

TRITA-MEK
ISSN 0348-467X
ISRN KTH/MEK/TR--01/06--SE

PASSIVE CONTROL OF MECHANICAL SYSTEMS

Bipedal Walking and Autobalancing

by

Jesper Adolfsson

May 2001
Technical Reports from
Royal Institute of Technology
Department of Mechanics
S-100 44 Stockholm, Sweden

Akademisk avhandling som med tillstånd av Kungliga Tekniska Högskolan i Stockholm framlägges till offentlig granskning för avläggande av teknologie doktorsexamen fredagen den 11:e maj 2001 kl 11.00 i hörsal M3, Brinellvägen 64, Kungliga Tekniska Högskolan, Stockholm.

©Jesper Adolfsson 2001

Passive Control of Mechanical Systems:

Bipedal Walking and Autobalancing

Jesper Adolfsson 2001

Department of Mechanics, Royal Institute of Technology
S-100 44 Stockholm, Sweden

Abstract

This thesis explores the dynamics of two mechanical systems capable of performing their tasks without the use of active control. The first system is an autobalancer, which can continuously balance rotating machinery. This is accomplished by using compensating masses that move in a viscous media about the axis of rotation. The second system is a passive bipedal walker, capable of human like gait in the absence of external active control. Since there is no control system to suppress errors in the mechanisms it is important to have a detailed knowledge of how parameters influence on these systems.

The dynamics of mechanical models, such as the three dimensional passive walker and autobalancer, are explored and understood using standard dynamical systems methods. This includes calculating the stability of equilibrium positions and limit cycles and creating bifurcation diagrams as parameters of the system are varied. In the case of the autobalancer it is possible to analytically solve for the equilibrium positions. In the passive walker the stability of limit cycles, which corresponds to periodic gaits, are studied. To find these limit cycles a Newton-Raphson root solving scheme is implemented. The model for the autobalancer results in a smooth dynamical system whereas the passive walker includes different kinds of discontinuities. These discontinuities are taken care of by introducing suitable mappings.

Results for the autobalancer include suggestions on how to choose parameters of the systems and how to avoid possible dangerous parameter combinations. It also explains some experimentally observed limit cycles.

In the case of the passive walker, it is shown how the planar passive walker can be extended into a fully three dimensional walker having dynamics in all spatial directions. Many parameter studies are reported which give an insight to the dynamics of the system. The thesis ends with an report on the search for an implementable passive walker.

The methods introduced in this thesis are not limited to the study of autobalancing and passive bipedal walking but are applicable to many mechanical systems with or without discontinuities.

Descriptors: autobalancing, bipedal walking, multibody systems, nonlinear dynamics, stability, bifurcations, limit cycles, variational equations, discontinuities

Preface

This thesis present studies, made on two different passive mechanical systems, the Autobalancer, capable of passively balancing rotating unbalance and the passive walker, capable of human like bipedal gait without using active control. This thesis includes modeling assumptions, simulations, and analysis performed on these systems. It is based on the following papers;

Paper 1 Adolfsson, J.: 1997 ‘A Study of Stability in Autobalancing Systems using Multiple Correction Masses’, Technical Report, KTH, Department of Mechanics

Paper 2 Adolfsson, J.: 2000 ‘A short introduction to walking’, Technical Report, KTH, Department of Mechanics

Paper 3 Adolfsson, J. & Nordmark, A.: 2000 ‘Planar Passive Walkers: Code Implementation Issues’, Technical Report, KTH, Department of Mechanics

Paper 4 Adolfsson, J., Dankowicz H., & Nordmark, A.: 1998 ‘3-D Stable Gait in Passive Bipedal Mechanisms’, In *Proceedings of Euromech 375, Biology and Technology of Walking*, pp 253-259

Paper 5 Adolfsson, J., Dankowicz H., & Nordmark, A.: 2000 ‘3D Passive Walkers: Finding Periodic Gaits in the Presence of Discontinuities’, In *Nonlinear Dynamics*, Volume 24, Number 2, February 2001, pp. 205-229

Paper 6 Dankowicz H., Adolfsson, J., & Nordmark, A.: 1998 ‘Existence of Stable 3D-Gait in Passive Bipedal Mechanisms’, In *Journal of Biomechanical Engineering*, Volume 123, Number 1, February 2001, pp. 40-46

Paper 7 Adolfsson, J.: 2000 ‘3D Passive walkers: Code Implementation Issues’, Technical Report, KTH, Department of Mechanics

Paper 8 Adolfsson, J.: 2000 ‘Finding an Implementable 3D Passive Walker using Continuation Methods’, Technical Report, KTH, Department of Mechanics

The papers are here re-set in the present used thesis format. Some of them are published as indicated above.

Contents

1	Introduction	1
2	AutoBalancing	3
2.1	Background	3
2.2	AutoBalancer	4
2.3	Conclusion	7
3	3D Passive Walking	9
3.1	Background	9
3.2	3D passive walker	11
3.3	Numerical simulation	12
3.4	Finding an implementable 3D passive walker	13
3.5	Conclusion	14
4	Mechanics	15
4.1	Sophia	18
5	Analysis methods	21
5.1	Nonlinear dynamics	21
5.1.1	Flow	21
5.1.2	Limit cycles	22
5.1.3	Stability of the dynamical system	22
5.1.4	Stability of limit cycles	23
5.1.5	Finding limit cycles	24
5.1.6	Variational equations	25
5.1.7	Discontinuities	25
5.2	Bifurcations	26
5.2.1	Saddle-node bifurcations	27
5.2.2	Pitchfork bifurcations	27
5.2.3	Hopf bifurcations	27
5.2.4	Period doubling	28
5.3	Conclusion	28
6	Acknowledgements	31

Chapter 1

Introduction

In our everyday life, we are surrounded by mechanisms, which perform many different tasks. The operation of most of these mechanisms can be understood intuitively by just looking at them. Examples of such mechanisms are wheels, pulleys, chains, link mechanisms, and even devices such as automotive vehicles. At the time of their invention, most of them have been made to work by cleverness and ingenuity, without a need for deeper theoretical understanding. As the theory of Newtonian mechanics was developed by Newton in 1666, it became possible to refine and optimize their operation by making theoretical models of the mechanisms. These theoretical models could now answer questions about the mechanism without constantly having to build new prototype mechanisms. Thus, greatly enhancing the understanding of the mechanism and thereby increasing the speed of development. However, it was not until the possibility of numerical calculations on modern computers that the major revolution came. Today, most mechanisms found around us have gone through many theoretical considerations. For example, it is not uncommon that a mechanism design has gone through one or more of the following steps; CAD system, simulation in a dynamics program, solid-mechanics / thermal / magnetic analysis by a finite element program, and fluid dynamics analysis before emerging as a final product. Today, all these steps are available through commercial or free computer programs running on standard computers. So how does this relate to the two mechanisms studied in this thesis? Here, we will study two mechanical systems, which have greatly benefited from modeling, simulation, and analysis of the dynamics using computers. The two mechanical systems are the autobalancer, capable of continuously balancing rotating unbalances, and the passive bipedal walker, capable of humanlike gait without external control. Both these systems have in common that they are passive by design, thus neither of them requires any active control to function. Since no control systems can suppress errors in the mechanism it is important to have a detailed knowledge of parameters influence on the system. The goal of the two studies is to find parameters that are suitable for implementation. However, in this thesis different approaches have been taken to reach this goal. The autobalancer, which at least dates back to

the 1930's, was an existing mechanical system that needed better understanding as to how parameters influence the system. Thus, the goal was reached by developing a working mechanism using theoretical modeling, simulation, and analysis. In the case of passive bipedal walking, previous walkers had been constrained to walk in a plane so as to prevent lateral dynamics. Here, the goal was reached by first extending previous models into three dimensions having dynamics in all spatial directions using theoretical modeling, simulation, and analysis. When this had been accomplished and implementable parameters had been found, a prototype was to be built to verify the theoretical results. Thus, in the case of the passive walker we did not *a priori* know that it should work, but relied on advanced mathematical methods to find a possible implementation. The three dimensional passive walker can therefore be seen as an experiment in ultimate virtual prototyping where the functionality was developed theoretically. However, only the simulated walker succeeded in walking passively. The implemented walker had some of the dynamics found in simulations but not all. It is believed that given enough time and resources, an actual three dimensional passive walker should succeed in walking.

This thesis can therefore be viewed as the study of two systems where the first is an example of traditional analysis, where the system under investigation already exists and works reasonably well and another where we develop a mechanical system from pure theoretical considerations.

In the following chapters short introductions to the autobalancing system and the three dimensional passive walking system will be given. A discussion is then given of the theoretical background and the mathematical methods used to analyze these two mechanical systems.

Chapter 2

AutoBalancing

2.1 Background

In 1994, Electrolux wanted to reduce vibrations during spin-drying in washing machines, due to the uneven distribution of wash. The company contacted the Department of Mechanics at KTH to get some help with the theoretical modeling of such a system. It was known that a Japanese manufacturer of home appliance had, a decade earlier, filed a patent for using autobalancing in washing machines. However, it was not known if any products had been manufactured using this technology. In 1994, there was no widespread use of autobalancers in washing machines and since this application had already been tried, some concerns were raised that there existed some "hidden" problems with the system. After the initial success to model the system it was decided to continue with the project. It was decided that a more advanced theoretical model should be developed simultaneously with building a prototype washing machine at Electrolux Wascator. Due to design issues, it was also decided that a small experimental autobalancer should be built, see Figure 2.1. This experimental autobalancer was to be used to verify the theoretical model. Autobalancing is not limited to washing machines. Other fields where it has been used or have been tested are;

- Grinding machines, the Atlas Copco Grinder GTG 40 can be equipped with a SKF autobalancing unit. Vibrations levels have been reduced so that the operator now can use the machine without time restrictions.
- Turning lathes
- Fans, dirt attaches to the blades during operation
- Refrigerator compressors, to reduce noise from piston movements
- Computer CD drives, to reduce vibrations from unbalanced CD's operating at much higher rotational speeds than originally intended for CDs

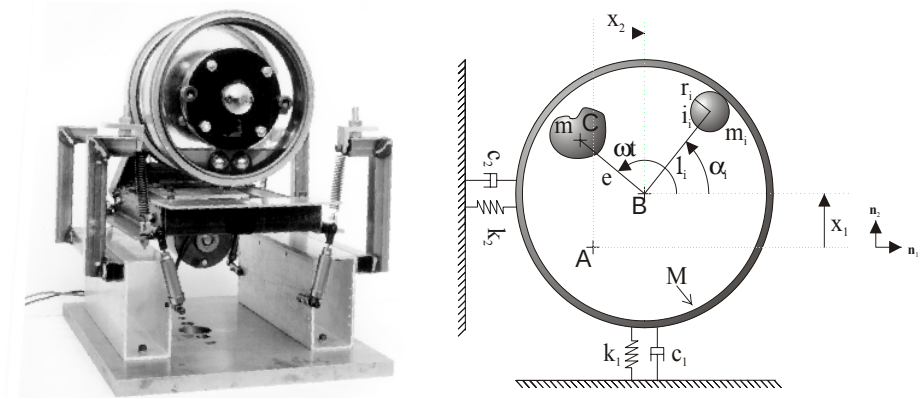


Figure 2.1: The left picture shows the experimental autobalancer equipped with two balance rings. Two balance rings are required to balance dynamic unbalance. Two compensating masses and the viscous media can also be seen. The right illustration shows the model used to analyze the autobalancer. By only using one balance ring placed at an appropriate location on the axis, the experimental autobalancer is well described by the mathematical model found in equations 2.1-2.3.

The history of autobalancing is quite interesting in itself since it is not clear who the original inventor of the mechanism is. The person who is usually believed to be the inventor is E.L. Thearle, who filed a patent in 1934 describing its use to dynamically balance machine tools, see [38]. However, it has been told that this technique was used at the end of the eighteenth century to balance rotating parts in combustion engines. This was supposedly abandoned since it later became possible to manufacture rotating parts with little inherent imbalance.

2.2 AutoBalancer

The cooperation between Electrolux and the Department of Mechanics resulted in a Master Thesis, see Adolffson [1], where it is shown that good agreement between the theoretical model and experimental model is possible. Thus, it was decided to continue with the more specific parameter investigations. A detailed description of the theory behind autobalancing can be found in Adolffson [2] (paper 1). In short, the idea is to have compensating masses, free to move in a circular path about the axis of rotation. At high enough rotational speed, these compensating masses will move to a position so as to reduce the vibration levels

in the system. The equations of motion used in all parameter studies are,

$$\overbrace{M'\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1} = me\omega^2 \cos(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\alpha}_i \sin(\alpha_i) + \dot{\alpha}_i^2 \cos(\alpha_i)), \quad (2.1)$$

$$\overbrace{M'\ddot{x}_2 + c_2\dot{x}_2 + k_2x_2} = me\omega^2 \sin(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\alpha}_i \cos(\alpha_i) - \dot{\alpha}_i^2 \sin(\alpha_i)), \quad (2.2)$$

$$(m_i + \frac{I_i}{r_i^2})l_i\ddot{\alpha}_i + l_i(\delta_i + \frac{\gamma_i}{r_i^2})(\dot{\alpha}_i - \omega) = m_i(\ddot{x}_1 \sin(\alpha_i) - \ddot{x}_2 \cos(\alpha_i)) \quad (2.3)$$

$$i = 1, \dots, n,$$

where the parts corresponding to a simple rotating unbalance are identified, see Figure 2.1 or Table 1 and 2 for a description of the parameters and state variables.

Table 1. Description of the parameters found in the autobalancing system.

Parameter	Description
M'	Total mass of the system
$k_{1,2}$	Spring constant in vertical and horizontal direction
$c_{1,2}$	Viscous damping constant in vertical and horizontal direction
m	Mass of unbalance
e	Distance from axis of rotation to c.m. of unbalance
m_i	Mass of compensating mass
I_i	Moment of inertia of compensating mass
r_i	Radius of the compensating mass
l_i	Distance from axis of rotation to c.m. position of compensating mass
δ_i	Viscous damping constant - translation of compensating mass
γ_i	Viscous damping constant - rotation of compensating mass
ω	Angular velocity of the rotation system

Table 2. Description of the states found in the autobalancing system.

State	Description
$x_{1,2}$	Horizontal and vertical position of the axis of rotation
$\dot{x}_{1,2}$	Horizontal and vertical velocity of the axis of rotation
$\ddot{x}_{1,2}$	Horizontal and vertical acceleration of the axis of rotation
α_i	Angular position of c.m. of compensating mass, measured relative to unbalance
$\dot{\alpha}_i$	Angular velocity of compensating mass
$\ddot{\alpha}_i$	Angular acceleration of compensating mass

Many of the questions answered in Adolfsson [2] (paper 1) are a result from phenomena discovered when working with the experimental autobalancer, and talking with the persons who built the autobalanced washing machine at Electrolux Wascator. For example, the operation of autobalancers requires some viscous media, typically oil, to operate. Due to varying temperature in washing machines, this oil might change its viscosity, potentially resulting in a failure of the autobalancer. Therefore, many of the parameter studies deals with studying the effects of varying the parameters sensitive to temperature changes. Another important questions is what happens when the autobalancer is anisotropically suspended. Typically, a washing machine is suspended in a way that results in different stiffness in the horizontal and vertical directions. For example, it is shown that given enough separation in the horizontal and vertical natural frequency, autobalancing operation is possible in regions between these two frequencies.

Generally, autobalancers require at least two compensating masses to function at varying amounts of unbalance. When using metal balls in a race the configuration where all metal balls are lined up side by side corresponds to the maximum imbalance load that the system is capable of balancing. By filling half the race with metal balls the maximum imbalance load is maximized. Adding an additional metal ball will only compensate for an already present metal ball. Thus, using only two metal balls in a race is usually not volume efficient since there is always the possibility of using many smaller metal balls, which will reduce the volume occupied by the race. However, using more than two compensating masses complicates the parameter studies, since a family of equilibriums will result. Hence, the stability of the family of equilibrium positions have to be scanned for safe operation.

Fitting an autobalancer on a rigidly mounted flexible shaft will result in equal stiffness in all directions. In this isotropic case, it is shown that the equations of motion 2.1-2.3 can be transformed to a time independent form, thus reducing the complexity of stability calculations and reducing the computational cost.

Most studies on the autobalancer were carried out by linearizing the equations of motion about the different equilibrium positions existing in the system. This was done to find out their stability characteristics. For safe and robust operation it is important that the equilibrium, corresponding to a balanced system, is globally attractive. Since the autobalancer is capable of continuously adjusting to different imbalance loads, it is important that all imbalance loads result in stable equilibriums. However, this is not always case, and could lead to a problem if not all imbalance loads are tested in a real system. After it has been established that all imbalance loads result in a stable balanced equilibrium, it is important to investigate that these equilibriums are globally attractive. This was checked in three different systems by performing 30000 simulations, with random initial conditions and random imbalance load. The investigated systems have the same capacity, in terms of imbalance load, but are configured differently. The first system have heavy compensating masses close to the axis of rotation and the last two have lighter compensating masses far from the axis. Histograms are presented showing the number of revolutions before the system

is satisfactorily balanced. The general result indicates that when the desired equilibrium positions are stable, they are also globally attractive. However, the system with the heavy compensating masses close to the axis balance the system faster than the other two. The local stability of the three systems are also compared and it appears like the local stability can be used as reasonable guide to optimize the system.

Other studies on autobalancing systems have been done by for example Hedaya *et al.* [12], which report similar stability results when internal and external damping are varied. These studies have been performed on a system similar to a washing machine. Friction effects and a study of the effects of the eccentricity of the race have been done by Majewski [25] and [26]. In reference [9], Bövik *et al.* shows that it is possible to have autobalancing in non-plane rotors by using multiple scale techniques. It is also interesting to note that some early theoretical studies, using simplified models, had come to certain conclusions regarding stability and rotational speed. These result do not seem to agree with experiments and more advanced models, see Kravchenko [18],[20], and [19].

2.3 Conclusion

There still isn't any widespread use of autobalancing technology, which might be explained by the one big disadvantage of autobalancers. Namely, that they don't work at rotational speeds lower than the lowest natural frequency of the system, see Adolfsson [2] (paper 1). Sometimes, it can even be disadvantageous to have an autobalancing system at low rotational speeds. Many devices have been developed to overcome this problem, such as clamping the compensating masses at low speeds, only releasing them at high speeds. However, introducing these locking devices removes the attractive simplicity of the mechanism and certainly increase its cost. Despite this, SKF, with its long history of making ball bearings, have developed autobalancing into a commercial product named DynaSpin, see [36]. They envisage use in washing machines, grinders, centrifuges, and optical storage devices. It will be interesting to see if they succeed in making autobalancing a successful commercial product.

Chapter 3

3D Passive Walking

3.1 Background

The activity of normal walking is something we perceive as requiring very little conscious effort. While on a higher level it is clear that some sort of control is necessary, such as that required for negotiating rough terrain and recovering from large disturbances, on a lower level it has been shown that the ability to walk is largely a consequence of the inertial and geometrical makeup of bipedal mechanisms, see Basmajian and Tuttle [8] and McGeer [28], [27], [29].

The models studied in this paper are usually referred to as passive walkers, see McGeer [28], since the source of energy is gravity alone and no external active control is applied. Sustained gait is realized by letting the walker descend an inclined plane to counteract the energy dissipated in collisions with the ground and in the knees. To accomplish walking on level ground, actuation would be needed. Different schemes for actuation have been thought of using for example a leaning torso as to provide a torque on the upper legs, see Howell and Baillieul [14] or impulsive pushing of the hind legs, see McGeer [30].

Most of the previous studied passive walkers have been constrained to move in a plane to prevent lateral dynamics. However, to be successful in a life like environment it is evident that a fully three dimensional walker is needed. Many successful attempts have been made to build active walking mechanisms. The most successful up to date is the Honda Biped [13]. This walker is capable of walking in relatively life like environments, such as climbing stairs. The P1 version weighs about 210kg and can operate for about 10 minutes on its own battery pack. The position of limbs are controlled using electrical direct drive servos. Since electrical motors usually have low torque and high speed, some sort of gearing is necessary. This gearing both amplifies inertia and friction, thus making it hard to control the actual force output on the limbs. Instead, it is most common to control the position and angle of different joints. The drawback of this is that it makes it hard to use the natural dynamics available to the mechanism, such as the natural pendulum motion of a swinging leg. In light

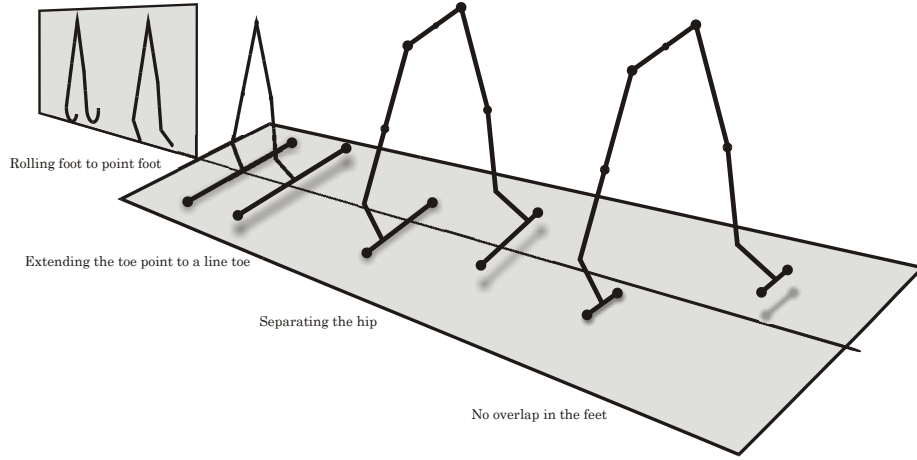


Figure 3.1: The steps taken to extend the planar passive walker into a 3D passive walker.

of this, we decided to try to extend the planar passive walker into a fully three dimensional passive walker, having dynamics in all spatial directions. The idea was to slowly extend previous found planar walkers into a three dimensional configuration, see Figure 3.1. The planar passive walker, originally devised by McGeer [27], had large radius feet and attempts to directly extend this configuration into three dimensions had resulted in unstable dynamics of the system, McGeer [30]. Other attempts had succeeded, albeit outside the passive category by using active control to stabilize the swaying motion, Kuo [21]. The first successful experimental attempt was the passive and knee-less Tinkertoy walker, see Coleman *et al.* [10], which was later modeled and shown stable, Mombaur *et al.* [10] [32]. However, it requires masses to be put on extended booms as to obtain the necessary moments of inertia. The path taken here was to add an extra contact point to the foot. This was made possible after realizing that the foot radius could be shrunk to zero, see Adolfsson and Nordmark [5] (paper 3). By starting with very wide feet, see Figure 3.1, the dynamics would essentially resemble those of the planar walkers. Thus initial conditions and parameters from the planar walker could be utilized. The extension into three dimensions resulted in a lot of new dynamics such as gaits heading obliquely down the plane, see Dankowicz *et al.* [11] (paper 6) and Piironen *et al.* [33] [34]. The initial extension was made using direct numerical simulation. Thus, the convergence to periodic gaits were slow. Therefore, a root finding algorithm was developed to locate periodic gaits, see Adolfsson *et al.* [6] (paper 5) and Adolfsson [3] (paper 7). This also made it possible to locate unstable periodic gaits. The reason for studying unstable gaits is that they can turn into stable gaits as parameters of the system are varied. Potentially, finding new stable gaits, which would have been hard or impossible to locate using direct numerical

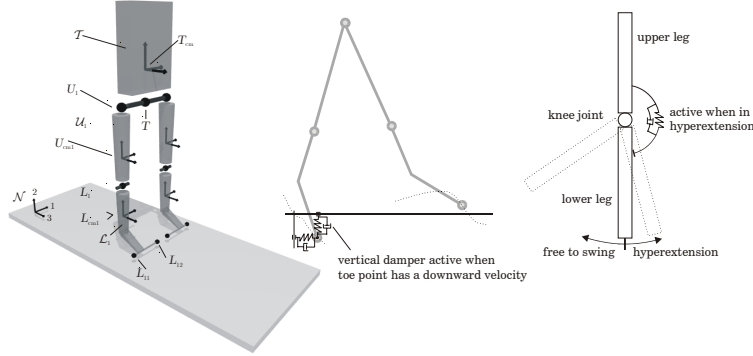


Figure 3.2: The different bodies of the 3D passive walker. Continuous impact dynamics are used in knee and ground plane.

simulation. Initial attempts with the root finding algorithm looked promising and work started on finding a 3D passive walker having no overlap in the feet and a geometry resembling a human, see Adolfsson [4] (paper 8).

3.2 3D passive walker

The mechanical model of the walker consists of five rigid bodies connected by hinge joints, as depicted in Figure 3.2. The bodies are the torso, the upper legs, and the lower legs. The walker makes regular contact with the ground plane through its four toe points. The 3D passive walker is modeled as a continuous system, *where the impact laws used for the planar walker have been replaced by springs and dampers*. These are activated based on the state the walker is in, such as knee-lock and contact with ground, thus avoiding the complexity of treating different impact sequences. A problem that could occur if impact models were used is that the toe points, representing the contact cylinder, could bounce between its ends infinitely many times in a finite time. To handle this would require some extra conditions on the impact models. Thus, the interaction between walker and ground plane is through springs and dampers attached from the initial toe contact point to the toe points. These springs/dampers are active as long as the toe point is below the ground plane, see right pane of Figure 3.2. In human walking, the stance-leg knee is never overstretched. Instead muscles prevent the knee from collapsing, see Inman *et al.* [15]. To prevent a collapse of the knee in our model, the knee has to be in hyperextension where it rests against a torsional spring/damper. For a detailed description on geometry, kinematics, and forces/torques acting on the walker see Adolfsson *et al.* [6] (paper 5) and Adolfsson [3] (paper 7).

Table 3. Continous and discrete state variables used in the 3D passive walker

Continous states	Description
q_1, q_2, q_3	Position of hip point T
q_4, q_5, q_6	Angles describing the orientation of the torso
q_7, q_9	Hip angles
q_8, q_{10}	Knee angles
p_1, p_3, p_5, p_7	Initial toe contact point relative to hip point T
p_2, p_4, p_6, p_8	Initial toe contact point relative to hip point T
u_1, u_2, u_3	Velocity of hip point T
u_4, u_5, u_6	Angular velocities of body T
Discrete states	
s_1, s_2	Knee 1 and 2 (locked/unlocked)
s_3, s_4	Toe 1 and 2 of leg 1 (attached/detached)
s_5, s_6	Toe 2 and 2 of leg 2 (attached/detached)

Table 4. Description of the parametrization of the 3D walker.

Body	Parameter description	nr
Torso	Described by a center of mass position, a mass, and six moments of inertia	10
Upper legs	See above	10
Lower legs	See above	10
Hip joints	Position on hip line	1
Knee joints	Relative to hip joint in upper leg direction	1
Toe points	Each toe point is described by three coordinates, two toe points on each leg	6
Knee dynamics	A angular knee stiffness and damping	2
Hip damping	To resist rotations of the torso	1
Ground plane	Stiffness and damping in three directions	6
Inclination	Inclination of the ground plane	1
Total		48

3.3 Numerical simulation

In Figure 3.3, a stick animation of the 3D walker is shown, sampled at the different events for twelve consecutive gait cycles. This particular walker has an overlap in the feet of 0.1m. The walker is started just after the impact of the right leg's toe points. The events for each toe of the legs are combined into one stick figure, since the time interval between the successive impacts or successive releases is very short. The event percentage is defined relative to the stride

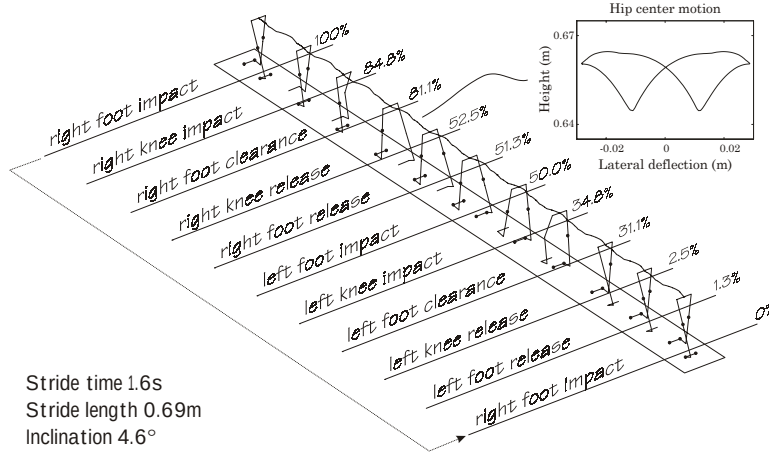


Figure 3.3: Stick animation of an stable periodic gait of the 3D walker. Toe clearance refers to the configuration having a nonzero local minimum in distance between toe points and the ground plane.

time. With methods described in Adolfsen *et al.* [6] (paper 5) it can be shown that this choice of parameters and initial conditions result in an stable periodic gait. The butterfly-shaped orbit of the center of the hip, as displayed in Figure 3.3, shows a similarity with results reported in Rose and Gamble [35].

3.4 Finding an implementable 3D passive walker

Initial attempts to reduce the foot overlap to zero resulted in unstable periodic gaits, see Adolfsen *et al.* [6] (paper 5). Thus, an extensive parameter search was undertaken to find a walker without an overlap. The 3D passive walker is described by 28 state variables and approximately 80 parameters. If the left and right side of the walker are mirrored in the sagittal plane, the number of parameters will be reduced to 48, see Table 4. The sheer number of parameters makes it hard to draw any reasonable conclusions about a particular parameter variation. Typically, the conclusion drawn from a one parameter variation might not hold if one changes the value of another parameter. This emphasizes the importance of having a very clear goal when performing parameter studies, such as finding an implementable 3D passive walker. Below are the requirements set up for an implementable 3D passive walker.

- The mass distribution should be realizable, thus masses should not have to be placed in awkward positions, such as on long extended booms, to get the required moment of inertia.
- Required friction between toe points and ground plane should be reasonable.

- There should be an anthropomorphic (human like) look of the 3D passive walker. This means that geometrical measures of the experimental walker should have similar ratios as a human. For example, the ratio of leg length to hip width should be close to the values found in humans.
- No overlap of the feet.
- Insensitive to parameter changes. This is especially important when the parameters are hard to control or measure in an experimental walker. For example, the actual force between ground and toe points might be very hard to model. Thus, a stable gait should exist for reasonable intervals in the parameters associated with the ground plane.
- Initial conditions that is possible to realize and have a reasonable basin of attraction. The latter might be hard to investigate due to the number of states of the walker.
- Be able to carry some payload, however not necessarily with an upright torso, which most likely would require active control.
- Knee mechanism should be possible to implement passively. The current design requires quite large angular motions of the lower legs during the locked state of the knees.

The search to fulfill these requirements and a presentation of the implemented 3D walker are given in Adolfsson [4] (paper 9).

3.5 Conclusion

It is believed that most of these requirements were fulfilled during the parameter search. However, we were not able to achieve a stable working walker in the laboratory. There are two possible reasons for this, the parameters of the experimental walker didn't match the parameters used in simulation or the walker was not started in the basin of attraction of found stable gaits. It could also be that it was a combination of these two reasons. But due to the lack of adequate measuring equipment, such as 3D motion measuring system, it is hard to know why it failed. For a more thorough discussion on why it failed and possible suggestions to make it work, see Adolfsson [4] (paper 9). Despite this, it is believed that the results presented here still support the assumption that human gait is largely a consequence of geometry and mass characteristics.

Chapter 4

Mechanics

To calculate the motion of a rigid body we use Newton's equation or the linear momentum principle

$$\frac{{}^N d}{dt}(m\mathbf{v}) = \mathbf{F}, \quad (4.1)$$

where N is the inertial reference frame where the time derivative is taken, $\mathbf{v}$ is the body center of mass velocity given in the inertial reference frame N , and $\mathbf{F}$ is the external force acting on the body. Together with the mass center motion, we also need to know how the orientation of the body evolves in time, which is governed by the angular momentum principle

$$\frac{{}^N d}{dt}(\mathbf{L}) = \mathbf{T}, \quad (4.2)$$

where $\mathbf{L}$ is the angular momentum of the body and $\mathbf{T}$ is the torque acting on the body. It is interesting to note that equation 4.1 and 4.2 does not contain any explicit dependence on the actual position or orientation of the body. This is obvious in equation 4.1, and can be seen in the latter equation by carrying out the differentiation of the angular momentum

$$\frac{{}^N d}{dt}(\mathbf{L}) = \frac{{}^N d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{I} \frac{{}^B d}{dt}(\boldsymbol{\omega}),$$

where both the angular velocity vector $\boldsymbol{\omega}$ and the moment of inertia dyad $\mathbf{I}$ are expressed in the reference frame B attached to the body. However, most forces and torques acting on bodies depends on both position and orientation. Some of these forces might be a result of constraints applied to the body. Constraint forces/torques play an important role since they can be viewed as control systems, operating on a much smaller time scale than the rest of the system as to maintain the constraints. Constraints can typically be divided into position/orientation constraints and velocity constraints. All position/orientation constraints (holonomic constraints) can be turned into velocity constraints through differentiation, however not all velocity constraints

can be turned into coordinate constraints. These, non-integrable constraints are called non-holonomic constraints. Examples of holonomic constraints are; hinges, joints, and guides. Examples of non-holonomic constraints are a skate on ice or a sphere rolling on a plane. Constraints on a mechanism results in forces/torques, therefore one usually divide the right hand side of equations 4.1 and 4.2 into

$$\mathbf{F} = \mathbf{F}^C + \mathbf{F}^A \text{ and } \mathbf{T} = \mathbf{T}^C + \mathbf{T}^A \quad (4.3)$$

where the division between constraint and applied forces/torques has been made.

Various methods have been developed to either remove or calculate these constraint forces from the equations of motion, such as for example Lagrange's method and Kane's method which removes the constraint forces/torques and differential algebraic formulations which calculates them. Naturally, all these techniques result in the same dynamics. However, Kane's method has proven to be practical in deriving equations of motions for engineering systems where the number of bodies and constraints are not too high.

The method of finding the equations of motion and stating the constraint forces can be viewed as a two step process. We start by describing the position and orientation of our bodies using generalized coordinates q_i . For free and unconstrained bodies this would require six generalized coordinates for each body. If we have simple constraints, such as hinges, joints and guides and the mechanism does not have any closed loops, it is usually possible to describe the configuration by using a minimum set of generalized coordinates. Since this is not always possible the second step is therefore to state the additional constraints as relations between the generalized coordinates or its time derivative, the generalized velocities.

Thus, by using generalized coordinates q_i we can express the position $\mathbf{x}_j$ and orientation $\mathbf{R}_j$ of our bodies as

$$\begin{aligned} \mathbf{x}_j &= \mathbf{x}_j(\mathbf{q}) \text{ and} \\ \mathbf{R}_j &= \mathbf{R}_j(\mathbf{q}) \end{aligned} \quad (4.4)$$

where the index j runs over the bodies. By differentiation of these relations we obtain a linear relation between our generalized velocities $\dot{q}_i$ and the physical velocities $\mathbf{v}_j$ and $\boldsymbol{\omega}_j$,

$$\begin{bmatrix} \mathbf{v}_1^T \\ \boldsymbol{\omega}_1^T \\ \vdots \end{bmatrix} = A(\mathbf{q})\dot{\mathbf{q}}. \quad (4.5)$$

where the columns of A is a set of basis vectors for the possible motions of the system. D'Alembert's principle of ideal constraints now says that the forces/torques acting on the bodies can be written as

$$[\mathbf{F}_1^b, \mathbf{T}_1^b, \dots] = \begin{pmatrix} \cdots & \mu_1 \mathbf{b}_1 & \cdots \\ \cdots & \mu_2 \mathbf{b}_2 & \cdots \\ & \vdots & \end{pmatrix} = \sum_i \mu_i \mathbf{b}_i = \mu B,$$

where B should satisfy $BA = 0$ and the scalars $\mu_i(t)$ can be determined. The additional constraints we might have can be described as relations between the generalized coordinates of the mechanism. Thus they can be written as

$$f_i(\mathbf{q}) = 0.$$

By differentiating these constraints with respect to time, we get,

$$\sum_k \frac{\partial f_i}{\partial q_k} \dot{q}_k = 0, C_{ik} = \frac{\partial f_i}{\partial q_k} = \begin{pmatrix} \cdots & \mathbf{c}_1 & \cdots \\ \cdots & \mathbf{c}_2 & \cdots \\ & \vdots & \end{pmatrix}.$$

If we have additional velocity constraints, they can also be added to C_{ij} . The null space of matrix C_{ij} now represents a subspace of possible motions of the system. A set of basis vectors of this subspace is usually referred to as the tangent vectors $\boldsymbol{\tau}_i$,

$$C(\mathbf{q})\boldsymbol{\tau}(\mathbf{q}) = 0, \text{ where, } \boldsymbol{\tau} = \begin{pmatrix} \vdots & \vdots & \\ \boldsymbol{\tau}_1, & \boldsymbol{\tau}_2, & \cdots \\ \vdots & \vdots & \end{pmatrix}.$$

The velocity configuration of the mechanism can be written in the tangent base as,

$$\dot{\mathbf{q}} = u_i \boldsymbol{\tau}_i \quad (4.6)$$

where the scalars u_i are the coordinates in this base and are named generalized speeds. Our physical tangent vectors $\boldsymbol{\beta}_i$ can be written as

$$\boldsymbol{\beta}_i = A \boldsymbol{\tau}_i$$

by using equation 4.5 in 4.6. D'Alembert's principle now states that the projection on A of the constraint forces/torques resulting from these constraints can be written as

$$[\mathbf{F}_1^c, \mathbf{T}_1^c, \dots] A = \sum_i \lambda_i \mathbf{c}_i = \lambda C$$

where $\lambda_i(t)$ are scalars. Taking the dot product between our physical tangent vectors $\boldsymbol{\beta}_i$ and the constraint forces/torques will result in

$$\begin{aligned} [\mathbf{F}_1^C, \mathbf{T}_1^C, \dots] \boldsymbol{\beta} &= ([\mathbf{F}_1^b, \mathbf{T}_1^b, \dots] + [\mathbf{F}_1^c, \mathbf{T}_1^c, \dots]) \boldsymbol{\beta} = \\ ([\dots] + [\dots]) A \boldsymbol{\tau} &= (\mu BA + \lambda C) \boldsymbol{\tau} = 0. \end{aligned}$$

Thus, by making the division between applied and constraint forces/torques in the right hand side of Newton's equation and the angular momentum principle we get

$$\frac{N}{dt} (m_j \mathbf{v}_j) = \mathbf{F}_j^A + \mathbf{F}_j^C = \mathbf{F}_j^A + \mathbf{F}_j^c + \mathbf{F}_j^b \text{ and} \quad (4.7)$$

$$\frac{N}{dt} (I_j \boldsymbol{\omega}_j) = \mathbf{T}_j^A + \mathbf{T}_j^C = \mathbf{T}_j^A + \mathbf{T}_j^c + \mathbf{T}_j^b. \quad (4.8)$$

Now, the dot product between the constraint forces/torques and the physical tangent vectors will vanish. Thus, we can take the dot product on the right hand side and left hand side of equation 4.7 and 4.8 to get rid of the constraint forces/torques

$$\left[\left(\frac{N}{dt} (m_1 \mathbf{v}_1) - \mathbf{F}_1^A \right), \left(\frac{N}{dt} (\mathbf{I}_1 \boldsymbol{\omega}_1) - \mathbf{T}_1^A \right), \dots \right] \cdot \boldsymbol{\beta}_i = 0,$$

which result in one equation for each $\boldsymbol{\beta}_i$.

4.1 Sophia

Sophia is a tool for deriving equations of motion and follows Kane's method; see Kane & Levinson [17]. **Sophia**, originally developed by Lesser [24] and [23], runs in computer algebra programs such as Maple and Mathematica. It contains tools for handling reference frames, rotations, frame-based differentiation, computing tangent vectors, and an implementation of Kane's method. An export utility, **exmex** has been developed by Lennartsson [22], which provides functionality for exporting equations of motion for numeric integration in Matlab. It also provides support for forming and exporting variational equations of dynamical systems, which can be used in stability calculations.

The extension to multi-body dynamics is done via a construct called K-vectors, which is a list of ordinary vectors. The velocity configuration K-vector contains the center of mass velocities and angular velocities of each body,

$$\mathbf{v}^< = \begin{bmatrix} \mathbf{v}^{<1} \\ \boldsymbol{\omega}^{<1} \\ \vdots \\ \mathbf{v}^{<K} \\ \boldsymbol{\omega}^{<K} \end{bmatrix}$$

A similar construct is used to hold the corresponding applied forces/torques on each body in the system,

$$\mathbf{R}_a^< = \begin{bmatrix} \mathbf{F}^{<1} \\ \mathbf{T}^{<1} \\ \vdots \\ \mathbf{F}^{<K} \\ \mathbf{T}^{<K} \end{bmatrix}.$$

By expressing the system velocity configuration $\mathbf{v}^<$ using a minimum set of generalized speeds u_i , it is possible to extract the tangent vectors, since $\mathbf{v}^<$ can be written as

$$\mathbf{v}^< = \sum_i u_i \boldsymbol{\beta}_i^< + \boldsymbol{\beta}^<(t).$$

Momentum and angular momentum is obtained by multiplying each row in $\mathbf{v}^<$ by either the mass m_i or the moment of inertia dyad $\mathbf{I}_i$,

$$\mathbf{P}^< = \begin{bmatrix} m_1 \mathbf{v}^{<1} \\ \mathbf{I}_1 \boldsymbol{\omega}^{<1} \\ \vdots \\ m_K \mathbf{v}^{<K} \\ \mathbf{I}_K \boldsymbol{\omega}^{<K} \end{bmatrix},$$

where K is the number of bodies. Using D'Alembert's principle it is now possible to get rid of the constraint forces/torques since $\mathbf{R}_c^< \bullet \boldsymbol{\beta}_i^< = 0$, where the fat dot product is an operation where the ordinary dot product between vectors is performed between each row and the result is summed up. By using these tangent vectors on Newton's equations and the angular momentum principle one gets

$$(\dot{\mathbf{P}}^< - \mathbf{R}_a^<) \bullet \boldsymbol{\beta}_i^< = 0,$$

which results in as many equations as there are $\boldsymbol{\beta}_i^<$'s. These equations, together with the KDE's, usually on the form

$$\dot{q}_i = W_{ij}(\mathbf{q})u_j,$$

can be retained on an implicit form or can put on a first order form,

$$\begin{aligned} \dot{q}_i &= g_i(q, u), \\ \dot{u}_i &= h_i(q, u, t), \end{aligned}$$

since the equations motion are linear in all derivatives. These first order form equations are suitable for numerical integration in standard ODE packages.

Chapter 5

Analysis methods

5.1 Nonlinear dynamics

Generally, it is not possible to analytically solve nonlinear ordinary differential equations (nonlinear ODEs). We therefore have to resort to numerical methods. Today, it exists many robust integrators, capable of handling different types of ODE's. In this thesis, all integration has been performed using Matlab's ODE suite, which range from non-stiff to stiff, low-order to high-order, and variable-order integrators.

A general nonlinear dynamical system is described by a set of differential equations having a nonlinear driving function, this can be written as

$$\dot{x} = f(x). \quad (5.1)$$

If the system doesn't explicitly depend on time it is said to be autonomous, otherwise it is said to be non-autonomous. Points x^* for which $f(x^*) = 0$ are called equilibrium points of the system. If the system is started at an equilibrium point, it will stay there and $x(t) = x^*$ for all times. If all neighboring points are attracted to this point, it is said to be stable. If some neighboring points are repelled, it is said to be unstable, see Figure 5.3 for an illustration of this. Depending on the complexity of the equations of motion, it can sometimes be convenient to put the equations of motion in the form:

$$M(x, t)\dot{x} = g(x, t), \text{ where } f = M^{-1}g. \quad (5.2)$$

This is useful when an explicit inversion of $M(x, t)$ is cumbersome.

5.1.1 Flow

Although it is not possible to generally write a closed form solution to equation 5.1, we can, given the initial condition $x(0) = x_0$, formally define the flow of

solutions $x(t) = \Phi(x_0, t)$ where Φ satisfies

$$\begin{aligned}\frac{\partial \Phi}{\partial t} &= F(\Phi) \text{ and} \\ \Phi(x_0, 0) &= x_0.\end{aligned}$$

5.1.2 Limit cycles

In nonlinear systems, self-sustained oscillations can occur, such as the famous Van der Pol oscillator, see José and Soletan [16]. These self-sustained oscillations are called limit cycles and are isolated and closed trajectories of the dynamical system. Thus, in terms of the flow a limit cycle has to fulfill

$$\Phi(x, t) = \Phi(x, t + T),$$

where $T \neq 0$ is the period time of the limit cycle.

The periodic gait of the passive walker is a limit cycle, since the walker returns to the same configuration, having the same velocities, after one stride. Stable limit cycles attract neighboring trajectories, thus if the stable limit cycle is slightly perturbed, the system will asymptotically return to the original limit cycle. Conversely, unstable limit cycles repel some neighboring trajectories.

5.1.3 Stability of the dynamical system

Stability of the equilibrium point x^* can be calculated by linearizing the non-linear dynamical system found in equation 5.1 about the equilibrium point. For a proof that linearization works for hyperbolic equilibrium points, Strogatz [37] gives a reference to Andronov *et al.* [7]. Introduce a small displacement Δx from x^* and Taylor expand the right hand side of equation 5.1

$$(\dot{x}^* + \Delta \dot{x}) = f(x^*) + \frac{\partial f}{\partial x} \Delta x + O(|\Delta x|^2).$$

However, for an equilibrium point $\dot{x}^* = f(x^*) = 0$ and we are left with

$$\Delta \dot{x} = \frac{\partial f}{\partial x} \Delta x,$$

neglecting higher order terms. The matrix $\partial f / \partial x$ is called the Jacobian of the system and controls how small disturbances about the fix point evolve in time. The general solution to this linear differential equation is (assuming no repeated eigenvalues of the Jacobian)

$$\Delta x(t) = \sum_i \alpha_i \mathbf{v}_i e^{\lambda_i t}$$

where λ_i are the eigenvalues of the Jacobian, $\mathbf{v}_i$ are the corresponding eigenvectors, and α_i are determined by $\Delta x(0)$. Hence, $\Delta x(t)$ will tend to zero if the real part of all λ_i s are less than zero. Thus, the equilibrium point is stable if all eigenvalues have a real part less than zero.

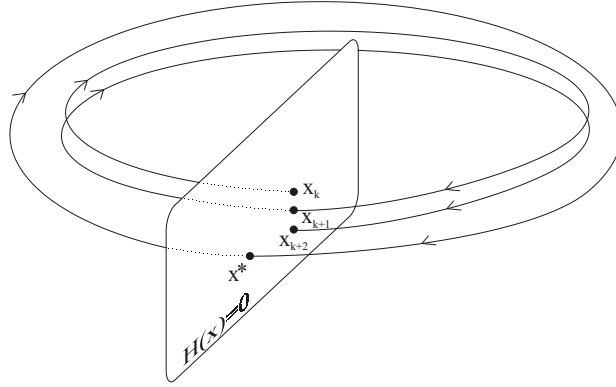


Figure 5.1: Consecutive intersections of the Poincaré section $H(x) = 0$ by a trajectory of the dynamical system. The limit cycle starts and ends in the same point x^* of the Poincaré section.

5.1.4 Stability of limit cycles

This method is fine as long as we have equilibrium points of the dynamical system, but what about limit cycles? To study their stability we introduce a Poincaré surface/section $H(x) = 0$ into the flow that the limit cycle intersects transversely, see Figure 5.1. Thus, we can now look at the consecutive points of intersection of this surface. This is called a Poincaré map,

$$x_{k+1} = f(x_k), \quad (5.3)$$

if all x_k belongs to the surface and the intersection x_k of this surface is followed by the intersection x_{k+1} . If we start with a point on the limit cycle and in the Poincaré section, the same point will be intersected turn after turn, $x^* = f(x^*)$, x^* is said to be a fixed point of the Poincaré map. We are now interested to see how points close to this fixed point map. Introduce a small disturbance d_0 and Taylor expand the right hand side of equation 5.3 about the fixed point,

$$x^* + d_1 = f(x^* + d_0) = f(x^*) + \frac{\partial f}{\partial x} d_0 + O(|d_0|^2).$$

However, $x^* = f(x^*)$, since x^* is a fixed point, and the result is,

$$d_1 = \frac{\partial f}{\partial x} d_0,$$

or more generally,

$$d_{k+1} = \frac{\partial f}{\partial x} d_k.$$

In a similar fashion to the vector field case, $\partial f/\partial x$ is said to be the Jacobian of the linearized Poincaré map for the fix point x^* . Asymptotic decay to x^* of all neighboring points is assured as long as the eigenvalues of the Jacobian are all inside the unit circle. This stability criterion can easily be seen if we assume that Jacobian has no repeated eigenvalues. If so, we can express the initial disturbance d_0 using the eigenvectors of the Jacobian,

$$d_0 = \sum_i \alpha_i \mathbf{v}_i.$$

Inserting this into the linearized Poincaré map we get,

$$d_1 = \frac{\partial f}{\partial x} \sum_i \alpha_i \mathbf{v}_i.$$

Since the $\mathbf{v}_i$'s are the eigenvectors of Jacobian and therefore $\partial f/\partial x \mathbf{v}_i = \lambda_i \mathbf{v}_i$. This results in

$$d_k = \sum_i (\lambda_i)^k \alpha_i \mathbf{v}_i,$$

and as long as all $|\lambda_i| < 1$, d_k will tend to zero as k goes to infinity.

5.1.5 Finding limit cycles

Much of the work relating to finding an implementable 3D passive walker has been focused on finding periodic gaits. Since periodic gaits of the walker correspond to limit cycles of the dynamical system a method of locating limit cycles has been implemented. The problem of finding limit cycles can be stated as that of finding x_0 and $T_0 \neq 0$ satisfying

$$\Phi(x_0, T_0) - x_0 = 0. \quad (5.4)$$

However, the number of unknowns exceeds the number of equations by one (all points on the limit cycle fulfill equation 5.4 above), so a Poincaré section is used to make the solution locally unique,

$$H(x_0) = 0.$$

If we now have a good initial approximation (x_0, T_0) to the solution of the above we can calculate an update $(\Delta x, \Delta T)$ that will take us closer to the solution by using the Newton-Raphson method,

$$\begin{bmatrix} \partial_x \Phi(x_0, T_0) - I & \partial_t \Phi(x_0, T_0) \\ \partial_x H(x_0) & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta T \end{bmatrix} = - \begin{bmatrix} \Phi(x_0, T_0) - x_0 \\ H(x_0) \end{bmatrix} \quad (5.5)$$

where I is the identity matrix, see Adolfsson [6] (paper 6).

5.1.6 Variational equations

The derivative of the flow $\partial_x \Phi(x_0, T_0)$, found in equation 5.5, can be obtained through the standard variational equations, see Adolfsson *et al.* [6] (paper 6). Forming and exporting the variational equations to Matlab code is supported by the **exmex** package, see Lennartsson [22]. This includes support for using implicit formulations, as in equation 5.2, or explicit formulations, as in equation 5.1.

5.1.7 Discontinuities

Different types of discontinuities can be encountered in engineering systems. For example, force discontinuities occur if the Coulomb friction model is used or if a particle goes from one media to another, such as a particle impacting with a water surface. Discontinuities involve the sudden change of state variables and/or their derivatives. This typically occurs in standard impact models, where velocities of the system change discretely according to some impact model. In the 3D passive walker, force discontinuities occur when the knee locks or when the foot impacts with the ground. Sudden change of state variables also occur, since the initial foot contact point is stored as a state of the system. Thus, it is discontinuously updated at each foot impact.

The standard use of variational equations has to be suitably modified at a discontinuity. Although a little bit intricate to derive, see Müller [31] or Adolfsson *et al.* [6] (paper 6), the resulting discrete Jacobian correction has a simple form

$$\Psi = \partial_x G + \frac{(F_a - \partial_x G F_b) \partial_x N}{\partial_x N F_b},$$

where $G_{i \rightarrow j}$ is the discrete change in state variables, F_a and F_b is the vector field of the dynamical system just after and just before the discontinuity, and $\partial_x N$ is the normal of the impact surface, see Figure 5.2 for an illustration. Applying this discrete Jacobian correction to the derivative of the flow just before the discontinuity will yield the derivative of the flow just after the discontinuity,

$$\partial_x \Phi_{\text{after}} = \Psi \partial_x \Phi_{\text{before}}. \quad (5.6)$$

5.2 Bifurcations

Bifurcations occur when the stability type of the system is changed. The stability type of the system might change as parameters are varied. The parameter value where the system changes stability type is called a bifurcation point. Here, we assume that the crossing occurs transversely. Thus, if we are looking at the variation of a parameter μ , we assume that

$$\left. \frac{\partial \operatorname{Re}(\lambda)}{\partial \mu} \right|_{\operatorname{Re}(\lambda)=0} \neq 0$$

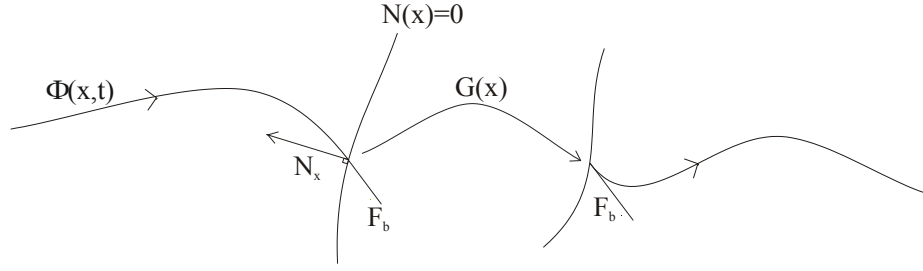


Figure 5.2: A trajectory of the dynamical system intersects a discontinuity surface $N(x) = 0$ where the state variables are updated to a new position in state space by the function $G(x)$. The vector field is given by F_b at the intersection of the discontinuity surface and F_a at the new updated position.

