AN ALGORITHM TO RESOLVE THE OPTIMAL LOCOMOTION PROBLEM
OF
MODULAR ROBOTS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAKAN MENCEK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

DECEMBER 2007

Approval of the Thesis:

AN ALGORITHM TO RESOLVE THE OPTIMAL LOCOMOTION PROBLEM OF MODULAR ROBOTS

submitted by **HAKAN MENCEK** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ______________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Kemal İder ______________
Head of Department, **Mechanical Engineering**

Prof. Dr. Reşit Soylu ______________
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members :

Prof. Dr. Kemal Özgören ______________
Mechanical Engineering Dept., METU

Prof. Dr. Reşit Soylu ______________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Buğra Koku ______________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. İlhan Konukseven ______________
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Erol Şahin ______________
Computer Engineering Dept., METU

Date : 05.12.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Hakan MENCEK

Signature :

ABSTRACT

AN ALGORITHM TO RESOLVE THE OPTIMAL LOCOMOTION PROBLEM OF MODULAR ROBOTS

Mencek, Hakan
M.S., Department of Mechanical Engineering
Supervisor : Prof. Dr. Reşit Soylu
December 2007, 115 pages

In this study, a novel optimal motion planning algorithm is developed for the locomotion of modular robots. The total energy consumption of the robot is considered to be the optimization criteria. In order to determine the energy consumption of the system, the kinematic and dynamic analyses of the system are performed. Due to the variable number of modules in the system, a recursive formulation is developed for both kinematic and dynamic analyses. Coulomb's static and dynamic friction models are used to model the frictional forces at the contact points.

In modular robot locomotion, the number of contact points and the positions of the contact points vary with time. As a result, the structure of the dynamic equilibrium equations changes. Depending upon the number and type of contacts (i.e., contact with static or dynamic friction), the dynamic equilibrium equations may lead to an overdetermined, regular or underdetermined system of equations. The last case implies that the system is statically indeterminate. A novel solution method, which takes into account the deflections of the flexible links in the modular robot, is introduced to resolve this statical indeterminacy problem.

Another important contribution is the identification of the singularities associated with the dynamic equilibrium equations. It is shown that these equations become singular when all tangential contact point velocities are in the same direction. The developed optimal motion planning algorithm ensures that such singularities are avoided.

The procedure is illustrated via a modular, self reconfigurable robot called MTRAN. However, the method may be easily extended to other modular robots by changing the structural parameters. In order to display the resulting motion, a visual simulation program is developed for MTRAN using the commercial software MATHEMATICA.

Keywords : Modular Robots, Optimal Locomotion Planning, Static Indeterminacy, Contact Point Determination, Energy Optimization

ÖZ

MODÜLER ROBOTLARIN HAREKET PLANLAMASI SORUNUNU ÇÖZEN BİR ALGORİTMA

Mencek, Hakan
Yüksek Lisans, Makina Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Reşit Soylu
Aralık 2007, 115 sayfa

Bu çalışmada, modüler robotların en uygun hareketini bulmak için özgün bir algoritma geliştirilmiştir. İyileştirme kriteri olarak toplam enerji tüketimi göz önüne alınmıştır. Enerji tüketimini hesaplamak için robotun kinematik ve dinamik analizleri yapılmıştır. Değişken modül sayısı sebebiyle bu analizler için yinelemeli bir formülasyon geliştirilmiştir. Temas noktalarındaki sürtünmeyi modellemek için Coulomb' un statik ve dinamik sürtünme modelleri kullanılmıştır.

Modüler robotların hareketi sırasında, temas noktalarının sayısı ve konumu zamana göre değişir. Bu yüzden dinamik denge denklemlerinin yapısı değişmektedir. Temas noktalarının sayısına ve konumuna bağlı olarak dinamik denge denklemlerinin sayısı, bilinmeyenlerin sayısından çok, bilinmeyen sayısına eşit veya bilinmeyen sayısından az olabilmektedir. Denklem sayısının bilinmeyen sayısından az olduğu durum statik olarak belirsiz bir sisteme yol açmaktadır. Bu statik belirsizlik sorununu çözmek için modüler robot üzerindeki esnek uzuvların uzamasını veya kısalmasını göz önüne alan özgün bir çözüm yöntemi geliştirilmiştir.

Diğer bir önemli katkı da dinamik denge denklemleri ile ilgili tekilliklerin belirlenmesidir. Bu çalışmada, modüler robotun temas noktalarındaki teğet hızları

aynı yönde olduğu takdirde dinamik denge denklemlerinin tekil olduğu gösterilmiştir. Geliştirilen hareket eniyileştirme algoritması, tekil durumlardan sakınacak bir biçimde düzenlenmiştir.

Geliştirilen yöntem modüler ve otonom olarak şekil değiştirebilen bir robot olan MTRAN kullanılarak açıklanmıştır. Fakat, bu yöntem bazı yapısal parametrelerin değiştirilmesi ile diğer modüler robotlara da uygulanabilir. Bu çalışmada ayrıca elde edilen hareketi göstermek için ticari bir paket yazılım olan MATHEMATICA kullanılarak görsel bir benzetim ortamı geliştirilmiştir.

Anahtar Kelimeler : Modüler Robotlar, Eniyileştirilmiş Hareket Planlama, Enerji Eniyileştirmesi, Statik Belirsizlik, Temas Noktaları Tespiti

To My Family

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my thesis supervisor Prof. Dr. Reşit SOYLU for his guidance, encouragement and continuous support throughout the study.

I also would like to specially thank to my colleagues Mr. Ender ÖZDEN, Mr. A. Özgü NURSAL and Mr. A. Gürcan ÖZKİL for their support in different parts of the study.

I finally express my sincere thanks to my family for their endless support and motivation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

β_i	Orientation angle of the middle link of the i^{th} module before the loading
$\bar{\alpha}_{pi}, \bar{\alpha}_{li}, \bar{\alpha}_{ai}$	Column matrix representations of the angular acceleration vectors of the passive, middle and active links of the i^{th} module around their mass centers represented in $F_0(O_0)$
$\bar{\omega}_{pi}, \bar{\omega}_{li}, \bar{\omega}_{ai}$	Column matrix representations of the angular velocity vectors of the passive, middle and active links of the i^{th} module around their mass centers represented in $F_0(O_0)$
$\bar{\phi}_{ai}$	Orientation angle column matrix of the active part of the i^{th} module
$\bar{\phi}_{pi}$	Orientation angle column matrix of the passive part of the i^{th} module
$\bar{a}_{pi}, \bar{a}_{li}, \bar{a}_{ai}$	Column matrix representations of the linear acceleration vectors of the mass centers of the passive, middle and active links of the i^{th} module represented in $F_0(O_0)$
$\bar{d}_{mod,i}$	6×1 column matrix storing the vertical distances of all critical points in the i^{th} module
$\bar{d}_{sys}$	$6n \times 1$ column matrix storing the vertical distances of all critical points in n module system
$\bar{r}$	Column matrix representation of $\vec{r}$ in $F_0(O_0)$
$\bar{r}_{pi}, \bar{r}_{li}, \bar{r}_{ai}$	Column matrix representations of the position vectors of the mass centers of the passive, middle and active links of the i^{th} module represented in $F_0(O_0)$

$\bar{v}_{pi}, \bar{v}_{li}, \bar{v}_{ai}$	Column matrix representations of the linear velocity vectors of the mass centers of the passive, middle and active links of the i^{th} module represented in $F_0(O_0)$
δ_i	Axial deflection of the middle link of the i^{th} module
ϵ	Numerical tolerance
F_{ai}^x, F_{ai}^y	Reaction forces acting on the joint of the active link of the i^{th} module along x and y axis of the global reference frame by the middle link of the i^{th} module
F_{pi}^x, F_{pi}^y	Reaction forces acting on the joint of the passive link of the i^{th} module along x and y axis of the global reference frame by the middle link of the i^{th} module
T_{pi}, T_{ai}	Motor torques of the actuator on the passive and active links of the i^{th} module respectively
$\gamma_s^1, \gamma_s^2, \gamma_s^3$	Euler angles corresponding to $1-2-3$ rotational sequence from the global reference frame to the body fixed reference frame
$\hat{C}_{pi}, \hat{C}_{li}, \hat{C}_{ai}$	Rotational transformation matrices of the passive, middle and active links of the i^{th} module for transformation from the body fixed reference frame to $F_0(O_0)$
$\psi_{ij}^1, \psi_{ij}^2, \psi_{ij}^3$	Euler angles corresponding to $1-2-3$ rotational sequence between the connected links of the i^{th} and j^{th} modules
θa_i	Rotation angle of the actuator between middle and active links of the i^{th} module, measured from $\vec{x_{li}}$ to $\vec{x_{ai}}$ around $\vec{z_{li}}$ in a right hand sense.
θp_i	Rotation angle of the actuator between passive and middle links of the i^{th} module, measured from $\vec{x_{pi}}$ to $\vec{x_{li}}$ around $\vec{z_{pi}}$ in a right hand sense.
$\tilde{r}$	Cross product matrix representation of vector $\bar{r}$
$\vec{x}_0, \vec{y}_0, \vec{z}_0$	Orthogonal unit vectors of global reference frame
$\vec{x}_{ai}, \vec{y}_{ai}, \vec{z}_{ai}$	Orthogonal unit vectors of the body fixed reference frame attached to the active part of i^{th} module

$\vec{x}_{cm}, \vec{y}_{cm}, \vec{z}_{cm}$	Orthogonal unit vectors of the body fixed reference frame attached to center of mass of the overall system
$\vec{x}_{li}, \vec{y}_{li}, \vec{z}_{li}$	Orthogonal unit vectors of the body fixed reference frame attached to the middle link of the i^{th} module
$\vec{x}_{pi}, \vec{y}_{pi}, \vec{z}_{pi}$	Orthogonal unit vectors of the body fixed reference frame attached to the passive part of the i^{th} module
ζ_γ	Phase shift of the time dependent periodic function of the yaw angle of the system
ζ_{pi}, ζ_{ai}	Phase shifts of the joint variable functions of the actuators on the passive and active links of the i^{th} module
ζ_y	Phase shift of the time dependent periodic function of the y coordinate of the center of mass of the system
A_γ	Amplitude of the time dependent periodic function of the yaw angle of the system
A_{pi}, A_{ai}	Amplitudes of the joint variable functions of the actuators on the passive and active links of the i^{th} module
A_y	Amplitude of the time dependent periodic function of the y coordinate of the center of mass of the system
CE	Consumed energy
CI	A set of index numbers of $\bar{d}_{sys}$, which satisfy the contact points requirement
d	Unit length of MTRAN module
d_{min}	Minimum distance of the system to the ground
$F_0(O_0)$	Global fixed reference frame with origin O_0
F^x_{ci}, F^y_{ci}	Connection reaction forces of the passive link of the i^{th} module applied by the corresponding active link along x and y axes of the global fixed reference frame
$F_{cm}(O_{cm})$	Body fixed reference frame of the center of mass of the system with origin O_{cm}

f_i	Frictional reaction force applied by the ground to the link in contact at the i^{th} contact
$Fa_i(Oa_i)$	Body fixed reference frame of active part of the i^{th} module with origin Oa_i
$Fl_i(Ol_i)$	Body fixed reference frame of middle link of the i^{th} module with origin Ol_i
$Fp_i(Op_i)$	Body fixed reference frame of passive part of the i^{th} module with origin Op_i
g	Gravitational acceleration constant
H_{ij}	j^{th} axial force acting the middle link of the i^{th} module
I_p, I_l, I_a	Mass moment of inertias of the passive, middle and active links of MTRAN module
M_{ci}	Connection reaction moment of the passive link of the i^{th} module applied by the corresponding active link along x and y axes of the global fixed reference frame
m_p, m_l, m_a	Masses of the passive, middle and active links of an MTRAN module
MD	Minimum distance
n	Number of modules in the system
N_{cp}	Critical point number of a link
N_{eqn}	Number of equations
N_f	Number of free parameters
N_i	Normal reaction force applied by the ground to the link in contact at the i^{th} contact
N_{mod}	Module number
N_{part}	Type of the corresponding link of an MTRAN module, 1 for the passive link and 2 for the active link
N_{unk}	Number of unknowns

p_γ	Period factor of the time dependent periodic function of the yaw angle of the system
p_{pi}, p_{ai}	Period factors of the joint variable functions of the actuators on the passive and active links of the i^{th} module
p_y	Period factor of the time dependent periodic function of the y coordinate of the center of mass of the system
$R(\bar{n}, \theta)$	Rotational transformation matrix due to the rotation around vector $\vec{n}$ by an angle θ
RC	Number of contact points ratio
RF	Frictional reaction force ratio
RN	Normal reaction force ratio
RV	Contact point velocities ratio
s_l	Joint variable s after the loading
s_u	Joint variable s before the loading
T	Period of locomotion
t	Time
t_f	Final time
V_{ij}	j^{th} normal force acting the middle link of the i^{th} module
V_x	Horizontal velocity of the center of mass of the system
$x_{ai}^m, y_{ai}^m, z_{ai}^m$	x, y and z components of $r\vec{a}_i$ expressed in $F_0(O_0)$
$x_{aj,i}^{cr}, y_{aj,i}^{cr}$	x and y coordinates of the j^{th} critical point of active part of the i^{th} module in $F_0(O_0)$
x_{ci}, y_{ci}	x and y coordinates of the i^{th} contact point in global fixed reference frame
$x_{pi}^m, y_{pi}^m, z_{pi}^m$	x, y and z components of $r\vec{p}_i$ expressed in $F_0(O_0)$
$x_{pj,i}^{cr}, y_{pj,i}^{cr}$	x and y coordinates of the j^{th} critical point of passive part of the i^{th} module in $F_0(O_0)$

x_s, y_s, z_s	x, y and z coordinates of the mass center of the system, represented in $F_0(O_0)$
y_{s0}	Shifting factor of the periodic function of the y coordinate of the center of mass of the system
LHS	Left hand side
LHS $+=$ RHS	Operator, which add the RHS to the LHS
RHS	Right hand side

CHAPTER 1

INTRODUCTION

Traditional robotic systems are designed to perform specific tasks in a certain environment. Although they carry out their tasks successfully, they are unable to adapt to the different environmental conditions. Therefore, they are unable to operate efficiently in an unknown environment. Besides, if a part of the system fails, the robot has to be turned off until the system is repaired. In order to work properly under similar conditions, the robots should be more flexible. They should be able to adapt to new environmental conditions. Also, they should be more intelligent to replace or repair the failed part of the system. A modular, self-reconfigurable robot can successfully accomplish these tasks by changing its shape to a more convenient form.

Modular robots are robotic systems consisting of a set of autonomous modules, which are connected in an appropriate configuration to perform a required task. The modules are able to connect to, or disconnect from, the other modules so that the system can change its shape to a suitable form for the required tasks. In addition, the modules can move inside the system, so that a failed module can be replaced by another one.

The motions of a modular robot may be separated into two, namely, reconfiguration and locomotion. Reconfiguration is the motion with shape change by connection and disconnection of the modules, whereas, the locomotion is the motion of the robot to a target location without any shape change. In this thesis, modular, self reconfigurable robots and their locomotion problems have been studied.

Modular robots are relatively new robotic systems compared to classical robot manipulators. Hence, their advantages, classification and area of applications are discussed in the following sections. Furthermore, the physically realized modular robots and their motion planning methods are also presented.

1.1 Advantages of Modular Robots

Modular robots have several advantages over traditional robotic systems with fixed structure. Some of them are listed below.

- Modular robots are easier and more economical to build, since production of the same standard parts are required, instead of the production of different and complex parts.
- Modular robots are fault tolerant. If a module fails during an operation, the rest of the system can continue to work by replacing the failed module by an operational one.
- Modular robots can adapt to the changing environmental conditions by reconfiguring into an appropriate form. For example, a modular robot in a legged form can climb stairs, then reconfigure into a snake like form and crawl under an obstacle.
- Modular robots can have the ability to self-assemble. A set of disconnected modules can assume a suitable shape and then carry out the task.
- Modular robots can perform multiple tasks by means of the same set of modules.

1.2 Classification of Modular Robots

Modular robots can be classified into different groups according to their module types, structures, working environments, capabilities and control systems.

- According to Module Type:
  - Homogeneous Modular Robots are composed of identical modules. In order to work independently, each module has its own actuator(s), power supply, processing unit and sensor(s).
  - Heterogeneous Modular Robots have different kinds of modules. Each type of module can be suitable for different requirements of the system. For example, one type of module may carry batteries while another type of module may hold actuator(s) and sensor(s).

- According to Connection Type:
  - Chain Type Modular Robots are composed of modules, which are connected in the form of a string or tree.
  - Lattice Type Modular Robots have modules connected in a regular two or three dimensional pattern, like a cubic or hexagonal grid [1].
  - Hybrid Type Modular Robots can connect in a lattice type configuration. Furthermore, they can also perform tasks as chain type modular robots.
- According to Capabilities:
  - Self Reconfigurable Modular Robots can reconfigure into another form automatically.
  - Manually Reconfigurable Modular Robots require additional help from the user to change their shape.
- According to Working Environment:
  - 2D Modular Robots operate in a planar (2D) space
  - 3D Modular Robots perform tasks in a 3D environment.
- According to Control System:
  - Distributed Control Modular Robots have modules which are controlled by their own control systems.
  - Centralized Control Modular Robots are controlled by means of a central control system.
- According to Reconfiguration Method:
  - Deterministic Modular Robots reconfigure into different forms by continuously controlling the position of each module.
  - Stochastic Modular Robots know the location of modules only when they are connected to the main system. These robots operate in an environment that enables Brownian motion[1] like in zero-gravity space or in fluid surroundings [3].

[1]Brownian motion is the erratic movement of microscopic particles in a fluid, as a result of collisions with the surrounding molecules [2]

1.3 Areas of Applications

Modular self reconfigurable robots are versatile, fault-tolerant and self-assembling systems. These features provide large potentials for robotic applications in different fields.

First of all, modular robots are useful for tasks in unknown environments. They can be used for missions in space, or on the ocean floor. A modular robot, in a suitable form for transportation, can be sent to an environment. Then the robot reconfigures into an appropriate shape to adapt to the conditions and carry out the required tasks. In addition, modular robots can carry out search and rescue missions in natural disaster areas. For example, a modular robot can enter inside of a collapsed building through a small hole, avoid obstacles by reconfiguring into different forms and search the ruins for any survivors.

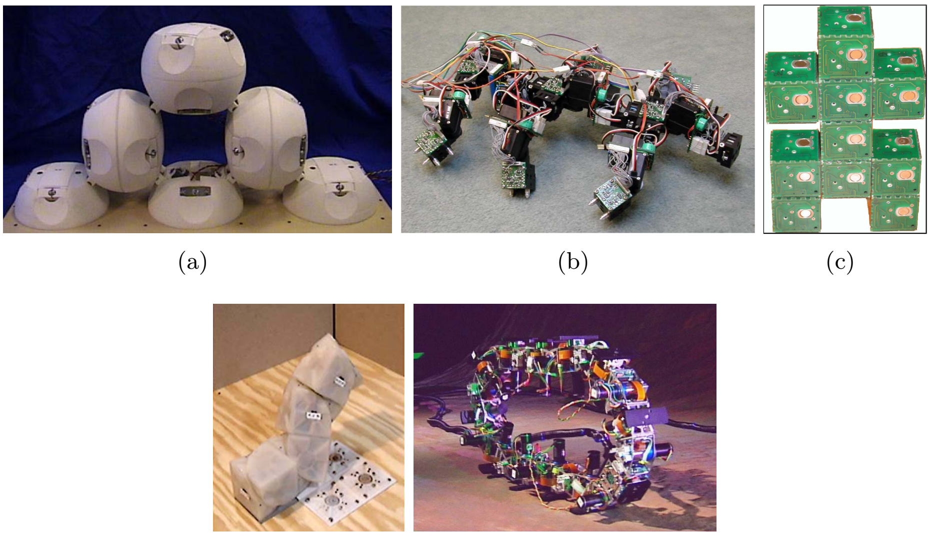
Modular robots can be used in the production lines in industry. Instead of using different and complex robot manipulators to perform various tasks, a set of modular robots may be used. They can perform different operations (welding, cleaning, painting, etc...) with different arrangements of the modules.

A long term vision for modular robots is called "Bucket of Stuff" [1]. In the future, consumers might have a container of self-reconfigurable modules. When required, the consumer calls for the robot to achieve a certain task, such as cleaning the room, changing the oil in the car and the robot reconfigures into a convenient form and performs the task.

In summary, the flexibility and versatility of modular robots lead to many possibilities for robotic applications in the future, making the studies in modular robotics well worthy.

1.4 Physically Realized Modular Robots

There has been a lot of researches on modular robots. Several modular robots are designed and built to study reconfiguration, locomotion and assembling of the systems. Some of the physically realized modular robots are listed in Table 1.1. Their structure types, work environments, module degrees of freedoms and developers are also presented in the same table. Physical appearances of some of these modular robots are shown in Figure 1.1.



(a) (b) (c)

(d) (e)

Figure 1.1: Example Modular Robots: (a) Atron [4] (b) CONRO [5] (c) Miche [6] (d) Molecube [7] (e) PolyBot [8]

Table 1.1: Physically Realized Modular Robots

Name	Structure	DOF	Workspace	Developer
ATRON	Lattice	1	3D	Jørgensen et al. [4]
Catoms	Lattice	0	2D	Kirby et al. [9]
CHOBIE	Lattice	2	2D	Koseki et al. [10]
CONRO	Chain	2	3D	Shen et al.[5]
Crystalline	Lattice	4	2D	Rus and Vona[11]
Fractum	Lattice	3	2D	Murata et al. [12]
I-Cubes	Lattice	3	3D	Unsal [13]
Metamorphic	Lattice	6	2D	Chirikjian[14]
Miche	Lattice	0	3D	Rus et al. [6]
Molecubes	Chain	1	3D	Zykov et al.[7]
MTRAN	Hybrid	2	3D	Murata et al. [15]
Pneumatic	Lattice	1	2D	Inou [16]
PolyBot	Chain	1	3D	Yim et al. [8]
Prog. Parts	Lattice	0	2D	Klavins et al.[17]
Stochastic	Lattice	0	3D	White et al. [3]
Superbot	Hybrid	3	3D	Shen et al. [18]
Telecube	Lattice	6	3D	Suh et al. [19]
Tetrobot	Chain	1	3D	Hamlin et al. [20]
Vertical	Lattice	1	2D	Hosokawa et al. [21]

1.5 Motion Planning of Modular Robots

Similar to classical robotic systems, modular robots require efficient motion strategies to accomplish their tasks successfully. In order to obtain the motion of a modular robot, motions of all modules in the system should be determined. Determination of the motions of all modules, for an efficient system motion, is called to be the motion planning problem of modular robots. Although a lot of effort has been spent on the realization of reconfiguration sequences and locomotion gaits of the modular robots, the studies to generate such motions automatically are limited. In this section, the research efforts for finding locomotion gaits and reconfiguration sequences for efficient modular robot motions are presented. Furthermore, the approach used in this thesis, for the solution of the motion planning problem, is introduced.

