

**DESIGN AND OPTIMIZATION OF A COMPASS ROBOT WITH
SUBJECT TO STABILITY CONSTRAINT**

A Thesis

by

ZOHREH KESHAVARZBAGHERI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2012

Major Subject: Mechanical Engineering

Design and Optimization of a Compass Robot with Subject to Stability Constraint

Copyright 2012 Zohreh KeshavarzBagheri

**DESIGN AND OPTIMIZATION OF A COMPASS ROBOT WITH
SUBJECT TO STABILITY CONSTRAINT**

A Thesis

by

ZOHREH KESHAVARZBAGHERI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Co-Chairs of Committee,	Reza Langari
	Richard J. Malak
Committee Member,	Nina P. Robson
Head of Department,	Jerald A. Caton

August 2012

Major Subject: Mechanical Engineering

ABSTRACT

Design and Optimization of a Compass Robot with Subject to Stability Constraint.

(August 2012)

Zohreh KeshavarzBagheri, B.S., University of Tabriz;

M.S., Khajeh Nasir-al-Deen Toosi University

Co-Chairs of Advisory Committee: Dr. Reza Langari

Dr. Richard J. Malak

In the first part of this thesis, the design of a compass robot is explored by considering its components and their interaction with each other. Three components including robot's structure, gear and motor are interacting during design process to achieve better performance, higher stability and lower cost. In addition, the modeling of the system is upgraded by considering the torque-velocity constraint in the motor. Adding this constraint of DC motor make the interaction of different components more complicated since it affects the gear and walking dynamics. After achieving the design method, different actuators (motor+ gear+ batteries) are selected for a given structure and the their performance is compared in the terms of cost, efficiency and their effect on the walking stability.

In the second part of the thesis, structural optimization of the compass robot with stability constraint is investigated. The stability of a compass robot as a hybrid system is

analyzed by Poincare map. Including stability analysis in the optimization process, makes it very complicated. In addition, the objective function of the system has to be evaluated in the convergent limit cycle. Different methods are examined to solve this problem. Limit cycle convergence is the best solution among the existing methods. By adding convergence constraint to the optimization, in addition of making the stability analysis valid, it helps the optimization estimates the correct objective function in each iteration.

Finally the optimization process is improved in two steps. The first step is using a predictive model in the optimization which covers the stable domain so that one does not need to check the stability of walking in each iteration. The Support Vector Domain Description (SVDD) approach which is applied to establish the stable domain, improve the decreases the optimization time. Another important step to upgrade the optimization is developing a computational algorithm which obtains the convergent limit cycle and its fixed-point in a short time. This algorithm speeds up the optimization time tremendously and allows the optimization search in a broader area. Combining SVDD approach in combination with Fixed-Point Finder Algorithm improve the optimization in the terms of time and broader area for search.

DEDICATION

To my dear father and mother

ACKNOWLEDGEMENTS

I would like to thank my committee Co-chairs, Dr. Reza Langari and Dr. Richard J. Malak for their guidance in nonlinear systems and their optimization process and its improvement. I would like to thank them for their support and help. Thanks also to my committee member, Dr. Nina Robson for her valuable comments on this thesis.

I also would like to thank my parents for supporting my path towards research. Finally, I would like to thank Amitava Guha for all his aids and supports. I appreciate all his time and attention for my thesis.

Thanks also to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

NOMENCLATURE

m	Mass of the Robot's Leg
M_h	Hip's Mass.
D	Domain
G	Guard
R	Reset Map
f_L	Lagrangian Vector Field
$M(\theta)$	Inertial Matrix
$N(\theta, \dot{\theta})$	Coriolis Matrix
$g(\theta)$	Gravity Matrix
ϕ	Ground Slope
SPG	Speed/ Torque Gradient (rpm/Nm)
Nls	Non-load Speed (rpm)
R_a	Armature Resistance (ohm)
i_a	Armature Current, A
V_a	Applied Armature Voltage, V
V_b	Back EMF, V
T_m	Torque Developed by Motor, N-m
J	Moment of Inertia of the Motor (kgm^2)

b	Viscous-friction Coefficient of the Motor and Load
K_b	Back EMF Constant (V/rpm)
K	Torque Constant (Nm/A)
NT	Nominal Torque (Nm)
T_m	Mechanical Time Constant
μ	Mass Ratio of the Robot Leg
β	Length Ratio of the Robot Leg
$M_n(\theta)$	Normalized Inertia Matrix
$N_n(\theta, \dot{\theta})$	Normalized Coriolis Matrix
$g_n(\theta)$	Normalized Gravity Matrix
γ	Slope of the Ground
$\tau(x_k)$	Period Time for One Step
P	Poincare Map
n	Gear Ratio
η	Gear Efficiency
T_g	Torque Applied by the Gear on the Robot's Leg
u_{des}	Desired Torque Computed by Control Symmetry
T_m	Motor Torque
J_{mhip}	Moment Inertia of the Motor on the Hip
J_{ghip}	Moment Inertia of the Gear on the Hip
N_1	Reduction Factor of the Gear

J_{mank}	Moment of Inertia of the Motor at the Foot of the Stance Leg
J_{gank}	Moment of Inertia of the Gear at the Foot of the Stance Leg
b_1	Viscous Friction Coefficient of the Motor
b_2	Viscous Friction Coefficient Between the Leg and Gear Train
$M_E(x)$	Mechanical Energy for One step
$L_k(x)$	Step Length
$u(x)$	Objective Function of the System

TABLE OF CONTENTS

	Page
1 INTRODUCTION: THE IMPORTANCE OF RESEARCH.....	1
1.1 Design of a Compass Robot with Interacting Components	3
1.2 Optimization of a Compass Robot with Subject to Stability Constraint.....	5
1.3 Different Solutions and Their Drawbacks.....	8
1.4 Steps for Optimization Improvement.....	11
1.5 Developing an Algorithm for Finding the Fixed-point	12
2 BACKGROUND ON COMPASS GAIT ROBOTS.....	14
2.1 Introduction to Hybrid Systems or Systems with Impulse Effect.....	14
2.2 Dynamic Modeling of a Compass Robot	16
2.2.1 2-D Compass Robot	16
2.2.2 Lagrangian Equation of Motion	17
2.2.3 Unilateral Constraint and Domain.....	18
2.2.4 Impact Equations.....	19
2.2.5 Normalized Equations	21
2.3 Controlled Symmetries Applied to the Robot for Walking on the Flat Ground.....	22
2.4 Simulation Results for Compass Robot	24
2.4.1 Passive Gait on the Slope Ground.....	25
2.4.2 Walking Robot on the Flat Ground by Controlled Symmetries.....	26
2.5 Influence of Robot Parameters on the Robot's Gait and Bifurcation Maps.....	28
3 STABILITY ANALYSIS OF A COMPASS ROBOT	33
3.1 Poincare Map.....	33
3.2 Simulation Results.....	36
4 DESIGN OF A COMPASS ROBOT WITH INTERACTING COMPONENTS....	38
4.1 Power Transmission Selection and Modeling.....	38
4.1.1 Motor.....	39
4.1.2 Gear	41
4.1.3 Adding Motor and Gear to the Robot.....	42

4.2	Design of a Compass Robot with Interacting Components	45
4.3	Speed Equation of the Motor based on the Design Process	46
4.4	Simulation Results	47
4.5	Motor Selection	47
4.6	Battery Selection	53
4.7	Characteristics of Different Selections of Actuator	55
4.8	Effect of the Motor on the Bifurcation Maps	55
4.9	Motor Selection	58
5	CHALLENGES AND DIFFICULTIES OF A COMPASS ROBOT OPTIMIZATION	60
5.1	Optimization Problem Statement	60
5.1.1	Weighting the Objective Functions	61
5.1.2	Instability Constraint	64
5.1.3	Optimization Problem Statement	64
5.2	Optimization Difficulties and Challenges	65
5.3	Different Solution for Optimization and Their Drawbacks	66
5.3.1	All-at-once Optimization	66
5.3.2	Nested Optimization	70
5.4	Using Convergence Principle for Finding the Fixed Point	76
5.4.1	Statement of Optimization Problem and the Results of Optimization ..	80
5.4.2	The Limitation of Convergence Method in Optimization	81
5.5	Support Vector Domain Description (SVDD) Method for Improving Optimization	84
5.5.1	SVDD Methodology and Application	84
5.5.2	Applying SVDD to the Optimization Process	86
5.5.3	Optimization Problem Statement by Applying SVDD Method	90
5.5.4	Optimization (SVDD) Result	92
5.6	Comparison between Ordinary Optimization and Optimization with SVDD ..	93
6	DEVELOPING AN ALGORITHM TO FIND THE FIXED-POINT FOR DIFFERENT STRUCTURE AND ITS EFFECT ON OPTIMIZATION	95
6.1	Algorithm for Finding the Fixed Point	96
6.2	Applying “Fixed-Point Finder” to Optimization Process	101
6.3	Applying “Fixed-Point Finder” along with SVDD	102
6.4	Comparison between Different Optimization Approaches	106
6.5	Robustness	108
6.6	Conclusion	111

7 CONCLUSION	113
REFERENCES	116
APPENDIX A	119
APPENDIX B	120
APPENDIX C	121
VITA	126

LIST OF FIGURES

	Page
Fig. 2-1. Model of a compass robot	17
Fig. 2-2. Rotating world by group action: walking robot on the flat ground	24
Fig. 2-3. Phase portrait of the robot walking on the slope	25
Fig. 2-4. Phase portrait of the robot walking on the flat ground by controlled symmetries	27
Fig. 2-5. Bifurcation diagram, α as a function of β for slope angle $\gamma=3^\circ$ and $\mu=2$	29
Fig. 2-6. Bifurcation diagram, α as a function of β for slope angle $\gamma=4^\circ$ and $\mu=2$	30
Fig. 2-7. Bifurcation diagram, α as a function of μ for slope angle $\gamma=3^\circ$ and $\beta=1$	31
Fig. 2-8. Bifurcation diagram, α as a function of μ for slope angle $\gamma=4^\circ$ and $\beta=1$	32
Fig. 3-1. Poincare map	34
Fig. 3-2. Poincare map for a system with impulse	35
Fig. 4-1. DC motor diagram: workspace is under the torque-speed line.....	40
Fig. 4-2: Compass robot with motors on the foots and on the hips.....	42
Fig. 4-3. Design of a compass robot with interacting elements	45
Fig. 4-4. Speed diagram of the motor at the foot	48
Fig. 4-5. Speed diagram of the motor at the hip.....	48
Fig. 4-6. Power-torque diagram of the motor at the foot	48
Fig. 4-7. Power-torque diagram of the motor at the hip.....	48
Fig. 4-8. Torque-speed diagram of the motor at the foot	49
Fig. 4-9. Torque-speed diagram of the motor at the hip.....	49

Fig. 4-10. Efficiency of the motor at the foot.....	49
Fig. 4-11. Efficiency of the motor at the hip.....	49
Fig. 4-12. Speed diagram of the motor at the foot	51
Fig. 4-13. Speed diagram of the motor at the hip.....	51
Fig. 4-14. Power-torque diagram of the motor at the foot	51
Fig. 4-15. Power-torque diagram of the motor at the hip.....	51
Fig. 4-16. Torque-speed diagram of the motor at the foot	52
Fig. 4-17. Torque-speed diagram of the motor at the hip	52
Fig. 4-18. Efficiency of the motor at the foot.....	52
Fig. 4-19. Efficiency of the motor at the hip.....	52
Fig. 4-20. Bifurcation map for $\beta=1.464$	58
Fig. 4-21. Bifurcation map for $\beta=1.9704$	58
Fig. 5-1. Scheme of the all-at-once optimization with fixed-point and stability constraint.....	68
Fig. 5-2. Comparison of the fixed-points selected by optimization and the real fixed- points for the relevant structure.....	70
Fig. 5-3. Scheme of the nested optimization with fixed-point and stability constraints .	72
Fig. 5-4. Comparison of the real fixed-point and the one founded by optimization (high difference).....	74
Fig. 5-5. Comparison of the real fixed-point and the one founded by optimization (small difference)	75
Fig. 5-6. Scheme of the convergence method for finding the fixed-point in optimization process.....	78
Fig. 5-7. Convergence to a stable limit cycle after finite steps	79
Fig. 5-8. Errors between the initial conditions in the first steps and fixed-point after 20 steps.....	79

Fig. 5-9. Searching domain for optimization for each selected structure in each iteration.....	83
Fig. 5-10. Sampled stable and unstable points for optimization algorithm.....	87
Fig. 5-11. Sampled stable and unstable points for optimization	88
Fig. 5-12. Contour plot of constraint $g(x)$, acceptable domain for optimization	89
Fig. 5-13. The value of constraint $g(x)$ for different magnitude of variables. acceptable domain is $g(x) < 0$	90
Fig. 5-14. Scheme of the optimization algorithm with SVDD constraint.....	92
Fig. 6-1. Sampled stable and unstable points evaluated by "Fixed-Point Finder algorithm"	98
Fig. 6-2. The first element of fixed-point calculated By FPF algorithm.....	100
Fig. 6-3. The third element of fixed-point calculated by FPF algorithm	100
Fig. 6-4. Acceptable domain for optimization	104
Fig. 6-5. Values of constraint $g(x)$. Acceptable domain is for $g(x) < 0$	104
Fig. 6-6. Mass ratio for different methods of optimization.....	109
Fig. 6-7. The domain which has to be excluded from SVDD domain.....	111

LIST OF TABLES

	Page
Table 4-1. Required Battery for Different Motors	68
Table 4-2. Characteristics of Different Motor Selection.....	69
Table 4-3. Values of μ and β and Poincare Map Results for Stability Analysis	71
Table 5-1. Comparison of Optimization Methods with SVDD and without SVDD.....	95
Table 6-1. The Reference Pair and the Consecutive Pairs which Their Fixed-points Have to Be Computed.....	98
Table 6-2. Comparison Between Different Methods of Optimization	122

1 INTRODUCTION: THE IMPORTANCE OF RESEARCH

The design process of a system requires integrating different components and investigating the interaction among them. As explained by Beitz, Pahl, and Wallace (2003), complex system design often integrates different isolated subsystems. Ensuring the ability of such systems to meet the desired performance requires proper modeling of the subsystems and their interaction. Modeling of complex systems becomes more difficult due to interactions between these subsystems. The objective of the first part of this thesis is detailed design of a compass robot by considering different components which interact with each other.

The compass gait, as described by Goswami, Thuirot, and Espiau (1996) is known as the simplest model of bipedal robots and is being studied as the basis for overall walking mechanism. Although a lot of research has been done on bipedal walking simulations, little has been conducted on the process of design regarding its components' interaction. A good design will reduce the required energy and increase the stability of the system. In this thesis, after studying the dynamics, stability and control of the compass robot, we will proceed to the design of the compass robot with three components, robot's structure, motor and gear.

This thesis follows the style of *The International Journal of Robotics Research*.

The design includes developing the robot's dynamics, motor's speed-torque equation based motor and gear. In the second section, the dynamics of a robot walking on the flat ground, its stable limit cycle and the required control law is obtained. The first step of this study is modeling the dynamics of the robot walking down the slope passively and finding its stable limit cycle. Such gaits have been studied in Goswami et al. (1996) and McGeer (1990). It has been shown that for certain shallow slopes, these passive bipeds have stable walking gaits without any control input. Spong and Bullo (2002) and Spong and Bullo (2005) showed the development of a feedback control law named "Controlled Symmetries" which allows the robot to walk on any slope and also on the flat ground. Controlled symmetries is used to shape the potential energy of the bipedal robotic walker so that it can walk on flat ground. This control procedure exploits the natural symmetries in biped robots to achieve natural walking gaits.

In the third section, gait stability of a walking robot is studied. Defining the gait stability of a robot is hard; however it has to be considered for purpose of design and optimization since it has a crucial effect on its performance. Stability analysis of nonlinear system has been investigated in different references such as Perko (1991), Westervelt, Grizzle, Chevallereau, Choi, and Morris (2007) and Hiskens (2001). Stability of the compass robot, which is a hybrid system is determined by the Poincare map. Westervelt et al. (2007) showed that exponential stability of a hybrid system can be determined by evaluating the eigenvalues of the Jacobian of the Poincare map linearized about a fixed point. J.W. Grizzle, Chevallereau, Ames, and Sinnet (2010) illustrated the derivation of the Poincare map for systems with Impulses. Hiskens (2001) has used

trajectory sensitivity matrix to define the Jacobian of the Poincare map and has illustrated the calculation of the sensitivity matrix for hybrid systems. Hiskens (2001) used a trajectory sensitivity analysis to efficiently determine the stability of limit cycles in hybrid systems and has applied this method to a compass gait robot in order to analyze the gait stability of the robot.

In the third section, the Poincare map method is explained follows above mentioned References. Then stability of the compass robot is calculated by Jacobian of Poincare map. The gait stability has a great effect on the performance and optimization of the robot. It has a vital influence on the optimization process of a walking robot.

1.1 Design of a Compass Robot with Interacting Components

After reviewing the background of hybrid systems and compass robots, detailed design of a compass robot is studied in Section 4. The design of a biped robot including the mechanical and electrical components has been done by Pedersen, Nielsen, and Christiansen (2006-2007). Ye, Dede, Nasser, and Tosunoglu (2005) and Tesfu, Schlattmann, and Ziemer (2008) have studied the integration of real time control electronics into dynamic mechanical system and selection of different component of a biped robot. Chevallereau and Sardain (2000) presented a design of a biped robot which combines the components selection (motor and gear) with the system trajectory. In this reference some reference trajectories are defined, the required torque and velocity in each joint is calculated and then the effect of the gear ratio on torque and velocity are determined. This paper has shown how actuator selection affects the trajectory and the

performance criteria such as the consumed energy and maximum walking speed. Oлару, Krut, and Pierrot (2009) have investigated the mechanical design of a biped robot using hollow shaft electrical actuators with a cable transmission instead of using classical actuators consisting of motor and gear. They showed that selection of the proposed actuator helps the robot to interact with its environment more smoothly. In addition, this actuator has lower weight and inertia and higher power to mass ratio.

In the previous designs, different components were selected individually without considering their effects on each other. For example, the motor's torque-speed constraint is a very important factor, which affects the robot's dynamics tremendously, and was ignored in the previous studies. In this thesis, the design of compass robot includes selection of different parts such as motor, robot's structure and gear, evaluation of their effects on the system performance and also their effects on each other. In addition, the torque-speed constraint of the motor, which has a huge impact on other components and the system performance, is considered in the design process. Adding this constraint of DC motor makes the interaction of different subsystems more complicated since it affects the gear and robot's dynamics.

In the design of the robot, proper actuator including motor, gear and batteries are selected according to the maximum required torque which can be computed by Control Symmetries applied to the robot's leg. Then, different actuators (motor + gear) are considered for the given structure of the robot and compared with each other in terms of cost, capacity and total mass which affects the stability of the robot. The effect of the mass of the actuator on the robot's stability is analyzed by studying bifurcation maps.

Ultimately, by considering all the effective factors, the best motor, gear and batteries will be selected for a given structure of the compass robot.

1.2 Optimization of a Compass Robot with Subject to Stability Constraint

In Section 5, structural optimization of a compass robot by considering stability constraint is studied. There has been a huge interest in gait optimization of biped robots. An optimal motion can be generated by minimizing a performance criterion that affects the dynamic. In this thesis, consumed energy and step length are the desired performance criteria of the optimization.

Many studies have been done on the optimization of robot. Oliveira, Costa, Rocha, Santos, and Ferreira (2011) worked on the motion of quadruped robot using bio-inspired Central Patterns Generators (CPGs), where he optimized speed, vibration and stability simultaneously by tuning the CPG parameters. In the afore mentioned work, stability was measured based on shortest distance between the projection of center of mass on the ground and the polygon formed by the vertical projection of the robot feet contact points on the ground and was mentioned as Wide Stability Margin (WSM). A gait is considered better with higher WSM values. Iida and Tedrake (2007) presented a minimalistic approach for optimization of a one-legged hopping robot with passive elastic joint. He used the joint angle as a stability criterion and exceeding a threshold value was considered to be the point of instability. The mentioned research presents stability analysis and optimization for specific purposes using parameters strictly relevant to their model.

Some of the robot optimization walking is done on the basis of a pre-computed reference trajectory and the goal of optimization is to minimize the difference between the robot walking model and the reference trajectory. In these optimization problems, the stability of the robot walking is not computed along with the optimization since the reference trajectory is assumed to be stable. Ames (2012) has applied optimization to generate walking gait which is best fit to human walking. In this reference the goal of optimization is to minimize the difference between the output of the robot and the output of the human walking data. Ames (2012) used from hybrid zero dynamic to check the stability condition. The final trajectory is shown to be stable by checking the eigenvalues of a linearized Poincare map. Lima, Gonçalves, Costa, and Moreira (2010) utilized pre-computed reference trajectory of the gait to optimize energy consumption. He also did not consider the gait stability in the walking model since the reference trajectory is considered to be stable and by tracking that, the stability of the gait will be guaranteed. Asta and Sariel-Talay (2011) has applied optimization to produce the desired motion trajectory for a humanoid robot. The purpose of the optimization is to find the best parameters of a “Partial Fourier Series” to improve the walking model. In the mentioned reference, the goal of “Fourier Series” is to generate a stable walking model, i.e., the produced walking by “Fourier Series” is assumed to be stable. Therefore there is no need to check the stability in the optimization process.

In this thesis, a generalized method of stability analysis is performed using the Poincare map, which is a widely used tool for stability analysis of hybrid systems with periodic behavior. This approach provides a mean of stability analysis which is

independent of the specific structural model. When the stability analysis by Poincare map involves in the optimization process, makes the optimization problem complicated and cumbersome to solve since the optimization process cannot satisfy the stability nonlinear constraint for different values of variables in each iteration.

Speed and consumed energy of the robot in one step (one period) are the objectives of the optimization, which are used to explore the best structure for the robot's leg. The optimization variables include the length and diameter of the robot's leg which affect the mass, inertia and other structure characteristics. The optimization starts from the known structure as its initial guess and then searches for better structures and dimensions which result minimum value of objective function. In every optimization iteration, new trajectories are generated due to changes in structural parameters i.e., length and diameter. The new trajectories of the robot can be represented by limit cycle. Limit cycle is an isolate periodic orbit. It is referred as isolated since all other trajectories have to converge to one unique periodic orbit (if it is stable) or converge from it (in unstable case).

The stability of the gait should be checked for every generated trajectory in different optimization iteration. In order to check the stability of the robot's gait, it is required to find the limit cycle and its related fixed point for relevant values of the optimization variables. In this thesis, fixed point corresponds to initial condition from which trajectory emanate and should be equal to the condition in the next period when the robot impacts the ground and starts. Since the generated trajectory is changing in every optimization iteration, its limit cycle and fixed point are also changing

accordingly. This makes the optimization process very complicated since finding the limit cycle and its fixed point is not an easy process.

Generally the fixed point is found manually by trial and error, but in optimization, when the algorithm proceeds to check the different values of the relevant variables, it needs a well-defined method to compute the fixed point automatically for different trajectory in each iteration. So, the fixed point challenge is considered as an optimization constraint which has to be satisfied in every iteration along with the stability constraint. The fixed point constraint implies that the limit cycle which begins from an initial condition (i.e., fixed point) should continue through the same point in the subsequent cycles after the robot impacts the ground. The stability constraint is defined by the eigenvalues of the Jacobian matrix of the Poincare map linearized about the fixed point. In this thesis, different methods for solving the optimization problem and their drawbacks are discussed. Then, the solutions for improving the optimization process in terms of time and accuracy are proposed.

1.3 Different Solutions and their Drawbacks

- All-at-once Optimization:

The first solution is to consider the initial condition of the periodic orbit (i.e. the fixed point) as optimization variables along with the robot's dimensions. In this method, the optimization needs to satisfy the fixed-point and Poincare map condition in each iteration. It means the developed trajectory for different values of variables in each iteration of optimization needs to pass through the same point (fixed-point) in

subsequent cycles and also it needs to be stable. So, the fixed point and the Poincare map are applied as constraints in the optimization process. This method of optimization is not successful for finding the optimum point due to intensive non-linearity of the dynamic equations, constraints and the two interacting nonlinear constraints. Optimization is not able to satisfy both constraints even when the initial guess is selected very close to the optimum point. Therefore, we proceed to find better solutions which are applicable and able to solve the optimization problem.

- Nested Optimization:

In the nested optimization method, the optimization problem is broken down into two levels. In the upper level, robot's structure is considered as optimization variable and at the lower level, the initial condition vector is considered as optimization variable set. The objective function of the lower level optimization is minimizing the error between the fixed-point and the initial condition and its goal is to find the fixed-point for the upper level variables. In every iteration, new values of dimensions are passed to the lower level. The lower level finds the fixed point for the corresponding dimensions and passes it to the upper level and to the stability constraint to verify the walking stability for corresponding dimension and to evaluate the corresponding objective functions.

This method is able to find the minimum objective function but it has major disadvantages. The accuracy of this method is low since the lower level does not always find the fixed-point. Sometimes it gets stuck in local minimum and the result is not exactly the fixed-point. Another drawback is that this method takes a very long time to

find the optimum point since in every iteration of the upper level, the lower level optimization has to run which is a time-consuming process itself.

- Convergence Principle as Constraint:

The nested optimization is able to perform the optimization. However it has minor errors and it takes a very long time. In this thesis, a new technique is presented which finds the fixed point within the optimization process quicker. The convergence principle is used in each step to find the fixed point for every generated trajectory with different values of variables. Ocken (1995) and Akian, Gaubert, and Lemmens (2011) have shown that a given orbit of a discrete dynamical system can converge to a fixed point or periodic orbit if they start from a proper initial condition. Ocken (1995), has used a computational method to obtain the attracting periodic orbit for a one-dimensional system.

In this thesis, the same idea is applied for finding the convergent periodic orbit of the compass robot for any dimension. If a robot starts from an initial condition close to its fixed point and within the domain of attraction, it will converge to its attractive periodic orbit after a finite number of steps and the fixed point for the corresponding limit cycle can be found. Therefore we substitute fixed point constraint for periodic orbit convergence constraint in the optimization problem, because if the convergence of limit cycle is satisfied, finding its fixed-point will be very easy and can be done very fast.

This method is similar to the first one, since it is also uses “all-at-once optimization” approach. The only difference is changing the constraint for finding the fixed point. The results of this method are satisfactory. The only limitation is its extensive computational,

as in every optimization step, convergence and stability have to be analyzed which slows down the optimization process significantly. However, it is still much faster than the nested optimization.

1.4 Steps for Optimization Improvement

As it was stated, the convergence method makes the optimization problem faster but it is still not satisfactory and we need to decrease the optimization time. In Section 5, we will discuss some ways to improve the optimization method in order to reduce the computation time and increase the accuracy of the method.

- Support Vector Domain Description (SVDD)

One approach, which is useful to obtain a faster optimization process, is to use a predictive model which specifies the domain over which the system is valid i.e., the system performance is acceptable. This model is very useful in optimization since it will search within a stable domain without extrapolating beyond the obtained predictive model. Support Vector Domain Descriptive (SVDD) is an approach developed by Tax and Duin (1999) which establishes a domain classification boundary for the model using a hypersphere. Tax and Duin (1999) developed SVDD following the SVM (Support Vector Machine) approach developed by Vapnik (1999). The difference between SVM and SVDD is that SVM establishes a hyperplane to classify between classes geometrically but SVDD uses a hypersphere to determine whether a point is in the data domain. Malak Jr and Paredis (2010) applied SVDD method to predictive modeling

problem and illustrates examples for applying this approach in system design and optimization.

In this thesis, SVDD is applied in order to establish a domain, which confines the stable area. In order to establish the stable area, it is needed to sample the consistent number of points and determine their stability by Poincare map. By using this method, the optimization algorithm does not need to compute the stability of the robot in each iteration. The Poincare map constraint is substituted by a nonlinear constraint which checks whether the candidate point is inside the stable domain or not. Since computation of Poincare Map for stability analysis is time consuming, this approach reduces the computation time significantly. However, we need to check the stability of the sampled points to generate the acceptable domain, but it takes less time in compare with the case when the Poincare involves in the each iteration of optimization.