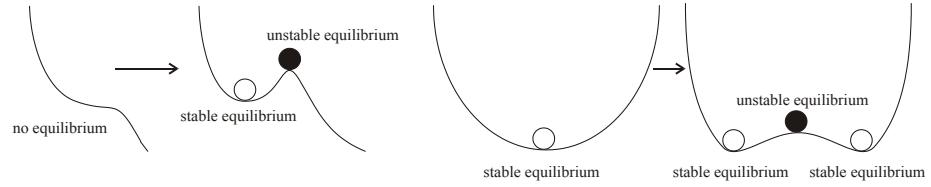


Figure 5.3: A bifurcation occurs when a system changes its stability type. Assume that the spheres move on a surface that is lowered into some viscous media. The equilibrium position of the left bowl will be stable. If the bowl is gradually transformed into the right bowl, the middle position will lose its stability and two new stable equilibrium positions are created. This is called a pitchfork bifurcation. The incline illustrates a saddle node bifurcation. No equilibrium exist in the left incline and as it is gradually changed into the right incline, one stable and one unstable equilibrium is created.

for an equilibrium point in a dynamical system and that

$$\left. \frac{\partial |\lambda|}{\partial \mu} \right|_{|\lambda|=1} \neq 0$$

for a fixed point of the Poincaré map.

Two different types of bifurcations are illustrated in Figure 5.3, where the stability of a small particle on an incline and in a bowl are shown. The two bifurcations that are illustrated are the saddle-node bifurcation and a super-critical pitchfork bifurcation, however there exists other bifurcations. The bifurcations occurring in the two studied systems are described below. Since stability criteria are different for equilibrium points of the dynamical system and fixed points of the Poincaré map, they are listed separately.

5.2.1 Saddle-node bifurcations

Saddle-node bifurcations occur in the passive walker system as the inclination is decreased. At some inclination one stable and one unstable branch are created. This bifurcation also occurs in the autobalancer, for some of the equilibrium positions, as the rotational speed is varied.

Flow: A real valued eigenvalue starts at 0 and either goes to the left half plane (stable branch) or goes to right half plane (unstable branch).

Poincaré map: A real valued eigenvalue starts at 1 and either becomes less than 1 (stable branch) or greater than 1 (unstable branch).

5.2.2 Pitchfork bifurcations

These bifurcations usually occur in systems with symmetry. The 3D walker has a right and left symmetry with respect to the legs. It is the super critical pitchfork bifurcation that usually occurs in the 3D walker. Typically, the gait heading straight down the plane becomes unstable and two oblique gaits emerge. These two oblique gaits are mirrored in each other, see Piiroinen *et al.* [33]. This bifurcation has not been observed in the autobalancer due to the lack of symmetry.

Flow: A real valued eigenvalue crosses the border between the right and left half plane.

Poincaré map: A real valued positive eigenvalue goes through the unit circle.

If the two forked solutions are stable, a super critical pitchfork bifurcation is said to occur, if the two forked solutions are unstable a sub critical pitchfork bifurcation is said to occur. The forked stable solutions always occur on the unstable side and vice versa.

5.2.3 Hopf bifurcations

Hopf bifurcation occurs both in the autobalancer and the 3D walker system. A super critical Hopf bifurcation occurs in the autobalancer if the damping on the compensating masses is decreased or if the suspension damping is increased. The result is small oscillations of the compensating masses.

Flow: A complex conjugate eigenvalue pair crosses the border between the right half plane and left half plane.

Poincaré map: An eigenvalue pair crosses the unit circle.

If there exist a small stable limit cycle on the unstable side, a super-critical Hopf bifurcation is said to occur, the amplitude of this oscillation is proportional to the square root of the distance from the bifurcation point. The sub-critical Hopf bifurcation occurs if there exist a small unstable limit cycle at the stable side. In engineering the sub-critical bifurcation is the most dangerous of the two types, since no stable limit cycle is available after the bifurcation has occurred.

5.2.4 Period doubling

Period doubling sequences occur in both systems analyzed in this thesis. In the autobalancer the previous mentioned limit cycles goes through a series of period doublings, eventually resulting in chaotic motion of the compensating masses. This occurs when the damping on the compensating masses are further decreased or if the suspension damping is further increased. In the 3D passive walker this sometimes occurs when varying the inclination of the plane. It always occurs after a pitchfork bifurcation has occurred, creating oblique gaits.

Just after the period doubling has occurred, the limit cycle will take twice the time to return to the same point in the Poincaré map compared to before the period doubling. In the planar walker, a sequence of period doublings is typically followed by chaotic regions.

Poincaré map: An eigenvalue goes through -1 .

5.3 Conclusion

In this thesis, the dynamics of two different mechanical models are explored using standard dynamical systems methods. The mechanical models are the autobalancer and the three dimensional passive walker. The two models differ in complexity and details but the analysis methods are similar. The model for the autobalancer results in a smooth dynamical system whereas the passive walker includes different kinds of discontinuities. These discontinuities are taken care of by introducing suitable mappings. In the case of the autobalancer it is possible to analytically solve for the time independent equilibrium positions. The equations of motion are then linearized about the equilibrium positions and stability can be assessed. It is not possible to analytically solve for the periodic gaits of the passive three dimensional walker. Hence, a Newton-Raphson root solving scheme for locating periodic gaits was implemented. Periodic gaits correspond to limit cycles of the dynamical system. Since this scheme involves integrating the variational equations of the flow, stability is easily assessed after the root solving scheme has converged to an equilibrium point.

The process of deriving analytical expressions for the equations of motion and the variational equations would not have been practical without symbolical manipulation programs. For example, the compiled C code for calculating the equations of motion of the passive walker are about 90kb and the variational equations is about 300kb.

The methods described in this thesis are not limited to the study of autobalancing and passive bipedal walking but are applicable to many mechanical systems with or without discontinuities.

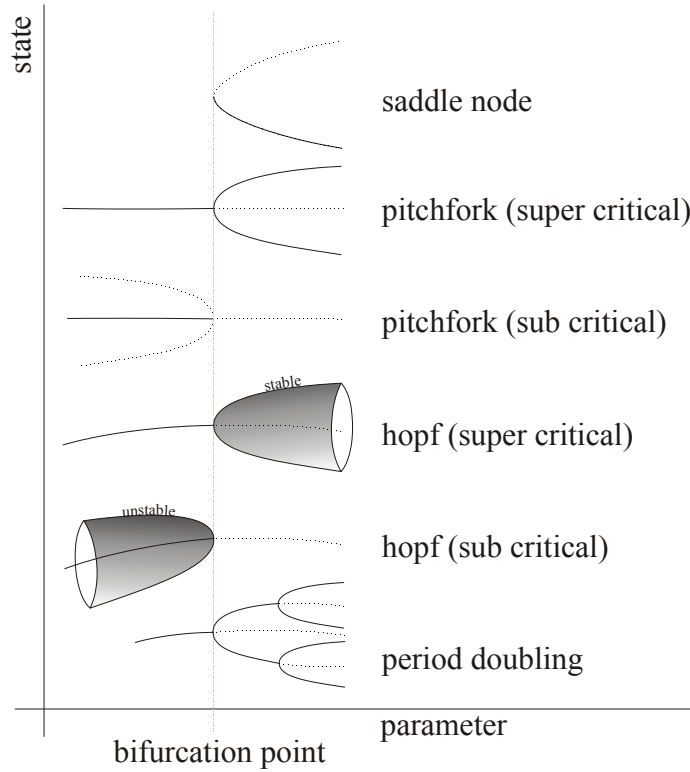


Figure 5.4: Illustration of bifurcations occurring in the autobalancing system and the 3D passive walker system. The saddle node bifurcation is characterized by the sudden birth of two solutions, one stable and one unstable. The pitchfork bifurcation is characterized by the creation of two forked solutions at the bifurcation point. Pitchfork bifurcations come in two flavors, super- and sub-critical. Hopf bifurcations occur when small limit cycles are born at the bifurcation point. This type of bifurcation can also be sub- or super-critical. The final bifurcation discussed here is the period doubling. These might occur when limit cycles change their stability in a certain way.

Chapter 6

Acknowledgements

I would like to express my gratitude to the people at the Department of Mechanics who have made it possible for me write this thesis.

First off all I would like to thank my advisor, Professor Martin Lesser for his support and advice.

I would also like to thank Dr. Arne Nordmark for sharing his knowledge and insights in the field of dynamics and mechanics, Dr. Harry Dankowicz for awakening my interest in passive walking and the many valuable and insightful discussions about interpreting research results, and Dr. Anders Lennartsson is gratefully acknowledged for his expert routines.

It has also been a pleasure to work with Dr. Hanno Essén, Tekn. Lic. Petri Piironen, Dr. Mats Fredriksson, Gitte Ekdahl, my room-mate Per J. Olsson, and the colleagues and friends at the Department of Mechanics.

Finally, I thank the members of my family and Helena, for their support.

Partial financial support from Volvo Research Foundation is gratefully acknowledged.

Bibliography

- [1] Adolfsson, J.: 1995, ‘Self Balancing of Rotating Machines: An Experimental and Theoretical Study’. Master’s thesis, KTH, Department of Mechanics.
- [2] Adolfsson, J.: 1997, ‘A Study of Stability in Autobalancing Systems using Multiple Correction Masses’. Technical report, KTH, Department of Mechanics.
- [3] Adolfsson, J.: 2000a, ‘3D Passive walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [4] Adolfsson, J.: 2000b, ‘Finding an Implementable 3D Passive Walker using Continuation Methods’. Technical report, KTH, Department of Mechanics.
- [5] Adolfsson, J.: 2000c, ‘Planar Passive Walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [6] Adolfsson, J., H. Dankowicz, and A. Nordmark: 2001, ‘3D Passive Walkers: Finding Periodic Gaits in the Presence of Discontinuities’. *Nonlinear Dynamics* **24**(2), 205–229.
- [7] Andronov, A. A., E. A. Leontovich, I. I. Gordon, and A. G. Maier: 1973, *Qualitative Theory of Second-Order Dynamic Systems*. Wiley.
- [8] Basmajian, J. V. and R. Tuttle: 1973, ‘EMG of Locomotion in Gorilla and Man’. *Control of Posture and Locomotion* pp. 599–609.
- [9] Bövik, P. and C. Högfors: 1986, ‘Autobalancing of rotors’. *Journal of Sound and Vibration* **111**(3), 429–440.
- [10] Coleman, M. and A. Ruina: 1998, ‘An Uncontrolled Toy That Can Walk But Cannot Stand Still’. *Physical Review Letters* **80**(16), 3658–3661.
- [11] Dankowicz, H., J. Adolfsson, and A. Nordmark: 2001, ‘Existence of Stable 3D-Gait in Passive Bipedal Mechanisms’. *Journal of Biomechanical Engineering* **123**(1), 40–46.
- [12] Hedaya, M. T. and R. S. Sharp: 1977, ‘An analysis of a new type of automatic balancer’. *Journal Mechanical Engineering Science* **19**(5), 221–226.

- [13] Honda, 'Honda Bipedal Robot'. <http://world.honda.com/robot/>.
- [14] Howell, G. W. and J. Baillieul: 1998, 'Simple Controllable Walking Mechanism which Exhibit Bifurcations'. In: *IEEE Conf. on Decision and Control, December 16-18, Tampa, FL*. pp. 3027–3032.
- [15] Inman, V. T., H. J. Ralston, and F. Todd: 1994, *Human Walking*, Chapt. 1. Williams & Wilkins.
- [16] José, V. J. and E. J. Saletan: 1998, *Classical Dynamics: A Contemporary Approach*. Cambridge University Press.
- [17] Kane and Levinson: 1985, *Dynamics: Theory and Applications*. McGraw-Hill.
- [18] Kravchenko, V. I.: 1983, 'Stability analysis of a row-type counterbalance'. *Mashinovedenie* (1), 25–27.
- [19] Kravchenko, V. I.: 1986a, 'Automatic balancing of rotor of multi-mass system with row-type automatic ball balancer'. *Mashinovedenie* (2), 95–99.
- [20] Kravchenko, V. I.: 1986b, 'Improving the dynamic reliability of loaded machinery elements by self balancing'. *Tyazheloe Mashinostoenie* (5), 19–21.
- [21] Kuo, A. D.: 1999, 'Stabilization of Lateral Motion in Passive Dynamic Walking'. *The International Journal of Robotics Research* **18**(9), 917–930.
- [22] Lennartsson, A.: 1999, 'Efficient Multibody Dynamics'. Ph.D. thesis, KTH, Department of Mechanics.
- [23] Lesser, M.: 1992, 'A geometrical interpretation of Kane's equations'. *Proc. R. Soc. Lond. A*, 69–87.
- [24] Lesser, M.: 1995, *The Analysis of Complex Nonlinear Mechanical Systems*. World Scientific Publishing Co. Pte. Ltd.
- [25] Majewski, T.: 1985, 'Position error occurrence in self balancers used on rigid rotors of rotating machinery'. *Mech. Mach. Theory* **23**(9), 71–78.
- [26] Majewski, T.: 1987, 'Synchronous vibration eliminator for an object having one degree of freedom'. *Journal of Sound and Vibration* **112**(3), 401–413.
- [27] McGeer, T.: 1990a, 'Passive Bipedal Running'. In: *Proceedings of the Royal Society of London: Biological Sciences*, Vol. 240. pp. 107–134.
- [28] McGeer, T.: 1990b, 'Passive Dynamic Walking'. *International Journal of Robotics Research* **9**, 62–82.
- [29] McGeer, T.: 1990c, 'Passive Walking with Knees'. In: *Proceedings of the IEEE Conference on Robotics and Automation*, Vol. 2. pp. 1640–1645.

- [30] McGeer, T.: 1993, 'Dynamics and Control of Bipedal Locomotion'. *Journal of Theoretical Biology* **163**, 277–314.
- [31] Müller, P. C.: 1995, 'Calculations of Lyapunov exponents for dynamic systems with discontinuities'. *Chaos, Solitons and Fractals* **5**, 1671–1681.
- [32] Mombaur, K., M. J. Coleman, M. Garcia, and A. Ruina: 2000, 'Prediction of Stable Walking for a Toy That Cannot Stand'. submitted for publication in PRL.
- [33] Piiroinen, P., H. Dankowicz, and A. Nordmark: 2000, 'Breaking Symmetries and Constraints: Transitions from 2D to 3D in Passive Walkers'. submitted for publication.
- [34] Piiroinen, P., H. Dankowicz, and A. Nordmark: 2001, 'On a Normal-Form Analysis for a Class of Passive Bipedal Walkers'. *International Journal of Bifurcation and Chaos* (11). accepted for publication.
- [35] Rose, J. and J. G. Gamble (eds.): 1994, *Human Walking*. Williams & Wilkins.
- [36] SKF: 2000, 'DynaSpin'. <http://dynaspin.skf.com/>.
- [37] Strogatz, S. H.: 1994, *Nonlinear Dynamics and Chaos*. Addison-Wesley Publishing Company.
- [38] Thearle, E. L.: 1934, 'US Patent: Means for dynamically balancing machine tools'.

Paper 1

P1

A Study of Stability in AutoBalancing Systems using Multiple Correction Masses

Jesper Adolfsson
Royal Institute of Technology
Department of Mechanics
S-100 44 STOCKHOLM
SWEDEN

January, 1997

ABSTRACT. The stability of an autobalancing system is investigated. This work is divided into three parts.

The first part is concerned with the stability in an isotropic two correction mass autobalancer. By isotropic it is meant that the system has the same stiffness and damping in the horizontal and vertical direction. It is shown that the equations of motion can be transformed, in the isotropic case, to a time independent form. This transformation reduces the complexity of the stability computations. The equilibrium positions are calculated. Linear stability analysis is done about these equilibriums. The influence of different parameters are investigated, such as the rotational speed, the suspension stiffness and damping, internal damping acting on the compensating masses and different mass configurations. It is shown that instability can occur when the rotational speed is above the natural frequency of the system. It is also shown that stability can depend on the amount of imbalance load in the system. When the internal damping acting on the correction masses is reduced, a super critical hopf bifurcation occurs.

The second part is concerned with the anisotropic autobalancer, i.e. with different stiffness and damping in horizontal and vertical directions. In this case the equations of motions are time dependant and stability analysis is performed by integrating the variational equations over one half period. It is shown that given enough separation in the horizontal and vertical natural frequency, regions of stability occur when the rotational speed is varied. It is also shown that some of the phenomena occuring in the isotropic autobalancer also occurs in the anisotropic autobalancer.

The third part studies the case when more than two compensating masses are used. This adds some complexity to the stability calculations since using three compensating masses gives a family of equilibrium positions. This means that the stability has to be calculated for all possible equilibrium configurations.

1 Introduction

Autobalancing of rotating machines using moving correction masses is best accomplished where one wants to correct imbalanced rotation varying in time. The type of imbalance can be both static, dynamic or a combination of both. Depending on the design of the autobalancers continuously or discrete balancing can be achieved. The discrete balancing usually involves some sort of locking mechanism of the correction masses. The need of a locking mechanism is due to the fact that continuous balancing can only be achieved when certain system parameters are chosen correctly. The most noticeable system parameter, regarding continuous balancing, is the rotary speed.

The term auto in autobalancing refers to the fact that it is a passive system. By passive it is meant that no active forces are needed to move the correction masses. Therefore no controllers are needed in the system.

The idea behind autobalancers is old and the first patents are from the 1930's. There is not a widespread use of autobalance although there exist many different patents on the subject, mostly regarding the design of locking mechanism of the moving correction masses. Why autobalancers have not been used more widely depends on several things. For example the forces that move the correction masses to their proper location is small compared to the normal forces. The normal forces tend to be very high in practical applications which also require high surface finish and high precision in balance rings and correction masses. However this is not a serious obstacle since ball bearing manufactures have mastered these skills. It therefore seems that the most likely manufacturer of autobalancers will be the ball bearing industry. Another serious drawback is that it does not work at all rotary speeds, and this is here analyzed in greater detail. This means that for some systems, operating at rotary speeds which are not in the autobalancing regime, it is not possible to use this technique. In some system it might still be possible to use it with a locking mechanism of the correction masses. Locking mechanisms have been tested in turning lathes and stationary grinding machines. During normal operation the spindle is rigidly connected to the machine and the correction masses are locked. When imbalance occur one can release the spindle so it is suspended with springs. This allows the machine to enter the autobalancing regime and consequently the correction masses are released. After autobalancing has occurred the correction masses are locked and the spindle is rigidly connected to the machine again. The locking mechanism is usually a mechanical device but other types exist. For example fluids with a melting point so that it is possible to have it solid when the correction masses are locked and melted when they move. The heating needed is usually accomplished by electrical means. Another important aspect of the problem is how system parameters should be chosen to achieve satisfactory autobalancing. It is commonly believed that the only criterion for autobalancing is that the rotary speed is above the natural

frequency of system (see section 3 for an explanation). However this is not true in general and only appears to be one of several requirements on the system, as will be shown later. Theoretical studies in this field have been performed to prove this common misbelief. These theoretical studies, see for example V.I. Kravchenko *et al.*, see [2] [4] [3], have reached this conclusion by simplifying the problem. The nice thing about these studies is that they have obtained analytical results. With some heuristic reasoning one is lead to the same conclusion (see section 3).

Autobalancers have successfully been used in grinding machines, such as the Atlas Copco Grinder GTG 40 equipped with an SKF Auto-Balancing Unit. Imbalance occur in grinding machines because of wear. This installation has reduced vibration levels to less than half of the vibrations occurring without the autobalancing unit. This means that the operator now can use the grinding machine without time restrictions. Other successful attempts have been washing machines where the load causes imbalance during spin drying. This has been tested, with good result, by Wascator in their commercial washing machines. Other areas where it has been tested with various results are refrigerator compressors to reduce noise from piston movements, fans operating in environments where dirt attaches to the blades such that when the dirt comes imbalance occurs.

2 Objectives

This work is mainly focused on how different parameters influence the autobalancing system. In previous work by the author, see [1] it is shown that a relatively simple analytical model accurately models a real autobalancing system. This was accomplished by numerical simulations and measurements on an experimental autobalancer built by the author, see picture 1. Since the model accurately describes the real system, stability analysis are made on the equations of motion. Calculations are performed on a plane isotropic and anisotropic two-ball balancer. In the last chapter it is shown how calculations on an autobalancer equipped with more than two balls can be made. Equations of motions are linearized and eigenvalues and eigenvectors are calculated about the equilibrium positions. Also normal form calculations are performed to analyze hopf bifurcations.

While working on the experimental machine a self sustained oscillation of the ball bearing balls, about their equilibrium position was found. This was not possible to explain with previous stability analysis. However, it will later be shown that these oscillations can be predicted by analytical/numerical means.

3 Illustrative model

There are two common designs of autobalancers. The most common design is letting balls move in a circular ring centered about the axis of rotation. This ring is usually filled with some viscous media such as oil. It will be

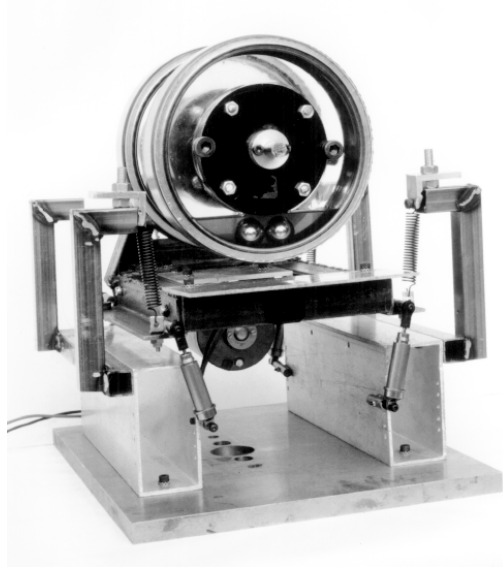


Figure 1. Picture of the experimental machine. The suspension is similar to a washing machine suspension. Two balance rings can be fitted on the axis of rotation. The correction masses are ballbearing balls.

shown later that choosing the right viscosity is of major importance to achieve good performance. This construction is used in the GTG 40 grinding machine mentioned above. The second design is to use pendulum arms mounted on the axis of rotation. This design has been used in turning lathes. There also exist autobalancers where the moving correction masses are small carriages with wheels mounted on them. The reason for using carriages is that it can be designed with a better weight to occupied volume ratio, therefore reducing the volume of the balance rings compared to ball balancers. However, the carriage is a more complex design compared to using balls and is therefore not widely used. However in respect to the models used in this work these designs are mechanically equivalent and can be modelled with the same analytical model.

To get an idea of how autobalancing works consider figure 2. Assume that the system has low damping compared to the critical damping (for an introduction to basic theory of vibration see for example Thomson, Theory of Vibration with Applications [5]). The behavior to exploit is that we will have a π phase shift when the rotary speed is above the natural frequency of the system.

When the driving angular velocity ω of the system is below the natural frequency of the system we will have zero phase shift between the line AB and

BC in figure 2. The natural frequency of the system is here defined as

$$\omega_n = \sqrt{\frac{k}{M + m + \sum_{i=1}^n m_i}}. \quad (1)$$

where M is the system mass and m is the mass of the unbalanced component and m_i is the mass of the i 'th compensating mass. If m and the m_i 's are small compared to M the expression for the natural frequency reduces to the normal expression $\omega_n = \sqrt{k/M}$. A small mass on a ring centered about the axis of rotation B would then move, as indicated in the figure, to the same side as the unbalance. This would increase the amplitude, line AB thus making the vibrations worse. However, when the system rotates at speeds above the natural frequency of the system we will have a π phase shift between the line AB and BC. A small mass, positioned the same way as before, would then tend to the opposite side of the unbalanced component. Assuming that this quasi static argument holds and that there are at least two masses moving symmetrically about the line BC the amplitude AB would decrease. In this quasi static argument it is assumed that the damping is small compared to the critical damping, which implies a phaseshift of π radians when ω goes from just below to just above the natural frequency. It is also assumed that there are no inertial forces due to the compensating masses acting on the system.

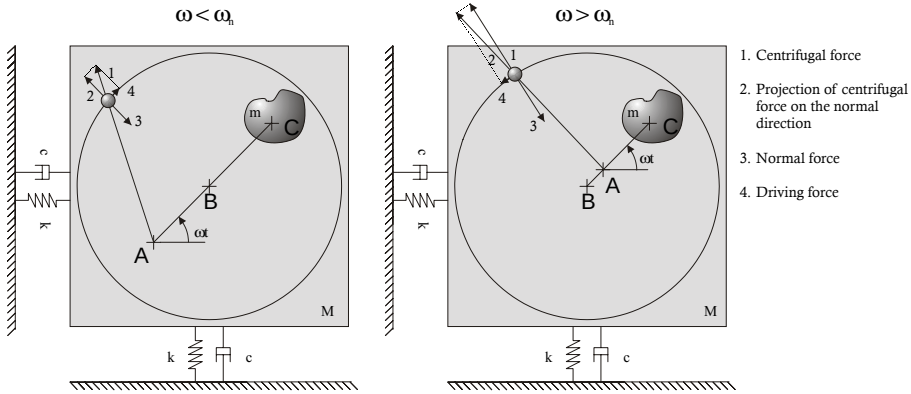


Figure 2. Figure showing the direction the centrifugal force would move a small imaginary mass on a ring when angular velocity is below and above the natural frequency of the system.

If more damping were present this phase shift would occur through a large region and this analysis would fail since it would not be clear what would happen when ω is in the range around ω_n .

4 Analytical model

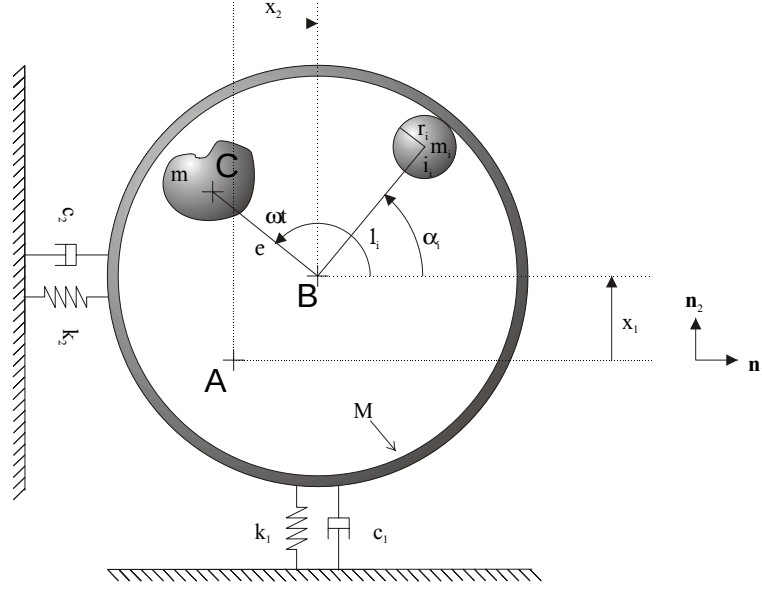


Figure 3. The auto balancing model.

4.1 DERIVATION OF EQUATIONS OF MOTION

To investigate the dynamics of autobalancing an analytical model constrained to move in a plane is used. This might appear to be a serious constraint since in most autobalancers the static unbalance is not situated in the same plane as the autobalancer. However, when using a one plane auto balancer one is forced to place the autobalancer as close as possible to the plane of unbalance. Therefore the plane analytical model will closely resemble an actual system, such as a grinding machine equipped with an autobalancer. If it is not possible to put the auto balancer in the plane of static unbalance or if you have dynamical unbalance one could use two auto balancers, letting the static or dynamic unbalance be situated anywhere between the autobalancers. This setup is for example used in washing machines. Actually it is possible to let the plane of static unbalance be situated outside the auto balancers.

In this model (see figure 3) the mass M is able to move in the $\mathbf{n}_1$ $\mathbf{n}_2$ plane. Linear springs and a linear viscous dampers are attached to mass M . Springs are considered to work in orthogonal directions to each other, the same applies to the dampers. The springs equilibrium position is the point A , therefore the displacement of mass M is measured from A to B with coordinates x_1 along $\mathbf{n}_1$ and x_2 along $\mathbf{n}_2$. The rotating mass is divided into two parts, the mass M with centre of mass at B and the mass m with centre of mass at C situated a distance e from B . The point C is assumed to be rotating about the geometrical centre B with angular velocity ω . The compensating masses, with mass m_i and rotational inertia i_i , are assumed be rotating, without slipping,

on a circular path centered about the geometrical centre B. The distance from B to the centre of mass of the compensating mass is l_i and the radius of the compensating masses are r_i . The angular coordinate is α_i . The forces acting on the compensating masses are assumed to linear in their relative speed compared to the rim. The viscous damping acting on the correction masses will be referred to as internal damping. The viscous damping from dampers will be referred to as external damping. The derivation of the equations of motion was made in Sophia, see [?]. Sophia is a set of routines for doing mechanics in computer algebra packages, such as Maple and Mathematica. Sophia was developed by Professor Martin Lesser at KTH. The equations of motion are

$$M'\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 = m\omega^2 \cos(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\alpha}_i \sin(\alpha_i) + \dot{\alpha}_i^2 \cos(\alpha_i)), \quad (2)$$

$$M'\ddot{x}_2 + c_2\dot{x}_2 + k_2x_2 = m\omega^2 \sin(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\alpha}_i \cos(\alpha_i) - \dot{\alpha}_i^2 \sin(\alpha_i)), \quad (3)$$

$$(m_i + \frac{I_i}{r_i^2})l_i\ddot{\alpha}_i + l_i(\delta_i + \frac{\gamma_i}{r_i^2})(\dot{\alpha}_i - \omega) = m_i(\ddot{x}_1 \sin(\alpha_i) - \ddot{x}_2 \cos(\alpha_i)) \quad (4)$$

$$i = 1, \dots, n,$$

where M' equals $(M + m + \sum_{i=1}^n m_i)$ and is introduced since it occurs frequently. Parameter m is superfluous since the masses M and m are rigidly connected. It is introduced since parameter analysis will be simplified by it.

5 The isotropic case

5.1 DERIVATION OF EQUATIONS

The isotropic case, where k_1 equals k_2 and c_1 equals c_2 is interesting since it is possible to get the equations of motion in a time independent form. An example of a isotropic system could be an autobalancer mounted on a rotating shaft, where the shaft is stiffly mounted. To get the equations in a time independent form is accomplished by expressing the position of point B relative to the point A with coordinates along the axis $\mathbf{a}_1$ and $\mathbf{a}_2$ (see figure 4). It can be expressed as a substitution

$$x_1 = q_1 \cos(\omega t) - q_2 \sin(\omega t) \quad , \quad (5)$$

$$x_2 = q_1 \sin(\omega t) + q_2 \cos(\omega t) \quad , \quad (6)$$

$$\alpha_i = \beta_i + \omega t \quad , \quad i = 1, \dots, n. \quad (7)$$

inserted into equations 2,3,4. Furthermore, since equations 2 and 3 are the projections of the time derivative of the linear momentum along $\mathbf{n}_1$ and $\mathbf{n}_2$ we now need the projections along $\mathbf{a}_1$ and $\mathbf{a}_2$.

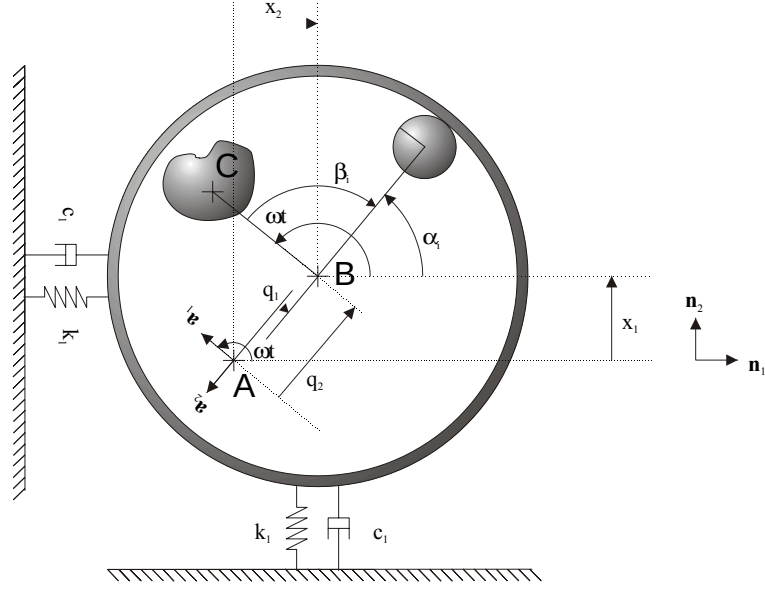


Figure 4. Coordinates in the rotating frame.

This is easily accomplished in Sophia since it has the ability to express vectors in all defined reference frames. The time independent equations can be written as

$$\begin{bmatrix} M' & 0 \\ 0 & M' \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c & -2\omega M' \\ 2\omega M' & -c \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} k - \omega^2 M' & -c\omega \\ c\omega & k - \omega^2 M' \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} me\omega^2 \\ 0 \end{bmatrix} + \sum_{i=1}^n m_i l_i \left(\begin{bmatrix} \ddot{\beta}_i & (\dot{\beta}_i + \omega)^2 \\ (\dot{\beta}_i + \omega)^2 & -\ddot{\beta}_i \end{bmatrix} \begin{bmatrix} \sin(\beta_i) \\ \cos(\beta_i) \end{bmatrix} \right) \quad (8)$$

and

$$\begin{aligned} (m_i + \frac{I_i}{r_i^2}) l_i \ddot{\beta}_i + (\delta_i + \frac{\gamma_i}{r_i^2}) l_i \dot{\beta}_i = \\ m_i ((\ddot{q}_1 - \omega^2 q_1 - 2\omega \dot{q}_2) \sin(\beta_i) + (-\ddot{q}_2 + \omega^2 q_2 - 2\omega \dot{q}_1) \cos(\beta_i)) \end{aligned} \quad (9)$$

$$i = 1, \dots, n.$$

We will first study a system with two correction masses. Using more than two correction masses will result in a system with a family of indifferent equilibrium positions. See chapter 7 for an explanation.

A Study of Stability in AutoBalancing

We will always assume that the correction masses have equal mass and equal distance from the axis of rotation, this will be the case in an autobalancer with two ball bearings in a circular race filled with some viscous media. We will also assume that the parameters, δ_i, γ_i , regarding the viscous damping acting on the correction masses are equal.

5.2 EQUILIBRIUM POSITIONS

To find the equilibrium positions we first rewrite the system of equations in 9 and set all time derivatives to zero, the resulting equations are

$$\begin{aligned} -q_1 \sin(\beta_1) + q_2 \cos(\beta_1) &= 0 \quad , \\ -q_1 \sin(\beta_2) + q_2 \cos(\beta_2) &= 0 \quad . \end{aligned} \quad (10)$$

This can be split into two kinds of solutions. The first is when $q_{1,2}$ equals zero and $\beta_{1,2}$ are given by the system of equations in 8 and can be expressed as

$$\beta_1 = \pi - \cos^{-1}\left(\frac{me}{2m_1 l_1}\right), \quad (11)$$

$$\beta_2 = \pi + \cos^{-1}\left(\frac{me}{2m_1 l_1}\right) \quad . \quad (12)$$

This is the preferred solution where the system is balanced and no vibrations exists. This is actually two solutions since β_1 and β_2 can be interchanged to produce a new solution. These solutions only exist when

$$|me| \leq 2m_1 l_1. \quad (13)$$

The absolute value should be interpreted that e , the distance from the geometrical centre to the mass centre of the unbalanced component, can be both positive and negative. Negative e only means that the unbalance is situated at the opposite side of the defined positive side in figure 3. To obtain the other solutions we rewrite the equations in 10 as

$$\frac{q_2}{q_1} = \tan(\beta_1) \quad , \quad (14)$$

$$\frac{q_2}{q_1} = \tan(\beta_2) \quad . \quad (15)$$

This show that if $q_{1,2}$ is not equal to zero the solution in terms of $\beta_{1,2}$ can be written as

$$\beta_2 = \beta_1 + n\pi, n = 0, 1, \dots \quad (16)$$

For $n = 0$, which means that the two correction masses are in the same place, the solution for $q_{1,2}$ in term of β_1 will be

$$q_1 = \frac{(k - \omega^2 M')(m\omega^2 + 2m_1 l_1 \omega^2 \cos(\beta_1)) + 2c\omega m_1 l_1 \omega^2 \sin(\beta_1)}{(k - \omega^2 M')^2 + (c\omega)^2}, \quad (17)$$

$$q_2 = \frac{2(k - \omega^2 M')m_1 l_1 \omega^2 \sin(\beta_1) - c\omega(m\omega^2 + 2m_1 l_1 \omega^2 \cos(\beta_1))}{(k - \omega^2 M')^2 + (c\omega)^2}. \quad (18)$$

Inserting 17 and 18 into 14 will result, after some manipulation in

$$(k - \omega^2 M')me \sin(\beta_1) + c\omega me \cos(\beta_1) + 2c\omega m_1 l_1 = 0 \quad (19)$$

which has a real solution if

$$(k - \omega^2 M')^2 + (c\omega)^2 \geq (c\omega)^2 \left(\frac{2m_1 l_1}{me} \right)^2. \quad (20)$$

Fixing all parameters except for ω , which we will vary, and letting $me < 2m_1 l_1$ we see that equation 20 will not have any solution for ω about $\sqrt{k/M'}$. The exact bifurcation points can be obtained by solving for the equality in equation 20. Also note that fixing all parameters, except for me , equation 20 won't have any solutions when me is below some value. The solution of 19 can be found in Appendix A. Equation 19 will, in general, have two solutions in β

Now turning back to equation 16 and setting $n = 1$, which physically means that the correction masses are on opposite sides, and following the same route as before results in

$$q_1 = \frac{(k - \omega^2 M') me \omega^2}{(k - \omega^2 M')^2 + (c\omega)^2}, \quad (21)$$

$$q_2 = \frac{-c\omega me \omega^2}{(k - \omega^2 M')^2 + (c\omega)^2}. \quad (22)$$

The solution for β_1 will then be

$$\beta_1 = \tan^{-1} \left(\frac{-c\omega}{k - \omega^2 M'} \right). \quad (23)$$

It is worth noting that equation 21 and 22 will be the equilibrium solution to a simple rotating unbalance isotropically suspended and that equation 23 will be the phase angle as it is usually defined. This is also two solutions since β_1 and β_2 can be interchanged to produce a new solution.

In all the system have six equilibrium positions in this coordinate representation, four of which are physically distinguishable. These four equilibrium positions are schematically drawn in figure 5 below.

5.3 LINEAR STABILITY ANALYSIS

It is now possible to linearize equation 8 and 9 to study the stability of the four equilibrium positions. The equations of motions are put on the standard form

$$M(x)\dot{x} = f(x). \quad (24)$$

where $x = [q_1, q_2, \beta_1, \beta_2, \dot{q}_1, \dot{q}_2, \dot{\beta}_1, \dot{\beta}_2]^T$, $M(x)$ and $f(x)$ are easily derived from 8 and 9. We will now study the eigenvalues and eigenvectors for A that control the stability about the equilibrium positions where

$$A = M^{-1}(x_{0,i}) \frac{\partial f(x_{0,i})}{\partial x} \quad (25)$$

where $x_{0,i}$ are the four equilibrium positions. A will now be a function of the system parameters. It is possible to derive A analytically and in this case it is done with the help of Maple. However, it is not possible to derive the analytical expression for the eigenvalues and eigenvectors of A (eight order system). We therefore insert numerical values for the parameters and equilibrium positions

A Study of Stability in AutoBalancing

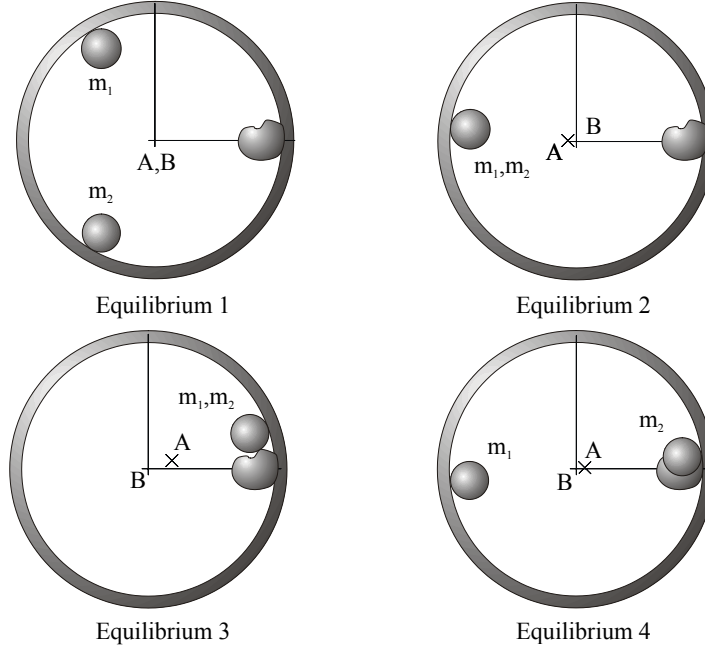


Figure 5. Schematically showing the 4 physical solutions. A is the centre of rotation. B is the geometrical centre. This figure is made for a particular choice of parameters. Note that all positions depend on the amount of unbalance load.

and calculate the eigenvalues and eigenvectors of A numerically. The criterion for stability is that the eigenvalue of A with the maximum real part is negative. This would assure asymptotic stability when we are close to the equilibrium position. We therefore define the stability as

$$\omega^* = \max_{i=1..2n} \text{real } \lambda_i \quad (26)$$

where λ_i are the eigenvalues of A . Thus, if ω^* is negative the equilibrium position is stable and if its positive the equilibrium is unstable. Therefore plotting ω^* as a function of the parameters we are interested in examining will show the stability of selected parameters. Since different λ_i will have maximum real part when parameters are varied, ω^* will in general be a non-smooth function of the varied parameters.

5.4 PARAMETER STUDY

What is usually of interest is to vary ω for a given system to see under what rotational speeds the system behaves as desired. It is also important, as we shall see, to investigate different amounts of unbalance by varying m and e , since we don't want a system where stability depends on the amount of unbalance.

Par	Value	Par	Value
M	1.000	m_1	0.010
k	1.000	I_1/r_i^2	0.004
c	0.700	l_1	1.000
m	0.010	δ_1	0.005
e	1.000	γ_i	0.000

Table 1: Parameters used for stability analysis.

These three, or two since it is the product of m and e that is important, are the most critical when one wants to install a predesigned autobalancing device in a pre-existing system without modifying the suspension of the system. We first pick some parameter values and compare with a previous stability analysis. By setting M equal to one all other masses will be measured in terms of this unit mass. We also set k and l_1 to one. This means that lengths will be measure in terms of l_1 and spring constants in terms of k_1 . For example the conversion factor between natural units of time and common units will be $\sqrt{Mk}^{-1}$.

In table 1 the rest of the parameters are found. These parameters correspond somewhat to the experimental machine used. By assigning a value to all parameters except for one parameter we can plot the eigenvalue with the maximum real part as a function of the varying parameter.

5.4.1 Stability when the rotational speed is varied

In figure 6 all parameters are assigned the values in table 1 and ω is varied between 0 and 6. The same plot is done for each equilibrium position.

The left graph in figure 6 shows, for low values of ω , that we have a stable solution for type 3 equilibrium position and the rest of the equilibrium points are unstable. The third equilibrium represents a solution where the correction masses are at the same position and at the same position as the unbalance. This shows that at *low rotational speeds the correction masses will increase the unbalance* compared to a system without autobalancing.

The cross indicates the natural frequency of the system as expressed in 1. According to previous stability analysis made by V.I. Kravchenko *et al.*, see [2] [4] [3] the region to the left of the cross is stable. In this graph the difference between our stability point, the crossing of zero, and the point obtained in previous work is small. We will however see a greater difference when other parameters are varied. The right picture shows that at higher ω the only stable equilibrium is type 1, which is the desired equilibrium position, for the selected parameters. One of the assumptions in earlier stability analysis is that external damping is small, which corresponds to small c here. Therefore, decreasing c should make the point where stability is attained come closer to the cross. In figure 7 the viscous damping c is reduced to 0.1. The result is the opposite of previous analysis. It is possible, by also decreasing the damping on the correction masses, parameter δ_1 , to move the crossing point even further away

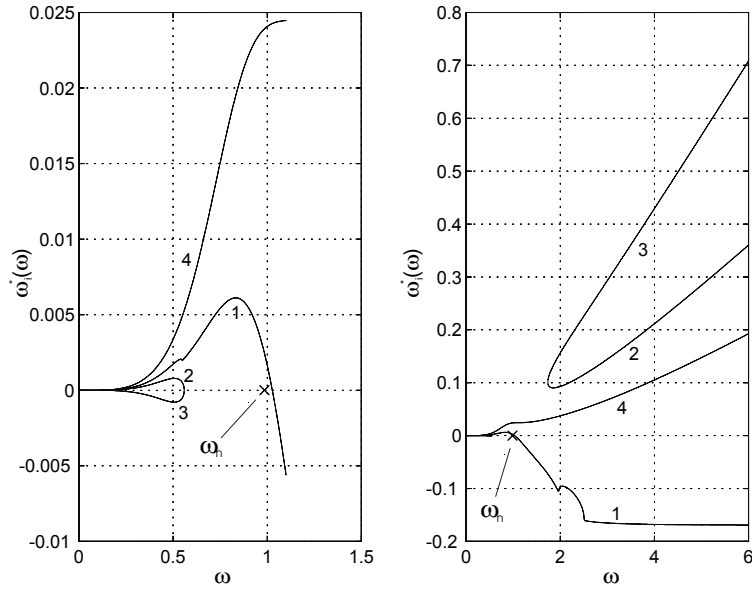


Figure 6. The stability as function of ω , the rotational speed. The numbers correspond to the equilibrium positions. The cross indicates the natural frequency as defined in expression 1.1. Parameters from table 3.1.

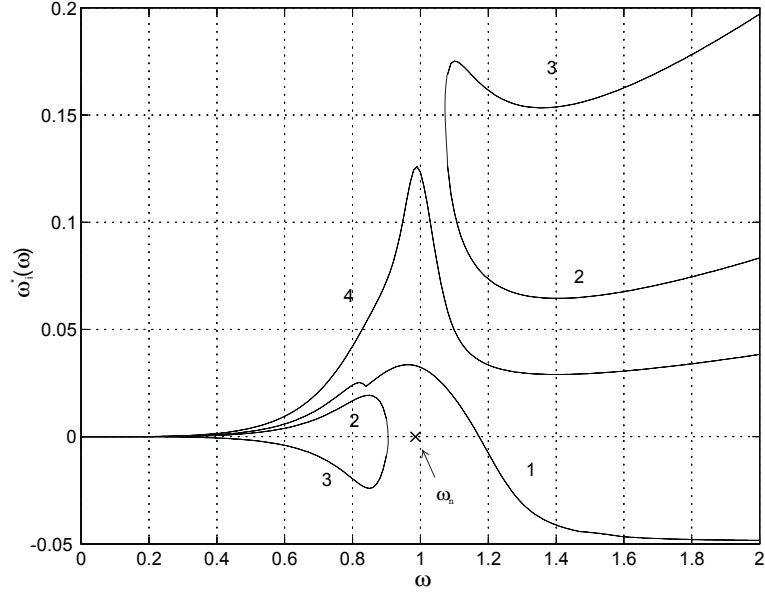


Figure 7. The result of decreasing c from 0.7 to 0.1 with the rest of the parameters from table 3.1.

from the natural frequency of the system. It is apparent that previous stability analysis oversimplified the problem.

5.4.2 Stability when the imbalance load is varied

Assume that we have a machine with parameters according to table 1 and running it at $\omega = 5.0$ as figure 6 tells us that this is a stable choice for ω and for the chosen m . It is now important to check stability when we vary the imbalance load. We vary the load by changing me in the range from zero load to 20 percent above the limit given by inequality 13. It is important to see what happens when the load is above the limit given by 13 since the wanted equilibrium position type 1 does not exist in this region. In figure 8 the result is displayed. At zero load we have an indifferent equilibrium since all positions where the compensating masses are at opposite sides are allowed. Note that equilibrium type 1 and 4 are both indifferent since they represent the same equilibrium position at zero load.

Close to maximum load, me is somewhat less than 0.002, the stability of equilibrium type 1 is lowered, point A in figure 8. At maximum load the stability of equilibrium type 1 and 2 coincide. When the load is above the maximum load capacity only equilibrium type 2 is stable. Only looking at equilibrium type 1 and 2 we have two stable and one unstable equilibrium when me is below maximum load (equilibrium type 1 is actually two equilibrium positions since

A Study of Stability in AutoBalancing

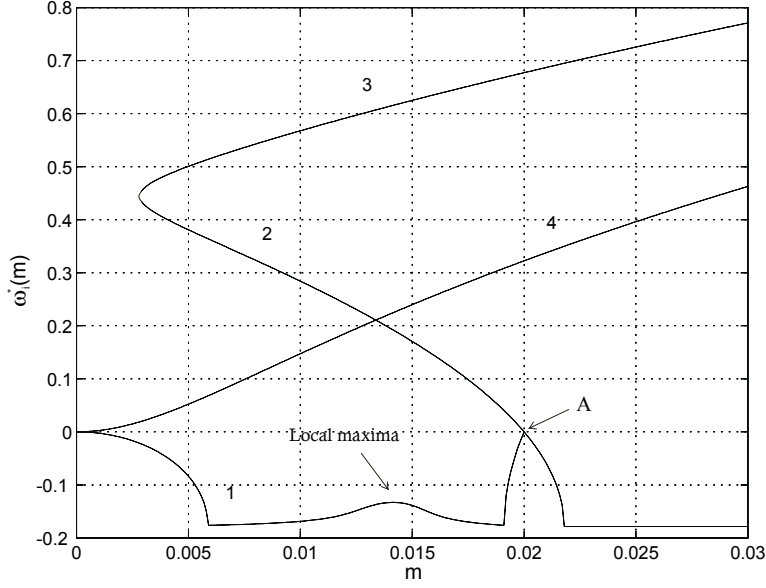


Figure 8. The result of varying the load by changing m . The numbers corresponds to the equilibrium positions. The angular velocity $\omega = 5$. The rest of the parameters from table 3.1.

β_1 and β_2 can be interchanged). When me is above 0.02 only one stable equilibrium exists (equilibrium type 2). We therefore have a supercritical pitchfork bifurcation at m equal to 0.02. *This is positive since it means that overloading the autobalancer, at high rotational speeds, will result in that the two correction masses will be at opposite side of the imbalance, ie. the autobalancer will work as well as possible.*

The local maxima at $me \simeq 0.014$ is interesting since it is possible to let it grow by increasing the external viscous damping and decreasing internal viscous damping. This is accomplished by increasing the external damping from $c = 0.7$ to $c = 2.7$ and decreasing the internal damping from $\delta_1 = 0.005$ to $\delta_1 = 0.002$. In figure 9 the local maxima has grown becoming unstable, since ω^* is greater than zero. We then have a system where the stability depends on the amount of load. This could potentially be a problem when using an autobalancer if is not tested with all admissible loads. In figure 9 the local maxima has grown and become unstable. Generally it seems like *increased external viscous damping requires increased internal damping*. We can test this hypothesis by doing a contour plot of ω^* as function of external damping c and internal damping δ_1 . In figure 10 contour plots with different loads are found, the zero contour line divides the stable region from the unstable. As seen in figure 10b-d the zero contour line is increasing in internal damping with increasing external damping. It is worth noting that *optimal damping, in terms of linear stability, depends on*

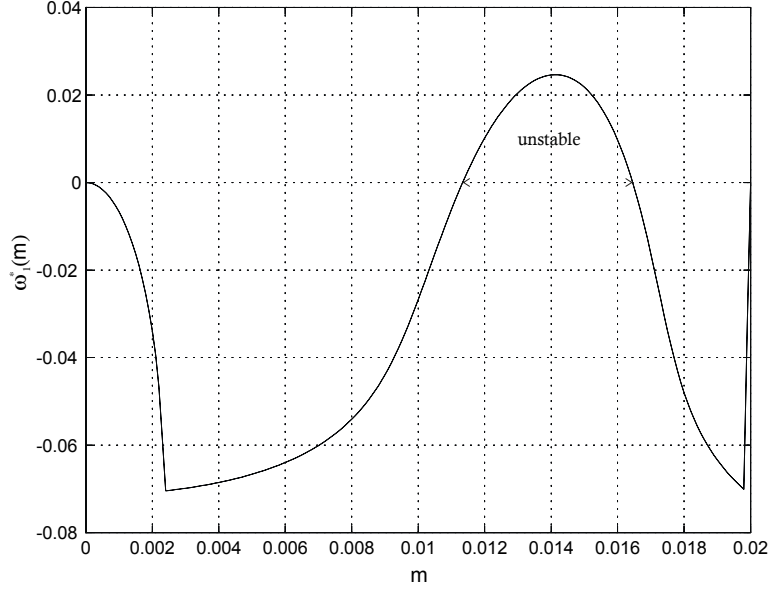


Figure 9. By increasing external damping and decreasing internal damping the system will be unstable for certain loads.

the amount of load. Therefore, choosing the right amount of internal damping and external damping has to be a compromise so that sufficient stability is guaranteed at all amounts of load taken from a figure such as 10. In figure 8 we see that the local maxima has a maximum at $m \simeq 0.014$. This correspond to an equilibrium position $\beta_1 \simeq \frac{3\pi}{4}$ and $\beta_2 \simeq \frac{5\pi}{4}$. From all contour plots in figure 11 we see that the zero contour line has a maximum in δ_1 at $m \simeq 0.014$. Therefore it seems natural to test a two correction mass autobalancer with a load so that the equilibrium positions of the two correction masses will be $\frac{3\pi}{4}$ and $\frac{5\pi}{4}$. This test would then assure stability for all other loads.