1.5.1 Reconfiguration

Reconfiguration is the shape change of the modular robot by means of the connection and disconnection between the modules. The connection and disconnection of the modules requires a lot of energy and long time. Therefore, efficient reconfiguration sequences are needed for modular robots.

One approach for the solution of the reconfiguration problem is dividing the problem into several subproblems. Kotay and Rus [22] have considered this approach and have divided the reconfiguration problem of a modular robot into three hierarchical levels, namely, trajectory planning, configuration planning and task-level planning. Trajectory planning level generates the motion for a single module, which moves from an initial position to a target position. Configuration planning level synthesizes the motion for a set of modules moving from a starting configuration to a goal configuration. Finally, task-level planning is used to plan the motion of the system according to the type of the required task. Similarly, Unsal and Khosla [23] also used hierarchical layers for the reconfiguration planning of I-Cubes. A multi-layered planner is developed to solve the reconfiguration problem in three levels, namely, metacube, cube and link levels. Metacube is defined as a uniform structure composed of a set of links and cubes. The planner uses Hungarian algorithm with Manhattan distance for the metacube motion planning and a heuristic search algorithm for the optimal cube motion planning. The link motions are generated by means of hard-coded basic link motions on the cube surfaces.

Evolutionary methods using genetic algorithms are also used in the solution of reconfiguration problem of modular robots. Yoshida et al. [24] developed a method called Evolutionary Reconfiguration Sequence Synthesis to solve the reconfiguration problem of MTRAN. In this method, different motion segments are defined for the basic module motions, including motor actuation and connection of modules. Then, a genetic algorithm is applied to find the reconfiguration sequences by taking the motion segments as the genes. The results are evaluated according to hardware limitations and module collisions. Several reconfiguration sequences were determined by means of this approach and they were experimentally verified with the MTRAN system. In addition, Mytilinaios et al. [25] and Zykov et al. [7] studied the self-replication of Molecubes by means of generic algorithms to find evolved physical replicators. In the genetic algorithm, a fitness function was used in order to have a feasible replication sequence.

Some novel algorithms are also developed for different types of modular robots to find reconfiguration sequences. For chain type modular robots, Yim et al. [26] presented a connectivity planning algorithm using canonical forms. This algorithm outputs attachment and detachment steps to reach the final configuration. Another planner is developed by Rus and Vona [11] for modular robots with compressible modules. Considering a set of modules, this planner takes a starting module, compresses it into an intermediate module and finally, expands the intermediate module into the target position.

Evaluation of the reconfiguration sequences according to some criterion has also been studied. Chiang and Chirikjian [27] introduced some metrics to include the type of reconfiguration in the optimization. These metrics were used to evaluate the performance of the reconfiguration motion according to the path length and the number of moves.

1.5.2 Locomotion

Locomotion is the translational motion of modular robots from one point to another without changing its configuration. For an efficient locomotion, different motion gaits should be obtained according to the requirement of the tasks.

To generate locomotion patterns for MTRAN, a locomotion pattern generation algorithm, using a central pattern generator, was developed by Yoshida et al. [24] and Kamimura et al. [28]. Neural oscillator model of central pattern generator was applied to each motor in the system. A genetic algorithm was used to evolve the initial values and the weights of the neural network in order to find a feasible locomotion gait. Then, the results were evaluated according to a fitness function designed to have faster locomotion in a straight line with minimum energy consumption. The dynamic analysis of the system during the pattern generation process was performed by means of a commercial dynamic simulation package, called Vortex. Similarly, Østergaard and Lund [29] successfully applied genetic algorithms to realize system locomotion of Atron. Experiments with different starting configurations were performed to evaluate the performance of the genetic algorithm. These experiments showed that evolutionary methods are useful for generating meaningful global behaviors in a self-reconfigurable modular robot in spite of a number of issues to be further studied.

Marbach and Ijspeert [30] used central pattern generators for a modular robotic system called YaMoR. Instead of a genetic algorithm, they successfully applied a classical function optimization algorithm, namely Powell's Method, to find the optimal locomotion gait. This optimization method is implemented into the central pattern generator and online optimization is achieved.

In this thesis, the optimal locomotion problem of modular robots is studied. The problem is converted into a parameter optimization problem. The variations of the joint variables are calculated by the optimization to have minimum energy consumption. In order to find the energy consumption and check the feasibility of the locomotion gait, a recursive formulation for the kinematic and dynamic analysis is developed. Coulomb friction is considered at the contact points and their effects are also included in the dynamic analysis. The method is illustrated by means of a modular, self reconfigurable robot, called Modular Transformer (MTRAN). All of the numerical and symbolic calculations are performed using the MATHEMATICA software and the numerical solution of the parameter optimization problem is achieved by means of the NMINIMIZE function of MATHEMATICA.

The outline of the thesis is as follows. In Chapter 2, the modular and self reconfigurable robot called the Modular Transformer is introduced. In Chapter 3, the kinematic analysis of modular robotic systems is discussed. The contact point determination of dynamic systems is explained in Chapter 4. The dynamic analysis of the modular robots is presented in Chapter 5. The definition and the solution of optimal locomotion problem is discussed in Chapter 6. Several case studies are presented in Chapter 7 in order to demonstrate the solution method, and finally, the conclusions of the study is given in Chapter 8.

CHAPTER 2

MODULAR TRANSFORMER

In this chapter, a modular and self reconfigurable robot, called Modular TRANsformer (MTRAN), is introduced. MTRAN is considered to be a hybrid type modular robot, since its modules can form both chain and lattice type structures. Since MTRAN has a hybrid type structure, it represents most of the modular robots in the literature. Therefore throughout this study, MTRAN system is used for the case studies to illustrate the methods followed for the solution of the optimal locomotion problem of modular robots.

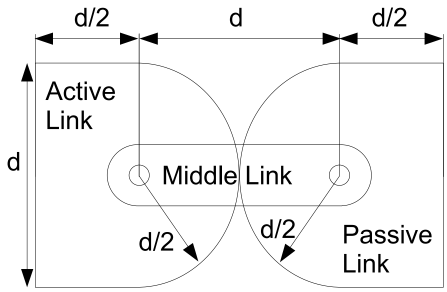
MTRAN is developed by Murata et al. in 1998 [15]. So far, several improvements have been made on the mechanical and electrical characteristics of the system. During this period, MTRAN version II and MTRAN version III have been designed. In this study, the specifications of the latest version is considered for the calculations.

MTRAN is a homogeneous modular robot composed of same type of modules. Each module has two degrees of freedom. Modules can move in a two dimensional space independently. Each module has male and female connection faces. Connection and disconnection between two modules are achieved automatically. By connecting several modules in a certain configuration, MTRAN is able to move in three dimensional space.

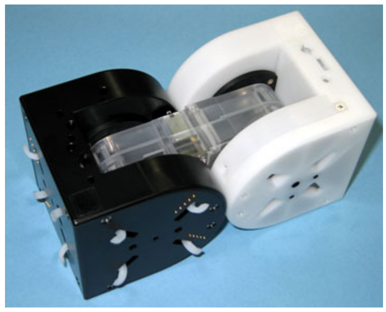
2.1 MTRAN III Module Design

Each MTRAN module consists of three parts, namely passive, middle and active links. Passive and active links have semi-cylindrical and semi-cubic form. The middle link is also called as the link. They are connected by the middle link via two actuators. If the diameter of the semi-circle in passive/active link is denoted by d, then the other link dimensions become d and $d/2$ as shown in Figure 2.1(a). For

MTRAN III, the value of d is $65mm$. Therefore each module is $65 \times 130 \times 65mm$ in size. The weight of a single module is $420g$ [31].



(a)



(b)

Figure 2.1: MTRAN III module : (a) Special dimensions (b) Physical appearance [32]

2.1.1 Actuation

Two servo motors connects three links and provides two degrees of freedom with parallel rotation axes as shown in Figure 2.2. They are mounted inside the middle link. Each motor has a rotational range of $\pm 90°$. The circuit board controlling the motors is also in the middle link. The position of the motors are controlled by a PID control system. Each motor is capable of providing $19.8kg - cm$ torque. This value is enough to lift up two modules [33].

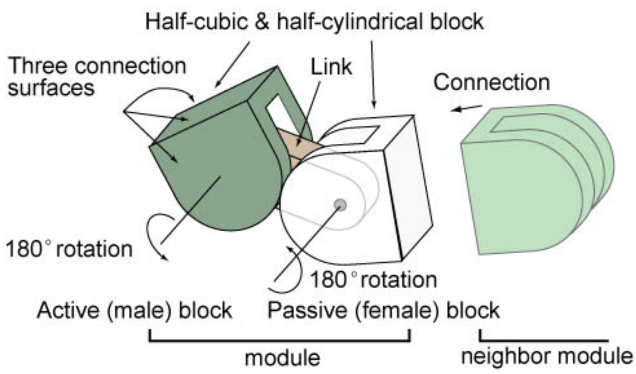


Figure 2.2: Module design of MTRAN III [32]

2.1.2 Connection Mechanism

The connection of the modules is achieved between passive and active links through the connection surfaces. Each link has three connection surfaces. Passive links has female connection surfaces, whereas active links has male connection surfaces. In the first two versions of the system, permanent magnets and SMA coils were used for the connection mechanism. However, a mechanical connector has been designed for the latest version of the module to replace the time and energy consuming magnetic system. More information related to the connection by permanent magnets and SMA coils can be found in [15] and [33]. For the connection between the modules of the latest version, retractable hook mechanism, which is shown in Figure 2.3, is installed in the connection surfaces of the active link. Several IR emitter and receivers are also installed on the connection surfaces to be used as target markers during the connection. They are used to detect the position of the connection surfaces with respect to each other. When the passive and active connection surfaces are close enough for the connection, a motor activates the retractable hook mechanism on the active link.

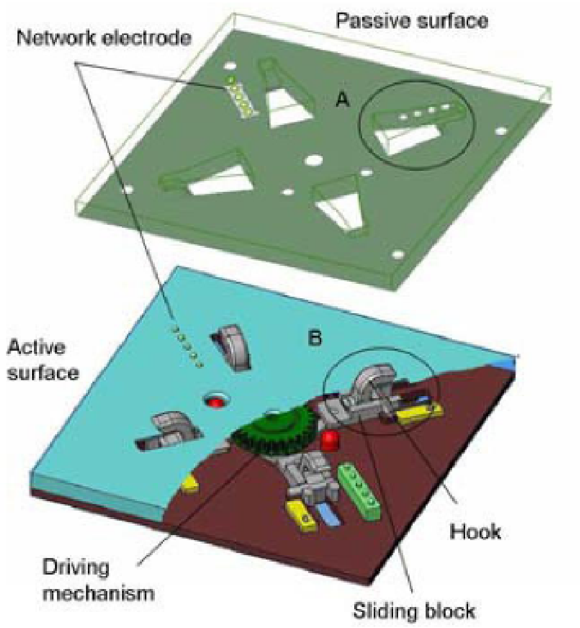


Figure 2.3: Connection mechanism of MTRAN III [31]

Then four hooks move into the slots on the passive connection surface and the connection is achieved. For disconnection of the two modules, driving the retractable hook mechanism in the reverse direction is enough. The tolerances for the MTRAN III to achieve connection is $5mm$ along the connection surface direction, $2mm$ perpendicular distance between modules and $10°$ in rotation [31].

2.1.3 Electrical Units

There are four processing units, one master and three slaves, in each MTRAN III module. Renesas SH II is used as the master processor and Renesas H8 is used as the slave processors. CAN bus connects all the master processors of the connected modules for communication, motion synchronization and cooperation. Communication of the modules is achieved through wireless connection via bluetooth. Each module is powered by means of lithium-polymer batteries [32].

2.2 Motion of the MTRAN System

Various simulations and experiments have been performed in order to verify the self-reconfiguration and system locomotion of the MTRAN system. These studies consider mainly three topics, namely, metamorphosis of large structures, the reconfiguration between different locomotion gaits and the locomotion of the system with different gaits [34].

2.2.1 Reconfiguration

Reconfiguration of modular robots can be considered in two different cases. One of them is the reconfiguration of small systems with a relatively low number of modules. The reconfiguration sequences of such systems can be found even manually, following a trial and error procedure. The other case is the reconfiguration of large systems composed of many modules. For this case, it is very difficult and time consuming to find the reconfiguration sequence of the overall system. Therefore the problem is divided into subproblems.

Kurokawa et al. [34] developed three regular structure families. These building blocks can change their position and/or orientation by using other modules. The regularity of these structures is also maintained during the reconfiguration sequence. The first type of structure is based on a parallel linear structure with additional two modules, called converters, to provide the conversion of motion direction. This type of a structure can generate caterpillar and crawler motions. The second type of structure is composed of four modules and two additional converter modules.

Straight motion and orthogonal turns are possible with such structures. The third type of structure is a planar structure having four modules connected to form either a square or a cross shape. The simulation snapshots of these structural types are shown in Figure 2.4.

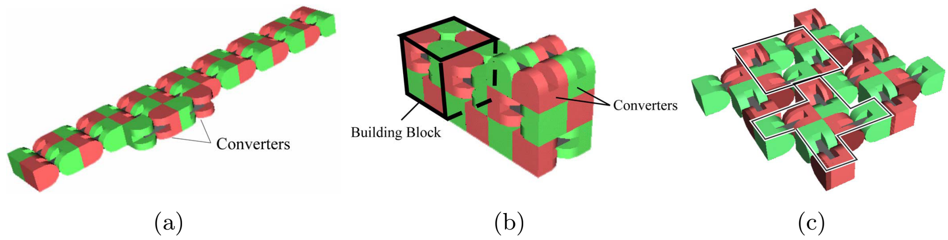


Figure 2.4: Different types of reconfiguration structures of MTRAN [34] : (a) Type -I (b) Type -II (c) Type -III

The reconfiguration of modules between different locomotion gaits are also studied. The sequences necessary for the reconfiguration of such systems are generated systematically using the structure types. In Figure 2.5, the reconfiguration from a four legged walker to a caterpillar is shown.

2.2.2 Locomotion

Locomotion of the MTRAN system is studied by considering several specially designed locomotion gaits. These gaits include four and six legged walkers, caterpillar configurations. Some of the locomotion gaits are shown in Figure 2.6.
The locomotion gaits shown in the figure are experimentally realized and the moving speeds of these gaits are measured. The fastest gait is the four legged walker having a speed of $20cm/s$ whereas the caterpillar and mini four legged walker gaits are slower than the other gaits (with a speed of $6cm/s$) [28].

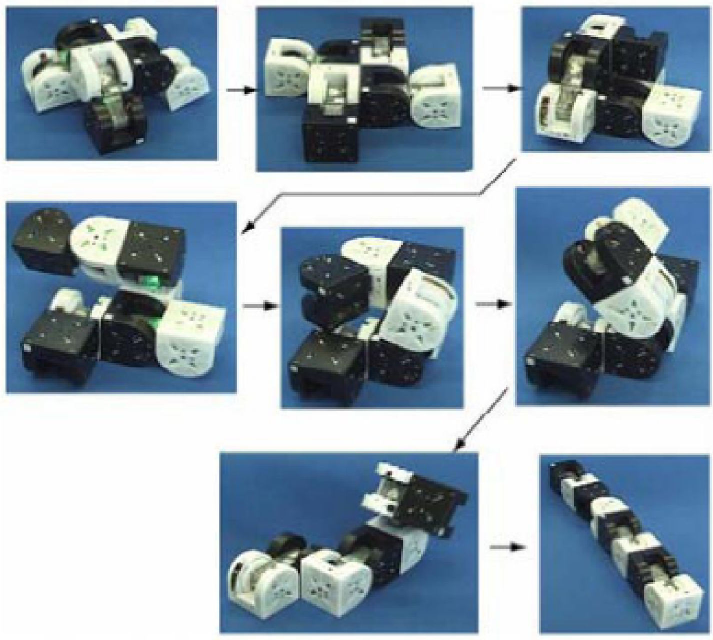
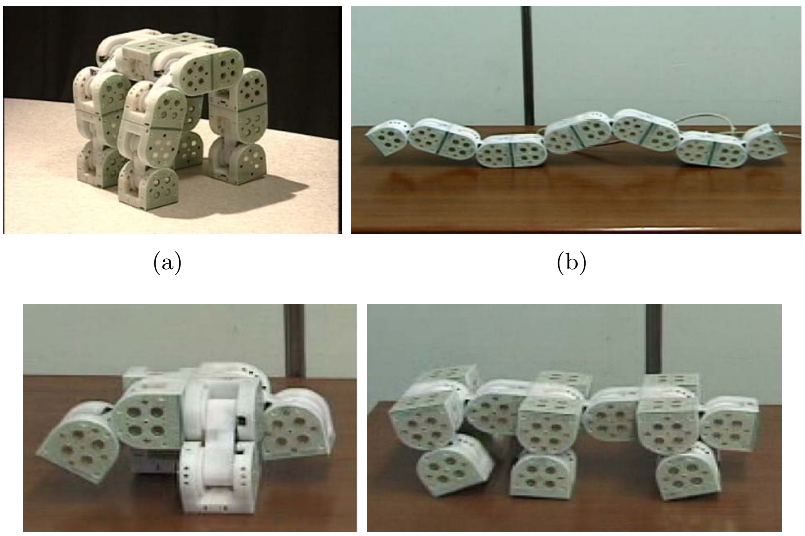


Figure 2.5: Reconfiguration of MTRAN III module from a walker to a caterpillar [31]



(a)

(b)

(c)

(d)

Figure 2.6: Different locomotion gaits of MTRAN [34] : (a) Four legged walker, (b) Caterpillar, (c) Six legged walker, (d) Mini four legged walker

CHAPTER 3

KINEMATIC ANALYSIS

In this chapter, the kinematic analysis of a modular robotic system consisting of n modules is explained. The kinematic analysis is performed to find the position, velocity and acceleration of all links for known values of joint variables. The procedure is discussed considering the structure of the MTRAN system. However, this method can be applied to other modular robots in the literature by adjusting the parameters (depending on the module geometry) and the joint types (according to the structure of the modular robot).

The kinematic analysis of a modular robot containing n modules is more efficiently performed via a recursive formulation for the position, velocity and acceleration analysis of the system. In this study, all terms in recursive formulation are represented in the global reference frame unless it is stated otherwise. Therefore, there is no need for transforming vectorial expressions between the reference frames in further calculations.

Evaluation order of the modules plays an important role in the execution time of the kinematic analysis. Therefore, a suitable evaluation order should be determined. In this study, the calculation starts with the module, on which the center of mass of the system in its fully open position lies. If it is at the connection of two modules, the module with smaller identifying number is evaluated first. Then, the modules connected to starting module are evaluated. This procedure is followed until all necessary calculations for all of the modules are performed. The evaluation order of a modular robot results in different evaluation directions within the module. For the MTRAN system, the direction is either from passive link to active link or from active link to passive link. The direction from passive link to active link indicates that the calculations for this module is performed by firstly calculating the unknown kinematic parameters of the active link using the kinematic parameters of the passive link of the previously evaluated module. Then the kinematic analysis of the middle and passive links is performed by means of the results of the kinematic analysis of the active link. The directions are shown in Figure 3.1. In the recursive formulation of the kinematic analysis, both cases are considered separately.

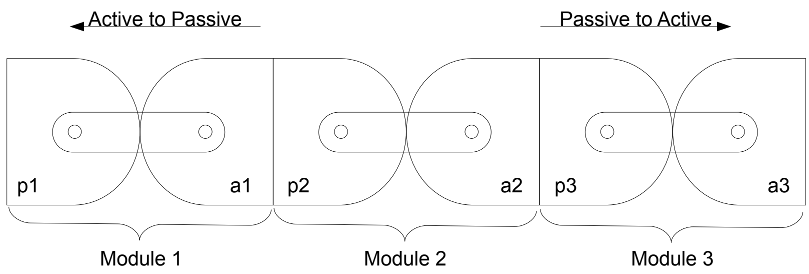


Figure 3.1: Evaluation Directions for the MTRAN system

For the analysis of the system, the center of masses of the passive and active links are assumed to be on the rotation axes of the joints. In addition, the center of mass of the middle link is assumed to be at the geometric center of the link.

Note that, the notation and the recursive formulation of the kinematic analysis are developed by examining the notation and recursive formulation in [35].

3.1 Notation

3.1.1 Body Fixed Reference Frames

The body fixed reference frames of the passive, middle and active links are shown in Figure 3.2. In this figure,

- $Fp_i(Op_i)$ is the body fixed reference frame attached to the passive link of the i^{th} module with origin Op_i coincident with the center of mass of the passive link
- $\vec{x}_{pi}, \vec{y}_{pi}, \vec{z}_{pi}$ are the orthogonal unit vectors of $Fp_i(Op_i)$
- $Fl_i(Ol_i)$ is the body fixed reference frame attached to the middle link of the i^{th} module with origin Ol_i coincident with the center of mass of the middle link
- $\vec{x}_{li}, \vec{y}_{li}, \vec{z}_{li}$ are the orthogonal unit vectors of $Fl_i(Ol_i)$

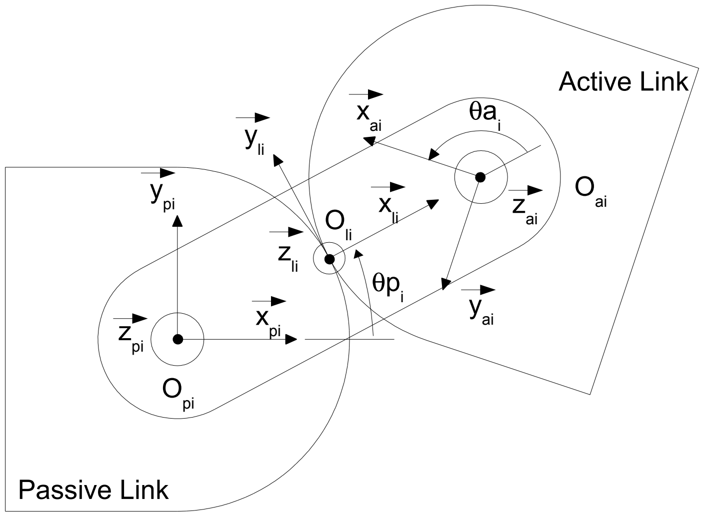


Figure 3.2: Body fixed reference frames of passive, middle and active links

- $Fa_i(Oa_i)$ is the body fixed reference frame attached to the active link of the i^{th} module with origin Oa_i coincident with the center of mass of the active link

- $\vec{x}_{ai}, \vec{y}_{ai}, \vec{z}_{ai}$ are the orthogonal unit vectors of $Fa_i(Oa_i)$

The first orthogonal unit vectors of the body fixed reference frames on the passive and active links are always directed to the circular part of the link and they are parallel to the edges of the link. Furthermore, the first orthogonal unit vector of the body fixed reference frame of the link is directed from the passive link to the active link. Besides, the third orthogonal unit vectors of passive, middle and active links are always parallel to each other with the same direction. Using the aforementioned conventions for positioning the body fixed reference frames on the links is extremely beneficial while analyzing large systems with many modules.