1.5 Developing an Algorithm for Finding the Fixed-point

By applying the SVDD method to optimization, we could decrease the optimization process time. However, the convergence constraint still has to be satisfied which is a very time consuming process because in every iteration it is required to find the objective function for the convergent limit cycle. In Section 6, an efficient computational method is introduced in which any given orbit converges to a fixed point or periodic orbit. The procedure is that for a given dimension of the robot's leg, the algorithm guesses a reasonable initial condition so that the limit cycle converges after a short

number of steps. This algorithm uses from convergence principle for finding the fixed points and period of corresponding limit cycle to any prescribed accuracy.

This algorithm can be used to create a time efficient and accurate module for computing the fixed point in optimization algorithm. In every iteration, optimization will call this module to find the fixed point for that dimension and then proceed to the next step. This method is very useful in optimization as it allows searching for better results in a broader area and by applying global optimization methods like Genetic Algorithm (matlab) the optimization process can be improved significantly.

2 BACKGROUND ON COMPASS GAIT ROBOTS

2.1 Introduction to Hybrid Systems or Systems with Impulse Effect

Hybrid systems are dynamical systems which are characterized by interaction between the continuous and discrete dynamics. Systems of these types are common across a number of engineering applications. In biped robots, discrete events have a significant influence on overall behavior and they are driven by unilateral constraints at the ground contact and impulse forces that occur at foot touchdown.

In this section, first we define the hybrid systems. Hybrid systems consist of several time-invariant ordinary equations linked by event-based switching mechanism and re-initialization rules Morris and Grizzle (2005). In the following, the definition of hybrid systems and hybrid flow and hybrid periodic orbits are given by Ames, Gregg, Wendel, and Sastry (2006) and Ames and Gregg (2007).

Definition: A simple hybrid systems can be defined as follows:

$$\mathcal{H} = (D, G, R, f, U) \tag{2-1}$$

- D is a manifold called the domain,
- G is an submanifold of D called the guard,
- R: G \rightarrow D is a smooth map called the reset map,
- f is a vector field, i.e., $\dot{x} = f(x)$ or $\dot{x} = f(x, u)$, in which case we call \mathcal{H} a controlled hybrid system on D.
- U is the set of admissible control inputs.

In the subsequent section, we will discuss domain, guard and reset map for a compass robot.

Hybrid Flows: A hybrid flow (or execution) is a tuple

$$\mathcal{X}^{\mathcal{H}} = (\Lambda, \mathcal{J}, \mathcal{X}) \quad (2-2)$$

where

$\Lambda = \{0, 1, 2, \dots\} \subset \mathbb{N}$ is a finite or infinite indexing set.

$\mathcal{J} = \{I_i\}_{i \in \Lambda}$ is a hybrid interval where $I_i = [\tau_i, \tau_{i+1}]$ if $i, i+1 \in \Lambda$ and $I_{N-1} = [\tau_{N-1}, \tau_N]$ or $[\tau_{N-1}, \tau_N)$ or $[\tau_{N-1}, \infty)$ if $|\Lambda| = N$, N is finite. Here, $\tau_i, \tau_{i+1}, \tau_N \in \mathbb{R}$ and $\tau_i \leq \tau_{i+1}$

$\mathcal{X} = \{x_i\}_{i \in \Lambda}$ is a collection of integral curves of f , i.e., $\dot{x}_i(t) = f(x_i(t))$ for all $i \in \Lambda$.

We require that the following conditions hold for every $i, i+1 \in \Lambda$,

$$\begin{aligned} x_i(\tau_{i+1}) &\in G \\ R(x_i(\tau_{i+1})) &= x_{i+1}(\tau_{i+1}) \\ \dot{x}_i(\tau) &= f(x_i(\tau)), \quad x_i(\tau) \notin G \end{aligned} \quad (2-3)$$

The initial condition for the hybrid flow is $c_0(\tau_0)$

Hybrid Periodic Orbits: In the compass robot, walking gait is represented as periodic orbit. A hybrid flow $\mathcal{X}^{\mathcal{H}} = (\Lambda, \mathcal{J}, \mathcal{X})$ of the hybrid system \mathcal{H} is periodic if the following conditions are satisfied Ames and Gregg (2007):

$$\begin{aligned} \Lambda &= \mathbb{N}, \\ \lim_{i \rightarrow \infty} \tau_i &= \infty, \\ x_i(\tau_i) &= x_{i+1}(\tau_{i+1}) \text{ for all } i \in \Lambda. \end{aligned}$$

A hybrid periodic orbit $\mathcal{O} \subset D$ is a subset of D such that:

$$\mathcal{O} = \bigcup_{i \in \mathbb{N}} \{x_i(t) : t \in I_i\} \quad (2-4)$$

For some periodic orbit flow $\mathcal{X}^{\mathcal{H}}$.

2.2 Dynamic Modeling of a Compass Robot

2.2.1 2-D Compass robot

Compass robot model represents the most elementary and simplest model of bipedal walking robots. The robot consists of two rigid legs which are connected by a hinge at the hip. The motion is 2-dimensional. The compass robot, its model derivation is explained in Goswami et al. (1996).

The gait of the robot consists of two stages: 1- Swing Stage. 2- Transition Stage. In the swing stage, the robot hip pivots around the contact point of the support leg and the ground. Another leg, called the swing or non-support leg swings forward. In the transition stage which occurs instantaneously, the swing leg impacts the ground and becomes the new stance leg. It is assumed that the impact of the swing leg with the ground is inelastic i.e. the robot configuration remains unchanged and the force in the impulse does not change the shape of the robot.

The configuration space for the compass robot is $Q_{2D} = \mathbb{R}^2$ with coordinates $\theta = [\theta_{ns}, \theta_s]^T$ where θ_{ns} is the angle of the non-stance leg from the vertical and θ_s is the angle of the stance leg from the vertical line. The state vector is represented by $q =$

$[\theta, \dot{\theta}]^T = [\theta_{ns}, \theta_s, \dot{\theta}_{ns}, \dot{\theta}_s]^T$. Fig (2-1) is demonstrated a configuration of a compass robot waking down the slope ground passively.

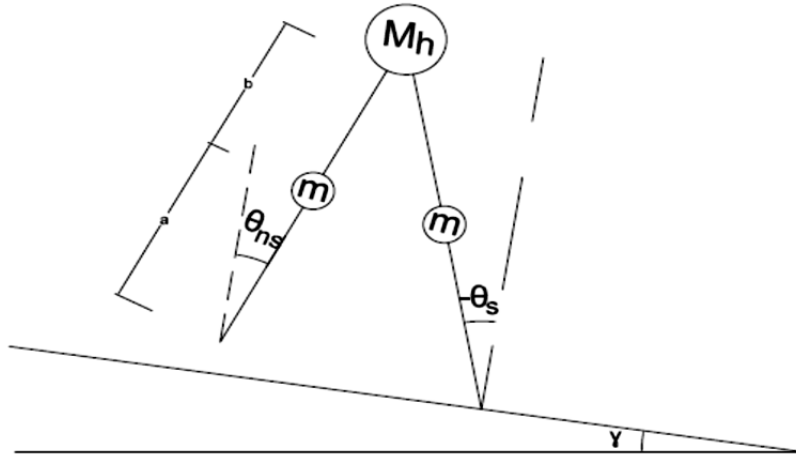


Fig. 2-1. Model of a compass robot

2.2.2 Lagrangian equation of motion

The compass biped resembles a double pendulum such that one link (the stance link) is fixed to the ground and another link (the swing or non-stance leg) swing freely about the hip. When the non-stance leg hits the ground, the velocities of the stance leg and non-stance leg change and can be determined according to conservation of momentum. The equation of the compass robot can be derived from the Euler-Lagrange equations. The Lagrangian $L: TQ \rightarrow \mathbb{R}$ modeling the mechanical system is:

$$L(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} - V(\theta) \quad (2-5)$$

where $M(\theta)$ is the inertia matrix and $V(\theta)$ is the potential energy. Euler-Lagrange equation without control input can be written as follows:

$$M(\theta)\ddot{\theta} + N(\theta, \dot{\theta})\dot{\theta} + g(\theta) = 0 \quad (2-6)$$

where $N(\theta, \dot{\theta})$ is the Coriolis matrix and $g(\theta) = \frac{\partial V_{2D}(\theta)}{\partial \theta}$ is the gravity matrix. The lagrangian vector field f_L can be written as follows:

$$(\dot{\theta}, \ddot{\theta}) = f_L(\theta, \dot{\theta}) = \begin{pmatrix} \dot{\theta} \\ -M^{-1}(\theta)[N(\theta, \dot{\theta})\dot{\theta} + g(\theta)] \end{pmatrix} \quad (2-7)$$

$$M(\theta) = \begin{pmatrix} mb^2 & -mlb\cos(\theta_s - \theta_{ns}) \\ -mlb\cos(\theta_s - \theta_{ns}) & M_H l^2 + m(l^2 + a^2) \end{pmatrix} \quad (2-8)$$

$$N(\theta, \dot{\theta}) = \begin{pmatrix} 0 & -mlb\dot{\theta}_s \sin(\theta_s - \theta_{ns}) \\ -mlb\dot{\theta}_{ns} \sin(\theta_s - \theta_{ns}) & 0 \end{pmatrix} \quad (2-9)$$

$$g(\theta) = \begin{pmatrix} mgbs\sin\theta_{ns} \\ -(M_H l + m(a + l)g\sin\theta_s) \end{pmatrix} \quad (2-10)$$

$$u = \begin{pmatrix} u_H \\ u_s \end{pmatrix} \quad (2-11)$$

See Goswami et al. (1996) for detailed derivation of $M(\theta)$, $N(\theta, \dot{\theta})$ and $g(\theta)$.

2.2.3 Unilateral constraint and domain

As mentioned in the previous section, the gait model consists of discrete phases. The dynamic model in each phase depends on the contact between the feet and the ground. Since the ground cannot pull on the foot, unilateral constraints exist on the forces and moments exerted by the ground on the foot J.W. Grizzle et al. (2010). In the compass robot, unilateral constraint is defined as the distance of the swing leg above the ground. While this constraint is satisfied, the robot is in the swing stage, so the robot

dynamics developed according to the given phase. When the constraint is violated, i.e., the distance of the swing leg and ground become zero, impact occurs and the phase and the dynamic model will change. The unilateral constraint is shown as $h: Q \rightarrow \mathbb{R}$ such that $h^{-1}(0)$ is a manifold, i.e., 0 is a regular value of h .

The domain and guard are constructed by utilizing the unilateral constraint that the swing leg is not allowed to pass through the ground. The domain and the guard of the system can be described as follows:

$$D = \{(\theta, \dot{\theta}) \in \mathbb{R}^4: h(\theta) \geq 0\} \quad (2-12)$$

$$G = \left\{ (\theta, \dot{\theta}) \in \mathbb{R}^4: h(\theta) = 0, \left(\frac{\partial h(\theta)}{\partial \theta} \right)^T \dot{\theta} < 0 \right\} \quad (2-13)$$

The guard of the system is the subset in which the unilateral constraint is violated and impact happens. In this thesis the unilateral constraint for a walking robot on the slope can be formulated as $h(\theta) = \theta_{ns} + \theta_s + 2\gamma$. For a walking robot on a flat ground where $\gamma = 0$, it results in $h(\theta) = \theta_{ns} + \theta_s$.

J.W. Grizzle et al. (2010) considered the guard as switching surface $S \subset \mathcal{X}$ at which solutions of the differential equation undergo a discrete transition and a re-initialization of the differential equation in the next period occurs.

2.2.4 Impact equations

The impact between the swing leg and the ground is modeled as a contact between two rigid bodies. Rigid impact models can be used to obtain an expression for the velocity of the legs after impact in terms of the velocity and position before the impact.

The basic assumptions for impact model as stated by J. W. Grizzle, Abba, and Plestan (2001) are as follows: 1) the impact takes place instantaneously; 2) the external forces during the impact can be represented by impulses; 3) impulsive forces change the velocities of the generalized coordinates, but the positions remain constant; and 4) the torques supplied by the actuators are not impulsive.

We denote the configuration before impact by $\theta^- = \left(\theta(T^-), \dot{\theta}(T^-)\right)^T$ and after impact by $\theta^+ = \left(\theta(T^+), \dot{\theta}(T^+)\right)^T$. By balancing momentum at impact, the velocities after impact can be obtained as follows Goswami et al. (1996):

$$\dot{\theta}(T^+) = H(\theta(T^-))\dot{\theta}(T^-) \quad (2-14)$$

if $\alpha := |\theta_{ns} - \theta_s|$, then

$$H = Q(\alpha)^{-1}P(\alpha) \quad (2-15)$$

$$Q(\alpha)\dot{\theta}(T^+) = P(\alpha)\dot{\theta}(T^-) \quad (2-16)$$

$P(\alpha)$ and $Q(\alpha)$ are the matrices which relates the pre-impact and post-impact angular velocities of the robot. These matrices are computed based on the conservation of momentum at the impact. It is shown in Goswami et al. (1996) that $P(\alpha)$ and $Q(\alpha)$ can be computed as follows:

$$P(\alpha) = \begin{bmatrix} -mab & -mab + (M_h l^2 + 2mal) \cos 2\alpha \\ 0 & -mab \end{bmatrix} \quad (2-17)$$

$$Q(\alpha) = \begin{bmatrix} mb^2 - mbl \cos 2\alpha & (ml^2 + ma^2 + M_h l^2) - mbl \cos 2\alpha \\ mb^2 & -mbl \cos 2\alpha \end{bmatrix} \quad (2-18)$$

$\dot{\theta}(T^+)$ which is the velocity after impact, can be used to re-initialize the model after impact. In order to do this, a change of coordinates is necessary since the former swing leg is now the stance leg. The position after impact can be written as follows:

$$\theta(T^+) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \theta(T^-) \quad (2-19)$$

The final result for expression $q^+ = [\theta^+, \dot{\theta}^+]^T$ in terms of $q^- = [\theta^-, \dot{\theta}^-]^T$ is:

$$q^+ = R(q^-) \quad (2-20)$$

$$R = \begin{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} & 0 \\ 0 & H(\alpha) \end{bmatrix} \quad (2-21)$$

Moreover, it follows from the robot geometry that at the impact moment (when both legs are on the ground):

$$\theta_{ns}^- + \theta_s^- = -2\phi \quad (2-22)$$

$$\theta_{ns}^- - \theta_s^- = 2\alpha$$

2.2.5 Normalized equations

The equation of walking robot can be normalized with respect to mass and length properties. The normalization is advantageous in terms of study of bifurcation maps of the robot. The swing leg and impact equation is derived in Goswami et al. (1996) and can be rewritten as follows:

$$M_n(\theta)\ddot{\theta} + N_n(\theta, \dot{\theta})\dot{\theta} + \frac{1}{a}g_n(\theta) = 0 \quad (2-23)$$

$$Q_n(\alpha)\dot{\theta}(T^+) = P_n(\alpha)\dot{\theta}(T^-) \quad (2-24)$$

where

$$M_n(\theta) = \begin{pmatrix} \beta^2 & -(1 + \beta)\beta \cos 2\alpha \\ -(1 + \beta)\beta \cos 2\alpha & \mu(1 + \beta)^2 + (1 + (1 + \beta)^2) \end{pmatrix} \quad (2-25)$$

$$N_n(\theta, \dot{\theta}) = \begin{pmatrix} 0 & (1 + \beta)\beta \dot{\theta}_s \sin(\theta_s - \theta_{ns}) \\ -(1 + \beta)\beta \dot{\theta}_{ns} \sin(\theta_s - \theta_{ns}) & 0 \end{pmatrix} \quad (2-26)$$

$$g_n(\theta) = \begin{pmatrix} g\beta \sin \theta_{ns} \\ -(\mu(1 + \beta) + (1 + (1 + \beta)))g \sin \theta_s \end{pmatrix} \quad (2-27)$$

$$Q_n(\alpha) = ma^2 \begin{pmatrix} -\beta & -\beta + (\mu(1 + \beta)^2 + 2(1 + \beta)) \cos 2\alpha \\ 0 & -\beta \end{pmatrix} \quad (2-28)$$

$$P_n(\alpha) = ma^2 \begin{pmatrix} \beta(\beta - (1 + \beta) \cos 2\alpha) & (1 + \beta)((1 + \beta) - \beta \cos 2\alpha) \\ \beta^2 & \dots + 1 + \mu(1 + \beta)^2 \\ & -\beta(1 + \beta) \cos 2\alpha \end{pmatrix} \quad (2-29)$$

where: $\mu = \frac{M_h}{m}$ = mass ratio, $\beta = \frac{b}{a}$ = Length ratio.

It is obvious that the equation of motion is just dependent on μ and β . It is shown in Goswami et al. (1996) that gait characteristics of a robot with arbitrary masses m and M_h can always be deduced from those of a robot whose masses are in the same proportion. In addition, gait characteristics of a robot with arbitrary lengths a and b can always be deduced from those of a robot whose lengths are in the same proportion.

2.3 Controlled Symmetries Applied to the Robot for Walking on the Flat Ground

A compass robot walking down a slope of γ degree by applied control forces can be represented as $\mathcal{H}^\gamma = (D^\gamma, G^\gamma, R, f^\gamma, U)$ where U is the set of admissible controls. Using the controlled Euler-Lagrange equation, the dynamic of the robot is given as follows:

$$M(\theta)\ddot{\theta} + N(\theta, \dot{\theta})\dot{\theta} + g(\theta) = BU \quad (2-30)$$

Control inputs U are introduced by controlled symmetries. B is a linear transformation which converts from relative to absolute coordinates. Note that this control law requires full action (the robot is not under-actuated) so B is full rank.

Controlled Symmetries were introduced by Spong and Bullo (2002) and later was completed by (Spong & Bullo, 2005). This control law works by shaping the potential energy of the associated robot to allow stable walking on flat ground. The idea of using this control law by Spong and Bullo (2005) is symmetries of hybrid system \mathcal{H} which is used for rotating the world by a “group action” to allow walking on flat ground. Group action Φ which rotates the world by γ degree can be represented as:

$$\Phi(\gamma, \theta) = (\theta_{ns} - \gamma, \theta_s - \gamma)^T \quad (2-31)$$

Here, $\gamma \in \mathbb{S}^1$ is the slope of the ground. In Fig 2-2, applying group action $\Phi(\gamma, \theta)$ to rotate the world by γ degrees in order to allow walking on the flat ground is shown. Using this group action, the following control feedback law can be formed:

$$U = K^\gamma(\theta) = B^{-1} \left(V(\theta) - V(\Phi(\gamma, \theta)) \right) \quad (2-32)$$

where $V(\theta)$ is the potential energy of the robot walking on the slope γ and $V(\Phi(\gamma, \theta))$ is the potential energy of the system with applied group action $\Phi(\gamma, \theta)$. Lagrangian and its associated vector field for a system with controlled symmetries are as follows:

$$L(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} - V(\Phi(\gamma, \theta)) \quad (2-33)$$

$$(\theta, \dot{\theta}) = f^\gamma(\theta, \dot{\theta}) = f(\theta, \dot{\theta}, K^\gamma(\theta)) \quad (2-34)$$

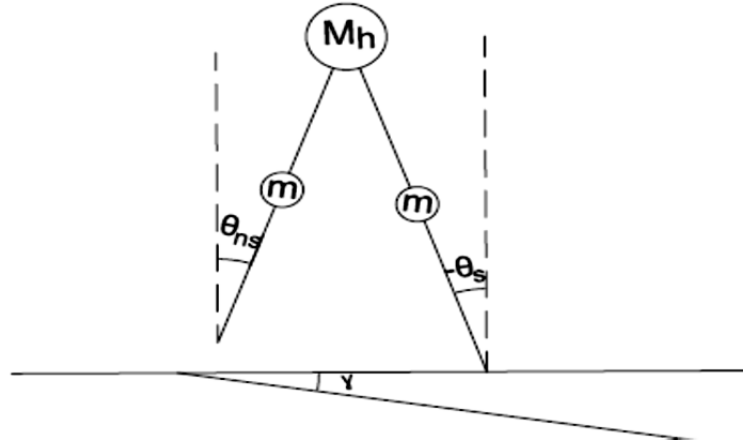


Fig. 2-2. Rotating world by group action: walking robot on the flat ground

Now, consider the following hybrid system:

$$\mathcal{H}^s = (D^0, G^0, R, f^0, U) \quad (2-35)$$

where D^0, G^0 are the domain and guard for the slope “0” degree (walking on the flat ground). According to Spong and Bullo (2005) \mathcal{H}^s can be related to the hybrid system \mathcal{H} (hybrid system without control input represented in formula (1-1)) as if $\mathcal{X}^{\mathcal{H}} = (\Lambda, \mathcal{J}, (\theta, \dot{\theta}))$ is the hybrid flow of system \mathcal{H} , then $\mathcal{X}^{\mathcal{H}^s} = (\Lambda, \mathcal{J}, (\Phi(\gamma, \theta), \dot{\theta}))$ is the hybrid flow of \mathcal{H}^s . It means if \mathcal{H} walks on the slope, then \mathcal{H}^s can walk on the flat ground.

2.4 Simulation Results for Compass Robot

The compass robot model is shown in Fig 2-1. The robot’s leg is a cylindrical bar with a uniform mass. The dimension and mass of the robot are as follows:

$M_h = 4$ kg: Mass at the hinge of the compass robot

$m = 2\text{kg}$: Mass of each leg of the robot

$l = 1\text{m}$: Length of each leg of the robot

$d = 0.0285$: Diameter of the robot's leg

$a = b = 0.5\text{m}$: Distances of the legs' centers of mass from respective foot and hinge

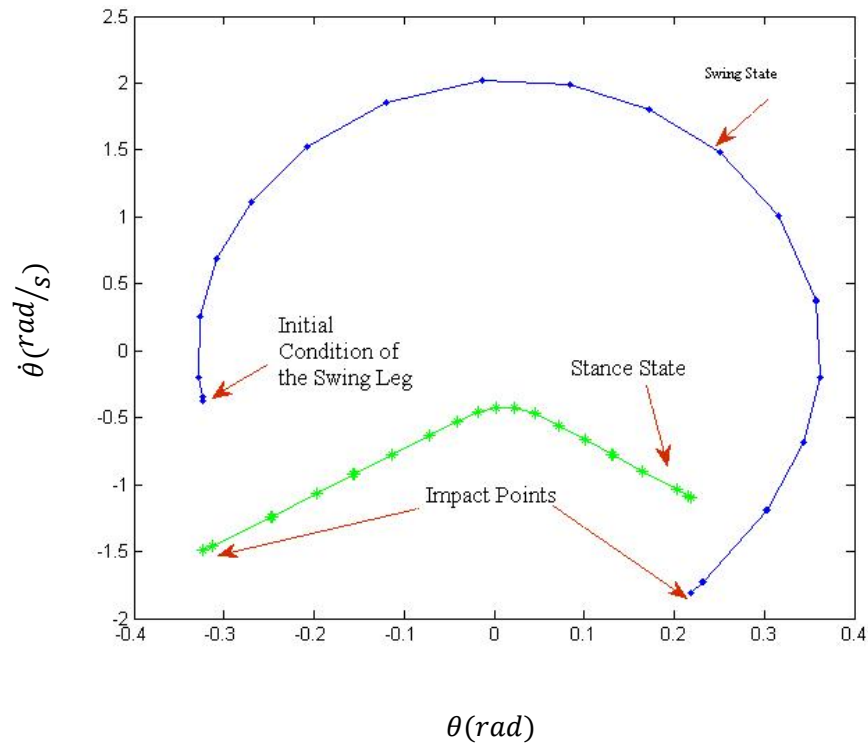
In this section, the gait of the compass robot is shown by phase portrait shown in Fig 2-3. The phase portrait involves the displacement and velocity of just one leg. The leg alternatively becomes the swing leg and the stance leg. The upper half of the cycle depicts the motion of the swing leg and the lower half of the cycle depicts the motion of the stance leg. Therefore, a complete walking cycle is formed from the limit cycles of the two legs in two different states. Since, this is a symmetric gait, the phase portrait of another leg is exactly the same.

2.4.1 Passive gait on the slope ground

The limit cycle of the compass robot walking on a slope ground passively is shown in Fig. (2-3). It can be seen from Fig (2-3) that, when impact happens, the angular velocity of the swing leg and stance leg have a sudden change but the position of two legs do not change. The legs just switch.

Initial Condition: $q_0 = [\theta_0, \dot{\theta}_0]^T$; $\gamma = 3^\circ$;

Guard: $h(\theta) = \theta_{ns} + \theta_s + 2\gamma$,



As it can be seen in Fig 2-3, when impact happens, the angular velocities of the stance leg and swing leg change instantaneously; however the angles of two legs are the same and do not change. Fig 2-3 can be considered as the complete walking limit cycle since the gait is symmetric.

2.4.2 Walking robot on the flat ground by controlled symmetries

As it was mentioned in section (2.3), if a compass robot walks down passively on a slope stable fashion, the controlled symmetries can result in the same stable walking on the flat ground. The law for controlled symmetries was given in (2-32). In this thesis, we specify independent control torques to act at the hip and the stance leg foot, so the

Matrix B is given as $B = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix}$. After applying controlled symmetries, the guard and the initial condition should be changed as follows:

$$\gamma = 0^\circ, \text{ Guard: } h(\theta) = \theta_{ns} + \theta_s = 0$$

$$\text{Initial Condition: } [\theta_0, \dot{\theta}_0]^T = [\theta_{0ns} + \gamma, \theta_{0s} + \gamma, \dot{\theta}_{0ns}, \dot{\theta}_{0s}]^T$$

The phase portrait of the robot on the flat ground is illustrated in Fig (2-4):

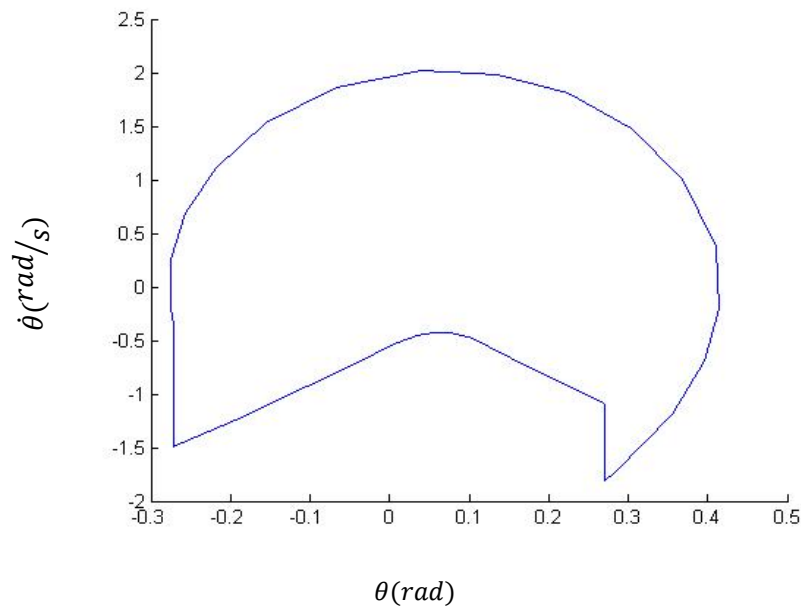


Fig. 2-4. Phase portrait of the robot walking on the flat ground by controlled symmetries

As it can be seen in Fig. 2-4, the limit cycle or phase portrait of the robot walking on the flat ground by controlled symmetries is exactly the same as the limit cycle of robot walking on the slope passively shown in Fig. 2-3. It means that applying controlled symmetries result in desired and predicted results.

2.5 Influence of Robot Parameters on the Robot's Gait and Bifurcation Maps

In this section, the effect of the robot parameters on the gait is studied. Two parameters, including mass ratio μ and length ratio β , are considered as effective variables on the robot's walking pattern. As it was shown in Section (2-2-5), the robot's dynamic can be normalized so that the walking equations can be written as a function of just 2 parameters μ and β . It helps to study the walking model based on two parameters μ and β . In the previous section, the values assigned to β and μ were "1" and "2" respectively. For higher values of these parameters, the robot first exhibits double period walking and then it will be four periodic walking and for very high values, it will be chaotic walking. The study of bifurcation map helps in designing the compass robot and selecting different components for the robot. In the study of bifurcation map, the slope angle γ on which controlled symmetries is developed is considered $\gamma = 3^\circ$ and $\gamma = 4^\circ$. In Fig. 2-5 and 2-6, the change of the half angle between the legs $\alpha = \frac{\theta_{ns}^- - \theta_s^-}{2}$ as function of β when $\mu = 2$ is demonstrated. In Fig. 2-7 and 2-8, the change of the α versus μ when $\beta = 1$ is represented.

