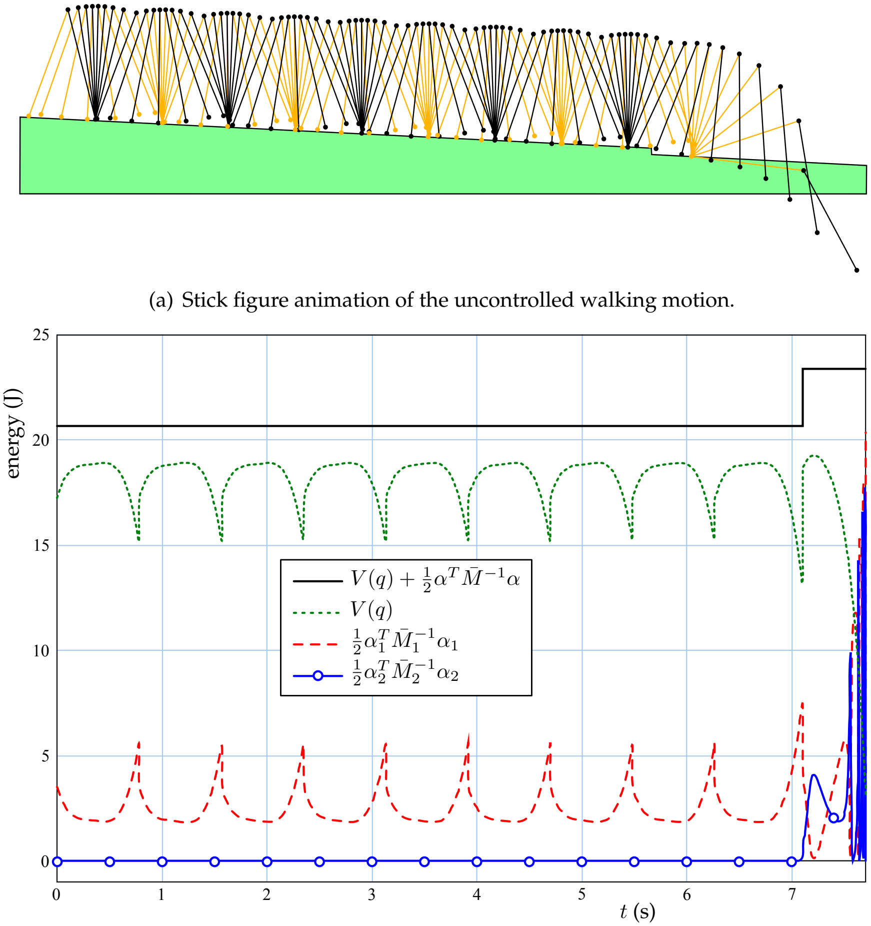
The results in this section have so far been formulated for general mechanical systems moving along general trajectories. We now apply them to the problem of walking robots, more specifically, to the compass-gait walker of Section 4.2 with a mass distribution equal to $m = 1$ kg and $m_H = 5$ kg. The results are an extended version of earlier work (Duindam & Stramigioli 2005*c*).

We already showed how the compass-gait walker can passively walk down a 3° slope if we start it at the initial conditions corresponding to the trajectory of Figure 4.8. This trajectory was shown to be stable, at least in the sense that the estimated initial conditions (using polynomial functions for the joint angles) converged to a limit cycle. The region of attraction of this passive motion is, however, not very large, as can be seen in Figure 5.8(a): when there is a disturbance in the floor (a 5 cm step down), the robot falls.

The goal of this section is to use the port-based tracking approach described before to increase the region of attraction of the passive walking cycle and hence increase the robustness against external disturbances. The controller should only act in case of disturbances, and otherwise just let the robot move along its natural passive walking trajectory. Thus, it is expected that the controlled system is still very efficient, and only uses actuator energy when a disturbance occurs. Of course, this only holds under the assumption that the actuators are back-driveable.

From the previous consideration, we can conclude that the desired behavior for the system should be a motion along the natural trajectory. This natural trajectory is taken as the computed trajectory of Figure 4.8, and it is shown as the solid half of the eight-figure in Figure 5.9(a). The dotted half is the motion of the angles q^3 and q^4 when the stance and swing legs are exchanged. As discussed in Chapter 4, we can use a coordinate relabeling to describe a walking cycle of two steps (one with each foot) as the concatenation of two equal single steps, once with and once without coordinate relabeling. We can use the same approach here in control, by computing the desired control behavior for half of a cycle and applying it either to one leg (ankle and hip) or to the other (ankle and negative hip), depending on which leg is on the ground. Thus, we can use only half of the cycle (the solid line) as the nominal curve Q_d, and construct a vector field towards this curve. When the swing foot impacts the ground, the (measured) coordinates are relabeled, and the computed torque is applied to the appropriate joints.

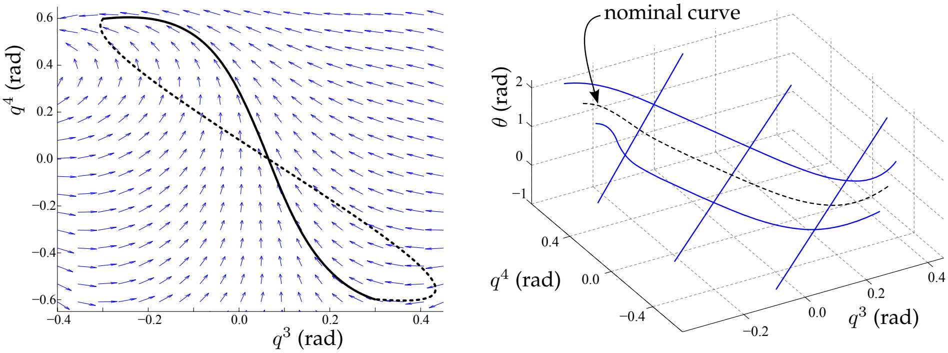
From this half of the trajectory, we compute a suitable vector field as follows, illustrated in Figure 5.9(b). Since the length of the vectors in the vector field is not important, it is sufficient to parameterize a vector at a point by only one variable;



(a) Stick figure animation of the uncontrolled walking motion.

(b) Evolution of the various energies during the walking motion.

Figure 5.8: Simulation of the compass-gait walker walking passively down a small step and falling.



(a) Nominal trajectory and vector field.

(b) Constructing curves for the vector field.

Figure 5.9: Construction of a suitable desired vector field $w(q)$ for downhill walking of the compass-gait walker. Based on the angle θ of the velocity vectors of a curve (half a walking cycle), a surface with other velocity angles is computed.

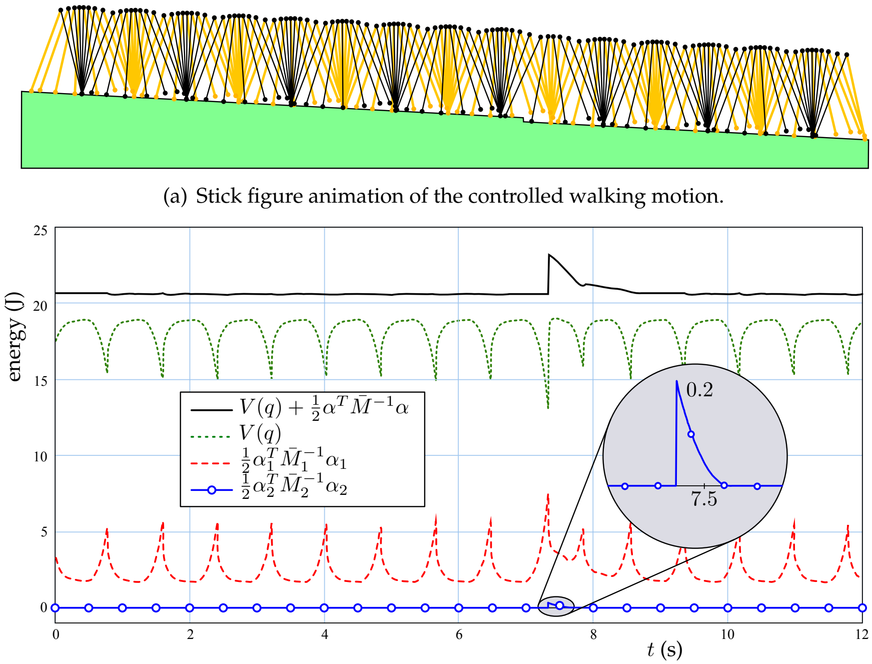
we take this variable to be the angle[1] θ between the vector and the vertically upward direction. We can then plot the velocity directions along Q_d as a curve in the three-dimensional (q^3, q^4, θ) space (the dotted line in the figure). From this curve, we can construct other curves roughly parallel to it, and we take the desired velocity to be roughly parallel and towards Q_d. This means that for points to the right of Q_d, θ is larger, and for points to the left of Q_d, θ is smaller. We construct two parallel curves and three orthogonal curves, as shown in Figure 5.9(b). We then fit a fourth-order polynomial surface of the form

$$\hat{\theta}(q^3, q^4) = \sum_{i=0}^{4} \sum_{j=0}^{i} a_{ij} (q^3)^{i-j} (q^4)^j = a_{00} + a_{10} q^3 + a_{01} q^4 + \dots \tag{5.28}$$

with $\hat{\theta}$ the approximated angle of the velocity vector at (q^3, q^4), and a_{ij} the fifteen parameters describing the surface. We fit this surface to the data in the six reference curves, in such a way that the error between θ and $\hat{\theta}$ over all data points is smallest in the least square sense. The resulting surface describes a vector field, which is plotted in Figure 5.9(a). We use this vector field as the desired vector field $w(q)$. This approach can be generalized to higher dimensions, by constructing the desired vector $w(q)$ at q as the sum of the velocity vector at the closest point of Q_d and a vector directed towards this point on the curve.

Using this vector field and the orthogonal direction (in M), we can find a suitable coordinate transformation $S(q)$ that decomposes the momentum of the

[1] We implicitly and shamelessly use the Euclidean metric here in the definitions of angle, parallel, and orthogonal. Even though the Euclidean metric has no physical meaning in (q^3, q^4) space, it is easy to work with and provides good results.



(a) Stick figure animation of the controlled walking motion.

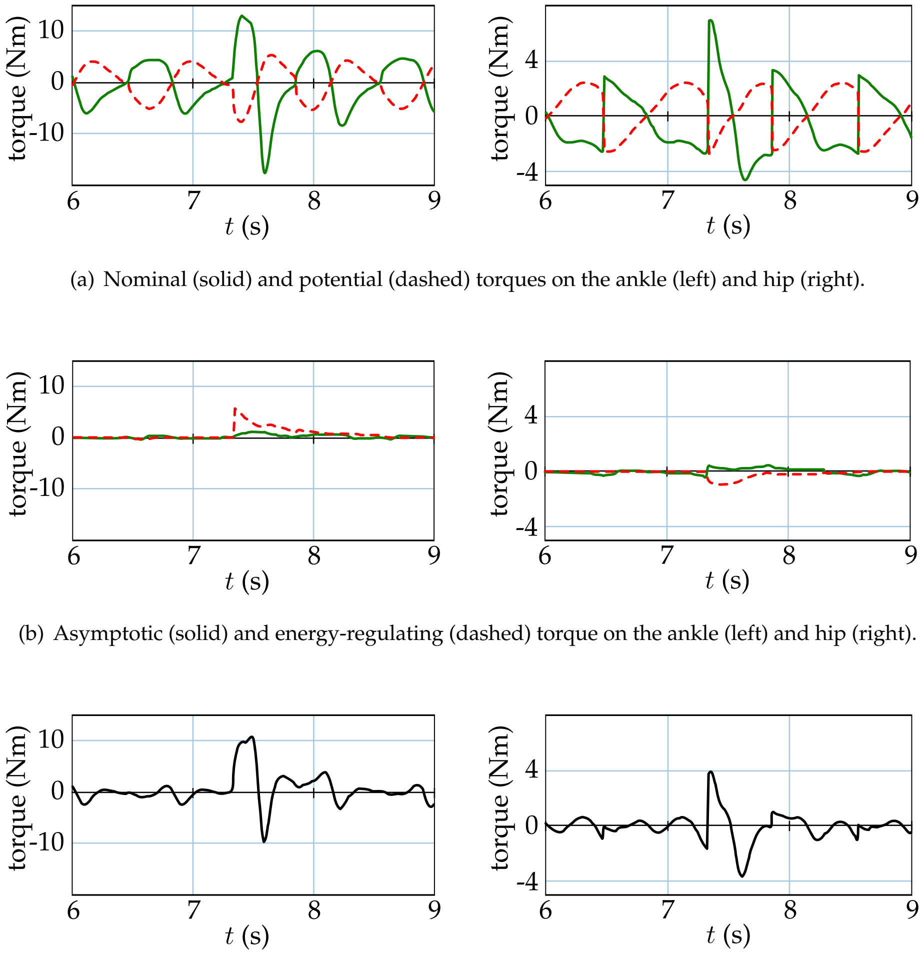
(b) Evolution of the various energies during the walking motion.

Figure 5.10: Simulation of the port-based tracking controller interconnected to the compass-gait walker, walking down a 5 cm step.

system in a desired component α_1 and an undesired component α_2. If we analyze the velocity components of the passive motion of Figure 5.8(a), we obtain that the α_2 component of the motion is roughly zero during nominal walking, and then becomes nonzero when the robot falls. This is to be expected, of course, since the passive trajectory was used as a reference to *define* the desired motion.

Figure 5.8(b) shows the decomposition of the energy of the system in potential energy and the two kinetic energies associated with α_1 and α_2. Here, the potential energy is taken to be equal for the beginning of each step, such that the total energy is constant during the walking motion (until the step down), instead of piecewise constant with steps down at each foot impact. This makes it easier to compare the different steps, and to implement the energy controller (5.27). During nominal walking, all kinetic energy is in the α_1 direction, while after the step down (the last second of the simulation), it oscillates wildly between the α_1 and α_2 directions.

From the transformation $S(q)$, we can compute the nominal controller (5.12), the asymptotic controller (5.18), the potential energy compensation (5.25), and



(a) Nominal (solid) and potential (dashed) torques on the ankle (left) and hip (right).

(b) Asymptotic (solid) and energy-regulating (dashed) torque on the ankle (left) and hip (right).

(c) Total control torque on the ankle (left) and hip (right).

Figure 5.11: Simulated control torques for the port-controlled walking motion of Figure 5.10, for $6 \leq t \leq 9$. The left figures show the various torques for the ankle joint, the right figure for the hip joint.

the energy management controller (5.27). As control parameters, we chose a proportional energy gain $K_p = 1.0$, and convergence parameter a as in (5.21) with $a_0 = 0.1$ and $a_1 = 0.001$. Figure 5.10 shows a simulation of the interconnection of this controller to the compass-gait walker, and Figure 5.11 shows the corresponding control torques.

With the port-based controller, the robot walks down the step without falling over. If the parameters K_p and a_0 are further tuned, steps up to roughly 11 cm (*i.e.* 11% of the leg length) can be taken without falling over. The uncontrolled walker can only walk down steps up to 2 cm. Hence, the controller really helps to increase the robustness of the walking motion to disturbances in the floor profile, and the performance is actually quite admirable: try walking down an unexpected 11 cm step with a blindfold on!

The reason for the limitation on performance (even though *e.g.* the point mass of Example 5.4 converges from any $q \neq 0$) is due to the limited conditions under which the simplified model is valid. In the controller design, the biped walker is assumed to behave like a two link manipulator fixed to the ground, and this assumption is only valid while the stance foot remains on the ground. In the case of large disturbances, *e.g.* in the ground profile, the control torques are large enough to cause a liftoff of the stance foot, which results in falling down.

If we look at the control torques in Figure 5.11, we see that the nominal torque and the partial potential-field compensating torque are the largest components, and that they mostly cancel each other out. This shows that, indeed, the potential forces are partially responsible for shaping the natural trajectory of the mechanism; without them, the controller would have had to supply the large nominal torques to keep the system on the trajectory.

Despite the nominal and potential torques canceling each other out, the required control torque u is still nonzero, even though the system is moving roughly on a natural passive trajectory, *i.e.* no torques should be required. The reason for the nonzero torques is that, although the vector field at each point is roughly directed along the natural curve or towards it, the integral curves of the vector field oscillate a little around the natural trajectory. So if the system perfectly follows the integral curves of the vector field, a small extra torque is needed to make this oscillation possible, and this oscillatory torque can be seen in the figures. The design of an accurate vector field is hence essential for low-torque requirements, and better methods than the crude least-squares approximation used here could improve the performance.

Finally, let us pose probably the most dreaded question for people in nonlinear control theory: why not use PID control? This is directly related to the earlier remark about limitations on performance due to the stance foot lifting off. During nominal walking, roughly along the reference trajectory, a PD controller tracking a time-varying setpoint performs very well, requiring very low torque (lower than for the port-based controller) to keep the system walking. This is not an amazing accomplishment, since the system by itself already remains close to the reference trajectory. However, when a step down occurs, the large position error causes the PD controller to respond with a large torque directed towards the setpoint. Many settings for the controller gains were tried, but either they were to small to stabilize the system, or too large and causing liftoff of the stance foot. This shows that, at least in this example, using a mild, energy-considerate controller that acts in the right direction works better than a traditional controller that applies a force directly towards the setpoint.

5.3 Planar stability using one actuator

The previous sections show how to compute efficient trajectories and control laws for fully actuated robots. Control laws such as these are very general and allow provable stability and asymptotic convergence. On the hand, many practical robots are not fully actuated. This can be for practical reasons (attaching motors on all degrees of freedom may not be feasible), but also because many walking robots can actually be controlled quite well using only a few actuators. However, many of the control laws for these underactuated systems are based on intuition, and may not be as general as the control laws for fully actuated systems. A notable exception is the class of robots with only one unactuated degree of freedom, for which some general approaches exist, see for example Shiriaev et al. (2005).

These last two sections of this chapter describe control laws for very specific underactuated walking robots. In this section, we consider the control of Dribbel, the experimental robot introduced in Section 4.3. The results described here are preliminary and mostly aimed at the conceptual testing of the experimental robot: whether it can walk at all, and how well the model developed in Section 4.3 resembles the real system. Most simulations and experimental results are based on work by Beekman (2004) and Dertien (2005).

In its current form, Dribbel has only one actuated joint (the hip), and it is hence severely underactuated. The power-continuous controller of Section 5.2 is designed for fully actuated systems, and adapting it to a system with one actuator is impossible: the only intrinsically power-continuous controller with a one-dimensional power-port is the controller $u = 0$. In addition, practical boundary conditions greatly reduce the design freedom: the stance leg is required to be kept straight, and the swing foot is required to have some ground clearance. This implies that the swing knee has to bend a decent amount during the swing phase (to provide ground clearance), but it also needs to stretch with enough speed before the end of the step (to lock it into place for the subsequent stance phase).

We propose a relatively simple hip controller that results in stable walking on a level floor. We show both simulation and experimental results.

5.3.1 Simulation of a stabilizing hip controller

We design the controller using the simulation model of Section 4.3, but extended with a model of the hip motor and reduction stage, and a compliant contact model (for easier simulation). Preliminary studies for a suitable motor were conducted, taking into account the maximum required torque and velocity, and the maximum allowed physical dimensions. Based on these studies, the choice was made to use the Maxon RE40 brushed DC motor, together with the Maxon GP42c gearbox. Detailed models of these components were taken from the 20sim motor library and added to the model of the mechanism. The maximum input current for the motor is 8 A, and hence any controller that is computed should require less than 8 A maximum current.