3.1.2 Definition of Joint Variables

Each MTRAN module has two rotary actuators. As a result, two variables are enough to define the relative position of the passive, middle and active links with respect to each other. Let θp_i be the joint variable of the revolute joint which connects passive and middle links of the i^{th} module. θp_i is measured from $\vec{x_{pi}}$ to $\vec{x_{li}}$ around $\vec{z_{pi}}$ in a right hand sense. Similarly, θa_i can be defined as the joint variable of the revolute joint between middle and active links of the i^{th} module. It is measured from $\vec{x_{li}}$ to $\vec{x_{ai}}$ around $\vec{z_{li}}$ in a right hand sense. The $\pm 90^\circ$ limitation of the actuators for MTRAN module limits the values of θp_i and θa_i. According to the definition of these angles, the limitations can be written as

$$-\frac{\pi}{2} \leq \theta p_i \leq \frac{\pi}{2}, \quad \frac{\pi}{2} \leq \theta a_i \leq \frac{3\pi}{2} \tag{3.1}$$

where θp_i and θa_i are in radians.

3.2 Position Analysis

Once the joint variables, θp_i and θa_i's, are specified, the relative positions of all links of the modular robot are known (with respect to any body fixed reference frame). In the position analysis, however, the positions of all links are needed with respect to an inertial, unmoving, reference frame. Therefore, besides the joint variables, six more parameters are needed in order to determine the position and orientation of a convenient body fixed reference frame with respect to the inertial frame. These six parameters may be taken to be the position (x, y and z coordinates) of a fixed point in the system, preferably the center of mass, and the orientation of the system ($1-2-3$ Euler's rotational sequence angles) with respect to the global reference frame. Once these parameters are specified, the position analysis of the system can be performed using the joint variables. The aforementioned six parameters are defined using an inertial reference frame and a center of mass reference frame. These reference frames are shown in Figure 3.3 for a three-link system. In this figure,

- $F_0(O_0)$ is the global reference frame with origin O_0
- $\vec{x}_0, \vec{y}_0$ and $\vec{z}_0$ are the orthogonal unit vectors of $F_0(O_0)$
- O_{cm} is the center of mass of the system at its fully open position, i.e., when all θp_i's are zero and all θa_i's are 180°. Note that O_{cm} is a fixed point on the modular robot.
- $F_{cm}(O_{cm})$ is the body fixed reference frame with origin O_{cm}
- $\vec{x}_{cm}, \vec{y}_{cm}$ and $\vec{z}_{cm}$ are the orthogonal unit vectors of $F_{cm}(O_{cm})$

$F_{cm}(O_{cm})$ reference frame has the same unit vectors as the body fixed reference frame attached to the link, on which O_{cm} lies. But the origin of the body fixed reference frame is shifted to O_{cm}. The six parameters that define the position and

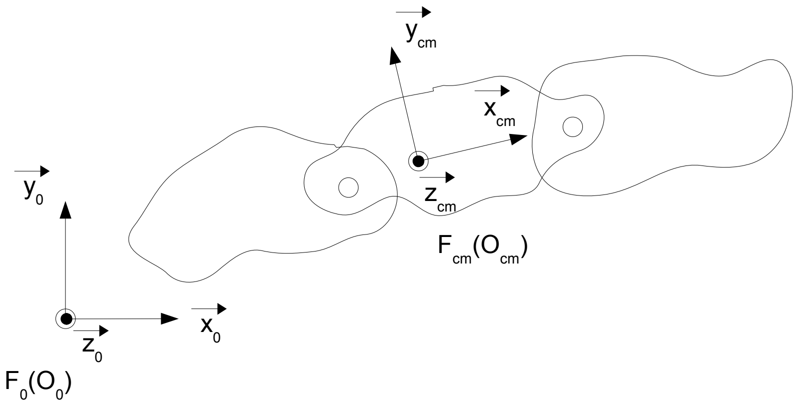


Figure 3.3: Global and center of mass reference frames of a three-link system

orientation of the body fixed $F_{cm}(O_{cm})$ frame with respect to the global reference frame $F_0(O_0)$ are selected to be $x_s, y_s, z_s, \gamma_s^1, \gamma_s^2$ and γ_s^3 where

- x_s, y_s and z_s are the x, y and z coordinates of O_{cm} in global reference frame,
- γ_s^1, γ_s^2 and γ_s^3 are the rotation angles of $1-2-3$ Euler's rotational sequence, which rotate $F_0(O_0)$ to $F_{cm}(O_{cm})$.

3.2.1 Center of Mass Calculation

The center of mass location of a dynamic system consisting of several bodies is not fixed relative to the links in the system. It always depends on the position of the links with respect to each other. In order to avoid confusion during the calculations, the center of mass of the system (O_{cm}) at the fully open position is taken for all calculations. In Figure 3.4, the fully open position of an MTRAN system with three modules and its center of mass are shown.

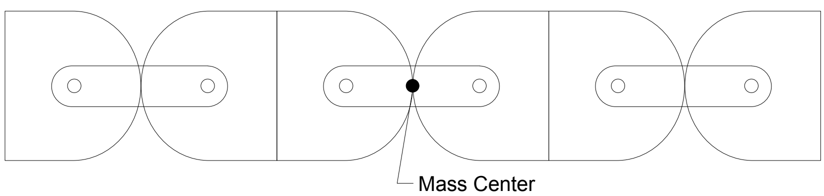


Figure 3.4: Three module MTRAN system in fully open position

The x, y and z coordinates of the center of mass for a k link system can be found by

$$x_s = \frac{\sum_{i=1}^{k} m_i x_i}{\sum_{i=1}^{k} m_i}, \quad y_s = \frac{\sum_{i=1}^{k} m_i y_i}{\sum_{i=1}^{k} m_i}, \quad z_s = \frac{\sum_{i=1}^{k} m_i z_i}{\sum_{i=1}^{k} m_i} \tag{3.2}$$

where x_i, y_i and z_i are the center of mass coordinates and m_i is the mass of the i^{th} link. Similarly, the x, y and z coordinates of the center of mass of an n module MTRAN system can be written as

$$\begin{aligned} x_s &= \frac{\sum_{i=1}^{n} (m_p x_{pi} + m_l x_{li} + m_a x_{ai})}{n\,(m_p + m_l + m_a)} \\ y_s &= \frac{\sum_{i=1}^{n} (m_p y_{pi} + m_l y_{li} + m_a y_{ai})}{n\,(m_p + m_l + m_a)} \\ z_s &= \frac{\sum_{i=1}^{n} (m_p z_{pi} + m_l z_{li} + m_a z_{ai})}{n\,(m_p + m_l + m_a)} \end{aligned} \tag{3.3}$$

where, m_p, m_l and m_a are the masses of the passive, middle and active links respectively.

3.2.2 Rodrigues' Formula

For the kinematic analysis of the system, the rotational transformation matrices of all body fixed reference frames with respect to the global reference frame are required. These rotational transformation matrices can be calculated by means of the Rodrigues' formula. The matrix form of this formula can be written as

$$R\,(\bar{n}, \theta) = \hat{I} + \tilde{n} \sin\theta + \tilde{n}\tilde{n}\,(1 - \cos\theta) \tag{3.4}$$

where $R\,(\bar{n}, \theta)$ represents the transformation matrix between two reference frames, due to the rotation of one reference frame with respect to the other (around $\vec{n}$) by an angle of θ, $\bar{n}$ is the column vector representation of $\vec{n}$, θ denotes the rotation angle around vector $\vec{n}$ and $\hat{I}$ is the 3×3 identity matrix. In addition, $\tilde{n}$ is the cross product matrix form of the column matrix representation of $\vec{n}$ in one of the reference frames, for which the transformation matrix is calculated. Note that, for column matrix $\bar{n}$, cross product matrix form $\tilde{n}$ is defined as

$$\bar{n} = \begin{bmatrix} n_1 \\ n_2 \\ n_3 \end{bmatrix} \quad \Rightarrow \quad \tilde{n} = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix} \tag{3.5}$$

The Rodrigues' formula is used to transform the column matrix representation of a vector from its corresponding body fixed reference frame to global reference frame.

3.2.3 Recursive Position Analysis Formulation

The formulation of the recursive position analysis of an n module MTRAN system will be presented next. Throughout this section, i represents the identifying number of the module currently being evaluated and j represents the identifying number of the module connected to the i^{th} module, which has been evaluated in the previous steps of the analysis. For the first module to be evaluated, the value of j is taken to be zero and the necessary recursive expression values for the first module are obtained via initialization. For the first module to be evaluated $j = 0$. Note that, the $j = 0$'th module rotational transformation matrix is the same as the rotational transformation matrix of the reference frame $F_{cm}(O_{cm})$ The calculation is repeated for all modules in the system according to the evaluation order.

The formulation of the recursive position analysis is discussed in three parts. Firstly, the initialization of the terms for the first calculation is given. Then, the expressions used for the evaluation process from passive link to active link are presented. Finally, the expressions used for the evaluation process from active link to passive link are given.

Definitions of $\bar{x}, \bar{y}$ and $\bar{z}$ used in the formulations are given below.

$$\bar{x} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \bar{y} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \bar{z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.6}$$

3.2.3.1 Rotational Transformation Matrices

- $\hat{C}_{pi}, \hat{C}_{li}$ and $\hat{C}_{ai}$ are the rotational transformation matrices of the passive, middle and active links of the i^{th} module respectively. They are used to transform a vector from the body fixed reference frame of the corresponding part of the i^{th} module to the global reference frame. For instance, if $\bar{v}_{pi}$ and $\bar{v}_0$ denote the the column matrix representation of a vector $\vec{v}$ in the reference frame of the passive part of the i^{th} module and global reference frame respectively, then $\bar{v}_0 = \hat{C}_{pi}\,\bar{v}_{pi}$

- $\hat{C}_{ij}$ is the transformation matrix used to transform vectors from the reference frames of the j^{th} module to the reference frame of the i^{th} module.

- ψ_{ij}^1, ψ_{ij}^2 and ψ_{ij}^3 are rotational angles of the Euler's rotational sequence for the $1-2-3$ rotational sequence from the body fixed reference frame of the j^{th} module to the body fixed reference frame of the i^{th} module. The values of these parameters depend on the connection surface arrangements. For example, if the connected passive and active links have parallel x axes, then ψ_{ij}^1 and ψ_{ij}^2 are equal to zero and ψ_{ij}^3 is equal to π.

The recursive rotational transformation matrices can be efficiently obtained, recursively, via the equations presented below.

- Initialization :

$$\begin{aligned} \hat{C}_{a0} &= \hat{C}_{p0} = \hat{I} \\ \hat{C}_{i0} &= \hat{C}_{cm} \end{aligned} \tag{3.7}$$

  Here, $\hat{C}_{cm}$ is the rotational transformation matrix between the global reference frame, $F_0(O_0)$, and the body fixed reference frame attached to the center of mass of the system at its fully open position, $F_{cm}(O_{cm})$. It can be expressed as

$$\hat{C}_{cm} = R\left(\bar{x}, \gamma_s^1\right) \cdot R\left(\bar{y}, \gamma_s^2\right) \cdot R\left(\bar{z}, \gamma_s^3\right) \tag{3.8}$$

- Evaluation from passive link to active link:

$$\begin{aligned} \hat{C}_{ij} &= R\left(\bar{x}, \psi_{ij}^1\right) \cdot R\left(\bar{y}, \psi_{ij}^2\right) \cdot R\left(\bar{z}, \psi_{ij}^3\right) \\ \hat{C}_{pi} &= \hat{C}_{aj} \cdot \hat{C}_{ij} \\ \hat{C}_{li} &= \hat{C}_{pi} \cdot R\left(\bar{z}, \theta p_i\right) \\ \hat{C}_{ai} &= \hat{C}_{li} \cdot R\left(\bar{z}, \theta a_i\right) \end{aligned} \tag{3.9}$$

- Evaluation from active link to passive link:

$$\begin{aligned} \hat{C}_{ij} &= R\left(\bar{x}, \psi_{ij}^1\right) \cdot R\left(\bar{y}, \psi_{ij}^2\right) \cdot R\left(\bar{z}, \psi_{ij}^3\right) \\ \hat{C}_{ai} &= \hat{C}_{pj} \cdot \hat{C}_{ij} \\ \hat{C}_{li} &= \hat{C}_{ai} \cdot R\left(\bar{z}, -\theta p_i\right) \\ \hat{C}_{pi} &= \hat{C}_{li} \cdot R\left(\bar{z}, -\theta a_i\right) \end{aligned} \tag{3.10}$$

3.2.3.2 Orientation of the Links

In three-dimensional space, three angles are used to indicate the orientation of each link with respect to the global reference frame. A column matrix is defined to represent these three orientation. Let $\bar{\phi}_{pi}$ and $\bar{\phi}_{ai}$ be the column matrices, whose components are the orientation angles of the passive and active links of the i^{th} module, given by

$$\bar{\phi}_{pi} = \begin{bmatrix} \phi_{pi}^x \\ \phi_{pi}^y \\ \phi_{pi}^z \end{bmatrix}, \quad \bar{\phi}_{ai} = \begin{bmatrix} \phi_{ai}^x \\ \phi_{ai}^y \\ \phi_{ai}^z \end{bmatrix} \tag{3.11}$$

where,

- ϕ_{pi}^x and ϕ_{ai}^x are the angles measured from $\vec{y}_0$ to the projections of the $\vec{y}_{pi}$ and $\vec{y}_{ai}$ on the y-z plane of the global reference frame, respectively around $\vec{x}_0$ in a right hand sense.
- ϕ_{pi}^y and ϕ_{ai}^y are the angles measured from $\vec{z}_0$ to the projections of the $\vec{z}_{pi}$ and $\vec{z}_{ai}$ on the x-z plane of the global reference frame, respectively around $\vec{y}_0$ in a right hand sense.
- ϕ_{pi}^z and ϕ_{ai}^z are the angles measured from $\vec{x}_0$ to the projections of the $\vec{x}_{pi}$ and $\vec{x}_{ai}$ on the x-y plane of the global reference frame, respectively around $\vec{z}_0$ in a right hand sense.

The orientation angles are calculated by means of dot and cross product operations between the corresponding vectors expressed in the global reference frame. The details are given in Appendix A.

3.2.3.3 Center of Mass Position of the Links

After calculating the rotational transformation matrices, the position vectors of the mass centers of the links are found recursively by using the formulation shown below. In the following formulation, $\bar{r}_{pi}, \bar{r}_{li}$ and $\bar{r}_{ai}$ are the column matrix representations of the position vectors of the mass centers of the passive, middle and active links of the i^{th} module represented in global reference frame. The initial value of the recursion has three parts. They are the global coordinates of the center of mass in the fully open configuration, the coordinate shifting to the nearest center of mass of the passive or active link and the recursion starting adjustment for the first term.

- Initialization :

$$\bar{r}_{p0} = \bar{r}_{a0} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} + \left(\frac{d}{2} \hat{C}_{cm} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

- Evaluation from passive link to active link:

$$\begin{aligned} \bar{r}_{pi} &= \bar{r}_{aj} - d\, \hat{C}_{aj} \cdot \bar{x} \\ \bar{r}_{li} &= \bar{r}_{pi} + \frac{d}{2}\, \hat{C}_{li} \cdot \bar{x} \\ \bar{r}_{ai} &= \bar{r}_{li} + \frac{d}{2}\, \hat{C}_{li} \cdot \bar{x} \end{aligned} \tag{3.13}$$

- Evaluation from active link to passive link:

$$\begin{aligned} \bar{r}_{ai} &= \bar{r}_{pj} - d\, \hat{C}_{pj} \cdot \bar{x} \\ \bar{r}_{li} &= \bar{r}_{pi} - \frac{d}{2}\, \hat{C}_{li} \cdot \bar{x} \\ \bar{r}_{pi} &= \bar{r}_{li} - \frac{d}{2}\, \hat{C}_{li} \cdot \bar{x} \end{aligned} \tag{3.14}$$

3.3 Velocity and Acceleration Analysis

In order to perform dynamic force analysis and check the feasibility of a given motion, the velocity and acceleration of all links are required. To find the velocity and acceleration of the links, firstly, the position analysis must be completed. In this section, the formulation used to determine the velocity and acceleration expressions of the links are explained.

Calculations of the velocity and acceleration terms require taking the time derivatives of the vectorial terms. These derivatives have to be evaluated in the global reference frame. Since all vectorial terms are represented in this reference frame, differentiating each component of the vector is sufficient to find its derivative with respect to time.

In order to differentiate the position expressions, the variables that depend on time should be identified firstly. The main terms, which depend on time are the joint variables θ_{pi} and θ_{ai}. Furthermore, the reference frame $F_{cm}(O_{cm})$ is a moving one. Hence, the six parameters, namely, $x_s, ys, z_s, \gamma_s^1, \gamma_s^2$ and γ_s^3 are also time dependent.

3.3.1 Recursive Velocity Analysis

3.3.1.1 Angular Velocities of the Links

In the following equations, $\bar{\omega}_{pi}, \bar{\omega}_{li}$ and $\bar{\omega}_{ai}$ are the column matrix representations of the angular velocity vectors of the passive, middle and active links of the i^{th} module expressed in the global reference frame.

- Initialization :

$$\tilde{\omega}_{p0} = \tilde{\omega}_{a0} = \dot{\hat{C}}_{cm}\,\hat{C}_{cm}^{t} \tag{3.15}$$

  Then,

$$\bar{\omega}_{p0} = \bar{\omega}_{a0} = col\left[\dot{\hat{C}}_{cm}\,\hat{C}_{cm}^{t}\right] \tag{3.16}$$

  Here, $col[\tilde{r}]$ is the column matrix form of the $\tilde{r}$ matrix, i.e., $\bar{r} = col[\tilde{r}]$. Note that, superscript t denotes matrix transpose.

- Evaluation from passive link to active link:

$$\begin{aligned} \bar{\omega}_{pi} &= \bar{\omega}_{aj} \\ \bar{\omega}_{li} &= \bar{\omega}_{pi} + \dot{\theta}p_i\,\hat{C}_{pi}\cdot\bar{z} \\ \bar{\omega}_{ai} &= \bar{\omega}_{li} + \dot{\theta}a_i\,\hat{C}_{li}\cdot\bar{z} \end{aligned} \tag{3.17}$$

- Evaluation from active link to passive link:

$$\begin{aligned} \bar{\omega}_{ai} &= \bar{\omega}_{pj} \\ \bar{\omega}_{li} &= \bar{\omega}_{ai} + \dot{\theta}a_i\,\hat{C}_{ai}\cdot\bar{z} \\ \bar{\omega}_{pi} &= \bar{\omega}_{li} + \dot{\theta}p_i\,\hat{C}_{li}\cdot\bar{z} \end{aligned} \tag{3.18}$$

3.3.1.2 Linear Center of Mass Velocities of the Links

Linear center of mass velocity of a link is the derivative of the position vector of its center of mass with respect to time. Therefore, the linear velocities of the center of mass of the links are given by

$$\begin{aligned} \bar{v}_{pi} &= \frac{d}{dt}\bar{r}_{pi} \\ \bar{v}_{li} &= \frac{d}{dt}\bar{r}_{li} \\ \bar{v}_{ai} &= \frac{d}{dt}\bar{r}_{ai} \end{aligned} \tag{3.19}$$

where, $\bar{v}_{pi}, \bar{v}_{li}$ and $\bar{v}_{ai}$ are column matrix form of the linear velocity vectors of the passive, middle and active link of the i^{th} module represented in the global reference frame.

3.3.2 Recursive Acceleration Analysis

3.3.2.1 Angular Accelerations of the Links

Angular acceleration of a link is the derivative of the angular velocity vector of the link with respect to time. For the i^{th} module of the MTRAN system, the column matrix form of the angular accelerations can be found as

$$\begin{aligned} \bar{\alpha}_{pi} &= \frac{d}{dt}\bar{\omega}_{pi} \\ \bar{\alpha}_{li} &= \frac{d}{dt}\bar{\omega}_{li} \\ \bar{\alpha}_{ai} &= \frac{d}{dt}\bar{\omega}_{ai} \end{aligned} \tag{3.20}$$

In these expressions, $\bar{\alpha}_{pi}, \bar{\alpha}_{li}$ and $\bar{\alpha}_{ai}$ are the column matrix form of the angular accelerations of the passive, middle and active links of the i^{th} module respectively. They are expressed in the global reference frame.

3.3.2.2 Linear Center of Mass Accelerations of the Links

Proceeding similar to the linear velocity calculations, the linear center of mass acceleration of a link can be found by differentiating the linear center of mass velocity with respect to time. Hence,

$$\begin{aligned} \bar{a}_{pi} &= \frac{d}{dt}\bar{v}_{pi} \\ \bar{a}_{li} &= \frac{d}{dt}\bar{v}_{li} \\ \bar{a}_{ai} &= \frac{d}{dt}\bar{v}_{ai} \end{aligned} \tag{3.21}$$

where, $\bar{a}_{pi}, \bar{a}_{li}$ and $\bar{a}_{ai}$ are the column matrix representations of the linear center of mass accelerations of the passive, middle and active links of the i^{th} module respectively. These column matrices are expressed in the global reference frame.