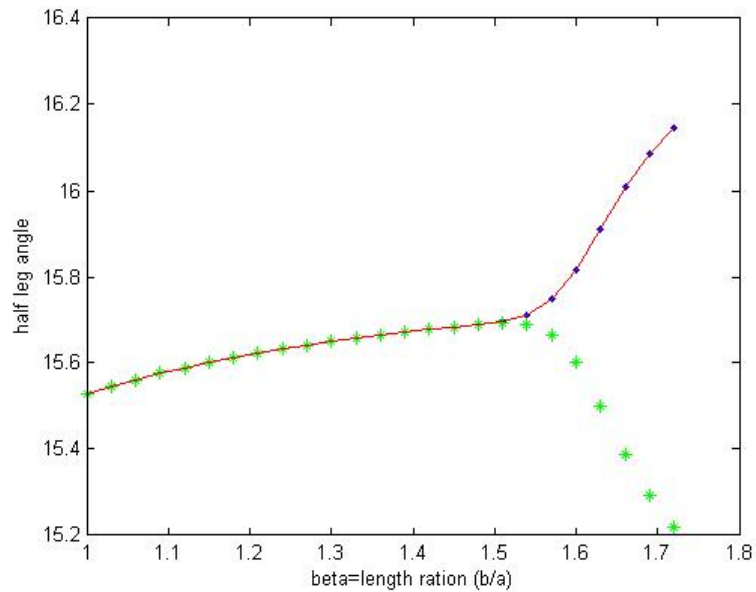


Fig. 2-5. Bifurcation diagram, α as a function of β for slope angle $\gamma=3^\circ$ and $\mu=2$

In Fig 2-5, the green line shows the value of α for the first step. The blue line corresponds to the value of α for the second step and the red line corresponds to the third step. It can be seen that for $\gamma=3^\circ$ and $\mu = 2$, changes in β , the length ratio, affects the walking model. It can be seen that for $\beta > 1.45$, there are two values of α . It means that the gait is now two-periodic i.e., the robot will take one short step in which α is smaller and one larger step with larger α angle. For example if $\beta = 1.6$, there are two values of α ; $\alpha = 1.55^\circ, \alpha = 1.8^\circ$. The shorter step corresponds to $\alpha = 1.55^\circ$ and the longer step is related to $\alpha = 1.8^\circ$.

:

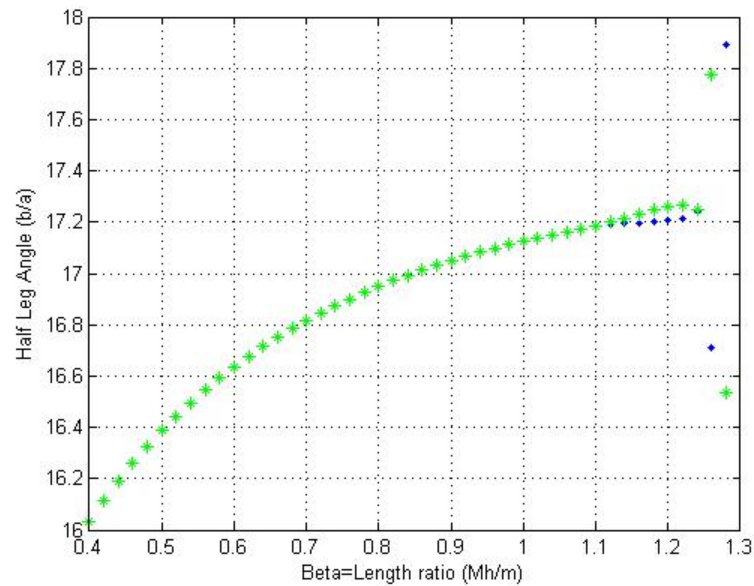


Fig. 2-6. Bifurcation diagram, α as a function of β for slope angle $\gamma=4^\circ$ and $\mu=2$

In Fig 2-6, the changes of α as the function of β for slope angle $\gamma=4^\circ$ is depicted. As it can be seen, the walking model is very sensitive to the length ratio. The gait becomes two periodic for $\beta = 1.1^\circ$ and for very small changes, at $\beta = 1.2^\circ$, the walking become four periodic. It can be seen that at $\beta = 1.2^\circ$, there is a sudden change in the value of α . The reason is that at this point, another bifurcation has happened and the walking becomes four periodic. (The four periodic is not shown in this plot and is not studied in this thesis).

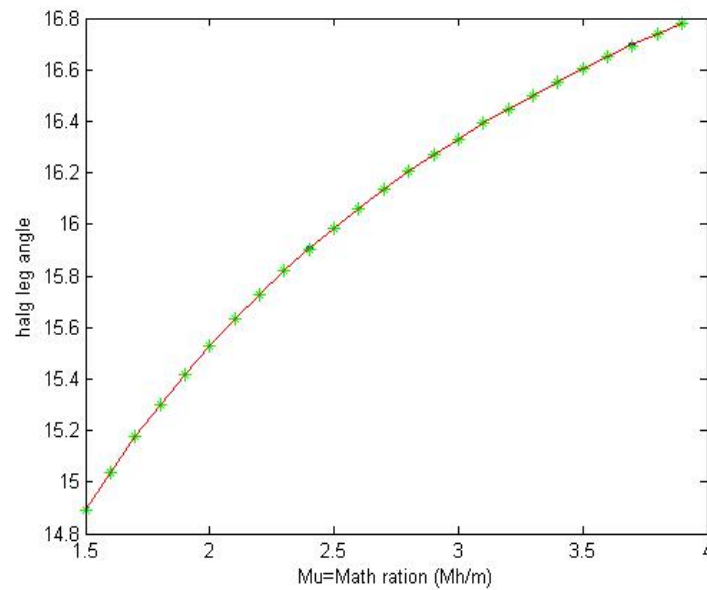


Fig. 2-7. Bifurcation diagram, α as a function of μ for slope angle $\gamma = 3^\circ$ and $\beta = 1$

Fig 2-7 shows that for slope angle $\gamma = 3^\circ$ and $\beta = 1$, changing the mass ratio does not affect on the walking model. As it can be seen, the gait remains one-periodic for different values of μ and no bifurcation has happened. It means that for $\gamma = 3^\circ$ and $\beta = 1$, the robot is not sensitive to mass ratio at all.

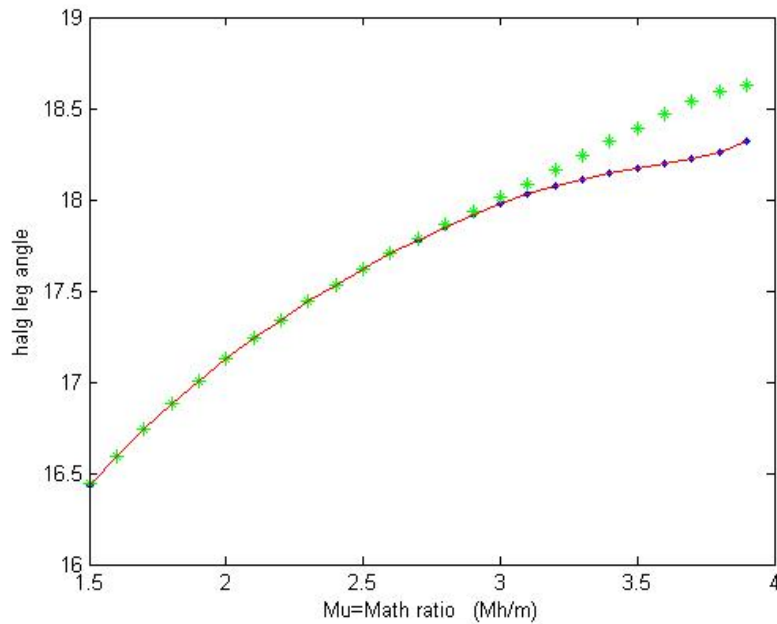


Fig. 2-8. Bifurcation diagram, α as a function of μ for slope angle $\gamma = 4^\circ$ and $\beta = 1$

As it can be seen from the Fig 2-8, the gait of the robot is double periodic for $\mu > 2.7$. In compare with Fig 2-7, it is shown that for slope angle $\gamma = 4^\circ$, the robot is sensitive to the mass ratio. When the mass ratio μ increases, the robot become double-periodic. It means that if $\gamma = 4^\circ$, the design of the compass robot should be consistent so that the bifurcation does not happen.

It has to be mentioned that in the above Fig, the gait of the robot was shown just for 2 periodic case. For higher values of μ and β , the gait will be 4-periodic, 8-periodic and finally it will become chaotic motion. By studying the bifurcation maps, the sensitivity of the robot to length ratio and mass ratio can be checked which is very helpful for designing a robot.

3 STABILITY ANALYSIS OF A COMPASS ROBOT

3.1 Poincare Map

The concept of gait stability for a walking robot is hard to define but is necessary for performance analysis. As mentioned in the previous section, robot motion as a hybrid system exhibits periodic behavior. The stability analysis of periodic, discrete motion is determined by the Poincare map. The concept of Poincare map for stability analysis of periodic orbits with event-based switching mechanism has been studied in many references conducted by Morris and Grizzle (2005), J.W. Grizzle et al. (2010), Westervelt et al. (2007) and Hiskens (2001).

A Poincare map samples the flow of a periodic system for every period. A gait is stable if, starting from a steady periodic trajectory, any finite disturbance leads to the same trajectory or to another nearby trajectory of similar shape. Consider a periodic trajectory $\phi(x_0, t)$ which is the solution of $\dot{x} = f(x)$ with initial condition x_0 . The Poincare map of the trajectory $\phi(x_0, t)$ is shown in Fig 3-1. It shows that $\phi(x_0, t)$ intersects the hyperplane Σ at point x^* which is referred to as the fixed point. The trajectory emanating from x^* will encounter the hyperplane Σ again after one period at the vicinity of x^* . The following mapping can be defined for intersection points:

$$P(x_i^*) := \phi(x_i^*, \tau(x_i^*)) \quad (3-1)$$

where $\tau(x_i)$ is the period time and P is the Poincare map.

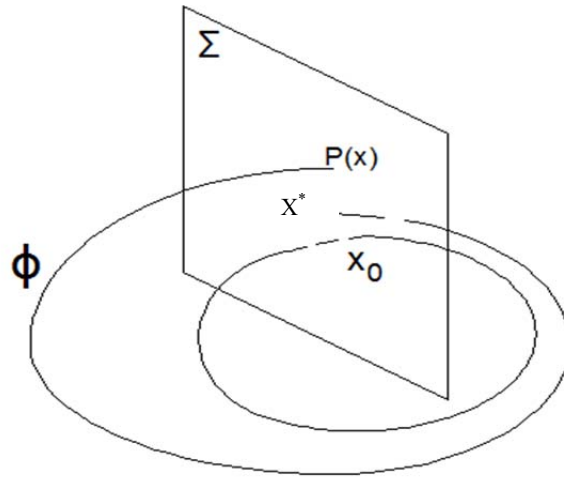


Fig. 0-1. Poincare map

For the compass robot, which is a system with an impulse events, the hyperplane Σ (Poincare section) is selected as the switching surface S on which the solution of the differential equation undergo a discrete transition that is an instantaneous re-initialization of the differential equation in the next period. Periodic orbit should be transversal to S so $\frac{\partial H}{\partial x}(x^*)f(x^*) \neq 0$ i.e., the vector field f is not tangent to S at x^* . Poincare map $P: S \rightarrow S$ represents the evolution of the robot's motion after the impact occurs. So Equation (3-1) can be rewritten as follows:

$$P(x_i^*) = \phi(R(x_i), T_1(R(x_i))) \quad (3-2)$$

where $x_i^* = R(x_i)$ is the re-initializing point of the trajectory for the next period. $i=1,2,\dots,n$ is the number of cycles in the periodic motion (number of steps by the robot)

and $T_1: \mathcal{X} \rightarrow \mathbb{R}$ is the period time which is the time from initializing to the first intersection with Surface S and is given by:

$$T_1(x_i) := \inf\{t \geq 0 \mid \phi(t, x_0) \in S\}, \text{ if } \exists t \text{ such that } \phi(t, x_0) \in S \quad (3-3)$$

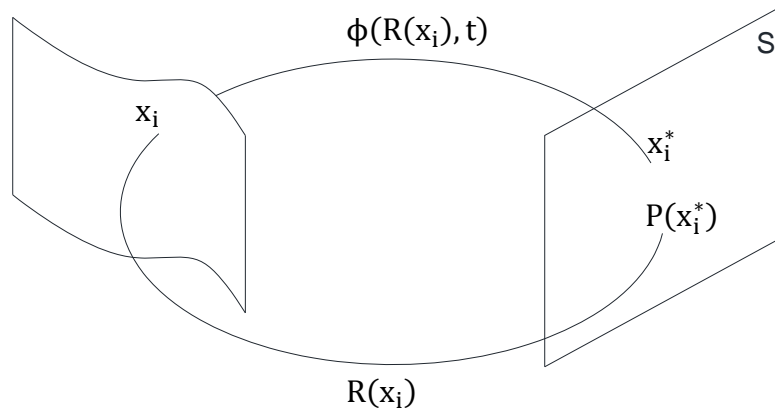


Fig. 0-2. Poincare map for a system with impulse

The theory of Poincaré Map for hybrid systems is given by Morris and Grizzle (2005) as follows: If a system with impulse effect $\mathcal{H} = (D, G, R, f)$ has a periodic orbit \mathcal{O} with $x^* := \mathcal{O} \cap S$ and \mathcal{O} is transversal to S , then the following are equivalent:

- (a) x^* is exponentially stable (resp. asymptotically stable or stable) fixed point of P .
- (b) \mathcal{O} is exponentially stable (resp. asymptotically stable or stable) periodic orbit.

The results in Westervelt et al. (2007) show that if \mathcal{O} is transversal to S and the hybrid system $\mathcal{H} = (D, G, R, f)$ is continuously differentiable system, then the Poincaré map P is differentiable at a fixed point x^* and stability of the system can be checked by the eigenvalues of Jacobian matrix that is linearized P at fixed point x^* . Hiskens (2001)

referred to the linearized P as trajectory sensitivity matrix $\Phi_x(x^*, T)$ which can be written as follows:

$$DP(x^*) = \Phi_x(x^*, T) = \frac{\partial \phi(x^*, T_I(x^*))}{\partial x^*} \quad (3-4)$$

One eigenvalue of $\Phi_x(x^*, T)$ is always “1” which is in the direction of the vector field $\phi(x^*)$. Remaining eigenvalues of $\Phi_x(x^*, T)$ which are referred to as characteristic multipliers have to be smaller than “1” to guarantee the stability of the system.

3.2 Simulation Results

In this section, the stability of the compass robot walking on the flat ground by controlled symmetries is analyzed by using the Poincare map. Note that the stability of the passive walking of the compass robot on the slope is equivalent to the stability of the controlled compass robot on the flat ground. As it was mentioned in the previous section, the surface Σ is considered as the switching surface S in which the guard condition happens. So, the guard condition $\theta_{ns} + \theta_s = 0$ forms the triggering surface and the initial condition considered as $x_0 = [\theta_{0ns} - \gamma, \theta_{0s} - \gamma, \dot{\theta}_{0ns}, \dot{\theta}_{0s}]^T$ which is the point immediately after the impact event, is considered as the fixed point x^* . Since, it is not easy to derive the flow ϕ , we just compute the eigenvalues of the Poincare map around the fixed point as:

$$DP(q^*) = \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_1}{\partial x_3} & \frac{\partial \phi_1}{\partial x_4} \\ \frac{\partial \phi_3}{\partial x_1} & \frac{\partial \phi_3}{\partial x_3} & \frac{\partial \phi_3}{\partial x_4} \\ \frac{\partial \phi_4}{\partial x_1} & \frac{\partial \phi_4}{\partial x_3} & \frac{\partial \phi_4}{\partial x_4} \end{bmatrix} \quad (3-5)$$

The eigenvalues of the linearized Poincare map for the robot with dimensions given in Section (2.4) after one period are obtained as follows:

$$-0.2226 \pm 0.5304i, 0.1307$$

The derivative of the trajectory about x_2 is not computed since its eigenvector is tangent to $\phi(x^*)$ that corresponds to eigenvalues “1”. Other eigenvalues are characteristic multipliers. Since all of other eigenvalues are smaller than one, the robot gait is stable under the assumed condition.

4 DESIGN OF A COMPASS ROBOT WITH INTERACTING COMPONENTS

The design process of a system requires integrating multiple components and investigating the interaction between them. Modeling of complex systems becomes more difficult due to the interactions between these subsystems. The objective of this section is detailed design of a compass robot by considering the robot, motor and gear set which interact with each other during the design process.

The required control input has been evaluated based on the conceptual controlled symmetries represented in section 2. The motor and batteries are selected based on the required control input and the required torque for the robot's legs. The modeling of the system is improved by considering the torque-speed constraint in the motor. Adding this constraint of DC motor affects the gear and the walking model. Finally, different selection of the motor at the hip and foot of the robot, their effect on the bifurcation map and gait stability will be discussed. The final decision is made based on the cost, efficiency and the effect of the motor on the stability.

4.1 Power Transmission Selection and Modeling

Selecting the power transmission of a robot i.e., motor, gear and batteries is based on the required load estimated in the previous section. The power transmission components are the most expensive ones and have the most contribution to the weight of

the robot. In order to have fast walking, it is better to keep the weight minimum. The motor and gear products are selected from Maxon since they have been used in several other biped robots like the robots studied by Pedersen et al. (2006-2007).

4.1.1 Motor

DC motors have the advantage of being controlled easily since the motor speed is proportional to voltage and its torque is proportional to current. This characteristic creates the main specification of DC motors which is the relationship between the motor's speed and torque. This characteristic is depicted in Fig 4-1 and can be stated as follows:

$$\Delta\omega_m = \text{SPG} * \Delta T_m \quad (4-1)$$

$$\omega_m = \text{Nls} - \text{SPG} * T_m$$

where:

SPG: Speed/ Torque Gradient (rpm/Nm)

Nls: Non-Load Speed (rpm)

T_m : Developed Torque by Motor, N-m

ω_m : Angular Velocity of the Motor

Some important characteristics of the motor are as follows:

R_a : Armature Resistance (ohm)

i_a : Armature Current, A

V_a : Applied Armature Voltage, V

V_b : Back emf, V

J : Moment of Inertia of the Motor (kgm^2)

b : Viscous-Friction Coefficient of the Motor and Load

K_b : Back emf Constant (V/rpm)

K : Torque Constant (Nm/A)

NT : Nominal Torque (Nm)

T_m : Mechanical Time Constant

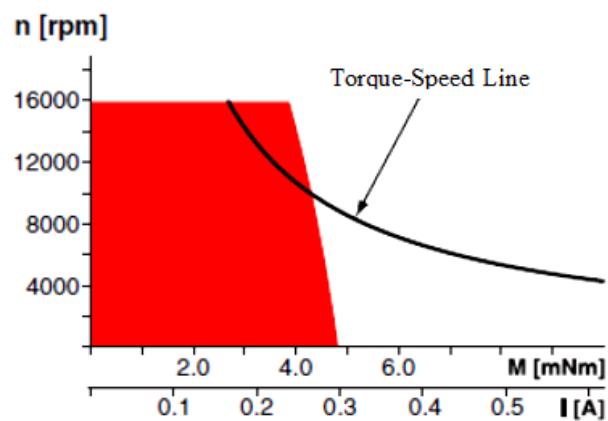


Fig. 4-1. DC motor diagram: workspace is under the torque-speed line

If the voltage is lowered, the torque-speed curve shifts downward. The motor can operate at any point under the torque-speed line but it should not exceed the maximum continuous torque and the maximum permissible speed to prevent overheating of the motor. This operation zone is shown as the red area in Fig 4-1.

The torque of the motor consists of two parts. The torque for rotating the motor and its components and the torque required by the gear which is considered as the load on the motor. The motor torque can be written as follows:

$$T_m = T + T_{load} = T + \left(\frac{N}{\eta}\right) * T_g \quad (4-2)$$

$$T = T_m - \left(\frac{N}{\eta}\right) * T_g$$

T : The torque for rotating the motor

T_{load} : The loaded torque used for rotating the gear train

T_g : The required torque applied via the gear train to the robot's leg

η : the efficiency of the gear train.

$$N = \frac{1}{n}$$

n : Reduction factor of the gear train

4.1.2 Gear

Selecting a proper gear train for motor is important. The proper gear for the robot should have the following qualities: high gear ratio, high efficiency, zero backlash and low weight. The required gear is also selected from Maxon Products.

For selecting the gear, it is noted that the output torque of the gear train should be equal to the control input needed by the robot's leg which is computed by controlled symmetries. It means:

$$T_g = u_{des} \quad (4-3)$$

u_{des} : Desired torque computed by control symmetry

T_g : Torque applied by the gear

4.1.3 Adding motor and gear to the robot

Applying controlled symmetries requires two control inputs at the hip and feet so two motors with their gear trains are required. Adding motor and gear at the hip and foot of the robot affects the moment of inertia of the leg and consequently affects the gait of the robot. So, selecting a combination of motor and gear which satisfies the gait requirement and at the same time has a reasonable price is very important. Fig 4-2 illustrates a Fig of the robot with motors.

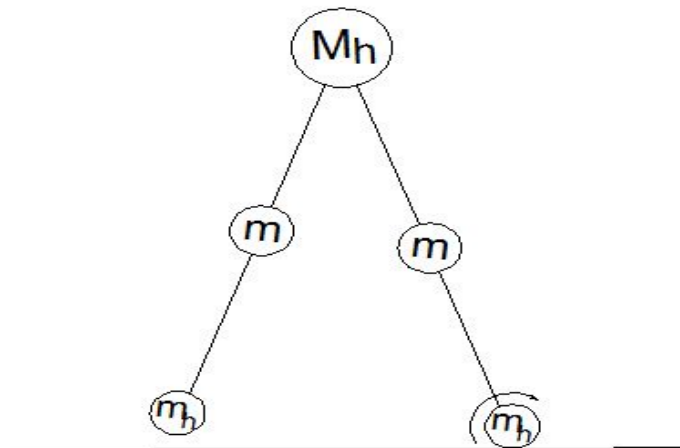


Fig. 4-2: Compass robot with motors on the foots and on the hips

In addition, determining the required speed of the motor for appropriate walking of the robot is very important. By deriving the speed equation of the motor, the speed controller of the motor can be designed properly. The dynamics equation of the motor is given by Ogata (2004) as follows:

$$J_{eq}\dot{\omega}_m + b_{eq}\omega_m = T \quad (4-4)$$

For the motor on the hip:

$$J_{eq} = J_{mhip} + N_1^2 * (J_{rest1} + J_{ghip}) \quad (4-5)$$

where:

J_{mhip} : Moment inertia of the motor at the hip

J_{ghip} : Moment inertia of the gear at the hip

$N_1 = \frac{1}{n_1}$, n_1 : Reduction factor of the gear at the hip

m_h : Weight of the motor + weight of the gear on the foot

m : Weight of the leg

J_{rest1} : Moment of inertia of rotational parts which consist of non-stance leg and the actuator on that.

The moment of inertia of rotational part can be calculated as follows:

$$J_{rest1} = M_{cent1} R_{cent1}^2 \quad (4-6)$$

$$M_{cent1} = m + m_h$$

where M_{cent1} is the total mass of the rotational part and R_{cent1}^2 is the radius of gyration of the rotational parts and can be stated as follows:

$$R_{cent1}^2 = \frac{mb^2 + m_h l^2}{M_{cent1}} \quad (4-7)$$

For the motor on the foot:

$$J_{eq} = J_{mank} + N_2^2 * (J_{rest2} + J_{gank}) \quad (4-8)$$

J_{mank} : Moment of inertia of the motor at the foot of the stance leg

J_{gank} : Moment of inertia of the gear at the foot of the stance leg

$N_2 = \frac{1}{n_2}$, n_2 : Reduction factor of the gear at the foot

M_h : Mass of the motor + Mass of the gear at the hip

J_{rest1} : Moment inertia of rotating part consisting of the stance leg, actuator on the hip, non-stance leg and actuator on that.

The moment inertia of rotating part can be calculated as follows:

$$J_{\text{rest2}} = M_{\text{cent2}} R_{\text{cent2}}^2 \quad (4-9)$$

$$M_{\text{cent2}} = 2m + M_h + m_h$$

where the R_{cent2}^2 is the radius of gyration which is calculated as follows:

$$R_{\text{cent2}}^2 = \frac{ma^2 + M_h l^2 + m(l^2 + b^2 - 2bl\cos\alpha) + m_h(l^2 + l^2 - 2l^2\cos\alpha)}{2m + M_h + m_h} \quad (4-10)$$

For both motors on the hip and foot:

$$b_{\text{eq}} = b_1 + N^2 * b_2 \quad (4-11)$$

b_1 : Viscous friction coefficient of the motor.

b_2 : Viscous friction coefficient between the leg and gear train (negligible).

4.2 Design of a Compass Robot with Interacting Components

The design of a compass robot includes developing the gait dynamics, motor's speed equation and gear dynamics, which relate to each other and exchanging data during the design process.

In Section 2, the dynamics of the robot gait and control strategy were discussed. In Section (4.1) the speed equation of the motor was derived. In this section, a general scheme design of the robot with interacting components is demonstrated. Fig 4-3 shows the design configuration concepts and related components.

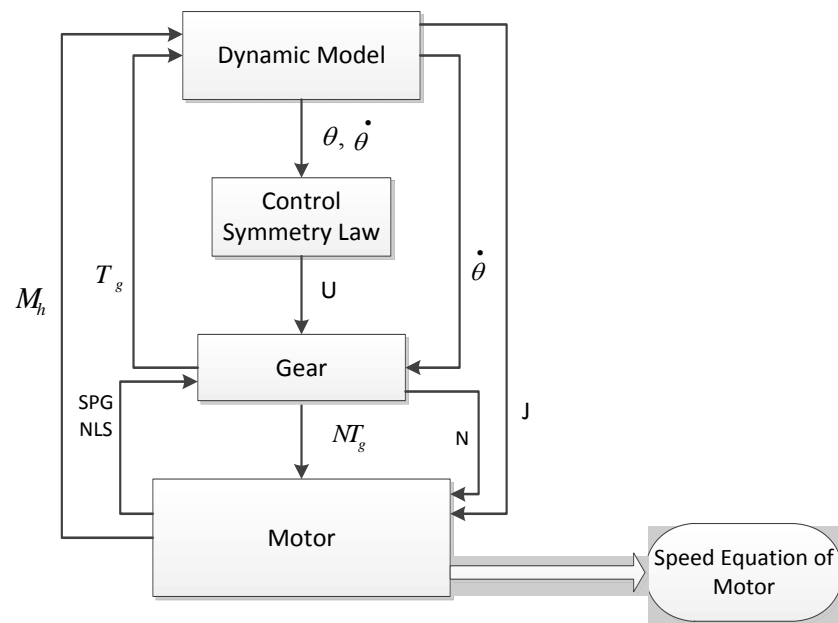


Fig. 4-3. Design of a compass robot with interacting elements

As it is shown in the above figure, speed equation of the motor is affected by the leg's moment of inertia, leg's angular velocity and load torque T_g . The modeling of the system is more complicated by considering the torque-velocity constraint of the motor.

By evaluating the interaction between the components and the torque-velocity constraint, the speed-equation of the motor can be derived more accurately. By considering all the contributing elements, a proper motor and gear have to be selected which satisfies the gait requirement.

4.3 Speed Equation of the Motor based on the Design Process

In this thesis, the relation between the speed and torque of the motor has been incorporated in the equation of the motor so that the changes of the motor's speed follow the proper torque-speed curve. In addition by following the design scheme shown in Fig 4-3, the effect of other components on the motor's dynamic is incorporated in the speed equation of the motor. By substituting (4-1) and (4-2) in (4-4), the speed equation of the motor can be written as follows:

$$J_{eq}\dot{\omega}_m + b_{eq}\omega_m + \left(\frac{N}{\eta}\right) * T_g = \frac{Nls - \omega_m}{SPG} \quad (4-12)$$

$$J_{eq}\dot{\omega}_m + \left(b_{eq} + \frac{1}{SPG}\right)\omega_m + \left(\frac{N}{\eta}\right) * T_g = \left(\frac{Nls}{SPG}\right) \quad (4-13)$$

As it has been mentioned T_g is calculated based on controlled symmetries law. In section (2-3), it was shown that controlled symmetries law which was computed by formula (1-32), is dependent on the gait variables, i.e. angle and angular velocity of the legs. So, the speed of the motor is also dependent to the gait variables as it was represented by Ames et al. (2006).