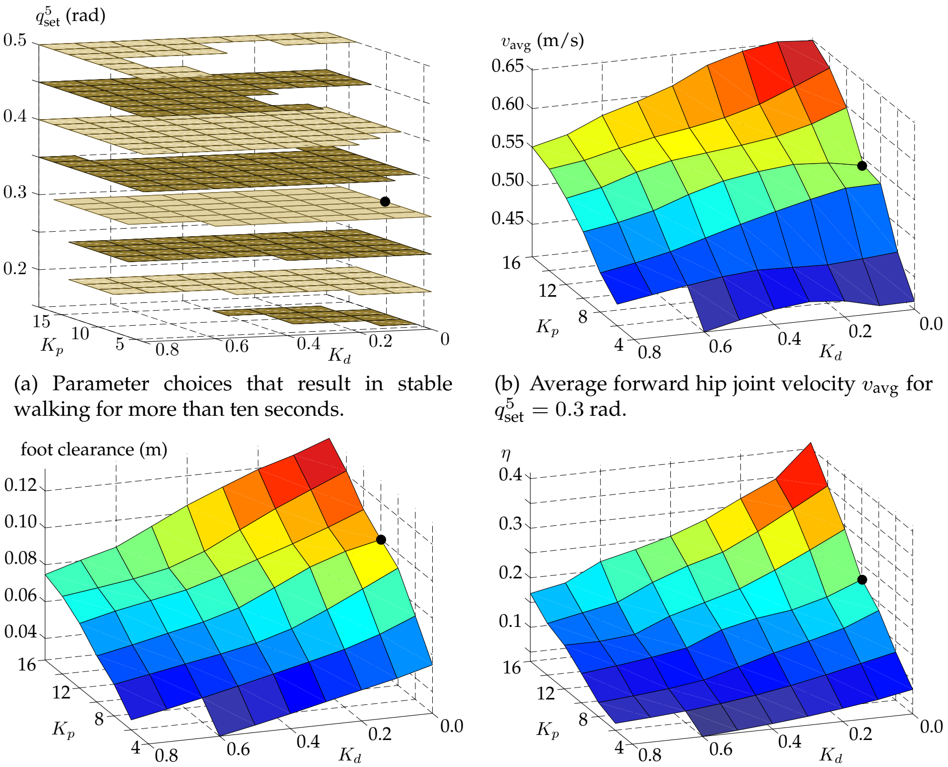
This extended model is used for controller simulation, to check whether the current remains within the limits, and to estimate the required mechanical power and efficiency. The *total* required power, including the power used for the electronics and the knee joint locking mechanism, cannot be estimated, since no models of the electronics or the knee locking actuator have been implemented.

As described before, the hip controller needs to ensure that the knee bends enough for ground clearance during the swing phase, but also stretches enough to lock into place for the stance phase. In addition, the hip controller should ensure that the forward velocity is large enough to prevent falling backwards. Finally, it should regulate the energy, such that the robot walks with a certain constant speed. As a first attempt to accomplish all this, we propose a simple PD controller of the form

$$i_{\text{motor}} = K_p(q^5_{\text{set}} - q^5) - K_d\dot{q}^5 \tag{5.29}$$

where q^5 is the hip angle, q^5_{set} is the setpoint for the hip angle, and $K_p, K_d \geq 0$ are controller gains. The magnitude of the setpoint q^5_{set} is a (constant) control parameter, but its sign is switched depending on what foot is on the ground. As soon as the swing foot touches the ground, the sign is reversed, such that the proportional control term produces a large control torque on the hip, causing the new swing leg to quickly swing forward while the inertia of the lower leg causes the swing knee to bend as required. Then, as the swing leg approaches the setpoint, the upper leg decelerates, and the swing knee locks back into place. The forward speed and energy can be regulated by changing the setpoint, similarly to the foot placement strategy discussed in Section 5.4.

To determine suitable control parameters, we perform simulations for various settings of the parameters, with setpoint q^5_{set} varying between 0.1 and 0.5, proportional gain K_p between 2 and 18, and derivative gain K_d between 0 and 0.5. The initial conditions for these simulations are the same for each simulation: the swing legs slightly backward, and a small initial forward velocity of the hip joint. The gait is considered stable if the robot does not fall within ten seconds of



(a) Parameter choices that result in stable walking for more than ten seconds.

(b) Average forward hip joint velocity v_{avg} for $q^5_{\text{set}} = 0.3$ rad.

(c) Average foot clearance of the swing foot for $q^5_{\text{set}} = 0.3$ rad.

(d) Specific cost of transport η for $q^5_{\text{set}} = 0.3$ rad.

Figure 5.12: Simulation results for a parameter sweep on the control parameters q^5_{set}, K_p, and K_d. The black dot indicates the chosen parameters for the simulation of Figure 5.13 and the experiment of Figure 5.15.

simulation time. During the last five seconds of the simulation (*i.e.* after possible transients have settled down), the following variables are measured: average forward hip velocity, average power consumption by the motor, and average foot clearance of the swing foot.

The results of the parameter sweep are shown in Figure 5.12. The first figure shows the region of control parameters which result in stable walking for at least ten seconds. Clearly, any choice of parameters should be inside this region, and not too close to the edge in order to allow for some modeling inaccuracies. Comparison of the power consumption for different setpoints q^5_{set} shows that the required power increases for increasing setpoints. We therefore choose the setpoint at $q^5_{\text{set}} = 0.3$ rad, such that it requires relatively little power, while still resulting

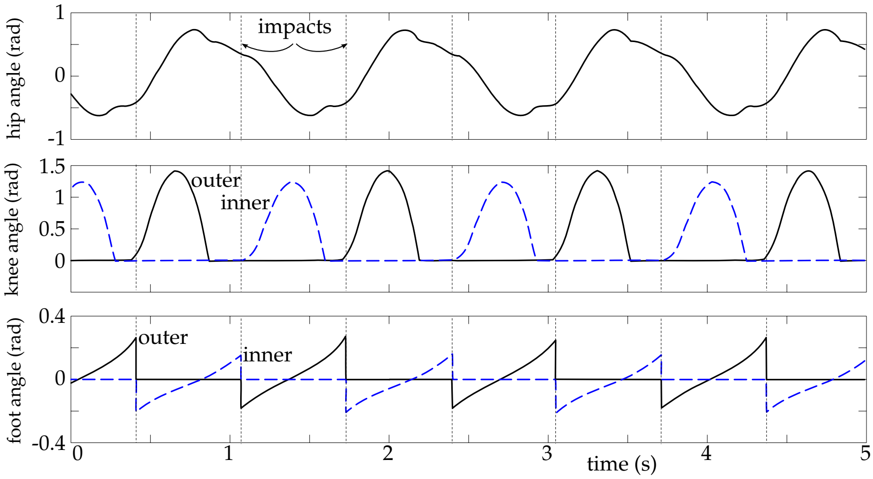


Figure 5.13: Simulation of the joint angles for steady state walking with control settings $K_p = 10$ A/rad, $K_d = 0$ C/rad, and $q^5_{\text{set}} = 0.3$ rad.

in stable walking for a broad range of controller gains K_p and K_d.

Figures 5.12(b)-(d) show the influence of the control gains on the average forward walking velocity, the average ground clearance, and the specific cost of transport η, which is computed as in (4.40)

$$\eta = \frac{P_{\text{mech}}}{mgv_{\text{avg}}} = \frac{P_{\text{mech}}}{9.2 \cdot 9.81 \cdot v_{\text{avg}}} \tag{5.30}$$

with $m = 9.2$ kg the total mass, and P_{mech} and v_{avg} the average mechanical power and velocity obtained from the simulation data. Note that P_{mech} includes the mechanical power dissipated by the actuator. The figures show that the mechanical cost increases for higher proportional gain and lower derivative gain, but that also the average foot clearance and forward velocity increase. For efficiency, the mechanical cost of transport should be small, while for stability, the average foot clearance should be large. We choose here the parameters $K_p = 10$ and $K_d = 0$, which give rise to a relatively low mechanical cost ($\eta = 0.22$) with still reasonable foot clearance. The peak current requirement is 7.5 A, which is still within the allowed range.

Figure 5.13 shows the joint angles corresponding to steady state walking for this choice of control settings. One of the interesting things to note in this figure is that the gait is not symmetric with respect to the two legs: the inner knee does not bend as much during the swing phase as the outer knee, and the step length (which is related to the stance foot angle) is shorter when the inner feet are on the ground than when the outer feet are on the ground. This asymmetry also

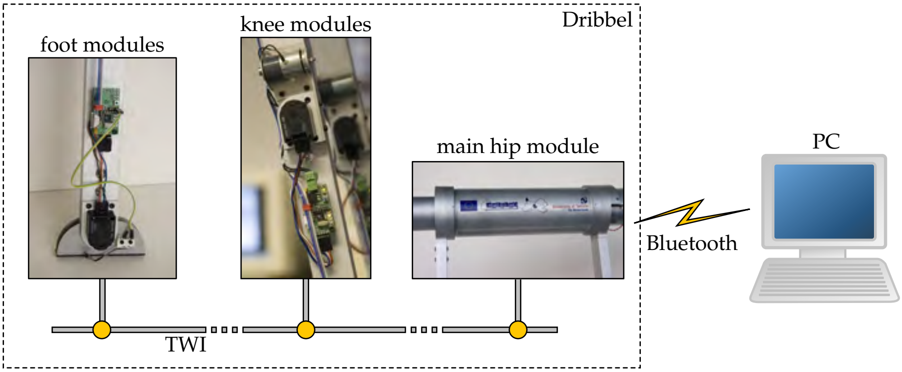


Figure 5.14: Communication channels in the Dribbel setup: two-wire interface (TWI) bus between the modules, and wireless Bluetooth to an external PC.

occurs for other control parameter values, and is due to a slight asymmetry in the definition of the controller: it determines the setpoint at -0.3 rad when only the outer feet are on the ground and +0.3 rad in all other cases (including the case that both the inner and outer feet are on the ground). Since the simulation model uses a compliant contact model, the double-stance phase is not instantaneous and hence the setpoint remains longer at +0.3 than -0.3, causing the slight asymmetry in the gait.

5.3.2 Experimental implementation and results

The electronics on Dribbel have been implemented as illustrated in Figure 5.14. Each joint of the robot features a small controller board that handles actuation (locking on the knees, motor control on the hip) and sensing (joint angles on all joints, contact switches on the feet). In addition, a larger control board has been inserted in the hip tube that handles computation of the steering value for the motor as well as communication with an external PC. Data is transferred between the modules through a two-wire interface (TWI) bus, and between the main control board and an external PC by wireless Bluetooth communication or a wired serial interface.

The PD controller (5.29) described in the previous section has been implemented on the main controller board, running on a 16 MHz ATMEL ATmega128 microcontroller. The control loop (sense, compute, actuate) runs at 100 Hz, while measurement data (such as joint angles and foot switch states) is transferred to the external PC at roughly 25 Hz. The setpoint and gains of the PD controller can be adjusted from the external PC through a terminal interface.

Figure 5.15 shows the *measured* joint angles for the control parameters as set

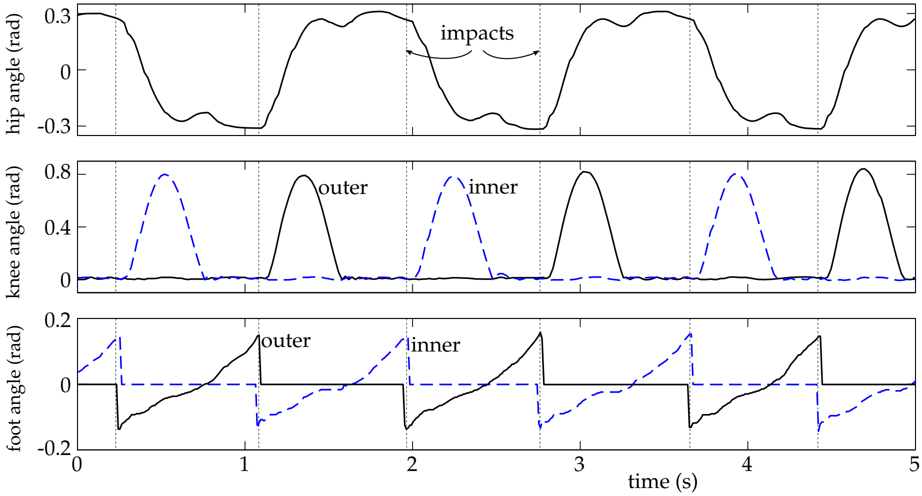


Figure 5.15: Measured joint angles during experimental walking with control settings $K_p = 10$ A/rad, $K_d = 0$ C/rad, and $q^5_{\text{set}} = 0.3$ rad.

in the *simulations* of Figure 5.13, *i.e.* $q^5_{\text{set}} = 0.3$, $K_p = 10$, and $K_d = 0$. The figure shows the measured hip and knee angles, as well as the measured foot angles during stance phase (the foot angles are undefined and set to zero during the swing phase). With these parameters, Dribbel can walk stably, but is very sensitive to initial conditions and floor flatness.

Power consumption was measured in standby mode (with the motor turned off), as well as during walking. The electronics boards consume roughly 11.2 W in standby mode, while the knee locking magnets each consume 3.74 W when active. During a walking cycle, the two stance knees are active constantly, and the swing knees are active only 60% of the time. Together, this results in an average power consumption by the electronics and knees of 23.2 W. The total average power consumption during walking is 30.0 W, which means that the average mechanical power consumption is 6.8 W. From the power consumption and the average velocity (about 0.32 m/s), we estimate the specific cost of transport (5.30) to be $\eta = 0.23$.

If we compare the measurements with the simulation, we see that both the knee, hip, and foot angles are smaller in practice than in the simulation. The hip angle in the experiment nicely damps towards the setpoint, while the hip angle in the simulation oscillates with a large overshoot. This can be explained by friction in the joints of the experimental robot, which is not modeled in the simulation. Another likely cause is the non-idealness of the mechanism, meaning that on impact of the feet, internal vibrations in the links and mechanical play in the joints result in higher energy loss than for an ideal rigid mechanism. This would account for the lower joint velocities, and the tendency for the robot to fall backward occasionally. Accurate velocity measurements on the joints could be used to compute the difference between the actual velocities after impact and the ideal velocities in case of a fully rigid impact (using the velocity projection discussed before). Finally, even though both the measured mechanical power and velocity are lower than the simulated ones, the specific cost of transport is very close: 0.23 in practice versus 0.22 in simulation.

It may come as a surprise that the practical cost of transportation is only slightly higher than the simulated cost, even though the experimental setup suffers from dissipation not included in the simulation model. The main reason for the relatively low cost in the experiment is the lower walking speed (0.32 m/s in practice versus 0.54 m/s in simulation). Lower speeds tend to require much less power than higher speeds. In fact, simulations show that the specific cost of transport can actually be decreased to zero by decreasing the speed towards zero: even though the speed appears in the denominator of (5.30), the required power for a certain speed decreases to zero faster. This is clear if we consider only the energy lost on impact: as shown by (3.45), the kinetic energy loss on impact is quadratic in the joint velocities, whereas the walking speed is linear in the joint velocities. Hence, if the joint velocities decrease to zero linearly (because of increasingly slower walking), the energy loss decreases to zero quadratically, and hence the specific mechanical cost converges to zero. This effect leads to the question whether the specific cost of transport is a good measure for comparison for robots walking at different speeds.

Overall, the comparison demonstrates that the simulation and experiment show similar behavior, but that more detailed models of the dissipation are required to make accurate predictions about angle trajectories. Still, the simulated model has been used to successfully predict controller values that result in stable experimental walking, which, at this point, was the main goal. Future work should adapt the model parameters to match the experimental results, such that a more precise prediction can be made and numerical optimization of the control parameters becomes meaningful.

5.4 3D stability by foot placement

As the final topic of this chapter, we consider the control of the three-dimensional robot of Section 4.4, but we assume all mass to be concentrated at the hip joint. This means that the motion of the trunk has no influence on the motion of the rest of the walker, and hence we leave it out. It also means that the swing leg only influences the walking motion by determining the time and position of the next impact with the ground; its motion between liftoff and touchdown does not influence the motion of the mass at the hip joint.

We first describe a model of this simplified system, and then discuss how *foot*

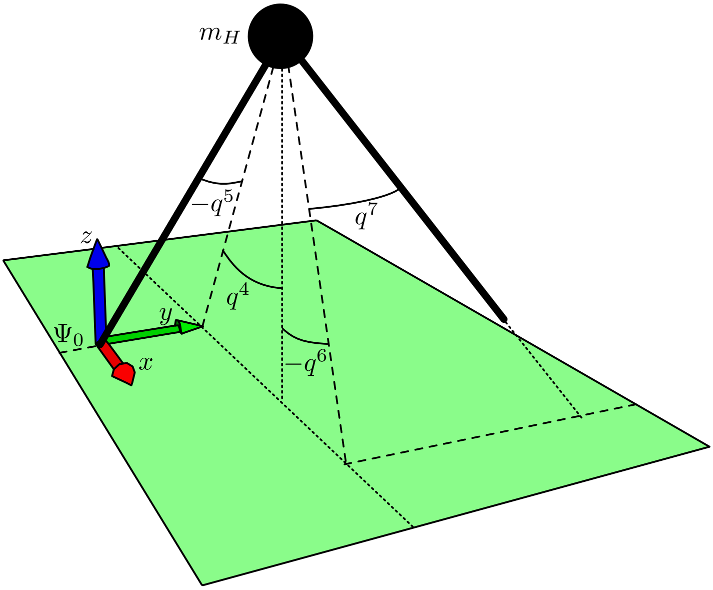


Figure 5.16: Control model of a simple three-dimensional walker with coordinates q^4 through q^7 and mass m_H concentrated at the hip.

placement can be used as a control method to stabilize it and to perform motion planning. The results in this section are largely based on the work of van Oort (2005), but reformulated in the port-Hamiltonian framework and extended towards motion planning.

5.4.1 Simplified model of the 3D walker

Figure 5.16 shows a representation of the simplified walker, using the coordinate choices of Figure 4.16, but leaving out the trunk. As indicated, we no longer assume the robot to be fully actuated, but instead control it by two means. First, we control the swing leg in order to control the time and location of the next impact, *i.e.* we implement a form of foot placement control. In addition, we compensate for the energy loss on impact by pushing off the stance leg just before touch-down of the swing leg. Without loss of generality, we assume the leg with coordinates q^4 and q^5 to be the stance leg, and the leg with coordinates q^6 and q^7 to be the swing leg. The results can be applied directly to the mirrored case, by using the coordinate relabeling (4.34).

The first control layer (foot placement) requires that we define control inputs on the position of the swing leg, *i.e.* control inputs on q^6 and q^7. Since the swing leg is massless, we cannot use torque inputs collocated with these angles, but instead, we use direct velocity input. This results in the following set of equations describing the dynamics of the walker:

$$\frac{d}{dt}x := \frac{d}{dt}\begin{bmatrix} q^4 \\ q^5 \\ q^6 \\ q^7 \\ p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \frac{\partial H}{\partial x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \frac{\partial H}{\partial x} \tag{5.31}$$

with energy function equal to

$$H(x) = \frac{1}{2}\begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} m_H \cos^2(q^5) & 0 \\ 0 & m_H \end{bmatrix}^{-1} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} + m_H g \cos(q^4)\cos(q^5) \tag{5.32}$$

The dynamics of the simplified walker can hence again be formulated as a port-Hamiltonian system, with input ports (u_1, y_1) and (u_2, y_2) to control the swing leg. Since the Hamiltonian does not depend on q^6 and q^7, the output variables y_1 and y_2 (the forces collocated with the input velocities u_1 and u_2) are zero at all times.

The second control layer (instantaneous push off) can be implemented together with the impact equations. Since we consider walking on level ground, and since the control ports in (5.31) do not change the energy of the system, the impact losses during walking will monotonously reduce the energy of the walker until it falls over. So, additional energy input is required. We include this by allowing the stance leg to instantaneously push off just before the impact of the swing leg. This idea is an approximation of the push-off actions of humans, who can flex their stance foot just before impact of the swing foot, thus adding mechanical energy to the walking cycle.

To illustrate the idea of the approach, consider the planar case, depicted in Figure 5.17. The velocity v_- of the hip mass before impact is tangent to a circle around the stance foot. Due to the collision, this velocity will be projected[2] to a velocity tangent to a circle around the new stance foot. Since it is a projection, it reduces the velocity of the mass, and hence reduces its kinetic energy. If, however, we inject kinetic energy by means of adding an instantaneous push-off velocity v_{po} in the direction of the stance leg, we change the magnitude and direction of the velocity vector to v_i, such that the final resulting velocity v_+ after projection is as desired. Here, we search for the push-off velocity such that the kinetic energies before and after push-off and impact are exactly equal, but the results can be generalized to regulate the energy to some other desired level.

[2] Since the dynamics of the walker are essentially those of a (constrained) point mass, we can think of this projection in the usual Euclidean way.

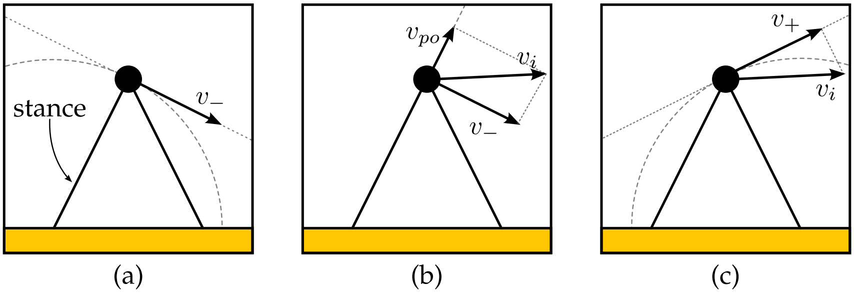


Figure 5.17: Just before collision of the swing foot (a), the velocity v_- of the hip is along a circle around the stance foot. Then (b), the stance leg pushes off, and increases this velocity by v_{po}. Finally (c), the velocity is projected by the collision to a vector v_+ along a circle around the new stance foot.