3.4 Two-Dimensional Analysis

In the previous sections, full (i.e., three-dimensional) position, velocity and acceleration analyses are explained. Clearly, it is possible to use this analysis to perform a two-dimensional kinematic analysis. This can be achieved by simply equating the terms related to the unused third dimension to zero. For a two dimensional analysis of the MTRAN system on the x-y plane, the following terms become zero.

- z_s, the z coordinate of the center of mass of the system
- γ_s^1 and γ_s^2, the first two rotation angles of the $1-2-3$ Euler's rotational sequence
- ψ_{ij}^1 and ψ_{ij}^2, the rotation angles between the connected links of the i^{th} and j^{th} modules around x and y axes respectively

Hence, the rotational transformation matrix between the global reference frame and the center of mass of the system reference frame becomes $\hat{C}_{cm} = R(\bar{z}, \gamma_s^3)$.

In addition to the terms listed above, ψ_{ij}^3, which is the rotation angle between the connected links of the i^{th} and j^{th} modules around z axis, becomes π for all connections. If the analyses is performed with these values, zero terms for the third component of the position, linear velocity and linear acceleration vectors for all links are obtained. Additionally, the first and second components of the angular vectorial expressions, such as the orientation angle, angular velocity and angular acceleration, become zero.

CHAPTER 4

DETERMINATION OF THE CONTACT POINTS

In this chapter, the approach used for the determination of the contact points of a modular robot is explained. Here, the contact points are defined to be those points on the modular robot which are in contact with the ground. The procedure is discussed by considering a planar case for simplicity. In the calculations, the y axis of the global reference frame is assumed to be coincident with the negative gravitational acceleration direction. Furthermore, the application of this procedure to an MTRAN system composed of n modules is illustrated.

The main objective of this chapter is to determine the so-called contact point related parameters of a modular robot when the position and orientation of the modular robot, moving in the x-y plane, is given. Here, the contact point related parameters refer to

- The minimum distance of the modular robot to the ground, d_{min}
- The number of contact points
- The coordinates of the contact points in the global reference frame.

The calculation of the contact point related parameters requires a complete position analysis of the system. Once the position analysis is performed, the following procedure is followed in order to calculate the necessary contact point parameters.

1. Determination of the critical points on the links of the modular robot
2. Determination of the minimum distance of the modular robot to the ground
3. Calculation of the global coordinates of the contact points

Each of the three steps above is discussed, in detail, in the following sections.

4.1 Critical Points

Critical points are those special points on the links of a modular robot, which are candidates for being contact points. A contact point must be one of the critical points. The locations of critical points strongly depend on the geometry of the links. In general, critical points will not lie on the concave parts of the links since there cannot be any contact between a point on a concave part of the robot and the ground, which is assumed to be a horizontal surface without any slope. In addition to the link geometry, the arrangement of the links plays an important role in the position of the critical points in the system. Some links can never touch the ground due to the physical structure of the system. Therefore, the critical points of such links can be neglected. In Figure 4.1, the critical points for different link shapes are shown. Triangular and rectangular links have three and four critical points respectively. However, the circular link only needs three critical points since a single critical point, P_3, represent all possible contact points on the circular part. Two critical points, which form a line segment that coincides with the boundary of a link, lead to a line contact. Links with a line contact has a line segment coincident with the ground at infinitely many points. Critical points are the end points of this line segment.

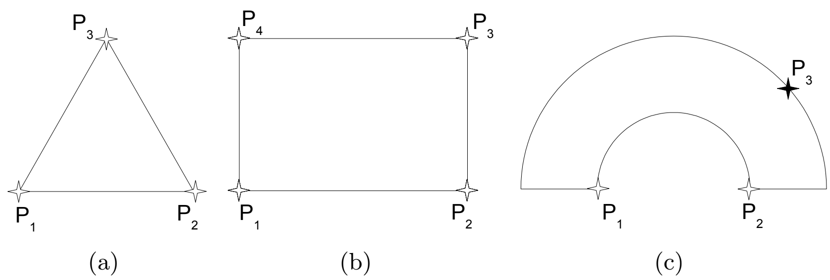
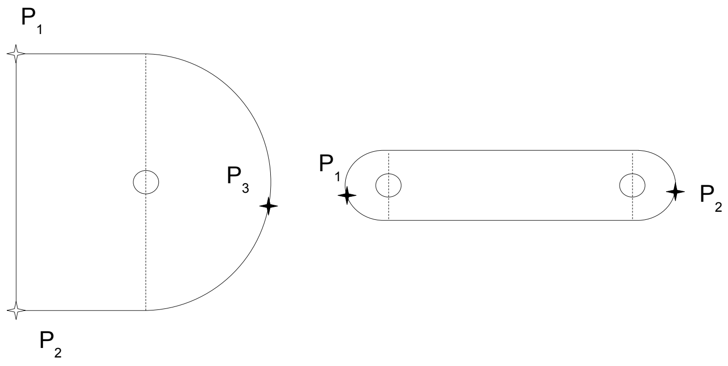


Figure 4.1: Critical points of links with different geometries (a) Triangular link with 3 critical points (b) Rectangular link with 4 critical points (c) Circular link with 3 critical points (P_3 represents any point on the semi circle)

Each module of the MTRAN system is composed of three links, namely, the passive, middle and active links. Passive and active links have the same shape, whereas the middle link has a different geometry. The critical points of these links are shown in Figure 4.2. The passive and active links have 3 critical points, whereas the middle link has 2 critical points. The summation of the critical points for a single MTRAN module is 8. However, the middle link cannot touch the ground due to the physical



(a) (b)

Figure 4.2: Critical points of MTRAN links (a) Passive/Active link with 3 critical points (P_3 represents any point on the semicircle) (b) Middle link with 2 critical points (P_1 and P_2 represent any point on the two semicircles)

limitations of the system. Hence, the number of critical points for a single MTRAN module reduces to 6.

4.2 Calculation of Minimum Distance to the Ground

Minimum distance of a modular robotic system to the ground, d_{min}, indicates the status of the system with respect to the ground. If the minimum distance of the system is less than zero, some part of the system is below the ground level, which is physically unrealizable for a system composed of rigid links. If the minimum distance is greater than zero, the system does not touch to the ground at any points. This case is also unfeasible unless the considered modular robot has the ability to fly. The last possible case is having zero distance to the ground, indicating that the system has contact with the ground at one or more points. This is the only practical case.

Consider the passive and active links with the critical points shown in Figure 4.3. In this figure,

- x^m_{pi}, y^m_{pi} are the x and y coordinates of the mass center of the passive part of the i^{th} module in the global reference frame,
- x^m_{ai}, y^m_{ai} are the x and y coordinates of the mass center of the active part of the i^{th} module in the global reference frame,

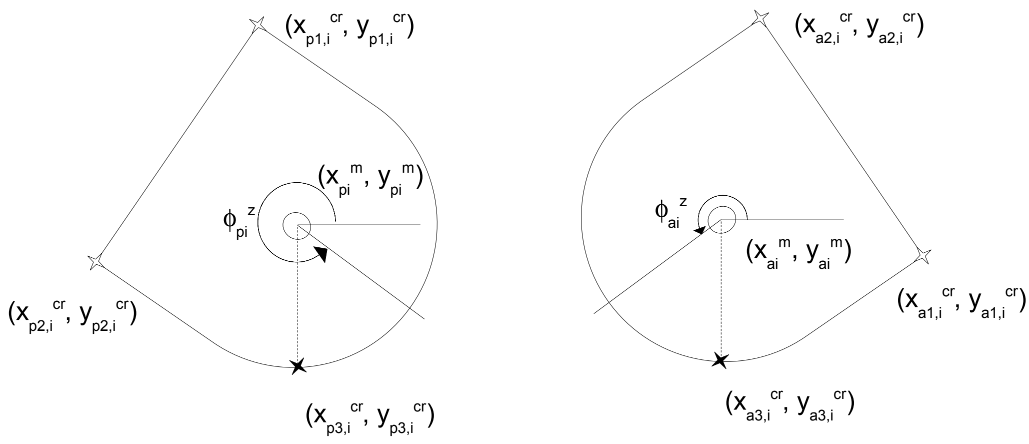


Figure 4.3: Contact point parameters of passive and active links of MTRAN module

- ϕ^z_{pi} is the yaw orientation angle of the passive part of the i^{th} module measured from $\vec{x}_0$ to $\vec{x_{pi}}$ around $\vec{z}_0$ in a right hand sense,
- ϕ^z_{ai} is the yaw orientation angle of the active part of the i^{th} module measured from $\vec{x}_0$ to $\vec{x_{ai}}$ around $\vec{z}_0$ in a right hand sense,
- $x^{cr}_{pj,i}, y^{cr}_{pj,i}$ are the x and y coordinates of the j^{th} critical point of passive link of the i^{th} module in the global reference frame,
- $x^{cr}_{aj,i}, y^{cr}_{aj,i}$ are the x and y coordinates of the j^{th} critical point of active link of the i^{th} module in the global reference frame,

It can be shown that the vertical distances of each critical point of the i^{th} passive link to the ground can be written as

$$\begin{aligned} y^{cr}_{p1,i} &= y^m_{pi} + \frac{d}{2}(\sin\left(\phi^z_{pi} + \pi/2\right) + \sin\left(\phi^z_{pi} + \pi\right)) \\ y^{cr}_{p2,i} &= y^{cr}_{p1,i} + d\sin\left(\phi^z_{pi} - \pi/2\right) \\ y^{cr}_{p3,i} &= y^m_{pi} - \frac{d}{2} \end{aligned} \tag{4.1}$$

Here, d is the unit length of the MTRAN module as shown in Figure 2.1(a). Repeating this calculation for the i^{th} active link, one obtains

$$
\begin{aligned}
y^{cr}_{a1,i} &= y^m_{ai} + \frac{d}{2}(\sin(\phi^z_{ai} + \pi/2) + \sin(\phi^z_{ai} + \pi)) \\
y^{cr}_{a2,i} &= y^{cr}_{a1,i} + d\sin(\phi^z_{ai} - \pi/2) \\
y^{cr}_{a3,i} &= y^m_{ai} - \frac{d}{2}
\end{aligned}
\tag{4.2}
$$

Note that the expressions for $y^{cr}_{p3,i}$ and $y^{cr}_{a3,i}$ will be correct iff the third critical point, on the semicircle, is actually a contact point.

Define, now,

$$
\bar{d}_{mod,i} = \begin{bmatrix} y^{cr}_{p1,i} \\ y^{cr}_{p2,i} \\ y^{cr}_{p3,i} \\ y^{cr}_{a1,i} \\ y^{cr}_{a2,i} \\ y^{cr}_{a3,i} \end{bmatrix}
\tag{4.3}
$$

where $\bar{d}_{mod,i}$ is a 6×1 column matrix, which stores the vertical distance of all critical points in the i^{th} module.

For an n module MTRAN system, Equations (4.1)-(4.3) are empoyed for each module, and the vertical distance of all critical points of the system are calculated and stored as in Equation (4.4). Here, $\bar{d}_{sys}$ is a $6n \times 1$ column matrix storing the vertical distance data of all critical points in the system.

$$
\bar{d}_{sys} = \begin{bmatrix} \bar{d}_{mod,1} \\ \bar{d}_{mod,2} \\ \cdots \\ \bar{d}_{mod,n} \end{bmatrix}
\tag{4.4}
$$

The smallest element of $\bar{d}_{sys}$ is the minimum distance of the system to the ground, i.e.,

$$
d_{min} = min_i \left[\bar{d}_{sys}(i)\right]
\tag{4.5}
$$

where $\bar{d}_{sys}(i)$ is the i^{th} element of $\bar{d}_{sys}$.

After calculating d_{min}, the status of the system with respect to the ground can be detected. As discussed before, d_{min} will be zero there is contact between the modular robot and the ground. However, d_{min} can be very close to zero due to the numerical round off errors in the calculations. Such cases should also be considered to correspond to a feasible contact. In order to distinguish these cases from other non-zero d_{min} cases, with values far away from zero, a tolerance, ϵ is used. The condition to have a feasible contact between the robot and the ground can be written as

$$-\epsilon \leq d_{min} \leq \epsilon \tag{4.6}$$

If this condition is satisfied, the system is assumed to touch the ground at one or more points. The coordinates of these contact points can be calculated using the $\bar{d}_{sys}$ data, which is discussed in the next section.

4.3 Calculation of Contact Point Coordinates

Once it is determined that the system has contact with the ground at one or more points, the locations of these points should be found using the $\bar{d}_{sys}$ data. The links, on which the contact points lie, can also be determined by checking the index of the found critical point distance in $\bar{d}_{sys}$.

The i^{th} element of $\bar{d}_{sys}$ is considered to be a contact point iff,

$$-\epsilon \leq \bar{d}_{sys}(i) \leq \epsilon \tag{4.7}$$

where ϵ is the user specified tolerance value used in the determination of d_{min}. Checking all elements of $\bar{d}_{sys}$ for this condition and recording the i indices that satisfy it, one obtains a set of index numbers, which is denoted by CI, indicating the position of the contact points in $\bar{d}_{sys}$. For each element of CI, the module number, the link type (passive or active) and the critical point number are required.

Let i denote an index number that is an element of CI. Using the definition of the d_{sys} vector, it follows that

$$N_{mod}(i) = \left\lceil \frac{i}{6} \right\rceil \tag{4.8}$$

$$N_{part}(i) = \begin{cases} 1 & if \quad 0 < mod(i,6) \leq 3 \\ 2 & if \quad mod(i,6) = 0 \vee mod(i,6) > 3 \end{cases} \tag{4.9}$$

$$N_{cp}(i) = \begin{cases} mod(i,3) & if \quad mod(i,3) \neq 0 \\ 3 & if \quad mod(i,3) = 0 \end{cases} \tag{4.10}$$

where $N_{mod}(i), N_{part}(i)$ and $N_{cp}(i)$ denote the module number, part number and critical point number corresponding to i. Here, part numbers 1 and 2 correspond to passive and active links respectively and $N_{mod}(i)$ is the nearest higher integer of the result of the division $i/6$. Once the module numbers, part numbers and critical point numbers of the contact points are determined, the x coordinates of of the contact points may be obtained via the following equations.

$$
\begin{aligned}
x^{cr}_{p1,i} &= x^{m}_{pi} + \frac{d}{2}\left(\cos\left(\phi^{z}_{pi} + \pi/2\right) + \cos\left(\phi^{z}_{pi} + \pi\right)\right) \\
x^{cr}_{p2,i} &= x^{cr}_{p1,i} + d\cos\left(\phi^{z}_{pi} - \pi/2\right) \\
x^{cr}_{p3,i} &= x^{m}_{pi} \\
x^{cr}_{a1,i} &= x^{m}_{ai} + \frac{d}{2}\left(\cos\left(\phi^{z}_{ai} + \pi/2\right) + \cos\left(\phi^{z}_{ai} + \pi\right)\right) \\
x^{cr}_{a2,i} &= x^{cr}_{a1,i} + d\cos\left(\phi^{z}_{ai} - \pi/2\right) \\
x^{cr}_{a3,i} &= x^{m}_{ai}
\end{aligned}
\tag{4.11}
$$

Note that the y coordinates of the contact points can be easily extracted from the d_{sys} vector.

CHAPTER 5

DYNAMIC ANALYSIS

In this chapter, the dynamic analysis of modular robotic systems is discussed. Dynamic analysis is performed after the position, velocity and acceleration analyses. In the dynamic analysis, the required actuator forces of the modules are calculated as a function of time for a specified motion of the system. The derivation procedure and the solution methods for the dynamic force and moment equations are illustrated by considering an n module MTRAN system moving in a two-dimensional space.

The x-y plane of the global reference frame is taken as the workspace with g being in the negative y direction. The approach used in this analysis can be expanded to a three dimensional workspace by considering the reaction forces along the z axis of the global reference frame and the moments about the x and y axes.

The forces and moments that act on an MTRAN system are:

- revolute joint reaction forces,
- actuator torques due to driving motors,
- gravitational forces,
- contact point forces,
- connection forces and moments (between the connected active and passive links)

The reaction forces are taken along the x and y axes of the global reference frame. The gravitational forces are assumed to act on the joints for the passive and active links since the center of masses of these links are on the joints. Contact point forces have normal and tangential components and act on the system at the contact points. The tangential force is due to Coulomb friction. Its direction is opposite to the tangential velocity direction at the contact point. Finally, connection forces consist of two reaction forces and one reaction moment applied to the connected active and passive links.

5.1 Dynamic Specifications of MTRAN Modules

In order to perform a dynamic analysis of the MTRAN system, the inertial parameters of the links in the system are required. However, all of the necessary information about the modules is not available. Therefore, the following assumptions are made in order to use approximate values for the unavailable data.

- The masses of the passive, middle and active links are equal and they are uniformly distributed on the link
- The body fixed reference frames of the links in the system are the principle reference frames

According to the assumptions, the masses of the links are taken as

$$m_p = m_a = m_l = 140g$$

In addition, the mass moment of inertias of the links (about the z axis of their body fixed reference frames) in an MTRAN module are calculated as

$$I_p = 8.774 \times 10^{-5} kg.m^2$$
$$I_a = 8.774 \times 10^{-5} kg.m^2$$
$$I_l = 8.580 \times 10^{-5} kg.m^2$$

Here, I_p, I_a and I_l are the mass moment of inertia of the passive, active and middle links around their center of masses. The details of the calculations to find these parameters are presented in Appendix A.

5.2 Free Body Diagrams

In this section, the forces and moments acting on an MTRAN system are described in detail using free body diagrams.

5.2.1 Reaction Forces, Actuator Torques and Gravitational Forces

Consider Figure 5.1. In this figure,

- F_{pi}^x and F_{pi}^y are the reaction forces acting on the joint of the passive link of the i^{th} module along the x and y axes of the global reference frame
- F_{ai}^x and F_{ai}^y are the reaction forces acting on the joint of the active link of the i^{th} module along the x and y axes of the global reference frame
- T_{pi} and T_{ai} are the actuator torques applied by the motor on the passive and active links of the i^{th} module
- g is the gravitational acceleration constant
- m_p, m_l and m_a are the masses of the passive, middle and active links respectively (which are assumed to be equal).

5.2.2 Contact Point Forces

Two forces are acting on the system at each contact point. These forces are the normal reaction force and the tangential friction force due to Coulomb friction. The contact point forces acting on the passive link of the i^{th} module are shown in Figure 5.2 for point and line contact cases. In this figure,

- x_{cj} and y_{cj} are the x and y coordinates of the j^{th} contact point in the global reference frame
- f_j is the Coulomb friction force (applied by the ground on the link) at the j^{th} contact point
- N_j is the normal reaction force (applied by the ground on the link) at the j^{th} contact point

5.2.3 Connection Forces and Moments

For every rigid connection between a passive and active link, there are two reaction forces along the x and y axes and a moment about the z axis of the global reference frame. The point of application of the forces are assumed to be at the mid point of the links as shown in Figure 5.3. The labels of the connection forces and moments are the same according to the module number of the passive link. In Figure 5.3,

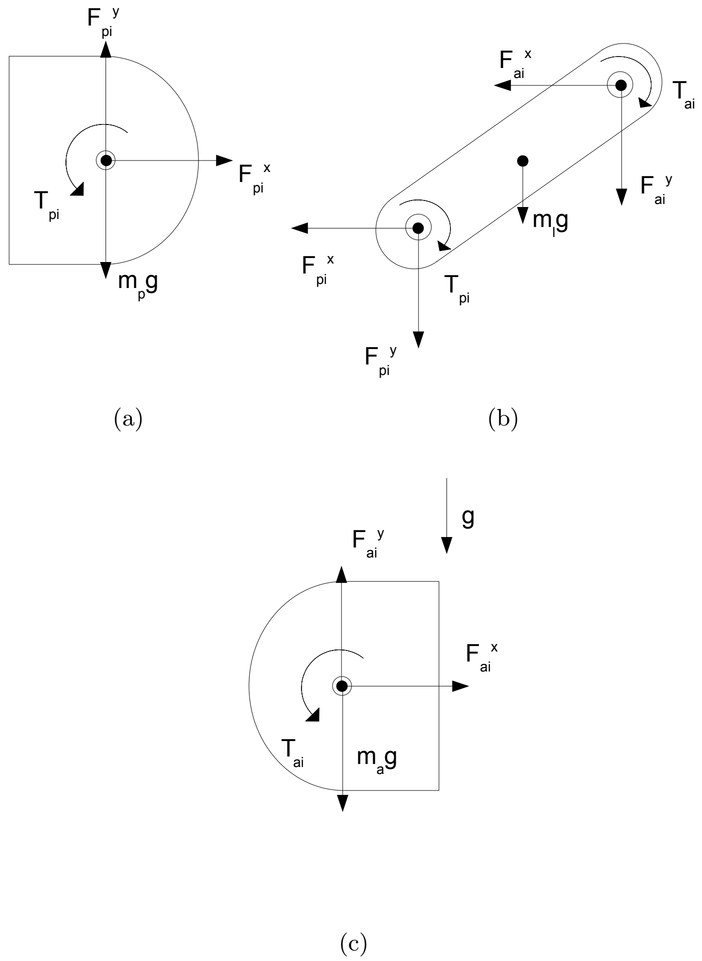


Figure 5.1: Free body diagrams for joint reaction forces and moments: (a) Passive link, (b) Middle link, (c) Active link

- F_{ci}^{x} and F_{ci}^{y} are the connection forces between the passive link of the i^{th} module and the active link of the j^{th} module along the x and y axes of the global reference frame. They are assumed to be applied in positive x and y directions to the active link of the j^{th} module.

- M_{ci} is the connection moment between the passive link of the i^{th} module and the active link of the j^{th} module about the z axis of the global reference frame. It is assumed to be applied in positive direction in a right hand sense to the active link of the j^{th} module.