4.4 Simulation Results

In this section, we follow the design method illustrated in Fig 4-3 and the speed equation of the motor found in Section (4-3) and then check the motor performance for the given motor and gear. In the first step, the maximum required torque by the robot's leg can be calculated by simulation results in section (2-4). The power of the motor can be estimated by multiplying the leg's maximum required torque by the gear train's ratio and also by a safety factor. The Torque of the motor is computed by Equation (4-2) and the power and efficiency of the motor for a compass robot with characteristics given in Section (2-4) are obtained as follows:

$$P = T_m * \omega_m \quad (4-14)$$

$$\text{Eff} = \frac{P_{\text{out}}}{P_{\text{in}}} = \frac{T_m * \omega_m}{T_m * \omega_m + I * R_a} \quad (4-15)$$

Note that the selection of the motor and gear is done using trial and error since many variables are involved in the design including gear ratio, no load speed, torque constant, etc. For example any of the motors i.e., 8w, 10 w or 50 w can be used with proper selection of gear train's reduction ratio. The power of the motor can have a wide range depending upon the gear train's ratio.

4.5 Motor Selection

In the first step, a Maxon motor RE25 10w (Order No. 118746) and gear GS 45A, n=310 (Order No. 301185) are selected for the hip. Motor RE25 10kw (Order No. 118746) and gear GS 45A, n= 9 (Order No. 301179) are selected for the foot. The

detailed characteristics of the two motors and their gears are given in Appendix A and B. The performance curves of the selected motors at the hip and at the foot of the compass robot are obtained as follows:

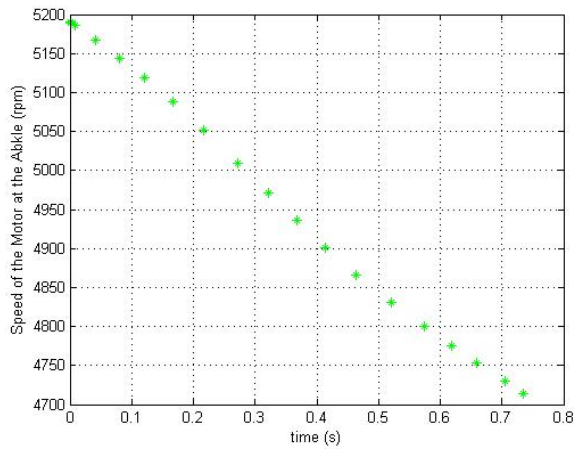


Fig. 4-4. Speed diagram of the motor at the foot

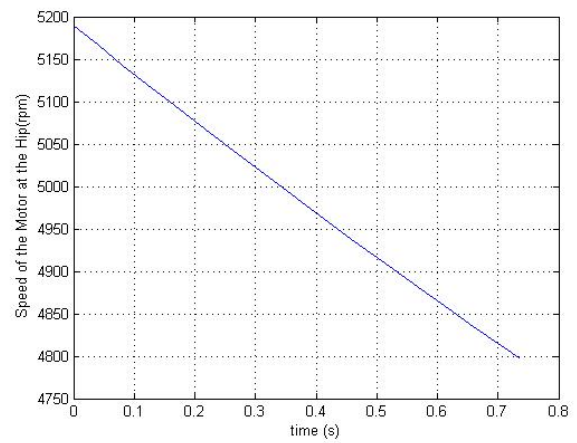


Fig. 4-5. Speed diagram of the motor at the hip

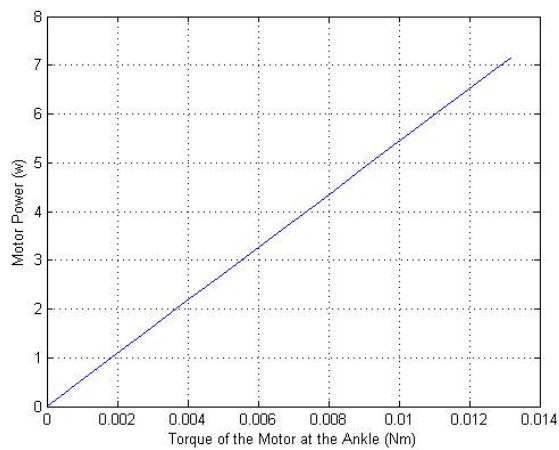


Fig. 4-6. Power-torque diagram of the motor at the foot

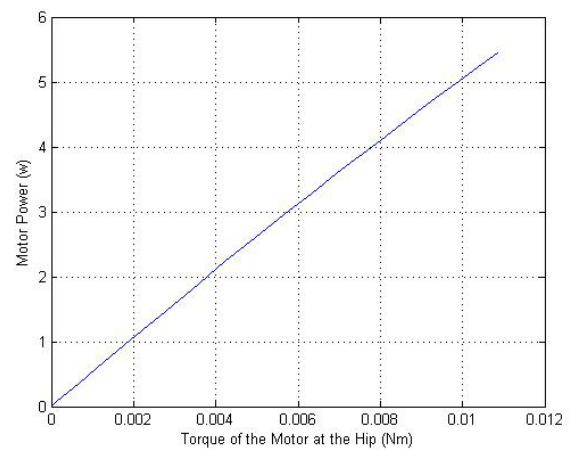


Fig. 4-7. Power-torque diagram of the motor at the hip

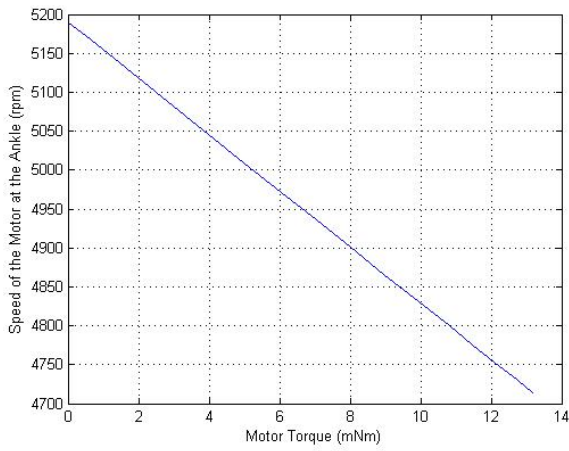


Fig. 4-8. Torque-speed diagram of the motor at the foot

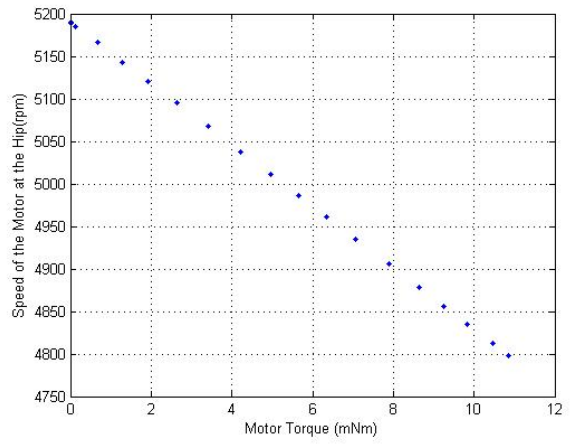


Fig. 4-9. Torque-speed diagram of the motor at the hip

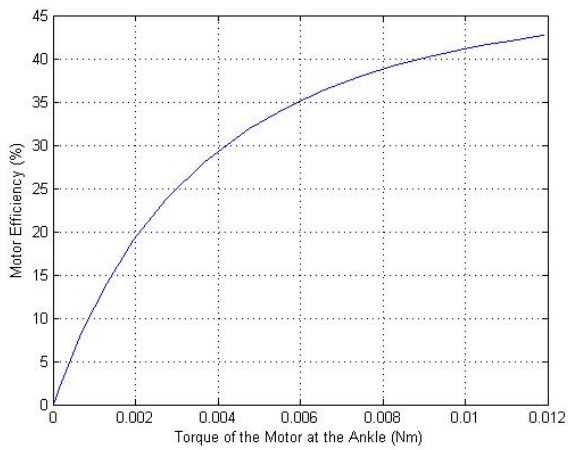


Fig. 4-10. Efficiency of the motor at the foot

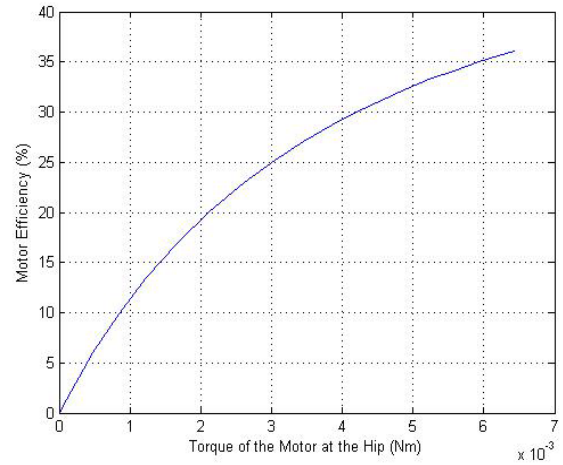


Fig. 4-11. Efficiency of the motor at the hip

Figs 4-4 and 4-5 show the required speed of the motor at the foot and at the hip respectively. As it can be seen in the above plots, the speed, power and torque of two motors at the hip and foot are in the permissible area. It should be noted that the power-torque diagram shows the performance of the motor in which the torque is smaller than the maximum continuous torque. It can be seen from Fig 4-4 that the speed is dropping. For the differential equation of the motor's speed derived in Section (4-3) and given by formula (4-14), it was assumed that the motor starts from No load speed which is the highest speed of the motor and as the motor's torque increases, the motor's speed will decrease. Figs 4-8 and 4-9 show the torque-speed diagram of the motor. As it can be seen from the plot, the motor satisfies the torque-speed constraint while it is producing the required torque for the robot. In addition, Power-torque diagram and efficiency diagram show the acceptable performance of the motor since they are following the general required curve and they are not exceeding from the maximum permissible value.

In the next steps, motor RE25 20 w (Order No. 118752) with gear GA 45, $n=310$ (Order No. 301187) is used at the hip. The motor at the foot is the same one which was used in the last step (RE25 10 w). The following plots show the performance curve of the corresponding motor.

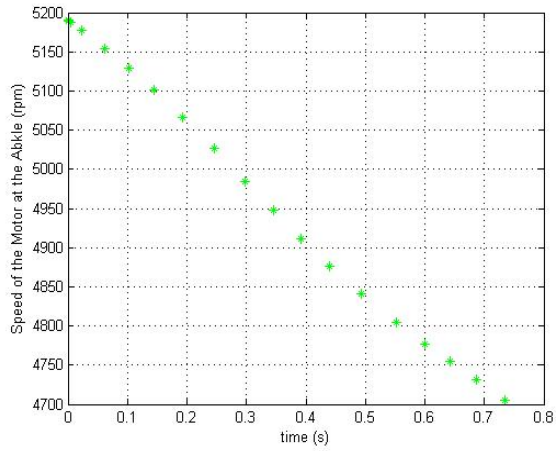


Fig. 4-12. Speed diagram of the motor at the foot

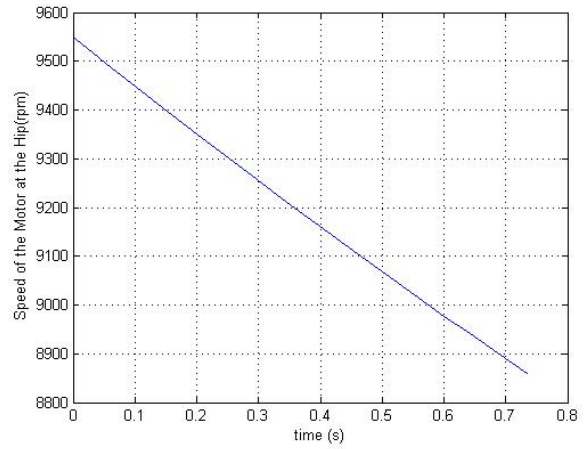


Fig. 4-13. Speed diagram of the motor at the hip

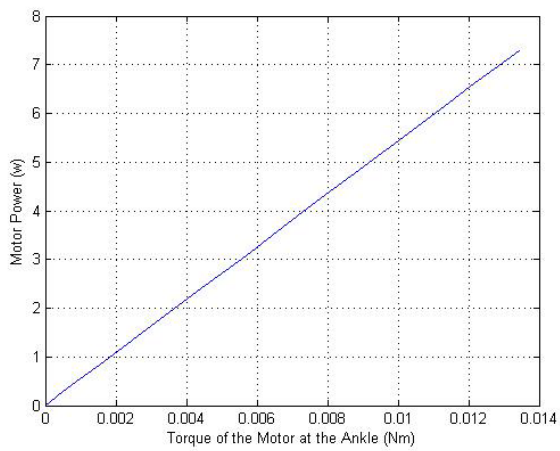


Fig. 4-14. Power-torque diagram of the motor at the foot

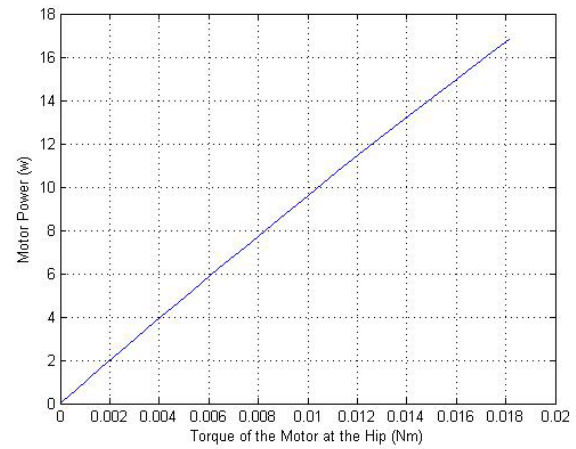


Fig. 4-15. Power-torque diagram of the motor at the hip

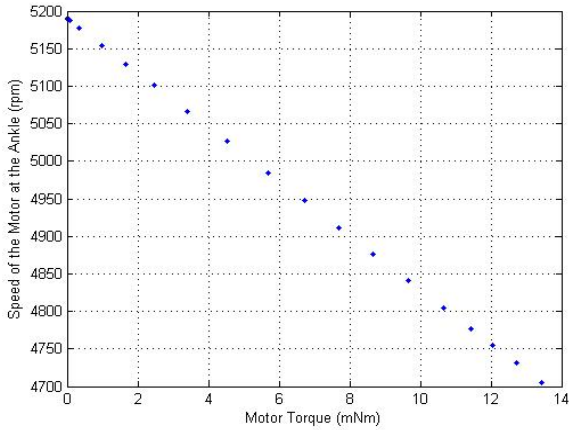


Fig. 4-16. Torque-speed diagram of the motor at the foot

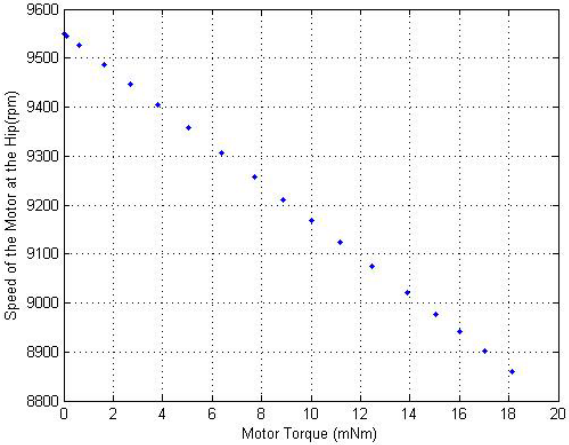


Fig. 4-17. Torque-speed diagram of the motor at the hip

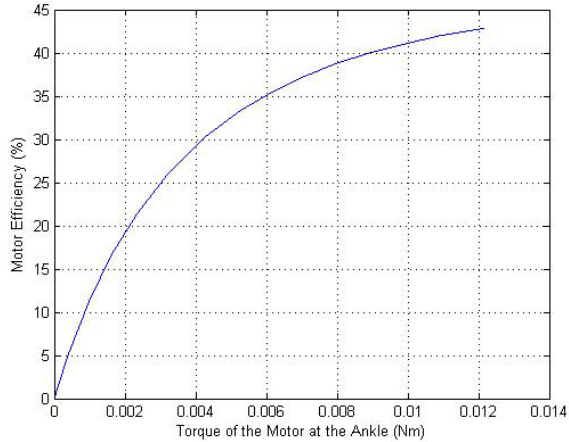


Fig. 4-18. Efficiency of the motor at the foot

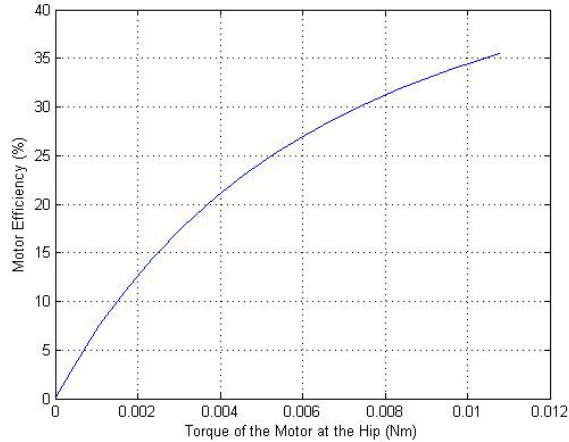


Fig. 4-19. Efficiency of the motor at the hip

Other selections of the motors are listed in the table (4-1) and their characteristic and performance plots are given in Appendix C.

4.6 Battery Selection

For every motor used in the robot, a suitable battery is needed in order to make the robot walking autonomous. The nominal voltage of the battery should be the same as the motor's voltage. The power needed for operating the robot has to be calculated. When a DC motor is subjected to an input voltage, an electrical power P_{el} is generated in the motor. P_{el} which is used for running the electric motor is divided into mechanical power P_{mech} and power loss P_j which is associated with the resistance of the winding. The motor input power P_{el} is calculated by knowing the voltage and current of the motor. The mechanical power associates the torque and speed of the motor. .

$$P_{el} = P_{mech} + P_j \quad (4-16)$$

$$P_{mech} = T_m \omega \quad (4-17)$$

$$P_j = Ri_a^2 \quad (4-18)$$

Mechanical power and power loss are calculated over a walking cycle for the motors at the hip and at the foot. The required power of the battery is the summation of mechanical power and power loss. It is noted that drawn current from battery (in the motor) can be calculated by knowing the torque and torque constant K :

$$i_a = \frac{T_m}{K} \quad (4-19)$$

It is noted that the required current by the motor should be less than the maximum current of the battery. After determining the maximum required current and power for each motor, the battery capacity can be determined and the battery can be selected. In

this thesis, a Nickel Metal Hydride (Ni-MH) battery is used since it can deliver a high starting current and it is rechargeable. (Note: the batteries are selected for 5 hours working).

Table 4-1. Required Battery for Different Motors

Selection	Motors	Max. Current (A)	Req. Power	Battery	Weight (kg)
(1)	Hip (RE25 20w)	2.603	320.4	24V 4.5Ah	1.7
	Foot (RE25 10w)	0.9293	132.73	24V 2.2Ah	0.62
(2)	Hip (RE25 10w)	0.8	103.3671	24V 2.2Ah	0.62
	Foot (RE25 10w)	0.9224	135.85	24V 2.2Ah	0.62
(3)	Hip (RE25 10w)	0.8	100.81	24V 2.2Ah	0.62
	Foot (Re-max 5w)	0.9655	44.28	9.6V 1.1Ah	0.198
(4)	Hip (RE25 10w)	0.8	102.77	24V 2.2Ah	0.62
	Foot (Re-max 5w)	1.3218	189.4	24V 4.2Ah	1.45

4.7 Characteristics of Different Selections of Actuator

In this section, the characteristics of different motors with their associated components (gear+battery) are summarized in table (4-2). This table is useful for final selection of the actuator.

Table 4-2. Characteristics of Different Motor Selection

	Motors	Weight (Battery+motor+Gear) on the Hip and Foot	Max. motor Efficiency	Cost (Battery+Motor+ Gear)
(1)	RE25 20w	$M_h = 1.4+0.130+0.255= 1.785$	Hip: 37%	$131.2+279+184=594.2$
	RE 25 10w	$m_h = 0.62+0.130+0.224=0.9740$	Foot: 43%	$40.1+270+160= 470.1$
(2)	RE25 10w	$M_h = 0.62+0.130+0.255=1$	Hip: 35%	$40.1+270+184=494.1$
	RE 25 10w	$m_h = 0.62+0.130+0.224=0.9740$	Foot: 44%	$40.1+270+160=470.1$
(3)	RE25 10w	$M_h = 0.62+0.130+0.255=1$	Hip: 35%	$40.1+270+184=494.1$
	RE-max 21 5w	$m_h = 0.198+0.042+0.224=0.464$	Foot: 34%	$112+88+160=360$
(4)	RE25 10w	$M_h = 0.62+0.130+0.255=1$	Hip: 35%,	$40.1+270+184=494.1$
	RE-max 29 15w	$m_h = 1.45+0.159+0.224= 1.833$	Foot: 42%	$110+150+160=420$

4.8 Effect of the Motor on the Bifurcation Maps

As it was shown in the previous section, different choices of motor are possible. However, they have different effect on the structure and consequently on the gait of the robot. The mass of the actuator (motor, gear train and battery) at the hip and at the foot affect the mass ratio $\mu = \frac{M_h}{m}$ and the length ratio $\beta = \frac{b}{a}$. Since M_h is summation of the

mass of the motor, gear train and battery at the hip and m is the summation of the mass of the leg and mass of the actuator (motor, gear train and battery) at the foot. In addition, the mass of the actuators at the hip and at the foot affect the center of the mass of the leg and consequently, change the length ratio of the leg. The mass ratio and length ratio of the robot's leg can be calculated as follows:

$$\mu = \frac{M_h}{m + m_h} \quad (4-20)$$

$$\beta = \frac{b}{a} \quad (4-21)$$

$$b = \frac{m * l/2 + m_h * l}{m + m_h}$$

In this section, the effect of different motor selection on the performance is discussed by bifurcation maps and Poicare maps. In Section 2, the bifurcation map of a compass robot was shown for $\mu = 2$ and $\beta = 1$. As it has been shown in Table 4-2, the values of μ and β is different for different motors. So, we need to derive a new bifurcation map for every selection. For analyzing Poincare map, as discussed in section 3, first we need to find the fixed-point for different motors. Then the stability is analyzed about the fixed-point by Poincare Map analysis. Table 4-3 shows the different values of mass ratio, length ratio for different selection and corresponding eigenvalues for stability analysis.

Table 4-3. Values of μ and β and Poincare Map Results for Stability Analysis

	Motors	Mass Ratio $\mu = \frac{M_h}{m+m_h}$	Length Ratio $\beta = \frac{b}{a}$	Eigenvalues (Poincare Map)
(1)	Hip: RE25 20w	0.6002	1.9740	Not Applicable (two-periodic gait)
	Foot: RE 25 10w			
(2)	Hip: RE25 10w	0.3362	1.9740	0.0798±0.656
	Foot: RE 25 10w			0.2403
(3)	Hip: RE25 10w	0.4058	1.4640	0.0568±0.661
	Foot: RE-max 21 5w			0.2285
(4)	Hip: RE25 10w	0.2609	2.833	Not applicable (Unstable Gait)
	Foot: RE-max 29 15w			

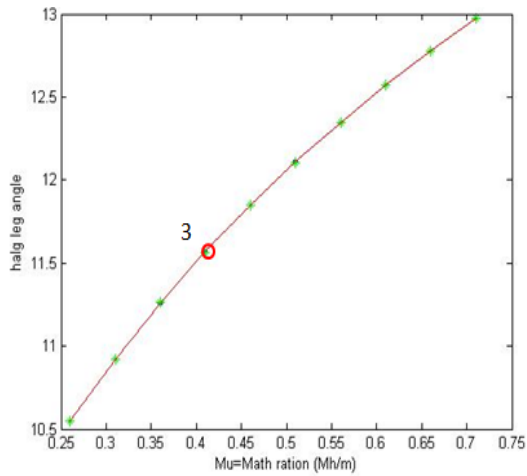
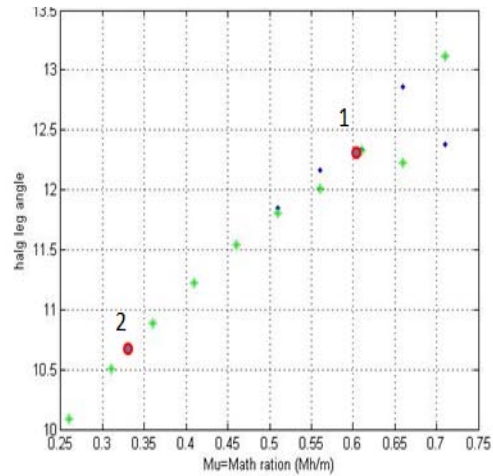
Fig. 4-20. Bifurcation map for $\beta=1.464$ Fig. 4-21. Bifurcation map for $\beta=1.9704$

Fig 4-20 shows the bifurcation map for selection 3 where $\beta=1.464$. It can be seen that the gait is one-periodic for different values of μ for corresponding value of length ratio, β . For $\beta=1.9704$ which corresponds to selection 1 and 2, the gait become two-periodic for $\mu \geq 0.52$. As it is shown in Fig 4-21, selection 2 is in the acceptable domain but selection 1 is within the 2-periodic domain. For selection 1, $\mu = 0.6$ and two different values of half leg angle α are found for that including $\alpha = 12^\circ$ and $\alpha = 12.3^\circ$. It means that the robot takes one small step in which $\alpha = 12^\circ$ and one larger step in which $\alpha = 12.3^\circ$. Even though, the values of α are very close to each other, but this selection is not acceptable since it is on the verge.

4.9 Motor Selection

As it can be seen from Figs 4-18 and 4-19, Selection 1 is not acceptable, since it result in 2-peridic gait. The only acceptable motors are Selection 2 (RE25 10w+ RE25

10w) and Selection 3 (RE25 10w+ RE-max 21 5w). In addition, the eigenvalues of the gait for these selections are less than 1 so the gait is stable.

In Next step, two selections “2” and “3” are compared in terms of motor efficiency and cost. As it can be seen in table (4-3), selection (3) has a lower price. However, the efficiency of the motor at the foot is also 10% lower than the one in selection (2). So, choosing between two options (2) and (3) is compromising between efficiency and price. So, selecting between two depends on designer’s preference. In this paper, since the efficiencies of two motors are not different too much instead, the price of selection (3) is 100% less than selection (2), selection (3) is more desirable.

5 CHALLENGES AND DIFFICULTIES OF A COMPASS ROBOT OPTIMIZATION

There has been a huge interest in gait optimization of biped robots. Most of the previous researches focused on improvement of robot's gait using computational intelligent techniques such as the researches done by Oliveira et al. (2011), Iida and Tedrake (2007), Ames (2012) and Asta and Sariel-Talay (2011). In these researches, the optimal motion was generated by minimizing the performance criteria which are usually the actuating torque or consumed energy. In this thesis, the structure of a compass robot is optimized with considering non-linear constraints presented by stability and convergence. The criteria of optimization is minimizing energy expenditure and maximizing gait velocity. The limitations introduced by non-linear model of the system and stability constraint which is also non-linear make optimization problem more complicated and challenging. The difficulties and challenges of the robot's optimization and the solution for solving the problem are explored in this section.

5.1 Optimization Problem Statement

In this thesis, the structural optimization of a compass robot is investigated. The robot's leg is cylindrical and the length and diameter of the leg are considered as optimization variables. Two objectives including minimizing the mechanical energy and maximizing the velocity of the robot are the goals of the optimization. For maximizing

the speed of the robot, it is enough to maximize its step length. In this problem, an improvement in one objective is gained at the cost of deteriorations of another objective, since maximizing the length step requires more energy. So, trade-off between two objectives is inevitable and optimization algorithm seeks for the solution which optimizes two criteria simultaneously. The optimization problem can be stated as follows:

$$\begin{aligned}
 x^* &= \underset{x}{\operatorname{argmax}} f(x) \\
 x_{lb} &\leq x \leq x_{ub} \\
 g_i(x) &\leq 0, \quad i = 1, \dots, m \\
 h_j(x) &= 0, \quad j = 1, \dots, p
 \end{aligned} \tag{5-1}$$

where the objective function $u(x)$ is a function of the design vector x . $x = [d, l]$ which consists of diameter and length of the leg respectively, g and h are inequality and equality constraints. x_{lb} and x_{ub} are the lower and upper bounds of the design variables.