5.4.3 Varying the mass of the compensating masses

The maximum amount of imbalance the autobalancer can compensate for, with zero resulting amplitude, is equal to $2m_1l_1$ assuming two compensating masses of equal mass and distance from the geometric centre. Assuming that we have limited space available where the autobalancer is mounted we could increase m_1 and decrease l_1 but still keeping two times their product to a constant value given by the maximum imbalance. The dynamics of the system will then change, which can be seen by looking at equations 8 and 9. In equation 8 nothing will be changed since the only place where m_i and l_i occur is as a product and that is kept constant by our assumption above. However, in equation 9 we will have a term like $\delta_i l_i$ multiplied by β_i and the right hand will be multiplied by m_i . Keeping some sort of balance between the viscous forces

A Study of Stability in AutoBalancing

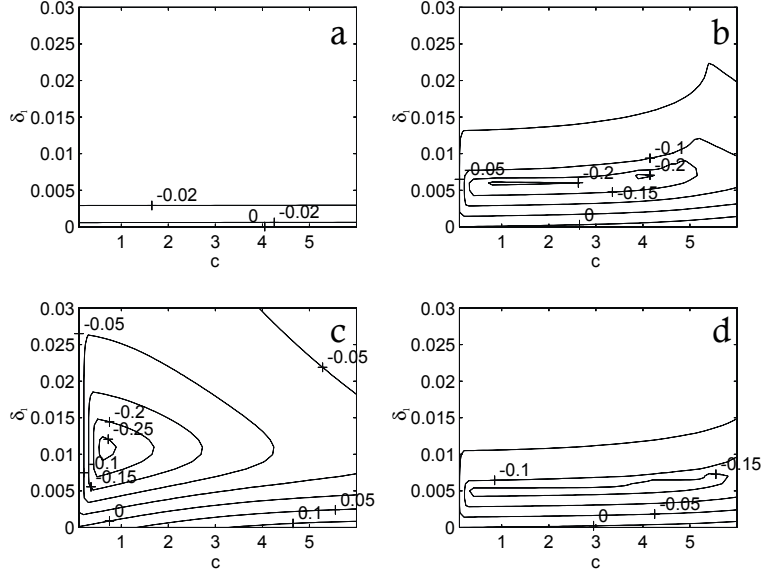


Figure 10. Contour plots of the maximum real part of eigenvalues as function of external damping, parameter c , and internal damping, parameter δ_1 . Parameters from table 3.1 and $\omega = 5$. The load in a $m = 0.002$, in b $m = 0.008$, in c $m = 0.014$ and in d $m = 0.019$.

and the right hand side forces we expect that increasing m_i and decreasing l_i will have to be compensated for by increasing δ_i . That this is the case can be seen in figure 12 where the border line between stable and unstable regions have been plotted for various m_1 and l_1 . It also seems like the border when c is increased tends to some limiting value in δ_1 . This would imply that if δ_1 is chosen above this limiting value stability will not depend on c . One could argue whether this low internal damping actually will occur in a real system. There are a lot of factors which actually could produce a system close to this boundary. For example in a washing machine with the drum mounted horizontally it is not desirable to activate the compensating masses at revolutions lower than the stable limit given by figure 6. To activate the compensating masses the viscous force must overcome the force from gravity so that the balls start moving with the outer rim. This might impose a constraint on the viscosity of the oil, lower viscosity will activate the compensating masses at higher revolutions. Another important factor is temperature. In a system which is operating under different temperature conditions the viscosity of oil might change dramatically, maybe so much that the stability boundary is crossed.

It is also interesting to see how the stability changes when this product, $2m_1l_1$, is held constant and the mass and length is varied. In figure 13 two cases are compared. Line one correspond to a system where the compensating

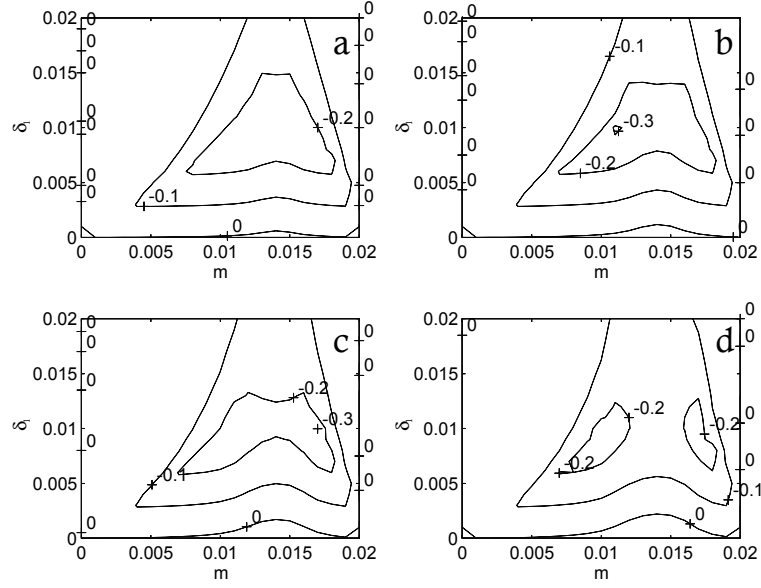


Figure 11. Contour plots of the maximum real part of eigenvalues as function of external load, parameter m , and internal damping, parameter δ_1 . Parameters from table 3.1 and $\omega = 5$. The external damping, parameter c equals , in a 0.5, in b 1.0, in c 1.5, in d 2.0.

mass is equal to 0.01. Line two correspond to a mass equal to 0.05. In both systems the product $2m_1l_1$ is held constant and is equal to 0.01. The parameter controlling the viscous damping, δ_1 , is increased in the second system following the reasoning above. As seen in the figure 13 the increased mass system shows a superior stability over the lower mass system. Also note that the first curve segment in the system with a heavier compensating mass decreases earlier and more rapidly than the low mass system. One would then expect that the system with a heavier compensating mass would perform better compared to a system with a lighter compensating mass. In section 5.6 it is shown that this the case.

5.5 HOPF BIFURCATIONS AND CHAOS

In figure 12 the border line between stable and unstable regions are drawn for some different choices of system parameters. As argued before there might be situations when the internal damping is low, even so low that the border line might be crossed. It is therefore important to know what will happen when crossing the stability boundary.

A bifurcation will occur when a equilibrium point changes stability. For an excellent introduction to bifurcation theory nonlinear dynamics see [?].

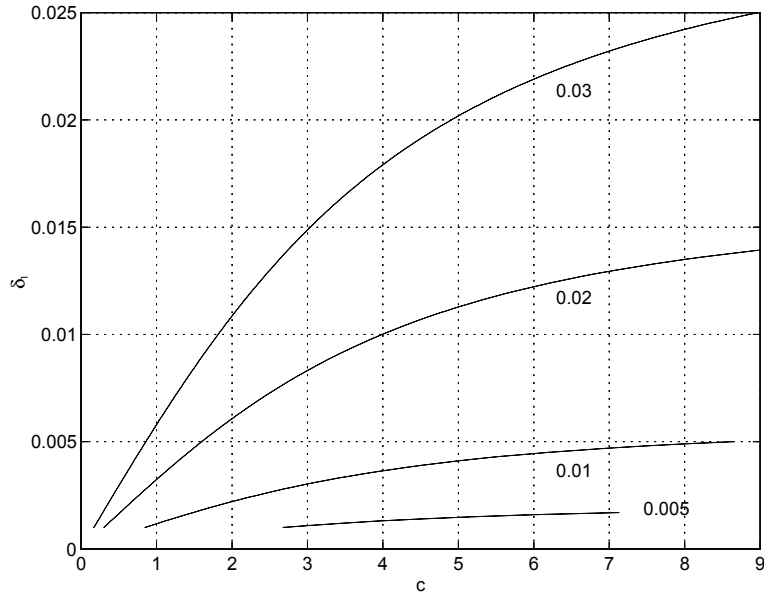


Figure 12. Border line dividing stable from unstable region when the mass and distance from geometrical centre to compensating masses are varied but their product is kept constant and equal to 0.01. Numbers indicating mass m_1 . Regions to the north-west of border line is stable. Parameters taken from table 3.1 except for $m_1 = 0.014$

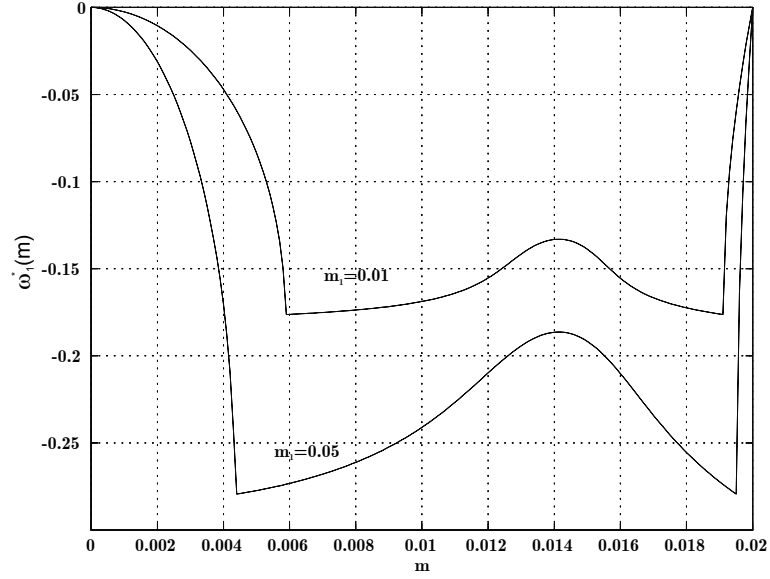


Figure 13. Comparison of stability when the compensating mass is increased. Line 1 $m_1 = 0.01, l_1 = 1$ and $\delta_1 = 0.005$. Line 2 $m_1 = 0.05, l_1 = 0.2$ and $\delta_1 = 0.04$. Rest of the parameters from table 3.1.

If two complex conjugate eigenvalues crosses from the left half plane into the right half plane a Hopf bifurcation typically occurs. Since the real part is positive in the right half plane the equilibrium point loses its stability. When crossing the imaginary axis in this way two things can happen. We might see oscillations in the system approaching some limiting amplitude resulting in a limit cycle. This is called a *supercritical* Hopf bifurcation. When the real part is small and positive the limit cycle will be sinusoidal with an angular frequency equal to the imaginary part of the crossing eigenvalues. We might also find sinusoidal oscillations that continues to grow exponentially away from the equilibrium point. In this case local analysis can not tell what will happen. This is called a subcritical Hopf bifurcation. In engineering this is potentially a much more dangerous bifurcation, since the supercritical bifurcation will have an amplitude proportional to $\sqrt{\mu - \mu_c}$, where μ is the bifurcation parameter we are interested in varying and $\mu = \mu_c$ when crossing the imaginary axis. This applies for small μ 's. Since our system have four degrees of freedom, which results in eight state variables, there is somewhat more to it. The dynamics of the system will approach a 2-dimensional surface embedded in the eight dimensional state space of our system. This surface is called the centre manifold. The trajectories approach this centre manifold because the other eigenvalues still have a negative real part. The shape of the surface and the dynamics on this surface depends on the parameters of the system.

If the stability boundary is crossed and a subcritical Hopf bifurcation occur, the system might experience violent vibrations due to the compensating masses movement. However, if there instead will be a supercritical Hopf bifurcation, the result will be a stable limit cycle. This stable limit cycle will manifest itself as small oscillation of the balls about their equilibrium position with resulting small vibrations of the suspended mass.

To examine this a numerical experiment can be made in the stable and the unstable regions. From figure 12 we have that at $\delta_1 = 0.0065$ the system will be in the stable region and that $\delta_1 = 0.0045$ it will be in the unstable region for $m = 0.02$. In figure 14 the phase portrait of $\beta_{1,2}$ is plotted for various values of δ_1 . All trajectories are plotted after the initial transients has dissappeared. Since $\beta_{1,2}$ can be interchanged to produce a new solution, both solutions are plotted. At $\delta_1 = 0.0065$ the system remains at the stable fixpoint. At $\delta_1 = 0.0045$ a 1 periodic limit cycle exists. *This limit cycle can be observed in the experimental machine* when external damping is high or internal damping is low. There is one noticable feature of this limit cycle, one of the compensating masses has a larger amplitude than the other. This is also observed in the experimental machine. The time of the oscillations also agrees with the measured. Continuing to lower the internal damping results in period doubling. At $\delta_1 = 0.0032$ we have a 2 periodic limit cycle and at $\delta_1 = 0.00305$ we have a 4 periodic limit cycle. This results at even lower internal damping in a chaotic motion of the compensating masses. When $\delta_1 = 0.00302$ we have a chaotic motion of the system. At $\delta_1 = 0.003$ something interesting happens, the two correction masses start changing positions with each other. This would of course be hard to realize since that would mean that the balls would have to move through each other. That some sort of Hopf bifurcation occur can be seen in figure 15 where one pair of eigenvalues are crossing the real axis when the internal damping δ_1 is varied from 0.01 to 0.003.

Using centre manifold theory and normal form analysis it is possible to calculate whether a stable or unstable Hopf bifurcation will occur. This was done for the border in figure 12 where $m = 0.02$. These calculations show that a stable Hopf bifurcation will occur for this particular border. This in combination with the experimental machine would indicate that a *stable Hopf bifurcation will occur when the internal damping acting on the compensating masses are lowered*.

5.6 NUMERICAL EXPERIMENTS

We have employed a local stability analysis, i.e. we examined the stability of the linearized equations of motion about the equilibrium position. It is therefore interesting to examine whether local stability will influence the complete motion. To see this we could devise some rules which determines a good performance from a poor one and run a series of time simulations. These simulations can then be compared to the local stability analysis. As usual the problem is to decide what separates a good performance from a poor one. To the authors knowledge there does not exist any accepted performance test to evaluate

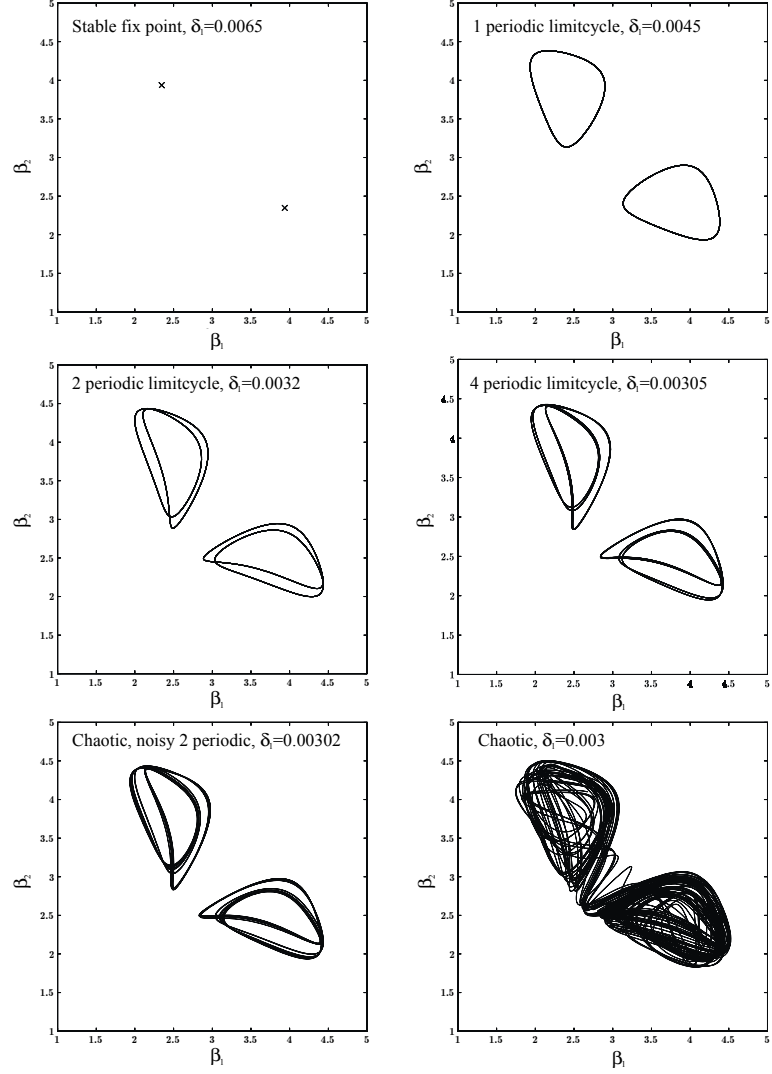


Figure 14. Phase portrait of $\beta_{1,2}$ when varying the internal damping δ_1 . Period doubling occurs and at low internal damping chaotic motion is found.

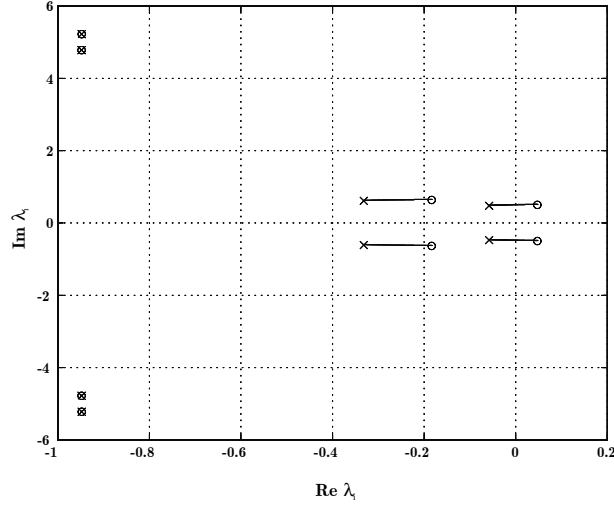


Figure 15. Showing the crossing of eigenvalues of the real axis when δ_1 is varied. Cross indicates $\delta_1 = 0.01$ and ring $\delta_1 = 0.003$.

autobalancers. The following is proposed as a candidate for such criteria. In most cases an autobalancer is fitted where one can not foresee the amount of unbalance and its position. Therefore the most common situation will be that the autobalancer is compensating for an unbalance and suddenly this amount of unbalance and its position change. A corresponding time simulation will then have initial conditions; all speeds are set to zero and the position of the axis of rotation is at the geometrical centre. The angular position of the correction masses are chosen randomly which corresponds to a system balancing some unbalance at some position. The new unbalance situated at zero angular position is randomly chosen and within the maximum balancing capability. We could then measure the time or number of revolutions until the amplitude of the oscillations are below some specified value. This value could be a fraction of the maximum amplitude in the simulation or some value determined by what is acceptable as amplitude. To test this evaluation method 10000 numerical simulations were made on some different choices of parameters. The measurement used were the number of revolutions before the amplitude becomes permanently smaller than one fifth of $2m_1l_1/M$. The amplitude $2m_1l_1/M$ correspond to the amplitude one would have if the unbalance me equals $2m_1l_1$ and the rotary speed ω is much greater than the natural frequency of the system. In figure 16 the stability of three different parameter choices is plotted, when the load is varied. The parameters used can be found in table 2. In the left table the parameters that are common to all three system are found. In figure 17 the

Par	Value(Common)	Par	1	2	3
M	1.000	c	1.8	0.7	0.7
k	1.000	δ_1	0.003	0.005	0.003
m	0 - 0.02	m_1	0.01	0.01	0,05
e	1.000	l_1	1.0	1.0	0.2
γ_i	0.000	I_1/r_i^2	0.004	0.004	0.040

Table 2: Parameters used for numerical comparison.

outcome is displayed as a histogram over the number of revolutions. The mean value and standard deviation is also displayed. The first system has relatively high external damping c_1 and low internal damping δ_1 compared to system two. System one has somewhat better stability at low loads but worse at the middle region compared to system two. System one and two have identical compensating masses. System two clearly behaves better than system one, both the mean value and the standard deviation is lower than in system one. Since figure 16 only displays the stability of the linearized system there is no reason to expect that the full nonlinear system will behave the same way. We are not even guaranteed that the system will tend to the equilibrium position. However these three histogram display the result of 30000 simulations and not one has not reached the equilibrium position. The idea is that when the system is reaching its equilibrium position it has to spend some time in the region where the linear model accurately models the behavior and therefore it should have some correspondence with the outcome in the histograms. In system three the mass of the compensating mass is increased five times over system one and two. The length l_1 is also reduced to one fifth of system one and two. Following the reasoning in section 5.4.3 the internal damping δ_1 is increased to compensate for the inertial forces due to a heavier mass. As seen in figure 16 this system show a higher stability than both system one and two. The outcome of the numerical simulations is displayed in histogram three of figure 17. Parameter set three shows both lower mean value and lower standard deviation than both system one and two. It therefore seems that one should *choose heavier compensating masses close to the axis of rotation over lighter masses far from the axis of rotation*. It also seems that the local stability tells something about the dynamics of the global system and can be used as a reasonable guide to optimize a system.

6 Anisotropic case

6.1 DERIVATION OF EQUATIONS

Most constructions where an autobalancer might fit in is clearly anisotropic. For example a common use of autobalancers are in washing machines. When spin drying the wash is not evenly distributed and as result of this an unbalance

A Study of Stability in AutoBalancing

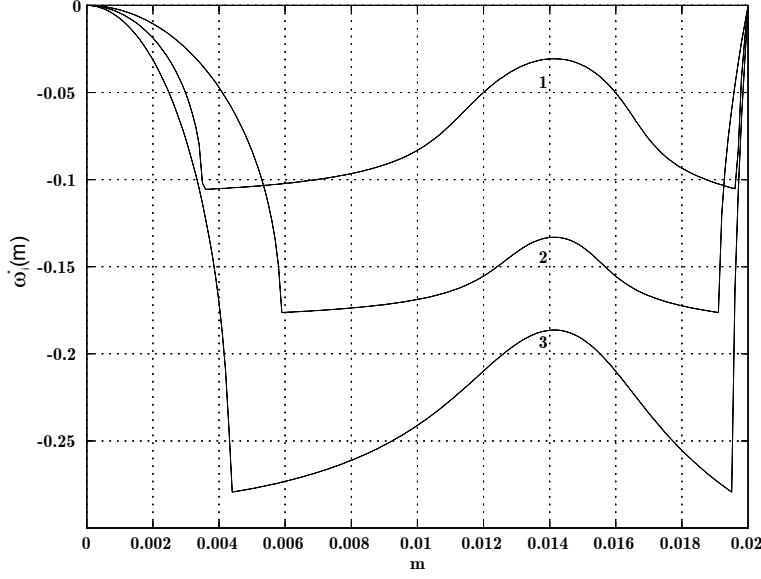


Figure 16. Stability when the load is varied. The three different plots correspond to the parameters found in table 3.1.

exist. The suspension in washing machines is not isotropic since the drum is usually standing on or hanging from springs almost vertically mounted. This will give a much lower natural frequency in the horizontal plane than in the vertical plane. It should be noted that using autobalancing techniques in washing machines requires two balancing rings, since the location along the axis of rotation of the unbalance is not static. Also there might exist dynamic unbalance in a washing machine. Using two balance rings solves both these problems. Another anisotropic case occurs when a grinding machine is operated. When a grinding machine is operated the suspension parameters will depend on the configuration and whether the grinder is in contact with the work piece or not.

The equations used in the anisotropic analysis are equations 2, 3 and 4 with the substitution 7. Using substitutions 5 and 6 will result in lengthy equations which are more time consuming to compute numerically than equations 2, 3 and 4 with the substitution 7. Substitution 7 is used so that the state, when the system is balanced, will be at zero speed in all state variables. Using substitution 7 the equations of motion will look like

$$M'\ddot{x}_1 + c_1\dot{x}_1 + k_1x_1 = me\omega^2 \cos(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\beta}_i \sin(\beta_i + \omega t) + (\dot{\beta}_i + \omega)^2 \cos(\beta_i + \omega t)), \quad (27)$$

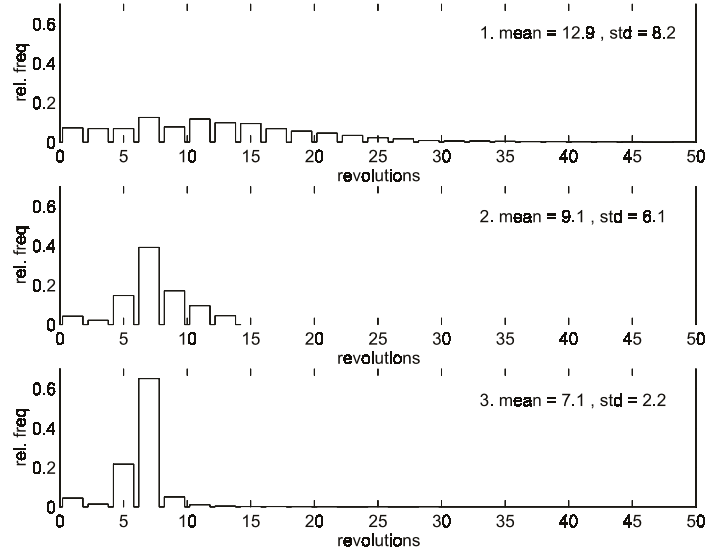


Figure 17. The relative frequency of number of turns before the amplitude is below one fifth of the maximum amplitude due to a simple rotating unbalance.

$$M'\ddot{x}_2 + c_2\dot{x}_2 + k_2x_2 = me\omega^2 \sin(\omega t) + \sum_{i=1}^{i=n} m_i l_i (\ddot{\beta}_i \cos(\beta_i + \omega t) - (\dot{\beta}_i + \omega)^2 \sin(\beta_i + \omega t)), \quad (28)$$

$$\begin{aligned} (m_i + \frac{I_i}{r_i^2})l_i\ddot{\beta}_i + l_i(\delta_i + \frac{2I_i}{r_i^2})\dot{\beta}_i &= m_i(\ddot{x}_1 \sin(\beta_i + \omega t) - \ddot{x}_2 \cos(\beta_i + \omega t)) \\ i &= 1, \dots, n \end{aligned} \quad (29)$$

where M' equals $(M + m + \sum_{i=1}^n m_i)$, as before, and is introduced since it occurs frequently. We still use the assumption that the parameters describing the compensating masses are equal and that the balancing device is capable of balancing the unbalance, i.e.

$$me \leq 2m_1 l_1. \quad (30)$$

The only equilibrium position in this coordinate representation will be

$$q_1 = 0, \quad (31)$$

$$q_2 = 0, \quad (32)$$

$$\beta_1 = \pi - \cos^{-1}\left(\frac{me}{2m_1 l_1}\right), \quad (33)$$

$$\beta_2 = \pi + \cos^{-1}\left(\frac{me}{2m_1 l_1}\right). \quad (34)$$

6.2 LINEAR STABILITY ANALYSIS

We now have an explicit time dependence in the equations of motion. The way to go for calculating the stability of the equilibrium position is to use Poincaré maps. The equations of motions can be written as

$$M(x, t)\dot{x} = f(x, t) \quad (35)$$

where $x = [x_1, x_2, \beta_1, \beta_2, \dot{x}_1, \dot{x}_2, \dot{\beta}_1, \dot{\beta}_2]^T$. Both $M(t, x)$ and $f(x, t)$ are $\frac{2\pi}{\omega}$ periodic, as can be seen from equations 27 to 29. It therefore seems natural to look at the configuration at intervals of $\frac{2\pi}{\omega}$ in t . The k 'th intersection of the Poincaré map will then be when $t = \frac{2\pi}{\omega}k$. The Poincaré map is then defined by

$$x_{k+1} = \Phi(x_k, 2\pi/\omega) \quad (36)$$

where $\Phi(x, t)$ is the flow of solutions i.e. $\Phi(x, t)$ represents all solutions to 35 with initial conditions $\Phi(x, 0) = x$. If we have a equilibrium x_0 , clearly $x_0 = \Phi(x_0, 2\pi/\omega)$. In the general case x may vary with time between the intersections. However, in our case the equilibrium is given by 31 to 34 and does not vary in time. Whether this equilibrium is stable or not depends on the derivatives of $\Phi(x_0, 2\pi/\omega)$ with respect to x . The criterion for a equilibrium position to be stable is that the eigenvalues of $\Phi_x(x_0, 2\pi/\omega)$ have a magnitude that is less than one. To calculate $\Phi_x(x_0, 2\pi/\omega)$ we start with inserting $\Phi(x, t)$ into 35, this gives

$$M(\Phi, t)\Phi_t = f(\Phi, t). \quad (37)$$

We then take the implicit derivative with respect to x , which result in,

$$M_x(\Phi, t)\Phi_x\Phi_t + M(\Phi, t)\Phi_{t,x} = f_x(\Phi, t)\Phi_x \quad (38)$$

In our case $\Phi(x_0, t) = x_0$, thus it does not depend on t and its derivative with respect to t is therefore zero. We are then left with

$$M(x_0, t)\Phi_{t,x} = f_x(x_0, t)\Phi_x. \quad (39)$$

Changing the order of differentiation of $\Phi_{t,x}$ and inverting M gives

$$\Phi_{x,t} = M^{-1}(x_0, t)f_x(x_0, t)\Phi_x. \quad (40)$$

which is a differential equation for Φ_x . The initial condition for Φ_x comes from differentiating $\Phi(x, 0) = x$ with respect to x , i.e. $\Phi_x(x, 0) = I$ where I is the identity matrix. This differential equation is, in our case, numerically integrated to produce $\Phi_x(x_0, 2\pi/\omega)$.

By looking more carefully at equations 27 to 28 we see that there exists a symmetry under the simple substitution

$$t \rightarrow t + \frac{\pi}{\omega}, \quad (41)$$

$$x_{1,2} \rightarrow -x_{1,2}, \quad (42)$$

$$\beta_{1,2} \rightarrow \beta_{1,2} \quad (43)$$

Par	Value	Par	Value
M	1.000	e	1.000
k_1	1.000	m_1	0.010
c_1	0.700	I_1/r_i^2	0.004
k_2	0.500	l_1	1.000
c_2	0.350	δ_1	0.005
m	0.010	γ_i	0.000

Table 3: Parameters used for the anisotropic stability analysis.

which will give equations 27 to 28 again. Therefore, we only have to integrate 40 over the interval $t : \epsilon : [0, \pi/\omega]$ and using the symmetry to calculate $\Phi_x(x_0, 2\pi/\omega)$. The symmetry can be written as

$$\Phi_x(x_0, 2\pi/\omega) = \left[\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Phi_x(x_0, \pi/\omega) \right]^2. \quad (44)$$

Since stability is governed by the eigenvalues of $\Phi_x(x_0, 2\pi/\omega)$ we define

$$\omega^{**} = \max_{i=1..2n} |\lambda_i|$$

where λ_i are the eigenvalues of $\Phi_x(x_0, 2\pi/\omega)$. The stability ω^{**} is then plotted as a function of the parameters we are interested in varying and when it is below one the equilibrium point is stable. The difference here compared to the isotropic case, found in section 5.3, is that ω^{**} will now show the amplitude decay per turn compared to ω^* , in the isotropic case, which show the exponential decay in time.

6.3 PARAMETER STUDY

We start our stability examination by varying ω for a system given by parameters from table 3 below. This particular choice of parameters will give a system where the natural frequencies will be quite close to each other and their ratio will be equal to $\sqrt{2}$. In figure 18 ω is varied from zero to seven. Note the similarities compared to figure 6. In previous stability analysis the region between the leftmost cross and middle cross was considered stable and the region between the middle and rightmost cross unstable. To the right of the rightmost cross the region was considered stable. By separating the natural frequencies in the horizontal and vertical direction with a higher ratio than given by the parameters in table 3 we will get a stable middle region. In figure 19 the ratio

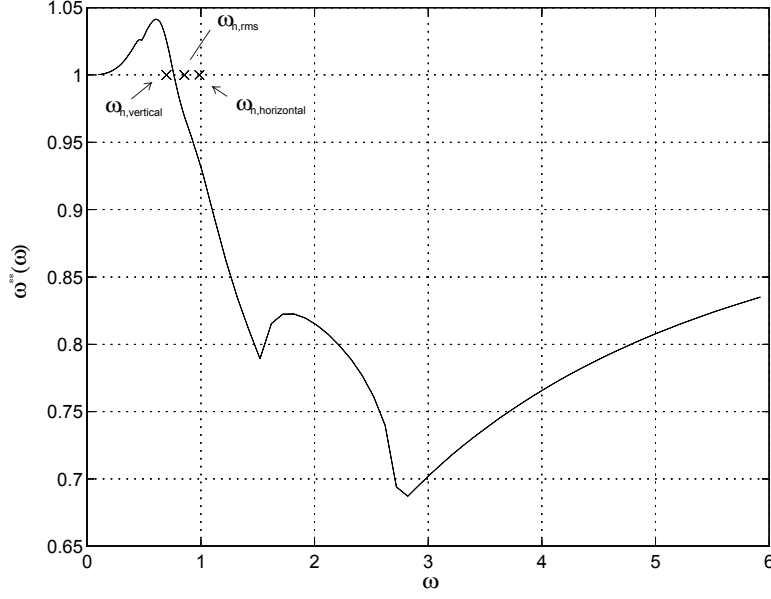


Figure 18. The stability as a function of the rotational speed. The crosses indicate the natural frequencies in the vertical direction and in the horizontal direction. The middle cross indicates the rms of the two frequencies. Parameters from table 4.1.

of natural frequency in the horizontal and vertical direction is 10. This explains why it is possible to mount an autobalancer more stiffly in one direction compared to the other and run it at a rotary speed between the natural frequencies in horizontal and vertical direction. In this case the previous stability results predicts the middle region quite accurately but fails to predict the stability in the high ω region. In general we will have a stable middle region when we have large separation of natural frequencies and no stable middle region when we have low separation. We can also infer from figure 19 that we should run an autobalancer with these parameters at $\omega \simeq 3.5$ if we want maximum decay in amplitude per turn. If instead we are interested in maximum decay in time we can run the autobalancer somewhere between $\omega \simeq 3.5$ and $\omega \simeq 6.3$.

6.3.1 Stability when the imbalance load is varied

The anisotropic case with varying load show the same behaviour as the isotropic case. In figure 20 the load is varied in the range where the autobalancer is capable of balancing. The figure shows the same features as the figure in the isotropic case. The local maxima at $m \simeq 0.014$ is still present, corresponding to a equilibrium position of the compensating masses where $\beta_1 \approx \frac{3\pi}{4}$ and $\beta_2 \approx \frac{5\pi}{4}$. It is possible by decreasing the viscous damping acting on the compensating masses to let the local maxima grow and become unstable. It is also useful to

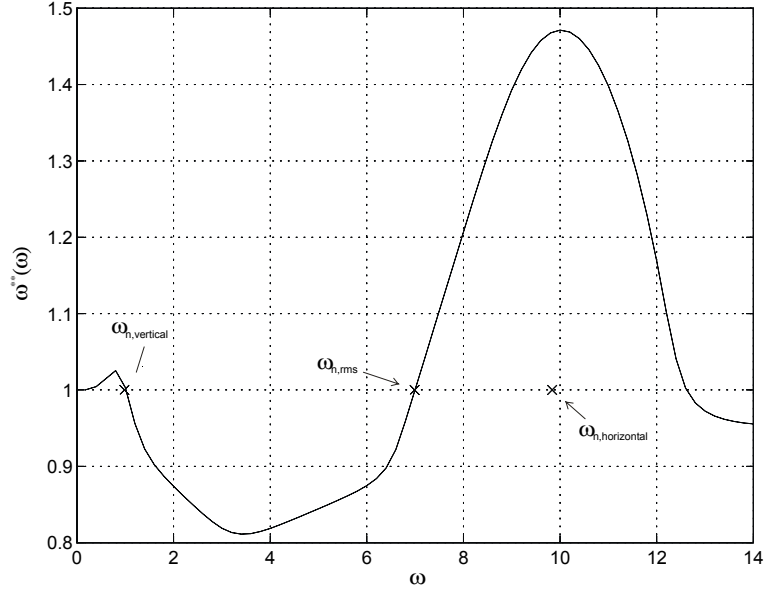


Figure 19. The stability as a function of ω when the spring constant in the horizontal and the vertical direction differs greatly. Parameter values from table 4.1 except for $k_2 = 100$, $c_2 = 0.7$ and $m = 0.014$.

examine the case when we are running at a rotational speed between the two natural frequencies. In figure 21, $\omega = 5$, which is between the two natural frequencies. The stability plot is similar to the other stability plots when m is varied.

7 Multiple correction masses

7.1 INTRODUCTION

Using only two correction masses is usually not volume efficient since there will be a lot of unused space in the balance rings. This, of course, depends on the design of the balancing ring, but since most balance rings use ballbearings with a radius which is less than the radius of the balance rings, more than two balls are more volume efficient. Using more than two correction masses results in a system capable of correcting for a higher imbalance load. The maximum number of correction masses to use is the amount which fills half the ring with correction masses.

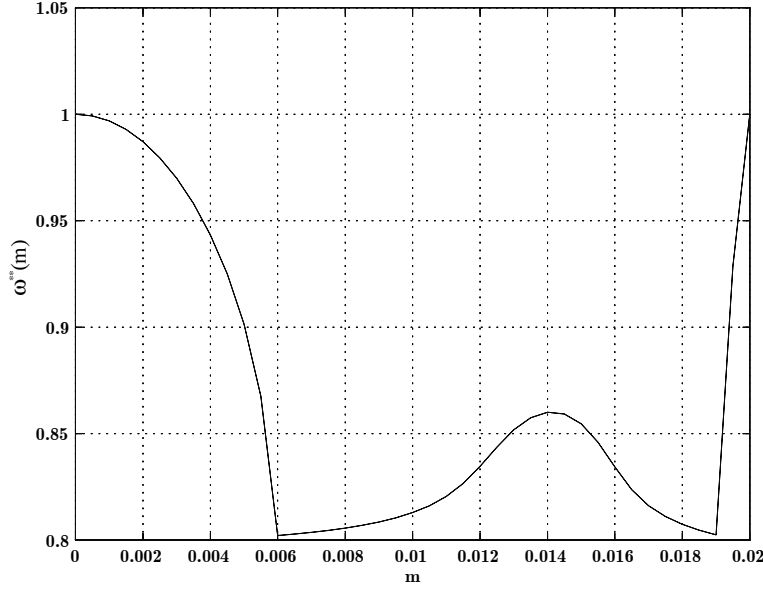


Figure 20. The result of varying the unbalance load by changing m . The angular velocity $\omega = 5$. The rest of the parameters from table 4.1.

7.2 EQUILIBRIUM POSITIONS

An aspect of using more than two correction masses is that we will get a family of equilibrium positions where the compensating masses correct for the unbalance. For example when we have three correction masses we will have a one parameter family of equilibrium positions. Figure 22 show that there will be a combined centre of mass of the unbalance and the third correction mass. The position of the combined centre of mass is labeled D in figure 22. We call this the new unbalance. By varying the angular position, angle β_3 , of the third correction mass, the point D will take different positions. In order to calculate the equilibrium position of the two other correction masses the distance from the geometrical centre to the centre of gravity of the new unbalance is needed. Labeling this distance, in the figure from B to D , with e' it can be expressed as

$$e' = \frac{\sqrt{(m_1 l_1 \sin \beta_3)^2 + (me + m_1 l_1 \cos \beta_3)^2}}{m + m_1}. \quad (45)$$

Now exchanging e for e' and m for $m + m_1$ the expressions for the equilibrium positions 11-12 are valid with the exception that an angle α is added to β_1 and β_2 . This angle compensates for the new angular position of the unbalance, see figure 22. Note that α is the angle between BC and BD and can be expressed as

$$\alpha = \text{atan2}(m_1 l_1 \sin \beta_3, me + m_1 l_1 \cos \beta_3) \quad (46)$$

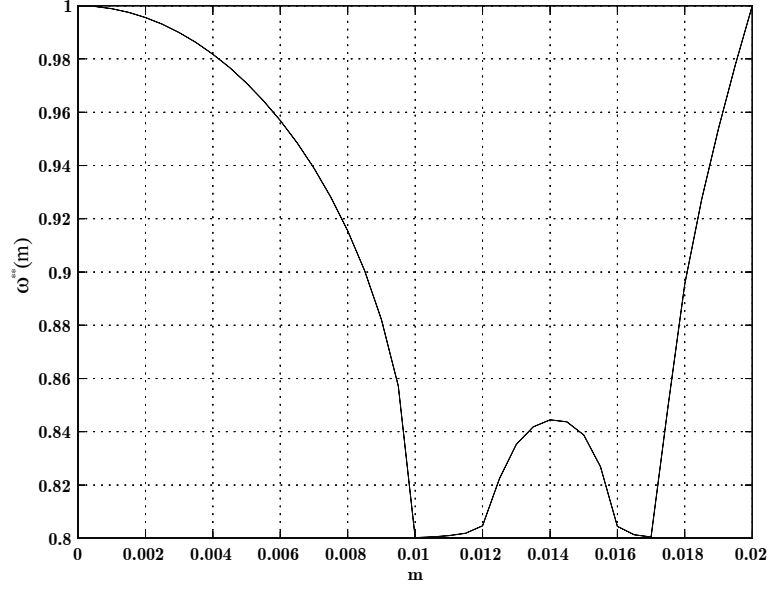


Figure 21. The result of varying the unbalance load by changing m when running at a revolutionary speed between the two natural frequencies. The angular velocity $\omega = 5$. Parameters from table 4.1 except $k_2 = 100$ and $c_2 = 0.7$.

where atan2 is the two argument arctangent function. We can not choose any value for β_3 since the last two correction masses must be able to correct for the new unbalance. This can be stated as

$$(m + m_1)e' \leq 2m_1l_1. \quad (47)$$

Inserting the expression for e' and equating this for β_3 gives

$$\cos \beta_3 \leq \frac{3(m_1l_1)^2 - (me)^2}{2mem_1l_1}. \quad (48)$$

This shows, as expected, that if the imbalance is less or equal to m_1l_1 we can freely vary β_3 . For the parameter study it is worth observing that we only need to vary β_3 in the region 0 to π since it will be symmetric to the region 0 to $-\pi$.

In the anisotropic case this equilibrium position will be the only one in this coordinate representation. However, the isotropic case will have some more equilibrium positions. To derive these equilibrium positions we return to equation 9. Setting all time derivatives to zero we get

$$-q_1 \sin \beta_i + q_2 \cos \beta_i = 0, i = 1, 2, 3. \quad (49)$$

The solutions of this, in terms of β_i are,

$$\beta_2 = \beta_1 + n\pi, : n = 0, 1, \quad (50)$$

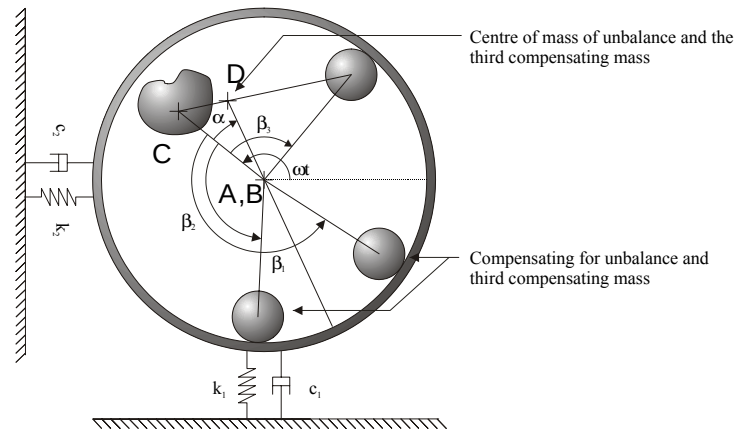


Figure 22. Using more than two correction masses will, depending on the unbalance load, result in a family of equilibrium positions. The equilibrium positions will therefore be indifferent.

$$\beta_3 = \beta_1 + k\pi, : k = 0, 1 \quad (51)$$

which corresponds to two physical solutions since $n = 0, k = 1$ is physically equal to $n = 1, k = 0$ and $n = 1$ and $k = 1$. The two physical solutions are that the correction masses are at the same angular position $\beta_1 = \beta_2 = \beta_3$ and that one of the correction masses is at the opposite side $\beta_1 = \beta_2 = \beta_3 - \pi$. If we would have four correction masses we would also have the case where there are two correction masses together on opposite sides. The solutions for q_1 and q_2 can easily be obtained from the equilibrium equations given in section 5.2. Obtaining the first solution when $\beta_1 = \beta_2 = \beta_3$ is done by replacing $2m_1$ with $3m_1$ in equations 17,18 and results in

$$q_1 = \frac{(k - \omega^2 M')(m_e \omega^2 + 3m_1 l_1 \omega^2 \cos \beta_1) + 3c\omega m_1 l_1 \omega^2 \sin \beta_1}{((k - \omega^2 M')^2 + (c\omega)^2)}, \quad (52)$$

$$q_2 = \frac{3(k - \omega^2 M')m_1 l_1 \omega^2 \sin \beta_1 - c\omega(m_e \omega^2 + 3m_1 l_1 \omega^2 \cos \beta_1)}{(k - \omega^2 M')^2 + (c\omega)^2}. \quad (53)$$

This will result in, in terms of β_1

$$(k - \omega^2 M')m_e \sin \beta_1 + c\omega m_e \cos \beta_1 + 3c\omega m_1 l_1 = 0 \quad (54)$$

which has a real solution if

$$(k - \omega^2 M')^2 + (c\omega)^2 \geq (c\omega)^2 \left(\frac{3m_1 l_1}{m_e} \right)^2. \quad (55)$$

This shows that depending on the choice of parameter values there might not exist a real solution. Equation 54 has two roots resulting in two physically different solutions. The next case, when $\beta_1 = \beta_2 = \beta_3 - \pi$, is similar to the above except that $3m_1$ has to be replaced by m_1 . The reason why this can be done is that the opposite side correction mass, labeled 3, will compensate for the correction mass labeled 2 leaving only correction mass number 1. This case will also result in two physically different solutions. *All in all we have five physically distinguishable solutions.* If we were to extend to use greater number of correction masses we would have for four correction masses, six equilibrium positions and five would give seven and so on.

These solutions only exist when both q_1 and q_2 are equal to zero. This is important since one could else believe that there exist solutions when $\beta_1 = \alpha$ and $\beta_2 = \alpha + \pi$. However this is not possible since compensating mass 3, angle β_3 , is not rigidly connected to the unbalance and would in this case start to move. This is the case for the other unbalanced solutions as well.

7.3 LINEAR STABILITY ANALYSIS

The stability analysis is performed by linearization of the equations of motion given by equation 8 and 9. The equations of motion are put in the form

$$M(x)\dot{x} = f(x) \quad (56)$$

A Study of Stability in AutoBalancing

Par	Value	Par	Value
M	1.000	e	1.000
k_1	1.000	m_1	0.010
c_1	0.700	I_1/r_i^2	0.004
k_2	0.500	l_1	1.000
c_2	0.350	δ_1	0.005
m	0 - 0.03	γ_i	0.000

Table 4: Parameters used for multiple mass stability analysis.

where x now equals $[q_1, q_2, \beta_1, \beta_2, \beta_3, \dot{q}_1, \dot{q}_2, \dot{\beta}_1, \dot{\beta}_2, \dot{\beta}_3]^T$. The linear stability is now governed by A , which equals

$$A(x_{0,i}) = M^{-1}(x_{0,i}) \frac{\partial f(x_{0,i})}{\partial x}, : i = 1, \dots, 5 \quad (57)$$

where $x_{0,i}$ are the five equilibrium positions given in the previous section. A can be derived analytically but the eigenvalues and eigenvectors are calculated numerically. Since the first equilibrium position, where the system is balanced and no vibrations exists, is indifferent, we expect that one of the eigenvalues of $A(x_{0,1})$ should be zero. The condition for stability is that the rest of the eigenvalues are less than zero. Therefore we define the stability as

$$\omega_j^{***} = \max_{i=1..2n-1} \text{Re } \lambda_i \quad (58)$$

where λ_i are the eigenvalues of $A(x_{0,j})$ and λ_{2n} is the zero eigenvalue and is therefore removed. It is important to check stability for all admissible values of β_3 since $x_{0,1}$ is a function of β_3 . If the stability condition holds for all admissible values of β_3 the system will certainly selfbalance if you are close to the equilibrium position. It is also important to check the stability of the other equilibrium positions since we don't want them to be stable in the operating region.

7.4 PARAMETER STUDY

The parameter study is complicated by the fact that we now have to vary not only the parameter we are interested in but also the equilibrium position. Starting with the parameters found in table 4 where we vary the load between 0 and 0.3 which is the maximum load the system is capable of balancing. The stability ω_j^{***} is plotted as a contour plot since this controls the asymptotic behaviour of small disturbances. In figure 23 the greyed area in the upper left corner corresponds to the condition found in 48 for β_3 . The zero at $m = 0.1$ and $\beta_3 = \pi$ is the result of another indifferent equilibrium position, namely that the balancing mass and correction mass are of equal mass. Since they are on opposite side the two other correction masses they may assume any position where they are on opposite side. From figure 24, where the stability for the five other equilibrium positions are plotted, we see that they are all unstable in the

regime where the autobalancer is capable of balancing the imbalance. When overloaded the balanced equilibrium cease to exist. Equilibrium type 5 then becomes the only stable one. This equilibrium position physically corresponds to the three compensating masses being at the opposite side of the unbalanced component. This means that the system is working as well as possible when overloaded.

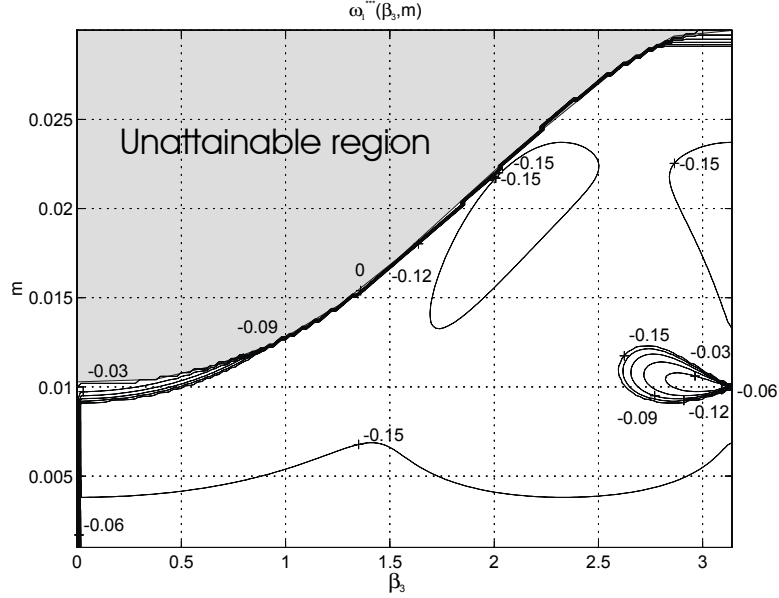


Figure 23. Contour plot of stability ω_1^{***} when m is varied from 0 to 0.3 and β_3 in the range 0 to the value given by equality in 5.4. Parameters from table 5.1.

It is also possible to get a system where stability depends on the amount of imbalance. This can be accomplished by lowering the damping acting on the correction masses. In figure 25 three regions are unstable when we lower δ_1 to 0.0007. For low amount of imbalance, m about 0.002, it does not exist any stable equilibrium. The case for m about 0.02 is somewhat better since there exist stable equilibrium for β_3 is some intervals. What might happen in a real system when starting in region 2 and 3 is that the correction masses might move outside the unstable region and stabilize in a stable region. This is somewhat supported by numerical simulations. However, this behaviour is not guaranteed.