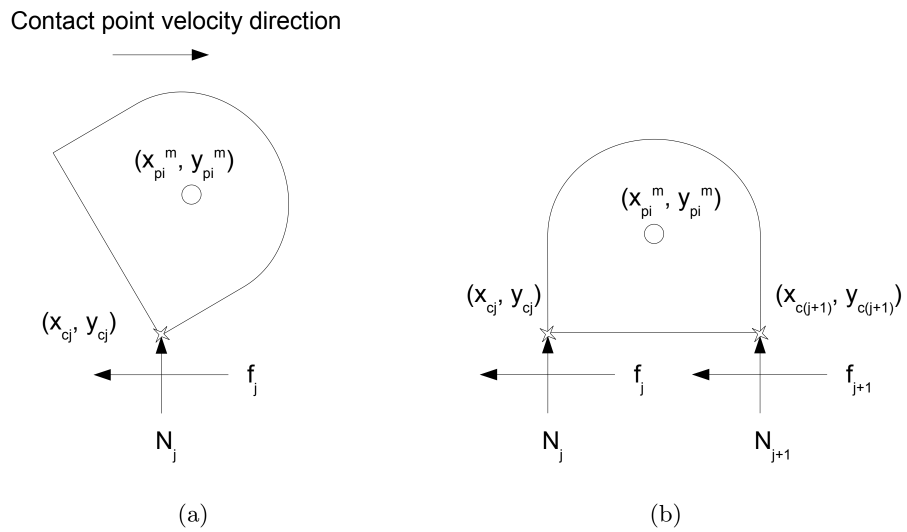


Figure 5.2: Reaction forces due to the contact of the link with the ground: (a) Point contact, (b) Line contact

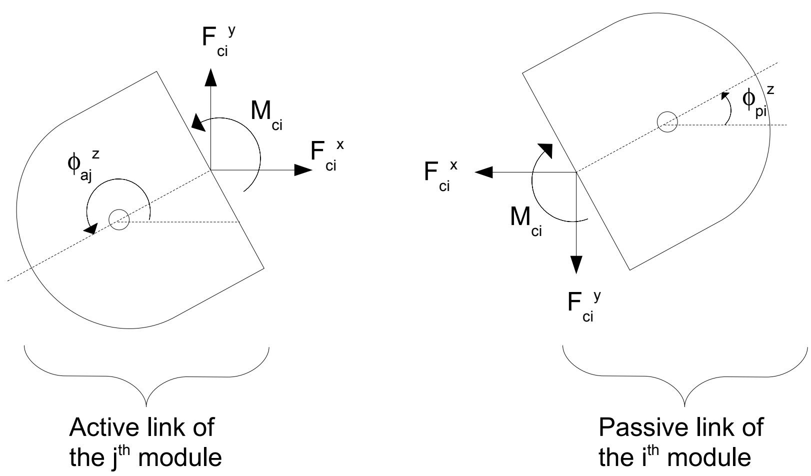


Figure 5.3: Reaction forces and moments due to the connection of two links

5.3 Newton-Euler Equations

The Newton-Euler equations, written for the passive, middle and active links of the i^{th} module of an MTRAN system moving in the global x-y plane, are given below.

- Passive link of the i^{th} module :

$$\Sigma Fx_{pi} = m_p a^x_{pi} \tag{5.1}$$
$$\Sigma Fy_{pi} = m_p a^y_{pi} \tag{5.2}$$
$$\Sigma Mo_{pi} = I_p \alpha^z_{pi} \tag{5.3}$$

- Middle link of the i^{th} module :

$$\Sigma Fx_{li} = m_l a^x_{li} \tag{5.4}$$
$$\Sigma Fy_{li} = m_l a^y_{li} \tag{5.5}$$
$$\Sigma Mo_{li} = I_l \alpha^z_{li} \tag{5.6}$$

- Active link of the i^{th} module :

$$\Sigma Fx_{ai} = m_a a^x_{ai} \tag{5.7}$$
$$\Sigma Fy_{ai} = m_a a^y_{ai} \tag{5.8}$$
$$\Sigma Mo_{ai} = I_a \alpha^z_{ai} \tag{5.9}$$

Considering the equations for the passive link, the following notation has been used.

- $\Sigma Fx_{pi}, \Sigma Fy_{pi}$ are the net force acting on the passive link of the i^{th} module along the global x and y directions, respectively,
- $\Sigma Mo_{pi}$is the net moment (around O_{pi}) acting on the passive link of the i^{th} module along the global z direction,
- a^x_{pi}, a^y_{pi} are the x and y components of the acceleration of the center of mass of the passive link of the i^{th} module,
- α^z_{pi} is the angular acceleration of the passive link of the i^{th} module (along the z axis of the global reference frame)

The notation employed for the middle and active links are similar to the one described above. In the following three sections, closed form expressions will be given for the left hand sides of Equations (5.1) - (5.9). In these expressions, the "+ =" operator is used. For instance,

$$A+=B \quad \Rightarrow A=A_{old}+B \tag{5.10}$$

where A_{old} is the value of A before the summation.

5.3.1 Contributions Due to Joint Reactions, Actuator Torques and Gravitational Forces

The following equations can be written for the i^{th} module by considering the free body diagrams in Figure 5.1. Note that initial values of ΣFx_{pi}, ΣFy_{pi}, ΣMo_{pi}, ΣFx_{li}, ΣFy_{li}, ΣMo_{li}, ΣFx_{ai}, ΣFy_{ai} and ΣMo_{ai} are all zero.

- Passive link of the i^{th} module:

$$\Sigma Fx_{pi} \mathrel{+}= F^x_{pi} \tag{5.11}$$

$$\Sigma Fy_{pi} \mathrel{+}= F^y_{pi} - m_p g \tag{5.12}$$

$$\Sigma Mo_{pi} \mathrel{+}= T_{pi} \tag{5.13}$$

- Middle link of the i^{th} module:

$$\Sigma Fx_{li} \mathrel{+}= -F^x_{pi} - F^x_{ai} \tag{5.14}$$

$$\Sigma Fy_{li} \mathrel{+}= -F^y_{pi} - F^y_{ai} - m_l g \tag{5.15}$$

$$\Sigma Mo_{li} \mathrel{+}= -T_{pi} - T_{ai} - \frac{d}{2}\left(F^x_{pi}\sin\beta_i - F^y_{pi}\cos\beta_i\right) + \tag{5.16}$$
$$+\frac{d}{2}\left(F^x_{ai}\sin\beta_i - F^y_{ai}\cos\beta_i\right)$$

where,

$$\beta_i = \phi^z_{pi} + \theta p_i \tag{5.17}$$

- Active link of the i^{th} module:

$$\Sigma Fx_{ai} \mathrel{+}= F^x_{ai} \tag{5.18}$$

$$\Sigma Fy_{ai} \mathrel{+}= F^y_{ai} - m_a g \tag{5.19}$$

$$\Sigma Mo_{ai} \mathrel{+}= T_{ai} \tag{5.20}$$

5.3.2 Contributions Due to Contact Point Forces

A normal reaction force and a tangential friction force act on the link at each contact point. The direction of the friction force depends on the direction of motion of the contact point such that it is always opposite to the direction of motion indicated by its tangential velocity.

Point Contact :

Consider the point contact case for a link touching the ground for the j^{th} contact point on the passive link of the i^{th} module,

$$\Sigma F x_{pi} \mathrel{+}= f_j \tag{5.21}$$

$$\Sigma F y_{pi} \mathrel{+}= N_j \tag{5.22}$$

$$\Sigma M o_{pi} \mathrel{+}= -f_j \left(y_{cj} - y_{pi}^m\right) + N_j \left(x_{cj} - x_{pi}^m\right) \tag{5.23}$$

Similarly, for the j^{th} contact point on the active link of the i^{th} module,

$$\Sigma F x_{ai} \mathrel{+}= f_j \tag{5.24}$$

$$\Sigma F y_{ai} \mathrel{+}= N_j \tag{5.25}$$

$$\Sigma M o_{ai} \mathrel{+}= -f_j \left(y_{cj} - y_{ai}^m\right) + N_j \left(x_{cj} - x_{ai}^m\right) \tag{5.26}$$

Line Contact :

If an MTRAN link has two contact points, then there exists a line segment with an infinite number of contact points. This type of contact is called line contact. Clearly, analysis with an infinite number of contact points is not possible. Therefore, the number of contact points is reduced to two by means of the following assertion proven in [36].

Assertion. *In the case of planar contact with Coulomb friction between two bodies, assume that contact between the two bodies occurs along a line segment. Excluding the two contact points at the two ends of the line segment, all of the contact points on the line segment are "redundant". Here, the term "redundant" is used in the sense that the combined effect of "any" set of valid contact forces due to the totality of the contact points on the line segment may equivalently be provided by the combined effect of a set of valid contact forces applied at the two ends of the line segment.*

where, a valid contact force is defined to be a contact force, which satisfies the normal and Coulomb friction force constraints.

Considering the j^{th} and $(j+1)^{th}$ contact points on the passive link of the i^{th} module (see Figure 5.2(b)), one obtains

$$\Sigma F x_{pi} \mathrel{+}= f_j + f_{j+1} \tag{5.27}$$

$$\Sigma F y_{pi} \mathrel{+}= N_j + N_{j+1} \tag{5.28}$$

$$\begin{aligned}\Sigma M o_{pi} \mathrel{+}= & -f_j\left(y_{cj} - y_{pi}^m\right) + N_j\left(x_{cj} - x_{pi}^m\right) - f_{j+1}\left(y_{c(j+1)} - y_{pi}^m\right) + \\ & N_{j+1}\left(x_{c(j+1)} - x_{pi}^m\right)\end{aligned} \tag{5.29}$$

Repeating the same calculation for the active link of the i^{th} module gives

$$\Sigma F x_{ai} \mathrel{+}= f_j + f_{j+1} \tag{5.30}$$

$$\Sigma F y_{ai} \mathrel{+}= N_j + N_{j+1} \tag{5.31}$$

$$\begin{aligned}\Sigma M o_{ai} \mathrel{+}= & -f_j\left(y_{cj} - y_{ai}^m\right) + N_j\left(x_{cj} - x_{ai}^m\right) - f_{j+1}\left(y_{c(j+1)} - y_{ai}^m\right) + \\ & N_{j+1}\left(x_{c(j+1)} - x_{ai}^m\right)\end{aligned} \tag{5.32}$$

5.3.3 Contributions Due to Connection Connection Forces and Moments

Consider Figure 5.3,

- For the passive link of the i^{th} module :

$$\Sigma F x_{pi} \mathrel{+}= -F_{ci}^x \tag{5.33}$$

$$\Sigma F y_{pi} \mathrel{+}= -F_{ci}^y \tag{5.34}$$

$$\Sigma M o_{pi} \mathrel{+}= -M_{ci} - \frac{d}{2}\left(F_{ci}^x \sin\phi_{pi}^z - F_{ci}^y \cos\phi_{pi}^z\right) \tag{5.35}$$

- For the active link of the j^{th} module :

$$\Sigma F x_{aj} \mathrel{+}= F_{ci}^x \tag{5.36}$$

$$\Sigma F y_{aj} \mathrel{+}= F_{ci}^y \tag{5.37}$$

$$\Sigma M o_{aj} \mathrel{+}= M_{ci} + \frac{d}{2}\left(-F_{ci}^x \sin\phi_{pi}^z + F_{ci}^y \cos\phi_{pi}^z\right) \tag{5.38}$$

5.3.4 Constraints on the Contact Point Forces

In addition to the dynamic equilibrium equations, there are also contact point force equations which relate the normal force to the tangential friction force according to the Coulomb friction model. The tangential friction force may be static or dynamic depending on the tangential velocity at the contact point. In order to determine the type of the friction force, the tangential velocity of the corresponding contact point is calculated. For the passive link of the i^{th} module, the tangential velocity of the j^{th} contact point may be written as

$$v_{tj} = \bar{v}_{pi}^{x} + \bar{\omega}_{pi}^{z} \left(y_{cj} - y_{pi}^{m} \right) \tag{5.39}$$

Similarly, the tangential velocity of the j^{th} contact point on the active link of the i^{th} module is

$$v_{tj} = \bar{v}_{ai}^{x} + \bar{\omega}_{ai}^{z} \left(y_{cj} - y_{ai}^{m} \right) \tag{5.40}$$

If the contact point has zero tangential velocity, then there exists static friction. Therefore, the relationship between the normal force and the friction force at the j^{th} contact point is

$$f_j \leq \mu_s N_j \tag{5.41}$$

where, μ_s is the static friction coefficient. If the contact point has nonzero tangential velocity, there exists dynamic Coulomb friction. Therefore, the friction force is given by the expression

$$f_j = -\operatorname{sgn} v_{tj} \, \left(\mu_d \, N_j \right) + \left(1 - |\operatorname{sgn} v_{tj}| \right) f_j \tag{5.42}$$

in the cases of both static and dynamic Coulomb friction. Here, μ_d is the dynamic friction coefficient and the sgn is the signum function, defined as

$$\operatorname{sgn} x = \begin{cases} 1 & if \quad x > 0 \\ 0 & if \quad x = 0 \\ -1 & if \quad x < 0 \end{cases} \tag{5.43}$$

In this study, static friction coefficient is taken as equal to the dynamic friction coefficient and denoted by $\mu = \mu_s = \mu_d$.

Clearly, the normal reaction force has to be greater than or equal to zero, i.e, for the j^{th} contact point:

$$N_j \geq 0 \tag{5.44}$$

since for non-penetrating bodies, it is impossible to have negative normal reaction forces.

Equations (5.41), (5.42) and (5.44) should be satisfied for each of the m contact points.

5.4 Solution of Dynamic Equilibrium Equations

All torques applied by the motors in the modular robots are necessary for the power and work calculation. Therefore the dynamic equilibrium equations should be solved for the motor torques by considering the contact point relations.

Consider a modular robotic system with n modules and m contact points. Depending on the velocity of the contact points, the friction force model can be static or dynamic. Let m_s be the number of contact points with static friction and m_d be the number of contact points with dynamic friction. Therefore, the number of contact points of the system can be written as

$$m = m_s + m_d \tag{5.45}$$

For the i^{th} module of the MTRAN system, the Newton - Euler equations are given by Equations (5.1) - (5.9). Hence, considering n modules, the number of equations due to the Newton-Euler equations becomes $9n$. There are also m_d additional equations (see Equation (5.42)) given by the dynamic Coulomb friction model. As a result, the number of equations is

$$N_{eqn} = 9n + m_d \tag{5.46}$$

The number of unknowns can also be found in terms of the number of modules (n) and the number of contact points (m_s and m_d). There exists 2 unknown motor torques and 4 unknown reaction forces on each module. Therefore; six unknowns are obtained by considering the reaction forces acting on the joints and the motor torques as the unknowns. Therefore, there are totally $6n$ joint reactions and motor torques as unknowns in the system. In addition, two force and one moment reactions exist at each active - passive connection. For an open chain system with $n-1$ connections, totally $3(n-1)$ connection forces and moments exist. Finally, there exists two contact reactions at each contact point, which leads to $2(m_s + m_d)$ unknown contact point reaction forces. Hence, the summation of the unknowns for an n module system with m contact points may be written as

$$\begin{aligned} N_{unk} &= 6n + 3(n-1) + 2(m_s + m_d) \\ &= 9n + 2m_s + 2m_d - 3 \end{aligned} \tag{5.47}$$

where N_{unk} denotes the number of unknowns.

Considering the contact points, there exists several inequalities which must be satisfied for a feasible motion. For each contact point, the normal reaction force must be greater than or equal to zero. This condition brings $m_s + m_d$ inequalities. Moreover, there are m_s extra inequalities (see Equation (5.41)) due to the static Coulomb friction model at the contact points with zero tangential velocity. As a result, the number of inequalities is given by

$$N_{ineq} = (m_s + m_d) + m_s = 2m_s + m_d \tag{5.48}$$

where, N_{ineq} represents the number of inequalities to be satisfied by the solution.

Equating the number of equations, given by Equation (5.46), to the number of unknowns, given by Equation (5.48), one obtains

$$2m_s + m_d = 3 \tag{5.49}$$

If Equation (5.49) is satisfied, one has a regular equation system where $N_{eqn} = N_{unk}$. If, on the other hand,

$$2m_s + m_d < 3 \tag{5.50}$$

one has an overconstrained equation system where $N_{eqn} > N_{unk}$. The case of

$$2m_s + m_d > 3 \tag{5.51}$$

corresponds to the underconstrained case where $N_{eqn} < N_{unk}$. Figure 5.4 shows different m_s, m_d combinations which leads to underconstrained, regular and overconstrained equation systems.

It should be noted that the cases with a single or no contact point, indicated by $\times$ in Figure 5.4, do not lead to a stable locomotion system. Therefore, they are not suitable for locomotion. The overconstrained case corresponding to $m_s = 0$ and $m_d = 2$, on the other hand, may lead to inconsistent set of equations and hence it is considered to be unsuitable for locomotion in this study.

For an underconstrained system the set of equations has a family of solutions instead of a unique solution. Using additional equations, the free parameters can be solved. The solution methods for the regular and underconstrained cases are discussed in the following sections.

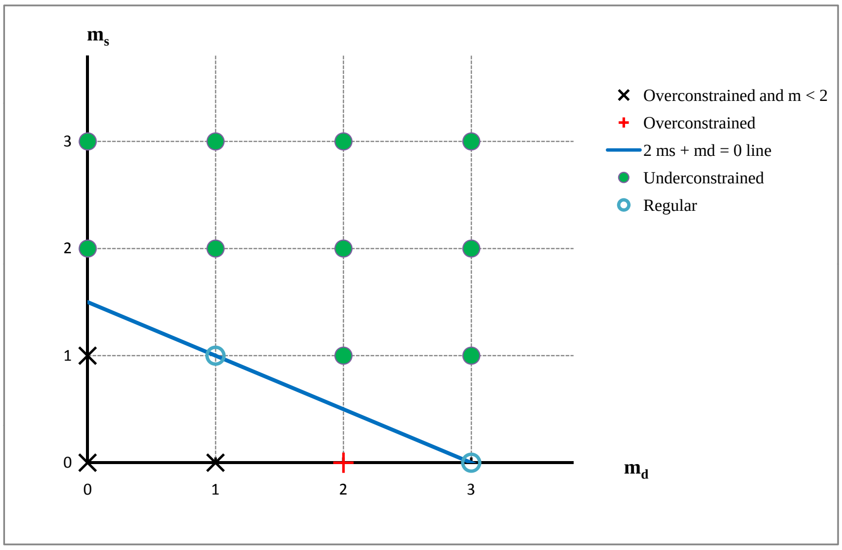


Figure 5.4: Conditions for number of static and dynamic frictional contacts to have under constraint, exact constraint or over constraint systems

5.4.1 Regular System of Equations

If the number of equations is equal to the number of unknowns in the system, then the motor torques can be solved uniquely as long as the determinant of the coefficient matrix is nonzero. In this case, the feasibility of the solution is questioned by checking the inequalities due to the Coulomb friction model and due to the normal reaction forces. If they are all satisfied, the solution is taken as a feasible one and the results are used in further calculations.

Although the number of equations is equal to the number of unknowns, the solution of the equations cannot be found if the determinant of the coefficient matrix is zero. In order to avoid such cases, the conditions that make the determinant of the coefficient matrix zero are studied and the results are presented in Appendix B.

The results show that there are two cases that lead to a zero determinant. First case is the coincident contact points. If the x and y coordinates of any two contact points are the same, the determinant becomes zero. However, it is impossible to satisfy this condition if the physical structure of the system is considered. The second case that makes the determinant of the coefficient matrix zero occurs when all tangential velocities at the contact points are in the same direction. Therefore, having all tangential velocities at the contact points in the same direction leads to infeasible motion and such cases are avoided.

5.4.2 Underconstrained System of Equations

If the number of equations is lower than the number of unknowns, all of the unknowns cannot be determined uniquely because of insufficient number of equations. Instead of a unique solution, a set of solutions is obtained in terms of some of the unknowns. In order to find the values of these unknowns, additional relations are essential. The solution of such systems may be separated into two parts. First part is the solution of the equations by regarding some of the unknowns as free parameters. The second part is the solution of the free parameters and checking the satisfaction of the inequalities due to Coulomb friction model.

5.4.2.1 Solution of Equations in terms of Free Parameters

For the solution of the indeterminate problem,

$$N_{unk} - N_{eqn} = 2m_s + m_d - 3 \tag{5.52}$$

(see Equations (5.46) and (5.48)) unknowns are selected as free parameters in order to equate the number of unknowns to the number of equations.

The free parameters are selected from the normal reaction forces at the contact points. Then, the equations are solved for the remaining unknowns in terms of the free parameters. Similar to the regular system of equations case, the determinant of the coefficient matrix cannot be equal to zero to ensure the uniqueness of the solution. Hence, the conditions, which yield zero determinant, are also investigated for different sets of free parameters. Results are the same as the regular system of equations case. Determinant of the coefficient matrix becomes zero if the contact points are coincident or the tangential velocities at the contact points are in the same direction. If these conditions are not satisfied, the equations can be solved uniquely in terms of the free parameters. The calculation of the free parameters are discussed in the next section.

5.4.2.2 Calculation of Free Parameters

In order to resolve the indeterminacy and determine suitable values for the free parameters, additional relations are required. These relations can be found by considering the deflections of the links. The following assumptions are made while considering the deflections.

- Passive and active links are rigid, whereas the middle links are flexible.
- Forces applied to a middle link, deflect the middle link along the axial direction
- Axial deflections of the middle links do not lead to separation of the contact points from the ground.

Using these assumptions, the changes in the orientations of the middle links due to the deflections of the middle links are calculated and compared with their values corresponding to the preloaded conditions. In order to minimize the difference in the values of the orientations between the preloaded and loaded conditions, the sum of the squares of their difference is minimized with respect to the free parameters to determine the approximate values of the free parameters. During the minimization, inequalities due to the Coulomb friction model and normal reaction forces are also considered in order to have a feasible solution. The results are then substituted into the expressions found (in terms of free parameters) for the motor torques. This procedure is discussed in detail in the following paragraphs.

The axial deflection of the middle link of the i^{th} module can be written as

$$\delta = \int_0^L \frac{H(x)}{EA} dx \tag{5.53}$$

Here, δ is the axial deflection of the link, $H(x)$ is the variable axial force changing with respect to the axial position x, L is the length of the link, E is the modulus of elasticity and A is the cross sectional area. In order to find the variation of $H(x)$, the axial force diagram of the middle link is drawn. This diagram is drawn by calculating the force components along the axial direction of the link, including the inertial effects. Note that the inertial loads are assumed to be lumped at the center of masses of the links. The free body diagram of the middle link is given in Figure 5.5.