5.4.2 Energy conservation by ankle push-off

To compute the required push-off velocity v_{po} as well as the new velocity v_+ after push-off and impact, we first describe the position and velocity of the hip mass in polar coordinates around the stance foot. This coordinate transformation relates the coordinates (x_h, y_h, z_h) of the hip expressed in Ψ_0 to the angles q^4, q^5, and the leg length r as follows.

$$\begin{bmatrix} x_h \\ y_h \\ z_h \end{bmatrix} = \begin{bmatrix} r\sin(q^4)\cos(q^5) \\ -r\sin(q^5) \\ r\cos(q^4)\cos(q^5) \end{bmatrix} \tag{5.33}$$

from which we find the velocity of the hip mass in linear coordinates as

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{z}_h \end{bmatrix} = \begin{bmatrix} r\cos(q^4)\cos(q^5) & -r\sin(q^4)\sin(q^5) & \sin(q^4)\cos(q^5) \\ 0 & -r\cos(q^5) & -\sin(q^5) \\ -r\sin(q^4)\cos(q^5) & -r\cos(q^4)\sin(q^5) & \cos(q^4)\cos(q^5) \end{bmatrix} \begin{bmatrix} \dot{q}^4 \\ \dot{q}^5 \\ \dot{r}_- \end{bmatrix} \tag{5.34}$$

Instead of polar coordinates around the (old) stance foot, we can also express the hip velocity in polar coordinates around the new stance foot, *i.e.* the swing foot.

$$\begin{bmatrix} \dot{x}_h \\ \dot{y}_h \\ \dot{z}_h \end{bmatrix} = \begin{bmatrix} r\cos(q^6)\cos(q^7) & -r\sin(q^6)\sin(q^7) & \sin(q^6)\cos(q^7) \\ 0 & -r\cos(q^7) & -\sin(q^7) \\ -r\sin(q^6)\cos(q^7) & -r\cos(q^6)\sin(q^7) & \cos(q^6)\cos(q^7) \end{bmatrix} \begin{bmatrix} \dot{q}^6 \\ \dot{q}^7 \\ \dot{r}_+ \end{bmatrix} \tag{5.35}$$

For non-singular configurations ($r \neq 0$ and $q^5, q^7 \neq 0$), the coordinate transformations are invertible, and hence we can express the velocity in polar coordinates in terms of the velocity in linear coordinates.

The radial velocity $\dot{r}_-$ before impact is precisely the magnitude of the push-off velocity v_{po}, and the velocity projection due to impact just means setting $\dot{r}_+$ to zero. Since $\dot{r}_-$ is only nonzero just before impact, and since $\dot{r}_+$ is zero immediately after impact, the leg length equals $r = 1$ both before and after impact ($\dot{r}$ is essentially a finitely high pulse with vanishing width, hence zero integral). The final velocity after both push-off and impact can be computed by the sequence

$$\begin{bmatrix}\dot{q}^4\\ \dot{q}^5\\ 0\end{bmatrix} \xrightarrow{\text{push off}} \begin{bmatrix}\dot{q}^4\\ \dot{q}^5\\ \dot{r}_-\end{bmatrix} \xrightarrow{(5.34)} \begin{bmatrix}\dot{x}_h\\ \dot{y}_h\\ \dot{z}_h\end{bmatrix} \xrightarrow{\text{inverse (5.35)}} \begin{bmatrix}\dot{q}^6\\ \dot{q}^7\\ \dot{r}_+\end{bmatrix} \xrightarrow{\text{impact}} \begin{bmatrix}\dot{q}^6\\ \dot{q}^7\\ 0\end{bmatrix} \tag{5.36}$$

The kinetic energy increases during push-off by an amount $\frac{1}{2}m_H\dot{r}_-^2$, and decreases during impact by an amount $\frac{1}{2}m_H\dot{r}_+^2$ (the two coordinate changes obviously do not change the kinetic energy). From the setup and the sketch of Figure 5.17, we can see that $\dot{r}_-$ should be positive to increase the forward energy, and that the radial velocity $\dot{r}_+$ just before impact is negative (the leg is trying to push into the ground). From these two considerations, we can compute the required push-off speed as $\dot{r}_- = -\dot{r}_+$. The velocity $\dot{r}_+$ can be expressed in terms of the pre-impact angles and velocities through (5.36), from which finally $\dot{r}_-$ can be solved as

$$\dot{r}_- = \frac{\dot{q}^5\cos(q^4-q^6)\cos(q^7)\sin(q^5) + \cos(q^5)\left(\dot{q}^4\cos(q^7)\sin(q^4-q^6) - \dot{q}^5\sin(q^7)\right)}{1+\cos(q^5)\cos(q^4-q^6)\cos(q^7)+\sin(q^5)\sin(q^7)} \tag{5.37}$$

where all variables should be evaluated just before impact. When this push-off velocity is applied just before impact, the kinetic energy before push-off is the same as the kinetic energy after impact.

5.4.3 Lateral stabilization and control by foot placement

We can combine the continuous dynamics (5.31), impact condition (4.35), push-off velocity (5.37), and impact projection (5.36) into one set of equations that represents the simplified 3d walker with a first layer of control: energy regulation by push-off. The second control layer is to implement foot placement, *i.e.* to find control inputs u_1 and u_2 that position the swing leg in such a way that stable walking is attained. From the model equations (5.31), we see that the dynamics of the swing leg are just simple integrators

$$\begin{aligned}\dot{q}^6 &= u_1\\ \dot{q}^7 &= u_2\end{aligned} \tag{5.38}$$

which can be easily controlled to some setpoint q^i_{ref} for $i = 6, 7$, for example by a proportional controller $u_i = K_u(q^i_{\text{ref}} - q^i)$ with $K_u > 0$ large enough. Of

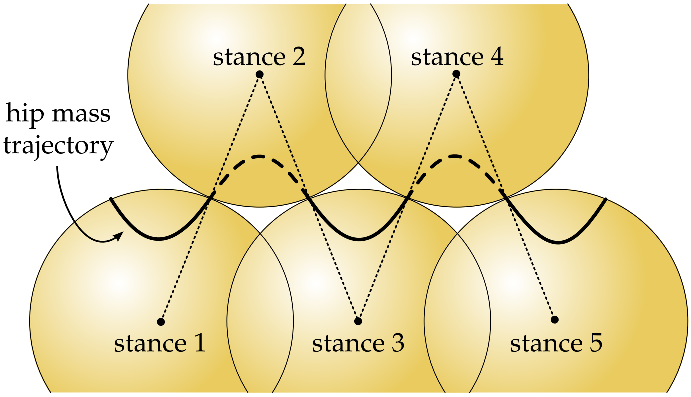


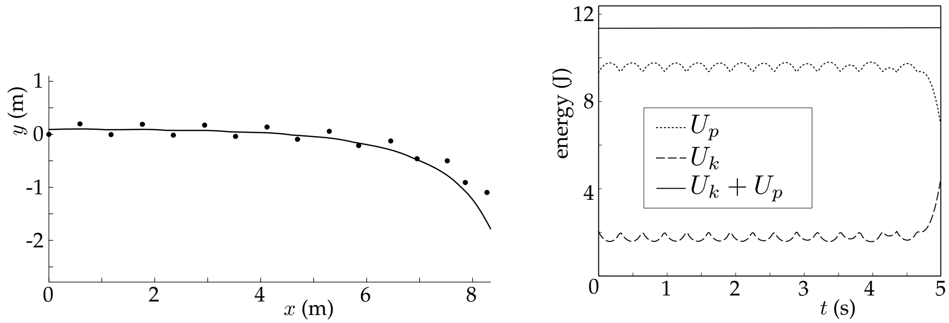
Figure 5.18: Top view of a repetitive walking motion with stance leg and swing leg aligned in a vertical plane on impact. The hip mass alternatively over a sphere around the current stance foot: for solid segments, one foot is the stance foot (odd numbers) and for dashed segments, the other foot is the stance foot (even numbers).

course, the swing leg should move to its desired position without touching the ground, which requires some kinematic planning. For the purpose of control by foot placement, though, this aspect is not very relevant, and we do not discuss it further in this thesis.

As a first step in foot placement control, we need to choose a nominal trajectory, *i.e.* a nominal stepping motion that we want the robot to converge to. We restrict our choices here to a trajectory such that, on impact, the stance leg and the swing leg define a vertical plane, which means that in a top view, they are aligned. In terms of coordinates, this means that on impact $q^6 = -q^4$ and $q^7 = -q^5$. Even with this restriction, the system still allows many cyclic motions, as can be seen from Figure 5.18. We can, within reasonable limits, freely choose the step length and step width (which determine the configuration on impact), as well as the forward velocity of the hip mass on impact. If this forward velocity is large enough, we can find a corresponding lateral velocity on impact, such that touch-down of the swing leg occurs exactly when

$$\begin{aligned} q^4(T) &= -q^6(T) = -q^4(0) \\ q^5(T) &= -q^7(T) = q^5(0) \end{aligned} \tag{5.39}$$

at the impact time T. Since the two legs are aligned on touch-down, the effect of push-off plus impact is that $\dot{x}_h, \dot{y}_h$ are unchanged during impact, and $\dot{z}_h$ is equal but mirrored. So, the configuration and velocity at the beginning of the second step is equal to the mirrored configuration at the beginning of the first step, thus



(a) Hip trajectory (line) and foot prints (dots).

(b) Kinetic and potential energy.

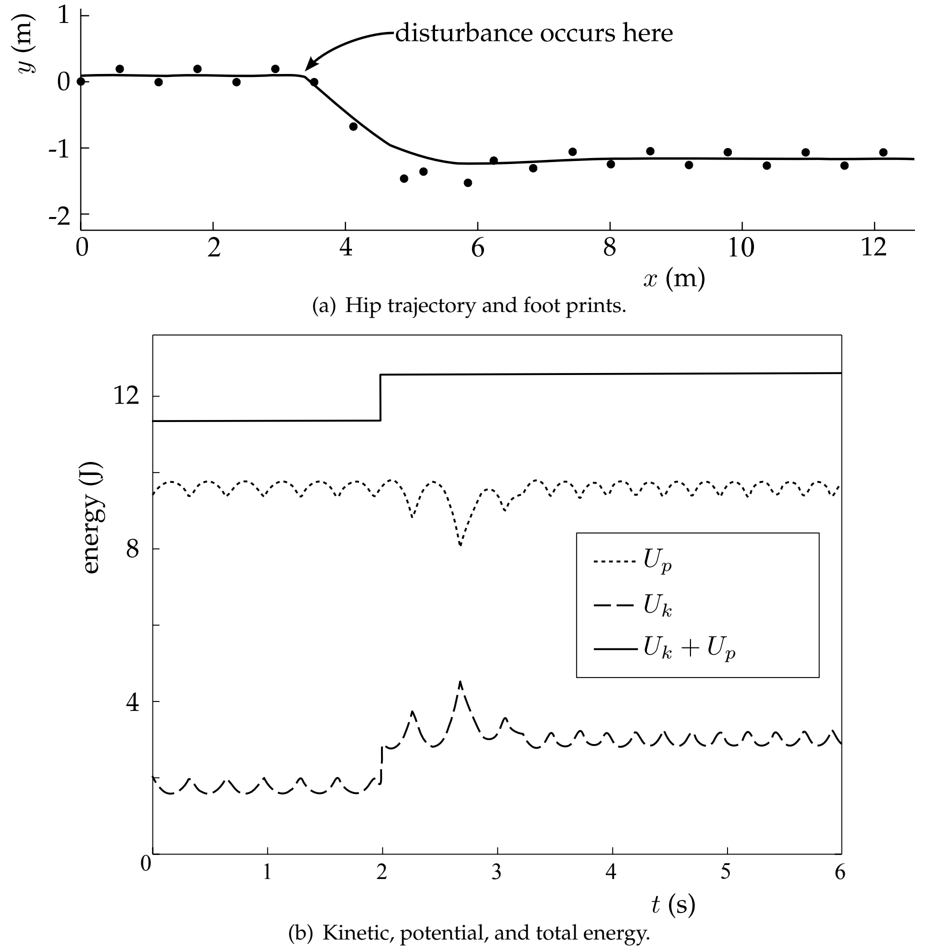
Figure 5.19: Fixed set-points for the swing leg result in unstable walking.

creating a cycle.

Unfortunately, most of the trajectories constructed in this way are unstable, in particular the practical trajectories for which the step length is larger than the step width. This can be seen for example in the simulation of Figure 5.19, where we chose $q^6_{\text{ref}} = -0.3$, $q^7_{\text{ref}} = 0.1$, $p_1 = 2$, and $p_2 = 0.1$, which is close to a cycle. For the first few steps, the motion of the hip mass oscillates between the two feet, but then it falls over to one side. This instability can be proved, as was done by van Oort (2005), by computing the eigenvalues of the linearization of the Poincare mapping discussed in Section 4.1. One of these eigenvalues has norm larger than one. Similar results were obtained by Kuo (1999) for a straight-legged 3D walker with curved feet. Note that by construction, the robot can only be unstable in the lateral (sideways) direction. It cannot fall backward if enough kinetic energy is given to push it over its highest point, and it cannot fall forward since the swing leg is assumed to be controlled quickly enough to be at the required position before touch-down.

To enhance the foot placement controller to stabilize the nominal trajectories, we first consider again the trajectory of Figure 5.18. Since the kinetic energy is constant throughout the push-off and impact phase, and since the gravitational force is conservative, the trajectories are symmetric around the midpoint of the step, *i.e.* when $q^4 = 0$, the hip mass should be at its highest point and we should have $\dot{q}^5 = 0$. Deviations from the nominal trajectory can thus be found by measuring $\dot{q}^5$ when $q^4 = 0$. If $\dot{q}^5 > 0$ at the midpoint, it means that the hip mass is moving too much over the stance leg, and the robot is threatening to fall to its right, *i.e.* in the $-y$ direction. If, on the other hand, $\dot{q}^5 < 0$, then the robot is threatening to fall to its left side, *i.e.* in the $+y$ direction.

The foot placement controller can be extended to stabilize these motions. Placing the foot at a different lateral position will change the lateral acceleration on the hip mass in the next step, due to the impact and normal forces on the stance



(a) Hip trajectory and foot prints.

(b) Kinetic, potential, and total energy.

Figure 5.20: Simulation of walking using the foot placement controller (5.40), with a lateral disturbance in the $-y$ direction at $t = 2$.

foot, which are roughly proportional to $\sin(q^5)$. If $\dot{q}^5 < 0$ in a step (towards the $-y$ direction), then the stance foot should be placed more outward, such that the lateral acceleration is more towards $+y$, thus stabilizing the system. From this principle, we can construct a simple control law for the swing leg as

$$\begin{aligned} q^6_{\text{ref}} &= q^6_{\text{nom}} \\ q^7_{\text{ref}} &= q^7_{\text{nom}} - K_l \, \dot{q}^5\big|_{q^4=0} \end{aligned} \tag{5.40}$$

where $K_l > 0$ is a control gain, and q^i_{nom} are the nominal angles, which can be derived from the desired step length and step width.

Figure 5.20 shows a simulation of this control law, where we chose the nominal angles as $q^6_{\text{nom}} = -0.3$, $q^7_{\text{nom}} = 0.1$, and the control gain $K_l = 0.3$. After a few steps (at $t = 2$), we introduce a disturbance in terms of an impulsive push at the hip mass in the $-y$ direction. The figure shows that the robot recovers from the disturbance and converges to walking along a different line in the x direction. Furthermore, the disturbance introduces extra energy (after $t = 2$), and since the push-off controller keeps the energy constant, the robot walks with a higher speed after the disturbance.

The robot converges to motion in the x direction, but the exact y position is determined by the initial condition and possible disturbances. We can control the y position indirectly, by extending the controller in order not to make $\dot{q}^5$ at the midpoint equal to zero, but equal to some reference value, which is computed to steer the walker up or down in the y direction. For example, we can define a reference signal

$$\dot{q}^5_{\text{ref}} = K_p \left(y_h - y_{h,\text{ref}}\right) + K_d \dot{y}_h \tag{5.41}$$

which is a form of proportional plus derivative control for the y position of the hip mass, with controller gains $K_p, K_d \geq 0$. Note the positive feedback in the signals y_h and $\dot{y}_h$, since motion in the positive y direction requires $\dot{q}^5$ to be negative. We can then build a controller by first limiting (5.41) between some minimum and maximum lateral velocity (such that the system does not become unstable), and then extending (5.40) to

$$\begin{aligned} q^6_{\text{ref}} &= q^6_{\text{nom}} \\ q^7_{\text{ref}} &= q^7_{\text{nom}} + K_l \left(\dot{q}^5_{\text{ref}} - \dot{q}^5\big|_{q^4=0}\right) \end{aligned} \tag{5.42}$$

Simulations of this controller are shown in Figures 5.21 and 5.22. Figure 5.21 shows the behavior for a reference signal equal to zero for all times, and the same lateral disturbance as in Figure 5.20. As expected, the walker still recovers from the disturbance, but now also walks back to the reference trajectory $y = 0$. Figure 5.22 illustrates the behavior for a non-trivial reference signal (but no lateral disturbance). For both simulations, we chose parameters $K_p = 1$, $K_d = 1.5$, and

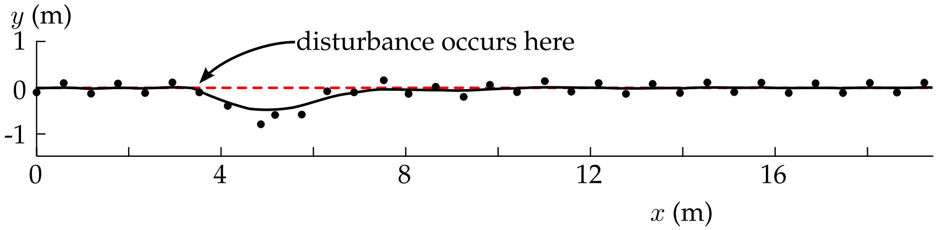


Figure 5.21: Reference trajectory (dotted), hip trajectory (solid), and foot prints (dots) when walking using the foot placement controller (5.42), and with a lateral disturbance in the $-y$ direction around $x = 4$ m.

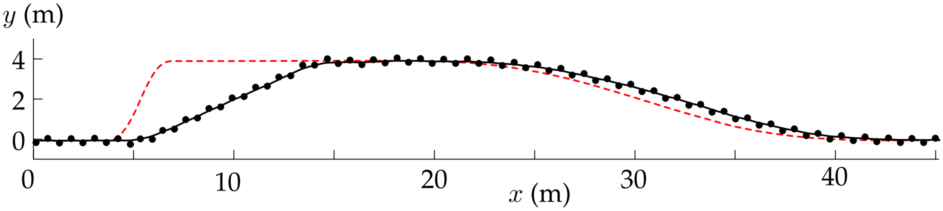


Figure 5.22: Reference trajectory (dotted), hip trajectory (solid), and foot prints (dots) when walking using the foot placement controller (5.42).

the reference velocity (5.41) was limited to maximally 0.3 rad/s in either direction.

Another extension in this direction would be to implement following a reference heading: walking along a different line than the fixed x direction. This can be implemented by evaluating $\dot{q}^5$ not when $q^4 = 0$, but when q^4 and q^5 are such that the stance leg is perpendicular to the line describing the desired heading. Furthermore, the nominal angles need to be computed from the step length and step width relative to the desired heading instead of relative to the x-axis.

All foot placement controllers presented here only require choosing a nominal stepping pattern, as well as some controller gains. They do not require knowledge about the exact cycle, and they are invariant to changes in the hip mass. This makes them very robust against parameter variations. For use on practical robots, the effect of legs with nonzero mass should be investigated, as well as the possibly positive effect of adding a trunk: by swinging the trunk in the opposite direction as the swing leg, the effects of the leg swinging on the motion of the hip mass may be reduced.

Chapter 6

Conclusions

6.1 Conclusions

The goals of this thesis, as formulated in Section 1.3, were to develop a port-Hamiltonian modeling framework for walking robots, to use this model to analyze several walking robots, and to use the knowledge obtained to design energy-efficient controllers for these robots. In this section, we discuss to what extent these research goals have been accomplished.

Develop a port-Hamiltonian modeling framework for walking robots

Chapters 2 and 3 developed a set of modeling tools, based on port-Hamiltonian formulations and ideas, that can be used to describe general rigid mechanisms, possibly in contact. Approximating walking robots as ideal rigid mechanisms is useful to study the simplified, essential properties of walking, and these approximations can also be used to design and test controllers. If more accurate simulation models of walking robots are needed, and especially in the case of stiff impacts, then the model should be extended to include non-rigid aspects, such as flexibility and mechanical play.