5.1.1 Weighting the objective functions

The most widely-used method for multi-objective optimization is the weighted sum method. This method converts multiple objectives into a single one by multiplying each objective function by a weighting factor and summing up all the weighted objective function. The weighting factors specify the importance of each objective function. Initial work on the weighted sum method can be found in Zadeh (1963). A general discussion

about multi-objective optimization and practical implementation is provided by Grodzevich (2006). The systems objective function can be written as:

$$f(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x)$$

$$\sum_{i=1}^2 \alpha_i = 1 \quad (5-2)$$

$$0 \leq \alpha_i \leq 1$$

where, $u_1(x)$ and $u_2(x)$ are objective functions of the system. α_1 and α_2 are the coefficients which show the importance of each objective function. Because of the equivalent importance of two objective functions in this thesis, it can be concluded that:

$$\alpha_1 = \alpha_2 = 0.5 \quad (5-3)$$

where $u_1(x)$ is mechanical energy, $u_2(x)$ is step length. The mechanical energy of the system is calculated based on the energy applied by the controlled symmetries in one step. As it was shown in Section 2, control law can be established by shaping the potential energy of associated robot to allow stable walking on the flat ground. So, the mechanical energy required for each step can be derived as follows:

$$f_1(x) = M_E(x) = (V(\theta) - V(\Phi(\gamma, \theta))) \quad (5-4)$$

The step length of the robot can be calculates as follows:

$$f_2(x) = L_k(x) = 2l \sin(\theta_{ns} - \theta_s) \quad (5-5)$$

- Normalization of Objective Functions

Weights of each objective function are assigned based on their preferences and importance. As different objective function has different magnitude range, the normalization of objectives in the same range is required in order to get a consistent optimal solution. For normalizing, we use from method represented in Grodzewich (2006) and Zadeh (1963) in which normalization was done by the difference of optimal function values in the Nadir and Utopia points. Nadir and Utopia points give the length of the intervals. The Utopia points provide the lower bound of optimal and the Nadir points are the upper bound of the set. For deriving Utopia and Nadir points for each objective function, we need to apply the optimization for each objective separately to find its minimum and its maximum. Utopia and Nadir points can be calculated as follows:

$$z_i^u = \underset{x}{\text{ARGMIN}} f_i(x) \quad , i = 1, 2 \quad (5-6)$$

$$z_i^N = \underset{x}{\text{ARGMIN}} (-f_i(x)) \quad , i = 1, 2 \quad (5-7)$$

The normalized objective function $U_{Ni}(x)$ can be found as follows which are bounded by:

$$0 \leq f_{Ni}(x) = \frac{f_i(x) - z_i^u}{z_i^N - z_i^u} \leq 1 \quad (5-8)$$

That gives the same intervals to each objective function. By knowing the Utopia and Nadir points and after normalizing, the final objective function of the system can be written as follows:

$$f(x) = \alpha_1 M_E(x) + \alpha_2 (1 - L_k(x)) \quad (5-9)$$

It is noted that the objective function $(1 - L_k(x))$ is used instead of $L_k(x)$ since we need to maximize step length. Minimizing $(1 - L_k(x))$ is equivalent to maximizing $L_k(x)$.

5.1.2 Instability constraint

Gait stability of the robot is the most important criteria in the design and optimization of the robot. Indeed, optimization algorithm has to search in the stable domain and finds the best structure which result in stable gait of the robot. Stability of compass robot was discussed in section (3-1) and it was shown that the gait stability can be determined by the eigenvalues of Poincare map linearized about the fixed point which is shown by Jacobian matrix $\Phi_\theta(\theta^*, T)$. One eigenvalue of Jacobian matrix is always “1”. The rest eigenvalues have to be smaller than “1” to guarantee the stability of the system otherwise, the system is unstable. This constraint can be shown as follows:

$$DP(\theta^*) = \Phi_\theta(\theta^*, T) = \left(\frac{\partial \phi(\theta^*, T)}{\partial \theta^*} \right) \quad (5-10)$$

$$C_{\text{noneq}}: g_i(x) = \text{eig}(\Phi_\theta(\theta^*, T)) - 1 < 0$$

i = number of eigenvalues of $\Phi_\theta(\theta^*, T)$

5.1.3 Optimization problem statement

After deriving the normalized objective functions, stability constraint and final objective function, the optimization problem can be stated as follows:

$$\begin{aligned}
x^* &= \underset{x}{\text{ARGMIN}} f(x) \\
x &= [d, l] \\
x_{lb} &\leq x \leq x_{up} \\
f(x) &= \alpha_1 M_E(x) + \alpha_2 (1 - L_k(x)) \\
C1: \quad g_i(x) &= \text{eig}(DP(\theta^*)) - 1 < 0, \quad i = 1, \dots, n - 1
\end{aligned} \tag{5-11}$$

The optimization variables $x = [d, l]$ are the diameter and length of the robot respectively and n is the number of system state variables.

5.2 Optimization Difficulties and Challenges

As mentioned in previous section, stability of the robot is considered as a non-linear inequality constraint for the optimization. So, when the optimization algorithm computes objective function for different variables' values in every iteration, the stability of the robot should be computed and checked as well. For checking the gait stability for different values of variables, it is required to derive the convergent limit cycle of the gait for relevant values (Structure of the leg) and its corresponding fixed point, and then compute the eigenvalues of Poincare map about that fixed-point. More important, the optimization needs to compute the objective function in the convergent limit cycle otherwise the values of objective function might be smaller or larger than the real one. The main challenge of optimization is that finding the convergent limit cycle and its corresponding fixed-point in each iteration of optimization is very cumbersome which makes the optimization hard to solve.

In this thesis, the fixed point corresponds to initial condition from which the trajectory emanates and it should be the same as the condition when the robot hits the ground. Generally the fixed point is found manually by trial and error, but in optimization, when algorithm proceeds to the next iteration with different variables' values, it needs a well-defined method to compute it very fast and accurately. In the following, different methods for solving the optimization problem and their drawbacks are discussed.

5.3 Different Solution for Optimization and Their Drawbacks

5.3.1 All-at-once optimization

The first solution for the problem is to consider the initial condition (i.e. fixed point) as optimization variables along with robot's structure. In this method, fixed point and eigenvalues of Poincare map are applied as constraints to the optimization. The fixed point constraint implies that the trajectory which begins from the fixed point has to continue through the same point in the next period after the robot hits the ground. This constraint can be shown by the difference of Poincare map $P(x_i^*)$ which is the state of the system after impact intersection of the trajectory with the hyperplane) and the initial condition as follows:

$$h(x) = \|x_k^* - P(x_k^*)\|_2 = 0 \quad (5-12)$$

$$P(x_k^*) = \phi(R(x_k), T_1(R(x_k)))$$

where $x_k^* = R(x_k)$ is the initializing point for trajectory ϕ and $T_1: \mathcal{X} \rightarrow \mathbb{R}$ is the period time and $P(x_k^*)$ is the reinitializing condition in the next period after the robot hits the

ground as it was shown in section (3). It should be mentioned that satisfying fixed-point constraint assures the convergence of limit cycle in the system.

A scheme of this approach of optimization is depicted in fig (5-1). The variables of the system consists of $[\theta, S]$. θ is the fixed point which corresponds to the initial condition and consists of 3 variables $\theta = [\theta_{0ns}, \dot{\theta}_{0ns}, \dot{\theta}_{0s}]^T$. S is the vector of robot's structure which consists of 2 variables including length and diameter of the robot's leg, $S = [d, l]$. It is noted that in the fixed-point constraint, we do not need to consider θ_{0s} since it always has a relationship with θ_{0ns} . For a compass robot with controlled symmetries $\theta_{0s} = \theta_{0ns}$. The statement of optimization for this method can be written as follows:

$$\begin{aligned}
 x^* &= \underset{x}{\text{ARGMIN}}_1(x) \\
 x &= [d, l, \theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s] \\
 x_{lb} &\leq x \leq x_{up} \\
 f(x) &= \alpha_1 M_E(x) + \alpha_2 (1 - L_k(x)) \\
 \text{C1:} \quad &h(x) = \|x_k^* - P(x_k^*)\|_2 = 0 \\
 \text{C2:} \quad &g_i(x) = \text{eig}(DP(\theta^*)) - 1 < 0
 \end{aligned} \tag{5-13}$$

This method is unable to find the optimum objective function because unless the θ vector is a fixed point, checking the stability constraint is not valid i.e., when the convergence of limit cycle is not assured, the stability analysis is not valid and even the value of objective functions are not accurate.

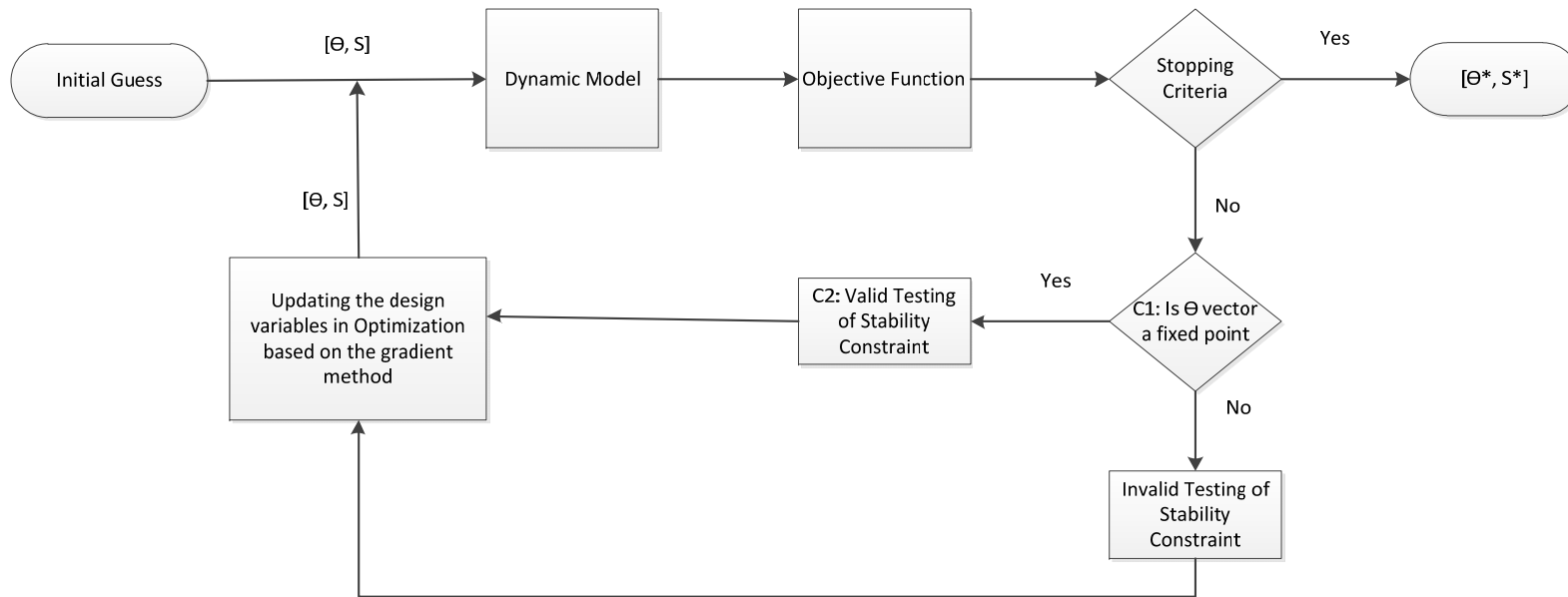


Fig. 5-1. Scheme of the all-at-once optimization with fixed-point and stability constraint

As it can be seen from Fig. 5-1, the optimization variables $[d, l, \theta_{ns}]$ are updated in every optimization iteration. During Optimization iteration with various values for θ and S , since the updated θ is not fixed points for corresponding updated S values, optimization fails to find the convergent limit cycle and analyze the Stability Constraint. However optimization proceeds to the next iteration with new updated variables in order to find the desired variables. The unlikely event where in an iteration the updated values of θ are the fixed point for corresponding updated S , it will again fail in the subsequent iteration. So, the final response of optimization will be invalid. Fig 5-2 shows 3 optimization variables $[d, l, \theta_{ns}]$. “*” points are the data which are selected by optimization in each iteration and “o” points show the condition in which θ_{ns} is one element of the fixed point for the relevant $[d, l]$. It can be seen that there is a huge difference between the fixed-point and the selected variables by optimization for relevant structure. It means that optimization cannot succeed in finding the fixed-point and the convergent limit cycle for the relevant structure in each iteration.

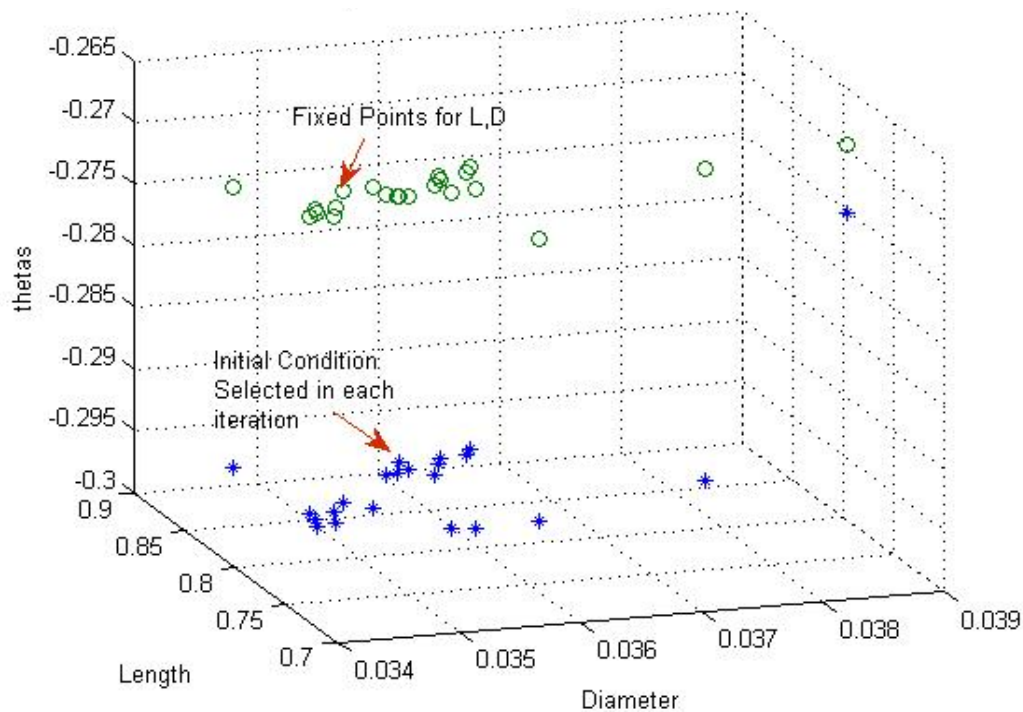


Fig. 5-2. Comparison of the fixed-points selected by optimization and the real fixed-points for the relevant structure

This method was applied with different optimization algorithms including Gradient-based interior-point method, Genetic algorithm Malab. None of the optimization methods can find the consistent result.

5.3.2 Nested optimization

The second technique for optimization problem is to break down the problem into two levels. Lower level optimization finds the fixed point for the corresponding structure variables. In the upper level, the robot's dimension is considered as optimization variables. In every iteration of the upper level, updated values of dimensional variables

are passed to the lower level of optimization. Lower level optimization finds the fixed point and passes them to the stability constraint to verify its stability. The scheme of this method is shown in Fig 5-2. The statement of optimization problem can be written as follows:

Lower level:

$$\theta^* = \underset{\theta}{\text{ARGMIN}} f_1(\theta)$$

$$\theta = [\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s] \quad (5-14)$$

$$\theta_{lb} \leq \theta \leq \theta_{up}$$

$$f_1(\theta) = \|\theta_k^* - P(\theta_k^*)\|_2$$

Upper Level:

$$x^* = \underset{x}{\text{ARGMIN}} f_2(x)$$

$$x = [l, d]$$

$$x_{lb} \leq x \leq x_{up} \quad (5-15)$$

$$f_2(x) = \alpha_1 * M_E(x) + \alpha_2 * (1 - L_k(x))$$

$$C1: g_i(x) = \text{eig}(DP(\theta^*)) - 1 < 0$$

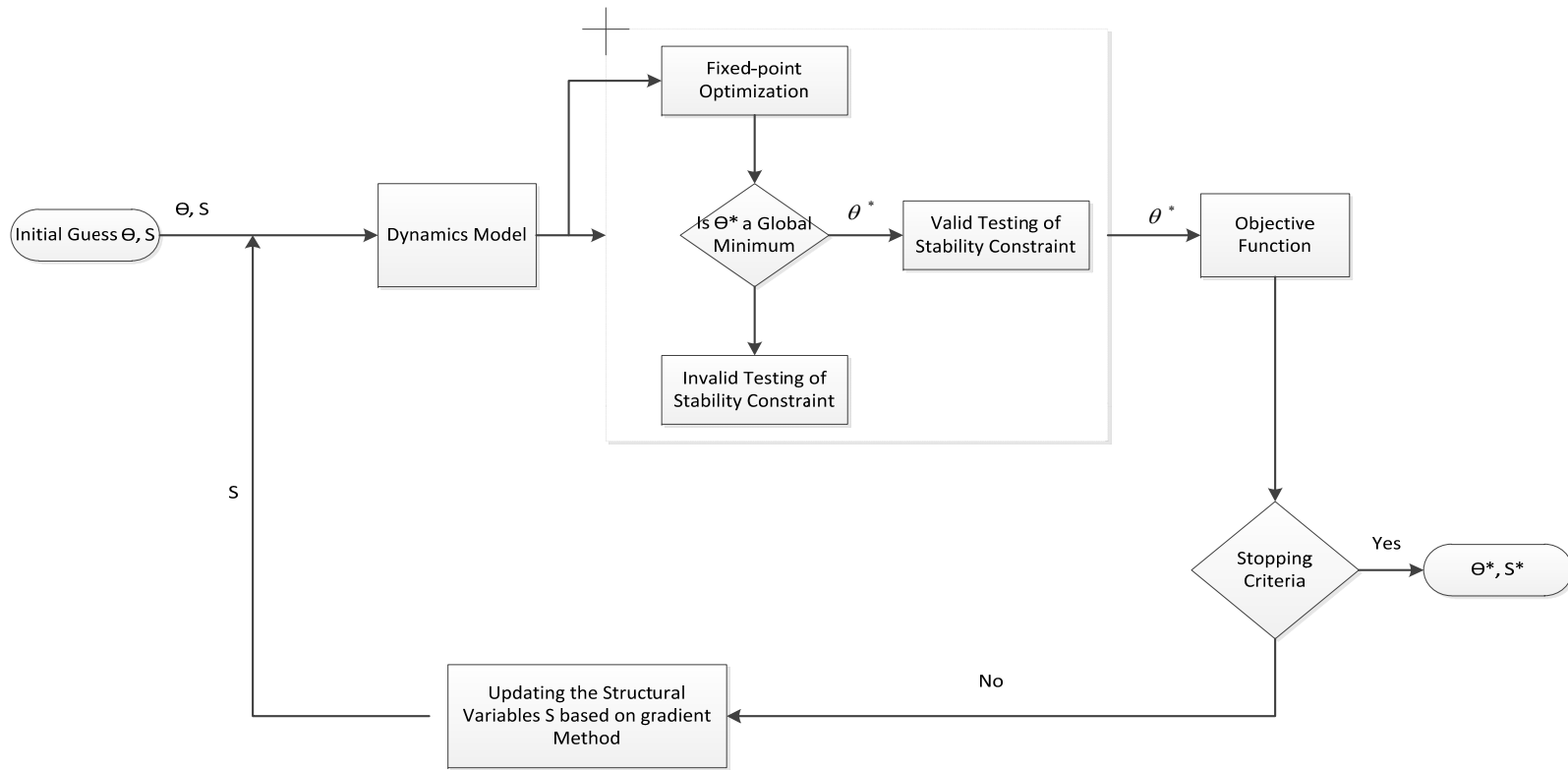
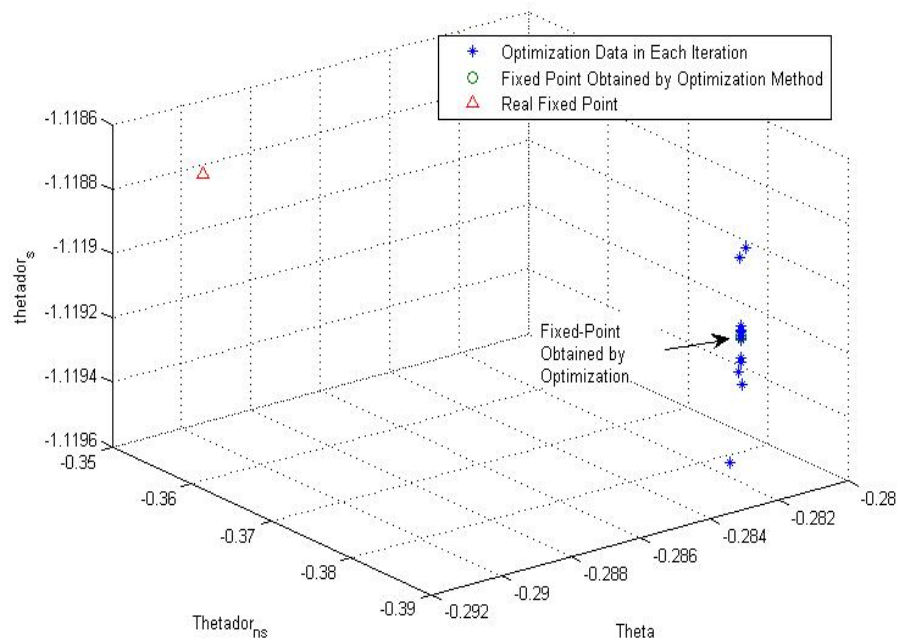


Fig. 5-3. Scheme of the nested optimization with fixed-point and stability constraints

This method of optimization succeeded to find the optimum solution. However, the accuracy is low, since the lower level optimization sometimes finds the local minimum. The accuracy of the lower level which is used to obtain the fixed-point can be improved by providing the initial guess θ from the final optimum value obtained in previous iteration of lower level optimization. This helps to find the global minimum rather than the local minimum. It also help that the local minimum get closer to global minimum (fixed-point). However, there are still some cases in which there is an error between the real fixed- point and the one found by lower level optimization because of the local minimum problem. Fig 5-4 shows the case in which the error between the fixed-point founded by optimization and the real fixed-point is considerable. The cases like this case lower the accuracy of the nested optimization. Figs 5-5 shows the case in which lower level optimization is able to find the exact fixed-point. The local minimum is very close to the fixed-point and it means that nested optimization is able to find the optimum point.



omparison of the optimization result for finding the fixed-point with the real fixed-Point

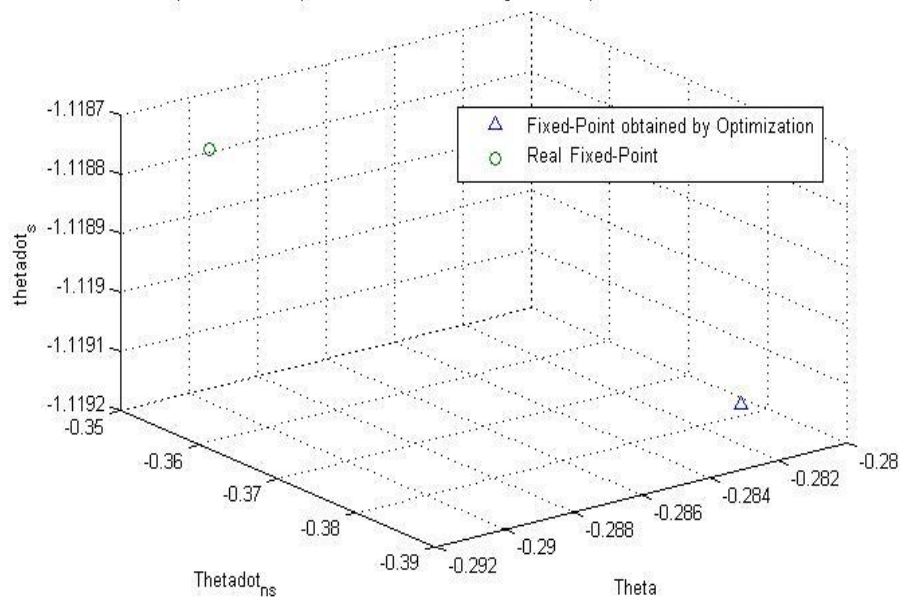


Fig. 5-4. Comparison of the real fixed-point and the one founded by optimization(high difference)

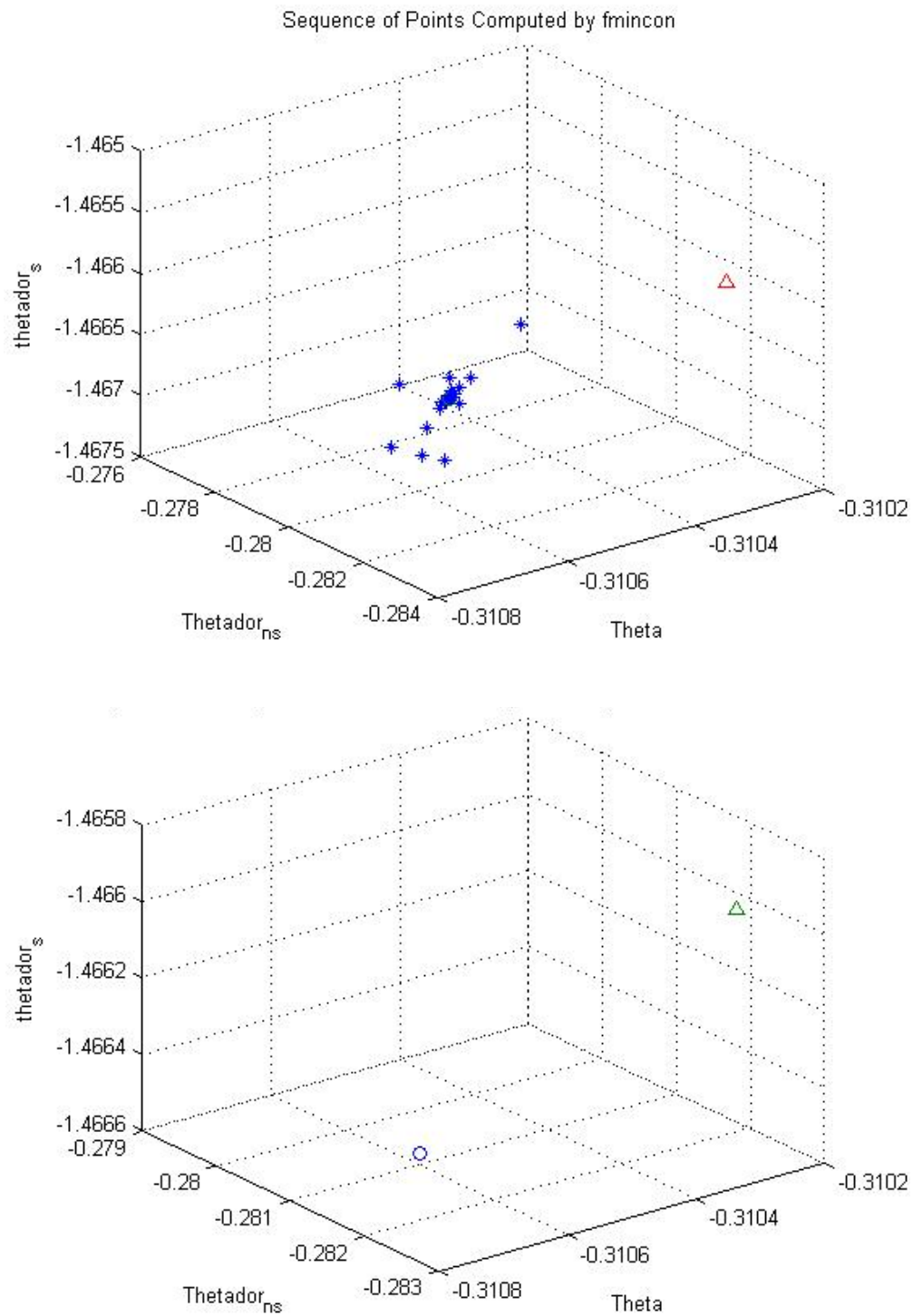


Fig. 5-5. Comparison of the real fixed-point and the one founded by optimization (small difference)

This method has another shortcoming. The nested optimization takes a long time since in every iteration of the upper level optimization, lower level optimization must be performed to find the fixed-point. The time of optimization is really high and specially, if the number of variables increases, the time will increase exponentially. The results of optimization for one period of walking are as follows:

Lower Level Initial Guess:

$$[\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]^T = [-0.2710, -0.3774, -1.067]^T$$

Upper Level Initial guess: $[d, l]^T = [0.045, 0.58]^T$

Optimum Dimensional Parameters: $d = 0.036, l = 0.5951$

Normalized Objective Function= 0.4695

Mechanical Energy: 55.135 J

Step Length: 0.3592 m

Time of Optimization: 1120s

5.4 Using Convergence Principle for Finding the Fixed Point

The first optimization method explained above was failed because of not being able to find the fixed point. The nested optimization was able to find the optimum, however the accuracy of the answer is low and the computational time of the optimization is very high. In this part, a new technique is introduced which can find the fixed point faster. Ocken (1995) and Akian et al. (2011) have shown that a given orbit of a discrete dynamical system can converge to a fixed point or periodic orbit. Ocken (1995) has

developed a computational method which obtains the attracting periodic orbit for a one-dimensional system.