In this figure, H_{i1}, H_{i2}, H_{i3} and V_{i1}, V_{i2}, V_{i3} are the axial and normal force components acting on the middle link of the i^{th} module. In addition, β_{iu} is the orientation angle of the middle link of the i^{th} module before the deflection. Equivalence of the normal components in the statically equivalent systems leads to

$$V_{i1} = F_{pi}^x \sin\beta_i - F_{pi}^y \cos\beta_i \tag{5.54}$$

$$V_{i2} = (m_l a_{li}^x) \sin\beta_i - (m_l a_{li}^x + m_l g) \cos\beta_i \tag{5.55}$$

$$V_{i3} = F_{ai}^x \sin\beta_i - F_{ai}^y \cos\beta_i \tag{5.56}$$

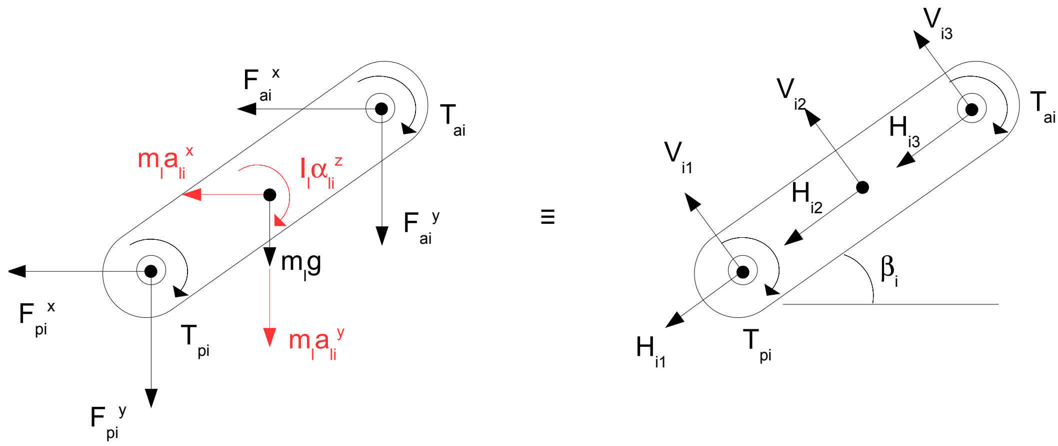


Figure 5.5: Free body diagram of the middle link with axial and normal force components

Similarly, equivalence of the axial components leads to

$$H_{i1} = F_{pi}^{x} \cos \beta_i + F_{pi}^{y} \sin \beta_i \tag{5.57}$$

$$H_{i2} = (m_l a_{li}^{x}) \cos \beta_i + (m_l a_{li}^{x} + m_l g) \sin \beta_i \tag{5.58}$$

$$H_{i3} = F_{ai}^{x} \cos \beta_i + F_{ai}^{y} \sin \beta_i \tag{5.59}$$

The axial force diagram of the middle link is shown in Figure 5.6.

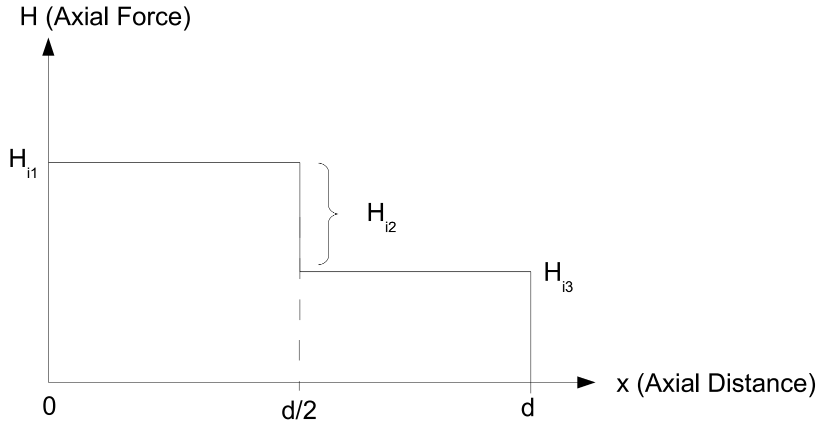


Figure 5.6: Axial force diagram of the middle link of the i^{th} module

The axial forces on the middle link are constant in two different regions, with values H_{i1} and H_{i3}. Therefore, the axial deflection expression in Equation (5.53) leads to the total deflection of the middle link of the i^{th} module given by

$$\delta_i = \frac{H_{i1}\,(d/2)}{EA} + \frac{H_{i3}\,(d/2)}{EA} = \frac{(H_{i1} + H_{i3})\,d}{2EA} \tag{5.60}$$

After finding the axial deflections of the middle links, the orientation angles of the middle links after the deflection should be calculated using the deflected lengths of the middle links. Let β_{il} denote the orientation angle of the middle link of the i^{th} module after the deformation. In order to calculate β_{il}, the system is considered as a multi-loop mechanism. The motors connecting the passive, middle and active links are taken as revolute joints. The contact points are also considered as joints depending on the type of the contact. Point contacts are taken as cam pairs due to the possibility of rotational and translational motion. If there exist a line contact, only translational motion is possible. Therefore line contacts are considered as prismatic joints. To illustrate the procedure, consider the two module system with four contact points shown in Figure 5.7. This system has six links, namely, links

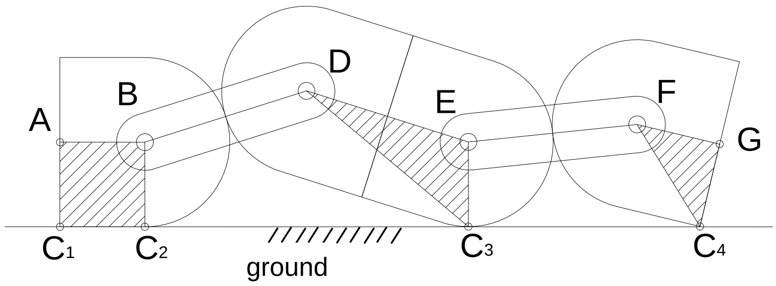


Figure 5.7: Two module system with four contact points

$AC_1C_2B, BD, DC_3E, EF, FC_4G$ and C_1C_4. There are four revolute joints at points B, D, E and F. The contact of the system with the ground consists of one line contact and two point contacts. Corresponding to one prismatic joint and two cam pairs.

The degree of freedom (DOF) of a mechanism, on the other hand, can be calculated via the equation

$$DOF = \lambda(l - j - 1) + \sum_{i=1}^{j} fi \tag{5.61}$$

where, $\lambda = 3$ for planar mechanisms, l is the number of links, j is the number of joints and $\sum fi$ is the totality of the degrees of freedom of the joints in the mechanism. Revolute and prismatic joints have 1 DOF and cam pairs have 2 DOF. Thus, the DOF of the system given in Figure 5.7 is given by

$$DOF = 3(6 - 7 - 1) + (4 \times 1 + 2 \times 2 + 1 \times 1) = 3 \tag{5.62}$$

Hence, three joint variables must be specified in order to find the deflected orientation angles of the links. The joint variables of the mechanism are defined as shown in Figure 5.8. These joint variables are s_1, s_2, s_3, $\lambda_2, \lambda_3, \lambda_4, \lambda_5$ and λ_6. Note that λ_3 and λ_4 are actually dependent ($\lambda_3 = \lambda_4 + \gamma_3$).

Consider the two module system shown in Figure 5.7 with no forces or torques applied at all and let the values of the joint variables be given by

$$s_1 = s_{1u}\,;\;\; s_2 = s_{2u}\,;\;\; s_3 = s_{3u}\,;\;\; \lambda_2 = \lambda_{2u}\,;\;\; \lambda_4 = \lambda_{4u}\,;\;\; \lambda_5 = \lambda_{5u}\,;\;\; \lambda_6 = \lambda_{6u}$$

Clearly, the values of these joint variables are known as well as the "dimensions" of the mechanism (i.e., $\overline{C_1C_2}, \overline{C_2B}, \overline{BD}, \overline{DC_3}, \overline{DE}, \overline{EC_3}, \overline{EF}, \overline{FC_4}, \overline{C_4G}$ and γ_3). Now, consider the actual case where there are forces and torques (including inertial ones) applied to the system. Let the values of the joint variables be now denoted by

$$s_1 = s_{1l}\,;\;\; s_2 = s_{2l}\,;\;\; s_3 = s_{3l}\,;\;\; \lambda_2 = \lambda_{2l}\,;\;\; \lambda_4 = \lambda_{4l}\,;\;\; \lambda_5 = \lambda_{5l}\,;\;\; \lambda_6 = \lambda_{6l}$$

The dimensions of the mechanism are the same as the "unloaded" case except for the flexible links (i.e., middle links BD and EF). The link lengths of these two flexible links for the loaded case are given by

$$\overline{BD}_l = \overline{BD} + \delta_{BD} \tag{5.63}$$

$$\overline{EF}_l = \overline{EF} + \delta_{EF} \tag{5.64}$$

where δ_{BD} and δ_{EF} are the axial deflections of links BD and EF respectively (see Equation (5.60)). One can perform position analysis of the "loaded" mechanism where the dimensions are the same as the "unloaded" one (except for $\overline{BD}_l$ and $\overline{EF}_l$ given by Equations (5.63) and (5.64)).

The inputs for the position analysis are

$$s_1 = s_{1u} = s_{1l}\,;\;\; s_2 = s_{2u} = s_{2l}\,;\;\; s_3 = s_{3u} = s_{3l}$$

which imply that the contact point locations do not change after the loading. The position analysis of the mechanism, which is detailed in the next paragraph, yields the unknown joint variables $\lambda_{2l}, \lambda_{4l}, \lambda_{5l}$ and λ_{6l}.

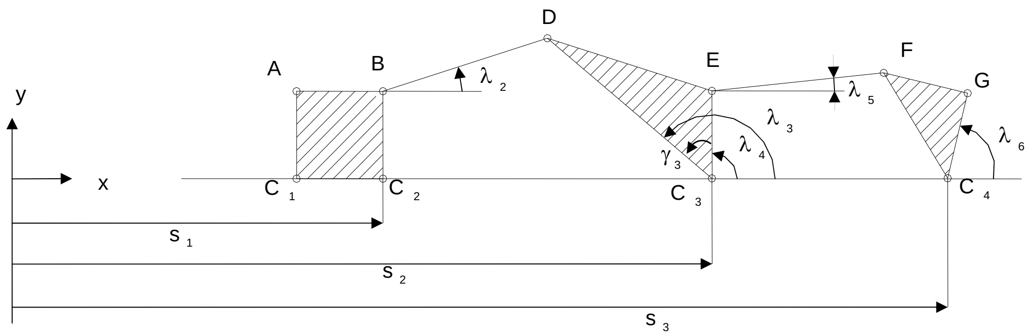


Figure 5.8: Mechanism model of the two module system

For the given input joint variables, the system can be separated into a four-bar and a slider-crank mechanism. The links of the slider-crank mechanism are C_2C_3, C_2B, BD and DC_3. For the input of $C_2C_3 = s_{2u} - s_{1u}$, the position analysis of the mechanism is performed to find λ_{2l}, which is equal to β_{1l}. The joint variable λ_{3l} is also found for further calculations with the four bar mechanism consisting of C_3E, EF, FC_4 and C_3C_4. The length of the link C_3C_4 is found by subtracting s_{2u} from s_{3u}. In addition, λ_{4l} is calculated by subtracting γ_3 from λ_{3l} and it is taken as the input of the four-bar mechanism. Performing position analysis to find the joint variable λ_{5l}, one obtains the other orientation angle β_{2l}. Proceeding similarly, the orientation angles of the middle links for other types of contact point combinations can be calculated. The modeling for different contact point are presented in Appendix D.

After finding the orientation angles of all middle links in the system after the loading, the sum of the square of the differences in the orientation angles before and after the loading for each link is calculated. This summation can be written as

$$Sum = \sum_{i=1}^{n} (\beta_{iu} - \beta_{il})^2 \tag{5.65}$$

This summation is minimized with respect to the unknown free parameters by also considering the inequalities due to friction modeling at the contact points (see

Equation (5.41) and (5.44)). The results of this minimization is substituted into the previously found solution for the dynamic analysis (in terms of free parameters) and a unique solution is obtained for the underconstrained system of dynamic equations.

CHAPTER 6

OPTIMIZATION

For a modular robotic system, an optimal control problem should be solved in order to find the optimal locomotion gaits. In this chapter, the modeling and the solution of the optimal locomotion problem by selecting the performance index, design parameters and physical constraints of the optimization problem are explained.

6.1 Optimal Locomotion Problem

Optimal locomotion problem of modular robots can be defined as the determination of the locomotion gait that minimizes or maximizes the value of a performance index. The solution of such a problem gives the variation of the joint variables with respect to time. Due to the time dependency, the optimization problem for finding the optimal locomotion modes has time dependent unknown functions. In order to solve this optimal control problem, it is converted into a parameter optimization problem. The conversion is achieved by replacing each unknown function by a time dependent expression consisting of a number of constant unknown parameters. The forms of the time dependent expressions, which are used to represent the unknown functions, depend on the preferred form of the time dependent joint variable expressions.

In this study, the locomotion of a modular robot is considered to be a periodic motion, so that the locomotion gait may be used repetitively to go from the starting point to the target point. For a periodic motion, all time dependent expressions, which are used to represent the locomotion, must also be periodic. A periodic locomotion gait must have a common system period which is the lowest common multiple of the periods of the time dependent locomotion functions.

6.2 Design Parameters

For a parameter optimization problem, the design parameters are unknown, constant terms. The values of these parameters are calculated in order to minimize or maximize the performance index. For a modular robot, several design parameters are used in the optimization problem to express the optimal locomotion gait. These parameters are explained in the following sections.

6.2.1 Joint Variable Parameters

The joint variables are expressed as time dependent periodic functions in order to have a periodic modular robot locomotion. The periods of the joint variable functions do not have to be the same. However, the overall system period must be an integer multiple of each joint variable period to complete one cycle of the locomotion gait at the same time.

In order to express the joint variables as a periodic function, the following general periodic function is considered.

$$f(t) = A \sin(\omega t + \zeta) \tag{6.1}$$

In this expression, A is the amplitude, ω is the frequency and ζ is the phase difference. For the periodic locomotion gaits of the modular robots, the overall system period must be an integer multiple of the period of the function. This condition can be implemented by rewriting the frequency of the periodic function as follows.

$$\omega = \frac{2\pi}{\frac{T}{p}} = \frac{2\pi p}{T} \tag{6.2}$$

Here, T represents the overall system period and p denotes an integer, which is called period factor in this study, obtained by dividing the system period by the function period. Therefore, the period of the function becomes T/p. Substituting Equation (6.2) into Equation (6.1) gives

$$f(t) = A \sin\left(\frac{2\pi p}{T} t + \zeta\right) \tag{6.3}$$

The application of this periodic function to express the time dependent terms leads to three design variables for each periodic function. These variables are the amplitude (A), phase shift (ζ) and period factor (p) of the function. Applying the function in Equation (6.3) to the joint variables of the i^{th} module of MTRAN system, one obtains

$$\theta p_i(t) = A_{pi} \sin\left(\frac{2\pi p_{pi}}{T}t + \zeta_{pi}\right) \tag{6.4}$$

$$\theta a_i(t) = \pi + A_{ai} \sin\left(\frac{2\pi p_{ai}}{T}t + \zeta_{ai}\right)$$

In these equations A_{pi}, A_{ai} are the amplitudes, p_{pi}, p_{ai} are the period factors and ζ_{pi}, ζ_{ai} are the phase shifts of the actuators on the passive and active links of the i^{th} module. Consequently, six design parameters are required to express the time dependent joint variables of each module. Hence, for an MTRAN system containing n modules, $6 \times n$ design parameters are necessary for the joint variables. Note that there is a slight difference between the two expressions in Equation (6.4). The difference is that $\theta a_i(t)$ is shifted by π in order to satisfy the constraint for the rotational range of the actuators of the active links.

If the period of the locomotion, T, is not specified, it is also considered as a design parameter for the parameter optimization problem.

6.2.2 System Position Parameters

System position parameters are used to locate the system with respect to the global, fixed reference frame. For a two dimensional analysis, three position and orientation parameters are required. These parameters are $x_s(t), y_s(t))$ and $\gamma_s^3(t)$. Similar to the joint variables, $y_s(t)$ and $\gamma_s^3(t)$ are also defined as periodic functions. Considering Equation (6.3), they may be written as

$$y_s(t) = y_{s0} + A_y \sin\left(\frac{2\pi p_y}{T}t + \zeta_y\right) \tag{6.5}$$

$$\gamma_s^3(t) = A_\gamma \sin\left(\frac{2\pi p_\gamma}{T}t + \zeta_\gamma\right)$$

Here, A_y is the amplitude, p_y is the period factor and ζ_y is the phase shift of the periodic function to represent the y coordinate of the center of mass of the system in the fully open configuration. Similarly, A_γ represents the amplitude, p_γ denotes the period factor and ζ_γ is the phase shift of the periodic function used to represent $\gamma_s^3(t)$. There is also a shifting term, denoted by y_{s0}, for $y_s(t)$. This term is used to increase the mean value of the periodic function to eliminate the possibility of having negative y coordinates. The value of y_{s0} is not known, therefore it is also used as a design parameter. Therefore, by representing $y_s(t)$ and $\gamma_s^3(t)$ as periodic functions with constant design parameters, seven additional design parameters are obtained for the optimization problem.

The x coordinate of the center of mass of the system in fully open configuration ($x_s(t)$) depends on the horizontal velocity of the system. Since its value should always be increasing, it cannot be expressed as a periodic function. However, the horizontal velocity of the center of mass of the system in fully open configuration should be a periodic function to have continuous motion. Considering this term as constant, the x coordinate of the center of mass of the system can be written as,

$$x_s(t) = V_x t \tag{6.6}$$

where, V_x is the constant horizontal velocity of the center of mass of the system in the fully open configuration. If it is unspecified, it can be treated as a design parameter to find the locomotion gait with maximum horizontal velocity.

In summary, the eight design parameters used to represent the system position are $A_y, A_\gamma, n_y, n_\gamma, \zeta_y, \zeta_\gamma, y_{s0}$ and V_x.

6.3 Performance Indices

In this section the expressions used as performance indices for the optimization are presented.

6.3.1 Minimum Distance

Modular robots must always have contact with the ground during the locomotion. Therefore, the following expression is considered in order to ensure continuous contact with the ground.

$$MD = \frac{1}{t_f} \int_0^{t_f} d_{min}(t)^2 dt \tag{6.7}$$

Here, $d_{min}(t)$ is the variation of the minimum distance of the system to the ground, MD is the average value of the square of the minimum distance over time t, and t_f is the final time. This expression may be used as a performance index or, as a constraint.

In order to have continuous contact with the ground, $d_{min}(t)$ must be identically zero leading to $MD = 0$. If MD equals zero, it is guaranteed that there exists at least one contact point between the system and the ground. However, MD cannot be exactly zero due to the round off errors. Hence, the values of MD "close" to zero are also considered to be satisfactory.

Locomotion of modular robots requires continuous contact with the ground. Hence, the minimum distance expression is used as a constraint to provide continuous contact. In this case,

$$MD \leq \epsilon \tag{6.8}$$

is used as a constraint for the optimization problem to compensate the round off errors during the calculations. Here, ϵ is a user defined tolerance value.

6.3.2 Consumed Energy

In order to find the energy spent by the system, the dynamic analysis of the system must be performed, which is discussed in Chapter 5. Assuming that the actuators are not regenerative, the average energy consumption of the i^{th} actuator in a dynamic system can be written as

$$CE_i = \frac{1}{t_f} \int_0^{t_f} |P_i(t)| dt \tag{6.9}$$

where,

$$|P_i(t)| = \begin{cases} |T_i(t)\,\dot{\theta}_i(t)| & \text{for rotational actuators} \\ |F_i(t)\,\dot{s}_i(t)| & \text{for linear actuators} \end{cases} \tag{6.10}$$

In these expressions, $P_i(t)$ is the i^{th} actuator power, $T_i(t)$ is the i^{th} motor torque, $\dot{\theta}_i(t)$ is the angular velocity of the i^{th} actuator, $F_i(t)$ is the force applied by the i^{th} linear actuator and $\dot{s}_i(t)$ is the linear velocity of the i^{th} actuator.

Considering all of the motors in an n module MTRAN system, the total power will be given by

$$|P(t)| = \sum_{i=1}^{n} \Big(|T_{pi}(t)\,\dot{\theta}_{pi}(t)| + |T_{ai}(t)\,\dot{\theta}_{ai}(t)| \Big) \tag{6.11}$$

Therefore, the average consumed energy becomes

$$CE = \frac{1}{t_f} \int_0^{t_f} \left[\sum_{i=1}^{n} \Big(|T_{pi}(t)\,\dot{\theta}_{pi}(t)| + |T_{ai}(t)\,\dot{\theta}_{ai}(t)| \Big) \right] dt \tag{6.12}$$

6.4 Constraints

6.4.1 Number of Contact Points Ratio

Suppose that the time interval for a full cycle of locomotion (i.e., $0 \leq t \leq T$) is divided into N equally spaced intervals. The (N+1) "nodes", along the time axis, are then defined by

$$t_i = i\left(\frac{T}{N}\right) \quad \text{for } i = 0, 1, 2, \ldots, N \tag{6.13}$$

A stable locomotion gait, on the other hand, must always have at least two contact points during the motion. In order to satisfy this condition, a ratio, namely, the percent constraint violation of the number of contact points (RC), is used. This ratio is defined as,

$$RC \triangleq \frac{\text{Number of nodes having 0 or 1 contact point}}{\text{Total number of nodes in one period}} \tag{6.14}$$

According to this definition, the value of RC must be equal to zero for having two or more contact points during the locomotion.

6.4.2 Contact Point Velocities Ratio

As discussed in Section 5.4.1, the velocity directions of the contact points should not be the same for three or more contact points in order to find a dynamically feasible locomotion gait. Therefore, a ratio called percent violation of the tangential velocity directions of the contact points, is defined. Similar to the ratio defined for the number of contact points, it also requires discretization of the locomotion period. This ratio is defined as

$$RV \triangleq \frac{\text{Number of nodes having the same direction for all contact point velocities}}{\text{Total number of nodes in one period}} \tag{6.15}$$

Similar to the ratio defined for the number of contact points, the value of RV must be zero so that the velocity directions of the contact points are not the same.