The results of Chapter 2 are an extension of classical results for serial mechanisms to the case of mechanisms with joints with more general configuration manifolds, such as Lie groups. This generalization is useful for systems that require singularity-free parameterizations of these joints, *e.g.* systems that are moving freely in three-dimensional space, such as walking robots. Since the derivation of the dynamics is a modular, joint-based process, it can be easily automated, and some of the results and ideas have been implemented in the 3D mechanics toolbox of the simulation program 20sim (Control Lab Products 2005).

Chapter 3 developed general models for two types of contact: compliant and rigid. Compliant contact models can be implemented as port-Hamiltonian subsystems and interconnected to the contacting bodies through power ports. As such, they are highly suitable for modular modeling purposes. A model of a walking robot, for example, is obtained simply by taking a model of the free mechanism and placing contact models between the feet and the ground. Rigid contact models are more suitable for analysis than simulation, as they simplify the fast dynamics of compliant contact to a state projection at one single time instant. However, practical implementation of rigid models requires direct modification of the equations of the mechanism, as well as cumbersome conditional checks to describe the contact release conditions, which become especially complex for multiple contacts. As such, rigid contact models are not modular and hence, at this stage, not suitable for inclusion in a model library or implementation in automatic modeling tools.

During the development of the modeling framework, no experimental tests were conducted to verify whether the various modeling assumptions were true. This approach was taken since the goal was to develop a *framework*, *i.e.* a set of tools that can be used to model a whole class of mechanisms, possibly in contact with other mechanisms or with a fixed world. The assumption of rigidity restricts the class of systems that can be considered, or at least reduces the accuracy of the predictions (as was seen in the comparison of simulation and experiment in Section 5.3), but it allows to draw general conclusions about the remaining class of systems, and the resulting models for these systems are simple enough to be used in calculations. Without the simplifying assumptions, the models just would not be manageable.

Analyze models of several walking robots

Three different walking robots were analyzed in Chapter 4: the standard planar compass-gait walker, an idealized rigid model of the experimental robot Dribbel, and a straight-legged three-dimensional walking robot with a trunk. Simulation and analysis models of these robots were developed using the tools of Chapters 2 and 3.

It was shown how the problem of finding natural, energy-efficient walking gaits can be formulated as an optimization problem. For the compass-gait robot walking down a small slope, the optimized motions were found to be purely passive motions, *i.e.* they require zero torque. The stability of these gaits, however, can not be determined from the optimization routine. For the experimental robot Dribbel walking down a slope, no stable passive motions could be found. For the three-dimensional robot, the most efficient fixed-speed level ground walking gait was computed, with the mass of the trunk distributed in different ways between the head and the hip. From these optimizations, it was seen that placing more mass at the hip joint results in more efficient motion. Hence, from a purely mechanical perspective, adding a directly actuated trunk to a straight-legged walker does not improve its efficiency.

The modeling framework developed in this thesis proved useful for simulation of walking robots, as well as analysis of their efficiency and prediction of efficient walking cycles. The analysis of the impact projection by itself allows estimation of the mechanical energy losses and suggests energy-efficient postures and velocities. However, impact is only part of the walking cycle, and hence these postures and velocity may very well be sub-optimal for the overall efficiency of a gait.

Design energy-efficient controllers for walking robots

Chapter 5 demonstrated that the mechanical structure of a walking robot can greatly influence and optimize its energy-efficiency. It showed how the compass-gait walker can passively walk down a certain slope at a whole range of walking speeds, simply by choosing the right mass distribution on the robot, or by adding an extra spring and damper.

The general control problem of tracking a certain desired curve can be simplified by choosing coordinates that reflect this control goal. It was shown that, once the coordinates have been chosen to encode this control goal system, designing the control law itself becomes almost trivial. Formulating the dynamics as a port-Hamiltonian system also allows an intuitive interpretation of the control objective as transporting energy from one storage location to another.

Walking robots can not only be stabilized by trajectory tracking control methods, but also by simple intuitive control algorithms, *e.g.* based on a simple PD controller on the hip joint (for Dribbel) or foot placement (for a three-dimensional walker). Formal proof of stability and convergence for such controllers, however, is much harder.

Overall conclusions

The research described in this thesis has been conducted as part of the GeoPlex project, and as such, another purpose was to investigate the advantages and disadvantages of using the port-Hamiltonian framework. A more commonly used framework for analyzing mechanical systems such as walking robots is the Lagrangian framework, meaning (roughly) the framework in which the state variables are positions and velocities, as opposed to positions and momenta.

For most mechanical systems, the momentum variables are related to the velocity variables by multiplication of an invertible mass matrix. For these systems, the two frameworks are mathematically equivalent, meaning that any model or controller expressed in one framework can be translated to a completely equivalent model or controller in the other framework. In particular, all modeling and control results in this thesis can be formulated equivalently in terms of velocities instead of momenta. As such, using either of the frameworks seems to be more a matter of personal taste than mathematical necessity.

However, since mathematical models are to be used not only in numerical computations by a computer, but also in manual analysis by a real person, the frameworks do have very distinct properties. As shown in this thesis, the structure of a port-Hamiltonian framework separates mechanical systems in energy-exchanging subsystems, and variables are considered in pairs of collocated power-variables. This structure can help in both the analysis and the controller design, especially when intrinsically passive or power-continuous controllers are to be designed. This was shown, for example, in the controller design in Section 5.2. For practical implementation, however, the Lagrangian framework is often better suited: in the polynomial approximation of the joint trajectories, for example, it is much easier to express the velocities in terms of the parameters of these polynomials than the momenta. Furthermore, specifying initial velocities is much more intuitive than specifying initial momenta.

Eventually, this author believes that the *ideas* from port-Hamiltonian framework, such as emphasis on energy flows, collocation of variables, and power-continuity of controllers, are indeed very useful and should be kept in mind when analyzing and designing physical systems, especially when energy-efficiency or passivity plays a role. Whether the final dynamic equations are better formulated in port-Hamiltonian or Lagrangian terms, however, depends on the particular task and is less important.

The modeling framework developed in this thesis allows for rapid prototyping of models of complex three-dimensional robotic mechanisms. Furthermore, the analysis method using parameterizations of both reference trajectories and mechanical structure is general, and can be applied to all kinds of walking robots.

General analytical tools, such as the ones developed in this thesis, should form a rigorous mathematical backup to verify, explain, generalize, and optimize ideas based on engineering intuition. Thus, results from one specific example can be extended to a class of systems, which, in the end, is what distinguishes science from ad hoc solutions.

6.2 Recommendations for future work

The answers to the research questions of this thesis have led to more questions, and hence several directions for future research. We briefly discuss these directions here, organized by topic.

Modeling framework and tools

- The most general joint types (including nonholonomic joints) still need to be implemented in software, *e.g.* in the 20sim 3D mechanics toolbox. Although the general definition of these joints might be mathematically quite involved, specific examples such as spherical joints can be implemented as just another joint type in the editor, with the mathematical details hidden from the user.

- Identifying the parameters in a port-Hamiltonian model generally involves estimating physical parameters, such as the mass of a body, or the stiffness and damping of the ground. It would be beneficial to have a solid identification method for port-Hamiltonian systems, *i.e.* a robust method that determines the parameters based on measurement data. Similarly to existing identification methods for classes of linear systems, a port-Hamiltonian identification method would compute the 'best fitting' port-Hamiltonian system of a certain order from measurement data.

- The local, differential approach to contact kinematics, as presented in this thesis, should be combined with global approaches from the field of computer graphics. These global approaches can then be used to give the contact positions on initialization and occasional resets to avoid drift, while the local approach tracks the contact positions and velocities continuously between resets.

- The problem of multiple rigid contact points can be formulated and analyzed as a linear complementarity problem (Cottle et al. 1992). This approach should be investigated, as it may suggest efficient ways to determine which contact constraints are active and which should be released. However, multiple impacts remain a delicate and sometimes even ill-defined problem, and hence, in the end, attempting to approximate multiple compliant impacts as rigid impacts may not even be possible.

- The experimental results with Dribbel showed that approximating a practical robot by a rigid mechanism can be a rather crude approximation, especially during rigid impacts. Flexibilities in the links and mechanical play in the joints result in larger energy loss on impact; as if a shock wave travels through the mechanism. It may be possible to implement this extra energy loss as another instantaneous projection operation on impact.

Analysis of walking robots

- The analysis of the three-dimensional walker with a trunk, as well as a similar study on a planar robot with trunk, demonstrate that least control torque is required when the trunk is not present and all its mass is concentrated at the hip joint. This begs the question why humans *do* have a trunk, and future research should investigate this. Perhaps the trunk is only useful when the system is underactuated (providing a counter weight for motion of the swing leg). It may also be useful in three-dimensional robots as a location to attach arms that counter swing the leg motion, thus avoiding rotation around the vertical axis. And maybe the only use of the trunk is to store computers and other equipment on it, *i.e.* it may be more efficient to have mass on the trunk instead of on the legs. Finally, efficiency may be improved by mechanically constraining the motion of the trunk, as done by Wisse (2004), such that no extra torques are required to keep the trunk upright.

- Continuing in this direction, more detailed models of human proportions should be investigated, *i.e.* three-dimensional models with human masses and inertias, a trunk, arms, and kneed legs. As discussed in Chapter 1, analyzing the mechanics of walking in nature can provide insight and suggest improvement for both natural and robotic walking. Thus, the results from the analysis of human-like mechanisms may explain how and why people walk the way they do, and can help in the rehabilitation of people who do not walk that way anymore. Furthermore, the actuation locations and strategies for humans could provide useful suggestions for the actuation of robotic walkers.

- The optimization technique for finding efficient gaits of a robot produces nominal trajectories, but tells us nothing about the stability of these trajectories. To describe the stability of a cycle in a practical way, a physically intuitive and coordinate-free definition of the region of attraction is needed, *i.e.* some way to measure the volume of the region in state space from which the system converges to this nominal trajectory.

- The numerical optimization for efficient trajectories at this point requires some human input to discard solutions if the optimization gets stuck in a local minimum with too high a cost. Suitable initial estimates for the optimization also need to be chosen manually. Online implementation is hence not yet possible, although it would be useful to adapt a reference trajectory to the particular circumstances. Future research should make the optimization more robust and reliable to allow this.

- One of the problems that prevent the automated search for efficient trajectories is the number of degrees of freedom in the optimization procedure. Especially for more complex robots, this large search space makes the optimization problem hard to solve. This complexity may be tackled by using lower-order polynomials first, in order to get a rough initial estimate of the efficient motion, and then using these estimates as initial guess for higher order polynomials. Another option would be to use a type of 'morphological expansion', as illustrated in Figure 6.1: a complex robot is approximated by simpler robots with fewer degrees of freedom, but with dynamic properties (mass and inertia) as close as possible to the complex robot. First, the motion of the simplest robot (left-most figure) is optimized to be as efficient

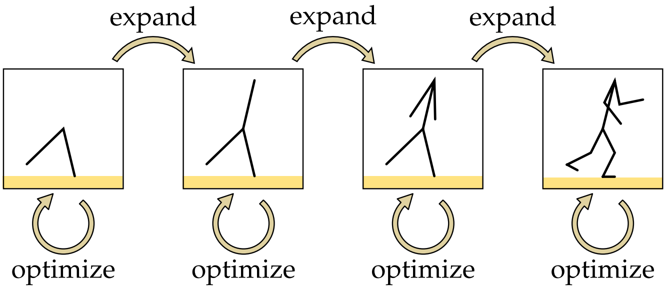


Figure 6.1: Idea of morphologically expanding the optimized solution for a simple walker to a more complex walker.

as possible. The resulting motion is then used as initial estimate for the optimization of a robot that is slightly more complex. This iteration continues until the optimal motion for the most complex robot is obtained.

Control of walking robots

- For specific practical robots for which actuators have been chosen, the cost function should be adapted to reflect these choices. In other words, the current, rather arbitrary, cost function of summed squared torque components should be replaced by something that really reflects the energy cost of actuation torques in certain directions.

- Similarly, it should be investigated whether underactuation can be included in the numerical search. Underactuated joints may be thought of as very inefficient motors with a very high associated cost. Optimal trajectories with a low cost hence will be such that little to no torque is required from these joints. It remains to be seen, though, whether subsequently approximating these little-torque actuators by zero-torque unactuated joints results in stable motion.

- Instead of computing the optimal torques as polynomial functions of time, it should be checked whether they can be described as polynomial (or other) functions of the joint angles. If this is possible, then the resulting functions can be thought of as caused by a certain potential field, namely the integral of the forces over the joint space. This gives a direct parameterization of the optimal stiffness distribution.

- Given the optimal spring, mass, and damper configuration for a certain speed, these mechanical elements should somehow be implemented in the mechanical structure. Several suggestions for implementation have already been given, but practical feasibility and efficiency still has to be proved.

- An automatic procedure should be devised to construct the desired vector fields as used in port-based asymptotic curve tracking. The current vector field for the simple compass-gait walker, based on manual extension from a single nominal curve, is already reasonably complex to construct manually. For higher-dimensional systems, this construction process becomes downright impossible. The automated method should take a desired curve as input, and construct a vector field directed towards this curve, for example by using a weighted combination of the Euclidean shortest distance vector to the curve and the velocity vector at the closest point of the curve. This *local* approach for points that are reasonably close to the desired curve, could then be augmented with *global* motion planning techniques, for example to avoid regions of the state space that correspond to toe stubbing.

- Concerning research on Dribbel, the experimental results presented in this thesis were very preliminary, using only a fixed-parameter switching PD controller on the hip joint. Although this results in stable walking (most of the time), the robustness against disturbances and initial conditions is not very high. Future work should investigate the possibilities of using the torque sensor on the hip motor, as well as extra inertial sensors. Based on measurements from the torque sensor, the control gains on the hip controller can be adapted (on or off-line). The measurements of the inertial sensors can be used online to adapt the setpoint for the hip angle. This effectively implementing a form of foot placement control, which can be used to control the forward speed and increase robustness against disturbances.

- The mechanical construction of Dribbel is based on a very crude knee-locking mechanism that accounts for more than a third of the total power consumption. Better mechanisms should be investigated, possibly using mechanical instead of magnetic locking forces. Furthermore, the use of different foot shapes (curved feet instead of the current point feet) can reduce the energy losses on impact, at the cost of more difficult foot angle measurement. Finally, the idea of using additional tunable passive mechanical elements to change the natural motion, demonstrated conceptually in Section 5.1 of this thesis for the compass-gait walker, should be implemented on Dribbel to see whether this is also a feasible control method in practice.

- The control method of foot placement has been demonstrated on a simple walker with massless legs. Future work should extend these results to legs with nonzero mass. In this case, the presence of a trunk and arms may be beneficial to counteract the inertial effects from the swing leg, *i.e.* the trunk could move such that the combined motion of trunk and swing leg least disturbs the hip mass from its trajectory as a three-dimensional inverted pendulum.

- Sideways position control by foot-placement should be augmented with horizontal speed control and adaptable push-off velocities. By placing the feet more forward or backward, the horizontal velocity can be increased or decreased. In this way, the robot can also be started and stopped by means of foot placement.

Appendix A

Mathematical Background

This appendix presents a short intuitive overview of the mathematical concepts used in this thesis. More detailed, precise, and extended treatments can be found in many textbooks, such as Lay (2002) and Trefethen & Bau (1997) for linear algebra, Dubrovin et al. (1984, 1985) and Burke (1985) for differential geometry, and Gilmore (1974) and Selig (2005) for Lie groups.

A.1 Linear algebra

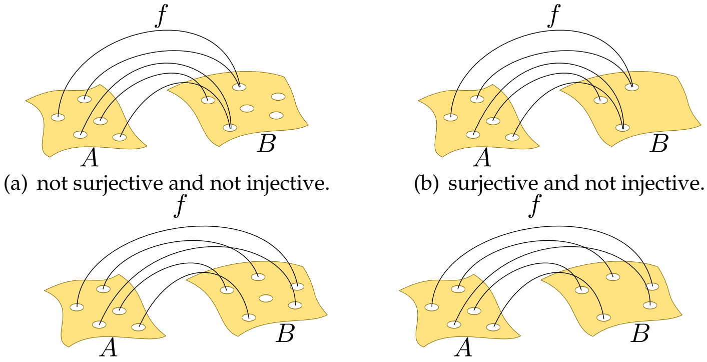
We start with two basic definitions of mappings and some possible properties. These properties are illustrated in Figure A.1.

Definition A.1 (mappings). *A mapping f between two sets A and B associates exactly one element of B to each element of A. We denote it abstractly as $f : A \rightarrow B$, and its action on an element $a \in A$ as $f(a) \mapsto b$ with $b \in B$. The set A is called the domain of f, and the set B its co-domain. The set of all $b \in B$ such that there exists an $a \in A$ with $f(a) \mapsto b$ is called the range of f.*

Definition A.2 (surjective, injective, bijective). *A mapping $f : A \rightarrow B$ is surjective (or* onto*), if its range is equal to its co-domain. It is injective (or* one-to-one*) if for every b in its range, there is exactly one $a \in A$ such that $f(a) \mapsto b$. A mapping is bijective (or* one-to-one and onto*) if it is injective and surjective.*

In addition, a *diffeomorphism* is a mapping between $\mathbb{R}^n$ and $\mathbb{R}^n$ that is injective, continuously differentiable, and has a continuously differentiable inverse.

Some examples of different types of mappings $f : \mathbb{R} \rightarrow \mathbb{R}$ are the functions $f(x) \mapsto x^2$ (not surjective, not injective), $f(x) \mapsto \tan(x)$ (surjective, not injective), $f(x) \mapsto e^x$ (not surjective, injective), and $f(x) \mapsto x^3$ (bijective).



(a) not surjective and not injective.

(b) surjective and not injective.

(c) not surjective and injective.

(d) surjective and injective.

Figure A.1: Examples of mappings $f : A \to B$ with different properties.

Definition A.3 (vector space). *A real vector space V is a set of elements (called vectors), one element called the identity (or zero-vector $\overrightarrow{0}$), and two operations $\oplus$ (addition of two vectors) and $\cdot$ (multiplication of a vector by a scalar), such that*

- *for all two elements $v_1, v_2 \in V$, also $v_1 \oplus v_2 \in V$;*
- *for all elements $v \in V$ and $x \in \mathbb{R}$, also $x \cdot v \in V$;*
- *for all elements $v \in V$, there exists a $v^{-1} \in V$ such that $v \oplus v^{-1} = \overrightarrow{0}$;*

and such that the following properties hold for all $v_1, v_2, v_3 \in V$ and $x_1, x_2 \in \mathbb{R}$.

$$\begin{aligned} v_1 \oplus \overrightarrow{0} &= v_1 & (v_1 \oplus v_2) \oplus v_3 &= v_1 \oplus (v_2 \oplus v_3) \\ 1 \cdot v_1 &= v_1 & (x_1 + x_2) \cdot v_1 &= (x_1 \cdot v_1) \oplus (x_2 \cdot v_1) \\ x_1 \cdot (x_2 \cdot v_1) &= (x_1 x_2) \cdot v_1 & x_1 \cdot (v_1 \oplus v_2) &= (x_1 \cdot v_1) \oplus (x_1 \cdot v_2) \end{aligned}$$

where $x_1 + x_2$ and $x_1 x_2$ are standard addition and multiplication of real numbers.

The abstract definition of a vector space includes many different spaces with a linear structure. Not only an obvious example like the space of all velocity vectors of a point mass is a vector space, but, for example, also the space of all 2×3 matrices, if we take element-wise addition as the $\oplus$ operator and the zero-matrix as the identity element.

Since a vector space is closed under addition and scalar multiplication, we can search for the smallest number of elements $e_i \in V$ such that any element of V can be constructed by addition and scalar multiplication of the elements e_i. If we can find $n < \infty$ elements e_i that accomplish this, then the vector space is said to be n-dimensional, and the e_i elements are called a basis of the vector space.

The dimension n of a vector space is unique, but the choice of basis e_i is not. By definition, we can express any element $v \in V$ as a linear combination of the basis elements, *i.e.* as

$$v = \sum_{i=1}^{n} v^i e_i = v^1 e_1 + v^2 e_2 + \ldots + v^n e_n \tag{A.1}$$

the n numbers $v^i \in \mathbb{R}$ can serve as coordinates for V, as they define a bijective mapping between $\mathbb{R}^n$ and V.