In this thesis, the same idea is applied for finding the convergent periodic orbit of the compass robot for any dimension. If a robot starts from an initial condition close to its fixed point and within its domain of attraction, it will converge to its attractive periodic orbit after a finite number of steps and the fixed point for the corresponding limit cycle can be found. Therefore the fixed point constraint will be changed to the periodic orbit convergence constraint. It has to be noted that convergence does not assure stability, because in some cases, the gait's limit cycle converges but with a very small disturbances it become unstable very easily. The Poincare map analyzes the stability of the system in presence of disturbances.

If this constraint is satisfied during optimization, the fixed point can be computed and sent to the stability constraint to check the gait's stability. However, I should be noted that convergence always happens. The reason is that the optimization procedure is gradient-based and optimization select the new combination of variables based on the variables in the previous iteration so the new selection guarantees the convergence of limit cycle. A schematic illustration for this procedure is depicted in Fig 5-6.

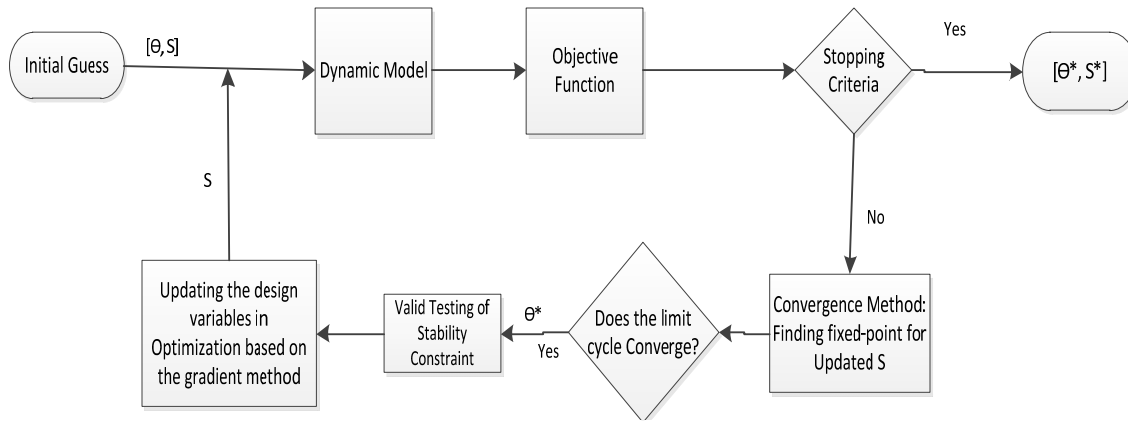


Fig. 5-6. Scheme of the convergence method for finding the fixed-point in optimization process

In Fig 5-7, the convergence to a stable limit cycle when the initial condition is in the domain of attraction of the fixed-point, is shown. Fig 5-8 shows the positional error and total error between the initial condition of a limit cycle and initial condition of the subsequent limit cycle. If the error converged to zero, it means that the limit cycle converges and the initial condition converges to fixed point for the corresponding dimensional parameters. The dimension of the robot, its initial condition and its fixed-point which is obtained after convergence are as follows:

$$d = 0.0425\text{m},$$

$$l = 0.57\text{m},$$

$$\text{Initial Condition: } x_0 = [-0.3135, 0.3135, -0.3320, -1.5368]^T;$$

$$\text{Fixed-Point: } x_{fp} = [-0.2953, 0.2953, -0.2849, -1.5112]^T;$$

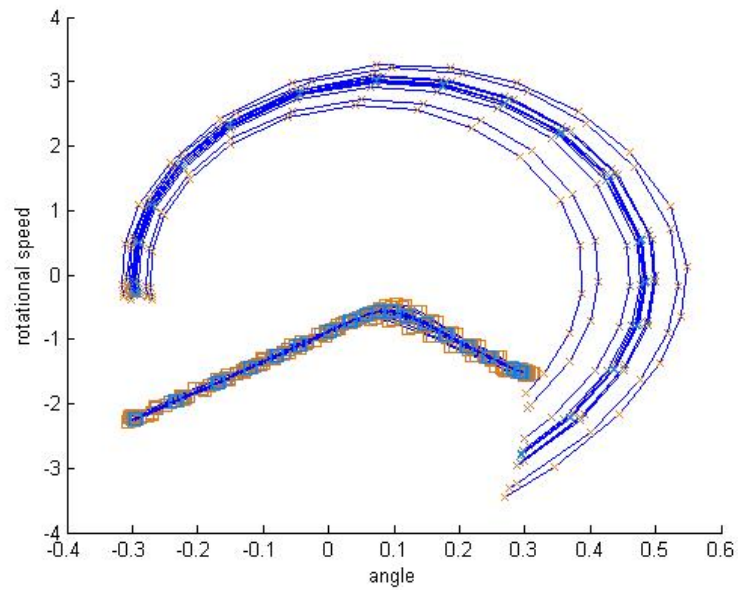


Fig. 5-7. Convergence to a stable limit cycle after finite steps

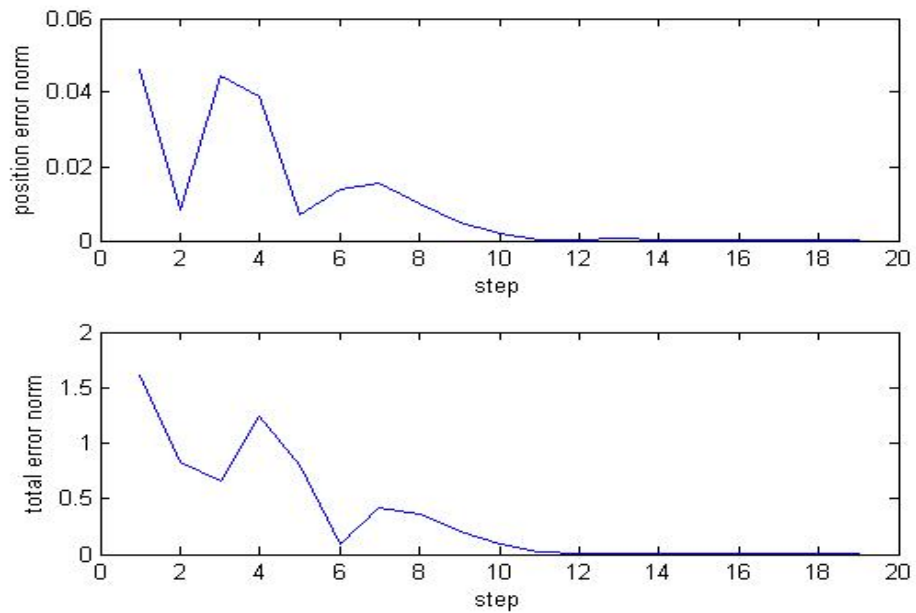


Fig. 5-8. Errors between the initial conditions in the first steps and fixed-point after 20 steps

5.4.1 Statement of optimization problem and the results of optimization

The optimization problem statement with convergence and stability constraints can be written as follows:

$$\begin{aligned}
 x^* &= \underset{x}{\text{ARGMIN}} f(x) \\
 x &= [d, l, \theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s] \\
 x_{lb} &\leq x \leq x_{up} \\
 f(x) &= \alpha_1 M_E(x) + \alpha_2 (1 - L_k(x)) \\
 \text{C1: } h(x) &= \|x_k^* - P(x_k^*)\|_2 = 0, \quad k = 20 \\
 \text{C2: } g_i(x) &= \text{eig}(DP(\theta^*)) < 1
 \end{aligned} \tag{5-16}$$

where "k" is the required steps that robot has to take (number of periodic cycles) and θ^* is the fixed point where $\theta^* = P(\theta^*)$ in the convergent limit cycle. In most cases, the limit cycle converges after 10-12 steps, but when the initial condition vector for the corresponding structure is far from the fixed-point, it needs more steps for convergence. Therefore, 20 steps are considered in the optimization process in order to guarantee the convergence of the limit cycle and finding its fixed-point.

The results of optimization of equation (5-12) for the following initial condition and optimization initial guess are as follows:

$$\text{Initial guess: } [d, l, \theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]^T = [0.034, 0.73, -0.3033, -0.2545, -1.33253]^T$$

$$\text{Optimum Dimensional Parameters: } d = 0.0337, l = 0.7329$$

$$\text{Fixed-Point for Optimum Structure: } [-0.3038, 0.3038, -0.2541, -1.3297]^T$$

$$\text{Normalized Objective Function} = 0.2631$$

Mechanical Energy: 70.42 J

Step Length: 0.43481 m

Total Time of Optimization: 746.2s

Time for checking the convergence constraint: 401.34 s

Time for checking the stability constraint: 178. 23 s

Note: Mechanical energy and step length are evaluated in the final step where we make sure the limit cycle has converged, since in the first step they might be larger or smaller than their real value in a stable limit cycle.

5.4.2 The limitation of convergence method in optimization

In the previous section, the dimension of the compass robot was optimized using convergence method. This method has some drawback and limitations. First, it takes a long time, since optimization needs to find a fixed-point after 20 steps in each iteration. It has to be mentioned that in some of the points, limit cycle convergence can be achieved in less than 20 steps. However, in the first iteration of the optimization where the step size of optimization is relatively large, the selected fixed-point (by optimization) might be farther from the real one for the corresponding structure. So, it needs to take more steps in order to converge to limit cycle. In addition, checking the stability by Poincare map for each fixed-point is a time-consuming process. As it can be seen from the results in previous section, the optimization time is about 746.2s. Even though, the time is improved and is less than the time of the nested optimization, it is still high. In addition, we consider just two optimization variables in this problem and the compass

robot has a very simple model. For the more complex systems with higher nonlinearity and more optimization parameters, the optimization time will be very large due to convergence and stability constraints.

The second shortcoming of this method which affects the optimization results tremendously is the domain of attraction for each point. Optimization algorithm starts from an initial guess $[S_0, \theta_0]$ consists of dimension and initial condition variables. When optimization variables are updated in each iteration, it tries to find the new set of variables $[S, \theta]$ which satisfies the convergence constraint after some steps. In the iterations in which optimization cannot find the convergent limit cycle for the updated set of variables, it does not calculate objective function in that case and proceeds to find the new set of variables. This constraint limitation confines the optimization to a limited set of variables. Further, in this case, using global optimization is not applicable and does not help to find a better answer.

For example, the domain of attraction for variable vectors: $[d, l, \theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]^T = [0.04, 0.55, -0.27357, -0.50847, -1.4707]^T$ is detected in Fig 5-9.

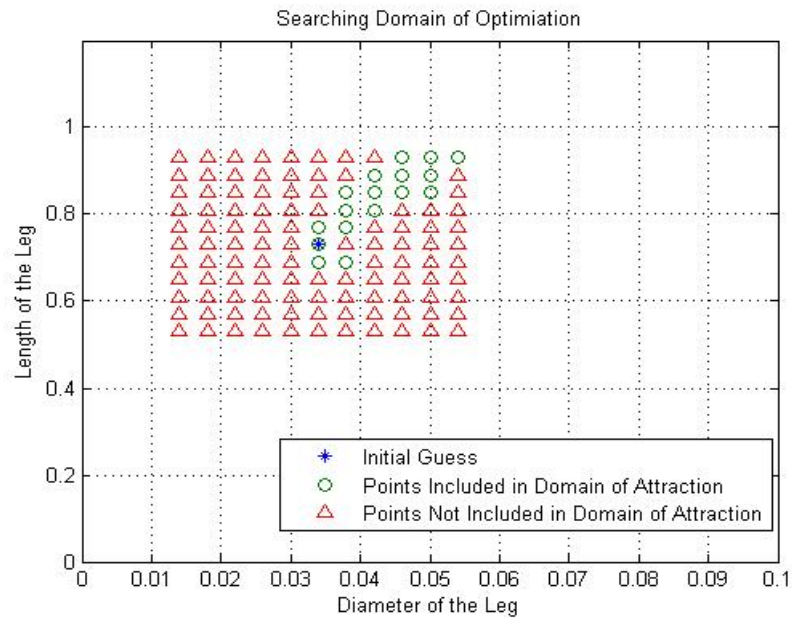


Fig. 5-9. Searching domain for optimization for each selected structure in each iteration

Hypothetically, if in an updated iteration, for the updated fixed-point $[\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]^T = [0.27357, -0.50847, -1.4707]^T$ which is shown by * in Fig 5-9, the optimization just can search in the dimensions which result in limit cycle convergence that is shown by “o” i.e., the optimization does not search in the area which leads the divergence of limit cycle. As it can be seen from Fig 5-9, the search area for optimization in every iteration is very limited.

5.5 Support Vector Domain Description (SVDD) Method for Improving Optimization

5.5.1 SVDD methodology and application

One method to improve the optimization process is to use a predictive model for the stable input data where the optimization avoid searching within the unstable domain. A predictive model allows a user to predict the unknown values of one variable as a function of the others. The main challenge with the predictive model is to define a mathematical description of the valid input data for the model.

Support Vector Domain Description (SVDD) is an approach which can be used to formalize the domain description for a predictive model. SVDD method which was first developed by Tax and Duin (1999) establishes a domain boundary using a hypersphere which envelopes the valid data. This approach is capable to form a domain classification boundary without examples of invalid data. Malak Jr and Paredis (2010) have discussed the procedure for applying the SVDD approach to predictive modeling problem and illustrated examples for applying it in the engineering systems designs.

For the basic SVDD, the goal is to find a hypersphere with minimum radius which contains a set of valid points $\{x_i, i = 1, \dots, N\}$. In mathematical description the hypersphere radius R have to be minimized with respect to the following constraint:

$$\|x_i - a\|^2 \leq R^2, \quad \forall i \quad (5-17)$$

where "a" is the location of the hypersphere center and $\|\cdot\|$ denotes the Euclidean norm and $\|x_i - a\|^2$ represents the distance of a given point to the hypersphere center. This constraint is due to the fact that all the valid data have to fall within the domain

boundary. The problem with this formula is that it does not consider the outliers. Malak Jr and Paredis (2010) have reformulated the problem using slack variables and a constant scalar parameter C in order to exclude the most extreme outliers. Parameter values in the range $1/N \leq C \leq 1$ results in identification of outliers. For finding the hypersphere center and its radius, one can use the Wolfe dual formulation and Lagrange multipliers which are explained by Richard J. Malak and Paredis (2009). Given a domain description, one can determine whether a candidate point is inside the valid domain by comparing its distance from the hypersphere center to the hypersphere radius.

One technique to improve the SVDD method is to use a nonlinear transformation of the data set which yields a simpler domain that is easy to describe. The strategy is to map the data into a higher-dimensional feature space. The domain description remains a hypersphere which can be described in the feature space. The kernel-based method improves the representational capabilities of SVDD since the classification model has a simple geometric (hypersphere).

Malak Jr and Paredis (2010) have used a Gaussian kernel function for nonlinear mapping. In this research, the same method is used to recast the SVDD as a kernel-based method. One important variable in Gaussian kernel function which affects the domain description is the “kernel width parameter, q ”. Larger values of q yield tighter domain boundaries for valid data sets. The smaller q yields a looser boundary for the data.

Identifying Gaussian Kernel width parameter q and C require trial and error for each data set. In this thesis, the value of regularization parameter is held at $C = 1$, since there is no outlier with a large distance from the other data. The width parameter q is selected

after trying different values and checking the boundary of the domain set. $q = 1$ results a consistent boundary for the valid input data set.

5.5.2 Applying SVDD to the optimization process

In this thesis, SVDD is applied in order to establish a domain which confines the stable area. By using this method, optimization algorithm does not need to compute stability of the robot in each iteration and for every point. Since computation of Poincare Map for stability analysis is time consuming, this approach reduces the computation time. By applying SVDD, Poincare map analysis is substituted by a nonlinear constraint which explores whether a candidate point in an iteration lies in the valid domain. The optimization still needs to check the convergence principle and find the convergent limit cycle in every step. The reason is that the limit cycle and its fixed-point affect the robot's trajectory and consequently the objective function value.

For using SVDD, first it is required to locate a domain which includes all the stable points. In order to find this domain, one needs to sample points with different dimensions, find their fixed points and then check their stability. By sampling about "52" points, the stable area is found. Sampling more points, provide opportunity for the optimization to search in a broader area but at the cost of more time for sampling. However, since in this thesis a gradient-based method (interior-point) is being used and local minimum is obtained, sampling more points does not produce any benefits. Providing a broader area is useful when a global optimization is being used. The sampled stable area can be bounded by applying SVDD method. In Figs 5-10 and 5-11,

the sampled stable and unstable points are shown. SVDD establishes a domain boundary around stable points with unstable points outside the boundary.

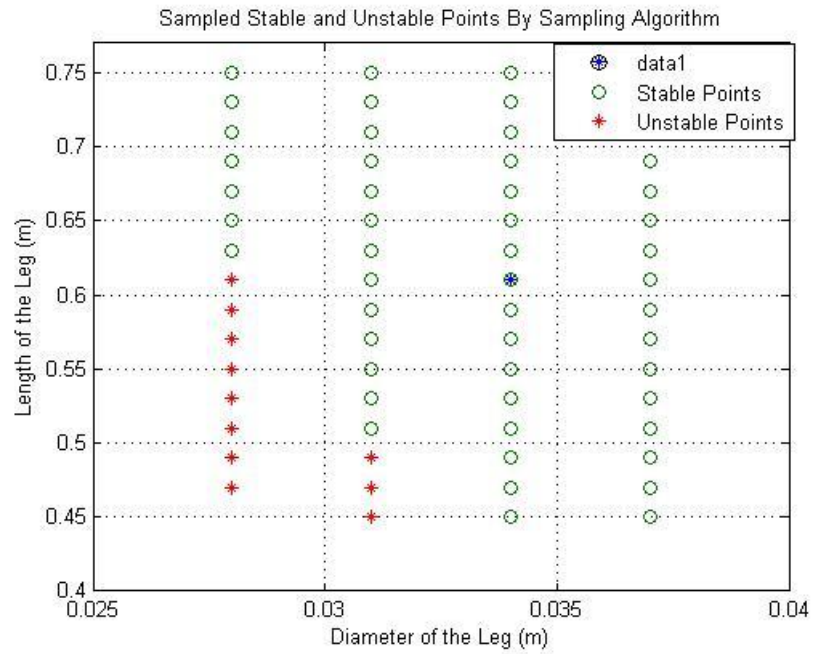


Fig. 5-10. Sampled stable and unstable points for optimization algorithm

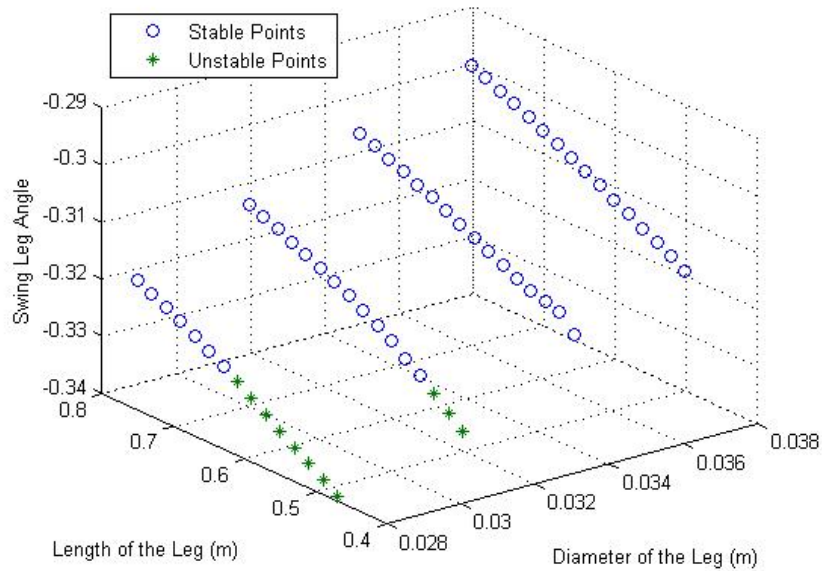


Fig. 5-11. Sampled stable and unstable points for optimization

In Fig 5-10, the points which are required to sample by SVDD are shown. The fixed-point of selected values of dimension are obtained manually and by trial and error. Then the stability of considered points is analyzed by the Poincare map. In Fig 5-10, the stable and unstable points are shown. Fig 5-11 shows the first element of fixed-point for the sampled points which are found manually. SVDD generate a domain descriptive to check whether a candidate point is in the domain. In order to check this condition, the following constraint has to be satisfied:

$$g(x) = R^2(x_i) - R_{\text{sph}}^2 \leq 0 \quad (5-18)$$

where R_{sph} is the radius of hypersphere created by SVDD and $R(x_i)$ is the distance between the SVDD hypersphere centroid and sampled points x_i . In this thesis, the stable

data are scaled in the range $[-1,1]$ which centers the hypersphere on 0. Finding the hypersphere centroid and scaling all other data around the centroid is performed according to lower and upper bounds. (only dimensional variables are included in the lower and upper bounds). Fig 5-12 shows the domain specified by SVDD approach.

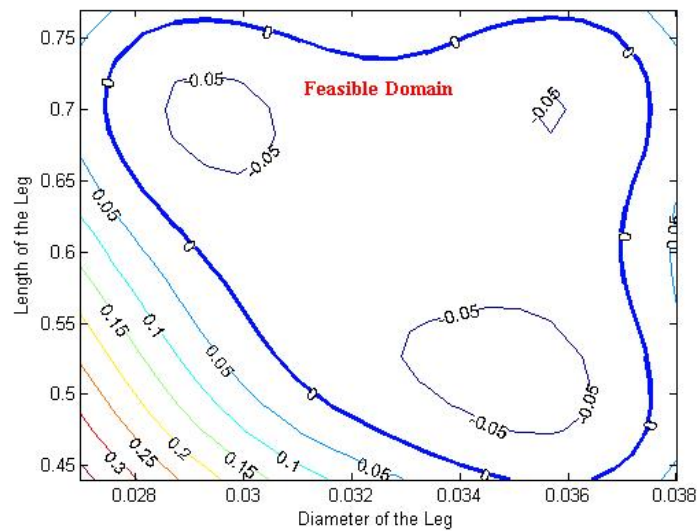


Fig. 5-12. Contour plot of constraint $g(x)$, acceptable domain for optimization

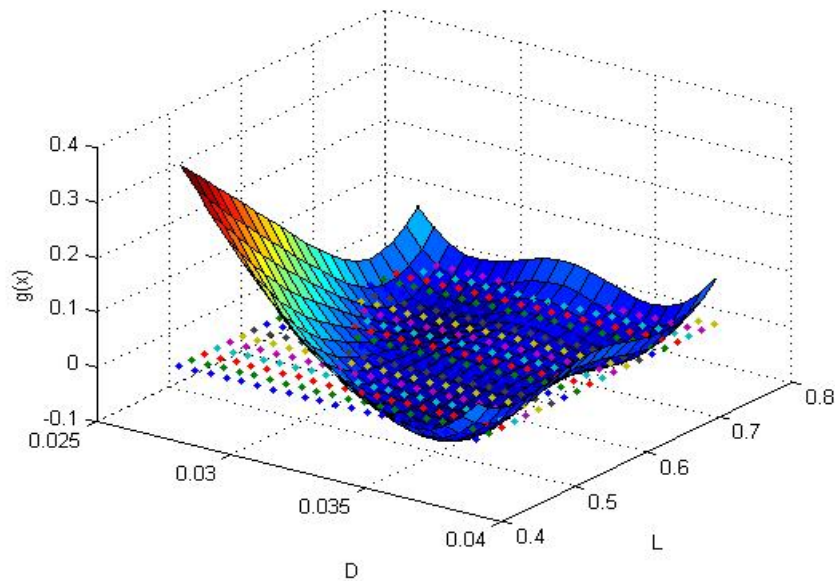


Fig. 5-13. The value of constraint $g(x)$ for different magnitude of variables. acceptable domain is $g(x) < 0$

As it can be seen in Fig 5-12, SVDD establishes a boundary around the stable domain. The boundary surrounds the area in which the nonlinear constraint $g(x) \leq 0$ is satisfied. This acceptable domain is shown in Fig 5-12 as the feasible domain. The optimization needs to search just within the feasible domain to find the optimum dimensions. Fig 5-13 demonstrates the values of $g(x)$ for different values of dimensions. As it was mentioned, the domain in which $g(x) \leq 0$ is the acceptable domain. In Fig 5-13, this domain is below the imaginary surface.