The percent violation of the tangential velocity directions of the contact points has some special conditions. For example, if the number of contact points is two, the sign of the velocities may be the same (see Appendix B). In addition, the tangential velocities of the contact points of a line contact are considered as a single velocity because there cannot be two tangential velocity with different values on the line contact of the link.

6.4.3 Normal Reaction Forces Ratio

In order to have a feasible locomotion, all normal reaction forces must be greater than or equal to zero throughout the motion. To ensure this condition, percent violation of the normal reaction forces ratio is defined. This ratio can be written as

$$RN \triangleq \frac{\text{Number of nodes having negative normal reaction forces}}{\text{Total number of nodes in one period}} \tag{6.16}$$

If all normal reaction forces at the contact points are non-negative throughout the motion, RN is zero.

6.4.4 Friction Forces Ratio

If the tangential velocity at a contact point is zero, the friction model is considered to be static. For static friction, Inequality (5.41) should be satisfied for all contact points having static friction. In order to guarantee this condition, percent violation of the friction forces ratio (RF) is defined. RF can be written as

$$RF \triangleq \frac{\text{Number of nodes having unsatisfied static friction inequalities}}{\text{Total number of nodes in one period}} \tag{6.17}$$

For a feasible solution, RF must be zero. This ratio is slightly different than the previously defined ratios. If all ratios are zero, the obtained locomotion can be performed on a surface with selected coefficient of friction in calculations. However; it is possible to find that the friction forces ratio is not zero while other ratios are still zero. In this case, the coefficient of friction may not be enough for the obtained locomotion pattern. Increasing the coefficient of friction (selecting a working environment with higher friction) may lead to a feasible motion.

6.4.5 Module Design Constraints

The operational range of the joint variables depends on the module design of the modular robot. Therefore, the limitations on the rotations or translations of the joint variables should be considered as constraints in optimization.

MTRAN system has two actuators with the following rotational limitations

$$-\frac{\pi}{2} \leq \theta p_i(t) \leq \frac{\pi}{2}, \quad \frac{\pi}{2} \leq \theta a_i(t) \leq \frac{3\pi}{2} \tag{6.18}$$

These limitations should be included in the optimization problem for a physically realizable locomotion. Clearly, these constraints should be expressed in terms of the design parameters. Consider the limitation for $\theta p_i(t)$. Substituting the periodic form of $\theta p_i(t)$ (see Equation (6.4)) into this inequality, one obtains

$$-\frac{\pi}{2} \leq A_{pi} \sin\left(\frac{2\pi p_{pi}}{T}t + \zeta_{pi}\right) \leq \frac{\pi}{2} \tag{6.19}$$

Since

$$-1 \leq \sin\left(\frac{2\pi p_{pi}}{T}t + \zeta_{pi}\right) \leq 1 \tag{6.20}$$

Inequality (6.19) implies

$$-\frac{\pi}{2} \leq A_{pi} \leq \frac{\pi}{2} \tag{6.21}$$

The constraints on A_{ai} can be found similarly. Substituting the periodic form of $\theta a_i(t)$ given by Equation (6.4) into the limitation for $\theta a_i(t)$ in Equation (6.18), one obtains

$$\frac{\pi}{2} \leq \pi + A_{ai} \sin\left(\frac{2\pi p_{ai}}{T}t + \zeta_{ai}\right) \leq \frac{3\pi}{2} \tag{6.22}$$

which leads to

$$-\frac{\pi}{2} \leq A_{ai} \sin\left(\frac{2\pi p_{ai}}{T}t + \zeta_{ai}\right) \leq \frac{\pi}{2} \tag{6.23}$$

Inequality (6.23) is exactly the same as the constraint for $\theta p_i(t)$ given by Equation (6.19). Therefore, proceeding similarly, one obtains

$$-\frac{\pi}{2} \leq A_{ai} \leq \frac{\pi}{2} \tag{6.24}$$

The constraints given by Equations (6.21) and (6.24) are considered for all modules in the system Hence, $2 \times n$ optimization constraints are necessary for the rotational limitations of the actuators.

6.4.6 Physical Constraints

Clearly, the period, T, and the horizontal velocity of the center of mass, V_x must be strictly positive, leading to the constraints

$$T > 0 \tag{6.25}$$

$$V_x > 0 \tag{6.26}$$

In order to have a periodic locomotion gait, all time dependent functions have been expressed as periodic functions as explained in the previous sections. Therefore, the locomotion period, T, must be the lowest common multiple of the periods of the time dependent periodic functions. To guarantee this condition, a design parameter, period factor, has been defined. It is obtained by dividing the locomotion period by the period of the corresponding periodic function. Therefore the period factor must be an integer.

6.4.7 Initial Position Limitations

The initial positions of the joint variables can be specified as optimization constraints to indicate the required starting positions of the locomotion gait. To illustrate, consider Equation (6.4) for $\theta p_i(t)$ of an MTRAN system, i.e.,

$$\theta p_i(t) = A_{pi} \sin\left(\frac{2\pi p_{pi}}{T} t + \zeta_{pi}\right) \tag{6.27}$$

Evaluating $\theta p_i(t)$ at $t = 0$, one obtains

$$\theta p_i(0) = A_{pi} \sin\left(\zeta_{pi}\right) \tag{6.28}$$

Equating the right hand side of Equation (6.28) to the specified initial value, one obtains the necessary limitation for the initial position of the actuator in the passive link of the i^{th} module. By repeating similar calculations for all actuators in the system, the initial position of the system can be defined by means of additional constraints.

The final position of the system is the same as the initial position due to the periodicity of the functions. Therefore, it is not necessary to add a constraint for the final position of the system.

6.5 Solution of Optimization Problem

The solution of the optimal locomotion problem requires the minimization of the selected performance index. The possible performance indices, on the other hand, require definite integrals to be evaluated. The integration and the minimization methods applied in this study are discussed in the following parts.

6.5.1 Symbo-Numeric Integration for Performance Indices

The performance indices of the optimization problem, which are discussed in Section 6.3, require integration of complex functions. It is not possible to find analytical solutions for these integrals. This is because the expressions inside the integral depend on the value of the minimum distance and the minimum distance depends on the numerical values of the expressions. Therefore, the integrals can only be evaluated numerically. However, the performance indices also depend on the unknown design parameters. Therefore, it is not possible to execute numerical integration. A convenient method of evaluating these integrals is to use symbo-numeric integration [37].

Consider, for instance, the following example,

$$I = \int_{0}^{t_f} f(a,b,c,t)dt \tag{6.29}$$

In this equation, a, b, c are unknown constant parameters and $f(a,b,c,t)$ is a very complex function, which cannot be evaluated analytically. Symbo-numeric integration yields

$$I = g(a,b,c) \tag{6.30}$$

where, $g(a,b,c)$ is obtained by numerically integrating $f(a,b,c,t)$.

The numerical integrals are calculated by means of the Gaussian Quadrature method. For instance, the definite integral $\int_{-1}^{1} f(x)dx$ can be approximately obtained via

$$\int_{-1}^{1} f(x)dx \cong \sum_{r=1}^{n} w_r f(x_r) \tag{6.31}$$

Here, x_r is the r^{th} absisca and w_r is the r^{th} weighting factor. Note that the limits of the integral are -1 and 1. Therefore, the limits of the integration should also be converted to these values if necessary. The values of the absiscas and the weighting factors are calculated by means of the GaussianQuadratureWeights function of the Mathematica software

The accuracy of the numerical integration depends on the number of quadrature points used for the evaluation. Increasing the number of points, more accurate results are obtained. However, the execution time of the integral is also increased. Therefore a compromise must be made.

6.5.2 Minimization of the Performance Index

The minimization of the performance index is achieved by means of the NMINIMIZE function of the MATHEMATICA software [38]. NMINIMIZE function numerically determines the design parameters which minimize the objective function subjected to a number of constraints. In order to use NMINIMIZE function, the method of minimization and the minimization options must be selected.

NMINIMIZE function uses four different numerical minimization methods. These methods are NelderMead, Simulated Annealing, Differential Evolution and Random Search. The algorithms used in these methods and their optimization techniques can be found in the Mathematica Documentation [38].

Some options of the NMINIMIZE function are used to define the maximum number of iterations, the tolerance of calculation and the accuracy of the minimization. The maximum number of iterations for NMINIMIZE is defined via the "MaxIterations" option. It is used to terminate the calculations if a solution is not found at the end of the maximum number of iterations. "Tolerance" option is used for the constraints in order to evaluate the acceptability of the constraint violation. In addition, two options are used for the accuracy of the final value of the objective function. These options are named as "AccuracyGoal" and "PrecisionGoal". MATHEMATICA attempts to ensure the numerical error for the result x to be less than $10^{-a} + |x|\, 10^{-p}$ where a is the AccuracyGoal and p is the PrecisionGoal. Specifying these options appropriately, the numerical error of the optimization can be adjusted to desired values.

The optimization problem involves several unknown design variables. As a result, it is possible to have many local extreme points. Although the algorithm of the NMINIMIZE function is developed to find the global optimum, it may converge to suboptimum values depending on the starting values of the design parameters. Hence, the NMINIMIZE function should be used for different initial values in order to find a better optimum value. The starting values of this function can be altered by means of the "RandomSeed" option. The default value of this option is zero. In this study the optimization algorithm is executed several times by changing the value of RandomSeed, in order to find better solutions.

6.6 Post Processing of the Solution

Solution of the optimal locomotion problem by means of the NMINIMIZE function yields the numerical values of the unknown design parameters. Substituting the values of the optimal design parameters into the corresponding expressions, one obtains the variations of the joint variables and the position and orientation parameters of the system with respect to time. These parameters are sufficient to describe the locomotion gait.

After finding the locomotion gait, the characteristic of the motion is investigated by means of visual simulation. This simulation visualize the motion of the links with respect to the ground. The details of the visual simulation is presented in Appendix D.

CHAPTER 7

CASE STUDIES

In this chapter, the algorithm developed to solve the optimal locomotion problem of modular robotic systems is illustrated via some case studies by using the MTRAN system. In addition, several snapshots of the visual simulation of the determined solutions are presented to show the motions of the links.

The flowchart of the optimization algorithm is shown Figure 7.1 and the necessary inputs are explained below.

- I - 1 (System and workspace definition parameters)
  - Module dimensions
  - Number of modules
  - Dynamic properties of the links (mass, inertia, etc . . .)
  - Coefficient of Coulomb friction, for dynamic and static cases, for the working environment
  - Tolerance for contact point evaluation
  - Number of Intervals for evaluating ratios (RC, RV, etc...) = 20
- I - 2 (Numerical integration parameters)
  - Gauss Quadrature integration precision
  - Number of points to be used for the Gauss Quadrature method
- I - 3 (Optimization parameters for the NMINIMIZE function)
  - Method (NelderMead, DifferentialEvolution, SimulatedAnnealing)
  - Numerical calculation options (MaxIterations, Tolerance, AccuracyGoal, PrecisionGoal, WorkingPrecision, RandomSeed)

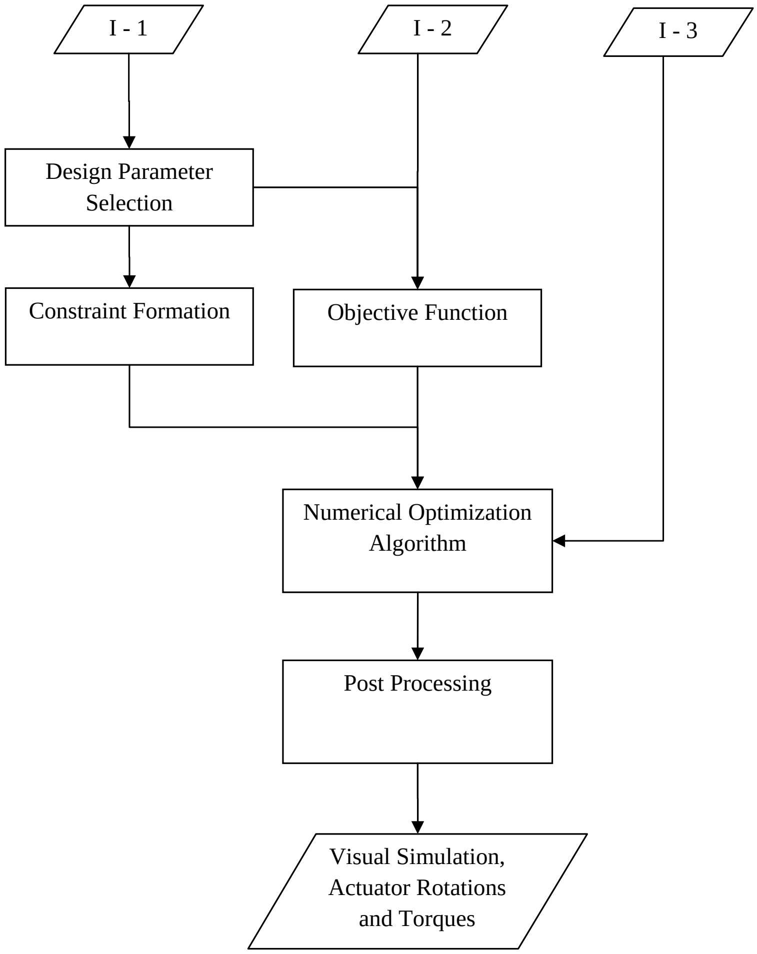


Figure 7.1: Flowchart of the optimization algorithm

After the specification of the inputs, the NMINIMIZE function tries to find an optimum value of the objective function, subjected to several constraints, by means of an iterative approach. Flowchart of a single iteration is presented in Figure 7.2. These iterations are continued until the specified accuracy is reached or the maximum number of iterations is exceeded.

The number of design parameters depends on the type of periodic function that is selected to be used and the number of modules. Considering a sine function for all time dependent joint variables, $6 \times n$ parameters are necessary to indicate the variation of the joint variables (with respect to time) for an n module system. Furthermore, there are 8 more design parameters which are used to express the position of the center of mass of the system, in the fully open configuration, as

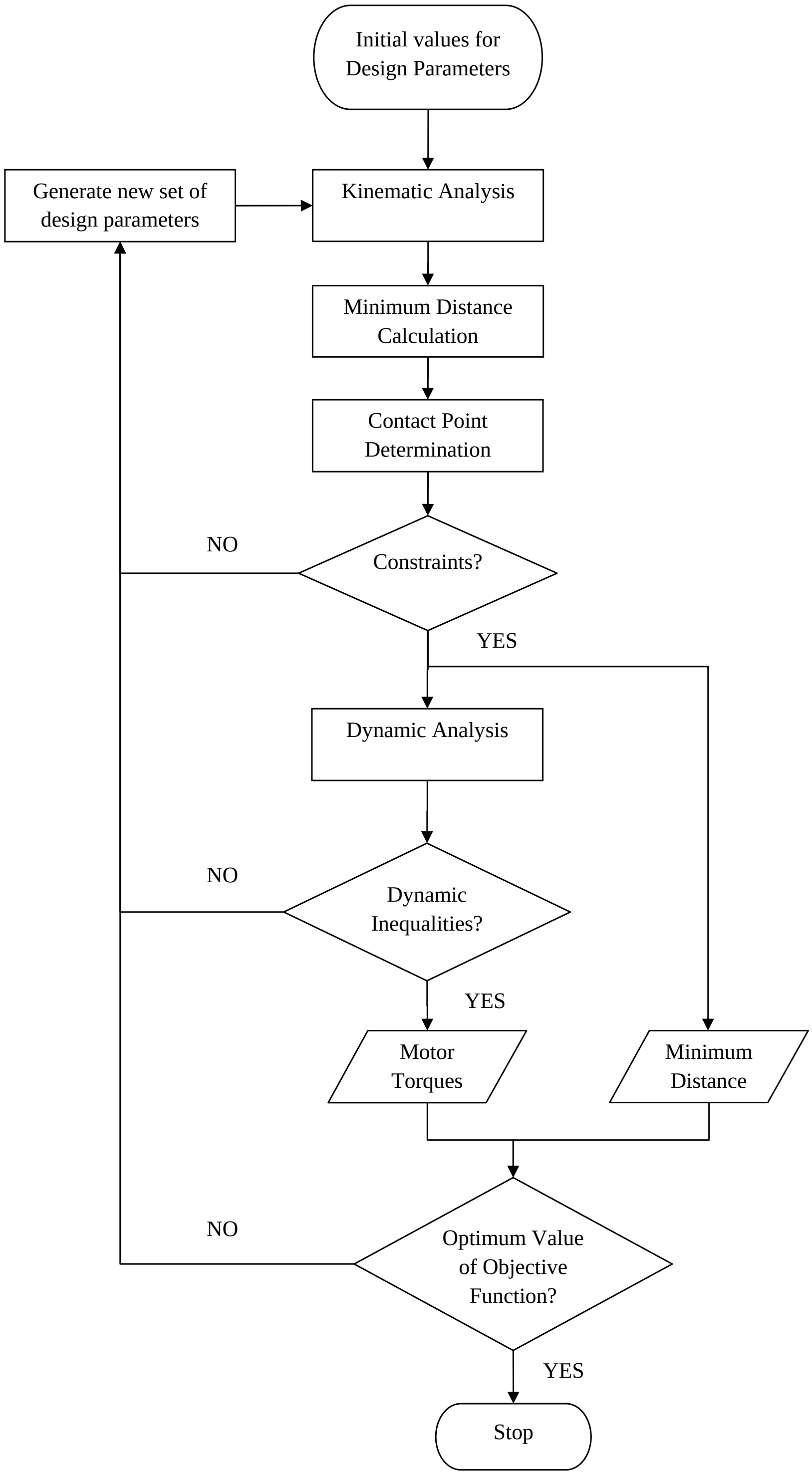


Figure 7.2: Flowchart of the optimization iterations

discussed in Section 6.2.2. Finally, the overall system period is also considered as a design parameter. Consequently, there are $6n + 9$ design parameters for an n module system.

In the case studies, the following abbreviations are used.

- MD : Minimum Distance
- CE : Consumed Energy
- RC : Number of Contact Points Ratio
- RV : Contact Point Velocities Ratio
- RN : Normal Reaction Forces Ratio
- RF : Friction Forces Ratio

In the following sections, the solution of the optimal locomotion problem for an MTRAN system consisting of two modules, is presented. For this analysis, necessary input parameters are taken as follows. Note that, some options of the NMINIMIZE function are changed during the case studies. Hence, they are presented in the corresponding sections.

- I - 1 :
  - $d = 65mm$
  - $n = 2$
  - $m_p = m_l = m_a = 0.140kg$
  - $I_p = I_a = 8.774 \times 10^{-5} kg.m^2$, $I_l = 8.580 \times 10^{-5} kg.m^2$
  - $\mu = \mu_s = \mu_d = 0.3$
  - $\epsilon = 2mm$
- I - 2 :
  - Gauss Quadrature integration precision = 40
  - Number of points to be used for the Gauss Quadrature method = 20
- I - 3 :
  - MaxIterations = 500
  - Tolerance = 10^{-5}

- AccuracyGoal = 20
- PrecisionGoal = 20
- WorkingPrecision = 40

After the specification of the necessary inputs, the design parameters are selected. For a two module system, there exists $(6 \times 2) + 9 = 21$ design parameters. In the following sections, the solution of the optimal locomotion problem for a two module MTRAN system is presented. The problem is solved by considering different objective functions and constraints. Limitations due to the design of the modules and due to the physical constraints are given to the optimization algorithm for all cases. However, in addition to these constraints, RC, RV, RN and MD expressions are not defined as constraints for all cases. Solution of the problem for these different cases and their results are discussed in the following sections.

All calculations presented in the following sections are performed on a desktop pc with Intel Core 2 Duo 6700 @ 2.66 GHz processor and 4 GB DDR2 800 MHz RAM.

7.1 Case 1 : Minimize MD

In this case, minimization of MD subjected to only module design and physical limitations is performed. Clearly, the minimum value of MD should be exactly zero.MD being equal to zero guarantees that the modular robotic system is in contact with the ground at all times. However, the number of contact points may not be sufficiently large for a feasible motion. Furthermore, in this case, the horizontal velocity does not affect the optimization process, since MD considers only the vertical distance of the system from the ground. Therefore, further analysis is needed for the determination of the horizontal velocity of the robot.

In Table 7.1, results of the optimization by means of different methods are presented. For each method, 51 different starting points are used by increasing the value of the RandomSeed option of NMinimize from 0 to 50, one by one. The data shown in Table 7.1 correspond to the best results obtained for the 51 trials. In order to decrease the necessary execution time for the optimization, AccuracyGoal and PrecisionGoal options are specified as 10 for this part only.

The locomotion parameters that corresponds to the best MD values, listed in Table 7.1, are given below. Note that, the horizontal position of the system, $x_s(t)$, is selected arbitrarily due to the fact that, the horizontal position does not affect the value of MD.