Definition A.4 (Lie algebra). *A Lie algebra is a vector space together with a binary operator $[,] : V \times V \to V$ (called Lie bracket or commutator), that satisfies the following properties for all $v_1, v_2, v_3 \in V$ and $a_1, a_2 \in \mathbb{R}$*

$$\textit{bilinearity:} \quad \begin{cases} [a_1 v_1 + a_2 v_2, v_3] = a_1[v_1, v_3] + a_2[v_2, v_3] \\ [v_1, a_1 v_2 + a_2 v_3] = a_1[v_1, v_2] + a_2[v_1, v_3] \end{cases} \tag{A.2}$$

$$\textit{skew-symmetry:} \quad [v_1, v_2] = -[v_2, v_1] \tag{A.3}$$

$$\textit{Jacobi's identity:} \quad [v_1, [v_2, v_3]] + [v_2, [v_3, v_1]] + [v_3, [v_1, v_2]] = 0 \tag{A.4}$$

An example of a Lie algebra is the vector space V of all $n \times n$ matrices with the Lie bracket defined as $[A, B] := AB - BA$ for $A, B \in V$. The concept of a Lie algebra is used in the context of Lie groups in Section A.3.

Definition A.5 (dual vector space). *The dual space V^* of a vector space V is the space of all linear mappings (called co-vectors) from V to $\mathbb{R}$,* i.e. *all mappings $f : V \to \mathbb{R}$ such that for all $v_i \in V$ and $x_i \in \mathbb{R}$.*

$$f\left((x_1 \cdot v_1) \oplus \ldots \oplus (x_k \cdot v_k)\right) = x_1 f(v_1) + \ldots + x_k f(v_k) \tag{A.5}$$

Definition A.6 (dual product). *The dual product is the natural pairing of an element $v \in V$ and an element $f \in V^*$ as $\langle f|v \rangle := f(v) \in \mathbb{R}$*

If we choose a basis e_i for V, we can express any element $v \in V$ as (A.1), and hence from (A.5) we see that the mapping of v by an element $f \in V^*$ can be written as

$$\langle f|v \rangle = f(v) = f\left(\sum_{i=1}^{n} v^i e_i\right) = \sum_{i=1}^{n} v^i f(e_i) \tag{A.6}$$

i.e. as a linear combination of the mappings of the basis elements. This shows that a dual element f is fully defined by how it maps the basis elements, and, since each element $f(e_i)$ is a single real number, it shows that the dimension of

V^* is equal to the dimension of V, *i.e.* the number of basis elements e_i. It also suggests a basis for the dual space V^*, which we denote by e^j and is defined by the condition

$$\langle e^j | e_i \rangle = e^j(e_i) = \delta_i^j := \begin{cases} 1 & \text{when } i = j \\ 0 & \text{when } i \neq j \end{cases} \tag{A.7}$$

where δ_j^i is the Kronecker delta. Any $f \in V^*$ can then be written as a linear combination of the basis elements e^i

$$f = \sum_{j=1}^{n} f_j e^j = f_1 e^1 + f_2 e^2 + \ldots + f_n e^n \tag{A.8}$$

with the numbers $f_j \in \mathbb{R}$ again defining coordinates for V^* in the basis e^j. With these choices of bases, computing the dual product (A.6) becomes

$$\langle f | v \rangle = \left(\sum_{j=1}^{n} f_j e^j \right) \left(\sum_{i=1}^{n} v^i e_i \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_j v^i e^j(e_i) = \sum_{i=1}^{n} f_i v^i \tag{A.9}$$

so, for these choices of bases, computing the dual product of a vector and a co-vector simply means summing the pair-wise products of their coordinates.

Vector spaces and their duals define an interesting mathematical structure, but they can also be used to represent a physical structure, namely as follows. Consider a robotic mechanism with n joints, for example the system of Figure 2.8 on page 46 for $n = 4$. The space of velocities $\dot{q}$ (at a point q) forms a vector space V, and we can choose a basis for example as $e_i = \dot{q}^i$, *i.e.* each basis element describes the unit-velocity of one joint, and zero velocity of the other joints.

This vector space V automatically induces a dual space V^* of abstract linear operators mapping a velocity to a real number. We can just ignore this dual space, but we can also think of it as the space of all collocated joint torques, *i.e.* the n-dimensional space with elements τ and basis elements $e^i = \tau_i$. From the structure of the vector space and its dual, we can pair elements as

$$\langle \tau | \dot{q} \rangle = \sum_{i=1}^{n} \tau_i \dot{q}^i \tag{A.10}$$

such that applying τ to $\dot{q}$ produces a real number. The reason for choosing this interpretation of a vector space and its dual becomes clear when we interpret also this real number: the dual product represents the mechanical power flowing into the system when it is moving with velocity $\dot{q}$ and with applied torques τ.

Associating the abstract mathematical concept of (dual) vector spaces to the practical physical concept of collocated power variables (force and velocity) can help reasoning about the physical concepts. The mathematical structure constrains computations to make sense. For example, computing the power as (A.10) only makes sense when τ and $\dot{q}$ are collocated, which is equivalent to V and V^* having dual bases as defined in (A.7). In this way, keeping the mathematical structure between physical variables in mind can help to avoid mistakes.

Definition A.7 (tensor). *Given a vector space V and its dual V^*, a tensor T is a mapping of the form*

$$T : \underbrace{V^* \times \cdots \times V^*}_{p \text{ times}} \times \underbrace{V \times \cdots \times V}_{q \text{ times}} \to \mathbb{R} \tag{A.11}$$

that is linear in all its arguments. The tensor T is said to have order $p+q$, order p contra-variant and order q co-variant, and is called a type (p,q) tensor.

Tensors are linear operators that map vectors and co-vectors to $\mathbb{R}$, and as such, are generalizations of the concepts of vectors and co-vectors. In fact, a co-vector is a type $(0,1)$ tensor, since it maps a vector (an element of one copy of V) to $\mathbb{R}$. Similarly, a vector is a type $(1,0)$ tensor, since it maps a co-vector (an element of one copy of V^*) to $\mathbb{R}$. Both mappings are defined by the dual product.

Basis elements and the corresponding coordinates for tensors can be constructed from coordinates for vectors and co-vectors, simply by taking the appropriate coordinates for each of the arguments. The $(i_1,\ldots,i_p,j_1,\ldots,j_q)$th coordinate of a type (p,q) tensor T, *i.e.* the result of applying T to the basis vectors $e^1,\ldots,e^p$ and co-vectors $e_1,\ldots,e_q$, is denoted by $T^{i_1,\ldots,i_p}_{j_1,\ldots,j_q}$. This convention of writing the contra-variant indices as superscripts and the co-variant indices as subscripts can be useful to quickly assess the type of tensor from its representation in coordinates.

A metric tensor, often denoted by g, is a symmetric positive-definite type $(0,2)$ tensor. It is symmetric in the sense that $g(v,w) = g(w,v)$ for any two vectors v, w, and positive-definite in the sense that $g(v,v) > 0$ for all vectors v except the zero-vector (in which case it returns zero by linearity of tensors). With a metric-tensor, we define the inner product between two vectors v and w as

$$v \cdot w := g(v,w) = g(w,v) = \sum_{i,j} g_{ij} v^i w^j \tag{A.12}$$

the length of a vector v as

$$|v| := \sqrt{v \cdot v} = \sqrt{g(v,v)} = \sqrt{\sum_{i,j} g_{ij} v^i v^j} \tag{A.13}$$

and the cosine of the angle between two nonzero vectors v and w as

$$\cos(\angle(v,w)) = \frac{v \cdot w}{|v||w|} \tag{A.14}$$

When $v \cdot w = 0$, the vectors v and w are said to be orthogonal in the metric g.

We can again relate the mathematical concept of a metric to physical variables. In this case, we take *e.g.* again the space of velocities $\dot{q}$ as the vector space V, and now the mass matrix M as a metric tensor, since it is indeed symmetric and positive definite. Then, when we apply the tensor M to two copies of $\dot{q}$, that is, we multiply the matrix with the vectors as $\dot{q}^T M \dot{q}$, we obtain a number that represents the physical quantity of twice the kinetic co-energy associated with the velocity $\dot{q}$.

We can also define a new tensor by applying the metric tensor only to one copy of V. The resulting tensor maps one tangent vector to $\mathbb{R}$, and is hence a tensor of type $(0, 1)$ — a co-vector. If we again take the physical example of a robot with velocity $\dot{q}$ and mass matrix M, the new tensor is $M\dot{q}$ — the generalized momentum (co) vector. Hence, we have seen two interpretations of the dual vector space: one as the space of forces, and one as the space of generalized momenta.

The distinction between the different types of tensors allow to assess what operations between them are possible. For example, a metric tensor is an operator mapping two vectors to $\mathbb{R}$, and hence it does not make mathematical sense to apply them to co-vectors, even though the coordinates of a metric tensor (represented by an $n \times n$ matrix) can be multiplied by the coordinates of a co-vector (represented by an n-dimensional column vector).

A.2 Differential geometry

> *"But how can you take such a map seriously," the poet snickered, "when it shows the earth flat, and you claim it's a sphere?" "What argument is that?" Abdul was indignant. "Could you depict a sphere in such a way that you could see everything on it? A map must serve to point out the way, and when you walk, you see the earth flat, not round."*
>
> — Umberto Eco, 'Baudolino'

The configuration space of a joint (or of a robot) is generally represented by an abstract space that is not directly equal to $\mathbb{R}^n$ for some suitable n. For example, Section 2.1.1 shows how the configuration of a rigid body is described by an element of $SE(3)$, and how using $\mathbb{R}^6$ (six numbers, *e.g.* including Euler angles) leads to singularities and other numerical problems that are not present in physics.

Differential geometry is a field of mathematics that makes exact the global properties of a configuration space such as $SE(3)$, while still allowing to do computations locally using real numbers. In this section, we give a very brief intuitive overview of the idea of differential geometry and how it can help to use some concepts from this field.

The central concept in differential geometry is the concept of a manifold. For the purpose of this thesis, we can think of a manifold as some kind of abstract space (such as $SE(3)$) that locally looks like $\mathbb{R}^n$. More precise, if we take a point in the abstract space, then the space around this point can locally be described by coordinates in an open subset of $\mathbb{R}^n$. An example of such a manifold is the surface of the earth, which globally is (more or less) a sphere, and locally (at each point) can be described by coordinates in $\mathbb{R}^2$, *i.e.* a flat chart. These charts, unfortunately, are not global, due to the topology of the sphere.

Once it becomes clear that a space is a manifold, *i.e.* once we have found enough local coordinate charts to $\mathbb{R}^n$ to cover the whole space, we can define global objects on the manifold (such as functions) by defining them first locally for each chart, and then checking certain compatibility conditions between the charts. These compatibility conditions ensure, for example, that a certain point in the abstract space, with two different coordinates in two different local charts, still has the same function value.

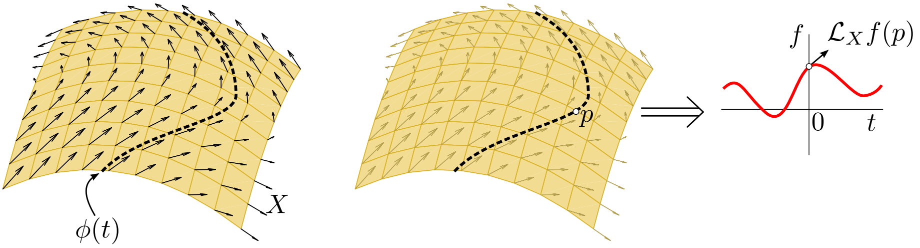
A manifold is called differentiable, if the mappings that change coordinates between different charts are diffeomorphisms. At each point p of a differentiable manifold $\mathcal{M}$, the space of all tangent vectors is called the tangent space, denoted $T_p\mathcal{M}$. This space is a linear vector space of dimension n, and describes all possible directions around p. The union of the tangent spaces over all points of $\mathcal{M}$ is called the tangent bundle, denoted $T\mathcal{M}$. An element of the tangent bundle consists of a point $p \in \mathcal{M}$ plus a vector in $T_p\mathcal{M}$: the tangent bundle is hence $2n$ dimensional.

The fact that the tangent space is a vector space allows to generalize the linear algebra concepts from Appendix A.1 to the setting of differential geometry. Since the tangent vector space at every point p has a dual, denoted by $T_p^*\mathcal{M}$, we can define the co-tangent or dual tangent bundle $T^*\mathcal{M}$ as the union of all dual tangent spaces. The concept of a tensor can be generalized to a *tensor field*, which is an object, defined on the manifold, that at each point p maps copies of the tangent space $T_p\mathcal{M}$ and dual tangent space $T_p^*\mathcal{M}$ to a real number, and that varies smoothly over $\mathcal{M}$. Note that tensor fields only operate on vectors and co-vectors that are elements of tangent and co-tangent space at the same point p.

An example of a tensor field is a vector field, which is a tensor field of type $(1, 0)$ that assigns to each point of the manifold a tangent vector. Figure A.2(a) shows an example. It also shows how, from a vector field, we can define its integral curves as the curves with velocity vector at all points equal to the value of the vector field at those points. Integral curves can be interpreted as the trajectories of a particle flowing along the vector field.

Given a function $f : \mathcal{M} \to \mathbb{R}$ of the points of the manifold, this function is obviously also defined for the points of the integral curves. If the function is differentiable, we define its Lie derivative along the vector field X at a point p as

$$(\mathcal{L}_X f)(p) := \frac{d}{dt}\left(f(\phi(t))\right)\bigg|_{t=0} \tag{A.15}$$



(a) Vector field X and one of its integral curves $\phi(t)$.

(b) A function f and its Lie derivative along X. The curve $\phi(t)$ is parameterized by t with $\phi(0) = p$.

Figure A.2: A vector field X on a manifold defines integral curves and Lie derivatives of functions at each point.

where $\phi(t)$ is an integral curve of X with $\phi(0) = p$. It can be shown that this expression is independent of the choice of integral curve. An example of a Lie derivative is shown in Figure A.2(b).

From two vector fields X and Y, we can also define a third vector field Z as the unique vector field such that for all functions f

$$\mathcal{L}_Z f = \mathcal{L}_{[X,Y]} f = \mathcal{L}_X(\mathcal{L}_Y f) - \mathcal{L}_Y(\mathcal{L}_X f) \tag{A.16}$$

This new vector field is called the Lie bracket of the two vector fields. It roughly represents the velocity when moving a little along X, then a little along Y, then a little along $-X$, and finally a little along $-Y$.

Another example of a tensor field is a metric tensor field, which assigns to each point of the manifold a metric tensor, *i.e.* a symmetric positive definite type $(0, 2)$ tensor. Such a tensor defines the metric concepts (dot-product, length, and angle) for tangent vectors at all points of the manifold.

Like the concept of a tensor, manifolds are mathematical structures that can be used to describe physical relations between variables. In this thesis, as in many works on robotics, the configuration space of a mechanism (describing all possible angles and positions of the joints in a mechanism) is thought of as a manifold Q . This means that, although almost all results are expressed in local coordinates $q^i \in \mathbb{R}^n$, it is realized that the global structure of the configuration space is not equal to $\mathbb{R}^n$. Similarly, the velocities $\dot{q}$ are thought of as belonging to the tangent space $T_q Q$, even though these two are usually expressed in coordinates $\dot{q}^i \in \mathbb{R}^n$. The mass matrix $M(q)$ is used as a metric field on Q and thus defines concepts like norms and orthogonality for velocity vectors. It also transforms velocities $\dot{q}$ to generalized momenta $p = M\dot{q}$, elements of the co-tangent space. Remembering that these physical variables are part of a mathematical structure is important, as it can help to avoid performing physically meaningless operations..

A.3 Lie groups

A.3.1 Definition and examples

Definition A.8 (group). *A group G is a set S together with a binary operator $\bullet : S \times S \to S$ and an element $I \in S$, such that for all $s_1, s_2, s_3 \in S$ we have*

$$\text{identity element:} \quad s_1 \bullet I = I \bullet s_1 = s_1 \tag{A.17}$$

$$\text{associativity:} \quad (s_1 \bullet s_2) \bullet s_3 = s_1 \bullet (s_2 \bullet s_3) \tag{A.18}$$

$$\text{inverse element:} \quad \exists\, s_1^{-1} \in S \text{ such that } s_1 \bullet s_1^{-1} = s_1^{-1} \bullet s_1 = I \tag{A.19}$$

If also $s_1 \bullet s_2 = s_2 \bullet s_1$ (commutativity property), the group is called abelian.

A simple example of a group is the set $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ together with the standard addition operator and identity element 0. It can be checked that this is indeed a group: adding zero to any element of the set indeed gives that same element, addition is associative, and for each element, the inverse is simply the negation of that element. Since summation is even commutative, this group is also abelian.

Definition A.9 (Lie group). *A Lie group $\mathcal{G}$ is a manifold that is also a group,* i.e. *it has a binary operator $\bullet : \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ and an identity element $I \in \mathcal{G}$ that satisfy the group properties.*

A Lie group is basically a group with a differentiable structure, which allows to talk about curves, velocities, tangent spaces, etcetera. We discuss a few examples of Lie groups that are useful for robotics, *i.e.* examples that describe positions and orientations in space. Since these groups are generally abstract, we also discuss matrix representations, *i.e.* sets of matrices with certain properties, which, together with the usual matrix multiplication as binary operator and the identity matrix as identity element, can be related one-to-one to abstract elements of the group. The matrix representations can be used in numerical computations as a type of singularity-free (though redundant) set of coordinates.

Example A.10 (translation). The group of all translations in n dimensions is denoted by $T(n)$, *e.g.* the group of translations in three dimensions is denoted by $T(3)$. Clearly these groups can be directly identified with $\mathbb{R}^n$, and so a matrix representation of $T(n)$ would be the space of n-dimensional column-vectors p_n, together with vector addition as the the binary operator, and the zero vector as the identity element. Another possible representation is as the set of all $(n+1 \times n+1)$ dimensional matrices structured as

$$\begin{bmatrix} I_n & p_n \\ 0 & 1 \end{bmatrix} \tag{A.20}$$

with p_n the translation vector, together with matrix multiplication as binary operator and the identity matrix as identity element. This method looks cumbersome and the matrix representation is highly redundant, but it proves useful when translations are combined with rotations. △

Example A.11 (fixed-axis rotation). The space of rotations around a fixed axis forms a group under the binary operator of combining rotations by performing one rotation after the other. The group is denoted by $SO(2)$ and can be identified with the circle $\mathbb{S}^1$. The group is one-dimensional, and in practice, it is often described by a single real number (the angle of rotation). This, however, neglects the fact that a full $360°$ rotation does not change the group element, although it does change the angle; this is the difference between a circle and a straight line. Instead, the group of rotations can be described by the set of special orthogonal 2×2 matrices (whence the name $SO(2)$), meaning 2×2 orthogonal matrices with determinant $+1$. These matrices have the form

$$R = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{A.21}$$

where ϕ is the angle of rotation. Together with matrix multiplication as the binary operator, and the identity matrix ($\phi = 0$) as the identity element, these matrices form a complete representation of the group of fixed-axis rotations. △

Example A.12 (spatial rotation). The space of free rotations around any axis in three dimensions forms a group, and is denoted by $SO(3)$. The group is three-dimensional, and is often represented locally by three angles, called the Euler angles, that describe three consecutive rotations around three (local) axes. Such a parameterization, however, has singularities, which results in non-smooth behavior of the coordinates around singularities. Instead, rotations can be fully and uniquely identified with the set of all special orthogonal 3×3 matrices (whence the name $SO(3)$), meaning 3×3 orthogonal matrices with determinant $+1$.

Another representation of the group of spatial rotations, not used in this thesis, is by unit quaternions. In this representation, a vector of the form

$$q = \begin{bmatrix} \cos(\frac{\theta}{2}) & n_1 \sin(\frac{\theta}{2}) & n_2 \sin(\frac{\theta}{2}) & n_3 \sin(\frac{\theta}{2}) \end{bmatrix} \tag{A.22}$$

is used to describe rotation around an axis $n = \begin{bmatrix} n_1 & n_2 & n_3 \end{bmatrix}$ with angle θ. The axis n is constrained to have unit norm (in the Euclidean sense), which means that also the vector q has unit norm. This representation is singularity free, but it doubly covers $SO(3)$, since the rotation angles α and $\alpha + 360°$ (for some α) define the same rotation, but are represented by different vectors q.

Instead of thinking of q as a unit vector in $\mathbb{R}^4$, it can also be thought of as a unit quaternion, which allows to use the mathematical structure of the space of unit quaternions. However, quaternions are not as easy to use in computations as rotation matrices, whence the choice for the matrix representation in this thesis. Still, quaternions are highly suitable for numerical implementation, and could be used when implementing the results from this thesis in software. △

Example A.13 (planar motion). We can combine the group of two-dimensional translations, *i.e.* translations in a plane, with the group of fixed-axis rotations and take the fixed axis to be orthogonal (in the Euclidean sense) to the translational plane. The resulting object is again a Lie group, and it describes all planar motions, that is, the set of all possible ways that an object can be positioned in a plane. This group is called the special Euclidean group of dimension two, denoted $SE(2)$.