8 Conclusions

This analysis has mainly been concerned with local stability analysis. One of the most important conclusions that one can draw from local analysis is that

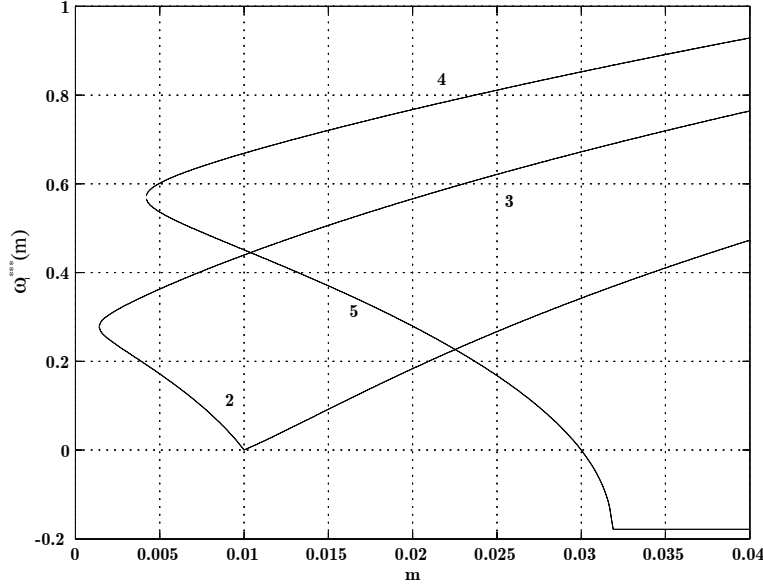


Figure 24. Plot of $\omega_{2.5}^{**}$ when m is varied from 0 to 0.4. Equilibrium position five become stable when m is above 0.3, which is the maximum amount of unbalance the system can correct for. This means that the system work as well as possible when overloaded. Parameters from tabel 5.1.

the natural frequency is not the border between stable and unstable balancing operations. However, it seems like running the autobalancer above the natural frequency is a necessary condition for stability.

The local analysis also tells us that the unbalanced equilibrium positions found in the isotropic case are not in general stable in regimes where the balanced equilibrium is stable. This would assure, for example in a system where the unbalance suddenly changes, that the correction masses would not move to an unbalanced equilibrium. The anisotropic case do not show these unbalanced equilibriums.

Another important conclusion is that the autobalancer work as well as possible when overloaded, i.e. all correction masses end up in a stable configuration at the opposite side of the unbalanced component.

This analysis also tells us something about the effects of internal damping compared to external damping. If we have high external¹ damping and low internal damping the autobalancer might be unstable. This effect is particularly noticeable when using two correction masses situated at an angle $\pm \frac{3\pi}{4}$ radians

¹ The damping from dampers are referred to as external damping and internal damping is the damping acting on the correction masses.

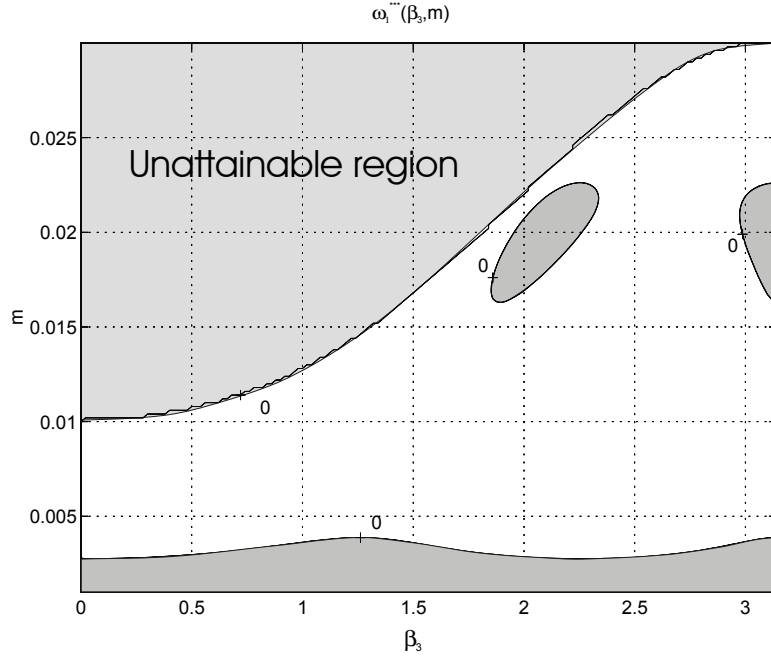


Figure 25. Contour plot of the stability when m is varied from 0 to 0.3 and β_3 in the range 0 to the value given by equality in 5.4. Parameters from table 5.1 except for the damping acting on the correction masses, $\delta_1=0.0007$. The dashed area are unattainable, see inequality 5.4. The dark grey area is unstable.

from the line connecting the geometrical centre and the centre of mass of the unbalanced component. This instability results in a periodic motion of the compensating masses about the balanced equilibrium position.

The anisotropic case, where we have a different spring constant and damping constant in the horizontal and vertical direction, shows a somewhat more complicated stability when the rotational speed is varied. We might then have, depending on the magnitude of separation in horizontal and vertical direction, alternating unstable and stable regions when the rotational speed is varied.

It is also shown that it is possible to extend this local stability analysis when more than two correction masses are used. However, we will then have indifferent equilibrium positions. The same instabilities show up in the multiple correction mass case, such as unstability due to low internal/high external damping and stability depending on the amount of unbalance.

Although local analysis does not tell what happens when the dynamics are far from the equilibrium position there seems to be a correspondence between the complete dynamics and the local dynamics. This is supported by

the numerical simulations. It therefore seems possible to use local analysis for guidance when designing an autobalancing system.

9 Acknowledgments

I would like to thank professor Martin Lesser for his support and guidance of this work. Without the Sophia code this work had been almost impossible.

I would also like to thank PhD Arne Nordmark for the time he has spent answering questions and explaining things to me about nonlinear dynamics.

I would also like to thank is Anders Lennartsson for using his routines that export Maple code to optimized Matlab code and all the rest of the people at the Department of Mechanics for their interest and suggestions.

Electrolux Research and Innovation let me use my research results obtained while working there and lent us the experimental auto balancer that I built as a part of this work.

Finally I would like to thank the Volvo Research Foundation's mechatronics grant that helped support this research.

Appendix A.

9.1 SOLUTION OF EQUATION 3.15

Equation 19 can be written as two equations,

$$(k - \omega^2 M')me \sin(\beta_1) + c\omega me \cos(\beta_1) + 2c\omega m_1 l_1 = 0 \quad (\text{A-1})$$

and,

$$\sin(\beta_1)^2 + \cos(\beta_1)^2 = 1 \quad (\text{A-2})$$

to get two well defined solutions. The solution can be written as,

$$\cos(\beta_1) = \frac{-bc \pm \sqrt{a^4 - a^2(b^2 - c^2)}}{a^2 + b^2} \quad (\text{A-3})$$

and,

$$\sin(\beta_1) = \frac{-b \cos(\beta_1) - c}{a} \quad (\text{A-4})$$

where,

$$\begin{aligned} a &= me(k - \omega M')^2 \\ b &= mec\omega \\ c &= 2c\omega m_1 l_1 \end{aligned} \quad (\text{A-5})$$

Bibliography

- [1] Adolfsson, J.: 1995, 'Self Balancing of Rotating Machines: An Experimental and Theoretical Study'. Master's thesis, KTH, Department of Mechanics.

- [2] Kravchenko, V. I.: 1983, 'Stability analysis of a row-type counterbalance'. *Mashinovedenie* (1), 25–27.
- [3] Kravchenko, V. I.: 1986a, 'Automatic balancing of rotor of multi-mass system with row-type automatic ball balancer'. *Mashinovedenie* (2), 95–99.
- [4] Kravchenko, V. I.: 1986b, 'Improving the dynamic reliability of loaded machinery elements by self balancing'. *Tyazheloe Mashinostoenie* (5), 19–21.
- [5] Thomson, W. T.: 1981, *Theory of Vibration with Applications, 2nd edition*. Prentice Hall.

Paper 2

P2

A short background to artificial walking

Jesper Adolfsson
Royal Institute of Technology
Department of Mechanics
S-100 44 STOCKHOLM
SWEDEN

November 20, 2000

ABSTRACT. This short background on artificial walking is meant to present some of the ideas that have influenced the author and others regarding artificial walking and in particular passive walking. Results from different fields of research are given below, which in some cases directly support the idea that human walking is largely a consequence of the geometry and the mass distribution. Some results are just presented because they are interesting in themselves. These research fields include medicine, biomechanics, toydesign, and robotics.

1 Introduction

Upright walking is the natural way of locomotion given to us by nature. The evolution from walking/climbing on all four extremities to walking on our legs freed the upper extremities to deal with other important tasks. Certainly, this gave us advantages compared to species still forced to walk on all four extremities. For example, we could now use our hands while we where moving around, such as hunting with tools in our hands. It was now also possible to cover relatively large distances on land, thus being able to move to new habitats as needed. Footprints and found limbs indicate that the upright posture was quite well established between 3,000,000 to 3,500,000 years ago (encycl.brittan). The earliest known evidence of bipedal gait are footprints in volcanic ash found in northern Tanzania and is dated 3.6 million years ago. These advantages certainly put an evolutionary pressure on further developing and refining our walking system. At first sight, it would seem that this change increased the complexity of our locomotion system, such as keeping the upright posture.

However, in some sense the complexity was lowered, such as only having to control two legs instead of four. In addition, as this thesis and other research in the field of passive walking all indicate, the complexity of bipedal walking can, to a large extent, be explained by the geometrical makeup of the walking mechanism. Thus, complex control strategies are not needed.

2 Biology

An indication that human gait is different from quadruped gait is evident from research in biology. For example, the role of the central pattern generator (CPG) has been studied extensively by biologists and has been used, for example, as a model for generating gait sequences in quadruped robots. The role of the CPG is to generate the locomotor stepping pattern autonomously. The CPG can change behaviour depending on speed requirements. In humans, it seems like the role of the CPG has been reduced to generate the first stumbling steps of infants, see Gait Analysis [5]. Experiments with cats have shown that it is still possible to generate walking patterns and simple obstacle avoidance after the spinal cord has been transected. However, in paraplegic humans it has not been possible to get the same walking patterns. Thus, it is believed that higher brain centers are responsible for the control of the locomotion in humans. Thus, it can be speculated that the reduced role of the CPG in humans is a consequence of the human capability of largely generating the walking motion passively. This also indicates that the human capability of bipedal walking is not innate but learned.

The activity of normal walking is something we perceive as requiring very little conscious effort. How is it then that something so deceptively simple as walking still poses such engineering difficulties? One of the problems is that we don't really know where to start searching for the answers. Before it was possible to record gait motion researchers were mainly forced to study humans in action. In 1836, the Weber brothers proposed the idea that the motion of the leg, during swing phase, could be explained by the motion of a double pendulum in a gravity field, see Human Walking [25]. This was observed by letting a human leg hang freely and releasing it from a pre-swing position and noting the similarities to a leg during normal walking, such as swing time and motion. Since this was before photography was available, the measurements had to be accomplished by using a hand-operated clock.

Some years later this idea was challenged by results from artificial stimulation of muscles and the fact that patients with paralysis of the muscles in the leg had to compensate during the swing phase or the foot would hit the ground. Thus, if the swing phase was purely passive the compensating movement of the hip shouldn't be necessary. However, it is possible that the relaxed position of the ankle made such a compensating movement necessary.

Later in the 19'th century, still photography was used to record the motion of different subjects, such as animals and humans in action. However, these images were just motion frozen and it was not until Muybridge in 1873 developed his technique of taking pictures in a sequence, that the whole motion could be captured. The original use of the Muybridge technique was to settle a bet whether a horse was completely in the air during gallop or not. By using electrical switches sequentially arranged and connected to a battery of cameras, it was shown that the horse indeed had all its feet in the air. This technique was later used by Marey to record the walking motion of humans. He used his results to calculate the mechanical energy needed for sustained walk for different walking speeds. The photographic technique was further developed and using stereometry photography, it was possible to study the motion in three dimensions. This technique combined with mass properties measurements of cadavers made it possible to estimate the forces that were needed for propulsion, see Human Walking [25].

One of the reasons for studying walking from a medical point of view comes from the need of differentiating pathological walk from ordinary walk. Thus, large quantities of data have been collected. These data range from foot pressure during the stance phase to the motion of various parts of the human body. This data is then compared with data obtained from patients and gait pathologies can be discovered, see Human Walking [25] and Gait Analysis [5].

Another common use of this data is to estimate the internal forces in humans, such as the forces in the skeleton, tendons, and muscles. This is accomplished by making a mathematical model having the right mass properties and then using the data as input for the motion of the model. From an engineering point of view, this data can be used to compare artificial walkers and their movement with real data.

3 Walking aids

Another driving force in walking research is the construction and development of walking aids. Many different designs have been implemented as to account for different kinds of disabilities, such as above- and below-knee amputees, see Human Walking [25]. Various mechanical designs have been developed, some of them with the intention of functionally duplicating the original limb as close as possible and in some cases as substitutes. From a mechanical engineering point of view some interesting designs are;

- Using four-bar-linkage knee mechanisms give a wide range of stability choices. Thus, the amputee can control the knee collapse.
- Flexible foot to make smoother impacts with the ground.
- Spring mechanisms for running.

It has been common to construct lightweight prosthesis, believing that it would be simpler to handle for the amputee. However, the effect of different mass properties has been studied by letting subjects walk with their own

lightweight prosthesis as well as ones with added mass. The result was that the subjects feeling increased when the mass properties were chosen as to resemble those of the original limbs, see Human Walking [25]. This indicates the importance of having the appropriate mass distribution of the walking mechanism.

It is also interesting to mention the device constructed by Nichols and Witt [20] to help children that had no legs. These children had usually been fitted with swivel-walkers or rockers. This aid consisted of a seat with a pair of pylons attached to it. These pylons had feet that were large enough as to allow the child to stand. The feet attached to these pylons could rotate about a vertical axis. Thus, the children could move forward by rocking the device so that they were standing on one foot and at the same time twist with their upper body as to move the other foot forward. However, the forward motion was slow and Nichols and Witt came up with a device having swinging legs that could increase the forward speed. The new device had legs that used a parallelogram linkage designed as to obtain a heel impact at the end of the swing phase. A crossbar was mounted between the legs so that when one leg swung forward the other one had to swing backwards. This provided the stability needed to stand still. To prevent collapse at the swing phase, a torsion spring was mounted between the crossbar and the seating. This prevented a too large opening angle between the legs. By tuning this spring in relation to the mass of the carrier, different walking frequencies could be obtained. What is interesting with this device, in relation to walking machines, is that a mechanical torso was built so as to provide the rocking motion. This actually turned the device into a bipedal walker. The walker, with the artificial torso, obtained a speed of 0.23m/s compared to 0.33m/s for a human rider. This can be compared to the previous swivel walkers, where speeds of 0.05m/s up to 0.18m/s had been obtained. The power consumption was as low as 4W, disregarding electrical and hydraulic losses.

4 Merits of walking

There are several merits of walking machines compared to their wheeled counterpart. The most obvious ones relate to the legs capability to avoid obstacles. The flexibility of movement in legs also gives the possibility of putting down the foot at a specified location. This can be favorable in poor ground conditions, such as avoiding soft ground in favor of more solid ground. During slippery conditions, legged vehicles can excel over wheeled ones by taking smaller steps, as to minimize horizontal forces. Depending on the design of the legs, such as the number of degrees of freedom at each joint, gives a great flexibility of positioning the body. This can be important in complex environments where difficult manipulation tasks have to be accomplished. In addition, using legs can reduce the impact on the surroundings. For example, the forest robot by Plustech Oy [21] is much gentler on its surroundings than a traditional forest machine using wheels. Thus, there are several aspects of legged walking that makes it both interesting and preferable over its wheeled counterparts. How-

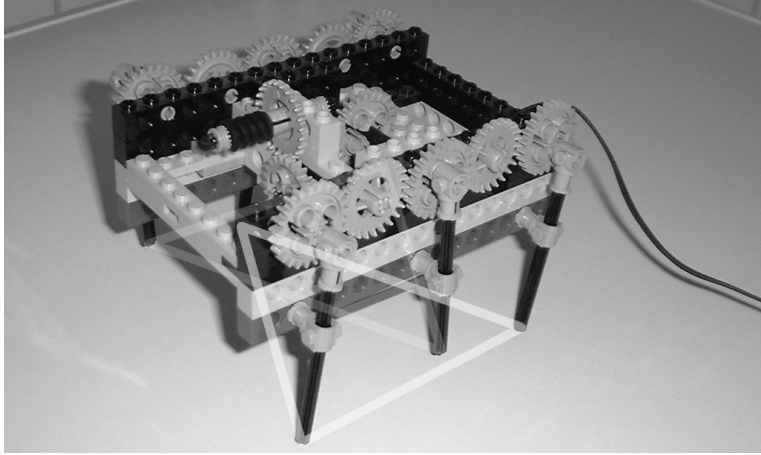


Figure 1. A picture of a statically stable lego walker. The current support pattern is shown by the brighter triangle. The lighter triangle is the other support pattern. These two support patterns will succeed each other, hence moving the walker forwards or backwards.

ever, one should not forget that wheels excel when it comes to speed, economy and a much lower complexity in terms of understanding and construction.

5 Walking machines

Walking machines can roughly be divided into statically stable walkers and dynamically stable walkers. Loosely speaking, statically stable walkers try to keep their projection of the center of gravity within the support pattern, see Figure 1. As long as they can maintain this condition and have enough traction at each foot, they will function. Their movement is accomplished by going from one stable state to a new stable state. Usually they move at low speeds and can therefore neglect inertial forces that could potentially destabilize the motion. These walkers come in many different shapes, everything from quadrupeds to insect like versions with many legs. For a nice introduction to walking machines see for example Todd [26]. For a good overview of recent walking machines, see for example Hardarson [10].

Dynamically stable walkers have the capability of letting the center of gravity move outside its support pattern without losing stability. Stability is in some cases obtained by using control laws and actuators and in other cases by designing the walker to have inherent stability. The latter are usually referred to as passive walkers. Controlled walkers in this category usually take into account inertial forces occurring as they move. Both dynamically stable bipeds and quadrupeds exist.

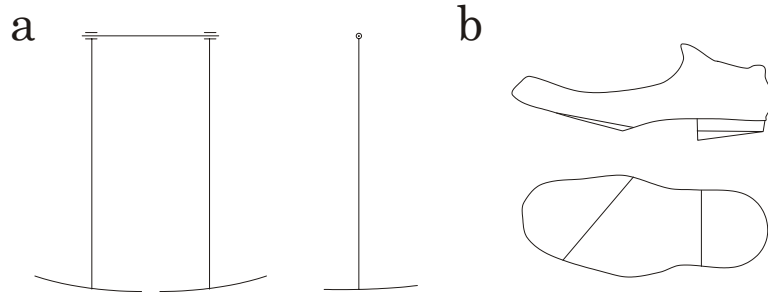


Figure 2. Pane (a) shows the design of a toy walker by Fallis. Pane (b) shows the construction of Bechstein and Uhlig that supposedly generated the rocking motion of the toy walker.

6 Historic and contemporary walking machines

The history of walking machines starts with different kinds of toy walkers. The toys were usually driven by clockworks that made the toy perform a repetitive motion as to propel it forward. Other toys used weights that were attached to the walker via a string. For example, by hanging the weight over the edge of a table, the force needed to move it forward was obtained. These toys included bipedal as well as quadrupeds and similar toys are still manufactured.

At the end of the 19'th century, toys capable of motion resembling human gait started to appear. One of these toys is described in the patent of Fallis, dated 1888, see reference [6]. This patent describes a toy that uses straight knee-less legs hinged at the shoulders. The feet had gently curved surfaces as to prevent the foot to hit the floor during the other legs swing phase, see Figure 2. The propulsion of this walker came from letting it walk down an inclined plane. In 1912, Bechstein and Uhlig filed a patent, see [2], describing a similar bipedal toy walker. Instead of using gently curved feet, they used a sole, which had differently inclined planes as to generate the side-to-side rocking motion. This patent later inspired Wisse in 1998 to build a kneed walker that could walk down an incline, see reference [29].

A driving force, partly associated with robotic walking, has been to overcome different limitations in humans. Such as increasing the strength and endurance of humans and helping people with disabilities. In the early sixties, General Electric undertook an investigation on how to build a bipedal walking machine capable of carrying a human operator. The idea was to let the operator have full control of the motion of different parts, such as legs and body. Since the intended size was much larger than a typical human, it was decided that it should be actuated by hydraulics. The operator controlled the hydraulics via force feedback devices. This project was considered possible (at least in the mechanical engineering sense) but it was never implemented due to other designs being more appealing, such as the four legged General Electric

walking truck. It is unfortunate that this walker never was built since if the operator would have been able to control the biped some interesting data could have been obtained. Such data could have given us the control laws used for keeping the balance, control laws regarding start and stop and so forth. This data seems to be very complicated to obtain from humans. As mentioned before, what was actually implemented by General Electric was the four-legged walking truck. This was controlled in a similar way to the proposed bipedal robot. The hind legs were controlled by the operators legs and the fore legs by the arms of the operator. However, this device put such a control load on the operator, thus, making it possible to operate for only a few minutes.

Another interesting device that was built by General Electric was the Hardy-Man. It was an exoskeleton capable of amplifying the force generated by the operator. This made it possible to lift heavy things. These three devices had in common that they were completely operator controlled, thus lacking any kind of higher control system to offload the operator.

Partly building on the experience from the walking truck, the adaptive suspension vehicle was constructed by the Ohio State University in collaboration with Adaptive Machine Technologies and the Stanford University, see reference [23]. This walker had six legs and walked in a statically stable manor. It had a number of operating modes specialized for different tasks. In normal operation over smooth terrain, it was possible to control elevation, heading and position in the plane, thus offloading the operator from the complex task of moving each leg to obtain a gait.

In the light of the successful implementation of statically stable walkers using control laws and actuation it is not too far fetched to believe that this would also work for designing dynamically stable walkers. One of the earliest bipedal robots is the Wabot by the Useda University [27] that was constructed in the early seventies. It was capable of walking, although always keeping its center of mass within the support pattern of the foot currently standing on the ground. Its size was comparable to a human and it used hydraulics to move its limbs. The most successful, up to date, is the famous Honda bipedal walking robot [11]. Little is known about this robot since it was developed internally at Honda. It is also believed that it took more than a decade to develop. Nevertheless, watching movies where it performs different tasks such as walking and climbing stairs is certainly impressive. Both these robots have in common that the limbs are directly controlled by servos, the Honda robot being controlled by direct drive servos. Since electrical motors usually have low torque and high speed, some sort of gearing is necessary. This gearing both amplifies inertia and friction, thus making it hard to control the actual force output. Instead, it is most common to control position/angles of different joints. The drawback of this is that it makes it hard to use the natural dynamics, such as the motion of a swinging leg, available to the mechanism. To address the problem of getting force control, the series elastic actuator was developed, see Williamson *et al.* [28]. The idea is to connect a servo in series with an elastic element, such as a spring. By controlling the elongation of the spring

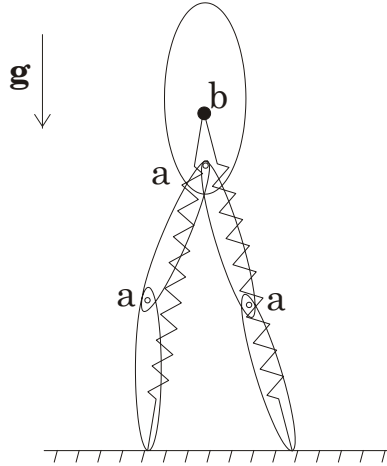


Figure 3. The springs represent the virtual force, which is generated at the joints labeled a. The attachment point of the virtual springs is labeled b.

it is possible to generate forces. Thus, turning the force control problem into a position control problem. This gives excellent control over the force but lowers the bandwidth of the controller. At least two planar walkers have been implemented using these actuators, the spring turkey and spring flamingo, see Pratt [22]. The walking motion was generated using something called virtual model control. The idea is to attach virtual springs or force elements between different parts of the robot and then calculate the corresponding forces/torques that realizes these virtual forces.

7 Hopping robots

Humans as well as animals usually have different walking patterns depending on the velocity. At low speeds, the walking gait is usually preferred and can be categorized by having a double support phase. As speed is increased, humans change from walking to running. Running can be described as having no double support phase and periods of no contact with the ground. The velocity at which the transition from walking to running occur can be obtained by theoretical reasoning, see Alexander [1],

$$v_{\text{transition}} < \sqrt{gl}, \quad (1)$$

where l is the length of the extended leg. For an adult human this corresponds to about 3 m/s. However, by moving the hip in an appropriate way, higher walking speeds can be obtained.

Running or hopping robots have been implemented with great success. These robots were pioneered by Raibert [24]. Starting with the one legged

A short background to artificial walking

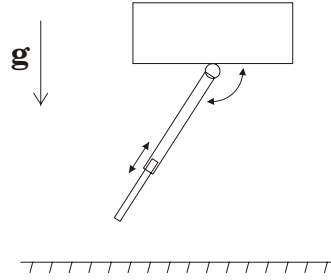


Figure 4. Raibert's hopping robot. The arrows show the movement that is controlled.

hopper which was attached to a tether boom so as to prevent lateral dynamics. Later versions had both two and four legs and were capable of moving in all three dimensions. One interesting aspect of these robots is that the number of controlled degrees of freedom is less than the mechanisms degrees of freedom and that the control does not depend on a reference trajectory.

8 Passive walkers

At the extreme of little or no control are the mechanisms that are referred to as passive walkers, originally pioneered by McGeer [14],[15],[16], and [17]. Passive walkers completely lack any kind of control and rely completely on the geometry and appropriate mass distribution. By letting the passive walker walk down an inclined plane gravity re-injects the energy lost at various stages of the walking cycle. A typical source of dissipation is the plastic impacts at knee lock and foot impact. A precursor to passive walking is ballistic walking, studied by Mochon *et al.* [18] and Formal'sky [7] for example. These walkers were usually studied on level ground. The term ballistic is used to describe that no actuation is used during the swing phase of the leg: thus the leg swings freely. One of the motivations for this comes from biological studies showing that muscle activity is very low during swing phase and is concentrated to the double support phase, see Human Walking [25] or Gait Analysis [5]. The task in ballistic walking is to find a set of initial velocities that return the walker into the same configuration after one step (with a switch of stance and swing leg). Since the final velocities will not match the initial velocities and impulse is introduced at the double support phase to make up for this. Animations of ballistic walkers produce a very human like walking pattern.

McGeer later showed that the impulse could come from the impact of the swing leg with the ground. This was accomplished by finding the appropriate mass distribution and geometry, as well as finding the right inclination of the plane. The first passive walker by McGeer was kneeless and constrained to move in a plane so as to prevent lateral dynamics. Since both legs had the

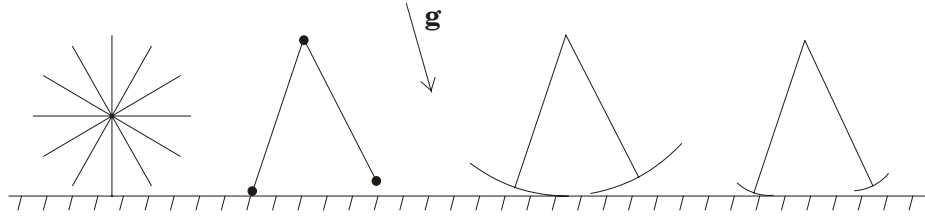


Figure 5. Some of the studied simple planar passive walkers; McGeers spoked wheel, Ruinas simplest passive walker, McGeers two legged walker having a foot radius equal to the leg length, and a similar walker but the radius is shrunk.

same length, foot scuffing at mid swing was neglected. Implemented walkers had devices that could contract the leg slightly so as to prevent the foot scuffing. McGeer later showed that it was possible to add a knee. This gave the walker a more human look and, as it turned out, avoided the foot-scuffing problem. Since simplifying can be a key to understanding, many simple models have been proposed as means to understand walking. Some examples of these are;

- Rimless wheels (McGeer) which models impacts and the up and down motion of the hip.
- The three point mass walker (Garcia [8]). Consisting of one hip mass and two masses situated at the foot. Shows chaotic walking patterns.
- The walking wheel (McGeer), which models the un-actuated swing of the legs.
- Straight legged bipedal walker (McGeer). Similar to the walking wheel but had a shorter foot radius.

Planar passive walkers have been implemented with great success by for example McGeer and Ruina and his group. Usually, lateral dynamics have been suppressed by building legs in pairs. The walker is started in a configuration where the outer legs move together and the inner legs are mechanically joined.

Sustained passive walking on level ground is not possible unless energy is constantly injected into the system as to counter the energy lost at impacts. However, one could of course imagine a walker capable of putting down its feet on the ground with zero velocity and having knee lock at zero angular velocity. Blajer and Schiehlen [3] have developed control strategies as to prevent relative velocities at foot impacts. However, the goal was not to minimize energy costs but to stabilize and find gentle motion of the walker. It is not at all clear how to mechanically build such a device since it will certainly involve some clever mechanism needed to obtain zero velocity impacts. However, there exists at least one mechanism, albeit not being a walking machine, which have these properties and that is the rolling wheel. In order to obtain walking on level ground some schemes have been proposed which are based on the passive walkers. For example, McGeer used impulsive pushing to model the push of

the trailing foot. He also suggested length cycling of the leg, as to mimic the push generated by knee flexure.

Adding a torso has also been used as a mean of propulsion. McGeer used an upright torso, which kept its position by using a rotational-spring/damper attached to the stance leg. Leaning the torso forward provided the additional torque needed to propel the walker. Where McGeer had studied the effect of different torso-inclinations, Howell and Baillieul [12] fixed the position of the torso in a bowed position and studied the effects of changing the center of mass position.

A natural next step was to find three-dimensional walkers capable of sustained stable gait down an incline. McGeer devised a walker capable of lateral motion but only found unstable periodic gaits. Kuo [13] used one of McGeer's kneeless designs and extended it laterally by including a hip. However, Kuo only found unstable walkers and proposed several control laws for stabilizing the motion. These control laws have been shown to work in simulations. In 1997, Coleman and Ruina [4] built an unpowered three-dimensional kneeless two-legged walker out of Tinkertoy parts. This walker is apparently stable (as seen in video clips) and has recently been shown to be stable, see [19]. The mass properties of the walker are not very human like, but clearly shows the possibility of passive walking in three dimensions. It is worth mentioning that some researchers in this field divide passive walkers into two categories depending on their stability properties. The first category includes walkers capable of standing upright when standing still, thus having at least one statically stable configuration. The second category are the ones that have to constantly move as to obtain stable behavior, such as the Tinkertoy Walker, which only has one contact point per foot. Whether the difference is fundamental or merely esthetical can be debated. One could argue in favor of passive walkers having statically stable equilibrium positions so that a passive walker would be able to stand passively and not have to rely on control and actuation for static equilibrium.

9 Animation

In most cases the simulation is used as a tool to understand or develop walking machines. However, in some cases a believable simulation is the goal in itself. Today most computer animations of walking are generated by using motion capture devices or keyframing. Motion capture is usually accomplished by putting markers on a person, who then performs the motion that should be recorded. However, this is both time-consuming and sometimes hard to get accuracy in readings, due to markers put on the skin and not on the joints [9]. Keyframing is accomplished by the animator specifying the geometric configuration at specific times and letting the software interpolate the in-between sequence. In addition, it is not possible to just increase the walking speed of a previously captured motion, since the resulting animation will not be believ-

able. However, simulations of walking and running offer a new way of obtaining realistic and believable animations.

Bibliography

- [1] Alexander, R. M.: 1995, ‘Simple models of human movement’. *Applied Mechanical Review* **48**, no8.
- [2] Bechstein and Uhlig: 1912, ‘Patent: Improvements in and relating to Toys’.
- [3] Blajer and Schiehlen: 1992, ‘Walking without impacts as a motion/force control problem’. *Journal of Dynamic Systems, Measurement, and Control* **114**, 660–665.
- [4] Coleman, M. and A. Ruina: 1998, ‘An Uncontrolled Toy That Can Walk But Cannot Stand Still’. *Physical Review Letters* **80(16)**, 3658–3661.
- [5] Craik, R. L. and C. A. Oatis (eds.): 1995, *Gait Analysis: theory and application*. Mosby-Year Book, Inc.
- [6] Fallis: 1888, ‘US Patent No. 376,588: Walking Toy’.
- [7] Formal’sky, A. M.: 1996, *Ballistic Locomotion of a Biped. Design and Control of Two Biped Machines*. Udine: CISM Advanced School on Modelling and Simulation of Human and Walking Robots Locomotion.
- [8] Garcia, M., A. Chatterjee, A. Ruina, and M. Coleman: 1998, ‘The Simplest Walking Model: Stability, Complexity, and Scaling’. *ASME Journal of Biomechanical Engineering* **120**, 281–288.
- [9] H., H. J.: 1998, ‘Animating Human Motion’. *Scientific American*.
- [10] Hardarson, F.: 1999, ‘Principles of Legged Robots’. Technical report, KTH, Mechatronics Lab.
- [11] Honda, ‘Honda Bipedal Robot’. <http://world.honda.com/robot/>.
- [12] Howell, G. W. and J. Baillieul: 1998, ‘Simple Controllable Walking Mechanism which Exhibit Bifurcations’. In: *IEEE Conf. on Decision and Control, December 16-18, Tampa, FL*. pp. 3027–3032.
- [13] Kuo, A. D.: 1999, ‘Stabilization of Lateral Motion in Passive Dynamic Walking’. *The International Journal of Robotics Research* **18(9)**, 917–930.

- [14] McGeer, T.: 1990a, ‘Passive Bipedal Running’. In: *Proceedings of the Royal Society of London: Biological Sciences*, Vol. 240. pp. 107–134.
- [15] McGeer, T.: 1990b, ‘Passive Dynamic Walking’. *International Journal of Robotics Research* **9**, 62–82.
- [16] McGeer, T.: 1990c, ‘Passive Walking with Knees’. In: *Proceedings of the IEEE Conference on Robotics and Automation*, Vol. 2. pp. 1640–1645.
- [17] McGeer, T.: 1993, ‘Dynamics and Control of Bipedal Locomotion’. *Journal of Theoretical Biology* **163**, 277–314.
- [18] Mochon, S. and T. McMahon: 1980, ‘Ballistic Walking: An Improved Model’. *Mathematical Biosciences* **52**, 241–260.
- [19] Mombaur, K., M. J. Coleman, M. Garcia, and A. Ruina: 2000, ‘Prediction of Stable Walking for a Toy That Cannot Stand’. submitted for publication in PRL.
- [20] Nichols, G. K. and D. C. Witt, ‘An Experimental Unpowered Walking Aid’. *Engineering in Medicine* **1**(1).
- [21] OY, P., ‘Forrest machine’. <http://www.plustech.fi>.
- [22] Pratt, J. E.: 1995, ‘Virtual Model Control of a Biped Walking Robot’. Master’s thesis, MIT, Department of Electrical Engineering and Computer Science.
- [23] Pugh, R., E. Ribble, V. Vohnout, T. Bihari, T. Walliser, M. Patterson, and K. Waldron: 1990, ‘Technical Description of teh Adaptive Suspension Vehicle’. *The International Journal of Robotics Research* **9**(1).
- [24] Raibert, M. H.: 1986, ‘Legged Robots that Balance’. *MIT Press, Cambridge, Mass.*
- [25] Rose, J. and J. G. Gamble (eds.): 1994, *Human Walking*. Williams & Wilkins.
- [26] Todd, D. J.: 1985, *Walking Machines: An Introduction to Legged Robots*. Kogan Page Ltd.
- [27] University, W., ‘Wabot’. <http://www.humanoid.waseda.ac.jp/index.html>.
- [28] Williamson, M.: 1995, ‘Series Elastic Actuators’. Technical report, M.I.T. Department of Electrical Engineering and Computer Science.

- [29] Wisse, ‘Video of bipedal walker having knees’. <http://www.ccmr.cornell.edu/ruinalab/pdw.html>.

Paper 3

P3

Planar Passive Walking

Jesper Adolfsson	Arne Nordmark
Royal Institute of Technology	Royal Institute of Technology
Department of Mechanics	Department of Mechanics
S-100 44 STOCKHOLM	S-100 44 STOCKHOLM
SWEDEN	SWEDEN

July, 2000

ABSTRACT. The planar passive walker was used as a starting point for the 3D passive walker. This system is modeled as a continuous system, which is regularly interrupted by collisional events such as knee collision and foot impact. Besides the collisional events, events which will terminate the gait sequence are also monitored. Hence, this report contains modelling assumptions for our planar passive walker, which includes geometry, forces and impact models.

Previously presented planar passive walkers had large radius feet, which was not suitable for extension into 3D. This report therefore presents the parameter variations resulting in a planar passive walker suitable for extension into a 3D passive walker. The linear stability of the system is shown for all parameter variations. The equations of motion have been derived using *Sophia* and *Maple* and this code is included.

1 Introduction

In this text, we show how passive walking of planar mechanisms studied by, for example, McGeer [11],[12],[13], and Ruina and collaborators, see Garcia [6], can be utilized as a starting point for 3D passive walkers, see also Dankowicz et al.[4]. However, it was felt that a thorough understanding of planar passive walking was necessary before modelling and simulating a 3D passive walker. Therefore, previous work by McGeer and Ruina was recreated using the tools developed at the Department of Mechanics, KTH. These tools include *Sophia*, which is a set of routines for deriving equations of motion in symbolical algebra programs, see Lesser [9], and the *exmex* routines used to export the equations of motion to Matlab, see Lennartsson [8].

Since the walker makes regular contact with the ground, impact dynamics is introduced. All impacts are treated as perfectly inelastic collisions, thus the velocities of the impact points are assumed to have zero relative velocity after the impact has occurred. Simulation of the resulting dynamical system

was performed in Matlab using it sets of ordinary differential equations (ODE) solvers. The dynamical system is assumed to be continuous between impacts, thus event handling is used to end the simulation as impacts occur. In addition, events that terminate the gait sequence are also monitored, for example such as the hip point hitting the ground. It should be noted that some differences exist between the present model and those studied in the above mentioned references, such as more restrictive criteria for successful gaits. In addition, the release of the hind leg knee occurs after the swing-leg foot has impacted with the ground.

The next section contains a brief overview of our modeling assumptions, impact conditions, impact modelling, and numerical results. In the third section it is shown how linear stability calculations were implemented. In the last section, a parameter variation is shown that ends in a design which is believed to be implementable in a 3D passive walker. The appendix contains the *Sophia*/Maple code used to derive the equations of motion and impact dynamics.

2 Modelling

2.1 GEOMETRY

The planar walker, see Figure 1, consists of four bodies constrained to move in a plane. The four bodies are the two identical upper legs and the two identical lower legs. The unconstrained mechanism has 6 degrees of freedom (DOF), labeled q_i , namely, the position of the hip, the two thigh angles and the two knee angles. Associated with each degree of freedom are the generalized velocities u_i . The generalized coordinates are related to the generalized velocities through the relation $\dot{q}_i = u_i$. All bodies are connected via frictionless hinge joints. The knee joints are at point P_3 and P_4 and the hip joint is at P_5 . The knee joint angles are constrained to positive values.

The centers of mass are specified with coordinates along and perpendicular to the lines $\overline{P_5P_4}, \overline{P_5P_3}$ for the thighs and $\overline{P_4P_1}, \overline{P_3P_2}$ for the lower legs. The moments of inertia I_i are given relative to the respective bodies centers of mass. The gravity vector is inclined an angle α as shown in the figure. Thus, the gravity vector has a positive component in the forward direction $\mathbf{n}_1$, which will propel the walker forward.

The walker makes regular contact with the ground plane at point C_1 and C_2 via its circular feet. The circular feet have a radius r_f and a center with an offset l_f from the points P_1 and P_2 . The contact points C_1 and C_2 are situated $-r_f \mathbf{n}_2$ from the foot center positions.

2.2 IMPACT MODELLING AND INTEGRATION

Impacts occurring in the system are treated as perfectly inelastic collisions. Thus, the relative velocity between the impacting bodies, at the impact point, is zero after the impact has occurred. Generalized impulses are introduced at

Planar Passive Walking

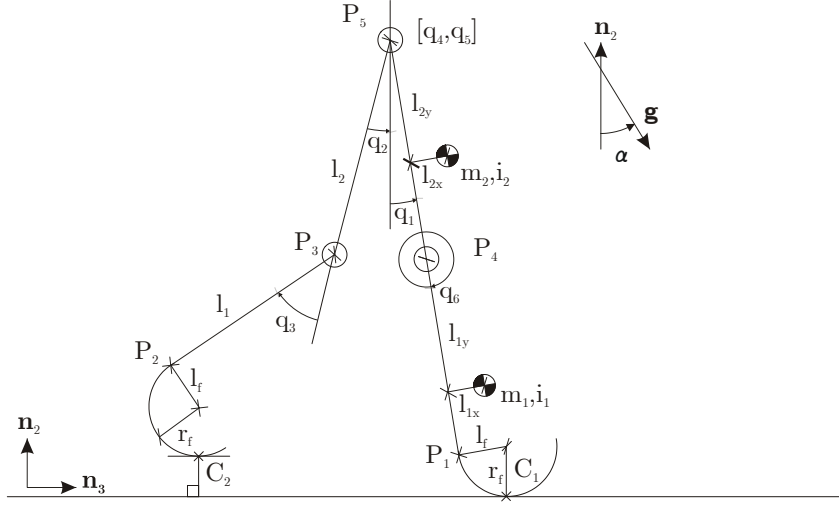


Figure 1. The geometry of the planar walker.

the instant of impact to bring the relative velocity to zero. At knee locking, an angular impulse is introduced at the knee joint. At foot impact, vertical and horizontal impulses from the plane are used at the initial contact point.

During an ordinary gait cycle, the planar walker goes through two continuous states, as illustrated in Figure 2. The first state is the three-link mode, which is ended at knee lock and the walker is transferred into the two-link mode. This mode ends at foot impact and the walker is returned to the three-link mode. At the end of each discrete state, some of the state variables make a discontinuous jump in their values.

In the following paragraphs, a detailed description will be given of the simulation and the impact modeling used in each step. In addition, the necessary checks to make sure that the solution is physically realizable are given and discussed.

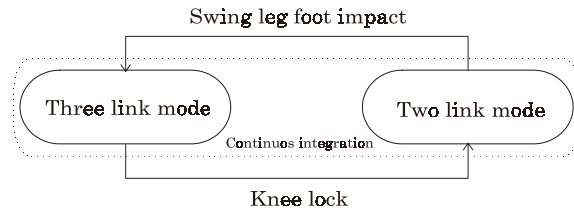


Figure 2. The walker goes through two continuous states during an ordinary gait cycles. These two states are connected via impact models.

2.3 TRANSITION FROM TWO-LINK MODE TO THREE-LINK MODE

The calculations at each impact are simplified by considering the mechanism as having all its six DOF available to it and introducing the appropriate impulses and constraints at each impact. As it turns out, the impact at knee lock can then be described by the same constraint and impulse equations as the impact of the swing-leg foot. However, as can be seen in figure 3 the geometrical configuration is different in the two cases.

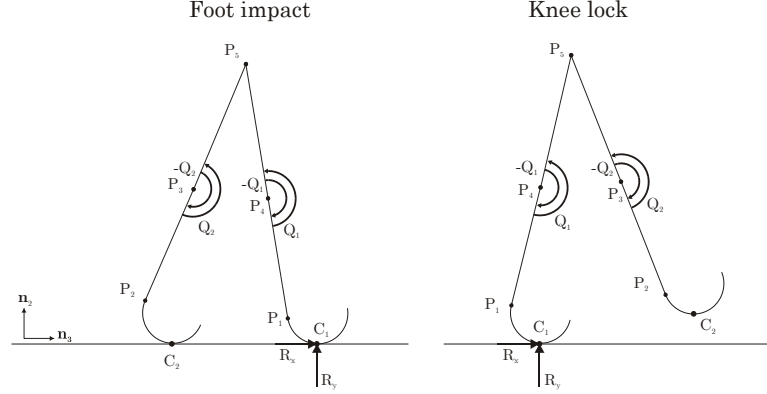


Figure 3. Impulses at foot impact and knee impact. Note that it is only the geometrical configuration that differentiate the two impacts.

Since the walker has different DOF, depending on its state, it is wise to give the initial conditions in the configuration having the least DOF. Immediately before foot impact, the state of the walker can be described by one of the leg angles and two angular velocities. Since the impacting-leg knee (soon to be the new stance-leg knee) is locked, only u_2 is needed to describe the stance leg angular velocity. The current stance-leg knee (soon to be new swing-leg knee) is also locked at this instant and therefore only u_1 is needed to describe the swing-leg angular velocity. Since the walker is invariant to translations in the forward direction, q_4 can be chosen at any value, preferably zero. From these four state variables $[q_1, q_4, u_1, u_2]$ the remaining ones are given by,

$$\begin{aligned} q_2 &= -2 \arctan((l_1 + l_2) / l_f) - q_1, \\ q_5 &= r_f - l_f \sin(q_1) + (l_1 + l_2) \cos(q_1), \\ u_4 &= u_1(-(l_1 + l_2) \cos(q_1) + l_f \sin(q_1) - r_f), \\ u_5 &= u_1(-l_f \cos(q_1) - (l_1 + l_2) \sin(q_1)), \\ q_3 &= q_6 = u_3 = 0, \end{aligned} \tag{1}$$

where the velocities of u_4 and u_5 are given by the rolling constraint of the stance leg (point C_2). The configuration of the walker is now given at the foot plane pre-collision instance. We now need to calculate the velocity jump

due to the impulses occurring at the foot impact. Since we assume that the mechanism is free at each impact, additional angular impulses are needed to maintain the locked state of the knees. Calculating these constraint impulses gives a possibility to check whether the constraints will be kept or broken. The equations that calculate the velocity jump can be stated as a matrix equation,

$$M(q)\Delta v = C(q)r \quad (2)$$

where $M(q)$ is the reduced mass matrix of the system, which is given in the appendix, and $\Delta v = [v^+ - v^-]$ is the resulting velocity jump due to the impulses r . The constraint matrix $C(q)$ relates the impulses at the point C_1 and the rotational impulses at the knees to the generalized velocities. This results in six equations for the ten unknowns (6 velocities and 4 impulses). The remaining 4 equations come from the velocity constraints, namely,

$$\begin{aligned} u_6 &= 0, u_3 = 0, \\ {}^N d/dt(\overline{OC_1}) &= \overline{0}, \end{aligned} \quad (3)$$

where the raised N indicates that the derivative is calculated in the inertial reference frame N . These velocity constraints are linear in the generalized speeds and expressing the zero velocity contact point constraint in the same directions as the impulses one can write the velocity constraint as,

$$C(q)^T v^+ = \overline{0} \quad (4)$$

where v^+ is the generalized velocity after the impact. Thus, if put on a matrix form, we can simultaneously solve for the new velocities v^+ and the impulses r , given the pre-collision velocity v^- . The matrix equation is given by,

$$\begin{bmatrix} M(q) & C(q) \\ C(q)^T & 0 \end{bmatrix} \begin{bmatrix} v^+ \\ r \end{bmatrix} = \begin{bmatrix} -M(q)v^- \\ 0 \end{bmatrix} \quad (5)$$

where $C(q_i)$ has 6x4 components and its non-zero elements equals,

$$\begin{aligned} C_{11} &= l_2 \cos(q_1) + l_f \sin(q_6 - q_1) + l_1 \cos(q_6 - q_1) + r_f, \\ C_{12} &= l_2 \sin(q_1) + l_f \cos(q_6 - q_1) - l_1 \sin(q_6 - q_1), \\ C_{61} &= -l_f \sin(q_6 - q_1) - l_1 \cos(q_6 - q_1) - r_f \\ C_{62} &= -l_f \cos(q_6 - q_1) + l_1 \sin(q_6 - q_1), \\ C_{34} &= C_{63} = C_{41} = C_{52} = 1 \end{aligned} \quad (6)$$

and

$$r = [R_x, R_y, Q_1, Q_2]^T, v = [u_1, \dots, u_6]^T. \quad (7)$$

2.4 POST COLLISION CHECKS

To ensure that the impact solutions are physically realizable, post collision checks are required. After the foot collision, the vertical velocity of the contact point of the new swing leg has to be positive,

$$\begin{aligned} {}^N d/dt(\overline{OC_2}) \cdot \mathbf{n}_2 &= l_f(u_2 \cos(q_2) - u_1 \cos(q_1)) + \\ &(l_1 + l_2)(u_2 \sin(q_2) - u_1 \sin(q_1)) > 0 \end{aligned} \quad (8)$$

Additionally, if simple mechanical stops are used, both rotational impulses have to be positive or the knees will collapse. However, in the simulations, only negative angular impulses at the hind-leg knee were found (point $P3$ in the left pane of Figure 3). Therefore, the knee is assumed locked until immediately after the foot collision has occurred. This implies a zero angular velocity of u_3 at the start of the swing phase. The final check that is made is to ensure that the acceleration of the new swing leg is positive, i.e. the new swing-leg foot should move away from the plane.

2.5 THREE-LINK MODE INTEGRATION

The constraints applied to the free mechanism during the three-link phase consists of the stance leg knee being locked, which can be written as,

$$q_6 = u_6 = 0 \quad (9)$$

and that the stance leg foot should be rolling. The rolling condition implies that the point C_1 should have a zero vertical position and that the horizontal velocity of the current contact point in the foot is zero. Since the position constraint is holonomic, it can be expressed as,

$$q_5 = -l_f \sin(q_1) + (l_1 + l_2) \cos(q_1) + r_f \quad (10)$$

The rolling without slipping constraint is,

$$u_4 = u_1(-(l_1 + l_2) \cos(q_1) + l_f \sin(q_1) - r_f) \quad (11)$$

which can be integrated to a holonomic constraint,

$$q_4 = -(l_1 + l_2) \sin(q_1) - l_f \cos(q_1) - r_f * q_1 + \text{const.} \quad (12)$$

where *const.* is an integration constant, which can be determined at the start of the integration. The system to integrate now consists of the states $[q_1, q_2, q_3, u_1, u_2, u_3]$, which is integrated until one of the below conditions occur.

2.6 CHECKS DURING THREE-LINK INTEGRATION

In this state, four different events, which terminate the gait simulation are monitored. Namely, that heel strike shouldn't occur before the knee locks. Heel strike occur when

$$l_f \sin(-q_3 + q_2) - l_2 \cos(q_2) - l_1 \cos(-q_3 + q_2) - l_f \sin(q_1) + (l_1 + l_2) \cos(q_1) = 0 \quad (13)$$

The walker shouldn't fall back, which occurs when the hip (point $P5$) hit the floor. This condition can be expressed as,

$$r_f - l_f \sin(q_1) + (l_1 + l_2) \cos(q_1) = 0 \quad (14)$$

In addition, to ensure that an over swing of the lower leg doesn't occur, q_3 has to be less than π . The correct termination of this integration is when knee lock

occur, $q_3 = 0$. After the integration has terminated, the states $[q_4, q_5, u_4, u_5]$ can be calculated using equations (10-12)

To be on the safe side, one could also check that the vertical component of the constraint force between the plane and rolling foot is always positive. However, all observed gaits have the velocity of the hip below $\sqrt{gl}$, which is the required velocity to leave the ground (compare the walker to an inverted pendulum), therefore, this was not checked during simulations.

2.7 TRANSITION FROM THREE-LINK MODE TO TWO-LINK MODE

As discussed earlier, the same equations as in the foot impact calculations can now be used again. However, as seen in the right pane of Figure 3, the geometrical configuration is different.

2.8 POST COLLISION CHECKS

Since it is assumed that the stance leg foot is only constrained by the plane, it is free to move upwards. This implies that the impulse R_y has to be positive, or the foot will leave the ground plane. If Coulomb friction laws are valid for impacts, see Brach [3] for a discussion of this topic, one could also make sure that the absolute value of the ratio of horizontal impulse to vertical impulse is below the friction coefficient,

$$\left| \frac{R_x}{R_y} \right| < \mu. \quad (15)$$

We also have to check that $q_1 > q_2$ otherwise the knee lock occurred when the swing leg was behind the stance leg. Since both legs are assumed to have equal geometry, the swing cannot take place when both knees are locked. This will eventually result in foot scuffing.

2.9 TWO-LINK MODE INTEGRATION

During the two-link mode, both knees are locked, $q_3 = q_6 = 0$, and consequently both angular knee velocities, u_3 and u_6 , are equal to zero. In addition, the rolling constraint has to be satisfied, see equation (11). The system to integrate now consists of the state variables $[q_1, q_2, u_1, u_2]$, which is integrated until one of the below conditions occur.

2.10 CHECKS DURING TWO-LINK MODE INTEGRATION

Two events, which terminate the gait simulation, might occur in this mode. The first one occurs if the hip hits the floor, see equation (14). The second one occurs if the swing leg rotates back to a configuration where $q_1 = q_2$. The termination that continue the gait cycle takes place when the swing-leg foot impacts with the ground and the legs have an opening angle. This occurs when,

$$q_1 + q_2 + 2 \arctan((l_1 + l_2) / l_f) = 0 \quad (16)$$

The additional states $[q_4, q_5, u_4, u_5]$ can now be calculated from equations (10-12). The walker is now in the pre-foot-collision state and the transition from two-link mode to three-link mode can take place again.