5.5.3 Optimization problem statement by applying SVDD method

As it was demonstrated in the previous section, the SVDD represents an additional nonlinear constraint $g(x)$ in the optimization problem in order to check whether the

candidate data lie inside the valid domain. By adding this constraint which confines the optimization to search in the stable domain, it is not necessary to check the stability constraint at each step, and it saves some time. Optimization still needs to check the convergence condition and find the fixed-point in every iteration since the objective function has to be evaluated after the convergence condition is satisfied. In addition, the fixed point (which is the initial condition in the limit cycle) affects the dynamic model of the system so it has to be computed for every updated structure in optimization iterations. The scheme of optimization with stability constraint is shown in Fig 5-14. The optimization problem can be stated as follows:

$$x^* = \underset{x}{\operatorname{argmax}} f(x)$$

$$x = [d, l]$$

$$x_{lb} \leq x \leq x_{up}$$

$$f(x) = \alpha_1 M_E(x) + \alpha_2 (1 - L_k(x)) \quad (5-19)$$

$$g(x) = R^2(x_i) - R_{sph}^2 \leq 0$$

$$h(x) = \|x_k^* - P(x_k^*)\|_2 = 0, \quad k = 20$$

$$\theta = [\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s], \quad \text{initial condition of gait}$$

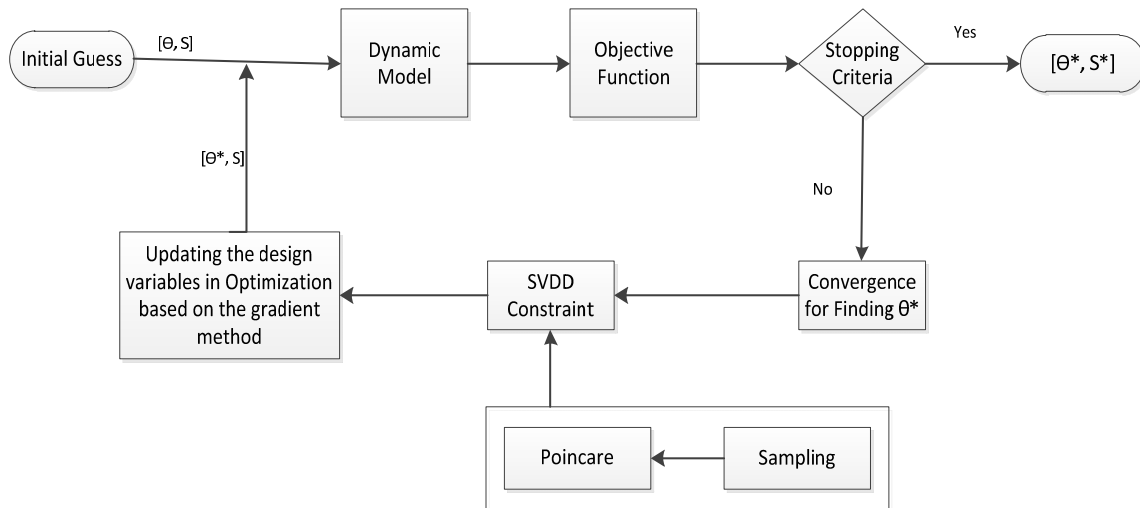


Fig. 5-14. Scheme of the optimization algorithm with SVDD constraint

5.5.4 Optimization (SVDD) result

The results of optimization by applying the optimization algorithm along with SVDD method, are as follows:

Initial guess: $[d, l, \theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]^T = [0.034, 0.73, -0.27357, -0.50847, -1.4707]^T$

Optimum Structure: $d = 0.028, l = 0.75$

Fixed-Point for Optimum Structure: $[-0.30224, 0.30224, -0.25087, -1.31504]^T$

Normalized Objective Function= 0.2203

Mechanical Energy: 61.98 J

Step Length: 0.4668m

Total Time of Optimization:

- T1: Time for deriving the fixed-points for sampling the data and checking the stability in order to establish the search domain=219.34 s

- T2: Time for running SVDD: 35.47s
- T3: Time of optimization: 434.4727s

Total time for optimization= $219.34+35.47+434.4747= 689.28s$

Time of optimization without sampling= 469.94 s

5.6 Comparison between Ordinary Optimization and Optimization with SVDD

As it was explained in the previous section, optimization with convergence and stability constraint is a time-consuming process. In section (5-5), SVDD method was used to replace the stability constraint with a constraint which confines the stable points and force the optimization algorithm search within this domain. Using this method is helpful to save time since checking the updated variables lie in the feasible domain takes less time than checking the stability constraint however sampling the data and checking their stability is still time-consuming process which increases the total optimization time. In the following, the table which compares optimization with SVDD and without SVDD and their result is shown.

Table 5-1. Comparison of Optimization Methods with SVDD and Without SVDD

	Optimum Objective Function	Mechanical Energy	Step Length	Optimization time	Optimization Time without sampling
Optimization with Convergence and Stability Constraint	0.2631	70.42	0.43481	746.2 s	746.2
Optimization with Convergence and SVDD Constraint	0.2203	61.98	0.4668	689.28 s	469.94

As it can be seen from the above table, both methods result the local optimum. The SVDD decreases the optimization time. However, it is not comparable with the entire time of optimization because sampling the data for SVDD domain consists of checking Poincare map for each sampled point which is a time-consuming process itself.

For solving the problem of data sampling, an algorithm is developed to compute the fixed-point and convergent limit cycle for every structure. This algorithm will be explained in the next section and is very useful for optimization in term of time and broader searching domain of optimization.

6 DEVELOPING AN ALGORITHM TO FIND THE FIXED-POINT FOR DIFFERENT STRUCTURE AND ITS EFFECT ON OPTIMIZATION

In the previous section, it was shown that the optimization process is a time-consuming process since the optimization algorithm needs to find the fixed-point for every updated variable. In section 5, SVDD (Support Vector Domain Descriptive) was replaced to the stability constraint in optimization algorithm in order to decrease the time of optimization. However, it was not very effective because of finding the convergent limit cycle and its corresponding fixed-point is still required which is time-consuming process. In addition the sampling data for SVDD domain also takes lots of time.

In this this section, a computational algorithm is established which can find the fixed-point for various magnitude of variables in a short time. Using this algorithm in optimization helps the process to converge to optimum point in a minimum time. In addition, this algorithm is very useful for improving optimization as it allows for optimization in a broader domain. Finally, by adding SVDD along with this algorithm, the optimization process is improved in terms of time while expanding the feasible domain for optimization. In the following, the algorithm for finding the fixed-points and its contribution to the optimization process is discussed in more details.

6.1 Algorithm for Finding the Fixed Point

In this section, an efficient computational method is developed which allows any given orbit with any initial condition to converge into its stable periodic orbit and fixed-point. This algorithm named “Fixed-Point Finder” can be used to create a time efficient and accurate module for computing the fixed-point in optimization algorithm. In each iteration, optimization will call this module to find the fixed point and convergent limit cycle for updated $[d, l]$ pair and then proceed to the next step. The “FPF” module consists of sampled pairs $[d, l]$ with the associated fixed-points. Following steps have to be taken in order to create the sampled data and use them in “FPF” module:

- 1- First, the optimization range and the desired pairs of $[d, l]$ in which the fixed-point needed to be computed are determined. Sampling pairs $[d, l]$ are generated with small increment between each consecutive term. It is noted that the increment between any two consecutive pair should not be too small which will increase the run period. It also should not be very large to keep the $[d, l]$ pair within the domain of attraction of the previous $[d, l]$ pair.
- 2- In the first step, the first $[d, l]$ pair is selected and its fixed-point is found using trial and error method and entered in the code. This point is referred as “reference pair”.

Table 6-1. The Reference Pair and the Consecutive Pairs Which Their Fixed-points have to Be Computed.

	d	l	Fixed Point			
Reference Pair	0.02	0.3	-0.3180544	0.318054	-	-
Consecutive pairs	0.02	0.35				
	0.025	0.3				
	0.025	0.35				
	-	-				

- 3- The developed algorithm “FPF” is capable of finding the remaining fixed points for other $[d, l]$ pairs based on the “reference pair”.
- 4- For the next $[d, l]$, the algorithm uses the fixed-point of “reference pair” as the initial condition of robot’s trajectory in order to find its limit cycle.
- 5- Since, the next $[d, l]$ pair is chosen within the domain of attraction of the reference pair, its limit cycle will converge to the attractive periodic orbit after infinite steps. then “Fixed-Point Finder” algorithm uses from convergence principle to obtain the periodic orbit of the next $[d, l]$ pair and its fixed point.
- 6- For the remaining $[d, l]$ the algorithm proceeds in the similar method. The only difference is that for the new $[d, l]$ pairs, it selects the previous known pair and uses its fixed-point as initial condition in order to find the limit cycle of new pair.
- 7- This process proceeds to find the fixed-point of all the $[d, l]$ pairs.
- 8- After sampling all the $[d, l]$ pairs in the optimization range and finding their fixed-points, the data can be used as a reference for “FPF”.

9- Optimization process uses this algorithm as a module for finding the fixed point of limit cycle for every updated structure in each iteration. For every updated dimensions of structure, “FPF” searches within its reference data list and find nearby $[d,l]$ pair which its distance from the updated $[d,l]$ is less than a specific threshold. It uses the fixed-point of the nearby pair as initial condition of trajectory for the updated structure to find its fixed point.

10-It is noted that this algorithm can also use for evaluating the eigenvalues of Poincare map in order to determine the stability of limit cycle for each sampled point faster. In Fig 6-1, the points which are sampled by “Fixed-point Finder” and the sampled stable and unstable points are shown.

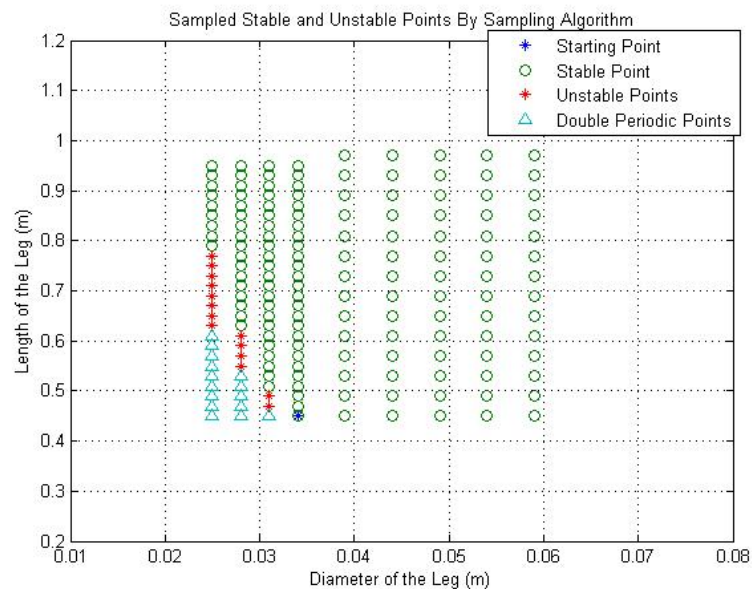


Fig. 6-1. Sampled stable and unstable points evaluated by
"Fixed-Point Finder algorithm"

As it is shown in Fig 6-1, the “FPF” algorithm starts with \ast point which is the reference point with a known fixed-point. It proceed to other sampled points respectively to find their fixed-point and analyze their stability. For example, the reference point is $[d,l] = [0.024,0.45]$ and the next point for which the fixed-point has to be found by “FPF” algorithm is $[d,l] = [0.028,0.45]$. It is noted that for $d \leq 0.025$, the size of the steps is smaller and for $d > 0.025$, the size of the steps is considered bigger. The reason is that for $d \leq 0.025$, some of the sampled points are unstable. It decreases the domain of attraction of them and also the domain of attraction of stable points close to them. Since, in the algorithm, finding the fixed-point of every pair $[d,l]$ is dependent on the previous point, it is necessary that every sampled $[d,l]$ lies in the domain of attraction of the previous point. Therefore, for $d \leq 0.025$, the step size between the sampled points are very small. As shown in Fig 6-1, for some $[d,l]$ pair, the robot’s gait is double periodic. The stability analysis in these points is not applicable since they do not have a specific fixed-point. In addition, they are considered as unfeasible domain since it is not desired that a optimum point result in double-periodic walking.

In Figs 6-2 and 6-3, the first and third elements of the fixed-point which are the swing leg angle and swing leg angular velocity respectively and are computed by the “Fixed-finder Algorithm” as shown.

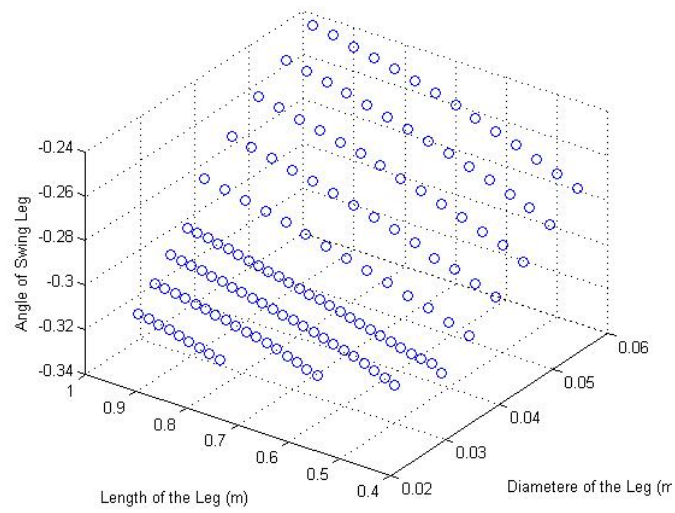


Fig. 6-2. The first element of fixed-point calculated By FPF algorithm

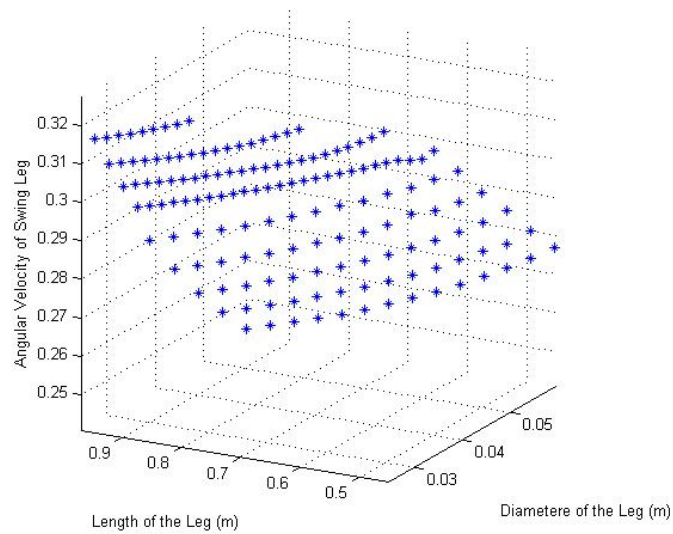


Fig. 6-3. The third element of fixed-point calculated by FPF algorithm

Fig 6-2 and 6-3 show the first and third element of the fixed-point for the sample data which are calculated by “FPF” algorithm. As it was explained, finding the fixed-point for each pair is dependent to the fixed-point of the previous pair. The “FPF” algorithm finds all the fixed point for sampled $[d, l]$ pairs very fast and accurately.

6.2 Applying “Fixed-Point Finder” to Optimization Process

As it was shown in Section 5, optimization algorithm with convergence and stability constraints is a very time- consuming process. By developing the “Fixed-Point Finder”, the optimization process can be accelerated since it does not need to check the convergence constraint in each iteration. The convergence limit cycle and fixed-point are rapidly computed in each iteration using “Fixed-Point Finder” algorithm and the stability of the limit cycle is evaluated for the updated variables. Applying this algorithm also allows the optimization process to search in a broader domain. In Section 5-4, it was shown that the search domain is limited by the domain of attraction of each updated point. By using this algorithm, which calculates the fixed-point for every updated structure, optimization gets the opportunity to search all over the assigned range.

The optimization problem with the “Fixed-Point Finder” is stated as follows:

$$\begin{aligned}
 x^* &= \underset{x}{\text{ARGMIN}} f(x) \\
 x &= [d, l] \\
 x_{lb} &\leq x \leq x_{up} \\
 f(x) &= \alpha_1 M_{\text{Eat}}(x) + \alpha_2 (1 - L_{\text{kat}}(x))
 \end{aligned}
 \tag{6-1}$$

$$g_i(x) = \text{eig}(\text{DP}(\theta^*)) < 1$$

where $M_{\text{Eat}}(x)$ and $L_{\text{kat}}(x)$ are the mechanical energy and step length which are calculated for the attractive limit cycle. The initial condition of the attractive limit cycle is $ic = [\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]$ which is calculated by “FPF” algorithm. The results of optimization for the following initial guess are as follows:

Initial guess: $[d, l]^T = [0.034, 0.73]^T$

Optimum Dimensional parameters: $d = 0.0315, l = 0.7274$

Fixed-Point for Optimum dimension: $[-0.3090, 0.3090, -0.2557, -1.3324]^T$

Normalized Objective Function= 0.2486

Mechanical Energy: 65.458 J

Step Length: 0.4428

Time of Optimization:

- T1: Time for sampling 174 data= 250.2s
- T2: Time of optimization: 241.23s
- Total time for optimization= 250.2+241.23= 491.43s

6.3 Applying “Fixed-Point Finder” along with SVDD

In the last section it was seen that the results of the optimization and its time was improved tremendously by applying “FPF” algorithm. In addition, by applying SVDD method which was explained in Section (5-5), the stability constraint can also be removed from optimization so the time of the process will be further decreased. Using “FPF” also speed up checking the Poincare map constraint because of calculating fixed-

point for the sampled data. In this section, both the “FPF” algorithm and the “SVDD” will be applied to the optimization problem. Therefore, two time-consuming constraints will be removed from the problem and optimization process will run faster.

As it was explained in Section 5-5, SVDD locates a domain which includes all the stable points. It establishes a domain boundary around stable points with unstable points outside the boundary. The constraint created by SVDD is stated as follows:

$$g(x) = R^2(x_i) - R_{\text{sph}}^2 \leq 0 \quad (6-2)$$

where R_{sph}^2 is the radius of hypersphere created by SVDD which includes all the stable points and $R^2(x_i)$ is the distance between the SVDD hyperrplane centroid and sampled points x_i . Fig 6-4 shows the domain specified by the SVDD method. It is noted that the domain in which the nonlinear constraint $g(x) \leq 0$ is satisfied represent the acceptable domain for optimization. Fig 6-5 shows the values of $g(x)$ versus different values of $[d, l]$ pairs.

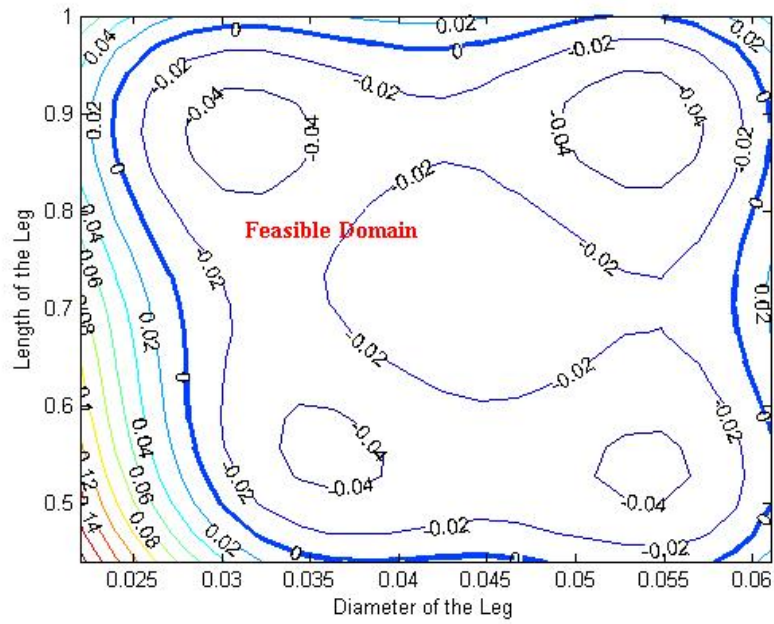


Fig. 6-4. Acceptable domain for optimization

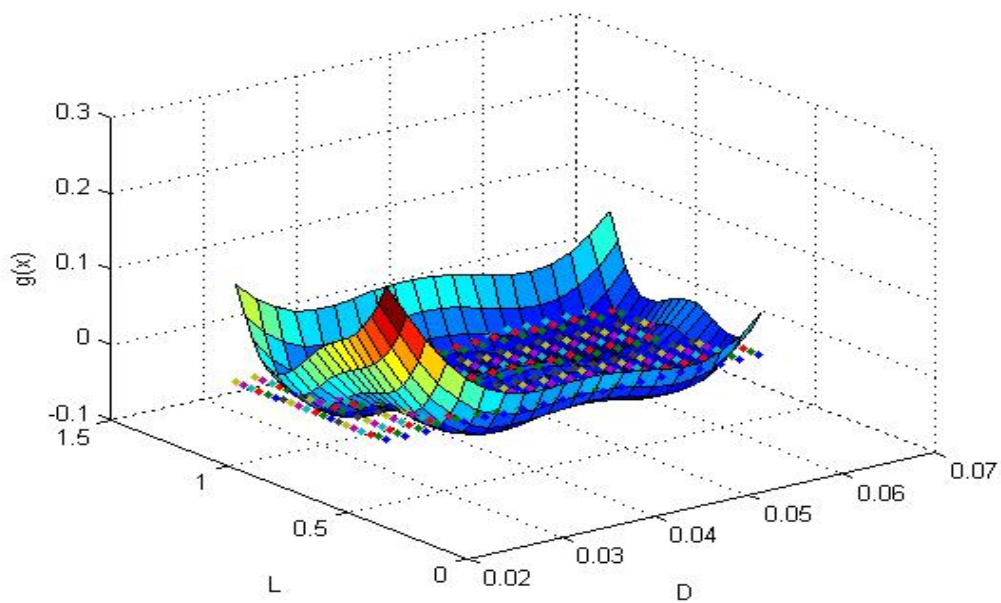


Fig. 6-5. Values of constraint $g(x)$. Acceptable domain is for $g(x) < 0$

As it can be seen in Fig 6-4, the boundary established by SVDD surround a broader area is broader in compare with Fig 5-12, since in this case, SVDD has taken the data from “FPF” algorithm which contains more sampled data. The domain in which the nonlinear constraint $g(x) \leq 0$ is satisfied is considered as the feasible point. The optimization needs to search within this area to find the optimum parameters. Fig 6-3 demonstrates the values of $g(x)$ for different values of $[d,l]$ parameters. As it was mentioned, the domain in which $g(x) \leq 0$ is the acceptable domain.

The optimization problem with “FPF” and SVDD is stated as follows:

$$\begin{aligned}
 x^* &= \underset{x}{\text{ARGMIN}} f(x) \\
 x &= [d, l] \\
 x_{lb} &\leq x \leq x_{up} \\
 f(x) &= \alpha_1 M_{\text{Eat}}(x) + \alpha_2 (1 - L_{\text{kat}}(x)) \\
 g(x) &= R^2(x_i) - R_{\text{sph}}^2 \leq 0
 \end{aligned} \tag{6-3}$$

where $M_{\text{Eat}}(x)$ and $L_{\text{kat}}(x)$ are the mechanical energy and step length calculated for the attractive limit cycle. $ic = [\theta_{ns}, \dot{\theta}_{ns}, \dot{\theta}_s]$ is the initial condition of the attractive limit cycle which is calculated by “FPF” algorithm. The results of optimization for the following initial l guess are as follows:

$$\text{Initial guess: } [d, l]^T = [0.034, 0.73]^T$$

$$\text{Optimum Structure: } d = 0.025, l = 0.9504$$

$$\text{Fixed-Point for Optimum Structure: } [-0.3145, 0.3145, -0.2272, -1.1616]^T$$

$$\text{Normalized Objective Function} = 0.1532$$

Mechanical Energy: 78.83 J

Step Length: 0.59311

Time of Optimization

- T1: Time for sampling 174 data and check their stability= 287.55s
- T2: Time for running SVDD: 53s
- T3: Time of optimization: 70.67 s
- Total time for optimization= 411.22 s

It should be noted that once we can obtain the sampled data list in “FPF” algorithm, we do not need to do the sampling for other optimization process and the same reference list can be used for other optimization runs. As it can be seen, the total time of optimization is decreased tremendously.

6.4 Comparison between Different Optimization Approaches

As it was illustrated in the previous section, optimization was improved tremendously by applying “FPF” and SVDD algorithm. By using FPF algorithm, the time for obtaining the fixed-point was decreased and also by replacing Poincare map constraint by the SVDD constraint, the optimization time was saved. In the following table, all the optimization application times and results are shown and compared. As it was mentioned in previous section, pre-computed sampled data can be reuse in any optimization algorithm.

Table 6-2. Comparison between Different Methods of Optimization

	Optimum Objective Function	Mechanical Energy	Step Length	Optimum Parameters	Optimization time+ Sampling	Optimization time
Nested Optimization with Poincare Constrain	0.4695	55.135	0.3592	d= 0.045, l= 0.58	1120s	1120s
Convergence and Poincare Constraint	0.2631	70.42	0.43481	d= 0.0337, l= 0.7329	746.2	746.2 s
Convergence and SVDD Constraint	0.2203	61.98	0.4668	d= 0.028, l= 0.75	689.28 s	469.94
Poincare Constraint and “FPF” algorithm	0.2486	65.458	0.4428	d= 0.0315, l= 0.7274	491.34	241.23 s
“FPF” algortihm and “SVDD” constraint	0.1532	78.83	0.59311	d= 0.025, l= 0.9504	411.22	123.67 s

It has to be mentioned that all the optimization in this thesis are performed in the computer Intel (R) Core (TM) Memory (Ram): 8 GB with system 64 bit.

As it can be seen from above table, applying SVDD alone improves the time of optimization but it is not very efficient since the convergence constraint is a very time-consuming process. By applying the “FPF” algorithm without SVDD, the time of optimization decrease a lot. By applying “FPF” and SVDD together, the time of optimization decreases more and feasible domain of optimization increases. It can be seen from above table 6-1 that the value of objective function has improved a lot. The reason is that the FPF+ SVDD method allows the optimization to search in a broader domain and to find better dimensions.

6.5 Robustness

In Fig 6-6, the result of different methods of optimization is shown on the bifurcation map. The bifurcation map allows one to check whether the obtained results are robust due to the changes in the values of the variables.

It has to be mentioned that the changes in the values of dimensional parameters does not affect the length ratio β since the center of mass does not change for all different parameter's values i.e., $\beta = 1$. The mass ratio μ is affected by dimensional parameters since the leg's mass is a function of its dimensions. In Fig 6-6, the length ratio is considered $\beta = 1$, and the mass ratio μ is shown for optimum dimensional parameters obtained by different methods of optimization.

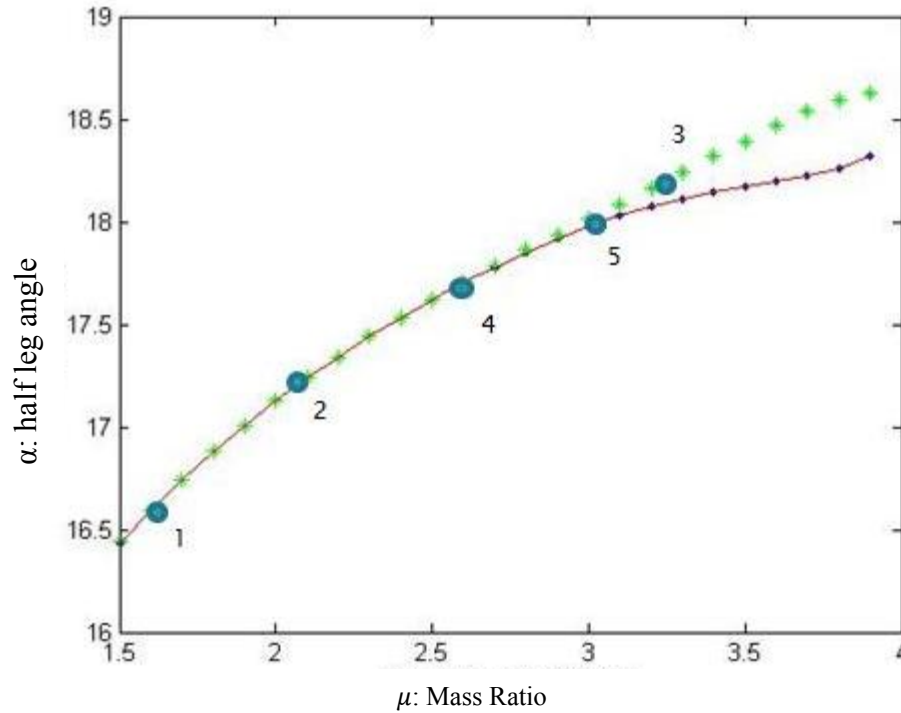


Fig. 6-6. Mass ratio for different methods of optimization

As it can be seen in Fig 6-6, the response of solution (1) (nested optimization), solution (2) (convergence method) and solution (4) (FPF Algorithm) are in the acceptable domain and their limit cycle is single-periodic. In these cases, the changes in the value of length or in the value of the diameter do not shift the limit cycle to the double-periodic limit cycle. Therefore the responses are robust and acceptable.

Solution (3) (Convergence method+ SVDD) result in double-periodic trajectory and it is not acceptable. However, the difference between the values of α in two consecutive cycles is very small, about 0.2° and the trajectory still can be considered as single-periodic one but the robustness is very weak and small changes in the length leads to a double-periodic motion. The reason of the weak robustness is that the border established

by SVDD is not accurately surrounded all the stable points. Some of the points are on the verge i.e., close to unstable or double-periodic points.

The answer of solution (5) (FPF algorithm + SVDD) is also acceptable (single-period) but not robust since it is very close to double-periodic motion. A small change (5%) in the value of the leg's length shifts the motion to the double-periodic one. In addition, if in some special conditions, the mass on the hip M_h increases for example if the robot wants to carry more load on its hip, the motion will shift to double-periodic one. As it was explained for solution (3), one of the reasons is that the border established around stable domain is not very accurate and is close to unstable domain. In addition, since the value of the objective function is optimum in the domain with larger length and smaller diameter, the optimization automatically shifts the search to that region which is also close to double-periodic motion. Specially, in solution (5) in which the optimization can search farther from its initial guess, this problem become more serious because the optimization will converge to that area. In order to solve this problem, one can establish the SVDD border such that it excludes the regions close to unstable and double-periodic domain. In Fig (6-7), the domain which has to be excluded form SVDD domain is shown. This allows the answer of the optimization be farther from double-periodic and unstable motion so it gives more robust answers.