As representation of this group, we can simple use a combination of a two-dimensional vector p to describe translation and a matrix R of the form (A.21) for the translation. This choice is often made in literature, but computations in this representation are cumbersome, since two consecutive motions need to be combined as

$$R_{13} = R_{23}R_{12} \qquad\qquad p_{13} = p_{23} + R_{23}p_{12} \tag{A.23}$$

which leads to long and tedious equations for multiple consecutive motions. Instead, we combine translation and rotation in one so-called homogeneous matrix of the form

$$H = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \tag{A.24}$$

where R is the rotation matrix (A.21). Consecutive planar motions can now be represented by simple matrix multiplications of the corresponding homogeneous matrices. The matrix representation of a translation as (A.20) is a special case of (A.24) for zero rotation, $R = I$. △

Example A.14 (three-dimensional motion). Similar to the planar situation, we can combine the group of translations in three dimensions $T(3)$ with the group of free-axis rotations $SO(3)$. The result is the special Euclidean group in three dimensions, or, $SE(3)$, that describes the space of all possible relative positions and orientations in three-dimensional space. It can also be represented by a matrix of the form (A.24) but now with R a three-dimensional rotation matrix, and p a three-dimensional translation vector. Again, consecutive motions are simply represented by matrix multiplication of the appropriate homogeneous matrices. △

The examples show how many useful transformations are actually Lie groups, and that these can be represented globally and without singularities by matrices with the appropriate properties. The realization that these transformations are Lie groups allows to perform certain useful operations on them.

First, because of the group structure, we can take an element of the group and combine it with another element of the group. This is called (left or right) translation, since effectively it transports one element of the group to another place, by means of group multiplication. In particular, since every element of a group has an inverse, we can transport a group element to the identity of the group. This transport can be done in two ways, either by pre- or post-multiplication with the inverse.

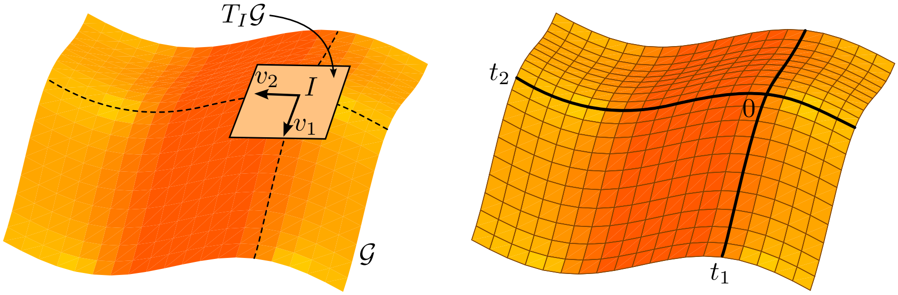
Secondly, since a Lie group has a differentiable manifold structure, we can talk about continuous and differentiable curves in the group, which represent smooth consecutive transformations, *i.e.* smooth motions of an object. The derivatives of these curves represent the (angular, linear, or combined) velocities of the moving objects.

Combining these two aspects (the group aspect and the manifold aspect), we can transport a curve $\gamma(t)$ near an element $A \in \mathcal{G}$ to a curve near the identity by applying A^{-1} to every element of the curve. We can then take the derivative of the transformed curve to obtain an element of the tangent space $T_I\mathcal{G}$ at the identity. Depending on whether left or right translation is chosen, different velocity vectors are obtained. Since in this way, velocity vectors at any point $A \in \mathcal{G}$ can be transported to the tangent space at the identity, this tangent space provides a common vector space which allows to compare and add different velocities. Furthermore, the tangent space at the identity can be given the structure of a Lie algebra by defining the appropriate Lie bracket on it. The tangent space to the identity of a group is thus usually called the Lie algebra of the group, and is denoted by $\mathfrak{g} := T_I\mathcal{G}$. Finally, as shown in Chapter 2, the tangent vectors at the identity of a group can have a clear physical interpretation, much more than tangent vectors at general points $A \in \mathcal{G}$.

A.3.2 Exponential coordinates

As a final property of Lie groups, we discuss exponential coordinates. These coordinates provide a general means to find local coordinates around any point on the Lie group. Since Lie groups allow to transport curves and tangent spaces between any points, we describe here only exponential coordinates around the identity element. The results can be directly translated to other points on the manifold.

We construct exponential coordinates using the approach illustrated in Figure A.3. As a first step, note that transporting tangent vectors from a point on the manifold to the identity can also be turned around: given a tangent vector $v \in \mathfrak{g}$,



(a) Two basis vectors at the group identity. (b) Construction of exponential coordinates.

Figure A.3: Exponential coordinates are constructed by the integral curves of left-invariant vector fields defined by basis elements at the group algebra.

we can define a vector field on the whole manifold by translating the vector to all points on the manifold. If this is done by left (right) translation, the resulting vector field is called a left (right) invariant vector field. If we focus only on left translation, and if the Lie group has a representation using square matrices, then the tangent vector (represented by a square matrix V) expands to a vector field XV at all elements $x \in \mathcal{G}$ with matrix representation X.

As a second step, we can consider the integral curves of this vector field, *i.e.* the curves $\gamma(t)$ satisfying

$$\frac{d\gamma(t)}{dt} = \gamma(t)V \tag{A.25}$$

with V constant. Equation (A.25) has an analytic solution, namely

$$\gamma(t) = e^{tV} := \sum_{i=0}^{\infty} \frac{(tV)^i}{i!} = I + tV + \frac{1}{2}t^2V^2 + \frac{1}{6}t^3V^3 + \dots \tag{A.26}$$

where we assumed without loss of generality that $\gamma(0) = I$. This equation describes the integral curve (parameterized by t) to a left-invariant vector field generated by an element of the Lie algebra. Furthermore, it can be shown that the exponential mapping provides a local diffeomorphism between elements $tV \in \mathfrak{g}$ around the zero vector and a neighborhood of $I \in \mathcal{G}$.

The third and final step to exponential coordinates is to choose a basis for the Lie algebra (which is a real vector space, and hence it has a basis), *i.e.* to choose nonzero elements $V_i \in \mathfrak{g}$ such that any tangent vector V can be expressed as a linear combination

$$V = \sum_{i=1}^{n} t_i V_i \tag{A.27}$$

for parameters $t_i \in \mathbb{R}$. Locally around $t = 0$, the exponential mapping then provides a local coordinate chart, which uniquely associates an element of the group to the set of coordinates t_i, namely as

$$\{t_i\} \in \mathbb{R}^n \quad \rightarrow \quad e^{\sum_{i=1}^{n} t_i V_i} \in \mathcal{G} \tag{A.28}$$

These coordinates $\{t_i\}$ are called exponential coordinates. Although they are only local coordinates, and although actually computing the mapping (A.28) can be cumbersome, exponential coordinates can still be useful in mathematical proofs, since they provide a representation of a general Lie group locally as $\mathbb{R}^n$. Computations can be performed on the exponential coordinates (using all available knowledge about $\mathbb{R}^n$) and then be transformed to the original representation in constrained matrices.

This approach can be used in Section 2.3, more specifically in Definition 2.9, which defines a globally parameterized rigid joint. In this definition, the joint is required to have a twice differentiable function $F_Q(\phi)$ assigning local coordinates $\phi \in \mathbb{R}^k$ around every allowed configuration $H_i^j(Q)$, parameterized by the matrix Q. Furthermore, it requires that the allowed relative twists $T_i^{i,j}$ (an element of the Lie algebra of $SE(3)$) be parameterized as $T_i^{i,j} = X(Q)v$, with $v = V_Q(\dot{Q})$ a linear function of $\dot{Q}$.

If the configuration space of the joint can be described by a proper subgroup of $SE(3)$, we can use the exponential mapping (A.28) as the coordinate mapping F, with corresponding coordinates $\phi_i = t_i$ as local coordinates, and $v_i = \dot{\phi}_i = \dot{t}_i$ as the velocities. We now consider the general case that the subgroup is in fact $SE(3)$ itself, and hence that $H_i^j(Q) = Q$. This means that we can choose the coordinates ϕ to be exponential coordinates around a fixed point $H_0 \in SE(3)$, and $F_Q(\phi)$ the exponential mapping, such that around H_0

$$H_i^j(\phi) = H_0 e^{\sum_k \phi_k V_k} \tag{A.29}$$

with V_k a fixed basis for $se(3)$. For joints that are lower-dimensional Lie groups in $SE(3)$, we constrain some ϕ_k to be zero, and thus generate only part of $SE(3)$ around H_0. In any case, the exponential mapping is clearly twice differentiable. Second, we can compute the velocity $\dot{H}_i^j$ in terms of $\dot{\phi}$ as

$$\begin{aligned}\dot{H}_i^j &= H_0 \frac{d}{dt}\left(e^A\right) = H_0 \frac{d}{dt}\left(I + A + \frac{1}{2}A^2 + \frac{1}{6}A^3 + \dots\right) \\ H_0^{-1}\dot{H}_j^i &= \dot{A} + \frac{1}{2}\left(\dot{A}A + A\dot{A}\right) + \frac{1}{6}\left(\dot{A}A^2 + A\dot{A}A + A^2\dot{A}\right) + \dots\end{aligned} \tag{A.30}$$

where we defined $A := \sum_k \phi_k V_k$ and hence $\dot{A} = \sum_k \dot{\phi}_k V_k$. The expression for $\dot{H}_i^j$ can hence be rewritten as a linear combination of $\dot{\phi}$, which is invertible around

H_0 (for $\phi \approx 0$), since the matrices V_k are linearly independent. Furthermore, we can compute the twist as

$$\begin{aligned}\tilde{T}_i^{i,j} &= H_j^i \dot{H}_i^j = e^{-A} H_0^{-1} \dot{H}_i^j \\ &= \left(I - A + \frac{1}{2}A^2 - \dots\right)\left(\dot{A} + \frac{1}{2}(\dot{A}A + A\dot{A}) + \frac{1}{6}(\dot{A}A^2 + A\dot{A}A + A^2\dot{A}) + \dots\right) \\ &= \dot{A} - \frac{1}{2}(A\dot{A} - \dot{A}A) + \frac{1}{6}(\dot{A}A^2 - 2A\dot{A}A + A^2\dot{A}) - \dots \qquad \text{(A.31)}\end{aligned}$$

It can be proved (Rossmann 2002) that this sequence actually equals

$$\tilde{T}_i^{i,j} = \sum_{l=0}^{\infty} \frac{(-1)^l}{(l+1)!} \operatorname{ad}_A^l \dot{A} \qquad \text{(A.32)}$$

with $\operatorname{ad}_X Y = [X, Y]$ the matrix commutator and $\operatorname{ad}_X^{k+1} Y := [X, \operatorname{ad}_X^k Y]$, but we only need the first few terms of this expression for Theorem 2.17, so we skip that proof here. If we choose the basis for $se(3)$ to be the standard Euclidean vector basis with corresponding matrix representation by the tilde operator defined in Definition 2.6, then we can write (A.32) in vector form as

$$T_i^{i,j} = \sum_{l=0}^{\infty} \frac{(-1)^l}{(l+1)!} \operatorname{ad}_\phi^l \dot{\phi} = \left(I - \frac{1}{2}\operatorname{ad}_\phi + \frac{1}{6}\operatorname{ad}_\phi^2 - \dots\right)\dot{\phi} \qquad \text{(A.33)}$$

where ad_ϕ is now taken as in Lemma 2.8. This shows that indeed, for Lie groups that are subgroups of $SE(3)$, exponential coordinates provide a system of local coordinates around an arbitrary element of the group that satisfy the requirements of Definition 2.9.

Appendix B

Port-Hamiltonian Systems and Bond Graphs

The framework of port-Hamiltonian systems is the theoretical modeling and control framework that is used mostly in this thesis. We present a brief discussion of the structure and properties of port-Hamiltonian system, as well as a brief overview of the graphical representation language of bond graphs, which is highly suitable to represent port-Hamiltonian systems. More details on port-Hamiltonian systems can be found, for example, in van der Schaft (2000), and on bond graphs in Karnopp et al. (1999).

B.1 Port-Hamiltonian systems

Port-Hamiltonian systems were first introduced by Maschke & van der Schaft (1992), then by the name of Port-Controlled Hamiltonian Systems. The framework of port-Hamiltonian systems is based on the idea of modeling energy flows inside systems, as opposed to information flows as in most traditional frameworks. It is aimed specifically at modeling physical systems, since energy plays an important role in these systems, and representing it explicitly can help in understanding them.

The concept of energy is present in all physical domains, and hence port-Hamiltonian systems can be used to model systems in these domains, as well as the interconnection of systems from different domains. In each domain, the flow of energy is characterized by two variables, called power-conjugate variables, which are elements of a vector space V and its dual V^* (see Appendix A.1), and whose dual product is equal to a physical power flow. For example, in mechanics, these variables are the collocated force and velocity (or torque and angular velocity) at a certain point, and in electric circuit theory, they are the voltage across and the current through a certain element. For a general domain, these

Table B.1: Choice of effort and flow variables in several physical domains.

physical domain	effort	flow
mechanical, translation	force	velocity
mechanical, rotation	torque	angular velocity
mechanical, spatial	wrench	twist
electrical circuits	voltage	current
magnetic	current	voltage
hydraulic	pressure	volume flow
thermal	temperature	entropy flow

variables are called *effort* and *flow*, and the standard choice of what variables are effort and flow is shown in Table B.1.

From the two power variables in a domain, we can obtain two energy variables by integration of the power variables. If we integrate the effort, we obtain as the energy variable the *generalized momentum*, by analogy of the integration of the force on a mass which yields its momentum. If we integrate the flow, we obtain as an energy variable the *generalized displacement*, by analogy of the integration of the velocity of a spring, which yields its displacement. However, the displacement state does not exist for the thermal domain.

▼ The integration of power variables should be done in a mathematically meaningful way. On $\mathbb{R}^n$, the integrals can be performed just by the usual integration, but *e.g.* on Lie groups, where the power variables are chosen to be in the Lie algebra or its dual, these power variables should first be translated to the tangent space of the appropriate group element, before integrating them. ▲

An energy function, now, is a mapping from the appropriate space of energy variables to $\mathbb{R}$. For example, the energy function for a point mass m is a mapping from the momentum variable p to $\mathbb{R}$ as $\frac{1}{2m}p^2$. This formal definition allows to distinguish between energy functions (functions of the energy variables), and co-energy functions (functions of the power variables). An example of a co-energy function is the kinetic co-energy of a point mass $\frac{1}{2}mv^2$, which is a function of the power variable v, the velocity. The energy and co-energy functions are related by the Legendre transformation.

Port-Hamiltonian systems are usually defined implicitly in a process depicted intuitively in Figure B.1. The central object under consideration is a state space manifold $\mathcal{X}$. On this manifold, a smooth energy function H is defined which assigns to each state $x \in \mathcal{X}$ an energy value. Vectors and co-vectors attached to a state x are identified with the rate of change $\dot{x}$ and the differential of the energy function dH, respectively. The dual product between the vectors and co-vectors hence equals the change of stored energy $\dot{H}$.

In order to turn such a Hamiltonian system into a *port*-Hamiltonian system,

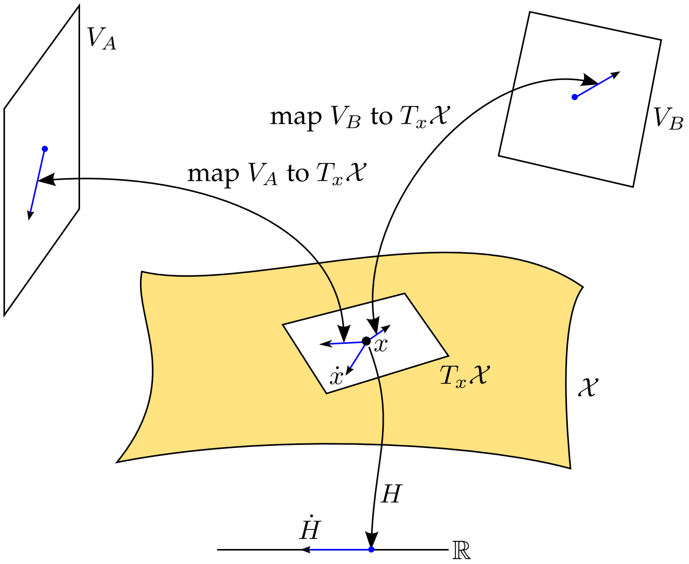


Figure B.1: Intuitive representation of the construction of a port-Hamiltonian system by mapping vector and co-vectors to the tangent and co-tangent space at $x \in \mathcal{X}$. Only the tangent and vector spaces is shown, not their duals.

input structures need to be defined that make the system open for interconnection. These input structures are called *power-ports*, and they are defined by a vector space and its dual, in such a way that the dual product between elements of the two spaces describes physical power flow. Hence, power-ports can be defined by choosing the appropriate vector spaces and their duals. These ports can be left open (allowing interconnection to other systems), or they can be terminated, for example, by a static dissipation relation that allows only certain combinations of vectors and co-vectors, satisying (among other things) that their dual product (the associated power) is non-negative.

Finally, in addition to the power-ports, the interconnection structure is defined. This structure (called the *Dirac structure*) describes how the power variables at the external power-ports need to be translated to the tangent and co-tangent space at x, and how all resulting elements in these spaces need to be related. This structure is power-continuous, such that energy is conserved in the system: the sum of the power coming in through all external power-ports is equal to the change of internal energy $\dot{H}$.

A port-Hamiltonian system described in this way contains only relations between variables, no explicit input-output formulations. This is useful to allow arbitrary interconnection to other systems without having to worry about causality, but for simulation and analysis, it is often convenient to use an input-output formulation. In this thesis, almost all port-Hamiltonian systems are formulated explicitly, *i.e.* as systems of the following form.

Definition B.1 (explicit port-Hamiltonian system). *An explicit port-Hamiltonian system is a system with state $x \in \mathcal{X}$, energy $H(x) : \mathcal{X} \to \mathbb{R}$, power port (u, y), and dynamic equations of the form*

$$\begin{aligned} \dot{x} &= \big(J(x) - R(x)\big)\frac{\partial H}{\partial x} + g(x)u \\ y &= g^T(x)\frac{\partial H}{\partial x} + \big(K(x) + S(x)\big)u \end{aligned} \tag{B.1}$$

with $J(x)$ and $K(x)$ skew-symmetric, and $R(x)$ and $S(x)$ positive semi-definite.

This explicit formulation is a special case of the general implicit formulation. The power-port (u, y) is mapped to the tangent space at x by the mapping $g(x)$, dissipation is applied directly at the tangent and co-tangent space at x (by the matrix $R(x)$) and at the input port (u, y) (by the matrix $S(x)$). Finally, the Dirac structure is defined by the matrices $J(x)$, $K(x)$, together with the mappings $g(x)$ and $g^T(x)$.

We discuss two properties of port-Hamiltonian systems that follow immediately from the structure of the representation (B.1). First, we can compute the change of internal energy $\dot{H}$ as

$$\begin{aligned} \frac{d}{dt}H(x) &= \frac{\partial^T H}{\partial x}\dot{x} = \frac{\partial^T H}{\partial x}\left(\big(J(x) - R(x)\big)\frac{\partial H}{\partial x} + g(x)u\right) \\ &= -\frac{\partial^T H}{\partial x}R(x)\frac{\partial H}{\partial x} + \left(\frac{\partial^T H}{\partial x}g(x) + u^T(K(x) + S(x) - K(x) - S(x))\right)u \\ &= -\frac{\partial^T H}{\partial x}R(x)\frac{\partial H}{\partial x} - u^T S(x)u + y^T u \end{aligned} \tag{B.2}$$

where we used the fact that $J(x)$ and $K(x)$ are skew-symmetric. Equation (B.2) shows that the increase in internal energy is equal to the power $y^T u$ supplied through the input port, minus the power lost by dissipation through $R(x)$ and $S(x)$. In particular, since $R(x)$ and $S(x)$ are positive semi-definite, the increase in internal energy is always less than or equal to the supplied power. This means that port-Hamiltonian systems are passive with respect to the supply rate $y^T u$ and storage function H (see van der Schaft (2000) for more details on passivity).

The second important property is that the port-interconnection of two port-Hamiltonian systems is again a port-Hamiltonian system. We can prove this simply by combining the equations for two port-Hamiltonian systems as shown in

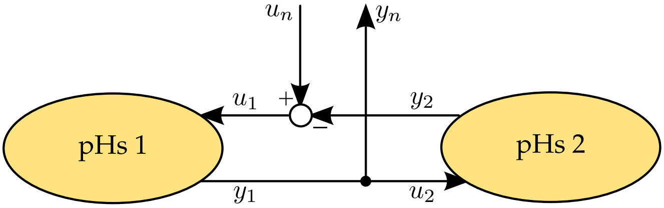


Figure B.2: The power-continuous interconnection of two port-Hamiltonian systems is again a port-Hamiltonian system.