3 Simulation

Simulation of the planar walker was performed in Matlab using the ode45 integrator (Runge-Kutta 45). This integrator is suitable for solving non-stiff ordinary differential equations. All integrators in Matlab support event detection, which were used to detect the end of the two continuous integrations. Both the absolute and relative tolerance was set to $1e-6$.

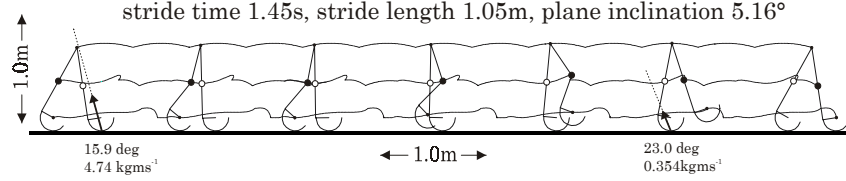


Figure 4. Simulation of a planar walker during 7 consecutive steps. Parameters and initial conditions can be found in Table I.

Table I. Parameter values and initial conditions used in the simulation found in Figure 4. Immediately before the double support only three state variables are needed to specify the complete state of the walker. The leg angle q_1 (the opening angle, which equals $q_1 - q_2$, can be calculated given q_1 and the double support configuration), the angular velocity of the new stance leg $\dot{q}_1$, and the angular velocity of the new swing leg $\dot{q}_2$.

Param	Value	Param	Value	Param	Value
m_1	1.013 kg	m_2	2.35 kg	l_f	0.123 m
i_1	0.0393 kg m ²	i_2	$0.231 \cdot 10^{-3}$ kg m ²	r_f	0.12 m
l_{1x}	0.0432 m	l_{2x}	0.0 m	g	9.81 m s ⁻²
l_{1y}	0.1663 m	l_{2y}	0.091 m	α	0.09 rad
l_1	0.3132 m	l_2	0.35 m		

State variable	q_1	$\dot{q}_1$	$\dot{q}_2$
Initial value	-0.5196 rad	-1.933 rad s ⁻¹	-2.037 rad s ⁻¹

Parameter values and initial conditions were obtained from previous work of Ruina and Collaborators, see [5]. However, the initial conditions had to be manually tuned before a stable gait was observed in the simulation. This is probably due to some modeling differences between our planar walker and the aforementioned walker. Stability was observed by plotting the three needed state variables $[q_1, u_1, u_2]$ in a three-dimensional diagram at subsequent steps. A stable gait is observed to converge to a point.

In Figure 4 a simulation of a stable planar walker is shown. The walker is drawn at different geometrical configurations during 7 consecutive steps (a stride consists of two steps, thus the walker is back to its original configuration

after two steps or one stride). The impulses from the plane are shown at foot impact and at knee locking. The impulse angle at foot impact requires a friction coefficient of $\mu = 0.28$ and at knee impact $\mu = 0.42$ or the walker will slip. The parameters for this particular walker can be found in Table I.

4 Stability

The stability of the gait was calculated by perturbing the stable periodic solution x_0 in each state variable. The accuracy of the periodic orbit was obtained to 1e-9 (relative and absolute) and the perturbations ε are of size 5e-6. By perturbing one state variable at the time and subtracting the periodic solution from the resulting state after one step, the Jacobian J of the system can be approximated, as indicated below,

$$\begin{aligned} x_0 + [\varepsilon, 0, 0]^T &\rightarrow \text{integration} \rightarrow x_1 \\ x_0 + [0, \varepsilon, 0]^T &\rightarrow \text{integration} \rightarrow x_2 \\ x_0 + [0, 0, \varepsilon]^T &\rightarrow \text{integration} \rightarrow x_3 \end{aligned}$$

$$J \simeq \frac{1}{\varepsilon} \begin{bmatrix} \vdots & \vdots & \vdots \\ x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Stability is now given by calculating the eigenvalues of this Jacobian. As long as all eigenvalues are inside the unit circle, the periodic solution is stable. Having access to the Jacobian also makes it possible to implement root-finding algorithms for locating periodic solutions, both stable and unstable. This approach was used to locate periodic gaits of the passive 3D walker, see Adolfsson [1] and [2]. However, in this chapter we only study periodic gaits, which are accessible through direct numerical simulation, i.e. stable periodic solutions. In Table II, the eigenvalues and eigenvectors of the periodic solution found in Table I is given

Table II. Eigenvalues and eigenvectors corresponding to the periodic solution given by the data in Table I.

	Eig.Vec.1	Eig.Vec.2	Eig.Vec.3
q_1	-0.0343	0.0047	0.0017
u_1	-0.0433	-0.0725	0.1088
u_2	0.9985	-0.9974	0.9941
Eig.Val	-0.7034	0.0088	0.1060

4.1 PARAMETER STUDY

Many interesting parameter studies have been done on the planar walker. For example, the inclination of the plane has been extensively studied by both McGeer, see [10],[11],[12], and Garcia, see [7] and [6]. By tuning the mass properties of the walker, Garcia [5], found periodic solutions for arbitrarily small slopes.

Since the goal is to extend the planar walker into three dimensions, a foot design that is suitable for this is needed. The approach taken here is to reduce the foot radius while maintaining an offset for the foot radius. In Figure 5, the

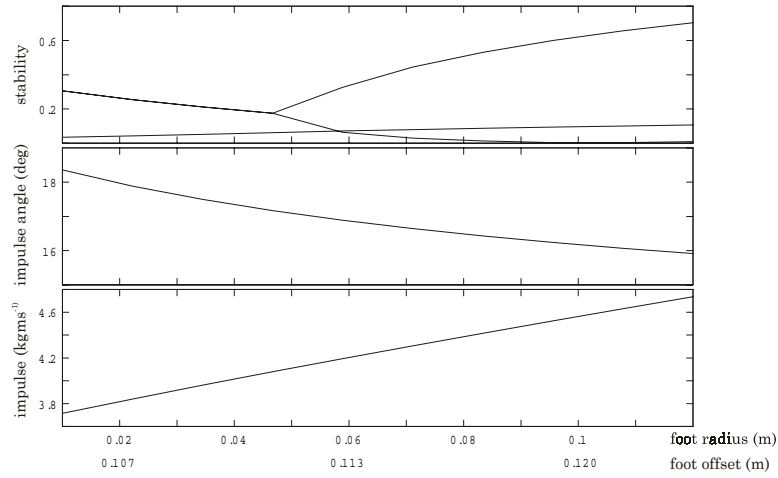


Figure 5. Parameter variation showing the stability of the planar walker as the foot is shrunk. Also shown, is the needed impulse and impulse angle at the foot impact.

foot radius is shrunk from 12cm (corresponding to the walker found in Figure 4) to 1cm (corresponding to the walker found in Figure 6). To remain stable, it was noted that the offset l_f had to be decreased slightly, from 12.3cm to 10.5cm. In figure 5 the stability of this particular parameter variation is shown, along with the needed impulse at foot impact. Stability is plotted as the absolute value of the eigenvalues of the Jacobian. As seen in the figure, it is stable over the whole parameter interval, with maximum stability at $f_r \approx 0.05\text{m}$. The y-shaped crossing at maximum stability is due to a complex-conjugate pair of eigenvalues turning into two real eigenvalues. The impulse from the plane, at foot impact, shows no dramatic change over the interval. Thus, it is expected that if slip does not occur in the large radius foot it will not occur in the small radius foot either.

5 Conclusions

In Figure 6, the small foot walker is drawn at different geometrical configurations for seven consecutive steps. The trajectories are quite similar to the ones found for the large radius foot, however, a somewhat smoother motion of the foot radius center can be observed. A difference in stride time and stride length can also be observed, which is probably due to the change of the length ($l_{1y} + r_f$) of the lower leg.

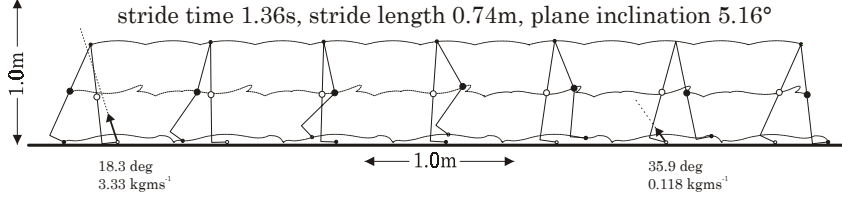


Figure 6. Simulation of a planar walker having small feet during 7 consecutive steps.

Although not shown here, it was possible to reduce the foot radius to arbitrarily small values. It was even, although not realizable, possible to use negative values of the foot radius. This suggests that the shape of the foot can be chosen quite freely, as long as the foot design allows for the inverted-pendulum motion to take place.

An advantage of this foot design is that it is easier to implement compared to the previous used large radius foot. In addition, it is now possible to extend this construction into three dimensions while guaranteeing the existence of periodic solutions, see Piironen [14]. In Figure 7, the path from planar passive walker into a 3D passive walking is shown. The point toe can now be extended into a line foot in three dimensions. By starting the three dimensional passive walker with zero hip distance and symmetrically overlapping feet, the periodic solution will exist. However, the stability characteristics are not known since additional degrees of freedom will be present.

Some comparisons between the planar passive walker and three-dimensional passive walker can be found in Dankowicz *et al.*, see [4].

6 Appendix

6.1 MAPLE AND SOPHIA CODE

This is the **Sophia**/Maple code for deriving the equations of motion and the impact equations for the planar passive walker. The **Sophia** package is a useful tool for analysis of multi-body dynamics, vector algebra and calculus, and has application to systems of ordinary differential equations. **Sophia** is implemented in Maple and has export routines to Matlab, (see Lesser [9] and

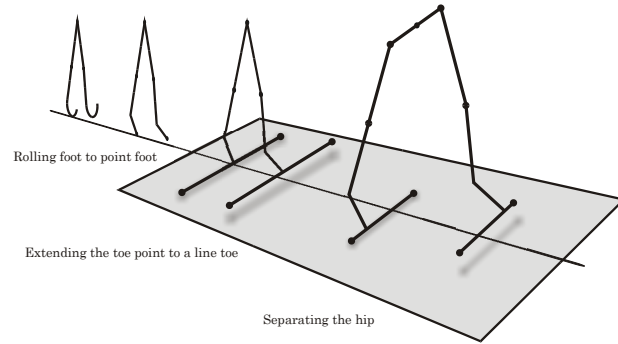


Figure 7. Illustrative picture showing the transition from a planar passive walker into a fully 3D passive walker having dynamics in all spatial directions.

Lennartsson [8]). **Sophia** uses Lesser's geometrical interpretation of Kane's method.

Maple code will be written with this **typeface** and comments with the ordinary typeface. Sophia can be downloaded at www.mech.kth.se/sophia and the code for deriving the equations of motion can be found at www.mech.kth.se/sophia/passivewalking

The Maple session is started by resetting the workspace and then reading the Sophia library.

```
> restart;
> read 'sophiaV5';
```

6.2 GEOMETRY

We start by defining the frame relations where N is the inertial reference frame.

Upper leg 1, reference frame A

```
> &rot[N,A,3,q1]:
```

Upper leg 2, reference frame B

```
> &rot[N,B,3,q2]:
```

Lower leg 2, reference frame CC

```
> &rot[B,CC,3,-q3]:
```

Lower leg 1, reference frame DD

```
> &rot[A,DD,3,-q6]:
```

The geometry of the walker is now defined and the centers of mass are labeled, refer to Figure 1.

Hip (point P_5)

```
j > P5 := N &ev [q4,q5,0]:
```

Planar Passive Walking

Upper leg 1 and leg 2 center of mass

```
> r1 := P5 &++ (A &ev [12x,-12y,0]):
> r2 := P5 &++ (B &ev [12x,-12y,0]):
```

Position of the knees, leg 2 & 1 (point P_3 and P_4)

```
> P3 := P5 &++ (B &ev [0,-12,0]):
> P4 := P5 &++ (A &ev [0,-12,0]):
```

Lower leg 2 and lower leg 1 center of mass

```
> r3 := P3 &++ (CC &ev [11x,-11y,0]):
> r4 := P4 &++ (DD &ev [11x,-11y,0]):
```

Circle foot contact point for leg 1 and 2 (point C_1 and C_2)

```
> C1 := P4 &++ (DD &ev [1f,-11,0]) &++ (N &ev [0,-rf,0]):
> C2 := P3 &++ (CC &ev [1f,-11,0]) &++ (N &ev [0,-rf,0]):
```

6.3 KINEMATICS

Compute the required center of mass velocities and angular velocities for each body.

Upper leg 1

```
> v1 := &simp subs(kde6,N &fdt r1):
> w1 := subs(kde6,N &aV A):
```

Upper leg 2

```
> v2 := &simp subs(kde6,N &fdt r2):
> w2 := subs(kde6,N &aV B):
```

Lower leg 2

```
> v3 := &simp subs(kde6,N &fdt r3):
> w3 := subs(kde6,N &aV CC):
```

Lower leg 1

```
> v4 := &simp subs(kde6,N &fdt r4):
> w4 := subs(kde6,N &aV DD):
```

Use simple kinematical differential equations, i.e. $\dot{q}_i = u_i$.

```
> &kde(6);kde6:=kde:
```

Assemble a K-vector of velocities

```
> vK := &KM[v1,w1,v2,w2,v3,w3,v4,w4]:
```

Momentum and angular momentum for each body

Upper leg 1

```
> p1 := m2 &** v1:
> h1 := i2 &** w1:
```

Upper leg 2

```
> p2 := m2 &** v2:
> h2 := i2 &** w2:
```

Lower leg 2

```
> p3 := m1 &** v3:
> h3 := i1 &** w3:
```

Lower leg 1

```
> p4 := m1 &** v4:
> h4 := i1 &** w4:
```

Assemble a momentum K-vector

```
> pK := &KM[p1,h1,p2,h2,p3,h3,p4,h4]:
```

Differentiate each component of the momentum K-vector with respect to time to obtain the inertia forces and substitute the kde.

```
> pKt := &Ksimp subs(kde6, N &Kfdt pK):
```

6.4 FORCES AND TORQUES

The applied forces and torques are now given. Since gravity is the only applied force on the walker, all torques will be zero.

Upper leg 1

```
> Ra1 := N &ev [m2*g*sin(alpha),-m2*g*cos(alpha),0]:
> Ta1 := N &ev [0,0,0]:
```

Upper leg 2

```
> Ra2 := N &ev [m2*g*sin(alpha),-m2*g*cos(alpha),0]:
> Ta2 := B &ev [0,0,0]:
```

Lower leg 2

```
> Ra3 := N &ev [m1*g*sin(alpha),-m1*g*cos(alpha),0]:
> Ta3 := CC &ev [0,0,0]:
```

Lower leg 1

```
> Ra4 := N &ev [m1*g*sin(alpha),-m1*g*cos(alpha),0]:
> Ta4 := DD &ev [0,0,0]:
```

Assemble applied forces and torques in a K-vector

```
> RK := &KM[Ra1,Ta1,Ra2,Ta2,Ra3,Ta3,Ra4,Ta4]:
```

6.5 KANE'S EQUATIONS

Compute the tangent vectors. Since the tangent vectors are assumed "orthogonal" to the constraint forces, they can be used to project away the constraint forces.

```
> tauK := KMtangents(vK,u,6):
```

Project the tangent K-vectors on the applied forces K-vector and the inertia forces K-vector

```
> GAF := tauK &kane RK:
> MGIF := tauK &kane pKt:
```

Extract Kane's equations from the Generalized Applied Forces (GAF) and the Minus Generalized Inertia Forces (MGIF)

```
> kaneeq := [seq(MGIF[i]-GAF[i],i=1..6)]:
> kaneeq := map(simplify,kaneeq):
```

Planar Passive Walking

```
> kaneeq := map(combine,kaneeq,trig):
```

Read the linalg library.

```
> with(linalg):
```

Put the equations on the form $M\dot{u} = F$ where M is the reduced mass matrix $\dot{u}$ is the vector of generalized accelerations, and F is the generalized force vector.

```
> TOT:=genmatrix(kaneeq,[u1t,u2t,u3t,u4t,u5t,u6t],flag):
> M:=submatrix(TOT,1..6,1..6):
> F:=submatrix(TOT,1..6,7..7):
```

M is the reduced mass matrix, which is also used in impact calculations.

6.6 THREE-LINK MODE

We now derive the equations of motion for the planar passive walker with the stance leg knee locked and the stance leg foot rolling on the plane. The constraints are $q_6 = 0$ (knee angle), the vertical position of the contact point C_1 must be zero, and the horizontal velocity of the contact point C_1 must also be zero. The reduced state of the walker will, in three-link mode, consist of $[q_1, q_2, q_3, q_4, u_1, u_2, u_3]$, thus, $[q_5, q_6, u_4, u_5, u_6]$ can be removed by the constraints.

Generalized coordinate constraints solved for q_5 (vertical position of hip) and q_6 (stance leg knee angle).

```
> qconstr3:=solve({(N &to C1) &c 2 = 0, q6=0},{q5,q6}):
```

The velocity of the instantaneous contact point C_1 is $(^N d/dt \overline{O P_4}) + \omega_4 \times \overline{P_4 C_1}$.

```
> v_contact1 := &simp (N &to (&simp subs(kde6,N &fdt P4)
&++ (w4 &xx (C1 &-- P4)))):
```

Differentiate the generalized coordinate constraint with respect to time

```
> qtconstr3:=map(diffTime,qconstr3):
```

Combine all generalized velocity constraints and express them as relations between the generalized velocities

```
> uconstr3:=simplify(subs(qconstr3,solve(subs(kde6,
qtconstr3) union {v_contact1 &c 1 = 0},{u4,u5,u6})))):
```

Compute the generalized acceleration constraints

```
> aconstr3:=subs(uconstr3,qconstr3,subs(kde6,
map(diffTime,uconstr3))):
```

Kinematical differential equations for three-link mode

```
> ki := subs(kde6,uconstr3,qconstr3,[q1t,q2t,q3t,q4t]);
> kde3 := {seq(q.i.t=ki[i],i=1..4)};
```

Velocity K-vector expressed with reduced state

```
> vK3 := &Ksimp subs(uconstr3,qconstr3,vK):
```

Compute the tangent K-vectors

```
> tauK3 := KMtangents(vK3,u,3):
```

Project the new tangent K-vectors on the applied forces and inertial forces

```
> GAF3 := tauK3 &kane RK:
> MGIF3 := tauK3 &kane pKt:
```

Substitute for the removed states $[q_5, q_6, u_4, u_5, u_6]$ and their derivatives. Extract Kane's equations from the Generalized Applied Forces (GAF) and the Minus Generalized Inertia Forces (MGIF)

```
> ndof := 3:
> kaneeq3 := simplify(subs(aconstr3,uconstr3,
qconstr3,[seq(MGIF3[i]-GAF3[i],i=1..ndof)])):
> kaneeq3 := map(simplify,kaneeq3):
> kaneeq3 := map(combine,kaneeq3,trig):
```

Put the equations on the form $M\dot{u} = F$ where M is the reduced mass matrix, $\dot{u}$ is the vector of generalized accelerations, and F is the generalized force vector.

```
> TOT3 := genmatrix(kaneeq3,[u1t,u2t,u3t],flag):
> M3 := submatrix(TOT3,1..3,1..3):
> F3 := submatrix(TOT3,1..3,4..4):
```

6.7 TWO-LINK MODE

In addition to the three-link mode constraints, the swing leg knee is now locked, $q_3 = 0$ (knee angle). The reduced state of the walker will, in two-link mode, consist of $[q_1, q_2, q_3, q_4, u_1, u_2]$.

```
> qconstr2 := qconstr3 union {q3=0}:
```

Differentiate the generalized coordinate constraint

```
> qtconstr2 := map(diffTime,qconstr2):
```

Generalized velocity constraint

```
> uconstr2 := uconstr3 union {u3=0}:
```

Generalized acceleration constraints

```
> aconstr2 := aconstr3 union {u3t=0}:
```

Velocity K-vector expressed with reduced state

```
> vK2 := &Ksimp subs(uconstr2,qconstr2,vK):
```

Kinematic differential equations for two-link mode

```
> ki:=subs(kde6,uconstr2,qconstr2,[q1t,q2t,q4t]):
> kde2:=[q1t=ki[1],q2t=ki[2],q4t=ki[3]]:
```

Compute the tangent K-vectors

```
> tauK2 := KMtangents(vK2,u,2):
```

Project the new tangent K-vectors on the applied forces and inertial forces

```
> GAF2 := tauK2 &kane RK:
> MGIF2 := tauK2 &kane pKt:
```

Substitute for the removed states $[q_5, q_6, u_3, u_4, u_5, u_6]$ and their derivatives. Extract Kane's equations from the Generalized Applied Forces (GAF) and the Minus Generalized Inertia Forces (MGIF)

```
> ndof := 2:
> kaneeq2 := simplify(subs(aconstr2,uconstr2,
```


Planar Passive Walking

```
qconstr2,[seq(MGIF2[i]-GAF2[i],i=1..ndof]))):
```

Put the equations on the form $M\dot{u} = F$ where M is the reduced mass matrix, $\dot{u}$ is the vector of generalized accelerations, and F is the generalized force vector

```
> TOT2 := genmatrix(kaneeq2,[u1t,u2t],flag):
> M2 := submatrix(TOT2,1..2,1..2):
> F2 := submatrix(TOT2,1..2,3..3):
```

6.8 IMPACT EQUATIONS

Both knees are assumed locked and therefore the constraint knee impulses (Q_1, Q_2) are introduced as to maintain the locked state of the knee. We want the equations on this form,

$$\begin{bmatrix} M & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} v^+ \\ r \end{bmatrix} = \begin{bmatrix} -Mv^- \\ 0 \end{bmatrix},$$

where C^T is the transpose of C , v^+ is the generalized velocity vector after impact and v^- is the generalized velocity vector before impact.

The impulse from the plane, at point C_1 , on the impacting foot (Lower leg 1) and the knee angular impulse is,

```
> iRa4 := N &ev [R1,R2,0]:
> iTa4 := &simp ((C1 &-- r4) &xx iRa4 &++
(N &ev [0,0,-Q1])):
```

Upper leg 2

```
> iRa2 := N &ev [0,0,0]:
> iTa2 := B &ev [0,0,Q2]:
```

Lower leg 2

```
> iRa3 := N &ev [0,0,0]:
> iTa3 := N &ev [0,0,-Q2]:
```

Upper leg 1

```
> iRa1 := N &ev [0,0,0]:
> iTa1 := DD &ev [0,0,Q1]:
```

Assemble the applied impulses and angular impulses in a K-vector.

```
> iRK:=&KM[iRa1,iTa1,iRa2,iTa2,iRa3,iTa3,iRa4,iTa4]:
```

Compute the generalized impulses

```
> genimp := map(combine,map(simplify,tauK &kane iRK),trig):
```

The generalized impulses are linear in the applied impulses and therefore it can be written as Cr , where $r = [R_1, R_2, Q_1, Q_2]^T$

```
> C := genmatrix(genimp,[R1,R2,Q1,Q2]):
```

Velocity constraints after impact are obtained from the condition that contact point C_1 should have zero velocity

```
> v_contact1 := &simp (N &to (&simp subs(kde6,N &fddt P4) &++
(w4 &xx (C1 &-- P4)))):
```

This constraint is linear in the generalized velocities

```
> CT:=map(combine,genmatrix([v_contact1 &c 1,
v_contact1 &c 2,u6=0,u3=0],[u1,u2,u3,u4,u5,u6]),trig);
```

Since the applied impulses are parallel to the directions of the velocity constraints and applied at the same point C_1 , the transposed C should equal CT .

6.9 MASS MATRIX AND GENERALIZED FORCES

Only non-zero elements of the diagonal and upper right part of the mass matrix is shown,

$$\begin{aligned}
M_{11} &= (l_{2y}^2 + l_{2x}^2)m_2 + (l_{1y}^2 + l_{1x}^2 + l_2^2 + 2l_2l_{1y}\cos(q_6) + 2l_{1x}l_2\sin(q_6))m_1 + i_1 + i_2 \\
M_{14} &= (l_{2y}\cos(q_1) - l_{2x}\sin(q_1))m_2 + (l_{1x}\sin(-q_1 + q_6) + l_{1y}\cos(-q_1 + q_6) + l_2\cos(q_1))m_1 \\
M_{15} &= (l_{2y}\sin(q_1) + l_{2x}\cos(q_1))m_2 + (l_{1x}\cos(-q_1 + q_6) + l_2\sin(q_1) - l_{1y}\sin(-q_1 + q_6))m_1 \\
M_{16} &= (-l_{1x}l_2\sin(q_6) - l_{1x}^2 - \cos(q_6)l_2l_{1y} - l_{1y}^2)m_1 - i_1 \\
M_{22} &= (l_{2y}^2 + l_{2x}^2)m_2 + (l_2^2 + l_{1x}^2 + l_{1y}^2 + 2l_2l_{1x}\sin(q_3) + 2l_2l_{1y}\cos(q_3))m_1 + i_1 + i_2 \\
M_{23} &= (-l_{1x}^2 - l_{1y}^2 - l_2l_{1y}\cos(q_3) - l_2l_{1x}\sin(q_3))m_1 - i_1 \\
M_{24} &= (l_{2y}\cos(q_2) - l_{2x}\sin(q_2))m_2 + (l_2\cos(q_2) + l_{1y}\cos(-q_2 + q_3) + l_{1x}\sin(-q_2 + q_3))m_1 \\
M_{25} &= (l_{2y}\sin(q_2) + l_{2x}\cos(q_2))m_2 + (-l_{1y}\sin(-q_2 + q_3) + l_2\sin(q_2) + l_{1x}\cos(-q_2 + q_3))m_1 \\
M_{33} &= (l_{1y}^2 + l_{1x}^2)m_1 + i_1 \\
M_{34} &= (-l_{1x}\sin(-q_2 + q_3) - l_{1y}\cos(-q_2 + q_3))m_1 \\
M_{35} &= (-l_{1x}\cos(-q_2 + q_3) + l_{1y}\sin(-q_2 + q_3))m_1 \\
M_{44} &= 2m_1 + 2m_2 \\
M_{46} &= (-l_{1y}\cos(-q_1 + q_6) - l_{1x}\sin(-q_1 + q_6))m_1 \\
M_{55} &= 2m_1 + 2m_2 \\
M_{56} &= (l_{1y}\sin(-q_1 + q_6) - l_{1x}\cos(-q_1 + q_6))m_1 \\
M_{66} &= (l_{1y}^2 + l_{1x}^2)m_1 + i_1
\end{aligned}$$

The generalized forces equals,

$$\begin{aligned}
F_1 &= (l_{2y}\sin(\alpha - q_1) - l_{2x}\cos(\alpha - q_1))m_2g + (l_2\sin(\alpha - q_1) - l_{1x}\cos(q_6 + \alpha - q_1) + l_{1y}\sin(q_6 + \alpha - q_1))m_1g + (-l_2\sin(q_6)u_6^2l_{1y} + 2l_2\sin(q_6)u_6u_1l_{1y} - 2\cos(q_6)l_2u_1u_6l_{1x} + \cos(q_6)l_2u_6^2l_{1x})m_1 \\
F_2 &= (-l_{2x}\cos(\alpha - q_2) + l_{2y}\sin(\alpha - q_2))gm_2 +
\end{aligned}$$

Planar Passive Walking

$$\begin{aligned}
& (-l_{1x} \cos(q_3 + \alpha - q_2) + l_2 \sin(\alpha - q_2) + \\
& l_{1y} \sin(q_3 + \alpha - q_2)) m_1 g + \\
& (-2 \cos(q_3) l_2 u_2 u_3 l_{1x} + \cos(q_3) l_2 u_3^2 l_{1x} - \\
& l_2 \sin(q_3) u_3^2 l_{1y} 2 l_2 \sin(q_3) u_3 u_2 l_{1y}) m_1 \\
F_3 = & (-l_{1y} \sin(q_3 + \alpha - q_2) + l_{1x} \cos(q_3 + \alpha - q_2)) m_1 g + \\
& (l_{1x} \cos(q_3) u_2^2 l_2 - l_{1y} u_2^2 l_2 \sin(q_3)) m_1 \\
F_4 = & (\sin(q_1) u_1^2 l_{2y} + \cos(q_2) u_2^2 l_{2x} + 2g \sin(\alpha) + \\
& \sin(q_2) u_2^2 l_{2y} + \cos(q_1) u_1^2 l_{2x}) m_2 + 2m_1 g \sin(\alpha) + \\
& (u_1^2 l_{1x} \cos(-q_1 + q_6) - u_1^2 l_{1y} \sin(-q_1 + q_6) - \\
& u_6^2 l_{1y} \sin(-q_1 + q_6) - u_2^2 l_{1y} \sin(-q_2 + q_3) + \\
& u_2^2 l_{1x} \cos(-q_2 + q_3) + \sin(q_1) u_1^2 l_2 + \\
& 2u_3 u_2 l_{1y} \sin(-q_2 + q_3) + \\
& 2u_6 u_1 l_{1y} \sin(-q_1 + q_6) - \\
& u_3^2 l_{1y} \sin(-q_2 + q_3) - 2u_2 u_3 l_{1x} \cos(-q_2 + q_3) - \\
& 2u_1 u_6 l_{1x} \cos(-q_1 + q_6) + u_3^2 l_{1x} \cos(-q_2 + q_3) + \\
& \sin(q_2) u_2^2 l_2 + u_6^2 l_{1x} \cos(-q_1 + q_6)) m_1 \\
\\
F_5 = & (\sin(q_2) u_2^2 l_{2x} - \cos(q_2) u_2^2 l_{2y} + \sin(q_1) u_1^2 l_{2x} - \\
& \cos(q_1) u_1^2 l_{2y} - 2g \cos(\alpha)) m_2 - 2m_1 g \cos(\alpha) + \\
& (2u_1 u_6 l_{1x} \sin(-q_1 + q_6) - u_6^2 l_{1x} \sin(-q_1 + q_6) - \\
& u_2^2 l_{1x} \sin(-q_2 + q_3) - u_2^2 l_{1y} \cos(-q_2 + q_3) + \\
& 2u_6 u_1 l_{1y} \cos(-q_1 + q_6) + 2u_2 u_3 l_{1x} \sin(-q_2 + q_3) - \\
& u_1^2 l_{1x} \sin(-q_1 + q_6) - u_1^2 l_{1y} \cos(-q_1 + q_6) - \\
& u_6^2 l_{1y} \cos(-q_1 + q_6) - u_3^2 l_{1x} \sin(-q_2 + q_3) - \\
& \cos(q_1) u_1^2 l_2 + 2u_3 u_2 l_{1y} \cos(-q_2 + q_3) - \\
& u_3^2 l_{1y} \cos(-q_2 + q_3) - \cos(q_2) u_2^2 l_2) m_1 \\
F_6 = & (-l_{1y} \sin(q_6 + \alpha - q_1) + l_{1x} \cos(q_6 + \alpha - q_1)) m_1 g + \\
& (-l_{1y} u_1^2 l_2 \sin(q_6) + l_{1x} \cos(q_6) u_1^2 l_2) m_1
\end{aligned}$$

Bibliography

- [1] Adolffsson, J.: 2000, ‘3D Passive walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [2] Adolffsson, J., H. Dankowicz, and A. Nordmark: 2001, ‘3D Passive Walkers: Finding Periodic Gaits in the Presence of Discontinuities’. *Nonlinear Dynamics* **24**(2), 205–229.

- [3] Brach, R. M.: 1991, *Mechanical Impact Dynamics*. A Wiley-Interscience Publication.
- [4] Dankowicz, H., J. Adolfsson, and A. Nordmark: 2001, 'Existence of Stable 3D-Gait in Passive Bipedal Mechanisms'. *Journal of Biomechanical Engineering* **123**(1), 40–46.
- [5] Garcia, M., A. Chatterjee, and A. Ruina: 1998a, 'Efficiency, Speed, and Scaling of Passive Dynamical Bipedal Walking'. *Dynamics and Stability of Systems*. submitted for publication.
- [6] Garcia, M., A. Chatterjee, A. Ruina, and M. Coleman: 1998b, 'Passive-dynamic Models of Human Gait'. In: *Proceedings of the Conference on Biomechanics and Neural Control of Human Movement*. pp. 32–33.
- [7] Garcia, M., A. Chatterjee, A. Ruina, and M. Coleman: 1998c, 'The Simplest Walking Model: Stability, Complexity, and Scaling'. *ASME Journal of Biomechanical Engineering* **120**, 281–288.
- [8] Lennartsson, A.: 1999, 'Efficient Multibody Dynamics'. Ph.D. thesis, KTH, Department of Mechanics.
- [9] Lesser, M.: 1995, *The Analysis of Complex Nonlinear Mechanical Systems*. World Scientific Publishing Co. Pte. Ltd.
- [10] McGeer, T.: 1990a, 'Passive Bipedal Running'. In: *Proceedings of the Royal Society of London: Biological Sciences*, Vol. 240. pp. 107–134.
- [11] McGeer, T.: 1990b, 'Passive Dynamic Walking'. *International Journal of Robotics Research* **9**, 62–82.
- [12] McGeer, T.: 1990c, 'Passive Walking with Knees'. In: *Proceedings of the IEEE Conference on Robotics and Automation*, Vol. 2. pp. 1640–1645.
- [13] McGeer, T.: 1993, 'Dynamics and Control of Bipedal Locomotion'. *Journal of Theoretical Biology* **163**, 277–314.
- [14] Piiroinen, P.: 2000, 'Passive Walking: Transition from 2D to 3D'. Technical report, KTH, Department of Mechanics.

Paper 4

P4

Because of the copyright we can not publish this paper on the Internet

Paper 5

P5

Because of the copyright we can not publish this paper on the Internet

Paper 6

P6

Because of the copyright we can not publish this paper on the Internet

Paper 7

P7

3D Passive walkers: Implementing code for simulation and stability calculations

Jesper Adolfsson
Royal Institute of Technology
Department of Mechanics
S-100 44 STOCKHOLM
SWEDEN

August, 2000

ABSTRACT. This text is meant to be an introduction to the implemented code for modelling, simulating, and analyzing the 3D passive walker. Since external force functions are used, special care has to be taken when deriving the variational equations. The variational equations are used for locating periodic gaits of the walker and to determine the stability of found periodic gaits. However, the standard use has to be modified due to the discontinuities in the 3D passive walker. Numerical computations were performed in Matlab. However, Matlab's standard ODE-file format had to be suitably modified to support discontinuity handling and external force functions. Also reported here are the implemented Matlab code for locating periodic gaits of the walker.

1 Introduction

Previous studied planar passive walkers were constrained to move in a plane to prevent lateral dynamics, for example see McGeer [10]. McGeer's original planar passive walker did not have any knees, thus foot scuffing was neglected during the swing phase. McGeer later introduced a knee, which solved the foot scuffing problem. All impacts were modeled as perfectly inelastic impacts using impact laws resulting in discontinuous jumps in velocity state variables. Extending such a mechanism into three dimensions presents the modeler with some design decisions, such as the foot design, knee design, and how the two legs should be linked together. Depending on design choices it might be necessary to switch the impact model, such as the replacement of discontinuous models with stiff springs and dampers. What usually necessitates this switch is the

problem of handling multiple contact points, which might result in infinite many bounces between contact points in finite time. Thus, most of the 3D walkers using discontinuous impact models have made sure that only one contact point exist at any give time and that the switch of stance leg to swing leg and vice versa will occur at the impact instance. Examples are;

- Keeping the shape of planar walker foot, thus, creating a knife edged foot in three dimensions, see McGeer [11].
- Introducing an additional curvature of the foot, thus, making it a contact surface, see Mombaur *et al.* [12] and Kuo [7].
- Using a light foot plate that instantaneously levels with the ground at foot impact, see Wisse *et al.* [18]. The impulse from the ground plane on the foot can therefore be assumed to take place at a single point.

It is worth noting that the two first designs, in the ideal case, won't restrict rotation about the normal direction of the ground plane at the contact point. However, the foot plate design can, depending on joint design, restrict rotations. In the first two cases, the rolling constraint gets more complicated in three dimensions. The foot surface will introduce some extra design parameters. For example Mombaur *et al.* [12], used toroidal feet in the successful modeling and simulation of the *Tinkertoy* walker, thus, only adding one additional parameter to the foot design.

In contrast to the planar knee-less walkers, 3D knee-less passive walkers don't need to suffer from foot scuffing since they can lean to the opposite side of the swing leg, see Kuo [7], Coleman and Ruina [4], and Nichols and Witt [13].

The foot design chosen here is to add an additional toe contact point for each foot, which necessitates the switch from discontinuous impact models to continuous impact models. An advantage of this design is that is now possible to start the 3D walker in a configuration that intuitively should retain the essential dynamics of the planar passive walker. This configuration is to some extent similar to the ones used for implemented passive walker. Namely, that the mass centers of each body are kept in a plane and the feet are placed symmetrically about this plane. However, as shown in Figure 1, the foot points are placed at the same distance from the sagittal plane.

The remainder of this work has the following form. In section 2, the modelling assumptions are presented and discussed in detail, which includes the kinematical modelling of the 3D passive walker and how forces and torques can be modelled and implemented using external functions. Section 3 deals with the intricacies of exporting the equations of motion and variational equations to the Matlab environment. In section 4, is is shown how the standard Matlab ODE-file was altered to support discontinuity handling and variational equa-

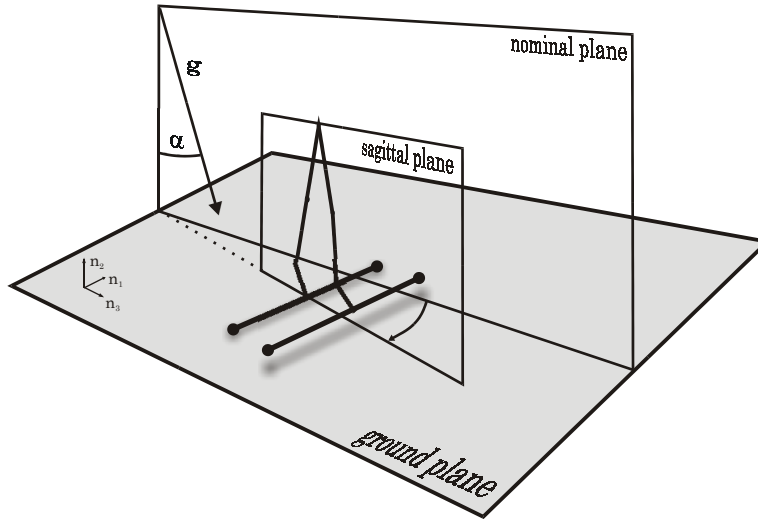


Figure 1. The toe points make regular contact with the ground plane. The nominal plane is defined by the initial hip point position and spanned by the gravity vector and the ground plane normal. The sagittal plane contains the center hip point and has a normal in the hip axis direction.

tions. The paper concludes with a discussion on how this framework can be used for parameter searches.

2 Modelling

The planar model used different sets of differential equations, see Adolphsson [2], depending on the state of the walker. Here, these impact models are replaced with linear springs and linear dampers. These springs and dampers are activated/deactivated depending on the current state of the walker, such as a foot point being in contact with ground or not or knee being locked or not. Thus, the same set of differential equations can be used throughout the simulations.

A drawback of this particular way of modeling contact forces is that it introduces extra modeling parameters, see Figure 2. The 3D passive walker was started using initial conditions from the planar passive walker. However, some manual tuning of these initial conditions was required before a gait was observed in the 3D walker. It is interesting to note that the first observed simulated gaits of the 3D passive walker didn't walk straight down the plane. Thus, the introduction of additional degrees of freedom and the switch to a continuous system introduced new interesting dynamics.

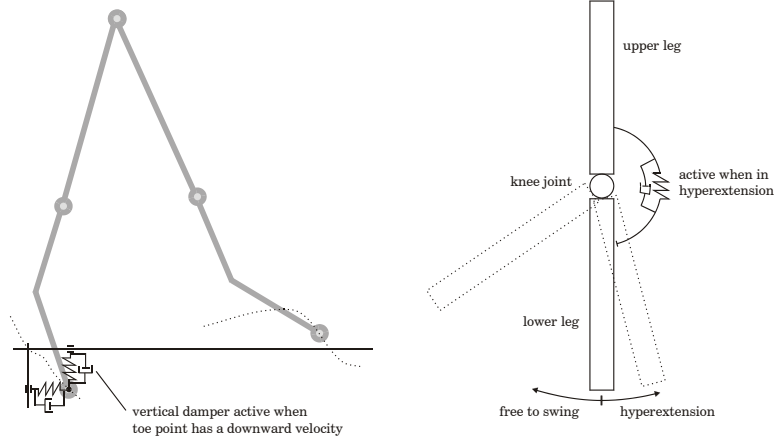


Figure 2. Stiff springs and dampers are used to model the contacts between the toe points and the ground plane. Stiff angular springs and dampers are used between the upper leg and lower leg to prevent a collapse of the knee during hyper extension.

2.1 GROUND PLANE MODELLING

Initially, the ground was modeled with stiff springs and high damping, as to mimic the inelastic impacts of the planar walker. This also prevents the toe contact points of having any larger motion while attached to the ground. It is now possible to investigate the effects of different ground plane characteristics, such as walking on a hard floor versus a soft floor. Additionally, it is straightforward to estimate the required friction coefficient by taking the ratio of the horizontal to the vertical force at each instant.

2.2 KNEE MODELLING

Since discontinuous impacts models were discarded in favor of continuous impact models, the inelastic knee impact model was also replaced with a continuous model. Thus, linear angular springs and linear angular dampers are activated/deactivated depending on the position of the knee, see right pane of Figure 2. In the planar case, the knee is assumed locked just until after the release of the hind leg. To model this in the 3D passive walker, the angular damping of the knee was assumed active in both directions as long as the knee is in hyperextension. In Adolfsson [1], it is shown how this can be passively implemented in an experimental walker.

2.3 TORSO

Another major difference between the planar walker and the 3D passive walker is the introduction of a torso. This torso acts as the link between the two legs. The torso was introduced for two reasons, firstly it was assumed that measurement equipment would have to be used in the experimental walker

and that the weight of this equipment would not be negligible. Secondly, for practical purposes it is necessary that the walker should be able to carry some payload. Thus, the effects of different torso mass configurations should be possible to investigate. However, there were no plans on using an upright torso with a center of mass situated above the hip line, which would most likely require active control to maintain the upright position. Therefore, most simulations and parameter studies presented here have the center of mass of the torso situated on the hip axis. Positioning of the center of mass on the hip line introduced a new problem. The coupling between the rotational motion of torso and the rest of the walker became too weak, thus potentially creating problems in the search of periodic gaits. To increase the coupling, a rotational damping between the legs and the torso was introduced.

2.4 SIMULATION AND ANALYSIS

Using direct numerical simulation, only stable periodic gaits are accessible. Therefore, a root finding scheme for the 3D walker was implemented to access unstable periodic gaits. Of course, the stable gaits are of main interest due to their possibility of being implemented passively. However, unstable gaits can turn into stable gaits, as parameters of the system are varied. Therefore, following unstable periodic gaits can expand the number of found solutions. In addition, it is believed that unstable or weakly unstable solutions are easier to stabilize with active control, compared to actively achieve the motion from ground up. Root finding is implemented with the Newton-Raphson scheme, which require derivative information. By using the export routine *exmex*, developed by Lennartsson [8], it is possible to export the variational equations. Since the 3D walker contains both force discontinuities and impact like changes of the state variables, the standard use of the variational equations needs to be suitably modified. In addition, the derivative information can be used to determine stability characteristics of a found periodic gait.

In the following text, some issues regarding the derivation of the equations of motions and variational equations are discussed. In the final chapters, the *Matlab* code used for integrating the equations of motion and variational equations is discussed, along with the implemented Newton-Raphson scheme for finding periodic gaits. In the appendix, the *Sophia/Maple* code for deriving the equations of motion and variational equations is given.

2.5 MAPLE/SOPHIA

The equations of motion (EOM) were derived with the aid of *Sophia*, see Lesser [9]. *Sophia* is implemented as a set of routines, which can be executed in different symbolical algebra programs, such as *Maple* and *Mathematica*. *Sophia* includes routines for deriving EOM for mechanical systems using Kane's method, see Kane and Levinson [6] or Lesser [9]. It is also useful for kinematical analysis of mechanisms. For example, *Sophia* has been used for deriving EOM in the field of robotics at ABB Robotics, vehicle dynamics see Vågstedt [17], pantographs see Drugge [5], and washing machines at Electrolux Research and

Innovation AB. It has also been used in education, both in its original form to teach constrained rigid multi-body dynamics and in a revised form, for teaching kinematics. Thus, *Sophia* is a versatile utility with many applications. Since the symbolical-algebra programs have rather limited numerical capabilities and performance, the *exmex* set of routines was developed by Lennartsson [8]. *Exmex* is currently only available for *Maple*. *Exmex* includes routines for exporting EOM to *Matlab*. For performance reasons, it exports them as *Matlab* compatible C files (mex-files). The EOM can be on both explicit and implicit form, or a combination of the two. It also includes routines for exporting variational equations (VE), which are here used for locating periodic gaits and stability calculations, see later chapters.

The *Sophia/Maple* code for deriving the EOM and VE can be found in the Appendix.

2.6 KINEMATICS

The passive walker consists of five bodies as depicted in Figure 3. The bodies are connected by hinge joints. The bodies are; an upper torso labeled $\mathcal{T}$, two upper legs labeled $\mathcal{U}_1$ and $\mathcal{U}_2$, and two lower legs labeled $\mathcal{L}_1$ and $\mathcal{L}_2$. We also consider the ground plane as a body $\mathcal{N}$. The mechanism has ten degrees of freedom. With each body $\mathcal{B}$ we associate three basis vectors $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$. We also specify a coordinate system for each body based on an origin B . A position of a point in this coordinate system will be labeled $[b_1, b_2, b_3]_{\mathcal{B}}$, vectors will use this notation too. Starting from the top, we have the reference point of the torso, labeled T which is given by $[q_1, q_2, q_3]_{\mathcal{N}}$. This reference point is situated on the hip line, which connects the two legs. The rotation from $\mathcal{N}$ to $\mathcal{T}$ is accomplished by rotating an angle q_6 about the 3-direction, an angle q_5 around the new 2-direction into an intermediate body $\mathcal{V}$. Finally, the orientation of body $\mathcal{B}$ is accomplished by rotating an angle q_4 around the 1-direction in body $\mathcal{V}$. All rotations are positive in the sense of the right hand rule. Connected to the torso by the hip hinge joints are the two upper legs. The orientations of the upper legs are given by a rotation from $\mathcal{V}$ an angle q_7 respectively q_9 about the 1-direction. Thus, the orientation of the legs doesn't depend on the angle q_4 . This is different from the definition found in Adolfsson *et al.* [3] where the orientation of the legs depended on q_4 . The reason for doing this is that under certain parameter choices the dynamics of the walker will be invariant to the orientation of the torso corresponding to the angle q_4 , see later chapters. Thus, q_4 need not to be taken into account in the search for periodic gaits.

The origins of the hip joints are situated on the hip axis. Connected to the upper legs by the knee hinge joints are the lower legs. Their orientations are accomplished by a rotation q_8 respectively q_{10} about the 1-direction. The origins of the lower legs are situated on the knee hinge joints. When all angles are zero all body-fixed axes coincide.

We introduce a set of generalized speeds $u_i, i = 1, \dots, 10$ corresponding to the generalized coordinates q_i described above. The velocity of reference point

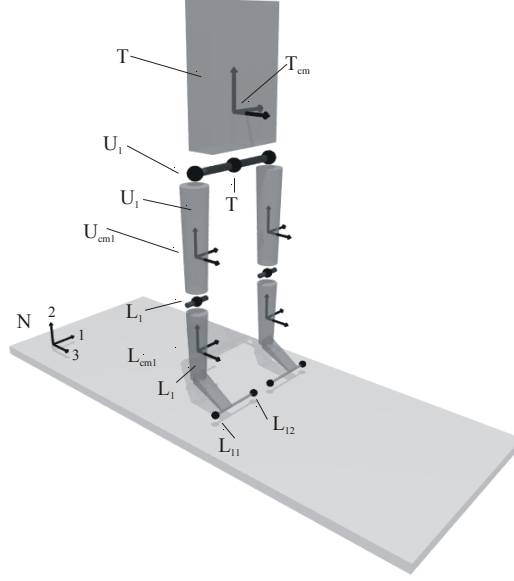


Figure 3. Illustrative picture of the 3D passive walker where bodies and points are labeled.

T is $[u_1, u_2, u_3]_{\mathcal{N}}$ and the angular velocity of the torso T is given by $[u_4, u_5, u_6]_{\mathcal{T}}$, thus the velocity components will also be independent of the rotation angle q_4 .

The angular velocities of the upper legs relative to the torso are given by $[u_7, 0, 0]_{\mathcal{T}}$ and $[u_9, 0, 0]_{\mathcal{T}}$. Similarly, the angular velocities of the lower legs are $[u_8, 0, 0]_{\mathcal{U}_1}$ and $[u_{10}, 0, 0]_{\mathcal{U}_2}$ relative to the upper legs. The kinematical differential equations (KDE) relating $\dot{q}_i$ to u_i are,

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} \cos(q_6) \cos(q_5) & -\sin(q_6) & \cos(q_6) \sin(q_5) \\ \sin(q_6) \cos(q_5) & \cos(q_6) & \sin(q_6) \sin(q_5) \\ -\sin(q_5) & 0 & \cos(q_5) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (1)$$

for the positions and,

$$\begin{bmatrix} \dot{q}_4 \\ \dot{q}_5 \\ \dot{q}_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \tan(q_5) \\ 0 & 1 & 0 \\ 0 & 0 & \cos(q_5)^{-1} \end{bmatrix} \begin{bmatrix} u_4 \\ u_5 \\ u_6 \end{bmatrix} \quad (2)$$

for the rotations of the torso. The relation between the hip and knee angles are, $\dot{q}_i = u_i, i = 7, 8, 9, 10$.

Since the forces between the ground plane and the toe points depends on the initial contact point we need to store the corresponding coordinates. One

possible solution is to add the initial contact points as state variables of the system. Since we are interested in finding periodic gaits, meaning that the state variables should return to their initial values after one step, we can give the contact coordinates relative to some reference point in the walker, for example T . At each foot point impact we update the corresponding states $p_i, i = 1, \dots, 8$ with the new coordinates, thus a periodic gait will result in a periodic behavior in the impact point states. The differential equations between the discontinuous updates of the contact point states in the 1-direction and 3-direction are,

$$\begin{bmatrix} \dot{p}_i \\ \dot{p}_j \end{bmatrix} = \begin{bmatrix} -\cos(q_6)\cos(q_5) & \sin(q_6) & -\cos(q_6)\sin(q_5) \\ -\sin(q_5) & 0 & \cos(q_5) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (3)$$

where $i = 1, 3, 5, 7$ (1-direction) and $j = 2, 4, 6, 8$ (3-direction). Since the contact points are given relative to the walker, it follows that changing the initial conditions, which corresponds to moving the walker in the plane (changing q_1 and q_3), the foot points will follow along. Therefore, the toe force will be independent of q_1 and q_3 . Thus, the dynamics will be indifferent to changes in q_1 and q_3 and we do not need to consider those state variables in the search for periodic gaits. If we include q_1 in the search for periodic gaits, we limit ourselves to gaits where the hip point T start and ends in the nominal plane.

In order to study very wide feet, Piironen, see [15], expressed the contact point positions in the $\mathcal{V}$ reference frame. Thus, the contact points will follow a rotation of the walker.

2.7 STATE VARIABLES

Along with the 28 continuous state variables there are six discrete state variables. These discrete states correspond to the knee being locked or unlocked and the toe contact points being attached/detached from the ground plane. When certain events occur in the system the discrete states change their values. Two types of events are monitored. A zero crossing of the knee angles (q_8, q_{10}) , positive values correspond to hyperextension of the knee. A zero crossing of the toe points L_{ij} vertical position. For convenience, the 28 continuous and 6 discrete states are listed in Table I. The unlocked knee and the detached foot point state is represented by a zero value and the locked knee and attached foot point state with the value one.

2.7.1 Mass properties

Each physical body has a center of mass labeled B_{cm} . The position of B_{cm} is given relative to the origo B . The moments of inertia are given about the center of mass and its six components are given relative to the body fixed basis vectors.

2.8 FORCES AND TORQUES

For modeling reasons, it is advantageous to divide the forces and torques acting on the walker into two categories. Firstly, continuous forces that are always

present, such as gravity and, secondly, intermittent forces, such as knee torques and toe point contact forces. The always-present forces lend themselves to be incorporated into the equations of motions at the modeling stage whereas the intermittent forces enters as general forces and torques in the modeling stage and are evaluated by external functions.