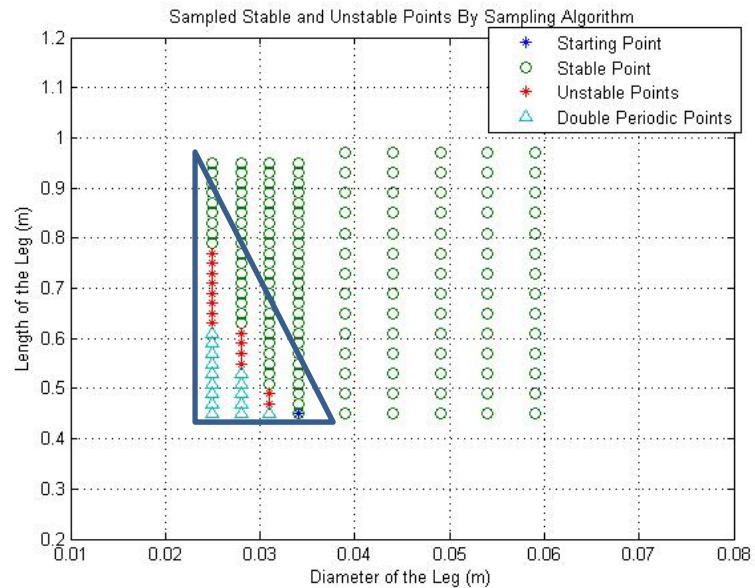


Fig. 6-7. The domain which has to be excluded from SVDD domain

6.6 Conclusion

Developing the FPF algorithm which finds the convergent limit cycle and its fixed-point in a short time is an important step to improve the optimization in terms of time and broader domain for search. This improvement can be seen from the results of optimization. The optimization time is $\frac{1}{4}$ of the nested method and the objective function is less than the one obtained in other methods. The problem of robustness can be solved by establishing the SVDD border farther from double-periodic and unstable points.

This method of optimization is very useful when we want to optimize the robots with higher dimension and non-linear dynamic. In addition, it should be noted that the compass robot considered in this thesis is a very simple robot and just two variables are considered as optimization variables. In more complex robot with more optimization

parameters, the time of optimization will be very large and this method will be very useful to decrease the process time.

7 CONCLUSION

The first part of the thesis focuses on the design of a compass robot by considering three components motor, gear and structure of the robot's leg. First, the robot's walking model and its required control input is derived. Then, this model is connected to the gear and motor model in a design process to evaluate the interaction of different components on each other. The torque-speed equation of the motor involving in the design process, affects the walking dynamics. Even though it is possible to ignore this constraint in the design as it was done in other researches, considering this constraint is helpful to estimate the effect of the motor on other parts and obtain a better performance of the system. After designing the system with its interacting components, one important step is to select the best actuator which consist of motor, gear and batteries on the hip and on the foot of the leg. Therefore, the different actuators are selected and their performance are compared through their efficiency and cost. In addition, the effect of the motor on the stability of the robot is investigated through Bifurcation map and Poincare map in order to insure the walking stability with the selected motor. The best motor was selected based on its efficiency, cost and its effect on the walking stability of the robot.

In the second part of the thesis, the structural optimization of the compass robot with stability constraint is investigated in order to achieve minimum consumed energy and maximum velocity. The main challenge of optimization of a compass robot as a hybrid system is the nonlinear stability constraint and its nonlinear, discrete dynamics which make the optimization problem very complicated. In this thesis, different

optimization strategies and their drawbacks are discussed. Then a new method, the periodic orbit convergence approach, is introduced to solve the problem. Applying the convergence principle to the optimization is used to obtain the convergent periodic orbit and its fixed point for different structure when the variables are updated in each optimization iteration. This constraint is helpful to analyze the stability around the fixed point and evaluate the system objective function in the convergent limit cycle. In addition, this approach is faster in compared with other methods such as nested optimization.

Even though, adding the convergence constraint improves the optimization extremely, it still has some limitation and drawbacks including time-consuming process and limited domain for search. One step for improving the optimization process is to use from a predictive model which examines whether a candidate point is in a feasible area or not. The kernel-based “Support Vector Domain Description”, SVDD is a method which creates a hypersphere which confines the stable domain. By applying SVDD to the optimization, the stability analysis by Poincare map can be replaced by the constraint which checks whether the updated variables in every optimization iteration are in the stable domain or not. Since the stability constraint is a time-consuming process, applying SVDD decreases the time of optimization.

Another important step which is done in this thesis in order to improve the optimization of hybrid systems is developing a computational algorithm named “FPF” which can find the convergent periodic orbit and consequently its fixed-point for every given structure of the robot. This algorithm is very useful in optimization problem since

it can compute the fixed-point very fast for every updated value of variables. Therefore, it replaces the convergent constraint which is a time-consuming process. Using this algorithm has two important advantages. First, it decreases the optimization time tremendously and second it increases the feasible domain in which the optimization explored for finding the best design variables. Combining “FPF” algorithm with SVDD is another step to improve the optimization time. As it was shown in the optimization results using FPF and SVDD methods together, improve the time and also the results of the optimization.

It has to be mentioned that compass robot is the simplest model of bipedal robots. So, for the complicated biped robots which have higher degree of freedom and more domains, using FPF and SVDD can be efficient to find the best performance criteria. Specially, when the numbers of variables are increased, this method is very useful to keep the optimization time low. Moreover, the “Fixed-Point Finder” algorithm is a proper method to find the convergent limit cycle not only in the biped robot but also in other hybrid systems. FPF algorithm along with SVDD method can be a very powerful tool to improve the optimization of Hybrid systems.

REFERENCES

- Akian, M., Gaubert, S., & Lemmens, B. (2011). Stability and convergence in discrete convex monotone dynamical systems. *Journal of Fixed Point Theory and Applications*, 9(2), 295-325.
- Ames, A. D. First steps toward automatically generating bipedal robotic walking from human data. In *Robot Motion and Control 2011* (pp. 89-116): Springer Verlag, London.
- Ames, A. D., Gregg, R., Wendel, E., & Sastry, S. (2006). *Towards the geometric reduction of controlled three-dimensional robotic bipedal walkers*. Paper presented at the 3rd Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, Nagoya, Japan.
- Ames, A. D., & Gregg, R. D. (2007). *Stably extending two-dimensional bipedal walking to three dimensions*. Paper presented at the American Control Conference, New York City, USA.
- Asta, S., & Sariel-Talay, S. (2011). *Nature-inspired optimization for biped robot locomotion and gait planning*. Paper presented at the Proceedings of the 2011 International Conference on Applications of Evolutionary Computation, Torino, Italy.
- Beitz, W., Pahl, G., & Wallace, K. (2003). *Engineering design: a systematic approach*. Springer, New York.
- Chevallereau, C., & Sardain, P. (2000). *Design and actuation optimization of a 4-axes biped robot for walking and running*. Paper presented at the Proceedings of IEEE International Conference on Robotics and Automation, San Francisco, CA.
- Goswami, A., Thuirot, B., & Espiau, B. (1996). Compass-like biped robot part I: Stability and bifurcation of passive gaits. *Technical Report 2996, Institut National de Recherche en Informatique et en Automatique*.
- Grizzle, J. W., Abba, G., & Plestan, F. (2001). Asymptotically stable walking for biped robots: analysis via systems with impulse effects. *IEEE Transactions on Automatic Control*, 48(1), 51-64.

- Grizzle, J. W., Chevallereau, C., Ames, A. D., & Sinnet, R. W. (2010). *3d bipedal robotic walking: Models, feedback control, and open problems*. Paper presented at the 8th IFAC Symposium on Nonlinear Control Systems, Bologna, Italy.
- Grodzevich, O. a. R. O. (2006). *Normalization and other topics in multi-objective optimization*. Paper presented at the Proceedings of the First Fields – MITACS Industrial Problems Workshop, Toronto, Canada.
- Hiskens, I. A. (2001). *Stability of hybrid system limit cycles: Application to the compass gait biped robot*. Paper presented at the 40th IEEE Conference on Decision and Control, Florida, USA.
- Iida, F., & Tedrake, R. (2007). *Motor control optimization of compliant one-legged locomotion in rough terrain*. Paper presented at the Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA.
- Lima, J., Gonçalves, J., Costa, P., & Moreira, A. P. (2010). *Humanoid robot gait planning resorting to an adaptive simulated annealing algorithm*. Paper presented at the 10th Conference on Autonomous Robot Systems and Competitions, Leiria, Portugal.
- Malak Jr, R. J., & Paredis, C. J. J. (2010). Using support vector machines to formalize the valid input domain of predictive models in systems design problems. *Journal of Mechanical Design*, 132, 101001.
- McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, 9(2), 62-82.
- Morris, B., & Grizzle, J. (2005). *A restricted Poincaré map for determining exponentially stable periodic orbits in systems with impulse effects: Application to bipedal robots*. Paper presented at the IEEE Conference on Decision and Control, CDC-ECC Seville, Spain.
- Ocken, S. (1995). Recognizing convergent orbits of discrete dynamical systems. *SIAM Journal on Applied Mathematics*, 55, 1134-1160.
- Ogata, K. (2004). *System dynamics*: Pearson/Prentice Hall, Englewood Cliffs.
- Olaru, I. M. C., Krut, S., & Pierrot, F. (2009). *Novel mechanical design of biped robot SHERPA using 2 DOF cable differential modular joints*. Paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA.

- Oliveira, M., Costa, L., Rocha, A., Santos, C., & Ferreira, M. (2011). Multiobjective optimization of a quadruped robot locomotion using a Genetic algorithm. *Soft Computing in Industrial Applications*, 96, 427-436.
- Pedersen, M. M., Nielsen, A. A., & Christiansen, L. F. (2007). Design of biped robot AAU-BOT1, *Master thesis, AALBORG University, Aalborg, Denmark*.
- Perko, L. (1991). *Differential equations and dynamical systems*: Springer-Verlag, New York.
- Richard J. Malak, J., & Paredis, C. J. J. (2009). Using support vector machines to formalize the valid input domain of models in data-driven predictive modeling for systems design. *ASME Conference Proceedings*, 2009(48999), 1423-1436.
- Spong, M. W., & Bullo, F. (2002). *Controlled symmetries and passive walking*. Paper presented at the 15th Triennial World Congress, Barcelona, Spain.
- Spong, M. W., & Bullo, F. (2005). Controlled symmetries and passive walking. *IEEE Transactions on Automatic Control*, 50(7), 1025-1032.
- Tax, D. M. J., & Duin, R. P. W. (1999). Support vector domain description. *Pattern Recognition Letters*, 20(11-13), 1191-1199.
- Tesfu, M. T., Schlattmann, J., & Ziemer, L. (2008). *Simulation based methodology for selection and integration of real time control electronics into complex dynamic mechanical systems*. Paper presented at the Proceedings of The IAJC-IJME International Conference, Nashville, TN.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5), 988-999.
- Westervelt, E. R., Grizzle, J. W., Chevallereau, C., Choi, J. H., & Morris, B. (2007). *Feedback control of dynamic bipedal robot locomotion*: CRC press, Boca Raton, Florida.
- Ye, S., Dede, M. I. C., Nasser, S., & Tosunoglu, S. (2005). *Cerberus the humanoid robot: Part II—component selection and manufacturing*. Paper presented at the Florida Conference on Recent Advances in Robotics, Florida, USA.
- Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1), 59-60.

APPENDIX A

	RE25 20w	RE 25 10w	RE-max 21 5w	RE-max 29 15w
	Order No. 118752	Order No. 118746	Order No. 221011	Order No. 226754
SPG (rpm/Nm)	$38.1 \cdot 10^3$	$36.1 \cdot 10^3$	$311 \cdot 10^3$	$32.3 \cdot 10^3$
Nls (rpm)	9550	5190	10400	5960
ST (Nm)	$257 \cdot 10^{-3}$	$144 \cdot 10^{-3}$	$33.6 \cdot 10^{-3}$	$185 \cdot 10^{-3}$
SC (A)	11	3.28	4.09	4.81
R _a (ohm)	2.19	7.31	2.20	4.99
J (kgm ²)	$10.7 \cdot 10^{-7}$	$10.5 \cdot 10^{-7}$	$2.18 \cdot 10^{-7}$	$13.2 \cdot 10^{-7}$
K _b (Volt/rpm)	1/407	1/217	1/1160	1/249
K (Nm/A)	$23.4 \cdot 10^{-3}$	$44 \cdot 10^{-3}$	$8.23 \cdot 10^{-3}$	38.4
NT (Nm)	$26.7 \cdot 10^{-3}$	$28.8 \cdot 10^{-3}$	$6.21 \cdot 10^{-3}$	$26.1 \cdot 10^{-3}$
T _m (s)	$4.28 \cdot 10^{-3}$	$3.97 \cdot 10^{-3}$	$7.08 \cdot 10^{-3}$	$4.48 \cdot 10^{-3}$

APPENDIX B**Gearhead GP 16 A , Order No. 134784:**

Gear Reduction Ratio (NN): 2458

Maximum Continiuouss Torque (Mt): 0.3

Maximum Efficiency (E2): 0.59%

Gear Weight (kg): 35×10^{-3}

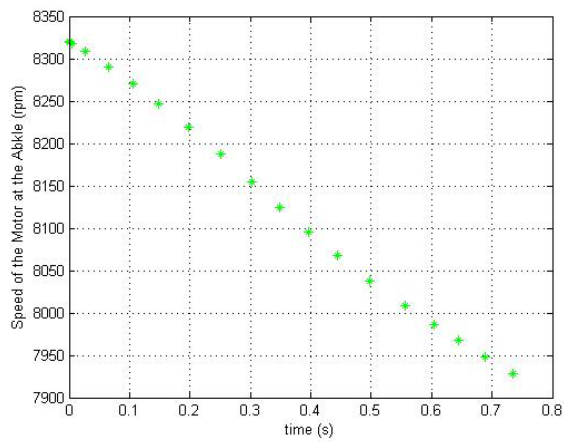
Gear Mass Inertia (kg/m^2): 0.05×10^{-7}

APPENDIX C

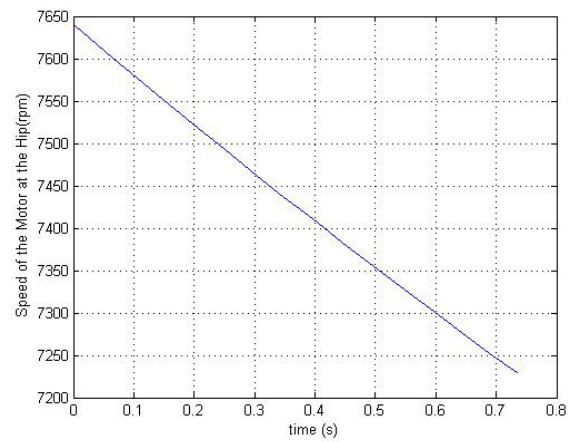
Performance plot:

Motor RE25 10w (Order No. 118746) and gear GS 45A, $n=310$ (Order No. 301185)

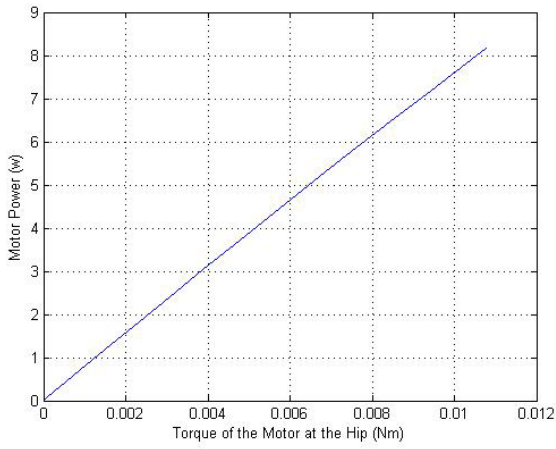
Motor RE-man21 5w (Order No. 221011) and gear GS 45A, $n=9$ (Order No. 301179)



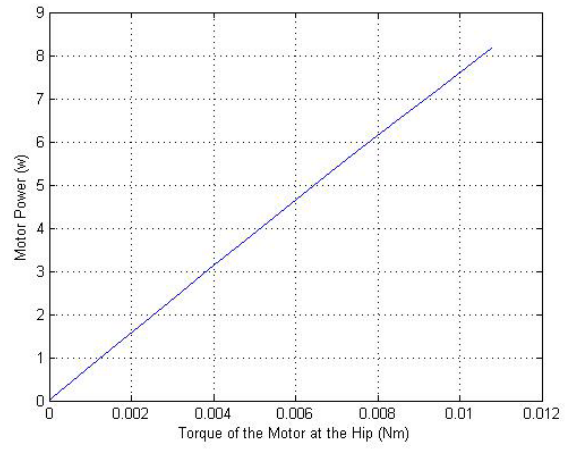
Speed Diagram of the Motor at the Foot



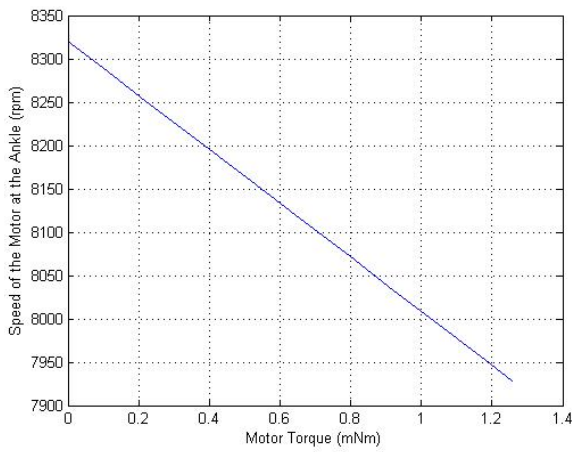
Speed Diagram of the Motor at the Hip



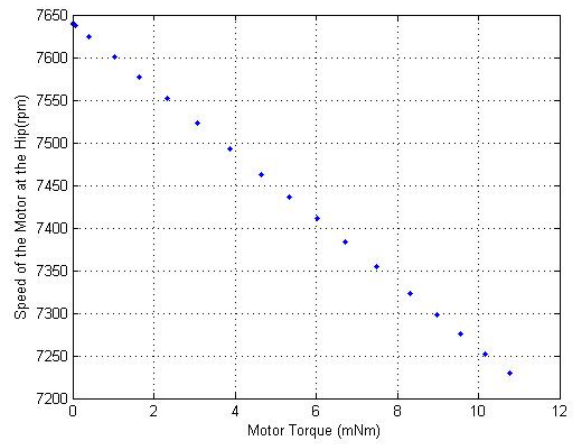
Power-Torque Diagram of the Motor at the Foot



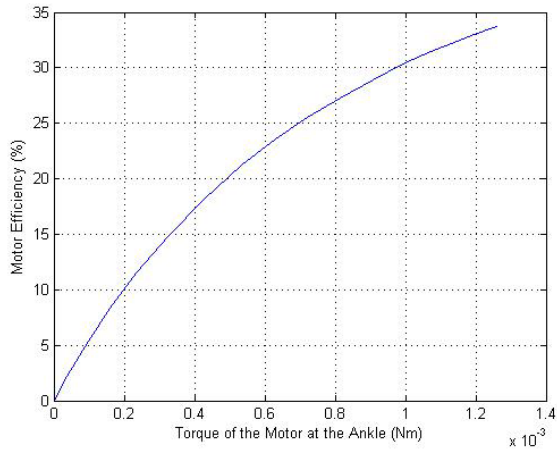
Power-Torque Diagram of the Motor at the Hip



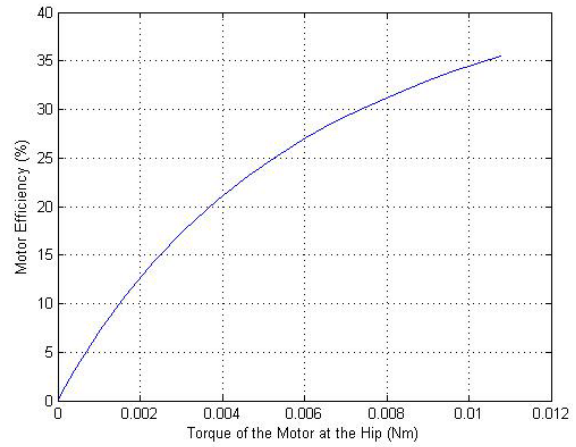
Torque-Speed Diagram of the Motor at the Foot



Torque-Speed Diagram of the Motor at the Hip



Efficiency of the Motor at the Foot

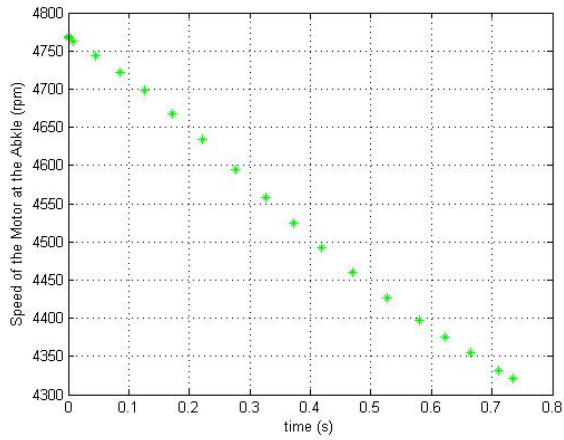


Efficiency of the Motor at the Hip

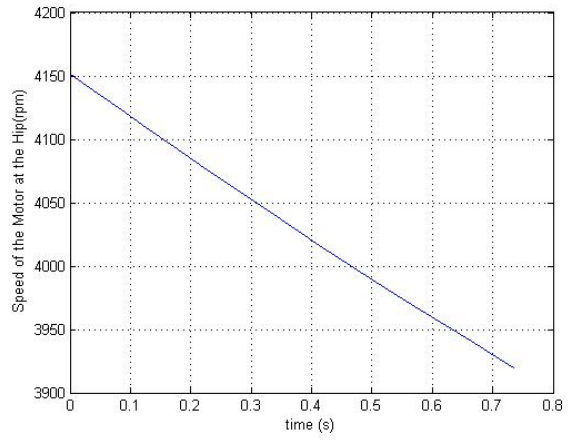
2-

Motor RE25 10w (Order No. 118746) and gear GS 45A, n=310 (Order No. 301185)

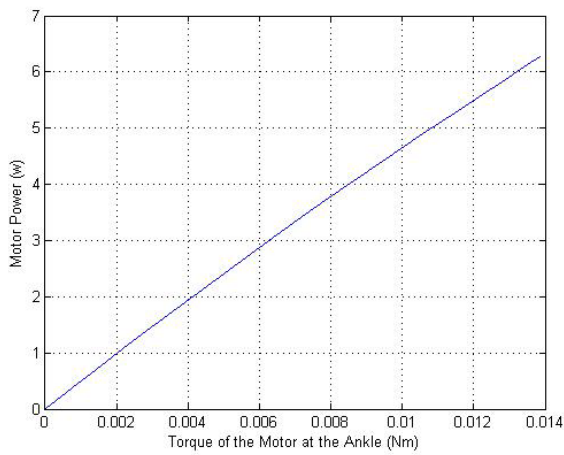
Motor RE-max 29 15w (Order No. 226754) and gear GS 45A, n=9 (Order No. 301179)



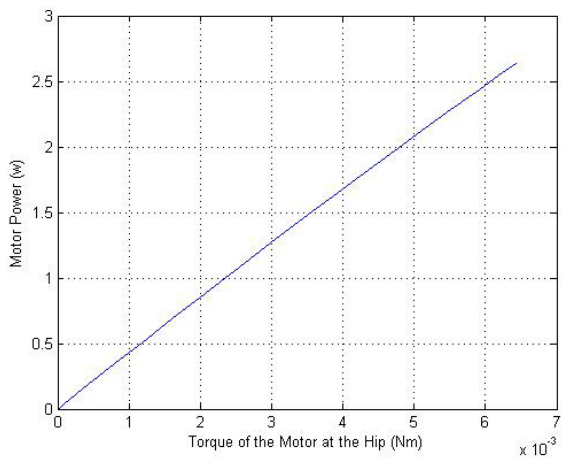
Speed Diagram of the Motor at the Foot



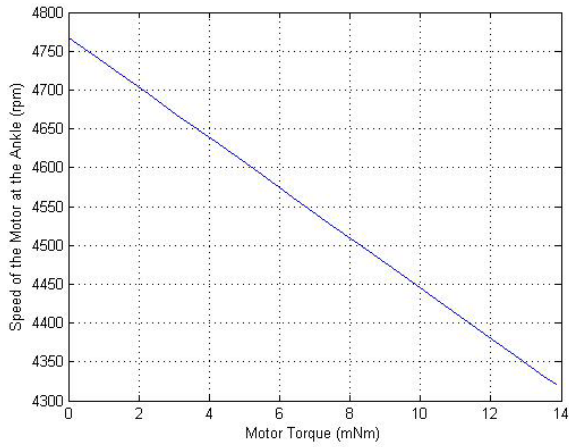
Speed Diagram of the Motor at the Hip



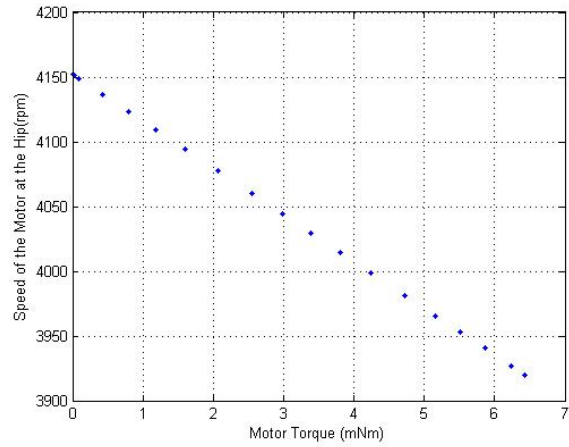
Power-Torque Diagram of the Motor at the Foot



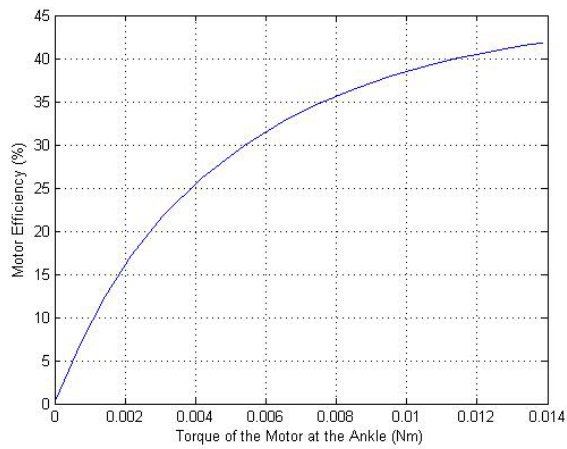
Power-Torque Diagram of the Motor at the Hip



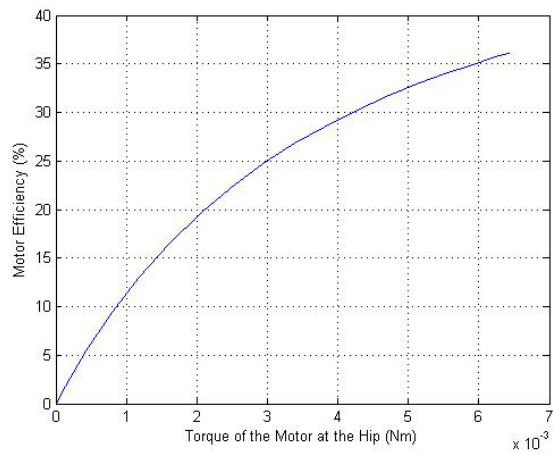
Torque-Speed Diagram of the Motor at the Foot



Torque-Speed Diagram of the Motor at the Hip



Efficiency of the Motor at the Foot



Efficiency of the Motor at the Hip

VITA

Name: Zohreh KeshavarzBagheri

Address: Texas A&M University,
Department of Mechanical Engineering,
3123 TAMU,
College Station, TX 77843-3123

Email Address: zohrehkeshavarz@yahoo.com

Education: B.S. Mechanical Engineering, Tabriz University, 2000
M.S., Khajeh Nasir-al-Deen Toosi University, 2005
M.S. Mechanical Engineering, Texas A&M University
2012