Figure B.2 and defined by the following equation.

$$\begin{aligned} y_1 &= u_2 = y_n \\ u_1 &= u_n - y_2 \end{aligned} \tag{B.3}$$

Such an interconnection is power-continuous, *i.e.* the external power $u_n^T y_n$ flowing into the system always instantaneously equals the sum of power flowing into the two subsystems. To avoid ill-defined interconnections and the corresponding algebraic loops, we assume here that the first system is strictly proper, *i.e.* that the two systems have the form

$$\begin{aligned} \dot{x}_1 &= (J_1 - R_1)\frac{\partial H_1}{\partial x_1} + g_1 u_1 & \dot{x}_2 &= (J_2 - R_2)\frac{\partial H_2}{\partial x_2} + g_2 u_2 \\ y_1 &= g_1^T \frac{\partial H_1}{\partial x_1} & y_2 &= g_2^T \frac{\partial H_2}{\partial x_2} + (K_2 + S_2) u_2 \end{aligned} \tag{B.4}$$

Manipulating (B.4) using the interconnection relations (B.3) results in the following expression

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} J_1 - R_1 - g_1(K_2 + S_2)g_1^T & -g_1 g_2^T \\ g_2 g_1^T & J_2 - R_2 \end{bmatrix} \begin{bmatrix} \frac{\partial H_n}{\partial x_1} \\ \frac{\partial H_n}{\partial x_2} \end{bmatrix} + \begin{bmatrix} g_1 \\ 0 \end{bmatrix} u_n \\ y_n &= \begin{bmatrix} g_1^T & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial H_n}{\partial x_1} \\ \frac{\partial H_n}{\partial x_2} \end{bmatrix} \end{aligned} \tag{B.5}$$

where $H_n(x_1, x_2) := H_1(x_1) + H_2(x_2)$ and (u_n, y_n) is the new input port. Equations (B.5) show that this interconnection of two port-Hamiltonian systems is again a port-Hamiltonian system of the form (B.1), with energy equal to the sum of the energies of the subsystems. Furthermore, the interconnection does not change the energy balance. For example, if the two original systems had no dissipation ($R_1 = R_2 = S_2 = 0$), then their interconnection does not have dissipation either. Different types of power-continuous interconnections give similar equations with the same conclusions.

Finally, we name a few special cases of port-Hamiltonian systems that are encountered often in practical modeling tasks.

- Energy-conserving port-Hamiltonian systems are port-Hamiltonian systems with $R(x) = S(x) = 0$, which means that for any state x and any input (u, y), the power supplied by the input port is exactly equal to the increase in internal energy;
- Strictly proper port-Hamiltonian systems are systems with $K = S = 0$, which means that there is no direct coupling between input u and output y, and the transfer functions (for linear systems) from input to output are strictly proper;
- Power-continuous port-Hamiltonian systems are of the form $y = Ku$ (so without internal state or energy) that do not store or dissipate energy, but only transform it in a power-continuous way.

B.2 Bond graphs

Bond graphs, introduced by Paynter (1961), are a graphical language for representing power-interconnections of physical elements. For this reason, bond graphs are a very suitable way to graphically represent port-Hamiltonian systems, as studied in detail, for example, by Golo (2002).

Definition B.2 (oriented graph). *An oriented graph is a set V of vertices, together with a set $E \subset V \times V$ of edges. An element $(v_i, v_j) \in E$ is called an (oriented) edge joining v_i to v_j.*

Bond graphs are a special type of oriented graph, in which the edges, now called bonds, represent a power-interconnection between two connected nodes. Each bond has two collocated power variables associated to it, one effort and one flow. The edge is a single line for one-dimensional power variables, and a double line for multi-dimensional power variables. A half-arrow or harpoon at one side of the bond indicates the reference direction for positive power: if the dual product of effort and flow is positive, then power flows in the direction of the half-arrow. Furthermore, a stroke on either side of the bond indicates the direction of computation, or, causality: the effort signal travels in the direction of the stroke, and hence the flow travels in the opposite direction. Making a bond graph causal (*i.e.* putting a causal stroke on each bond in a sensible way, to be discussed later) allows to write down the explicit port-Hamiltonian equations corresponding to the bond graph.

The nodes in bond graphs can represent one of several basic functions, namely storage, dissipation, source, and power-continuous interconnection. Figure B.3

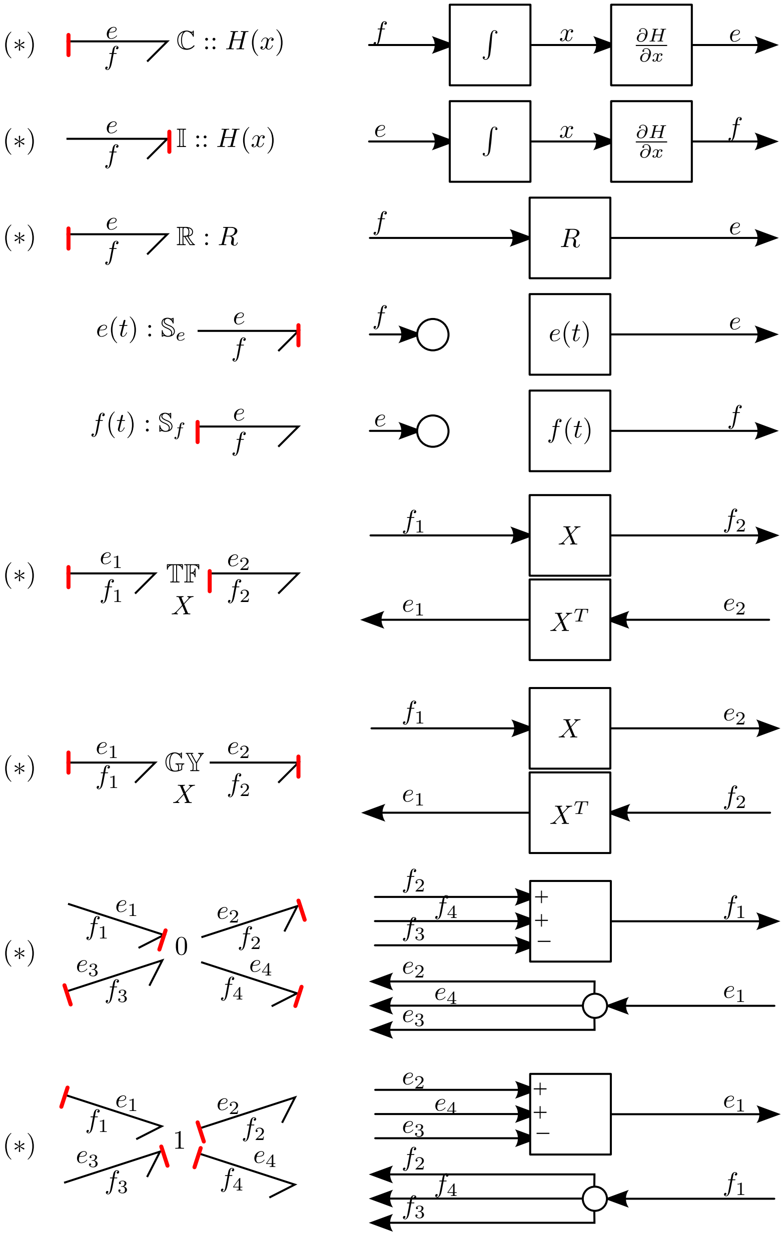


Figure B.3: Several bond graph elements and their equivalent block diagrams. Elements marked by (∗) have more possibilities for the causality; one is shown.

shows several often-used bond graph elements, together with their equivalent block diagrams (when in causal form). Next to the symbol for the element, some other expression can appear, separated by a colon or double colon. This expression indicates the value of the parameter that defines that element, or, in the case of a double colon, the energy function associated with the element (only valid for storage elements).

A storage element integrates one of the power variables on the connected bond to an energy variable x (its state), and computes the other power variable as the partial derivative of an energy function H to the state. Its dynamics are given by the equation

$$\begin{aligned} \dot{x} &= u \\ y &= \frac{\partial H}{\partial x} \end{aligned} \tag{B.6}$$

where (u, y) defines its input power-port. If the input u is an effort (and hence the output y is a flow), the element is denoted by the symbol $\mathbb{I}$ (for inertial), while if the input is a flow (and hence the output an effort), the element is denoted by the symbol $\mathbb{C}$ (for capacitor). Indeed, the inertial and capacitive element are of the appropriate type, when the efforts and flows are chosen as in Table B.1. Storage elements can have multiple connecting ports, even connecting multiple domains, and their energy functions are then functions of multiple energy variables. Still, the energy variables are computed by integrating one of the power variables on the bond, and the other collocated power variable equals the partial derivative of the energy function with respect to this energy variable.

A dissipative element is denoted by the symbol $\mathbb{R}$ (resistive) and is defined by a static relation between the connected effort and flow such that $e^T f \geq 0$ at all times. Hence, power always flows towards the $\mathbb{R}$ element, making it represent dissipation, or better, irreversible transduction.

A source element determines one of the power variables on the bond, irrespective of the other power variable. An effort source (denoted $\mathbb{S}_e$) fixes the effort value, while a flow source (denoted $\mathbb{S}_f$) fixes the flow value. Sources can be used to model systems with negligible back effect, such as the connection of a (good) battery to a load with a high input impedance: the voltage is fully determined by the battery, irrespective of the current that the load draws.

Power-continuous interconnections can take several forms: transformers, gyrators, and junctions. A transformer $\mathbb{TF}$ is connected by two bonds (one incoming, one outgoing), and is defined by a static linear relation between the efforts on the two bonds in one direction as well as the same but transposed relation between the two flows in the other directions. In other words, either $f_2 = Xf_1$ and $e_1 = X^T e_2$, or $f_1 = Xf_2$ and $e_2 = X^T e_1$, for some matrix X. In either case, it follows immediately that the power $e_1^T f_1$ on one bond is equal to the power $e_2^T f_2$ on the other bond.

A gyrator is a similarly defined power-continuous interconnection element, and it is denoted by the symbol $\mathbb{GY}$. A gyrator is also connected by two bonds (one incoming, one outgoing), but now the effort variable on one bond is statically related to the flow variable on the other bond. So in this case, either $e_2 = Xf_1$ and $e_1 = X^T f_2$, or $f_1 = Xe_2$ and $f_2 = X^T e_1$. A gyrator can also be connected by only one bond, in which case the effort and flow on this bond are related as $e = Yf$ or $f = Ye$ with Y skew-symmetric, such that the instantaneous power is always $e^T f = 0$. Obviously, this only makes sense for multi-dimensional bonds.

Finally, junctions are power-continuous elements that can take any number of bonds. The power variables on these bonds are constrained by the junction in the following way: one power-variable must be equal on all connected bonds, and the other power-variables on the bonds must sum up to zero, taking into account the direction of the bond. These constraints ensure that the total instantaneous power flowing into the junction is always zero. A 1-junction is constrained to have the flows on all bonds equal, and a 0-junction is constrained to have the efforts on all bonds equal.

All static elements can be modulated by external signals, for example, to change the transformation ratio in a transformer, or to represent state-dependent centrifugal forces by a modulated gyrator. If an element is modulated, an extra $\mathbb{M}$ is put on front of the symbol, such as $\mathbb{MTF}$ for a modulated transformer. Note that storage elements cannot be modulated directly, since this would mean that the stored energy can be changed even though no power is flowing through the power-port. This would violate the whole idea of representing an energy balance as a bond graph, and hence storage elements should only be changed through power-ports. Stramigioli & Duindam (2001) show an example of how the rest length and center of stiffness of a spatial spring can be changed through the appropriate power-port.

If desired, *e.g.* to derive the explicit dynamic equations, a bond graph can be made causal, which means that a causal stroke is placed at one side of each bond. These strokes should be placed in such a way, that the relations corresponding to each element can be written as explicit assignment statements. This means, for example, that only one of the efforts at a 0-junction can be the input, and that if one bond on a transformer has an incoming effort, then the other bond must have an outgoing effort. Systematic (and automated) methods exist to assign causality to a bond graph, see *e.g.* Golo (2002).

The bond graph of a physical system can be built up by interconnection of the appropriate bond graph elements. A general explicit port-Hamiltonian system of the form of Definition B.1 is given by the bond graph of Figure B.4, where we assumed the input vector u to be an effort. Generally, though, larger physical systems possess more structure in the energy function or the interconnection, and this can be exploited and displayed in the bond graph.

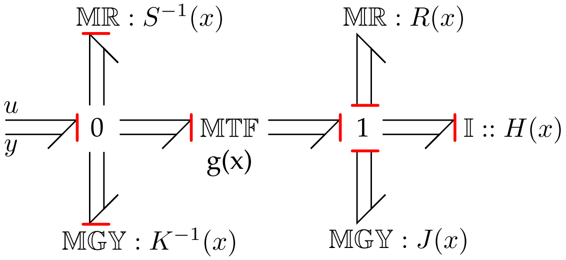


Figure B.4: Bond graph of the general explicit port-Hamiltonian system (B.1).

Bibliography

Acary, V. & Brogliato, B. (2003), Towards a Multiple Impact Law: the 3-Ball Example, *in* 'Actes du Sixième Colloque National de Calculs des Structures', Vol. 2, pp. 337–344.

Alexander, R. M. (1990), 'Three Uses for Springs in Legged Locomotion', *The International Journal of Robotics Research* **9**(2), pp. 53–61.

Ambrose, R. O., Aldridge, H., Askew, R. S., Burridge, R. R., Bluethmann, W., Diftler, M., Lovchik, C., Magruder, D. & Rehnmark, F. (2000), 'Robonaut: NASA's Space Humanoid', *IEEE Intelligent Systems* **15**(4), pp. 57–63.

Ambrosius, F. (2005), Interpolation of 3D Surfaces for Contact Modeling. B.Sc. report, University of Twente.

Asano, F., Luo, Z.-W. & Yamakita, M. (2004), Some Extensions of Passive Walking Formula to Active Biped Robots, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', Vol. 4, pp. 3797–3802.

Ascher, U. M. & Petzold, L. R. (1998), *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Soc. for Industrial and Applied Math.

Beekman, N. (2004), Analysis and Development of a 2D walking machine, Master's thesis, University of Twente.

Blankenstein, G. (2003), Symmetries and Locomotion of a 2D Mechanical Network: the Snakeboard, *in* 'Lecture Notes for the Euron/GeoPleX Summer School', Bertinoro, Italy.

Bloch, A. M. (2003), *Nonholonomic Mechanics and Control*, Interdisciplinary Applied Mathematics (24), Springer-Verlag.

Bloch, A. M., Krishnaprasad, P. S., Marsden, J. E. & Murray, R. M. (1996), 'Nonholonomic Mechanical Systems with Symmetry', *Archive for Rational Mechanics and Analysis* **136**, pp. 21–99.

Breazeal, C. (2003), 'Towards Sociable Robots', *Robotics and Autonomous Systems* **42**, pp. 167–175.

Breedveld, P. C. (2004), 'Port-based Modeling of Mechatronic Systems', *Mathematics and Computers in Simulation* **66**, pp. 99–127.

Bullo, F. & Zefran, M. (2001), 'On Mechanical Control Systems with Nonholonomic Constraints and Symmetries', *Systems and Control Letters* **45**(1), pp. 133–143.

Burke, W. L. (1985), *Applied Differential Geometry*, Cambridge University Press.

Canudas de Wit, C., Olsson, H., Åström, K. J. & Lischinsky, P. (1995), 'A New Model for Control of Systems with Friction', *IEEE Transactions on Automatic Control* **40**(3), pp. 419–425.

Chatterjee, A. & Ruina, A. (1998), 'Two Interpretations of Rigidity in Rigid Body Collisions', *Journal of Applied Mechanics* **65**(4), pp. 894–900.

Chevallereau, C. (2003), 'Time-Scaling Control for an Underactuated Biped Robot', *IEEE Transactions on Robotics and Automation* **19**(2), pp. 362–368.

Cohen, J. D., Lin, M. C., Manocha, D. & Ponamgi, M. (1995), I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments, *in* 'Symposium on Interactive 3D Graphics', pp. 189–218.

Collins, S. H., Wisse, M. & Ruina, A. (2001), 'A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees', *International Journal of Robotics Research* **20**(7), pp. 607–615.

Control Lab Products (2005), '20sim Version 3.6'.
URL: *http://www.20sim.com*

Cottle, R. W., Pang, J.-S. & Stone, R. E. (1992), *The Linear Complementarity Problem*, Computer Science and Scientific Computing, Academic Press.

Dertien, E. C. (2005), Realisation of an Energy-Efficient Walking Robot, Master's thesis, University of Twente.

Dubrovin, B. A., Fomenko, A. T. & Novikov, S. P. (1984), *Modern Geometry — Methods and Applications, Vol. I*, Graduate Texts in Mathematics 93, Springer-Verlag.

Dubrovin, B. A., Fomenko, A. T. & Novikov, S. P. (1985), *Modern Geometry — Methods and Applications, Vol. II*, Graduate Texts in Mathematics 104, Springer-Verlag.

Duindam, V. (2006), Port-Based Modeling and Control for Efficient Bipedal Walking Robots, PhD thesis, University of Twente. ISBN 90-365-2318-4.
URL: *http://purl.org/utwente/50829*

Duindam, V., Blankenstein, G. & Stramigioli, S. (2004), Port-Based Modeling and Analysis of Snakeboard Locomotion, *in* 'Proceeding of the International Symposium on Mathematical Theory of Networks and Systems'.

Duindam, V. & Stramigioli, S. (2003*a*), Modeling the Kinematics and Dynamics of Compliant Contact, *in* 'Proceedings of the International Conference on Robotics and Automation', pp. 4029–4034.

Duindam, V. & Stramigioli, S. (2003*b*), Passive Asymptotic Curve Tracking, *in* 'Proceedings of the IFAC Workshop on Lagr. and Hamilt. Methods for Nonlinear Control', pp. 229–234.

Duindam, V. & Stramigioli, S. (2004*a*), Energy-Based Model-Reduction and Control of Nonholonomic Mechanical Systems, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', pp. 4584–4589.

Duindam, V. & Stramigioli, S. (2004*b*), 'Port-Based Asymptotic Curve Tracking for Mechanical Systems', *European Journal of Control* **10**(5), pp. 411–420.

Duindam, V. & Stramigioli, S. (2005*a*), Optimization for Mass and Stiffness Distribution for Efficient Bipedal Walking. Presented at the IROS workshop on Morphology, Control, and Passive Dynamics.

Duindam, V. & Stramigioli, S. (2005*b*), Optimization for Mass and Stiffness Distribution for Efficient Bipedal Walking, *in* 'Proceedings of the International Symposium on Nonlinear Theory and Its Applications'.

Duindam, V. & Stramigioli, S. (2005*c*), Port-Based Control of a Compass-Gait Bipedal Robot, *in* 'Proceedings of the 16th IFAC World Congress'. Electronic proceedings.

Duindam, V., Stramigioli, S. & Scherpen, J. M. A. (2004), 'Passive Compensation of Nonlinear Robot Dynamics', *IEEE Transactions on Robotics and Automation* **20**(3), pp. 480–487.

Ekkelenkamp, R., Veneman, J. & van der Kooij, H. (2005), LOPES : Selective Control of Gait Functions During the Gait Rehabilitation of CVA Patients, *in* 'Proceedings of the 9th IEEE International Conference on Rehabilitation Robotics', pp. 361–364.

Fasse, E. D. (1997), 'On the Spatial Compliance of Robotic Manipulators', *ASME Journal of Dynamic Systems, Measurement and Control* **119**, pp. 839–844.

Fasse, E. D. (2000), Some Applications of Screw Theory to Lumped-Parameter Modeling of Visco-Elastically Coupled Bodies, *in* 'Proceedings of the Ball Symposium'.

Flügge, W. (1975), *Viscoelasticity*, second edn, Springer-Verlag.

Garcia, M., Chatterjee, A. & Ruina, A. (1998), Speed, Efficiency, and Stability of Small-Slope 2D Passive-Dynamic Bipedal Walking, *in* 'Proceedings of the IEEE Conference on Robotics and Automation', pp. 2351–2356.

Garcia, M., Chatterjee, A. & Ruina, A. (2000), 'Efficiency, Speed and Scaling of Two-Dimensional Passive Dynamic Walking', *Dynamics and Stability of Systems* **15**(2), pp. 75–99.