These external functions can be further divided into functions that have the generalized states as arguments and the functions that have functions of the generalized states as arguments. This can symbolically be expressed as,

$$\begin{aligned} f_1 &= f_1(q, u, p) \\ f_2 &= f_2(x(q, u, p)) \end{aligned} \quad (4)$$

When deriving the VE, special care has to be taken when the arguments of the functions are functions of the generalized states, since the inner derivative needs to be computed. This dependence is accomplished by explicitly declare that x is a function of (q, u, p) . Since Sophia/Maple is aware of this dependency, it can use the chain rule of differentiation to evaluate the derivatives of these force functions with respect to the generalized state. Thus, we only have to specify the derivative of the force functions with respect to the arguments of the function. We can symbolically write the derivatives of f_2 as,

$$\frac{df_2}{dq} = \frac{\partial f_2}{\partial x} \frac{dx}{dq} \quad (5)$$

We supply $\partial f_2/\partial x$ as a C function whereas dx/dq is computed by export routine *exmex*. This provides a numerically efficient solution, since all expressions of type dx/dq can be optimized together with the EOM and the VE. Thus, there is no difference in the external C file between functions that depend directly or indirectly on the generalized states.

The next sections describe how the forces and torques enter the Sophia/Maple code at the modeling stage. Since the same external C-function can be used for calculating forces or torques at different places, the arguments that differentiate the function calls are shown. For example, different toe points use the same function to calculate the force. In addition, the C implementation is given.

2.8.1 Gravity

Since the walker is assumed to walk down an incline, gravity is inclined an angle α with respect to the ground plane normal, as depicted in Figure 1. The plane containing the gravity vector is referred to as the nominal plane. The gravity force, which is acting on all bodies, can be written as,

$$F_i^{\text{grav}} = -m_i g [0, \cos(\alpha), \sin(\alpha)]_{\mathcal{N}} \quad (6)$$

where g is the gravity constant and m_i represents the mass of each body. The gravity force is incorporated into the EOM.

Table I. Continous and discrete state variables used in the 3D passive walker

Continous states	Description
q_1, q_2, q_3	Position of hip point T given in body $\mathcal{N}_2$
q_4, q_5, q_6	Angles describing the orientation of the torso
q_7, q_9	Hip angles
q_8, q_{10}	Knee angles
p_1, p_3, p_5, p_7	Initial toe contact point relative to hip point T (1-dir. in $\mathcal{N}$)
p_2, p_4, p_6, p_8	Initial toe contact point relative to hip point T (2-dir. in $\mathcal{N}$)
u_1, u_2, u_3	Velocity of hip point T given in body $\mathcal{V}$
u_4, u_5, u_6	Angular velocities of body T given in reference frame $\mathcal{V}$
Discrete states	
s_1, s_2	Knee 1 and 2
s_3, s_4	Toe 1 and 2 of leg 1
s_5, s_6	Toe 1 and 2 of leg 2

2.8.2 Hip torque

Torques are introduced between the torso and the upper legs to increase the rotational coupling between the two bodies. These torques are aligned with the hip axis and the torque applied to the torso is,

$$T_{\text{torso}} = [T_{\text{hl}_1}^e + T_{\text{hl}_2}^e, 0, 0]_{\mathcal{V}} \quad (7)$$

and

$$\begin{aligned} T_{\text{ul}_1} &= [-T_{\text{hl}_1}^e, 0, 0]_{\mathcal{V}} \\ T_{\text{ul}_2} &= [-T_{\text{hl}_2}^e, 0, 0]_{\mathcal{V}} \end{aligned} \quad (8)$$

for the two upper legs. The actual torques are calculated in an external C-function, which returns $T_{\text{hl}_1}^e$ and $T_{\text{hl}_2}^e$.

$$\begin{aligned} T_{\text{hl}_1}^e &= \text{hip}(q_4, q_7, u_4, u_7, \text{hip_k}, \text{hip_delta}, \text{hip_d}) \\ T_{\text{hl}_2}^e &= \text{hip}(q_4, q_9, u_4, u_9, \text{hip_k}, \text{hip_delta}, \text{hip_d}) \end{aligned} \quad (9)$$

Thus, the same parameters are used for leg one and two. The corresponding C function, which include support for a hip spring, is found in the code below.

```
double hip(double torso_q, double leg_q, double torso_u,
           double leg_u, double hip_k, double hip_delta, double hip_d)
{
    return hip_k*(leg_q-torso_q-hip_delta)+
           hip_d*(leg_u- torso_u);
}
```

Expressing the necessary derivatives of this function is straightforward and in the code below, the above C function is differentiated with respect to the torso angular velocity.

```
double hip_torso_u(double torso_q,double leg_q,double torso_u,
    double leg_u,double hip_k,double hip_delta,double hip_d)
{
    return -hip_d;
}
```

2.8.3 Knee torque

The lower leg is free to swing as long as it is not in hyperextension. However, when the knee is in hyperextension, corresponding to knee angles q_8 and q_{10} being positive, a collapse is prevented by an applied torque. This torque is aligned with the hip axis and acts between the upper leg and the lower leg. At the modeling stage, the torque acting on upper leg one is expressed as,

$$T_{ul_1} = [T_{kl_1}^e, 0, 0]_{\mathcal{UL}_1} \quad (10)$$

and the torque acting on lower leg one is accordingly,

$$T_{ll_1} = [-T_{kl_1}^e, 0, 0]_{\mathcal{UL}_1} \quad (11)$$

Similar to the hip case, the actual torque is calculated in an external C function, which return the value for $T_{kl_1}^e$ and $T_{kl_2}^e$.

$$\begin{aligned} T_{kl_1}^e &= \text{knee}(s_1, q_8, u_8, \text{knee_k}, \text{knee_d}, \text{knee_d_reb}) \\ T_{kl_2}^e &= \text{knee}(s_2, q_{10}, u_{10}, \text{knee_k}, \text{knee_d}, \text{knee_d_reb}) \end{aligned} \quad (12)$$

The corresponding C function is found below.

```
double knee(double s, double q,double u,double knee_k,
    double knee_d, double knee_d_reb)
{
    if (s>0.0)
        return (knee_k*q + (u <= 0.0 ? knee_d*u : knee_d_reb*u));
    else
        return 0.0;
}
```

The first argument of this C function is the discrete state of the knee. A non-zero torque is only returned if the knee is in hyperextension, which corresponds to s_1 and s_2 equal to one. The C function supports different damping constants,

depending on the rotational direction of the lower leg. When the rebound damping is positive, the knee will be sticky. When set to zero, the knee torque will model a simple mechanical stop. The derivatives with respect to q and u are given below.

```
double knee_q(double s, double q, double u, double knee_k,
              double knee_d, double knee_d_reb)
{
    if(s>0.0) return knee_k; else return 0.0;
}

double knee_u(double s, double q, double u, double knee_k,
              double knee_d, double knee_d_reb)
{
    if (s>0.0)
        return (u <= 0.0 ? knee_d : knee_d_reb);
    else
        return 0.0;
}
```

2.8.4 Toe point forces

The interaction between the ground plane and the walker occur at the four toe points L_{ij} , $i = 1, 2$ and $j = 1, 2$. These toe points are situated on the lower legs as shown in Figure 3. The resulting force has components along each of the three basis directions in $\mathcal{N}$. There is no torque at the contact point, which is one of the reasons for using two contact points at each foot. Two contact points will only allow rotation around the axis defined by the line going through the two contact points. At the modeling stage, the applied force on lower leg one is given by,

$$F_{l1} = [f_{111}^e + f_{121}^e, f_{112}^e + f_{122}^e, f_{113}^e + f_{123}^e]_{\mathcal{N}}. \quad (13)$$

The torque resulting from this force, acting on the lower leg, equals,

$$T_{l1} = [t_{111}^e + t_{121}^e, t_{112}^e + t_{122}^e, t_{113}^e + t_{123}^e]_{\mathcal{N}}, \quad (14)$$

where $t_{ij_k}^e$ is computed from the force expression using the common sub expression feature of *cmex*. Thus,

$$[t_{111}^e, t_{112}^e, t_{113}^e]_{\mathcal{N}} = \overline{L_{cm1}L_{11}} \times [f_{111}^e, f_{112}^e, f_{113}^e]_{\mathcal{N}} \quad (15)$$

where the vector from the toe contact point L_{11} to the center of mass of the lower leg L_{cm1} is expressed in the $\mathcal{N}$ reference frame. The forces and torques for the other contact points are calculated in the same way. Since each direction in $\mathcal{N}$ is treated independently, three different function calls are made. The external C function calls to calculate f_{111}^e , f_{112}^e , and f_{113}^e are,

$$f_{111}^e = \text{toefx}(s_4, x, v_x, kX, dX, p_1) \quad (16)$$

3D Passive walkers: Implementing code

$$\begin{aligned} f_{11_2}^e &= \text{toefy}(s_4, y, v_y, kX, dX) \\ f_{11_3}^e &= \text{toefz}(s_4, z, v_z, kX, dX, p_2) \end{aligned}$$

where s_4 reflects the discrete state of the toe point being attached/detached to the ground plane. The arguments x, y , and z are the position of the toe point given in reference frame $\mathcal{N}$ and vx, vy , and vz are the corresponding velocities. These six quantities, which all depend on the generalized states of the system, are given as common sub expressions in Maple/Sophia. In the C code found below, two functions that models the vertical force differently are given. The first one models a ground plane where the damper is only active when the velocity is into the ground plane. Thus, at rebound, the vertical force will only come from the spring. The second force function models a ground plane where the damper and spring are active as long as the force on the toe point is positive.

```
double toefy(double stat, double y, double vy,
             double kY, double dY)
{
    if(stat>0.0)
        return -kY*y-(vy<=0.0 ? 1.0 : 0.0)*dY*vy;
    else
        return 0.0;
}

double toefy(double stat, double y, double vy,
             double kY, double dY, double dYrebound)
{
    double Fy;
    if(stat>0.0)
    {
        Fy = -kY*y-(vy<0 ? dY : dYrebound)*vy;
        return (Fy>0.0 ? Fy : 0.0);
    }
    else
        return 0.0;
}
```

The horizontal force functions `toefx` are given in the code found below, The argument `posx` is the initial toe contact point coordinate in the $\mathbf{n}_1$ -direction. It should be noted that with the second vertical force model, it is possible to have forces in the ground plane directions while the vertical force component is zero. This is somewhat unphysical and could be avoided by triggering the detachment of the toe point when the vertical force becomes zero. However, handling a possible reattachment, occurring shortly after the release, would then need some special considerations, such as introducing dynamics for the

ground plane. It is believed that the success of the passive 3D walker will not depend on the exact form of ground plane model. Below is the needed derivatives with respect to the function arguments given. Note that the derivative with respect to the initial contact coordinate posx is also required, since posx belongs to the states of the system that need to be varied.

```
double toefx(double stat, double x, double vx,
             double kX, double dX, double posx)
{
    if(stat>0.0) return -kX*(x-posx)-dX*vx; else return 0.0;
}

double toefx_x(double stat, double x, double vx,
               double kX, double dX, double posx)
{
    if(stat>0.0) return -kX; else return 0.0;
}

double toefx_vx(double stat, double x, double vx,
                double kX, double dX, double posx)
{
    if(stat>0.0) return -dX; else return 0.0;
}

double toefx_p(double stat, double x, double vx,
               double kX, double dX, double posx)
{
    if(stat>0.0) return +kX; else return 0.0;
}
```

2.9 HELPER FUNCTIONS

Since the toes initial-contact-point states p_i have to be updated at each ground impact, helper functions are used to calculate this initial position. Since these functions return the position of the toe point given the state and parameters of the system, it is necessary that the state variable correspond to an impact configuration. For efficiency, these functions are coded as C functions.

The rotation matrices, which give the directions of the basis vectors of each reference frame, are also exported as C functions. These functions are mainly used by the visualization code, which is described in later sections.

3 Exporting EOM and VE

Exporting the EOM and VE to *Matlab* C code is accomplished by using the *exmex* routine, which is developed by Lennartsson [8]. For simulation of the 3D walker, the kinematical differential equations (KDE) and EOM are exported together. For integration of the variational equations the KDE, EOM, and VE are exported together. Thus, optimization can take place across all the equations.

The KDE are put on explicit form whereas the EOM and VE are put on implicit form due to their complexity. The common subexpression feature of *exmex* is used to evaluate external function calls (force/torque functions) and the arguments used in these calls, see Figure 4. During derivation of the VE, Maple needs to know the external functions dependence on state variables. However, this syntax is not C compatible and a substitution of the VE substitution list is needed to make the functions "C-friendly". In Figure 4, an example of the process of exporting the EOM and VE is shown. Before exporting the EOM, the VE substitution list needs to be substituted into the common sub expressions. When generating the VE, this substitution should be delayed until after *exmex* has internally generated the VE. Therefore, this list is passed as an argument to *exmex*. Note that the order in the list is important.

Since Maple's optimize command is used to reduce the computational cost, special care has to be taken when compiling the exported C file. It seems like C compilers have trouble when compiling already optimized computational sequences. Therefore, this feature has to be turned off during compilation or the result will be excessively long compilation times, especially when handling large files such as the VE.

When using Maple's "optimize" command, the resulting size of the compiled file will be about 300kb. Using the option "tryhard" in optimize it is possible to further reduce the size to about 70kb. However, it takes Maple about 40 times longer to generate the EOM code using this option and 300 times longer to generate the VE code.

Together with the EOM, the necessary C code to detect the six events is also exported. The produced file is compatible with all *Matlab* ODE integrators.

3.1 DISCONTINUITY MAPPINGS

If there is a discontinuity associated with an event, during integration of the VE, a discrete mapping needs to be applied to the derivative of the flow. The derivative of this mapping can be written as,

$$M_{a \rightarrow b} = \frac{\partial G}{\partial x} \Big|_{x_b} + \frac{\left(F_a(x_a) - \frac{\partial G}{\partial x} \Big|_{x_b} F_b(x_b) \right) \frac{\partial N}{\partial x} \Big|_{x_b}}{\frac{\partial N}{\partial x} \Big|_{x_b} F_b(x_b)} \quad (17)$$

where G is the discontinuous jump in state variables, F_a and F_b is the vector field before respectively after the event and N is the event surface. The subscripts a and b refer to the instant just before the event respectively just after

the event occurred. For a derivation of this mapping, see Adolfsson *et al.* [3]. The evaluation of the vector field (F_a and F_b) is accomplished by calling the exported EOM C file with the discrete and continuous state corresponding to just before and just after the event occurred. The derivative of the event surface, $\partial N/\partial x$, is derived in Maple by differentiating the function that triggers the event. This derivation is with respect to all state variables. The expressions for the knee events are particularly simple and are not exported as C files. The expressions for the toe point events are more complicated and are exported from Maple as C files. Finally, the quantity $\partial G/\partial x$ only differs from the identity matrix during toe point impact, since at knee locking no jump in the state variables occurs. At toe point impact, the toe point states will be updated with the new impact position. The expression for $\partial G/\partial x$ is derived in Maple and exported as a C function.

4 Numerical integration and periodic orbit search

4.1 INTEGRATION OF THE EOM

Numerical integration of the EOM is performed in the *Matlab* environment. One of the advantages of *Matlab* is its set of robust direct numerical integrators which range from non-stiff to stiff, low-order to high-order, and variable-order integrators. The speed of these integrators is moderate due to their implementation in *Matlab*'s own language.

A common interface is introduced for direct numerical integration of dynamical systems containing continuous states, discrete states, and VE. Thus, the structure of Matlab's ODE file was suitably modified into a statefile, which include support for discrete states, event response and discontinuity mappings. In addition, an extra top layer, named stateflow, was introduced for calling the event response functions and handling the continuous and discrete states, see Figure 5. The statefile now needs to evaluate the following,

- EOM
- Jacobian of the EOM
- *EOM and VE together*
- Event detection
- *Event response*
- $\partial N/\partial x$, $\partial G/\partial x$

where italic text describe the additions compared to Matlab's original ODE file. Since some integrators need to evaluate the Jacobian of the EOM, support for extracting the Jacobian from the VE was also added. As seen in Figure 5, the event response function is called by the stateflow function. These events are detected by the event trigger function, which terminates the current integration and return execution to stateflow. Some differences exist when integrating the VE, see next section. In addition, the event response function prints a status

message that tells what event occurred and at what time. This can be used to monitor the success of the simulation.

Since Matlab's integrators does not support discrete states and *exmex* only export files compatible with these integrators, the discrete states have to be transferred into the parameter list by the statefile layer.

During integration of the EOM, the ode15s integrator is used because it produces the fastest execution times. The integrator is a variable order integrator for stiff differential equations. The 3D walker system is stiff due to the high forces associated with the stiffness of the ground plane. This integrator requires the evaluation of the Jacobian. This evaluation can be accomplished either numerically or analytically, but the particular choice does not seem to have any larger influence on integration times. In Table II, some statistics regarding code size and execution times are shown. As seen in this table, there is little difference in execution time between the larger and smaller file. The reason for this is that most of the time is spent in the integrator and not in the function evaluating the derivatives.

Table II. Statistics using a PII 300MHz, Matlab 5.3, all integration tolerances set to 1e-6, and no visualization.

Desc.	dll file size	exec. time	% in func.	time
EOM	200kb	21.8s	19	4.1s
EOM, tryhard	90kb	18.6s	7.6	1.4s
VE	850kb	375s	49	181s
VE, tryhard	312kb	253s	31.6	80s

4.2 VE INTEGRATION

During integration of the VE, an extra layer is introduced between stateflow layer and statefile layer, see Figure 4. The purpose of this layer is twofold. Firstly, it calls the event response function, which updates the discrete and continuous states. Secondly, it calls the VE event function, which returns $\partial N/\partial x$ and $\partial G/\partial x$. With this information and two calls to evaluate the vector field before and after the event (F_a and F_b), it is possible to calculate the discrete mapping, equation (17), which is applied to the derivative of the flow.

The ode113 integrator is used for integration of the VE. It is a variable order integrator for non-stiff differential equations. A stiff solver would have been preferred but all ODE solvers of *Matlab* require the evaluation of the Jacobian. Evaluating the Jacobian would be excessively hard due to the number of states associated with integration of the VE (28 states for the EOM plus $28 \cdot 28$ states for the VE).

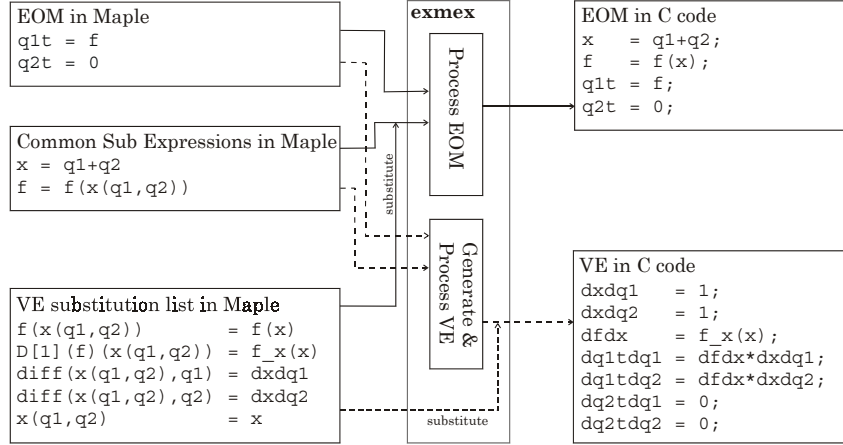


Figure 4. Showing the processing of the EOM and common subexpression feature to generate both EOM and VE in C code.

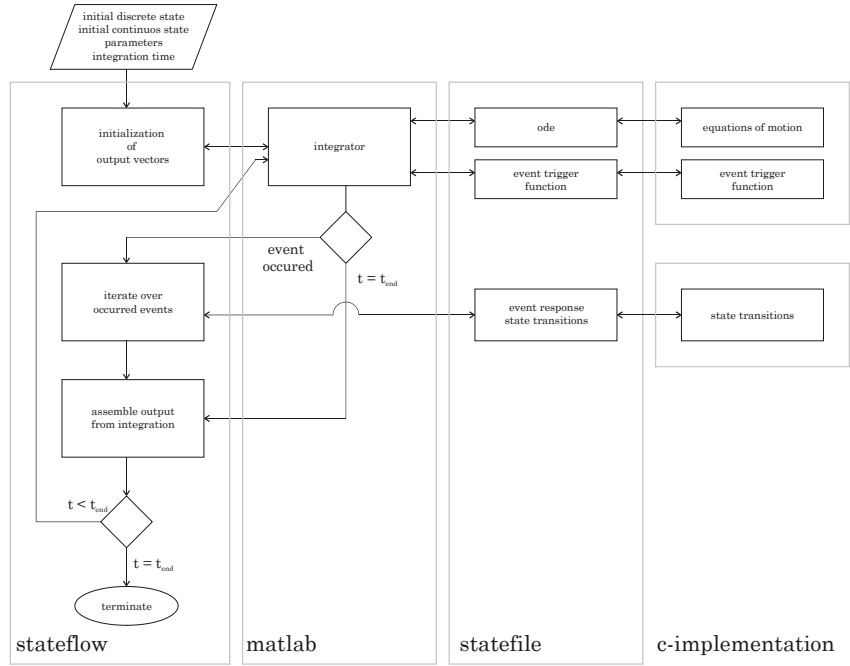


Figure 5. The different layers used during simulation of the EOM.

3D Passive walkers: Implementing code

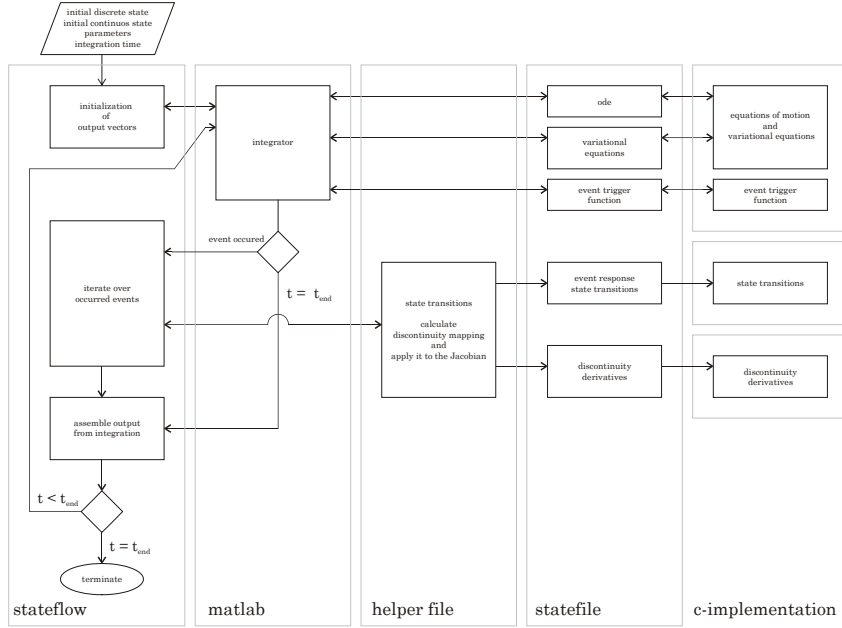


Figure 6. The different layers used during integration of the equations of motions (EOM) and variational equations (VE) together.

4.3 VISUALIZATION

During simulation, it is possible to animate the 3D walker. This visualization can provide clues why a certain simulation has failed. During normal parameter variations, it is wise to turn the animation off to reduce the execution time.

There is no mechanism for transferring the parameters of the simulation to the visualization function. Therefore, the parameter set has to be declared global in the *Matlab* environment.

4.4 PERIODIC ORBIT SEARCH

The search for periodic orbits uses the Newton-Raphson scheme, which require the integration of the VE. The goal of the search is to find an initial condition x_0 and a integration time T_0 which returns the walker to the same configuration x_0 after integrating the EOM a time T_0 . This search can be written as

$$\Phi(x_0, T_0) - x_0 = 0 \quad (18)$$

where the flow function Φ satisfies $\Phi(x_0, t) = x(t)$ given $x(0) = x_0$. Since every point on the trajectory of a periodic solution satisfy the above condition, a Poincaré section is introduced to make the solution unique,

$$H(x_0) = 0. \quad (19)$$

Given a good initial approximation of a periodic orbit (x_0, T_0) we would now like to get an update $(\Delta x_0, \Delta T_0)$ which takes us closer to the periodic orbit. Using the Newton-Raphson scheme, the equations to solve are,

$$\begin{bmatrix} \frac{\partial \Phi}{\partial x}|_{x_0, T_0} - I & \frac{\partial \Phi}{\partial t}|_{x_0, T_0} \\ \frac{\partial H}{\partial x}|_{x_0} & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta T \end{bmatrix} = - \begin{bmatrix} \Phi(x_0, T_0) - x_0 \\ H(x_0) \end{bmatrix} \quad (20)$$

where $\partial \Phi / \partial x$ is obtained by integrating the VE with the identity matrix as initial condition, $\partial \Phi / \partial t$ is the vector field at the end of the integration. With a sufficiently good initial approximation, the scheme is expected to converge. In the implemented code, convergence is said to be reached when the difference between the quadratic norm of two consecutive iterations is below some pre-defined value. Although it is unphysical to compare angles and lengths, this method has proven adequate.

The Poincaré section that is used corresponds to one of the knee angles having a specific value. As the parameter is varied during the parameter search, it can happen that an event moves through the Poincaré section. Therefore, an automatic change of Poincaré section was implemented. This works by finding the configuration having maximum time distance to any event. In Figure 7, a flow chart is shown which describes the search for periodic orbits.

By using extrapolation, it is possible to speed up the search, requiring less iterations. Termination of the search is done when the end of the parameter path is reached or if the Newton-Raphson iterations doesn't converge within a specified number of iterations. It is not uncommon that convergence isn't obtained, Some possible causes are,

- No solution exists for the parameter set. Usually a saddle node or grazing bifurcation has occurred. The grazing bifurcation might be due to foot scuffing.
- The extrapolated initial condition and new parameter set is too far from the periodic solution and therefore the walker has fallen during the simulation.
- The extrapolated initial condition and new parameter set is too far from the periodic solution and therefore the Newton-Raphson scheme doesn't converge.

4.5 ADDITIONAL NOTES

To easily store and manage data about found periodic gaits, a Matlab structure was used to organize the information, see below. Among other things, the structure contains information about the state after one stride, y_1 . This can be used to find the step length as well as to find out if the walker walks at an angle, i.e. doesn't return to the nominal plane after one stride. Walking at an angle results in different values in the first component of y_0 and y_1 , which corresponds to the hip center position. The threshold value, which is also found in the structure, is used to signal when the Poincaré cut is too close to one

3D Passive walkers: Implementing code

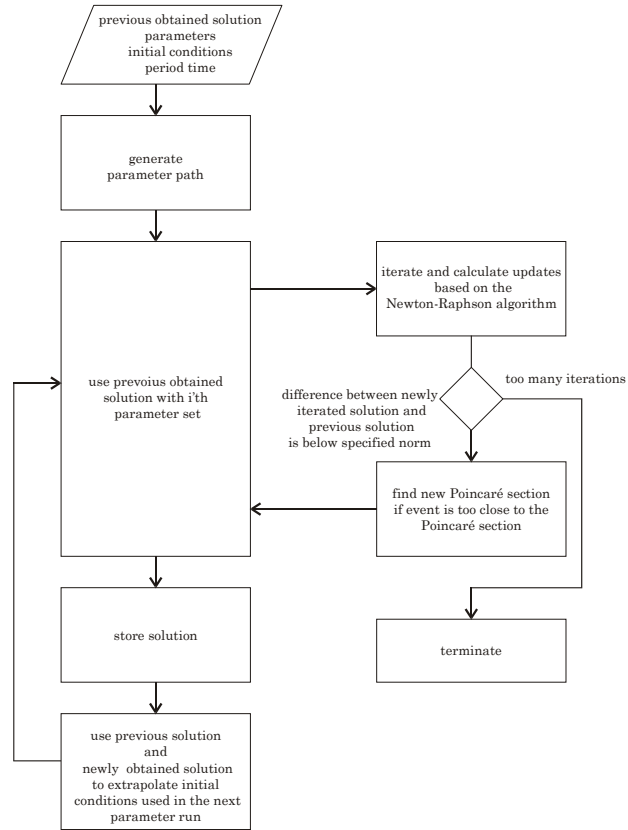


Figure 7. Graph showing the main features of the code used for finding periodic gaits during parameter variations.

of the event surfaces and therefore needs to be changed. Because of this, the value used for the Poincaré cut is also stored.

Table III. The Matlab structure used to store essential data about a found periodic gait.

ws =		
ver:	'2.0'	version number
date:	'99-10-07'	creation date/time
time:	'10.50'	
auth:	'Jesper Adolfsson'	
desc:	'Foot-width test'	
y0:	[1x28 double]	initial condition
T0:	1.6230	period time
stat0:	[-1 -1 1 1 -1 1]	initial discrete state
params:	[86x1 double]	parameters of the system
J1:	[28x28 double]	Jacobian of periodic orbit
y1:	[1x28 double]	configuration after one stride
delta_t_thresh:	0.0500	threshold for pc-event
pc_in_y7:	0.3000	currently used Poincaré cut

4.6 SIMULATION EXAMPLE

Below, the command for invoking a simulation and the resulting output is shown. As simulation is progressing the type and the time an event occurred is printed. If visualization is turned on, the 3D-walker is drawn at the times found in the integration time vector (third argument of stateflow).

Table IV. The output resulting from an ordinary gait cycle. Brackets with only numbers represents the toe points and bracket containing k1 and k2 represents the knees. Empty brackets correspond to an unlocked/detached state.

```
>> [t,y,s,te,ye,ie,se]=stateflow('ode15s','walker_3d_ode',
[0:0.05:ws.T0],ws.y0,options,ws.stat0,ws.params);
```

```
[ ] [ ] [k1] [k2] [22] [21] 0.324756(s)
[11] [ ] [k1] [k2] [22] [21] 0.564310(s)
[11] [12] [k1] [k2] [22] [21] 0.574678(s)
[11] [12] [k1] [k2] [22] [ ] 0.582882(s)
[11] [12] [k1] [k2] [ ] [ ] 0.593838(s)
[11] [12] [k1] [ ] [ ] [ ] 0.612024(s)
[11] [12] [k1] [k2] [ ] [ ] 1.136243(s)
[11] [12] [k1] [k2] [ ] [21] 1.375908(s)
[11] [12] [k1] [k2] [22] [21] 1.386299(s)
[ ] [12] [k1] [k2] [22] [21] 1.394484(s)
```

```
[ ] [ ] [k1] [k2] [22] [21] 1.405455(s)
[ ] [ ] [ ] [ ] [k2] [22] [21] 1.423651(s)
```

In Figure 8, the walker is sampled at different characteristic events, during eleven consecutive steps. In the upper right pane, the butterfly shaped motion of the hip center point T is shown. From this figure, it is clear that the mechanism has motion in all spatial directions (cf. Human Walking [16]). The parameters and initial conditions used, result in a weakly asymptotically stable periodic gait. The largest eigenvalue of the linearized system is 0.9685. The parameters and initial conditions of the 3D walker can be found in Table III and IV. Note that this particular walker has an overlap in the feet of 0.10m. In Figure 9, the time history for the knee and hip angle is shown. In addition, the times for different events are shown. Note that after knee impact, the knee angle is less than the hip angle, corresponding to a locked knee.

4.7 STABILITY CALCULATIONS

The search for periodic gaits are time consuming due to the size of the code that compute the variational equations. Also, as previously mentioned, it is common that the automatic search terminates due to different reasons, see previous bullet list. Therefore, manual search can sometimes be preferred over automatic search, such as interactively trying new parameter paths to avoid problems. However, a comparably fast computer is needed to perform manual search. The code has been tested on a computer equipped with the Compaq Alpha EV6 processor yielding good performance when integrating the variational equations, which typically takes 20-30 seconds for one stride. Using this computer, it typically takes a couple of working days to find all connected solution branches of a found periodic gait.

5 Conclusion

The methods and code presented here provide a framework for simulation and finding periodic gaits of a 3D passive walker. The methods and implemented code are not only useful for studying 3D passive walking, but could also be adapted for investigating other systems. The developed *Matlab* framework, which handles continuous and discrete systems and integration of variational equations, provide a good balance between flexibility and execution speed. The integration of the variational equations provides improved robustness, speed, and accuracy compared to using numerical derivatives. Using numerical derivatives, one has to assure that the obtained derivatives are accurate. Using integration of variational equations, the accuracy is controlled by the accuracy of the integration. In addition, calculating the Jacobian based on numerical differentiation would require at least 28 integrations (same as the number of states) of the equations of motion. This would approximately take twice the time compared to integrating the variational equations, see Table II.

Table V. Parameters used in the simulation of the 3D walker found in Figure 8. See Figure 3 for a description of the mass and point labels used.

Body	mass (kg)	moments of inertia (kg m^2)					
		x	y	z	xy	xz	yz
$\mathcal{T}$	0.800	2.0000	2.0180	2.4000	0.0	0.0	0.0
$\mathcal{U}_1, \mathcal{U}_2$	2.354	0.0080	0.0010	0.0100	0.0	0.0	0.0
$\mathcal{L}_1, \mathcal{L}_2$	1.013	0.0393	0.0010	0.0100	0.0	0.0	0.0

Point Label	x (m)	y (m)	z (m)	Ref. frame
T_{cm}	0.0000	0.0000	0.0000	$\mathcal{T}$
U_1	-0.1000	0.0000	0.0000	$\mathcal{V}$
U_2	0.1000	0.0000	0.0000	$\mathcal{V}$
$U_{\text{cm}1}, U_{\text{cm}2}$	0.0000	-0.0910	0.0000	$\mathcal{U}_1, \mathcal{U}_2$
L_1, L_2	0.0000	-0.3500	0.0000	$\mathcal{U}_1, \mathcal{U}_2$
$L_{\text{cm}1}, L_{\text{cm}2}$	0.0000	-0.1663	0.0432	$\mathcal{L}_1, \mathcal{L}_2$
L_{11}	0.0000	-0.3132	0.1050	$\mathcal{L}_1$
L_{12}	-0.1500	-0.3132	0.1050	$\mathcal{L}_1$
L_{21}	0.0000	-0.3132	0.1050	$\mathcal{L}_2$
L_{22}	0.1500	-0.3132	0.1050	$\mathcal{L}_2$

Hip/Knee	Value	Plane	Value
k_{knee}	500.0 Nm rad^{-1}	k_x, k_y, k_z	5000 Nm^{-1}
c_{knee}	20.0 Nms rad^{-1}	c_x, c_y, c_z	500 Nsm^{-1}
c_{hip}	0.04 Nms rad^{-1}	α	0.08 rad

Table VI. Initial condition used for the direct numerical simulation of the 3D passive walker found in Figure 8.

Configuration			
State	Value	State	Value
q_1	0 m	q_{10}	$-7.1645 \cdot 10^{-3}$ rad
q_2	$6.6037 \cdot 10^{-1}$ m	q_{11}	$1.3116 \cdot 10^{-1}$ m
q_3	m	q_{12}	$-2.7171 \cdot 10^{-1}$ m
q_4	$-2.7728 \cdot 10^{-1}$ rad	q_{13}	$-1.8835 \cdot 10^{-2}$ m
q_5	$1.4256 \cdot 10^{-2}$ rad	q_{14}	$-2.7199 \cdot 10^{-1}$ m
q_6	$4.9098 \cdot 10^{-2}$ rad	q_{15}	$-6.5930 \cdot 10^{-2}$ m
q_7	$2.2721 \cdot 10^{-2}$ rad	q_{16}	$7.4855 \cdot 10^{-2}$ m
q_8	$7.9218 \cdot 10^{-1}$ rad	q_{17}	$8.4060 \cdot 10^{-2}$ m
q_9	$4.7757 \cdot 10^{-2}$ rad	q_{18}	$7.4581 \cdot 10^{-2}$ m
Velocities			
State	Value	State	Value
u_1	$-1.8026 \cdot 10^{-2}$ ms^{-1}	u_6	$1.8713 \cdot 10^{-2}$ rads^{-1}
u_2	$1.6363 \cdot 10^{-3}$ ms^{-1}	u_7	$-3.9031 \cdot 10^{-0}$ rads^{-1}
u_3	$1.2288 \cdot 10^{-1}$ ms^{-1}	u_8	$5.2527 \cdot 10^{-0}$ rads^{-1}
u_4	$3.6005 \cdot 10^{-4}$ rads^{-1}	u_9	$1.9532 \cdot 10^{-1}$ rads^{-1}
u_5	$5.6527 \cdot 10^{-2}$ rads^{-1}	u_{10}	$-1.5636 \cdot 10^{-2}$ rads^{-1}

3D Passive walkers: Implementing code

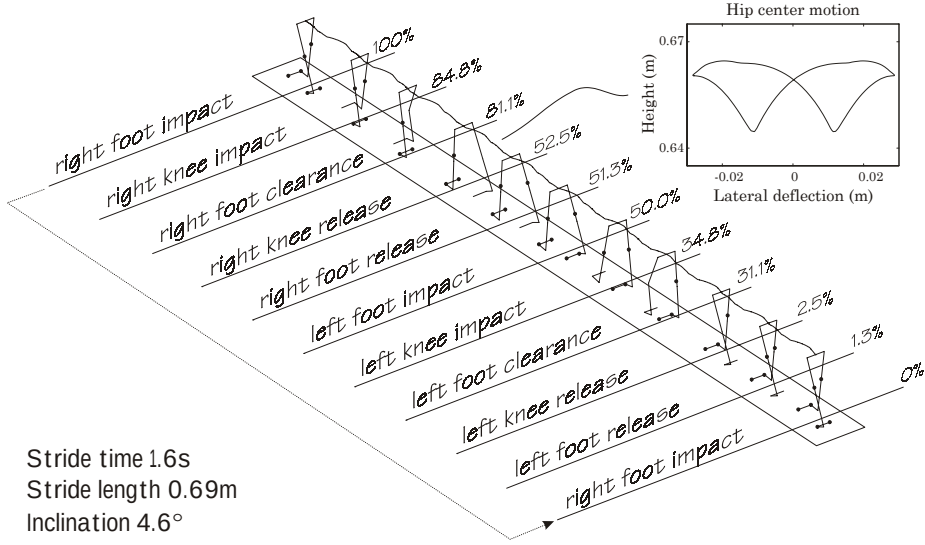


Figure 8. Stick animation of a stable periodic gait of the 3D passive walker. Parameters and initial conditions from Table IV and V. Toe clearance refers to the configuration having nonzero local minimum in the toe point point distance to the ground plane.

5.1 OUTLOOK

One of the goals of the parameter search is to find parameter sets that are possible to implement. Note that the parameters used here are hard to realize due to the high moment of inertia radius for the torso, 1.7m. As mentioned before, the parameter path to an implementable configuration might include many detours in parameter space. Therefore, fast execution of the integration of the variational equations is important. As seen in Table II, most of the execution time is spent in the integrators, thus faster integrators would significantly improve execution speed. This could be accomplished by switching from the integrators implemented in Matlab code to their C versions, which are available in the Matlab C/C++ library. Another speed increase would be to get rid of the extra layers associated with the developed framework. However, this might include major changes of the code and export routines. The time spent in the integrator, during integration of the equations of motion, account for almost 60% of the execution time. Thus, a switch to faster integrators, in combination with using the 'tryhard' option to optimize could result in almost real-time performance. This could inspire many new interesting studies, such as real-time control by a human operator. In essence, one could learn how to walk, start, and stop in a simulated environment. Since it would be possible

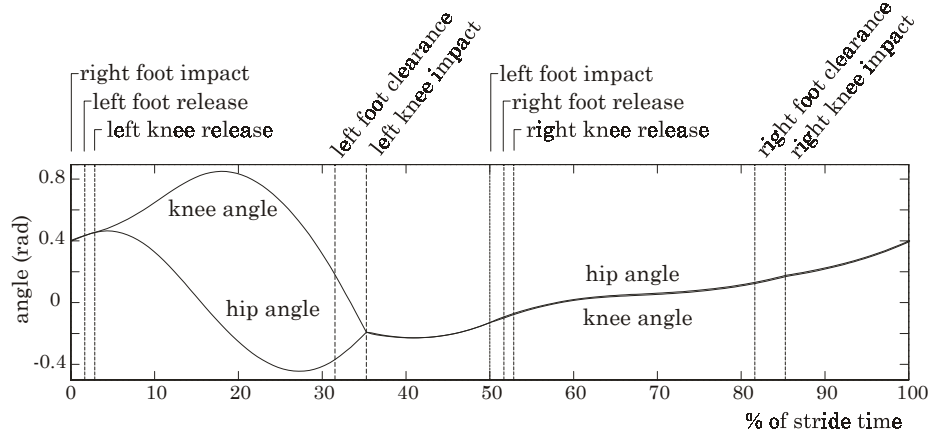


Figure 9. The time evolution for the upper leg and lower leg. Upper leg angle correspond to q_7 and lower leg angle to $q_7 + q_8$.

to record the control signals, ideas on how human control their walking might be obtained.

Since the dynamical systems usually undergo bifurcations during parameter searches, an automatic scheme for finding the resulting solution branches would be desirable. Schemes like this have been implemented in various programs, see Olsson [14].

A database for storing found periodic gaits would be beneficial in the search for new possible configurations. Such a database should store all previous found gaits and be able to answer which of the previous found periodic gaits is closest to the new wanted configuration, thus simplify the search.

6 Appendix

6.1 MAPLE/SOPHIA CODE

The Maple sessions is reset and the Sophia library is read:

```
> restart;
> read 'sophiaV5';
```

The continuous state variables are declared as being functions of time:

```
> &kde(10);
> dependsTime(seq(p.g,g=1..8));
```

6.1.1 Frame relations

Frame relations are defined, starting with the rotation from inertial reference frame $\mathcal{N}$ to the intermediate frame $\mathcal{V}$ and torso reference frame $\mathcal{T}$ and continuing with left and right upper and lower legs:

```

> chainSimpRot([N,T1,3,q6],[T1,V,2,q5],[V,T,1,q4]);
> chainSimpRot([V,UL1,1,q7],[UL1,LL1,1,q8]);
> chainSimpRot([V,UL1,1,q7],[UL1,LL1,1,q8]);

```

6.1.2 Geometry

All relevant points of the walker are defined. The absolute positions of the centers of mass are defined using partitioned vectors. This reduce the computational complexity during derivation of the equations of motion, see Lennartsson [8].

```

> rbody_hip_center:=N &ev [q1,q2,q3];
> r_torso_cm_rel :=B &ev [rx,ry,rz];

> rleg1_jointpos :=V &ev [r11,0,0];
> rleg2_jointpos :=V &ev [r12,0,0];

> rleg1_upper_cm_rel:= UL1 &ev [r11u1,r11u2,r11u3];
> rleg2_upper_cm_rel:= UL2 &ev [r12u1,r12u2,r12u3];

> rleg1_upper_jointpos_rel:= UL1 &ev [0,11u,0];
> rleg2_upper_jointpos_rel:= UL2 &ev [0,12u,0];

> rleg1_lower_cm_rel:= LL1 &ev [r11l1,r11l2,r11l3];
> rleg2_lower_cm_rel:= LL2 &ev [r12l1,r12l2,r12l3];

> rbody_cm:=mkc(rbody_hip_center,r_torso_cm_rel);

> leg1upper_cm:=mkc(rbody_hip_center,
  rleg1_jointpos, rleg1_upper_cm_rel);

> rleg2upper_cm:=mkc(rbody_hip_center,
  rleg2_jointpos,rleg2_upper_cm_rel);

> rleg1lower_cm:=mkc(rbody_hip_center,
  rleg1_jointpos,rleg1_upper_jointpos_rel,
  rleg1_lower_cm_rel);

> rleg2lower_cm:=mkc(rbody_hip_center,
  rleg2_jointpos,rleg2_upper_jointpos_rel,
  rleg2_lower_cm_rel);

```

6.1.3 KDE

The KDE is derived by expressing the velocity of the point T in the intermediary reference frame $\mathcal{V}$ (no q_4 rotation) and setting the expression for velocities and angular velocities the generalized speeds and generalized velocities equal. Note that the KDE for the toe contact points are included in the new KDE:

```

> v_body_hipcenter:= V &ev [u1,u2,u3];

```

```

> omega_body:=V &ev [u4,u5,u6];
> v_body_hipcenter_qt:= diffFrameTime(rbody_hip_center,N);
> omega_body_qt:=N &aV B;

> ikde:={seq((V &to v_body_hipcenter &c i) =
  (V &to v_body_hipcenter_qt) &c i,i=1..3),
  seq((V &to omega_body &c i) =
  (V &to omega_body_qt) &c i,i=1..3),
  seq(p.(2*i-1).t=-q1t,i=1..4),seq(p.(2*i).t=-q3t,i=1..4)};

> kde_new := simplify(solve(ikde,
  {seq(q.i.t,i=1..6),seq(p.i.t,i=1..8)}) union
  {seq(q.i.t=u.i,i=7..10)} );

```

6.1.4 Velocities and angular velocities

Velocities of the center of mass points and angular velocities expressed in the generalized speeds. Please note the `cdft` command (frame-based differentiation), which is used on partitioned vectors. The `ccpt` command is used to remove zero elements in the partitioned vectors.

```

> v_body_cm := ccpt(map(Esimplify,subs(kde_new,cdft(rbody_cm,N)))));
> vleg1upper_cm :=
  ccpt(map(Esimplify,subs(kde_new,cdft(rleg1upper_cm,N)))));

> vleg2upper_cm :=
  ccpt(map(Esimplify,subs(kde_new,cdft(rleg2upper_cm,N)))));
> vleg1lower_cm :=
  ccpt(map(Esimplify,subs(kde_new,cdft(rleg1lower_cm,N)))));
> vleg2lower_cm :=
  ccpt(map(Esimplify,subs(kde_new,cdft(rleg2lower_cm,N)))));

> omega_leg1upper := &simp subs(kde_new,N &aV UL1):
> omega_leg2upper := &simp subs(kde_new,N &aV UL2):
> omega_leg1lower := &simp subs(kde_new,N &aV LL1):
> omega_leg2lower := &simp subs(kde_new,N &aV LL2):

```

6.1.5 Momentum and angular momentum

The `csm` command is used to perform scalar multiplication on partitioned vectors. The angular momentum is stated using ordinary Sophia syntax.

```

> pbody:=csm(m_body,v_body_cm):
> pleg1upper:=csm(ml1u,vleg1upper_cm):
> pleg2upper:=csm(ml2u,vleg2upper_cm):
> pleg1lower:=csm(ml1l,vleg1lower_cm):
> pleg2lower:=csm(ml2l,vleg2lower_cm):

> I_body:=EinertiaDyad(I_B11,I_B22,I_B33,I_B12,I_B13,I_B23,B):
> hbody:=I_body &o omega_body:

```

3D Passive walkers: Implementing code

```

> I_leg1upper:=EinertiaDyad(I_LU111,I_LU122,I_LU133,
  I_LU112,I_LU113,I_LU123,UL1):
> hleg1upper:=I_leg1upper &o omega_leg1upper:

> I_leg2upper:=EinertiaDyad(I_LU211,I_LU222,I_LU233,
  I_LU212,I_LU213,I_LU223,UL2):
> hleg2upper:=I_leg2upper &o omega_leg2upper:

> I_leg1lower:=EinertiaDyad(I_LL111,I_LL122,I_LL133,
  I_LL112,I_LL113,I_LL123,LL1):
> hleg1lower:=I_leg1lower &o omega_leg1lower:

> I_leg2lower:=EinertiaDyad(I_LL211,I_LL222,I_LL233,
  I_LL212,I_LL213,I_LL223,LL2):
> hleg2lower:=I_leg2lower &o omega_leg2lower:

```

6.1.6 Inertial forces and torques

The inertial forces and torques are transferred into partitioned vectors.

```

> ptbody      :=map(Esimplify,subs(kde_new,cdft(pbody,N))):
> ptleg1upper:=map(Esimplify,subs(kde_new,cdft(pleg1upper,N))):
> ptleg2upper:=map(Esimplify,subs(kde_new,cdft(pleg2upper,N))):
> ptleg1lower:=map(Esimplify,subs(kde_new,cdft(pleg1lower,N))):
> ptleg2lower:=map(Esimplify,subs(kde_new,cdft(pleg2lower,N))):

> htbody:=mkc( &simp subs(kde_new,N &fdt hbody) ):
> htleg1upper:=mkc( &simp subs(kde_new,N &fdt hleg1upper) ):
> htleg2upper:=mkc( &simp subs(kde_new,N &fdt hleg2upper) ):
> htleg1lower:=mkc( &simp subs(kde_new,N &fdt hleg1lower) ):
> htleg2lower:=mkc( &simp subs(kde_new,N &fdt hleg2lower) ):

```

6.1.7 Velocity K-vector and tangent vector

```

> vK:=[ v_body_cm, vleg1upper_cm, vleg2upper_cm,
  vleg1lower_cm, vleg2lower_cm, mkc(omega_body),
  mkc(omega_leg1upper), mkc(omega_leg2upper),
  mkc(omega_leg1lower), mkc(omega_leg2lower)]

> tauK:=Kctau(vK,[seq(u.i,i=1..10)]):

```

6.1.8 Toe forces and torques

Maple procedure for extracting the variables that an expression depends on, given a list of possible variables:

```

> check_args:=proc(expr,args)
> local temp_args,i;
> temp_args:=[]:
> for i in args do

```

```

> if depends(expr,i) then temp_args:=[op(temp_args),i]; fi;
> od;
> RETURN(op(temp_args));
> end:

> args:=seq(q.i,i=1..10),seq(u.i,i=1..10);

```

6.1.9 First foot first toe

The position and velocity for the first toe point of the first foot:

```

> rtoe11:=&simp (N &to (rbody_hip_center &+ rleg1_jointpos &+
  rleg1_upper_jointpos_rel &+ (LL1 &ev [xt11,yt11,zt11])) ):
> vtoe11:=&simp (N &to subs(kde_new,N &fdt rtoe11)):

```

Find the variables that the above expressions depends on. Since the toe position is given relative to the hip point T , the expressions are subtracted with q_1 and q_3 in the appropriate directions.

```

> x11_args:=check_args((rtoe11 &c 1)-q1,[args]):
> y11_args:=check_args(rtoe11 &c 2,[args]):
> z11_args:=check_args((rtoe11 &c 3)-q3,[args]):

> vx11_args:=check_args(vtoe11 &c 1,[args]):
> vy11_args:=check_args(vtoe11 &c 2,[args]):
> vz11_args:=check_args(vtoe11 &c 3,[args]):

```

The torque acting on the lower leg is derived:

```

> toe11_cm111_cross_f:=&simp ((N &to ((LL1 &ev [xt11,yt11,zt11])
  &-- (LL1 &ev [r1111,r1112,r1113]))) &xx
  (N &ev [toefx(stat11,x11(x11_args),vx11(vx11_args),kX,dX,p1),
  toefy(stat11,y11(y11_args),vy11(vy11_args),kY,dY),
  toefz(stat11,z11(z11_args),vz11(vz11_args),kZ,dZ,p2)])):

```

Common sub expressions used later in force calculations:

```

> toe11_list:=[
  x11=(rtoe11 &c 1)-q1,
  y11=rtoe11 &c 2,
  z11=(rtoe11 &c 3)-q3,

  vx11=vtoe11 &c 1,
  vy11=vtoe11 &c 2,
  vz11=vtoe11 &c 3,

  fx11=toefx(stat11,x11(x11_args),vx11(vx11_args),kX,dX,p1),
  fy11=toefy(stat11,y11(y11_args),vy11(vy11_args),kY,dY),
  fz11=toefz(stat11,z11(z11_args),vz11(vz11_args),kZ,dZ,p2),

  tx11=toe11_cm111_cross_f &c 1,
  ty11=toe11_cm111_cross_f &c 2,

```

3D Passive walkers: Implementing code

```
tz11=toe11_cm111_cross_f &c 3
]:
```

The VE substitution list associated with this toe is created:

```
> dset11:=[
  D[2](toefx)(stat11,x11(x11_args),vx11(vx11_args),kX,dX,p1)=
    toefx_x(stat11,x11,vx11,kX,dX,p1),
  D[3](toefx)(stat11,x11(x11_args),vx11(vx11_args),kX,dX,p1)=
    toefx_vx(stat11,x11,vx11,kX,dX,p1),
  D[2](toefy)(stat11,y11(y11_args),vy11(vy11_args),kY,dY)=
    toefy_y(stat11,y11,vy11,kY,dY),
  D[3](toefy)(stat11,y11(y11_args),vy11(vy11_args),kY,dY)=
    toefy_vy(stat11,y11,vy11,kY,dY),
  D[2](toefz)(stat11,z11(z11_args),vz11(vz11_args),kZ,dZ,p2)=
    toefz_z(stat11,z11,vz11,kZ,dZ,p2),
  D[3](toefz)(stat11,z11(z11_args),vz11(vz11_args),kZ,dZ,p2)=
    toefz_vz(stat11,z11,vz11,kZ,dZ,p2),
  D[6](toefx)(stat11,x11(x11_args),vx11(vx11_args),kX,dX,p1)=
    toefx_p(stat11,x11,vx11,kX,dX,p1),
  D[6](toefz)(stat11,z11(z11_args),vz11(vz11_args),kZ,dZ,p2)=
    toefz_p(stat11,z11,vz11,kZ,dZ,p2),

  seq(diff(x11(x11_args),i)=dx11d.i,i=x11_args),
  seq(diff(vx11(vx11_args),i)=dvx11d.i,i=vx11_args),
  x11(x11_args)=x11,
  vx11(vx11_args)=vx11,

  seq(diff(y11(y11_args),i)=dy11d.i,i=y11_args),
  seq(diff(vy11(vy11_args),i)=dvy11d.i,i=vy11_args),
  y11(y11_args)=y11,
  vy11(vy11_args)=vy11,

  seq(diff(z11(z11_args),i)=dz11d.i,i=z11_args),
  seq(diff(vz11(vz11_args),i)=dvz11d.i,i=vz11_args),
  z11(z11_args)=z11,
  vz11(vz11_args)=vz11
]:
```

6.1.10 First foot second toe

Instead of repeating all the above with parameters relating to the second toe, a reuse of the above list is done and substitutions are made where necessary. The toe point dependence on state variables is the same as the first toe.

```
> x12_args:=x11_args:
> y12_args:=y11_args:
> z12_args:=z11_args:
```

```

> vx12_args:=vx11_args:
> vy12_args:=vy11_args:
> vz12_args:=vz11_args:

```

Substitution list:

```

> from_toe11_to_toe12_sset:={
  xt11=xt12,yt11=yt12,zt11=zt12,
  rl111=r1111,r1112=r1112,r1113=r1113,
  toefx(stat11,x11(x11_args))=toefx(stat12,x12(x12_args)),
  toefy(stat11,y11(y11_args))=toefy(stat12,y12(y12_args)),
  toefz(stat11,z11(z11_args))=toefz(stat12,z12(z12_args)),
  x11=x12,vx11=vx12,
  y11=y12,vy11=vy12,
  z11=z12,vz11=vz12,
  stat11=stat12,
  p1=p3,p2=p4,
  seq(dx11d.i=dx12d.i,i=x12_args),
  seq(dvx11d.i=dvx12d.i,i=vx12_args),
  seq(dy11d.i=dy12d.i,i=y12_args),
  seq(dvy11d.i=dvy12d.i,i=vy12_args),
  seq(dz11d.i=dz12d.i,i=z12_args),
  seq(dvz11d.i=dvz12d.i,i=vz12_args),
  fx11=fx12,fy11=fy12,fz11=fz12,
  tx11=tx12,ty11=ty12,tz11=tz12
}:

```

This list is now substituted into the position and velocity expression, as well as torque and VE substitution list:

```

> rtoe12:=subs(from_toe11_to_toe12_sset,rtoe11):
> vtoe12:=subs(from_toe11_to_toe12_sset,vtoe11):
> toe12_cm111_cross_f:=subs(from_toe11_to_toe12_sset,
  toe11_cm111_cross_f):
> toe12_list:=subs(from_toe11_to_toe12_sset,toe11_list):
> dset12:=subs(from_toe11_to_toe12_sset,dset11):

```

6.1.11 Second foot first toe

The toe points of this leg is functions of the angles q_9 and q_{10} , thus a state variable substitution has to take place together with the parameter substitution.

```

> from_leg1_to_leg2:={q7=q9,q8=q10,u7=u9,u8=u10}:
> x21_args:=seq(subs(from_leg1_to_leg2,[x11_args])[i],
  i=1..nops([x11_args])):
> y21_args:=seq(subs(from_leg1_to_leg2,[y11_args])[i],
  i=1..nops([y11_args])):
> z21_args:=seq(subs(from_leg1_to_leg2,[z11_args])[i],
  i=1..nops([z11_args])):
> vx21_args:=seq(subs(from_leg1_to_leg2,[vx11_args])[i],
  i=1..nops([vx11_args])):

```

3D Passive walkers: Implementing code

```

> vy21_args:=seq(subs(from_leg1_to_leg2,[vy11_args])[i],
  i=1..nops([vy11_args])):
> vz21_args:=seq(subs(from_leg1_to_leg2,[vz11_args])[i],
  i=1..nops([vz11_args])):

> from_toe11_to_toe21_sset:={
  xt11=xt21,yt11=yt21,zt11=zt21,
  r1111=r1211,r1112=r1212,r1113=r1213,
  r111=r121,
  l1u=l2u,
  toefx(stat11,x11(x11_args))=toefx(stat21,x21(x21_args)),
  toefy(stat11,y11(y11_args))=toefy(stat21,y21(y21_args)),
  toefz(stat11,z11(z11_args))=toefz(stat21,z21(z21_args)),
  x11=x21,vx11=vx21,
  y11=y21,vy11=vy21
  z11=z21,vz11=vz21,
  stat11=stat21,
  p1=p5,p2=p6,
  seq(dx11d.i=dx21d.i,i=x21_args),
  seq(dvx11d.i=dvx21d.i,i=vx21_args),
  dx11dq7=dx21dq9,dx11dq8=dx21dq10,
  dvx11dq7=dvx21dq9,dvx11dq8=dvx21dq10,
  dx11du7=dx21du9,dx11du8=dx21du10,
  dvx11du7=dvx21du9,dvx11du8=dvx21du10,
  seq(dy11d.i=dy21d.i,i=y21_args),
  seq(dvy11d.i=dvy21d.i,i=vy21_args),
  dy11dq7=dy21dq9,dy11dq8=dy21dq10,
  dvx11dq7=dvy21dq9,dvy11dq8=dvy21dq10,
  dy11du7=dy21du9,dy11du8=dy21du10,
  dvx11du7=dvy21du9,dvy11du8=dvy21du10,
  seq(dz11d.i=dz21d.i,i=z21_args),
  seq(dvz11d.i=dvz21d.i,i=vz21_args),
  dz11dq7=dz21dq9,dz11dq8=dz21dq10,
  dvz11dq7=dvz21dq9,dvz11dq8=dvz21dq10,
  dz11du7=dz21du9,dz11du8=dz21du10,
  dvz11du7=dvz21du9,dvz11du8=dvz21du10,
  fx11=fx21,fy11=fy21,fz11=fz21,
  tx11=tx21,ty11=ty21,tz11=tz21
}:

> rt21:=subs(from_leg1_to_leg2,
  subs(from_toe11_to_toe21_sset,rt21)):
> vt21:=subs(from_leg1_to_leg2,
  subs(from_toe11_to_toe21_sset,vt21)):
> toe21_cm112_cross_f:=subs(from_leg1_to_leg2,
  subs(from_toe11_to_toe21_sset,toe11_cm111_cross_f)):

```



```

> toe21_list:=subs(from_leg1_to_leg2,
  subs(from_toe11_to_toe21_sset,toe11_list)):
> dset21:=subs(from_leg1_to_leg2,
  subs(from_toe11_to_toe21_sset,dset11)):

```

6.1.12 Second foot second toe

See first foot second toe for an explanation.

```

> x22_args:=x21_args:
> y22_args:=y21_args:
> z22_args:=z21_args:
> vx22_args:=vx21_args:
> vy22_args:=vy21_args:
> vz22_args:=vz21_args:

> from_toe21_to_toe22_sset:={
  xt21=xt22,yt21=yt22,zt21=zt22,
  r1211=r1211,r2112=r1212,r1213=r1213,
  toefx(stat21,x21(x21_args))=toefx(stat22,x22(x22_args)),
  toefy(stat21,y21(y21_args))=toefy(stat22,y22(y22_args)),
  toefz(stat21,z21(z21_args))=toefz(stat22,z22(z22_args)),
  x21=x22,vx21=vx22,
  y21=y22,vy21=vy22,
  z21=z22,vz21=vz22,
  stat21=stat22,
  p5=p7,p6=p8,
  seq(dx21d.i=dx22d.i,i=x22_args),
  seq(dvx21d.i=dvx22d.i,i=vx22_args),
  seq(dy21d.i=dy22d.i,i=y22_args),
  seq(dvy21d.i=dvy22d.i,i=vy22_args),
  seq(dz21d.i=dz22d.i,i=z22_args),
  seq(dvz21d.i=dvz22d.i,i=vz22_args),
  fx21=fx22,fy21=fy22,fz21=fz22,
  tx21=tx22,ty21=ty22,tz21=tz22
}:

> rtoe22:=subs(from_toe21_to_toe22_sset,rtoe21):
> vtoe22:=subs(from_toe21_to_toe22_sset,vtoe21):
> toe22_cm112_cross_f:=subs(from_toe21_to_toe22_sset,
  toe21_cm112_cross_f):
> toe22_list:=subs(from_toe21_to_toe22_sset,toe21_list):
> dset22:=subs(from_toe21_to_toe22_sset,dset21):

```

6.1.13 Knee

Knee torque functions used for both legs and the associated VE substitution list:

```

> knee1_list:=[Tleg11=knee(statk1,q8,u8,knee_k,knee_d,knee_d_reb)]:

```

3D Passive walkers: Implementing code

```
> knee2_list:=[Tleg2l=
    knee(statk2,q10,u10,knee_k,knee_d,knee_d_reb)]:

> dset_knees:=[
    diff(knee(statk1,q8,u8,knee_k,knee_d,knee_d_reb),q8)=
    knee_q(statk1,q8,u8,knee_k,knee_d,knee_d_reb),

    diff(knee(statk1,q8,u8,knee_k,knee_d,knee_d_reb),u8)=
    knee_u(statk1,q8,u8,knee_k,knee_d,knee_d_reb),

    diff(knee(statk2,q10,u10,knee_k,knee_d,knee_d_reb),q10)=
    knee_q(statk2,q10,u10,knee_k,knee_d,knee_d_reb),

    diff(knee(statk2,q10,u10,knee_k,knee_d,knee_d_reb),u10)=
    knee_u(statk2,q10,u10,knee_k,knee_d,knee_d_reb)
]:
```

6.1.14 Hip

Hip torque function used for both legs and the associated VE substitution list:

```
> hip1_list:=[Tleg1=hip(q4,q7,u4,u7,hip_k,hip_delta,hip_d)]:
> hip2_list:=[Tleg2=hip(q4,q9,u4,u9,hip_k,hip_delta,hip_d)]:

> dset_hips:=[
    diff(hip(q4,q7,u4,u7,hip_k,hip_delta,hip_d),q4) =
    hip_q_1(q4,q7,u4,u7,hip_k,hip_delta,hip_d),

    diff(hip(q4,q7,u4,u7,hip_k,hip_delta,hip_d),q7) =
    hip_q_2(q4,q7,u4,u7,hip_k,hip_delta,hip_d),

    diff(hip(q4,q7,u4,u7,hip_k,hip_delta,hip_d),u4) =
    hip_u_1(q4,q7,u4,u7,hip_k,hip_delta,hip_d),

    diff(hip(q4,q7,u4,u7,hip_k,hip_delta,hip_d),u7) =
    hip_u_2(q4,q7,u4,u7,hip_k,hip_delta,hip_d),

    diff(hip(q4,q9,u4,u9,hip_k,hip_delta,hip_d),q4) =
    hip_q_1(q4,q9,u4,u9,hip_k,hip_delta,hip_d),

    diff(hip(q4,q9,u4,u9,hip_k,hip_delta,hip_d),q9) =
    hip_q_2(q4,q9,u4,u9,hip_k,hip_delta,hip_d),

    diff(hip(q4,q9,u4,u9,hip_k,hip_delta,hip_d),u4) =
    hip_u_1(q4,q9,u4,u9,hip_k,hip_delta,hip_d),

    diff(hip(q4,q9,u4,u9,hip_k,hip_delta,hip_d),u9) =
    hip_u_2(q4,q9,u4,u9,hip_k,hip_delta,hip_d)
```

]:

6.1.15 Forces and torques

These are the forces and torques applied to the walker:

```
> Fbody:=N &ev [0, -m_body*g*cos(theta), -m_body*g*sin(theta)]:
> Tbody:=V &ev [Tleg1+Tleg2,0,0]:

> Fleg1upper:=N &ev [0,-ml1u*g*cos(theta),-ml1u*g*sin(theta)]:
> Tleg1upper:=V &ev [-Tleg1+Tleg1l,0,0]:

> Fleg2upper:=N &ev [0,-ml2u*g*cos(theta),-ml2u*g*sin(theta)]:
> Tleg2upper:=V &ev [-Tleg2+Tleg2l,0,0]:

> Fleg1lower:=N &ev [fx11+fx12,fy11+fy12-ml1l*g*cos(theta),
    fz11+fz12-ml1l*g*sin(theta)]:

> Tleg1lower:=(UL1 &ev [-Tleg1l,0,0]) &++
    (N &ev [tx11+tx12,ty11+ty12,tz11+tz12]):

> Fleg2lower:=N &ev [fx21+fx22,fy21+fy22-ml2l*g*cos(theta),
    fz21+fz22-ml2l*g*sin(theta)]:
> Tleg2lower:=(UL2 &ev [-Tleg2l,0,0]) &++
    (N &ev [tx21+tx22,ty21+ty22,tz21+tz22]):
```

6.1.16 Kane's method

Assemble the inertial forces and applied forces into K-vectors of partitioned vectors:

```
> pKt:=[ptbody, ptleg1upper, ptleg2upper, ptleg1lower, ptleg2lower,
    htbody, htleg1upper, htleg2upper, htleg1lower, htleg2lower]:

> RKt:=[mkc(Fbody),mkc(Fleg1upper),mkc(Fleg2upper),mkc(Fleg1lower),
    mkc(Fleg2lower), mkc(Tbody),mkc(Tleg1upper),mkc(Tleg2upper),
    mkc(Tleg1lower),mkc(Tleg2lower)]:

> kane_eq:=map(simplify,ckane(tauK,pKt,RKt)):
```

6.1.17 Matlab export

The common sub expressions are assembled:

```
> cse:=[op(toe11_list),op(toe12_list),
    op(toe21_list),op(toe22_list),op(knee1_list),op(knee2_list),
    op(hip1_list),op(hip2_list)]:
```

The VE substitution list is assembled:

```
> dset:=[op(dset11),op(dset12),op(dset21),op(dset22),
    op(dset_knees),op(dset_hips)]:
```

3D Passive walkers: Implementing code

```
> uts := [u1t,u2t,u3t,u4t,u5t,u6t,u7t,u8t,u9t,u10t]:

> kde_new_ordered:= [seq(q.i.t=subs(kde_new,q.i.t),i=1..10),
    seq(p.i.t=subs(kde_new,p.i.t),i=1..8)]:

> qts := [q1t,q2t,q3t,q4t,q5t,q6t,q7t,q8t,q9t,q10t,
    p1t,p2t,p3t,p4t,p5t,p6t,p7t,p8t]:

> vars:=[q1,q2,q3,q4,q5,q6,q7,q8,q9,q10,
    p1,p2,p3,p4,p5,p6,p7,p8,u1,u2,u3,u4,u5,u6,u7,u8,u9,u10]:
```

The names of the parameters are listed. Note that space is reserved for the discrete states in the parameter list. At simulation time the discrete states are transferred into the parameter list since *Matlab*'s ODE solvers does not support discrete states

```
> para:=[
    rx,ry,rz,
    rl11,
    rl21,
    rl1u1,rl1u2,rl1u3,
    rl2u1,rl2u2,rl2u3,
    l1u,l2u,
    rl1l1,rl1l2,rl1l3,
    rl2l1,rl2l2,rl2l3,
    I_B11,I_B22,I_B33,I_B12,I_B13,I_B23,
    I_LU111,I_LU122,I_LU133,I_LU112,I_LU113,I_LU123,
    I_LU211,I_LU222,I_LU233,I_LU212,I_LU213,I_LU223,
    I_LL111,I_LL122,I_LL133,I_LL112,I_LL113,I_LL123,
    I_LL211,I_LL222,I_LL233,I_LL212,I_LL213,I_LL223,
    m_body,m11u,m12u,m11l,m12l,g,theta,
    kX,dX,kY,dY,kZ,dZ,
    knee_k,knee_d,knee_d_reb,
    hip_k,hip_delta,hip_d,
    statk1,statk2,
    xt11,yt11,zt11,stat11,
    xt12,yt12,zt12,stat12,
    xt21,yt21,zt21,stat21,
    xt22,yt22,zt22,stat22
]:
```

The list containing the event functions are expressed. The list with all elements set to one, tells *Matlab* to stop integrating when the corresponding event occurs. The list of "stats" tell *Matlab* to trigger the event on the correct flank.

```
> b:=[
    [
        [y11=rtoe11 &c 2,
```

```

    y12=rtoe12 &c 2,
    y21=rtoe21 &c 2,
    y22=rtoe22 &c 2
  ],
  [y11,y12,y21,y22,q8,q10]
],
[1,1,1,1,1,1],
[stat11,stat12,stat21,stat22,statk1,statk2]
]:

```

Include file and the path where file should be exported:

```

> inc:='includes'=['','walker_3D_external.c']':
> path:='':

```

6.1.18 Export of equations of motion

This exports the KDE and EOM to a Matlab compatible C file, which can be used directly by the ODE integrators of Matlab. Note that the VE substitution list `dset` is substituted into the common sub expression `cse` as to make them C friendly.

```

> exmex('walker_3D',path,[[op(kde_new_ordered)]],
    [kane_eq,uts],[op(subs(dset,cse))],
    [op(qts),op(uts)],vars,parameters=para,
    'v5'=['events'=b],inc);

```

6.1.19 Export of variational equations

The option `'variationaleqs'` is used to generate the variation equations. Here, the VE substitution list is given as an argument since the substitution needs to take place after `exmex` has internally generated the equations.

```

> exmex('walker_3D_stab_1',path,[[op(kde_new_ordered)]],
    [kane_eq,uts],[op(cse)], [op(qts),op(uts)],
    vars,variationaleqs,parameters=para,
    diffset=dset,'v5'=['events'=b],inc);

```

6.1.20 Helper functions

Export of functions to calculate the position of the toes and the basis vectors of each frame:

```

> toe11 := matrix(1,3,[rtoe11 &c 1, rtoe11 &c 2, rtoe11 &c 3]):
> toe12 := matrix(1,3,[rtoe12 &c 1, rtoe12 &c 2, rtoe12 &c 3]):
> toe21 := matrix(1,3,[rtoe21 &c 1, rtoe21 &c 2, rtoe21 &c 3]):
> toe22 := matrix(1,3,[rtoe22 &c 1, rtoe22 &c 2, rtoe22 &c 3]):

> exmat('pos_toe11',path,toe11,vars,para,'matlab');
> exmat('pos_toe12',path,toe12,vars,para,'matlab');
> exmat('pos_toe21',path,toe21,vars,para,'matlab');
> exmat('pos_toe22',path,toe22,vars,para,'matlab');

```

3D Passive walkers: Implementing code

```
> exmat('VtoN',path,Rmx(N,V),vars,'matlab');
> exmat('UL1toV',path,Rmx(V,UL1),vars,'matlab');
> exmat('UL2toV',path,Rmx(V,UL2),vars,'matlab');
```

Export of the derivatives of the discontinuous jump in state variables during toe impact:

```
> G11:=[seq(q.i,i=1..10),(rtoe11 &c 1)-q1,
        (rtoe11 &c 3)-q3,p3,p4,p5,p6,p7,p8,seq(u.i,i=1..10)];
> G12:=[seq(q.i,i=1..10),p1,p2,(rtoe12 &c 1)-q1,
        (rtoe12 &c 3)-q3,p5,p6,p7,p8,seq(u.i,i=1..10)];

> G21:=[seq(q.i,i=1..10),p1,p2,p3,p4,(rtoe21 &c 1)-q1,
        (rtoe21 &c 3)-q3,p7,p8,seq(u.i,i=1..10)];

> G22:=[seq(q.i,i=1..10),p1,p2,p3,p4,p5,p6,
        (rtoe22 &c 1)-q1,(rtoe22 &c 3)-q3,seq(u.i,i=1..10)];

> qus:=[seq(q.i,i=1..10),seq(p.i,i=1..8),seq(u.i,i=1..10)];

> Gx11 := matrix(28,28,[]);
> Gx12 := matrix(28,28,[]);
> Gx21 := matrix(28,28,[]);
> Gx22 := matrix(28,28,[]);
>
> for i from 1 to 28 do
>   for j from 1 to 28 do
>     Gx11[i,j]:=diff(G11[i], qus[j]):
>     Gx12[i,j]:=diff(G12[i], qus[j]):
>     Gx21[i,j]:=diff(G21[i], qus[j]):
>     Gx22[i,j]:=diff(G22[i], qus[j]):
>   od:
> od;

> exmat('Gx11',path,Gx11,qus,para,'matlab');
> exmat('Gx12',path,Gx12,qus,para,'matlab');
> exmat('Gx21',path,Gx21,qus,para,'matlab');
> exmat('Gx22',path,Gx22,qus,para,'matlab');
```

Derivatives of the discontinuity surfaces associated with toe impacts:

```
> qus:=[seq(q.i,i=1..10),seq(p.i,i=1..8),seq(u.i,i=1..10)];

> mxtoe11 := matrix(28,1,[]);
> mxtoe12 := matrix(28,1,[]);
> mxtoe21 := matrix(28,1,[]);
> mxtoe22 := matrix(28,1,[]);

> i:=1;
```

```
> for ii in qus do
>   mxtoe11[i,1]:=diff(rtoe11 &c 2, ii);
>   mxtoe12[i,1]:=diff(rtoe12 &c 2, ii);
>   mxtoe21[i,1]:=diff(rtoe21 &c 2, ii);
>   mxtoe22[i,1]:=diff(rtoe22 &c 2, ii);
> i:=i+1;
> od:

> exmat('Ntoe11',path,mxtoe11,qus,para,'matlab');
> exmat('Ntoe12',path,mxtoe12,qus,para,'matlab');
> exmat('Ntoe21',path,mxtoe21,qus,para,'matlab');
> exmat('Ntoe22',path,mxtoe22,qus,para,'matlab');
```

Bibliography

- [1] Adolfsson, J.: 2000a, ‘Finding an Implementable 3D Passive Walker using Continuation Methods’. Technical report, KTH, Department of Mechanics.
- [2] Adolfsson, J.: 2000b, ‘Planar Passive Walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [3] Adolfsson, J., H. Dankowicz, and A. Nordmark: 2001, ‘3D Passive Walkers: Finding Periodic Gaits in the Presence of Discontinuities’. *Nonlinear Dynamics* **24**(2), 205–229.
- [4] Coleman, M. and A. Ruina: 1998, ‘An Uncontrolled Toy That Can Walk But Cannot Stand Still’. *Physical Review Letters* **80**(16), 3658–3661.
- [5] Drugge, L.: 2000, ‘Modelling and Simulation of Pantograph-Catenary Dynamics’. Ph.D. thesis, LTU, Department of Mechanical Engineering, Division of Computer Aided Design.
- [6] Kane and Levinson: 1985, *Dynamics: Theory and Applications*. McGraw-Hill.
- [7] Kuo, A. D.: 1999, ‘Stabilization of Lateral Motion in Passive Dynamic Walking’. *The International Journal of Robotics Research* **18**(9), 917–930.
- [8] Lennartsson, A.: 1999, ‘Efficient Multibody Dynamics’. Ph.D. thesis, KTH, Department of Mechanics.
- [9] Lesser, M.: 1995, *The Analysis of Complex Nonlinear Mechanical Systems*. World Scientific Publishing Co. Pte. Ltd.

- [10] McGeer, T.: 1990, ‘Passive Bipedal Running’. In: *Proceedings of the Royal Society of London: Biological Sciences*, Vol. 240. pp. 107–134.
- [11] McGeer, T.: 1993, ‘Dynamics and Control of Bipedal Locomotion’. *Journal of Theoretical Biology* **163**, 277–314.
- [12] Mombaur, K., M. J. Coleman, M. Garcia, and A. Ruina: 2000, ‘Prediction of Stable Walking for a Toy That Cannot Stand’. submitted for publication in PRL.
- [13] Nichols, G. K. and D. C. Witt, ‘An Experimental Unpowered Walking Aid’. *Engineering in Medicine* **1**(1).
- [14] Olsson, A.: 1997, ‘Object-oriented finite element algorithms’. Technical report, KTH, Department of Structural Engineering.
- [15] Piiroinen, P.: 2000, ‘Passive Walking: Transition from 2D to 3D’. Technical report, KTH, Department of Mechanics.
- [16] Rose, J. and J. G. Gamble (eds.): 1994, *Human Walking*. Williams & Wilkins.
- [17] Vågstedt, N.-G.: 1995, ‘On Cornering Characteristics of Ground Vehicle Axles’. Ph.D. thesis, KTH, Department of Vehicle Engineering.
- [18] Wisse, M., A. L. Schwab, and R. Q. vd Linde: 2000, ‘A 3D Passive Dynamic Biped with Yaw and Roll Compensation’. to appear in ROBOTICA.

Paper 8

P8

Finding an implementable 3D passive walker using continuation methods

Jesper Adolfsson
Royal Institute of Technology
Department of Mechanics
S-100 44 STOCKHOLM
SWEDEN

October, 2000

ABSTRACT. This text describes some of the parameter searches made to find an implementable 3D passive walker. The starting points of the parameter search are previous found 3D walkers, see for example Adolfsson *et. al.* [3]. However, previous found walkers had a number of features not suitable for implementation, which are believed to be corrected by the presented here walker. The parameter search was accomplished by using continuation methods, which makes it possible to locate periodic gaits as parameters of the system are varied, see Adolfsson *et. al.* [3] and Adolfsson [1]. To be a realizable walker it is important that reasonable margins exists in both parameter space and initial conditions. Thus, a large part of this text is devoted to investigating the influence of parameters that are difficult to control in an experimental environment. Also reported here are attempts to develop an experimental 3D walker.

1 Introduction

The 3D walker consists of five bodies, an upper torso, two upper legs, and two lower legs, see Figure 1. The walker makes regular contact with the ground through its four toe points, two on each lower leg. During gait the walker can be seen as a constrained double pendulum (swing leg) attached to an inverted pendulum (stance leg). For a detailed description of the geometry, mass distribution, and forces acting on the 3D passive walker see Adolfsson *et al.* [3] and Adolfsson [1]. To counteract the energy lost in knee impacts and toe impacts, the walker descends an incline, thus gravity acts as the source of

energy. Studies of related passive 3D mechanisms have been done by Mombaur *et al.* [14], Wisse *et al.* [16], and Coleman *et al.* [5].

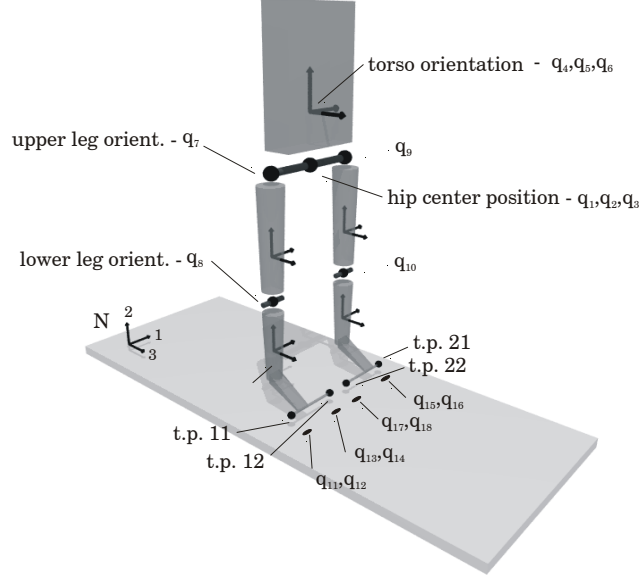


Figure 1. The geometry of the 3D walker with state variables. Note that the angle of the upper legs are not given relative to the torso rotation q_4 about the hip axis. Abbreviations: t.p. stands for toe point and i.c.p. stands for initial contact point.

The main goal of the parameter search was to find an implementable passive 3D walker, having dynamics in all spatial directions. This was accomplished by using continuation methods, where periodic gaits could be followed while varying the parameters of the system. In some cases, direct numerical simulation was also used. The continuation algorithm uses the Newton-Raphson method to locate periodic gaits of the 3D passive walker. Periodic orbits of the dynamical system correspond to periodic gaits of the 3D passive walker. For a detailed description of the methods used and the implementation of the methods, see Adolfsson *et al.* [3] and Adolfsson [1].

The 3D passive walker is described by 28 state variables and approximately 80 parameters. If the left and right side of the walker is mirrored in the sagittal plane, see Figure 2 for a definition of planes, the number of parameters will be reduced to about 40. The sheer number of parameters makes it hard to draw

any decisive conclusions for a particular parameter variation. Typically, the conclusion drawn from a one-dimensional parameter variation might not hold if one changes the value of another parameter. Therefore, the results presented here should be regarded as suggestions or insights, which might or might not work under different parameter conditions. This emphasizes the importance of having a very clear goal when performing parameter studies, such as finding an implementable 3D passive walker.

The parameter searches are presented in two different ways, bifurcation diagrams and goal-oriented diagrams. Bifurcation diagrams are the classical dynamical systems approach where new solutions, born in bifurcations, are followed and classified. The other type of parameter search have a specific goal in mind, such as increasing the mass of the torso. This is more of an engineering approach to the problem. However, having decided a particular parameter path to obtain a goal, it might turn out that no periodic solutions exist at some point along the path; therefore a new path in parameter space might be needed. For example, the toe points might hit the floor (foot scuffing) during the swing phase as parameters are varied. This usually results in non existence of periodic solutions, see Piiroinen [15] for a study where the periodic solution survives foot scuffing. Sometimes a whole new strategy to the problem might be required, such as introducing new forces into the model.

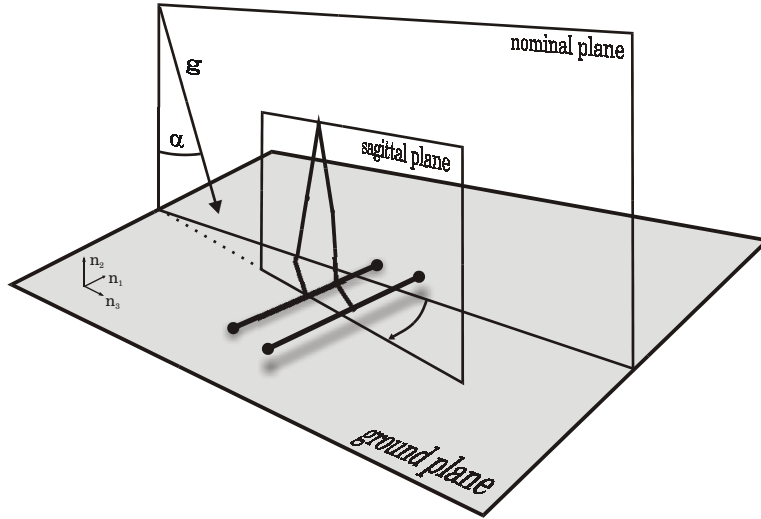


Figure 2. Definition of planes. The walker makes regular contact with the ground plane. The nominal plane is defined by the initial lateral position of the hip center and spanned by the $\mathbf{n}_2$ and $\mathbf{n}_3$ direction. Thus, a walker heading straight down the plane will start and end in the nominal plane. The sagittal plane is defined by the hip center position and the up and forward directions of the walker. Symetric walkers have legs that are mirrored in this plane. Gravity is inclined an angle α .

1.1 GOAL

As previously mentioned, a clearly defined goal is needed, which in this study is to find an implementable passive 3D walker. Below are the requirements listed for an implementable 3D passive walker.

- The mass distribution should be realizable, thus masses should not have to be placed in awkward positions, such as on long extended booms, to get the required moment of inertia.
- There should be an anthropomorphic (human like) look of the 3D passive walker. This means that geometrical measures of the experimental walker should have similar ratios as a human. For example, the ratio of leg length to hip width should be close to the values found in humans.
- No overlap of the feet.
- Insensitive to parameter changes. This is especially important when the parameters are hard to control or measure in an experimental walker. For example, the actual force between ground and toe points might be very hard to model. Thus, a stable gait should exist for reasonable intervals in the parameters associated with the ground.
- Initial conditions that is possible to realize and have a reasonable basin of attraction. The latter might be hard to investigate due to the number of states of the walker.
- Be able to carry some payload, however not necessarily with an upright torso, which most likely would require active control.
- Knee mechanism should be possible to implement passively. The current design requires quite large angular motions of the lower legs during the locked state of the knees.

During the search to fulfill the above requirements, many steps were made so as to investigate discovered phenomena. This included investigating the effects of letting the toe points being below the ground plane at certain times. Some of these parameter studies didn't directly help to reach the goal but gave valuable insights into the dynamics of the system.

The 3D passive walkers studied originate in parameters from a planar passive walker studied by Garcia *et al.* [9], [8], [7], and [4]. Also see Adolfsson *et al.* [2] for a description of the planar passive walker and how to implement code for simulation of planar passive walking. Since the parameters originate from a planar walker, the 3D passive walker was started in a configuration that mimics the dynamics of the planar passive walker. This configuration is similar to the walker depicted in Figure 2. The first parameter variations were made using direct numerical simulation, thus only stable periodic gaits were accessible. These parameter variations included separating the legs on the hip axis and decreasing the overlap in the feet. However, the need for faster convergence of found stable gaits and the need to access unstable periodic gaits led to the development of a root finder, see Adolfsson *et al.* [3]. The implemented

root finder uses the Newton-Raphson scheme, which will converge to the periodic solution, given an initial state reasonable close to the periodic solution. The stable periodic solutions are of main interest due to the possibility of being implemented passively. However, by following unstable periodic solutions, the solution might bifurcate into a stable solution. It has also been argued that unstable solutions might require less active control to stabilize the motion compared to artificially generating the walking pattern.

1.2 INTERPRETING NUMERICAL RESULTS

Automating the search using a root finder brings up the important question of whether the obtained numerical results can be understood by some simpler reasoning. For example, it will later be shown that changing the position of the center of mass of the legs seems to require that the moment of inertia be changed in a specific way. Could this have been understood by looking at a simple pendulum? Generally, it has been very hard to understand the results by some simpler reasoning and someone using these results should make sure that they are valid for their specific system.

The Newton-Raphson root finder scheme gives the Jacobian of the linearized system as a by-product. The stability characteristics can be obtained by calculating the eigenvalues and eigenvectors of the Jacobian. As long as all eigenvalues have a magnitude less than one, the system is stable. It has been argued that valuable insight can be obtained by looking at the elements of a particular eigenvector. For example, McGeer [10],[11],[12], and [13] give the eigenvectors in the planar passive walker illustrative names, such as speed, totter, and swing. These names are related to what they are supposed to control. These eigenvectors are given in a specific Poincaré section corresponding to the configuration having the least number of degrees of freedom. For a description of this configuration see Adolfsson *et al.* [2]. These eigenvectors might look different when viewed in a different Poincaré section. Therefore, to get the whole picture they should at least be viewed over an entire gait cycle.

To investigate if eigenvectors could give valuable insight into the dynamics of the 3D passive walker, animations of eigenvectors were performed on both stable and unstable periodic gaits. This was accomplished by adding the time evolution of a small disturbance to the periodic gait. The disturbance was in the direction of an eigenvector. This can be written as,

$$\mathbf{x}_{\text{anim}}(t) = \mathbf{x}(t) + J(t)c\mathbf{v}_i, \quad (1)$$

where $\mathbf{x}(t)$ is the found periodic solution, $J(t)$ is the Jacobian at time t , and $c\mathbf{v}_i$ is a constant times the i 'th eigenvector of the Jacobian. However, no success was made in interpreting the animation $\mathbf{x}_{\text{anim}}(t)$. The animation resulted in very strange walking motion, where legs started to oscillate back and forth together with oscillating motions of the hip. This shows, as can be deduced by looking at the elements of the eigenvector, that strong coupling of the motion takes place over all state variables. In addition, it is hard to draw conclusions

from comparing angular state variables with positional state variables in the eigenvector, because there is no apparent way of comparing them.

However, in some cases the eigenvectors can be of help. The dynamics of the 3D passive walker are indifferent to placement in the plane. Thus, two eigenvalues reflect this by being one and the only nonzero element in the corresponding eigenvector is the states that describe the position in the ground plane. In addition, when pitchfork bifurcations occur the corresponding eigenvector pair can be used to give the directions of the newborn solution branches.

2 Parameter studies

The following sections will present some of the many parameter studies that were carried out to find an implementable 3D passive walker. The order they are presented roughly follows the chronological order in which they were carried out.

2.1 GOING FROM 2D TO 3D

By taking initial conditions and parameter values from the planar passive walker, see Adolfsson *et al.* [2], it was possible to start the 3D passive walker in a configuration similar to the planar walker, see Figure 2. Ground plane stiffness was high to mimic the plastic impacts found in the planar walker. Knee stiffness was also high for the same reason. After some manual tuning of initial conditions a gait was observed, which became periodic after transients had settled down. However, an oblique walk resulted, where the hip center did not remain in the nominal plane after one stride. By using direct numerical simulation, it was possible to slowly introduce a hip distance and to shrink the foot width. In Figure 3, a walker with dynamics in all spatial directions is shown, as can be seen it also walks at an angle down the inclination. The overlap in the feet is 0.1m. Walking at an angle results in different time histories for the two legs. By looking at the horizontal line in Figure 4, this asymmetry can be seen. Since the system is symmetric about the sagittal plane, there must exist an oblique gait mirrored in the nominal plane. Later, it was discovered that by lowering the stiffness of the ground plane, the walker headed straight down the plane, thus starting and ending in the nominal plane. From a dynamical systems point of view, a super critical pitchfork bifurcation occurs when increasing the stiffness of the plane. Thus, a stable gait heading straight down the plane becomes unstable and two gaits, symmetric with respect to the nominal plane, are born.

In reference Adolfsson *et al.* [3], a path from an unconstrained 2D walker to a 3D walker having motion in all spatial directions is reported. Only gaits heading straight down the plane were considered in this parameter variation. At the end of this parameter variation, the toe points are symmetrically placed about the leg, the distance between toe points is only 2.2cm. However, stability was lost before the overlap was reduced to zero. Thus, a restart had to be made to find a passive 3D walker without an overlap. An interesting result from this

Finding an implementable 3D passive walker

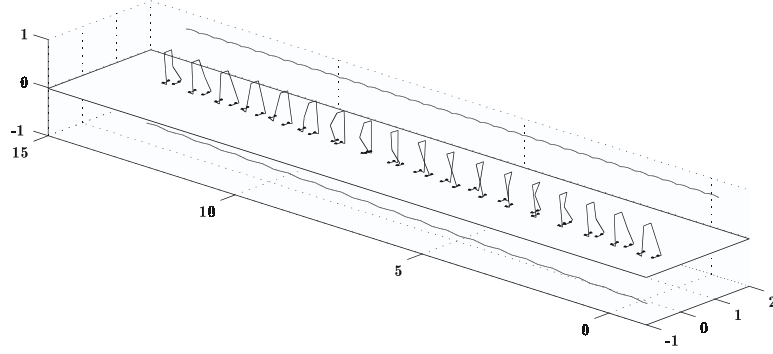


Figure 3. Walker with overlap in the feet walking obliquely down the incline.

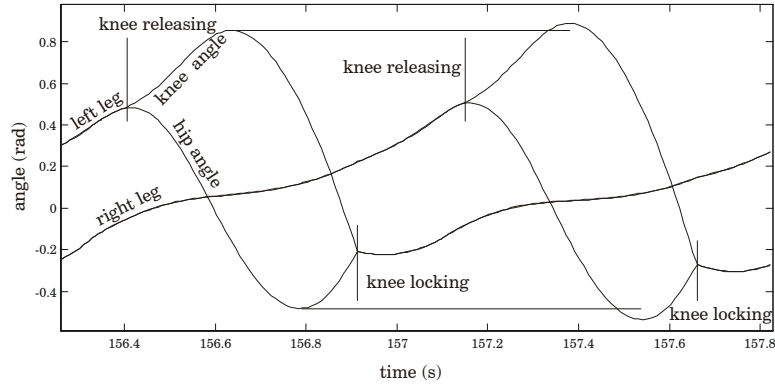


Figure 4. Oblique walking result in different time histories for the two legs. This is evident from the looking at the amplitudes of the hip and knee angles of the two legs.

parameter variation is that the swaying amplitude tends to zero, as the distance between the legs goes to zero, the toe points being placed symmetrically about the legs. In addition, this, essentially 2D symmetric walker, is here unstable. Thus, there probably exist two stable periodic gaits, mirrored in the nominal plane, walking obliquely.

2.2 KNEE MODELING IN 2D AND 3D

The switch from an impact model to a continuous model introduces new dynamics to the system. In the impact model, the knee is released after the hind leg foot has been released from the ground. Thus, the relative angular velocity of the lower leg is always zero at the beginning of the swing phase. In the continuous model, an angular spring and angular damper is active between the lower leg and upper leg, as long as the knee is in hyperextension. The result is that the continuous model allows non-zero velocities at the knee

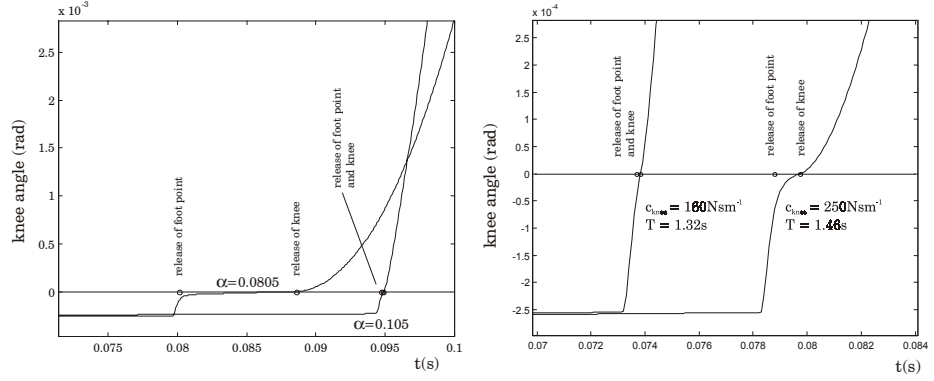


Figure 5. Left pane illustrates the difference in angular velocity at knee release for two different inclinations. The right pane illustrates the same thing, but for different knee damping. The left pane also show the knee damping effect on stride time.

release which is not possible using the impact models. This is illustrated in the left diagram of Figure 5, where the time history of the lower leg orientation, relative to the upper leg, is shown for two different inclinations. At the lower inclination, the knee release velocity is almost zero, thus mimicking the impact model. However, when the inclination of the plane is increased, the release velocity is no longer zero. In Dankowicz *et al.* [6], bifurcation diagrams are drawn for a planar passive walker using impact models and continuous models having similar mass distributions. For low inclinations, the two diagrams are fairly similar, but as the inclination is increased little resemblance is shown. A plausible explanation is the model switch, which can be seen from looking at the stride time for two different levels of knee damping. Since the inclination of the plane have a large influence on gait speed and stride time it is not wise to compare stride times in the left diagram. However, in the right diagram of Figure 5, only the knee damping is changed. The result is that at high damping the release velocity is low, whereas it is high for low damping. Looking at the stride times, there is a 10% increase between the lower and higher damping. Therefore, to mimic the knee release velocity of previous impact models, tuning of the knee stiffness and damping might be needed as other parameters of the walker are changed.

As a side note, this is similar to catching a particle with a spring and damper system. To make it behave inelastically, the spring and damping constants will depend on the mass of the caught particle.

2.3 THE SEARCH FOR A 3D WALKER WITHOUT OVERLAP

Since the initial attempt to reduce the overlap to zero resulted in a highly unstable walker, see Adolfsson *et al.* [3], a new approach was needed. This included the change of many parameters, such as trying different centers of

Finding an implementable 3D passive walker

mass locations for the legs, changing the amount of mass of various parts, and altering the ground plane characteristics. The major parameter changes from the walker reported in Adolfsson *et al.* [3] and Adolfsson *et al.* [1] are;

- Increased the hip width from 0.20m to 0.26m.
- Lowered the upper leg center of mass position from 9cm to 14cm below the hip line.
- Moved the lower leg center of mass up from 16cm to 8cm below the knee.
- Major decrease of the torso moment of inertia about all axes, from approximately 2kgm^2 to 0.3kgm^2 .
- Increased the upper leg moment of inertia from 0.008kgm^2 to 0.02kgm^2 about the axis through the upper leg center of mass, parallel to the hip and knee joint axes.
- Decreased the lower leg moment of inertia from 0.04kgm^2 to 0.012kgm^2 about an axis through the upper leg center of mass, parallel to the hip and knee joint axes.
- Increased the mass of the torso from 0.8kg to 5.0kg
- Lowered the mass of the upper legs from 2.4kg to 2.2kg
- Increased the ground plane stiffness in the vertical direction from 5000N/m to 13000N/m and damping from 500Ns/m to 770Ns/m.
- Major decrease of knee stiffness, from 500Nm/rad to 100Nm/rad and 20Nms/rad to 1Nms/rad.
- Decreased toe width from 15cm to 8.6cm and reduced the overlap from 10cm to -5.5cm (no overlap).
- Lowered the toe points 3mm and had them at slightly different height (1mm)

The motivation for lowering the moment of inertia and increasing the mass of the torso was to make it feasible to implement. Thus, it was desirable that the radius should be close to the hip width, $\sqrt{0.3/5} \approx 0.24\text{m}$ in the no overlap configuration, compared to $\sqrt{2/0.8} \approx 1.6\text{m}$ in the overlap configuration. The increase in ground stiffness was first made to make up for the increased weight of the walker. However, as the foot width was decreased it was observed that stability could be maintained by increasing the stiffness of the ground plane. The major breakthrough came unexpectedly when the knee stiffness and damping was reduced considerably. This increased stability and introduced a bounce in the knee, which can be seen in Figure 6. The sustained knee lock (hyperextension) now occurred after the toe points of the swing leg had impacted with the ground. This "new" knee dynamics once again highlight the importance of the knee modeling on the walker dynamics. The lower stiffness and damping in the knee allowed for larger relative motion between the upper leg and lower leg during knee lock. This made it feasible to use the implemented knee locking mechanism, which is described in later sections.

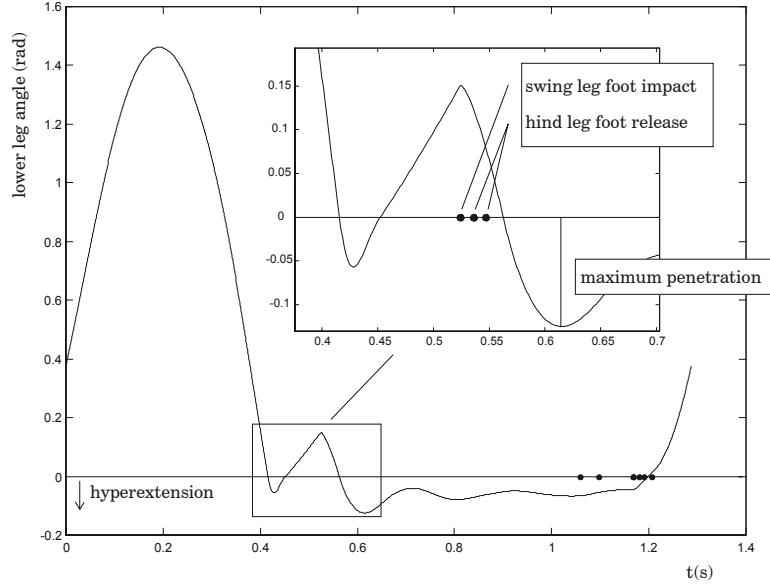


Figure 6. When lowering the knee stiffness and knee damping a bounce in the knee occurred. The enlargement show that sustained knee lock only occur after the toe points of the swing leg foot have attached to the ground plane.

During these parameter variations, it was observed that the instability occurring, when lowering the lower-legs moment of inertia, could be countered by moving the lower-legs center of mass upwards. A similar observation was made for the upper legs. When the moment of inertia of the upper legs was increased, instability occurred. This was remedied by moving the lower-leg center of mass downwards. This "rule of thumb" seems to hold and was later used to adjust mass distributions on the upper and lower legs.

In Figure 7 a root locus diagram is shown for the parameter variation starting with the no-overlap walker and ending in with the overlap walker. As seen in the diagram, both the no-overlap walker (large arrow) and overlap walker are stable. However, between the start and end, an unstable region exists. It should be noted that this particular parameter variation is not the one that was used to find the no overlap configuration, since it would have been hard to a priori know that it would end up as a stable configuration. Instead, effort was spent on gradually changing parameters and keeping stability, to find a walker satisfying the previous stated requirements.

As a reference, two extra sets of parameters are included in the root locus diagram. These points correspond to parameters "beyond" the no-overlap walker. Since one eigenvalue pair is outside the unit circle, the parameter sets are unstable. Somewhat concerning is the eigenvalues sensitivity to parameter variations for the no-overlap walkers. Potentially this will make it hard to implement the

Finding an implementable 3D passive walker

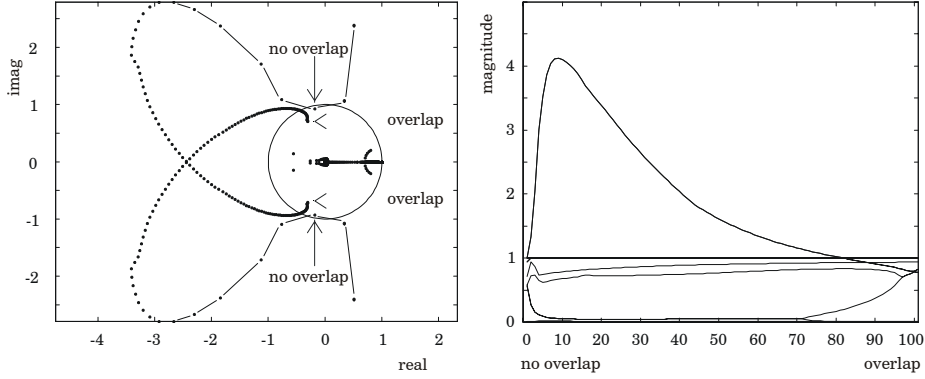


Figure 7. The right pane show a root locus for a parameter variation from a walker with an overlap in the feet to a walker with no overlap. The points are equally spaced in parameter space. Notice the high parameter sensitivity at the no-overlap point compared to the overlap point. The right pane show the the absolute value of the eigen values for the parameter variation.

walker, since tight tolerances will be needed. This sensitivity is illustrated by the large distance between the points at the no-overlap region compared to the more closely located points at the overlap region. All bifurcations occurring in this parameter variation are Hopf bifurcations. The eigenvalue having the maximum absolute value is 0.942 for the no-overlap walker and 0.937 for the overlap walker.

2.4 HIP AND CENTER OF MASS MOTION

One characteristic of the 3D passive walker is the lateral swaying motion of the hip center. This can be clearly seen in Figure 8, where the trajectories of the hip center and center of mass for all parts are drawn. Also shown are the toe contact points and the center of mass position at toe impact/release. Shortly after the toe impacts occur, the hind-leg toes release. During the short time interval between to impact and release, the toe points create a support pattern, which would seem to provide static stability, i.e. if all velocities were set to zero the walker would still stand up. However, trying this in a simulation resulted in the walker falling. Drawing a line between the toe points shows a slight toe in. Drawing a line perpendicular to the previous line, starting at the inner toe, show that the center of mass is on the inside of this line at the hind leg toe release.