Geppert, L. (2004), 'Qrio, the Robot that Could', *IEEE Spectrum* **41**(5), pp. 34–37.

Gilmore, R. (1974), *Lie Groups, Lie Algebras, and Some of Their Applications*, John Wiley & Sons.

Glocker, C. (2001), 'On Frictionless Impact Models in Rigid-Body Systems', *Philosophical Transactions of the Royal Society London* **359**(1789), pp. 2385–2404.

Glocker, C. (2004), 'Concepts for Modeling Impacts without Friction', *Acta Mechanica* **168**, pp. 1–19.

Goldstein, H. (1980), *Classical Mechanics*, second edn, Addison-Wesley.

Golo, G. (2002), Interconnection Structures in Port-Based Modeling: Tools for Analysis and Simulation, PhD thesis, University of Twente.

Gomes, M. & Ruina, A. (2005), A Walking Model with No Energy Cost. In revision, J. of Theor. Biology.

Goswami, A., Thuilot, B. & Espiau, B. (1998), 'A Study of the Passive Gait of a Compass-like Biped Robot: Symmetry and Chaos', *International Journal of Robotics Research* **17**(12), pp. 1282–1301.

Harwin, W. S., Rahman, T. & Foulds, R. A. (1995), 'A Review of Design Issues in Rehabilitation Robotics with Reference to North American Research', *IEEE Transactions on Rehabilitation Engineering* **3**(1), pp. 3–13.

Hogan, N. (1985), 'Impedance Control: An Approach to Manipulation', *Journal of Dynamical Systems, Measurement, and Control* **107**(1), pp. 1–24.

Hubbard, P. M. (1996), 'Approximating Polyhedra with Spheres for Time-Critical Collision Detection', *ACM Transactions on Graphics* **15**(3), pp. 179–210.

Hunt, K. H. & Crossley, F. R. E. (1985), 'Coefficient of Restitution Interpreted as Damping in Vibroimpact', *ASME Journal of Applied Mechanics* **2**(3), pp. 289–307.

Hurst, J. W., Chestnutt, J. & Rizzi, A. (2004), An Actuator with Physically Variable Stiffness for Highly Dynamic Legged Locomotion, *in* 'Proceedings of the International Conference on Robotics and Automation', Vol. 5, pp. 4662–4667.

Jiménez, P., Thomas, F. & Torras, C. (2001), '3D Collision Detection: a Survey', *Computers and Graphics* **25**(2), pp. 269–285.

Johnson, K. L. (1985), *Contact Mechanics*, Cambridge University Press.

Kanda, T., Hirano, T. & Eaton, D. (2004), 'Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial', *Human-Computer Interaction* **19**, pp. 61–84.

Karnopp, D. C., Margolis, D. L. & Rosenberg, R. C. (1999), *System Dynamics: Modeling and Simulation of Mechatronic Systems*, third edn, Wiley-Interscience.

Kim, J.-Y., Park, I.-W., Lee, J., Kim, M.-S., Cho, B.-K. & Oh, J.-H. (2005), System Design and Dynamic Walking of Humanoid Robot KHR-2, *in* 'Proceedings of the IEEE Conference on Robotics and Automation', pp. 1443–1448.

Klosowski, J. T., Held, M., Mitchell, J. S. B., Sowizral, H. & Zikan, K. (1998), 'Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs', *Transactions on Visualization and Computer Graphics* **4**(1), pp. 21–36.

Koditschek, D. E. & Bühler, M. (1991), 'Analysis of a Simplified Hopping Robot', *The International Journal of Robotics Research* **10**(6), pp. 587–605.

Kuo, A. D. (1999), 'Stabilization of Lateral Motion in Passive Dynamic Walking', *The International Journal of Robotics Research* **18**(9), pp. 917–930.

Lay, D. C. (2002), *Linear Algebra and Its Applications*, 3rd edn, Addison Wesley.

Lewis, A., Ostrowski, J., Murray, R. M. & Burdick, J. W. (1994), Nonholonomic Mechanics and Locomotion: the Snakeboard Example, *in* 'Proceedings of the IEEE Conference on Robotics and Automation'.

Li, P. Y. & Horowitz, R. (1995), Passive Velocity Field Control of Mechanical Manipulators, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', pp. 2764–2770.

Li, P. Y. & Horowitz, R. (1999), 'Passive Velocity Field Control of Mechanical Manipulators', *IEEE Transactions on Robotics and Automation* **15**(4), pp. 751–763.

Lipkin, H. (1985), Geometry and Mappings of Screws with Applications to the Hybrid Control of Robotic Manipulators, PhD thesis, University of Florida.

Lončarić, J. (1985), Geometrical Analysis of Compliant Mechanisms in Robotics, PhD thesis, Harvard University, Cambridge (MA).

Marsden, J. E. & Ratiu, T. S. (1999), *Introduction to Mechanics and Symmetry*, Texts in Applied Mathematics (17), second edn, Springer-Verlag.

Maschke, B. M. & van der Schaft, A. J. (1992), Port-Controlled Hamiltonian Systems: Modelling Origins and System-Theoretic Properties, *in* 'IFAC Symposium on Nonlinear Control Systems', pp. 282–288.

McGeer, T. (1989), Powered Flight, Childś Play, Silly Wheels, and Walking Machines, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', Vol. 3, pp. 1592–1597.

McGeer, T. (1990*a*), 'Passive Dynamic Walking', *The International Journal of Robotics Research* **9**(2), pp. 62–82.

McGeer, T. (1990*b*), Passive Walking with Knees, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', Vol. 3, pp. 1640–1645.

McGeer, T. (1991), Passive Dynamic Biped Catalogue, *in* 'Proc. 2nd Int. Symp. of Experimental Robotics', Springer-Verlag, pp. 465–490.

McGeer, T. (1993), 'Dynamics and Control of Bipedal Locomotion', *Journal of Theoretical Biology* **163**(3), pp. 277–314.

Milnor, J. (1978), 'Analytic Proof of the Hairy Ball Theorem and the Brouwer Fixed Point Theorem', *American Mathematics Monthly* **85**(7), pp. 521–524.

Montana, D. J. (1989*a*), 'The Kinematics of Compliant Contact and Grasp', *International Journal of Robotics Research* **7**(3), pp. 17–32.

Montana, D. J. (1989*b*), The Kinematics of Contact with Compliance, *in* 'Proceedings of the IEEE Conference on Robotics and Automation', pp. 770–774.

Murray, R. M., Li, Z. & Sastry, S. S. (1994), *A Mathematical Introduction to Robotic Manipulation*, CRC Press.

Ortega, R., van der Schaft, A. J., Mareels, I. & Maschke, B. M. (2001), 'Putting Energy Back in Control', *IEEE Control Systems Magazine* **21**(2), pp. 18–33.

Ostrowski, J. P. (1999), 'Computing Reduced Equations for Robotic Systems with Constraints and Symmetries', *IEEE Transactions on Robotics and Automation* **15**(1), pp. 111–123.

Ostrowski, J. P. & Burdick, J. W. (1998), 'The Geometric Mechanics of Undulatory Robotic Locomotion', *International Journal of Robotics Research* **17**(7), pp. 683–701.

Paynter, H. M. (1961), *Analysis and Design of Engineering Systems*, M.I.T. Press.

Penrose, L. S. & Penrose, R. (1958), 'Impossible Objects: A Special Type of Visual Illusion', *British Journal of Psychology* **49**, pp. 31–33.

Pfeiffer, F. & Glocker, C. (1996), *Multibody Dynamics with Unilateral Contacts*, John Wiley & Sons, Inc.

Polderman, J. W. & Willems, J. C. (1998), *Introduction to Mathematical Systems Theory – A Behavioral Approach*, Texts in Applied Mathematics (26), Springer-Verlag.

Pratt, J. E., Krupp, B. T., Morse, C. J. & Collins, S. H. (2004), The RoboKnee: An Exoskeleton for Enhancing Strength and Endurance During Walking, *in* 'Proceedings of the IEEE Conference on Robotics and Automation', pp. 2430–2435.

Pratt, J. & Pratt, G. (1999), Exploiting Natural Dynamics in the Control of a 3D Bipedal Walking Simulation, *in* 'Proceedings of the International Conference on Climbing and Walking Robots'.

Rossmann, W. (2002), *Lie Groups: An Introduction Through Linear Groups*, Oxford University Press.

Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N. & Fujimura, K. (2002), The Intelligent ASIMO: System Overview and Integration, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems', Vol. 3, pp. 2478–2483.

Secchi, C., Stramigioli, S. & Melchiorri, C. (2001), Geometric Grasping and Telemanipulation, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 1763–1768.

Selig, J. M. (2005), *Geometric Fundamentals of Robotics*, second edn, Springer-Verlag.

Shiriaev, A., Perram, J. W. & Canudas de Wit, C. (2005), 'Constructive Tool for Orbital Stabilization of Underactuated Nonlinear Systems: Virtual Constraints Approach', *IEEE Transactions on Automatic Control* **50**(8), pp. 1164–1176.

Snakeboard U.S.A. (2005), 'Snakeboard'.
URL: *http://www.snakeboard.com*

Spong, M. W. & Bullo, F. (2005), 'Controlled Symmetries and Passive Walking', *IEEE Transactions on Automatic Control* **50**(7), pp. 1025–1031.

Stramigioli, S. (2001), *Modeling and IPC Control of Interactive Mechanical Systems – A Coordinate-free Approach*, Springer-Verlag.

Stramigioli, S. & Duindam, V. (2001), Variable Spatial Springs for Robot Control Applications, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems', pp. 1906–1911.

Stramigioli, S. & Duindam, V. (2004), 'Port Based Modeling of Spatial Visco-Elastic Contacts', *European Journal of Control* **10**(5), pp. 505–514.

Stramigioli, S., van der Schaft, A. J., Maschke, B. M. & Melchiorri, C. (2002), 'Geometric Scattering in Robotic Telemanipulation', *IEEE Transactions on Robotics and Automation* **18**(4), pp. 588–596.

Suri, S., Hubbard, P. M. & Hughes, J. F. (1999), 'Analyzing Bounding Boxes for Object Intersection', *ACM Transactions on Graphics* **18**(3), pp. 257–277.

Takegaki, M. & Arimoto, S. (1981), 'A New Feedback Method for Dynamic Control of Manipulators', *ASME Journal of Dynamic Systems, Measurement, and Control* **103**(2), pp. 119–125.

Tanie, K. (2005), 'Letś Work More on Practical Problems!', *IEEE Robotics and Automation Magazine* **12**(2), pp. 3, 6.

The Mathworks (2005), 'Matlab R14'.
URL: *http://www.mathworks.com*

Thoma, J. U. (1975), 'Entropy and Mass Flow for Energy Conversion', *Journal of the Franklin Institute* **299**(2), pp. 89–96.

Trefethen, L. & Bau, D. (1997), *Numerical Linear Algebra*, Soc. for Industrial and Applied Math.

van Amerongen, J. & Breedveld, P. C. (2003), 'Modelling of Physical Systems for the Design and Control of Mechatronic Systems', *Annual Reviews in Control* **27**, pp. 87–117.

van den Bogert, A. J. (2003), 'Exotendons for assistance of human locomotion', *Biomedical Engineering Online* **2**(17).

van der Linde, R. Q. (2000), Bipedal Walking with Active Springs – Gait Synthesis and Prototype Design, PhD thesis, Delft University of Technology.

van der Schaft, A. J. (2000), *L_2-Gain and Passivity Techniques in Nonlinear Control*, Communications and Control Engineering, Springer-Verlag.

van der Schaft, A. J. & Maschke, B. M. (1994), 'On the Hamiltonian Formulation of Nonholonomic Mechanical Systems', *Reports on Mathematical Physics* **34**, pp. 225–233.

van Oort, G. (2005), Strategies for Stabilizing a 3D Dynamically Walking Robot, Master's thesis, University of Twente.

Vela, P. A. (2003), Averaging and Control of Nonlinear Systems, PhD thesis, California Institute of Technology.

Visser, M. & Stramigioli, S. (2006), Generalized Theory and Families of Spatial Springs. Submitted to IEEE Transactions on Robotics.

Visser, M., Stramigioli, S. & Heemskerk, C. (2002), Screw Bondgraph Contact Dynamics, *in* 'Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems'.

Vukobratović, M. (2004), 'Zero-Moment Point — Thirty Five years of Its Life', *International Journal of Humanoid Robotics* **1**(1), pp. 157–173.

Westervelt, E., Grizzle, J. W. & Koditschek, D. E. (2003), 'Hybrid Zero Dynamics of Planar Biped Walkers', *IEEE Transactions on Automatic Control* **48**(1), pp. 42–56.

Whittaker, E. T. (1998), *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, fourth edn, Cambridge University Press.

Wiggins, S. (2003), *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Texts in Applied Mathematics, second edn, Springer-Verlag.

Willems, J. C. (1991), 'Paradigms and Puzzles in the Theory of Dynamical Systems', *IEEE Transactions on Automatic Control* **36**(3), pp. 259–294.

Wisse, M. (2004), 'Essentials of Dynamic Walking — Analysis and design of two-legged robots'.

Wisse, M. & van Frankenhuyzen, J. (2003), Design and construction of Mike; a 2D autonomous biped based on passive dynamic walking, *in* '2nd International Symposium on Adaptive Motion of Animals and Machines'.

Wolfram Research (2005), 'Mathematica 5.1'.
URL: *http://www.wolfram.com*

Yamakita, M. & Asano, F. (2001), 'Extended passive velocity field control with variable velocity fields for a kneed biped', *Advanced Robotics* **15**(2), pp. 139–168.

Yamakita, M., Asano, F. & Furuta, K. (2000), Passive Velocity Field Control of a Biped Walking Robot, *in* 'Proceedings of the IEEE International Conference on Robotics and Automation', Vol. 3, pp. 3057–3062.

Zefran, M. & Kumar, V. (2002), 'Geometric Approach to the Study of the Cartesian Stiffness Matrix', *ASME Journal of Mechanical Design* **124**, pp. 30–38.

Index

Acknowledgments

First and foremost, I would like to thank my supervisor and good friend, *professor* Stefano Stramigioli. Despite (or because of) our differences in personality, approach to research, and culture, the past years have been very fruitful and inspiring for me. Your relentless and unstoppable enthusiasm for everything from girls and geometry to risotto and robotics has often been both highly motivating and a little embarrassing at the same time. Most of what I know about robotics and differential geometry is from our many discussions and from all the books you suggested me to read, and I am very grateful for that. On top of that, it has been great to always be and feel welcome in your home to play with your kids and try to eat spaghetti without a knife.

Even though Stefano was appointed professor just in time to be my first promotor, Job van Amerongen, head of the control lab, is my second promotor and has been truly active in discussing and commenting on the draft versions of this thesis. To me, your input has formed a very helpful complement to the comments of Stefano, by pinpointing the soft spots in my argumentation and formulation, asking the right nasty questions, and of course providing the invaluable help with Photoshop and even LaTeX!

Besides Stefano and Job, I would like to thank the other members of my dissertation committee: prof. C. Melchiorri, prof. A.J. van der Schaft, dr. J.M.A. Scherpen, and prof. P.H. Veltink. I really appreciate the time you have taken to read and comment on my thesis, travel all the way to Twente, and ruthlessly interrogate me at my defense ceremony. I also thank prof. van der Schaft for his detailed comments and suggestions for improvement.

The research in this thesis has been conducted as part of the project GeoPlex, which was sponsored by the European Commission under the Fifth Framework Programme. Being in this project has allowed me to get to know something about the many diverse research fields represented in GeoPlex (from flexible structures through power-electronics to chemical engineering), not to mention the semi-annual meetings for fruitful discussions, fancy dinners, and the occasional skiing trip. Thanks to all fellow Geoplexers for making this possible!

Apart from the cooperation with the European colleagues, I have had the pleasure of collaborating with a number of B.Sc. and M.Sc. students at the University of Twente, who worked on various projects, more or less related to the research in this thesis. I especially would like to thank Eddy, Edwin, Ester, Gijs, Niels, and Yanzhen, for designing, building, and improving our walking robot Dribbel.

Then, a very big 'hartstikke bedankt' to all members of the control engineering lab and of CLP, for the joint lunches, borrels, klootschiet adventures (I think I still owe you a kloot), and general good social working environment. In particular, I would like to thank Carla for taking such good care of all of us; Philippus for always being willing to go ice skating, see a movie, contemplate life, or just drink a beer in that strange communist bar; Mark for the adventurous holidays, freezing tennis lessons, and some specific signs of insanity we shared; Norbert for being my patient office mate and making me realize that Fortran is not as ancient and retarded as modern society wants us to believe; and Bojan for being my other office mate and for proving that the number of icons on a desktop can indeed approach infinity.

I am very happy that, through conferences, workshops, and international exchanges, I have met many interesting people outside the Netherlands, some of whom have become close friends. Debora, bella ragazza, it was a great pleasure to have you as an office mate during your time in Twente, and as a close personal friend ever since. Michela, of all the foreign visitors I have met, you were definitely the one with the most stubborn (and successful!) drive to learn Dutch and ice-skating, although Satoru, Mister Japan, probably beat you at ice-skating on pure willpower. Liudvika, I hope I will be able to enjoy your hospitality and delicious Lithuanian food many more times in the future, wherever you end up after the last of your gazillion trips. Alessandro, Cristian, Nicola, and Pippo, thanks so much for introducing and explaining, in excruciating detail, the finesses of Italian food, people, and general culture. Patric and Dani, it was great fun hanging out with you at all the conferences, and in Croatia, where you proved that Croatians are pretty close competition for Italians when it comes to reckless driving. Vishy and Deepa, will I ever get used to the idea of coming over for a visit without making an appointment, or to using spices when making hutspot with chicken? And finally, a very big thank you to the Paulino family, for making me feel so welcome in their various warm and sunny Florida homes, and for letting me take their money in the endless poker games.

Tot slot wil ik mijn familie bedanken. Mijn ouders en zusje hebben me op vele manieren geholpen om tot dit punt te komen. Van technische hulp bij de wiskunde toen ik nog jong was, tot mentale en financiële steun toen de wiskunde wat moeilijker werd. Ik ben blij dat jullie trots op me zijn, ook al hebben jullie waarschijnlijk geen idee wat ik daar in het oosten uitspook. And Jennifer, what could I say to you that you do not already know? It is amazing that, despite the distance and our busy lives, we have still managed to build a wonderful and close relationship. I am looking forward to the rest of it!

About the Author

Vincent Duindam was born on October 21st, 1977 in Leiderdorp, a small town in the west of the Netherlands. After completing his secondary education cum laude in 1996, he started his bachelor/master studies in Electrical Engineering at the Delft University of Technology in the Netherlands. During an internship in the summer of 2000, he spent three months at the California Institute of Technology, USA, where he worked with prof. Joel Burdick and dr. Kristi Morgansen on the design and control of a robotic fish. In 2001, he obtained his M.Sc. degree cum laude under the supervision of prof. Stefano Stramigioli on the subjects of robotics and geometrical mechanics.

After an adventurous backpacking trip through south-east Asia, he started his Ph.D. position at the University of Twente, Netherlands, in the Control Engineering lab of prof. Job van Amerongen, with Stefano Stramigioli as his research supervisor. His Ph.D. research on walking robots was part of the European-sponsored project *Geometric Network Modeling and Control of Complex Physical Systems* (GeoPlex).

Besides walking robots, his research interests include the general modeling and control of physical systems, geometric mechanics, human-machine interaction, biologically inspired control, augmented reality, and computer graphics.

Research on walking robots has shown that the process of walking, in itself, requires little energy. Indeed, many robots have been built that walk with high efficiency. General analysis and control tools for such efficient walkers, however, are lacking, and many results are based on engineering intuition and ad hoc solutions.

This thesis aims to provide a framework for modeling, analysis, and efficient control of walking robots. The framework uses a port-Hamiltonian system description to express the dynamics of rigid mechanisms and their interaction with the ground. The structure of the resulting models forms the basis for the development of general analysis and control techniques.

The proposed framework extends well-known modeling methods to a broad class of rigid mechanisms with a configuration space described by any combination of Euclidean components (such as linear joints), Lie group/algebra components (such as ball joints), and nonholonomic components (such as nonslipping wheels). Two different 3D contact models are presented: one for compliant contact, and one for rigid contact.

Using the structure of the models, the problem of finding efficient walking gaits is cast as a numerical optimization problem. This setting allows one to optimize not only the joint trajectories but also the mechanical structure of a walking robot. Finally, three control techniques for efficient walking are presented. The first technique uses the computed optimal trajectories to define a power-continuous asymptotic tracking controller. The second technique stabilizes an experimental kneed walking robot by means of a single controller on the hip joint. The third technique uses foot placement to increase the robustness of a three-dimensional walking robot.

ISBN 90-365-2318-4