2.5 FOOT SCUFFING AND BEYOND

At some point, as inclination is decreased, periodic gaits are usually born in a saddle node bifurcation. This can be seen in many references, McGeer [10],[11],[12],[13], Garcia *et al.* [9],[8],[7]. This seems to hold for both planar

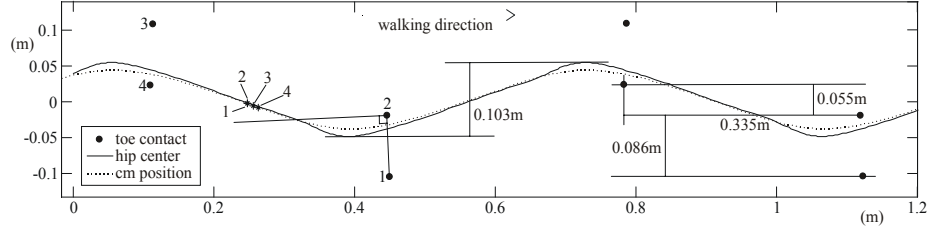


Figure 8. The trajectory of the hip center point and total center of mass for the 3D walker. The numbers correspond to toe impact/release from the ground plane. First the toes labeled 1 and 2 impact with the ground. The corresponding hip center position is also labeled 1 and 2. Secondly, the toe labeled 3 and 4 release from the ground. The corresponding hip center position is also labeled 3 and 4.

walkers and the here studied 3D passive walkers. In the reference Dankowicz *et al.* [6], bifurcation diagrams are drawn for planar walkers using impact models, planar walkers using continuous models, and 3D walkers using continuous models. All continuous models have in common that the solution is born in a saddle node bifurcation. However, most of these solutions do not end in a bifurcation as inclination is decreased, but terminate due to foot scuffing. Foot scuffing is loosely speaking a toe point impact with the ground, during swing phase, at an inappropriate time. By disregarding foot scuffing, i.e. allowing the toe point to penetrate the ground plane without applying any force, it is possible to follow the periodic gaits as inclination is further decreased. In Figure 9, a bifurcation diagram for a 3D walker with an overlap is shown. The periodic gait is born in a saddle node bifurcation at inclination 0.112 radians. As the inclination is increased, both solutions terminate in foot scuffing, (point A and E). However, the foot scuff occurs at different walker configurations in the two branches. This is illustrated in Figure 10. In the left pane, the time histories for the vertical positions of the toe points are shown for an inclination of 0.08 radians. There are two local minima for the toe points of the swing leg, labeled hind dip and front dip. As inclination is increased, either of these two local minima will touch the ground plane. In the unstable branch, the touch will occur at the front dip and in the stable branch at the hind dip. This is illustrated in the right pane of Figure 10 for the unstable branch. By disregarding this penetration, it is possible to follow the periodic gait, as inclination is further decreased. However, at point C, it is not possible to further follow the gaits due to walker not having any toe attachment point. This is illustrated in Figure 11, where the arrow indicates that the attachment point is moving below the ground plane. The next possible attachment point is 0.6s later, which is too much of a jump for the periodic gait to survive. The stable lower branch ends in a similar way, first the hind dip moves through the ground plane and finally there is no attachment point left. However, before this happens, the stable

Finding an implementable 3D passive walker

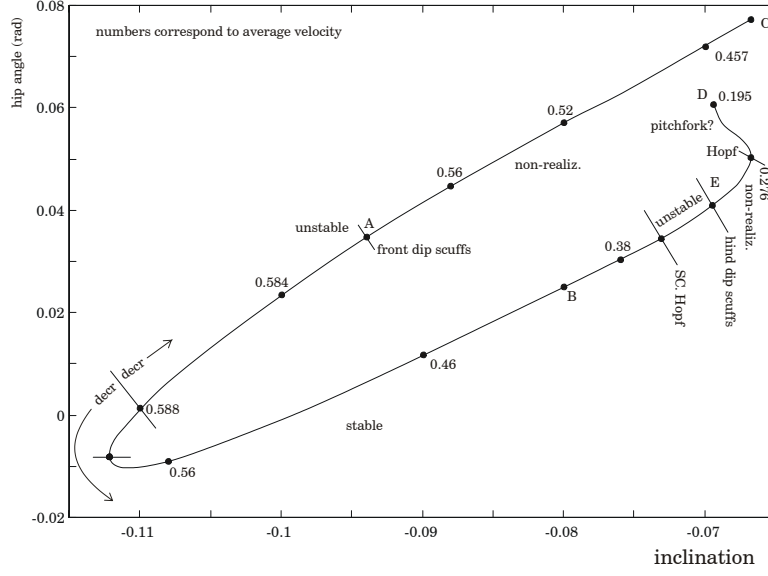


Figure 9. Bifurcation diagram for a 3D walker with an overlap in the feet when varying the inclination of the ground plane. When inclination is decreased beyond point A, in the unstable branch, and point E, in the stable branch, foot scuffing occurs. By disregarding the foot scuffing, it is possible to follow periodic gaits as inclination is further decreased. The branches terminate when it is no longer possible for the foot points to attach to the ground.

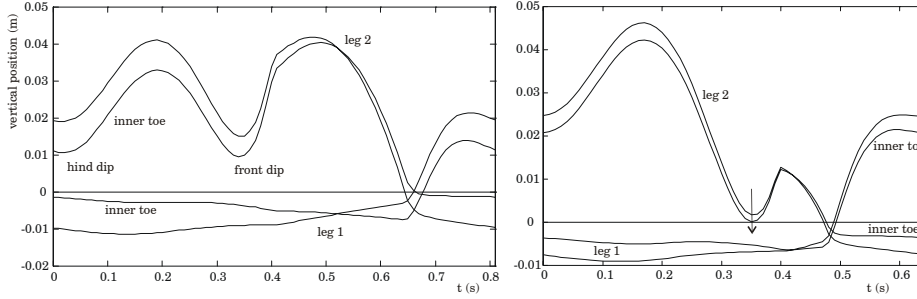


Figure 10. The vertical toe point positions as a function of time for one step. The left pane show the two characteristic dips occurring at an ordinary gait cycle. The right pane shows what happens when the inclination is decreased, one of the toe points will start touching the ground plane. By disregarding this foot scuffing, it is possible to follow gaits at further decrease of the inclination. Notice that the vertical positions are shown for one step. Leg one will perform the motion of leg two and vice versa after one step.

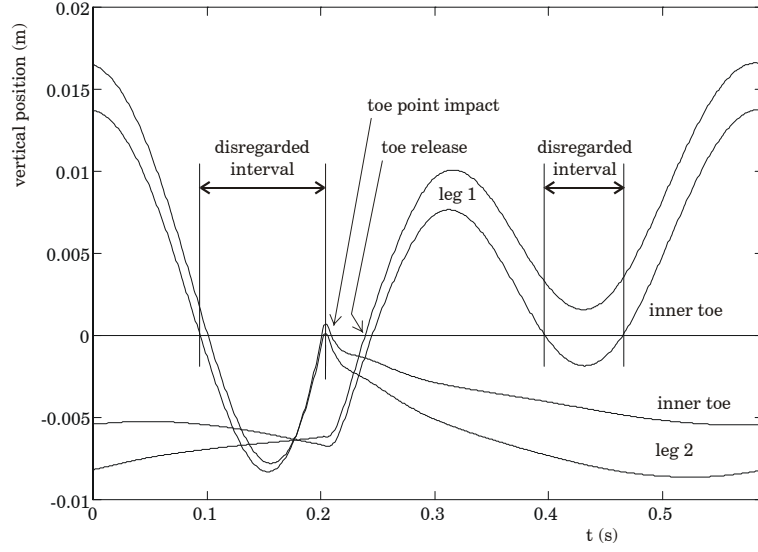


Figure 11. At some point it is not possible to disregard the scuffing since the attachment point will move through the ground plane. The next possible attachment point is too far away for the gait to survive.

branch goes through a super critical Hopf bifurcation at point E and a Hopf bifurcation at point F.

Just before the branch ends, it goes through a saddle node bifurcation. Thus, in a short inclination interval, there exists three periodic gaits heading straight down the plane. Also shown in Figure 9 is the average velocity at selected inclinations. Note that the maximum velocity does not occur at maximum inclination, but at some lower inclination in the unstable branch. This seems to generally hold for passive walkers. Also note the big difference in gait velocities at the end of the two branches (point C and D). Thus, it is very unlikely that the stable and unstable branches will ever meet.

2.6 GROUND PLANE STIFFNESS

Depending on the foot design, the actual characteristics and dynamics of the ground plane might only be estimated. Thus, it is important that the ground plane model allows for some variance in the parameters describing it. The actual design can be implemented in various ways. For example, the toe points could be fitted with soft soles, to allow some motion at the contact point. In addition, one could instead put small hard spheres at the toe points and walk on a soft ground. Both solutions are believed to work, but might, to some extent, disagree with the model used for the ground plane. These discrepancies include, non-linear effects of the material and slip stick phenomena. It is also possible to reduce uncertainties by clever design of the foot. For example,

Finding an implementable 3D passive walker

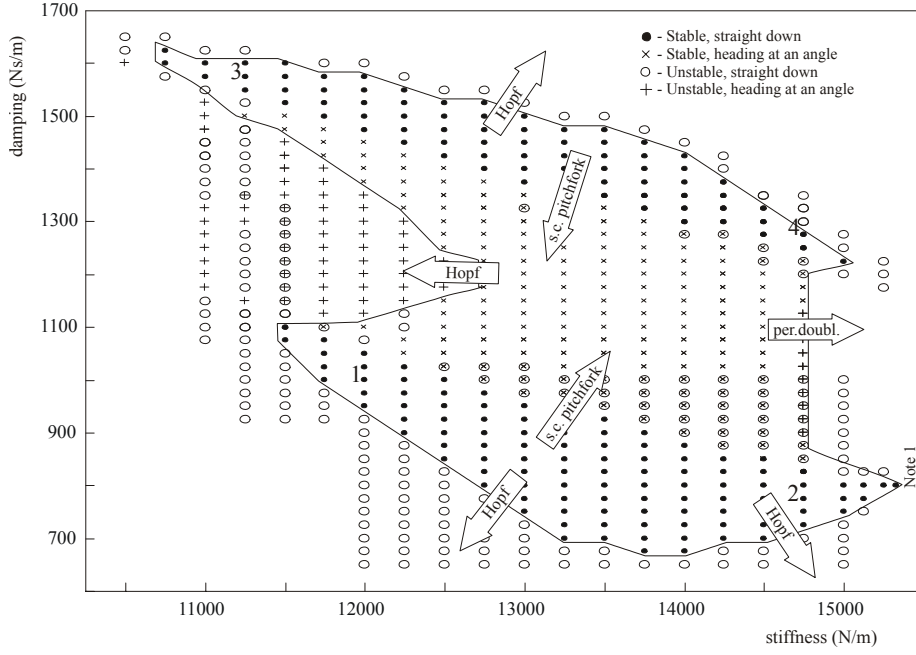


Figure 12. Parameter variation to determine the stable region in ground plane vertical stiffness and damping. Notice that some of the gaits in the middle of the region walk stable at an angle. Note 1: It was not possible to further increase the stiffness due to one of the toe points releasing from the ground plane in the middle of the stance phase.

Wisse *et al.* [16], have devised a light foot, hinged to the lower legs. Thus, the dynamics of the light foot can be disregarded and stiffness can be controlled via angular springs and dampers.

In Figure 12, the region of stable periodic gaits are determined in vertical stiffness and damping of the ground plane.

This region contains both stable gaits heading straight down the plane and oblique stable gaits. The stable region is surrounded by Hopf bifurcations. It has not been investigated whether these Hopf bifurcations are sub- or super-critical, but experience shows that super-critical Hopf bifurcations only exist in a short parameter interval close to the bifurcation point. The stable gaits heading at an angle exist between two regions of stable gaits heading straight down the plane. At the borders between these two regions super critical pitchfork bifurcations occur. The stable gaits, heading at an angle, go through a period doubling sequence as stiffness is increased. The stable range in stiffness is about 4000N/m and 500Ns/m in damping. This corresponds to a tolerance of about $\pm 15\%$ in stiffness and $\pm 20\%$ in damping. Note that not all unstable periodic gaits heading straight down the plane were located. However, it is

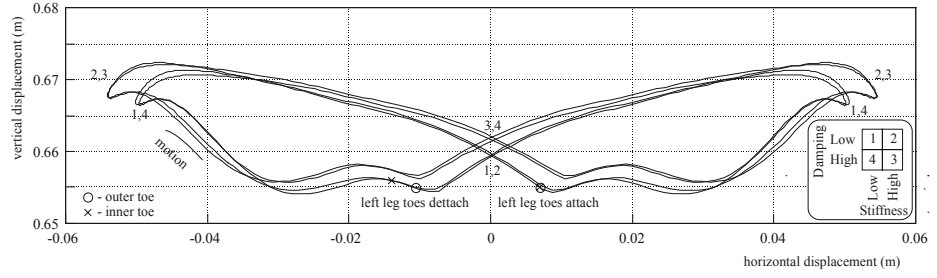


Figure 13. Hip center trajectory for four different ground plane characteristics. Stiffness seems to control the swaying amplitude whereas damping seems to control the height at the middle region.

very likely that there exists such a solution for each found oblique periodic gait. The period doubling sequence of oblique gaits were not followed. Therefore, the right border, at the arrow indicating period doubling, could probably be moved a little bit to the right, thus increasing the area of the stable region. This ground plane model assumes that damping is only present during motion into the ground plane. In Adolfsson [1], an alternative ground plane model is discussed, where damping is present during upward motion depending on the upward velocity. That model gives similar results to the present model, although the vertical damping needs to be lower.

To investigate the ground plane characteristics effect on the dynamics of the gait, four different locations were selected in the stable region, numbers 1-4 in Figure 12 give these locations. The dynamics of the walker was investigated by looking at the hip center trajectory during one stride. In Figure 13, the result is shown. The butterfly-shaped trajectories are quite similar but some distinct differences exist. At low damping, the walkers have midpoints lower than the high damping walkers. However, the high stiffness walkers have larger swaying amplitude, compared to the corresponding low stiffness walkers. Thus, the swaying amplitude seems to be controlled by the stiffness of the ground. Also shown in the figure is the corresponding hip center position when the left legs toe events occur. The toe points attaches to the ground at almost the same hip center position, whereas the detachment occurs at different positions.

2.7 KNEE STIFFNESS

Although the devised design of the knee should agree with the model used, it is good to have some margin against parameter uncertainties. In addition, the knee model does not include dry friction, which is frequently present between sliding surfaces. In Figure 14, a parameter study, similar to the previous study, is shown, where the angular stiffness and angular damping of the knee is varied. The ground plane characteristics is chosen from the lower stable region in Figure 14, being 13500N/m and 800Ns/m in the vertical direction. As can be seen in

Finding an implementable 3D passive walker

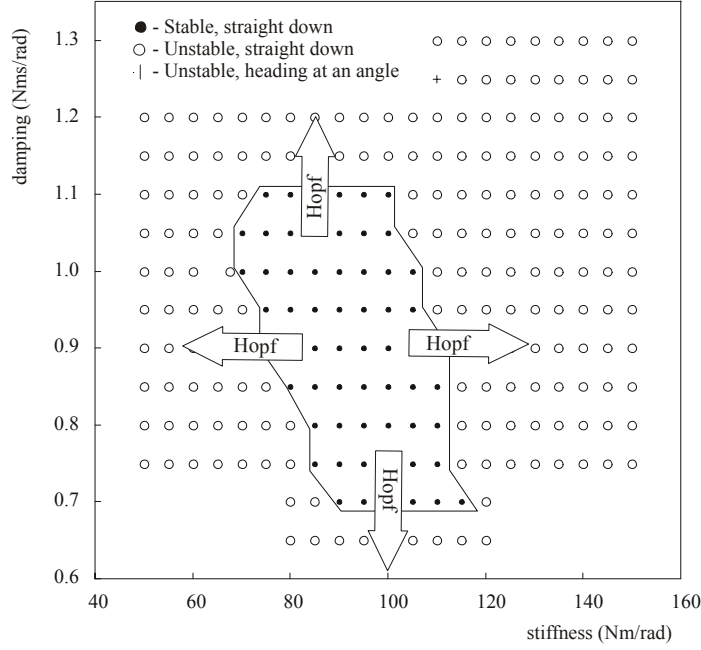


Figure 14. Parameter study to determine the limits in knee stiffness and knee damping for stable periodic gaits.

the figure, there exists a region of stable gaits, surrounded by Hopf bifurcations, which render the gaits unstable. The gaits found were seen to head straight down the plane, except for one parameter combination (the plus sign in Figure 14). The range in angular stiffness is about 40Nm/rad and the range in angular damping is circa 0.4 Nms/rad. This results in similar tolerance percentage as in the ground plane characteristics study, being about $\pm 18\%$ in angular stiffness and $\pm 20\%$ in angular damping.

2.8 FRICTION LEVELS

In Figure 15, the no slip friction coefficients between the toe points and ground plane are plotted for the inner and outer toe. The required combined friction coefficient can be written as,

$$\mu_{\text{required}} = \frac{\sqrt{(F_{11})^2 + (F_{13})^2}}{F_{12}}. \quad (2)$$

For the outer toe, the combined friction coefficient is below one for most of the times, except for a short period of time at the end of the contact. At the very end, the required friction level is about 20, which would result in a slip. However the time spent outside the unit circle is less than 2/100 of a second and is not expected to cause any problem. The inner toe has a local peak in

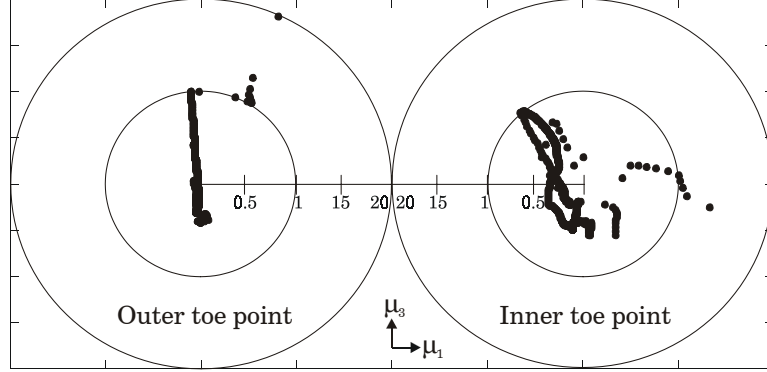


Figure 15. Required friction levels during the stance phase for the inner and outer toe points. Notice the change of scale inside and outside of the unit circle. The high friction levels are only required for very short intervals of time, typically 10ms. Thus, these peaks are not expected to cause any larger slip at the toe points.

the middle of the contact interval, where the required peak friction level is 1.3. However, the time spent outside the unit circle only lasts for 3/100 of a second.

2.9 INCLINATION

The range in inclinations for stable operation of the walker is between 4.3 degrees and 4.7 degrees. At the lower inclination, a Hopf bifurcation occurs. At the higher inclination a super critical pitchfork bifurcation occurs, thus creating two stable branches walking obliquely down the plane. These two branches only remains stable for a short interval and become unstable in a Hopf bifurcation.

3 Robustness

Equally important as to allow for a reasonable margin in parameter space is to have some margin of error in the initial conditions. Without this, it will be hard to initiate the gait of the walker. It has been demonstrated by both Ruina and McGeer that it is possible, albeit with some practice needed, to manually throw the planar walker down the incline. Thus, it is desirable that the initial condition margin for the 3D passive walker should be similar to those found for planar passive walkers. However, with current computer speed it is not feasible to find this margin for all possible combinations of state variables. The search was therefore limited to disturbing one state variable at a time from the found periodic orbit. In addition, these disturbances are not necessarily similar to

those occurring during an experiment. For example, an error in torso attitude certainly involves some error in the horizontal position of the torso.

The margins were calculated for five consecutive configurations, equally spaced over one step, as shown in Figure 21 in the appendix. This investigation was done to see if a particular configuration was more suitable for initiating the gait, i.e. had a larger margin compared to other configurations. For comparison, the same study was performed on the 3D walker having an overlap in the feet. Finding the margins were done by first finding a disturbance, which would make the walker fall within 20 strides. After that, interval halving was used to find the border between failure and success. These borders were obtained with an relative accuracy of 1%. Of course, it is possible that there exist regions with multiple borders of failure and success. This was also observed

when performing the same stability on the next step in the stride. However, the differences between margins were small. In the appendix, a graphical illustration of the robustness study is shown. This illustration was made to visually identify preferable configurations for initiation of the gait. However, no outstanding configuration was visually identified, which would imply that other considerations could be made, such as initiating gait at an easily obtained configuration.

In Table 1, a statistical comparison between the walker with an overlap in the feet and the walker without overlap is shown. This was calculated by looking at the ratio of state space volume for the two walkers,

$$m_j = \sqrt[n]{\prod_{i=\mathbf{s}_j} \frac{\Delta o_i}{\Delta n o_i}} = \exp \left(\frac{1}{n} \sum_{i=\mathbf{s}_j} \log \left(\frac{\Delta o_i}{\Delta n o_i} \right) \right) \quad (3)$$

where Δo_i is the range in initial conditions for the walker with an overlap in state i , similarly, $\Delta n o_i$ is the range for the walker without an overlap. The vector $\mathbf{s}_j$ is a vector of state indices. The mean, m_j , was then averaged over the five configurations,

$$M_j = \sqrt[5]{\prod_{i=1}^5 m_j} = \exp \left(\frac{1}{5} \sum_{i=1}^5 \log m_j \right). \quad (4)$$

The standard deviation is calculated on $\log m_j$. In most states, the data have small standard deviation, which indicates that the ratio is fairly similar in the different configurations. The general result is that the walker with an overlap has the larger margin in almost all initial conditions at all configurations. For example, the margin in initial velocities of the torso reference point is almost seven times larger in favor of the overlap configuration. Somewhat surprisingly is the low advantage for the overlap walker in the angular velocities of the torso.

Table I. Statistical comparison between a 3D walker with an overlap in the feet and a 3D walker without an overlap. The numbers are calculated by averaging the logarithm of the ratios between maximum possible disturbances in the two walkers. The average is calculated from five walker configurations equally spaced in time. Thus, a value greater than one means that the overlap walker is less sensitive to disturbances than the no-overlap walker. The standard deviation is given as a factor.

State ($\mathbf{s}_j$)	mean	std($\ast\div$)	median
height (q_2)	7.7	2.3	5.4
orientation (q_5, q_6)	8.9	1.1	8.6
leg angles (q_7, q_8, q_9, q_{10})	3.0	1.5	2.7
toe pos. lateral ($q_{11}, q_{13}, q_{15}, q_{17}$)	13.5	1.5	12.0
toe pos. sagital ($q_{12}, q_{14}, q_{16}, q_{18}$)	9.0	2.1	11.9
hip center velocity (u_1, u_2, u_3)	7.0	1.2	7.1
orient. angular velocities (u_5, u_6)	1.2	1.3	1.2
leg angular velocities (u_7, u_8, u_9, u_{10})	2.4	1.5	2.2
all states except u_4	4.5	1.1	4.6

4 Implementation

Early walker configurations, with an overlap in the feet, had a total mass of 7.5kg. The current design was made to support this weight. However, this design was hard to realize due to the torsos high radius of gyration. The moment of inertia was therefore reduced, which unfortunately resulted in unstable periodic gaits. However, this was possible to correct by increasing the mass of the torso. To retain similar ground plane penetrations, the ground plane stiffness had to be increased. The current configuration, having no overlap, has a total mass of 11.4kg. Since the increase of mass was made at the torso, the legs needed to support this additional mass. The experimental mechanism turned out to be slightly weak to support this added mass. This resulted in too much bending of the legs and hip axis, which resulted in the two legs hitting each other during the swing phase. In Figure 16, a picture of the experimental walker is shown.

4.1 KNEE MECHANISM

To make the implemented knee locking mechanism behave similarly to the modeled knee, the design in Figure 17 was used. This design is passive and only requires a simple link mechanism between the lower leg and upper leg. The upper link is the spring/damper element. Its operation is as follows. When the lower leg is free to swing the spring/damper link will be fully extended and the two links will have negligible influence on the motion. When the lower leg goes into hyperextension, the link mechanism will be constrained by the

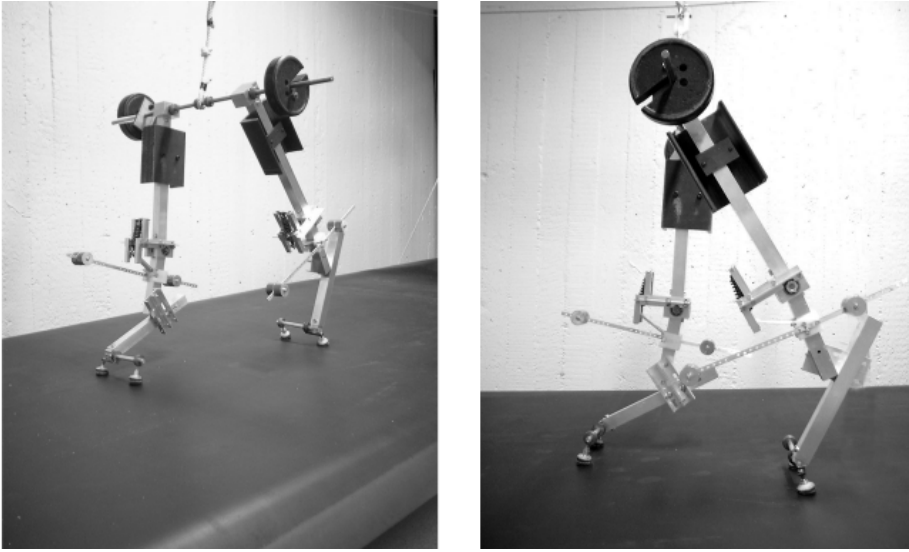


Figure 16. Photographs of the experimental walker, which didn't succeed in walking. The weights on the hip axis are placed at a location to provide for the necessary moment of inertia. The weight of the lower legs made it necessary to put the added weights on booms as to obtain the right moment of inertia.

slot and the spring/damper element will compress. During forward swing, the link mechanism will slide against the surface furthest away. During backward swing, the link mechanism will slide against the opposite surface. To make sure that smooth sliding occurs, the distance between the surfaces is slightly larger than the radius of the sliding bar. With this design, the damper will generate a force, both in forward and backward swing, as long as the link mechanism is constrained. This agrees with the theoretical model.

For a sound operation of the knee, it's important that there exist sufficient motion in the slot. Therefore, the angular spring and angular damping should have suitable values as to allow for some motion. Previously presented 3D walkers, see Adolfsson *et al.* [3], had high knee stiffness, resulting in very little motion. However, the current design have a maximum angular hyperextension of about 14 degrees, see Figure 6. This corresponds to a movement in the slot of 24mm. This should be sufficient for the spring/damper element to perform adequately.

The spring/damper element is a Kyosho "Turbo Big Shock", intended for use in radio controlled off road vehicles. The level of damping can be adjusted by changing the viscosity of oil or by changing the diameter of the piston plate hole. Stiffness is controlled by a change of springs. The springs can also be pre-loaded, although that was not tried. Parameters were obtained by the linear

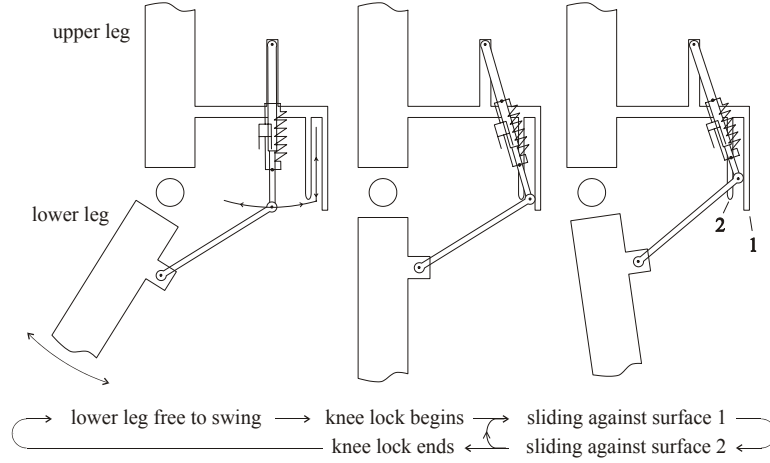


Figure 17. Illustration of the knee mechanism and its different modes of operation. The left pane show a lower leg that is free to swing. When the lower leg swings forward, the link mechanism will be guided into the slot, and the spring/damper will start to work (middle and right pane).

model,

$$I_{\text{lowerleg}}\ddot{\alpha} + c_{\text{knee}}\dot{\alpha} + k_{\text{knee}}\alpha = 0 \quad (5)$$

to experimental data. The agreement between measured data and the model was good. Fixating the upper leg and releasing the lower leg, while measuring the angle of the lower leg, gave the needed experimental data. In Figure 18, a picture of the implemented knee mechanism is shown.

4.2 TOE POINTS

It was originally conceived that the walker should have toe lines walking on a soft ground plane. The toe lines were to be implemented with a 10mm bar, thus making it possible to have an overlap in the feet. However, as the configuration having no overlap was discovered, it was decided to use a hard ground plane and to put the softness in the toes. This design would make it easier to adjust the toe positions and to adjust the compliance of the toe points. In Figure 19, the design of the toe points is shown. The vertical position can be adjusted by releasing the lock nuts and the horizontal position by releasing the lock ring. Compliance in the toe points are obtained by using a soft foam rubber layer. Below the foam rubber is a rubber layer as to obtain a high levels of friction. All layers are glued together. Foam rubber is a highly nonlinear material, which makes it quite hard to estimate adequate model parameters. Stiffness was measured by statically loading the toe points and recording the corresponding deflection. Damping was not estimated, however, very little bounce was observed when dropping a mass, attached to the toe points, on a hard surface.

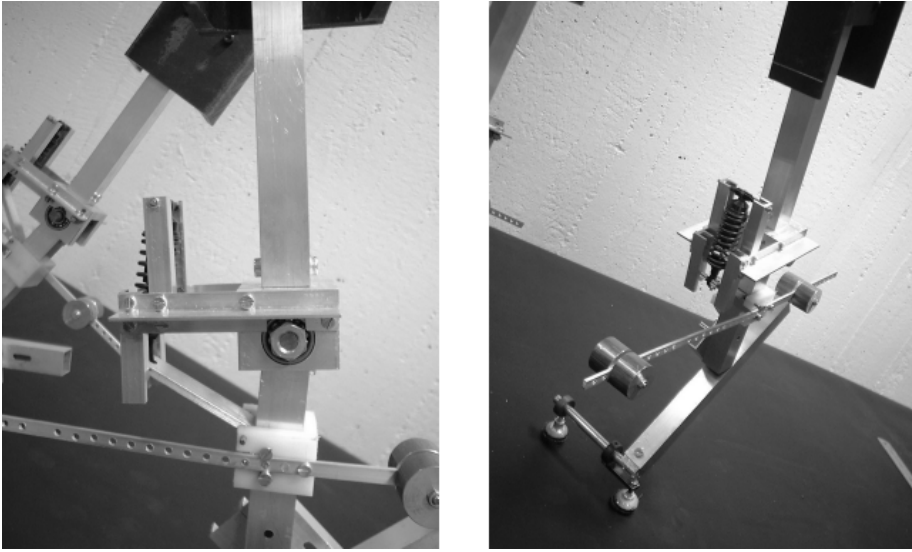


Figure 18. Close up of the implemented knee mechanism. The damper is inside the coil spring.

4.3 MASS DISTRIBUTION

The weight of different parts were obtained by using an accurate scale. The mass centers of the upper and lower legs were found by balancing the parts on a sharp edge at a few different locations. By finding the intersection of these edges, the mass center location could be found. When the center of mass position and weight was known, the part was hinged like a pendulum, with the center of mass situated some distance away from the hinge axis. The period times for small oscillations were measured, and from this, the moment of inertia was calculated. Masses were then added at appropriate locations to obtain mass distributions corresponding to the parameters found in Appendix 6.1. With the exception of the lower legs, this extra mass could be fitted naturally on the existing structure. There was little mass to be added at the lower legs and to obtain the right moment of inertia this masses were placed on extended booms, see Figure 18. By redistributing the mass or lightening the structure, it should be possible to remove these extended booms.

4.4 INCLINE

The ground plane measured 1.5m x 6m and its inclination is adjustable by changing the height at one of the ends, see Figure 20. A safety wire, attached between the ends of the incline, was used to prevent damage if the walker would fall. The walker is attached to the safety wire using a slack rope, attached on the hip axis of the walker, and at a pulley running on the wire. To reduce toe point slip, the incline was covered with a rubber mat. The short range of stable

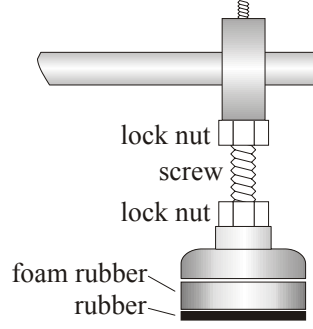


Figure 19. The implemented toe points are adjustable by releasing the lock nut. The rubber sole is used to get a high levels of friction. The toe point is also laterally adjustable by releasing and moving the lock ring. Foam rubber is used to get compliance of the toe point.

inclinations, between 4.3 and 4.7 degrees, corresponds to a range in height of 40mm.

4.5 EXPERIMENTS

Despite many efforts to manually throw the experimental walker down the incline in an appropriate way, it was not possible to obtain a sustained gait. However, after the stance leg foot impacts with the ground, the swing leg release from the ground and performs a motion resembling the simulated motion. The bounce of the lower leg at knee impact was also observed, see Figure 6.

5 Conclusions and future work

Initially it was believed that initiating the 3D passive walker should only be slightly harder compared to starting the planar passive walkers. This optimistic view was enhanced by watching the success of the small 3D Tinkertoy walker, see Coleman and Ruina [5]. However, this did not prove to be the case and we were forced to seek the cause of this difficulty. However, why the experimental walker didn't walk is hard to answer due to the lack of adequate measuring equipment. Either there exists errors in the parameters of the walker, or the gait was not initiated correctly. Of course, it could also be a combination of the two. To investigate this a catapult device for initiating gait could be built. This device should provide for repetitive initial conditions. Secondly, a 3D motion measuring system should be used to compare trajectories of the experimental walker to the simulated trajectories. Using such a system, it should be possible to establish that initial conditions are within the margins given by the robustness study. If the walker still doesn't walk, the error is probably in parameters or modelling. However, the 3D measuring system at hand was considered not good enough as to provide the needed data and no



Figure 20. The inclined ground plane. The inclination is adjustable by changing the height at one end. The walker is attached to the safety wire via a slack rope and a pulley.

other system was available. Since this part was missing, a catapult device was not built.

However, some issues could have been dealt with if time had been available. This includes improving the construction of the experimental walker. The structure of the walker needs to be strengthened as to minimize bending and play. In the hip axis, this could be done by replacing the middle part with a tube. This tube should have a appropriate radius as to improve bending stiffness over the current design. The knee and hip joints seem to be stiff enough, however the toe axis is too weak. As previously mentioned, the structure was too weak to carry its own weight resulting in collision between the "non-overlapping" toe points. This could be corrected by attaching an element capable of taking compression forces from the end of the toe axis to a point just below the knee.

These suggestions are only small improvements over the current design and it might be necessary to rethink large parts of the design. Such as implementing and simulating contact forces that are easier to model. This could include a foot with a contact patch instead of a line. In addition, a more precise knee locking mechanism might be needed. For example, this mechanism could release the knee after the hind leg has released from the floor. This would however include some mechatronic device to control the knee lock and release.

6 Appendix

6.1 PARAMETERS AND INITIAL CONDITIONS

Parameters used in the simulation of the no-overlap 3D walker. See Adolfsson *et. al.* [3] for a detailed description of the mass and point labels used.

Body	mass (kg)	moments of inertia (kg m^2)					
		x	y	z	xy	xz	yz
$\mathcal{T}$	5.000	0.3000	0.3000	0.3000	0.0	0.0	0.0
$\mathcal{U}_1, \mathcal{U}_2$	2.200	0.0200	0.0010	0.0100	0.0	0.0	0.0
$\mathcal{L}_1, \mathcal{L}_2$	1.013	0.0120	0.0010	0.0100	0.0	0.0	0.0

Point Label	x (m)	y (m)	z (m)	Ref. frame
T_{cm}	0.0000	0.0000	0.0000	$\mathcal{T}$
U_1	-0.1295	0.0000	0.0000	$\mathcal{V}$
U_2	0.1295	0.0000	0.0000	$\mathcal{V}$
$U_{\text{cm}1}, U_{\text{cm}2}$	0.0000	-0.1400	0.0000	$\mathcal{U}_1, \mathcal{U}_2$
L_1, L_2	0.0000	-0.3500	0.0000	$\mathcal{U}_1, \mathcal{U}_2$
$L_{\text{cm}1}, L_{\text{cm}2}$	0.0000	-0.0800	0.0432	$\mathcal{L}_1, \mathcal{L}_2$
L_{11}	-0.0160	-0.3164	0.1050	$\mathcal{L}_1$
L_{12}	-0.1020	-0.3174	0.1050	$\mathcal{L}_1$
L_{21}	0.0160	-0.3164	0.1050	$\mathcal{L}_2$
L_{22}	0.1020	-0.3174	0.1050	$\mathcal{L}_2$

Hip/Knee	Value	Plane	Value
k_{knee}	100.0 Nm rad^{-1}	k_x, k_z	6000 Nm^{-1}
c_{knee}	1.0 Nm s rad^{-1}	k_y	13000 Nm^{-1}
c_{hip}	0.04 Nm s rad^{-1}	c_x, c_z	500 N s m^{-1}
α	0.08 rad	c_y	770 N s m^{-1}

Finding an implementable 3D passive walker

Initial conditions used for the direct numerical simulation of the no-overlap 3D passive walker. Toe 21 and 22 are attached to the ground and knee two is locked. Knee two corresponds to leg angles q_9 and q_{10} .

Configuration			
State	Value	State	Value
q_1	0 m	q_{10}	$-1.0255 \cdot 10^{-1}$ rad
q_2	$6.5804 \cdot 10^{-1}$ m	q_{11}	$1.4251 \cdot 10^{-1}$ m
q_3	0 m	q_{12}	$-2.2477 \cdot 10^{-1}$ m
q_4	$-0.0000 \cdot 10^{-0}$ rad	q_{13}	$-5.6571 \cdot 10^{-2}$ m
q_5	$4.7396 \cdot 10^{-2}$ rad	q_{14}	$-2.2798 \cdot 10^{-1}$ m
q_6	$5.5742 \cdot 10^{-2}$ rad	q_{15}	$-7.0790 \cdot 10^{-2}$ m
q_7	$3.0000 \cdot 10^{-1}$ rad	q_{16}	$1.1198 \cdot 10^{-2}$ m
q_8	$3.7535 \cdot 10^{-1}$ rad	q_{17}	$1.5149 \cdot 10^{-2}$ m
q_9	$4.2700 \cdot 10^{-2}$ rad	q_{18}	$1.0877 \cdot 10^{-1}$ m
Velocities			
State	Value	State	Value
u_1	$-1.9585 \cdot 10^{-1}$ m s $^{-1}$	u_6	$2.6336 \cdot 10^{-1}$ rad s $^{-1}$
u_2	$1.7444 \cdot 10^{-1}$ m s $^{-1}$	u_7	$-3.0461 \cdot 10^{-0}$ rad s $^{-1}$
u_3	$4.0490 \cdot 10^{-1}$ m s $^{-1}$	u_8	$7.5069 \cdot 10^{-0}$ rad s $^{-1}$
u_4	$4.3913 \cdot 10^{-2}$ rad s $^{-1}$	u_9	$9.6030 \cdot 10^{-3}$ rad s $^{-1}$
u_5	$1.5456 \cdot 10^{-1}$ rad s $^{-1}$	u_{10}	$1.3435 \cdot 10^{-0}$ rad s $^{-1}$

6.2 RAW DATA FROM ROBUSTNESS STUDY.

Numbers show the maximum possible disturbance away from the periodic solution at five different configurations. The "no overlap" and "overlap" columns refer to the feet configuration of the studied walkers. Both maximum positive and maximum negative disturbance are listed. For a graphical illustration of this data and the walker configurations see Figure 21 and 22. Note that the toe point state variables are removed when the corresponding toe point is not attached to the ground.

configuration 0	state	no overlap	overlap	no overlap	overlap
	q2	0.0008	0.0039	-0.0006	-0.0022
	q5	0.0113	0.1707	-0.0161	-0.0858
	q6	0.0052	0.0831	-0.0094	-0.0231
	q7	0.0174	0.0445	-0.0140	-0.0210
	q8	0.1193	0.1222	-0.1490	-0.1252
	q9	0.0060	0.0148	-0.0027	-0.0118
	q10	0.0080	0.0284	-0.0084	-0.0127
	q15	0.0302	0.1664	-0.0283	-0.2664
	q16	0.0173	0.0391	-0.0227	-0.0208
	q17	0.0108	0.1186	-0.0102	-0.0413
	q18	0.0049	0.0414	-0.0083	-0.0126
	u1	0.0082	0.2184	-0.0071	-0.0754
	u2	0.3016	1.2609	-0.2984	-0.2598
	u3	0.0127	0.0801	-0.0239	-0.0338
	u4	4.2900	3.5112	-2.4150	-2.7094
	u5	0.4281	1.0193	-0.3047	-0.7483
	u6	2.3906	0.2493	-1.5078	-0.9565
	u7	1.8203	25.7080	-0.4297	-0.6790
	u8	3.6719	10.1404	-3.0156	-2.1797
	u9	0.3211	0.7595	-0.1934	-0.8168
	u10	3.3250	20.3672	-2.9625	-9.6797

Finding an implementable 3D passive walker

configuration 1	state	no overlap	overlap	no overlap	overlap
	q2	0.0023	0.0141	-0.0053	-0.0083
	q5	0.0230	0.5316	-0.0406	-0.0988
	q6	0.0213	0.0864	-0.0150	-0.0825
	q7	0.0104	0.0608	-0.0165	-0.0208
	q8	0.0336	0.1298	-0.0158	-0.0412
	q9	0.0334	0.1908	-0.0197	-0.0405
	q10	0.1031	0.0871	-0.0547	-0.0820
	q11	0.0055	0.1122	-0.0029	-0.0288
	q12	0.0005	0.0153	-0.0005	-0.0041
	q13	0.0049	0.1097	-0.0078	-0.0276
	q14	0.0012	0.0134	-0.0010	-0.0071
	u1	0.0126	0.2460	-0.0115	-0.0654
	u2	0.0309	0.1785	-0.0387	-0.0850
	u3	0.0156	0.1047	-0.0272	-0.0351
	u4	2.9537	2.9076	-2.4075	-2.6648
	u5	0.2083	0.6330	-0.3266	-0.5720
	u6	0.4534	0.2789	-0.3618	-0.4222
	u7	0.4655	0.9716	-0.3073	-1.0381
	u8	11.5811	21.8473	-11.5093	-11.3384
	u9	0.1475	0.5486	-0.1445	-0.2410
	u10	0.3086	0.9746	-0.6194	-0.6967
configuration 2	state	no overlap	overlap	no overlap	overlap
	q2	0.0010	0.0425	-0.0058	-0.0146
	q5	0.0110	0.2807	-0.0151	-0.1450
	q6	0.0213	0.1019	-0.0341	-0.0949
	q7	0.0295	0.1091	-0.0116	-0.0242
	q8	0.0389	0.2637	-0.0088	-0.0333
	q9	0.0375	0.0543	-0.0327	-0.0647
	q10	0.3447	0.3914	-0.2140	-0.1938
	q11	0.0063	0.1071	-0.0059	-0.0378
	q12	0.0010	0.0113	-0.0013	-0.0065
	q13	0.0053	0.0852	-0.0096	-0.0365
	q14	0.0013	0.0161	-0.0010	-0.0112
	u1	0.0261	0.2308	-0.0136	-0.0693
	u2	0.0729	0.3932	-0.0716	-0.3932
	u3	0.0301	0.2469	-0.0340	-0.0580
	u4	1.2688	2.4304	-1.2703	-1.9156
	u5	0.2997	0.5546	-0.3342	-0.5639
	u6	0.6425	0.2356	-0.3735	-0.4077
	u7	0.7894	2.0181	-0.6279	-1.4952
	u8	25.3687	26.1237	-12.0532	-16.5180
	u9	0.0956	0.4028	-0.1182	-0.1636
	u10	0.2618	0.6626	-0.3644	-0.4019

configuration 3	state	no overlap	overlap	no overlap	overlap
	q2	0.0110	0.0702	-0.0056	-0.0122
	q5	0.0122	0.2698	-0.0228	-0.1671
	q6	0.0279	0.0901	-0.0265	-0.0748
	q7	0.0197	0.1100	-0.0220	-0.0366
	q8	0.0163	0.2186	-0.0107	-0.0472
	q9	0.0740	0.0645	-0.0410	-0.1992
	q10	0.0968	0.1942	-0.1139	-0.7493
	q11	0.0083	0.1239	-0.0069	-0.0541
	q12	0.0008	0.0135	-0.0017	-0.0083
	q13	0.0098	0.0774	-0.0205	-0.0717
	q14	0.0020	0.0181	-0.0011	-0.0105
	u1	0.0354	0.2363	-0.0155	-0.0649
	u2	0.0400	0.3872	-0.0583	-0.3401
	u3	0.0549	0.6018	-0.0569	-0.0983
	u4	2.3047	2.6993	-1.3896	-2.0653
	u5	0.2897	0.5112	-0.3260	-0.7465
	u6	0.8744	0.2317	-0.7588	-0.6359
	u7	0.8461	4.3837	-0.8207	-2.0433
	u8	17.7763	40.0307	-18.1475	-32.3564
	u9	0.0887	0.2864	-0.1339	-0.1504
	u10	0.3787	0.6427	-0.6655	-0.5296
configuration 4	state	no overlap	overlap	no overlap	overlap
	q2	0.0019	0.0852	-0.0016	-0.0117
	q5	0.0150	0.1569	-0.0219	-0.1498
	q6	0.0133	0.1410	-0.0324	-0.0884
	q7	0.0400	0.0997	-0.0340	-0.0504
	q8	0.0760	0.3483	-0.0106	-0.0508
	q9	0.0184	0.0726	-0.0426	-0.0455
	q10	0.0384	0.1746	-0.1228	-0.1363
	q11	0.0083	1.0993	-0.0105	-0.0799
	q12	0.0014	0.0250	-0.0013	-0.0077
	q13	0.0080	0.0558	-0.0165	-0.0865
	q14	0.0048	0.0241	-0.0021	-0.0094
	u1	0.0221	0.2367	-0.0096	-0.0642
	u2	0.0466	0.2298	-0.0361	-0.2396
	u3	0.0238	0.3550	-0.0498	-0.1127
	u4	1.7178	3.2696	-1.7766	-2.7417
	u5	0.1427	0.5715	-0.1799	-0.4639
	u6	0.5331	0.2426	-0.4709	-0.4199
	u7	0.5546	1.5057	-0.8223	-1.5722
	u8	11.0471	25.5011	-10.7568	-14.4995
	u9	0.1718	0.3444	-0.1929	-0.1883
	u10	0.5065	0.7486	-1.0051	-0.7136

6.3 GRAPHICAL ILLUSTRATION OF ROBUSTNESS DATA

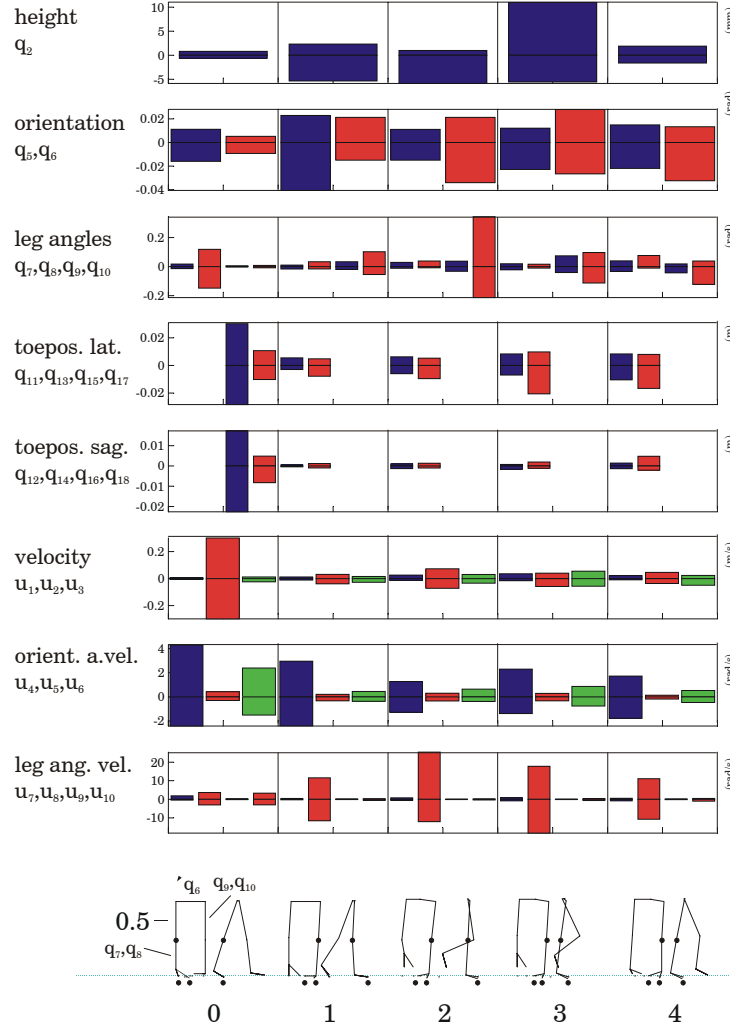


Figure 21. Study of robustness in a 3D passive walker with no-overlap in the feet. The graphs show the maximum possible disturbance from the stable periodic gait at five different configurations of the walker.

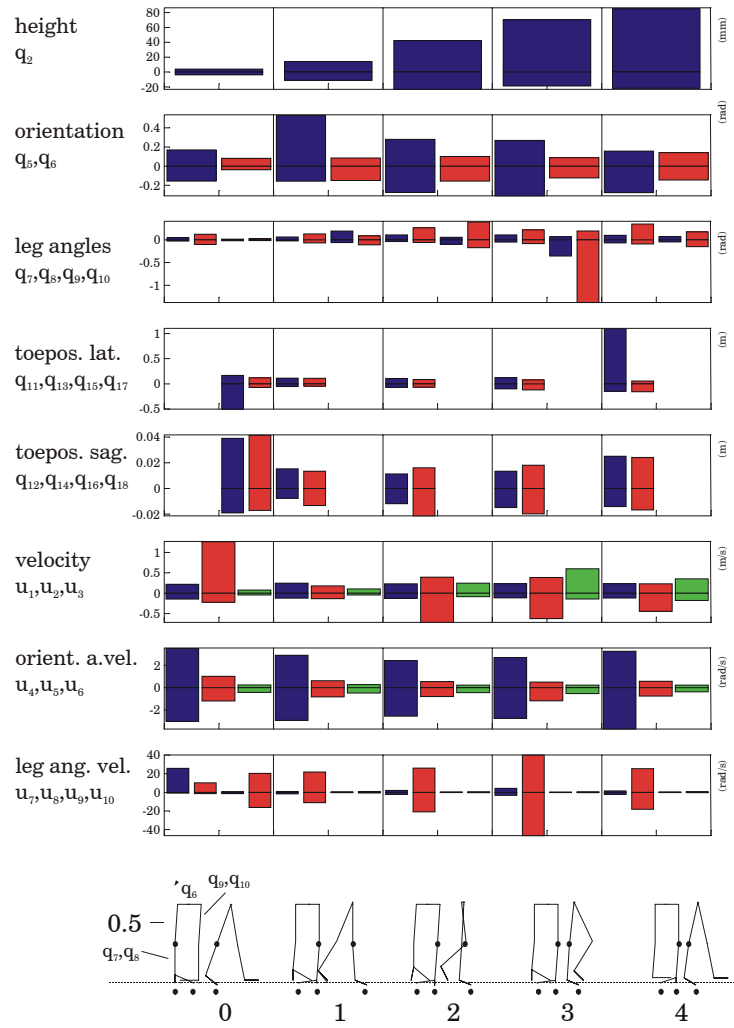


Figure 22. Study of robustness in a 3D passive walker with overlap in the feet. The graphs show the maximum possible disturbance from the stable periodic gait at five different configurations of the walker.

Bibliography

- [1] Adolfsen, J.: 2000a, ‘3D Passive walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [2] Adolfsen, J.: 2000b, ‘Planar Passive Walkers: Code Implementation Issues’. Technical report, KTH, Department of Mechanics.
- [3] Adolfsen, J., H. Dankowicz, and A. Nordmark: 2001, ‘3D Passive Walkers: Finding Periodic Gaits in the Presence of Discontinuities’. *Nonlinear Dynamics* **24**(2), 205–229.
- [4] Chatterjee, A. and M. Garcia: 1998, ‘Small Slope Implies Low Speed in Passive Dynamic Walking’. *Dynamics and Stability of Systems*. submitted for publication.
- [5] Coleman, M. and A. Ruina: 1998, ‘An Uncontrolled Toy That Can Walk But Cannot Stand Still’. *Physical Review Letters* **80**(16), 3658–3661.
- [6] Dankowicz, H., J. Adolfsen, and A. Nordmark: 2001, ‘Existence of Stable 3D-Gait in Passive Bipedal Mechanisms’. *Journal of Biomechanical Engineering* **123**(1), 40–46.
- [7] Garcia, M., A. Chatterjee, and A. Ruina: 1998a, ‘Efficiency, Speed, and Scaling of Passive Dynamical Bipedal Walking’. *Dynamics and Stability of Systems*. submitted for publication.
- [8] Garcia, M., A. Chatterjee, A. Ruina, and M. Coleman: 1998b, ‘Passive-dynamic Models of Human Gait’. In: *Proceedings of the Conference on Biomechanics and Neural Control of Human Movement*. pp. 32–33.
- [9] Garcia, M., A. Chatterjee, A. Ruina, and M. Coleman: 1998c, ‘The Simplest Walking Model: Stability, Complexity, and Scaling’. *ASME Journal of Biomechanical Engineering* **120**, 281–288.
- [10] McGeer, T.: 1990a, ‘Passive Bipedal Running’. In: *Proceedings of the Royal Society of London: Biological Sciences*, Vol. 240. pp. 107–134.
- [11] McGeer, T.: 1990b, ‘Passive Dynamic Walking’. *International Journal of Robotics Research* **9**, 62–82.
- [12] McGeer, T.: 1990c, ‘Passive Walking with Knees’. In: *Proceedings of the IEEE Conference on Robotics and Automation*, Vol. 2. pp. 1640–1645.

- [13] McGeer, T.: 1993, ‘Dynamics and Control of Bipedal Locomotion’. *Journal of Theoretical Biology* **163**, 277–314.
- [14] Mombaur, K., M. J. Coleman, M. Garcia, and A. Ruina: 2000, ‘Prediction of Stable Walking for a Toy That Cannot Stand’. submitted for publication in PRL.
- [15] Piiroinen, P.: 2000, ‘Passive Walking: Transition from 2D to 3D’. Technical report, KTH, Department of Mechanics.
- [16] Wisse, M., A. L. Schwab, and R. Q. vd Linde: 2000, ‘A 3D Passive Dynamic Biped with Yaw and Roll Compensation’. to appear in ROBOTICA.