Table 7.1: Results of the minimization of MD by means of different optimization methods

Method	**Best MD** (mm^2)	**Total Time** (h)	**Average Time for one trial** (s)
Nelder-Mead	0.0012983	4.49	317.2
Differential Evolution	0.0018671	6.13	432.7
Simulated Annealing	0.0064715	5.16	364.5

- Nelder-Mead Solution Method :
    - $\theta p_1(t) = -0.118 \sin(2.538 - 1.902t)$
    - $\theta a_1(t) = \pi - 0.048 \sin(5.764 - 1.902t)$
    - $\theta p_2(t) = -0.006 \sin(0.0075 + 3.803t)$
    - $\theta a_2(t) = \pi + 0.118 \sin(2.403 + 1.902t)$
    - $x_s(t) = 0.863t\, mm$
    - $y_s(t) = 33.717 + 1.118 \sin(4.922 - 1.902t)\ mm$
    - $\gamma_s^3(t) = 0.013 \sin(3.213 - 1.902t)$
    - $T = 3.304\, s$
- Differential Evolution Solution Method :
    - $\theta p_1(t) = -0.120 \sin(2.651 + 0.992t)$
    - $\theta a_1(t) = \pi + 0.044 \sin(6.251 - 0.992t)$
    - $\theta p_2(t) = 0.033 \sin(0.152 + 0.992t)$
    - $\theta a_2(t) = \pi - 0.004 \sin(4.371 - 2.975t)$
    - $x_s(t) = 41.650t\, mm$
    - $y_s(t) = 33.891 - 0.720 \sin(6.283 + 1.984t)\ mm$
    - $\gamma_s^3(t) = 0.0211 \sin(5.447 - 0.992t)$
    - $T = 6.335\, s$
- Simulated Annealing Solution Method :
    - $\theta p_1(t) = 0.056 \sin(4.037 - 2.381t)$
    - $\theta a_1(t) = \pi - 0.0495 \sin(2.381t)$
    - $\theta p_2(t) = -0.020 \sin(0.407 - 4.761t)$
    - $\theta a_2(t) = \pi + 0.0495 \sin(1.236 + 2.381t)$

- $x_s(t) = 35.946t\, mm$
- $y_s(t) = 34.201 - 1.644 \sin(0.095 - 2.381t)\ mm$
- $\gamma_s^3(t) = -0.007 \sin(3.428 - 4.761t)$
- $T = 2.639\, s$

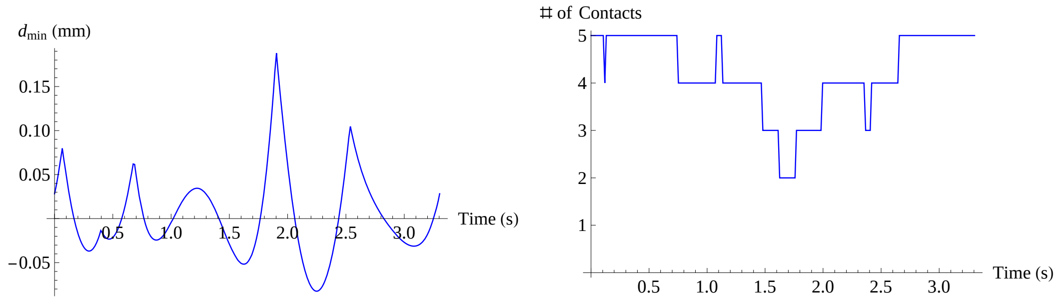
The results indicate that, all methods successfully find a minimum of MD. However, the Differential Evolution method requires more time with respect to the other methods. The variations of the number of contact points and the minimum distance (with respect to time) corresponding to the results given above are shown in Figure 7.3. Although the number of contact points is 2 or more, throughout the motion, for the solutions with the Nelder Mead and Simulated Annealing methods, the number of contact points drops to one for the solution with the Differential Evolution method. In order to avoid such cases, a constraint related to the number of contact points, which is discussed in the next section, is defined.

7.2 Case 2 : Minimize MD subjected to RC = 0

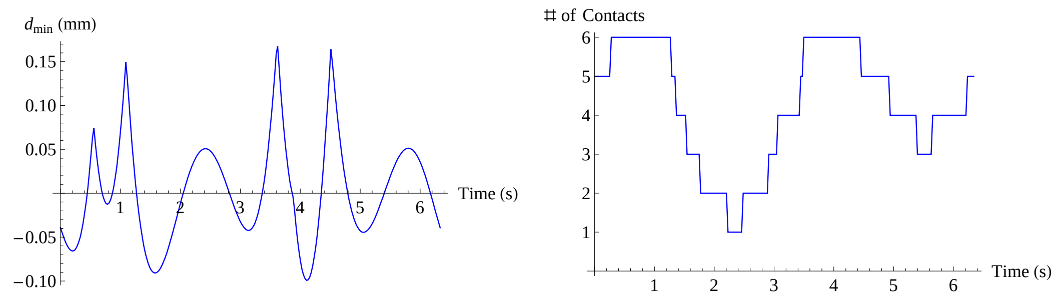
In order to have stable solutions with two or more contact points, throughout the motion, the number of contacts ratio, which is explained in Section 6.4.1, is used. If the number of contact points is greater than or equal to two for the whole period then RC becomes zero. Hence, the addition of RC = 0 condition to the optimization is necessary to obtain a locomotion with at least two contact points during the motion.

Similar to the previous case, 51 different initial values are used for each of the three different optimization methods. In the evaluation of RC, equally spaced 20 intervals are used per period. The results are presented in Table 7.2. To satisfy the constraint RC = 0 with the NMINIMIZE function, the AccuracyGoal and the PrecisionGoal options are set to 20, whereas the WorkingPrecision is increased to 40.

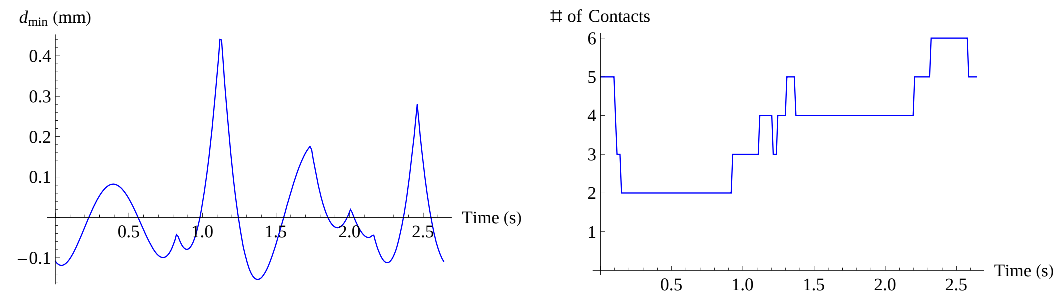
The resulting locomotion parameters, corresponding to the best solutions for the three different methods are presented below. Similar to the previous case, the horizontal velocity is completely arbitrary.



(a)



(b)



(c)

Figure 7.3: Minimum distance (d_{min}) and the number of contact points (♯ of Contacts) with respect to time for (a) Nelder-Mead method, (b) Differential Evolution method, (c) Simulated Annealing method

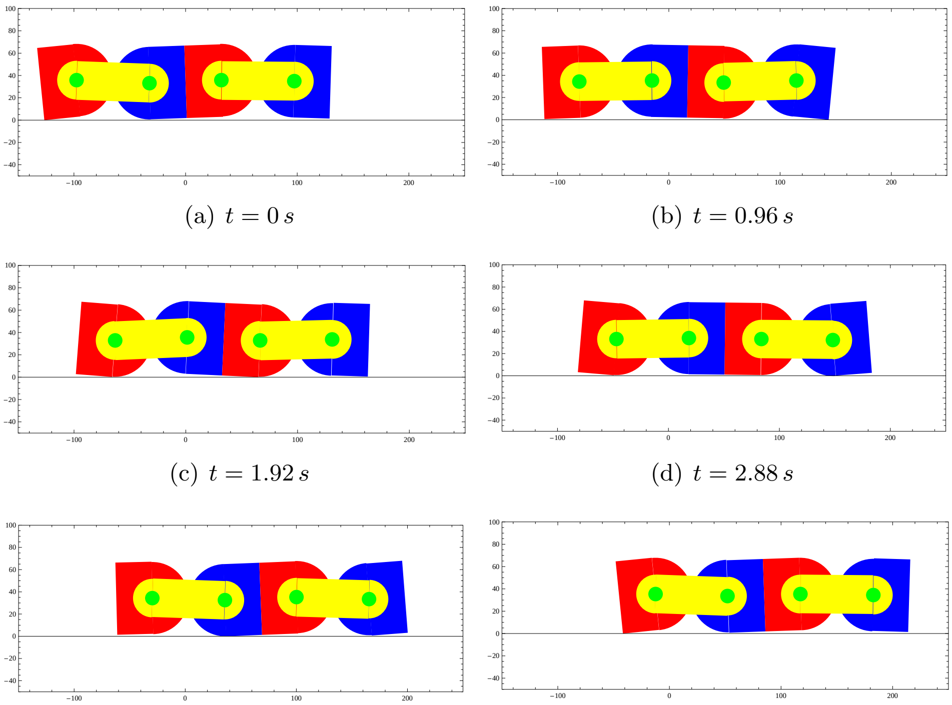
Table 7.2: Results of minimization of MD, which is subjected to the RC = 0 condition, by means of different optimization methods

Method	**Best MD** (mm^2)	**Total Time** (h)	**Average Time for one trial** (s)
Nelder-Mead	0.0044389	7.45	525.6
Differential Evolution	0.0013775	7.37	520.3
Simulated Annealing	0.0035569	8.56	604.4

- Nelder-Mead Method Solution :
    - $\theta p_1(t) = 0.138 \sin(4.580 - 1.307t)$
    - $\theta a_1(t) = \pi + 0.089 \sin(0.918 - 1.307t)$
    - $\theta p_2(t) = -0.073 \sin(2.493 + 1.307t)$
    - $\theta a_2(t) = \pi - 0.121 \sin(2.970 - 1.307t)$
    - $x_s(t) = 17.664t\, mm$
    - $y_s(t) = 33.990 - 0.522 \sin(5.593 - 1.307t)\ mm$
    - $\gamma_s^3(t) = 0.046 \sin(0.862 - 1.307t)$
    - $T = 4.805\, s$
- Differential Evolution Method Solution :
    - $\theta p_1(t) = 0.104 \sin(0.022 + 0.789t)$
    - $\theta a_1(t) = \pi - 0.006 \sin(1.571 + 1.578t)$
    - $\theta p_2(t) = -0.010 \sin(1.612 + 1.578t)$
    - $\theta a_2(t) = \pi - 0.062 \sin(0.019 + 0.789t)$
    - $x_s(t) = 17.188t\, mm$
    - $y_s(t) = 33.465 + 0.064 \sin(2.629 + 2.367t)\ mm$
    - $\gamma_s^3(t) = -0.014 \sin(6.283 - 0.789t)$
    - $T = 7.964\, s$
- Simulated Annealing Method Solution :
    - $\theta p_1(t) = -0.058 \sin(4.533 + 0.118t)$
    - $\theta a_1(t) = \pi + 0.029 \sin(5.320 + 0.118t)$
    - $\theta p_2(t) = 0.020 \sin(2.726 - 0.118t)$

- $\theta a_2(t) = \pi - 0.077 \sin(2.387 - 0.118t)$
- $x_s(t) = 34.218t\, mm$
- $y_s(t) = 33.477 - 1.042 \sin(3.634 - 0.118t)\; mm$
- $\gamma_s^3(t) = -8.56610^{-12} \sin(6.278 + 0.118t)$
- $T = 0.100\, s$

According to the results shown in Table 7.2, all methods successfully determine a set of design parameters which minimize the MD. In addition the solutions have two or more contact points during the motion Hence, they satisfy the constraint RC $= 0$ as shown in Figure 7.5.



(a) $t = 0\, s$ (b) $t = 0.96\, s$

(c) $t = 1.92\, s$ (d) $t = 2.88\, s$

(e) $t = 3.84\, s$ (f) $t = 4.80\, s$

Figure 7.4: Some snapshots of the visual simulation of the system corresponding to the solution obtained by Nelder-Mead method

If the total execution times of the optimizations for the 51 different starting points are considered for different methods, it can be seen that the Differential Evolution method requires the longest time and the Nelder-Mead method requires the shortest time. Therefore, it is better to use Nelder-Mead method for the minimization of MD subjected to RC $= 0$. The motion of the system for the solution obtained by

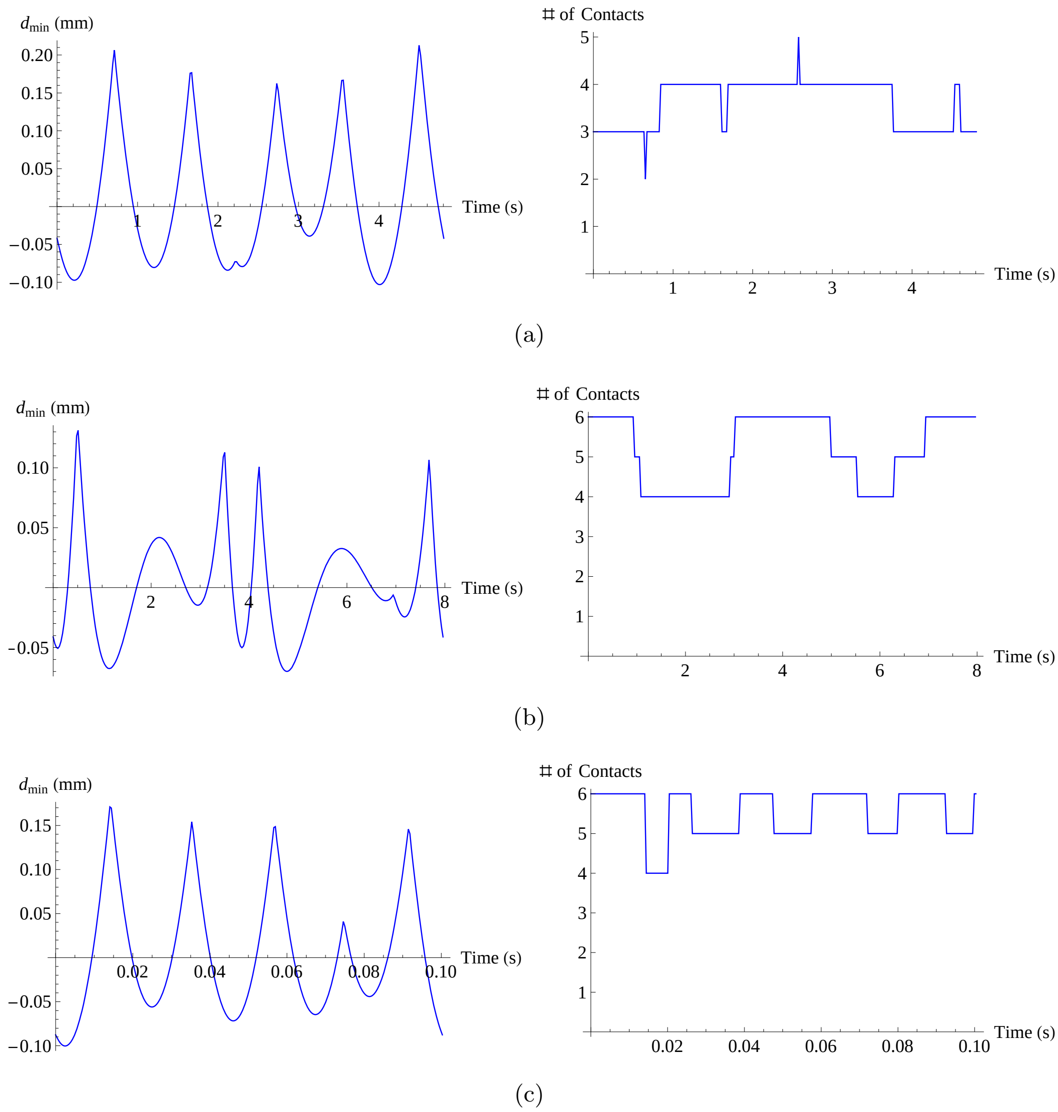
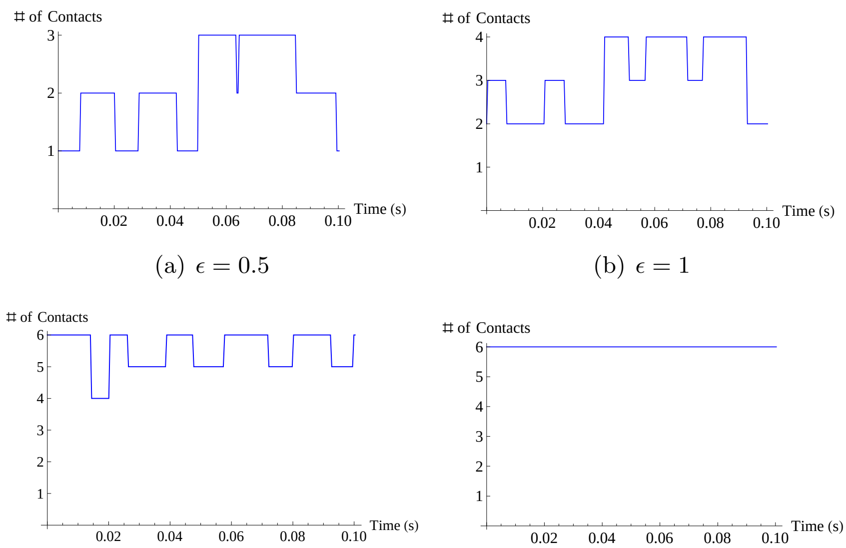


Figure 7.5: Minimum distance (d_{min}) and the number of contact points (♯ of Contacts) with respect to time for (a) Nelder-Mead method, (b) Differential Evolution method, (c) Simulated Annealing method

Nelder-Mead method is visualized via the visual simulation program and several snapshots of the simulation are given in Figure 7.4.

The number of contact points depend upon the numerical tolerance (ϵ) that is selected. If ϵ is selected to be too small, it may be concluded that there is no contact with the ground. On the other hand, if it is selected to be too large, all of the links may seem to be in contact with the ground. As an example, consider the solution obtained by the Simulated Annealing method. For $\epsilon = 2\,mm$, it can be seen that the number of contact points is changing between 4 and 6. However; if the value of the tolerance is decreased to $0.5\,mm$, the constraint RC $= 0$ cannot be satisfied. If ϵ is taken to be $1\,mm$, then $RC = 0$ condition is satisfied. Therefore, the value of the tolerance plays an important role in the determination of the contact points. The number of contact points variations for different ϵ values are presented in Figure 7.6.



(a) $\epsilon = 0.5$ (b) $\epsilon = 1$ (c) $\epsilon = 2$ (d) $\epsilon = 3$

Figure 7.6: Number of contact points vs time for different values of tolerance

7.3 Case 3 : Solution subjected to RC = 0, RV = 0, RN = 0 and RF = 0

In order to minimize the total energy consumed by the actuators of the modular robot, firstly dynamically feasible solutions are determined. As discussed in the previous sections, a dynamically feasible solution requires $RC = 0$, $RV = 0$, $RN = 0$ and $RF = 0$.

The solutions which satisfy the $RC = 0$ constraint are easily determined. However; the determination of the solutions which satisfy the remaining constraints are difficult. In order to find such solutions, the NMINIMIZE function is employed using different optimization methods. The Nelder - Mead, Simulated Annealing and Differential Evolution methods are used with different starting points to find solutions, which minimizes a constant, pseudo objective function subjected to the constraints $RC = 0$, $RV = 0$, $RN = 0$ and $RF = 0$. These attempts have not led to a feasible solution satisfying all of the constraints. A possible reason of this failure is the size of the search space. Considering a two-module MTRAN system, for instance, there are 21 design parameters to be determined for an optimal locomotion, which leads to a 21-dimensional search space. Therefore, the NMINIMIZE function could not find a solution which satisfies all constraints.

On the other hand, for the solutions which satisfy the $RC = 0$ constraint, it is possible to change some of the design parameters and still fulfill the $RC = 0$ constraint. Horizontal velocity can be considered to be such a design parameter, since it does not affect RC. Furthermore, the system period, T, has been selected to be the second design variable to be manipulated to satisfy the constraints $RV = 0$, $RN = 0$ and $RF = 0$.

Applying a divide and conquer approach, the determination of the solutions, which satisfy $RC = 0$, $RV = 0$, $RN = 0$ and $RF = 0$, is separated into two steps. Firstly, many solutions satisfying the $RC = 0$ condition is determined. Then, the remaining constraints are satisfied by changing the values of the horizontal velocity and the system period. In this procedure, the values of the design parameters, excluding the horizontal velocity of the system and system period, are taken from the previously determined solutions.

By means of this procedure, the following solution, which satisfies $RC = 0$, $RV = 0$, $RN = 0$ and $RF = 0$, has been determined by dividing the period into 50 intervals. The variations of the minimum distance of the system to the ground and the number of contact points for this solution are shown in Figure 7.7. The resulting locomotion parameters for this solution are as follows.

- $\theta p_1(t) = -0.202 \sin(5.609 + 15.708t)$
- $\theta a_1(t) = \pi + 0.129 \sin(4.699 + 15.708t)$

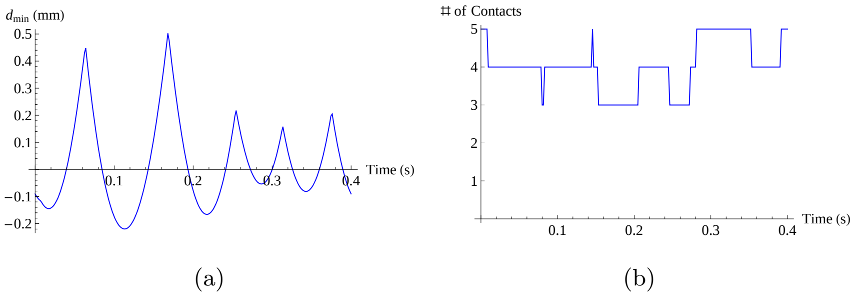


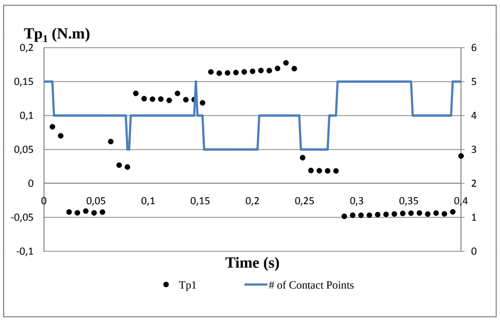
Figure 7.7: Minimum distance (d_{min}) and the number of contact points (♯ of Contacts) with respect to time for the solution, which satisfies all constraints

- $\theta p_2(t) = 0.074 \sin(1.244 + 15.708t)$
- $\theta a_2(t) = \pi - 0.124 \sin(4.022 + 15.708t)$
- $x_s(t) = 6t\, mm$
- $y_s(t) = 34.730 - 0.011 \sin(4.100 + 15.708t)\ mm$
- $\gamma_s^3(t) = 0.074 \sin(1.841 - 15.708t)$
- $T = 0.4\, s$

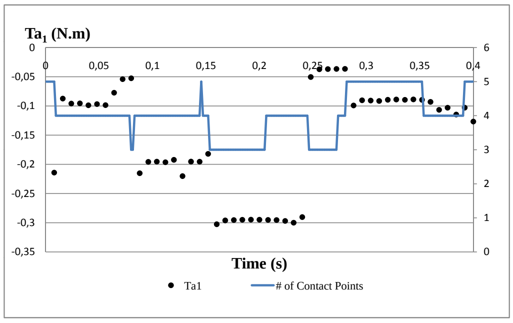
For this solution, the variations of the motor torques with respect to time for a single period are presented in Figure 7.8. As shown in the plots, the motor torques are not continuous. This is due to the fact that the number of contact points does not change continuously (with respect to time). As soon as a contact between a link and the ground occurs, the normal and friction reaction forces are applied to this contact point, which results in sudden changes in the dynamic structure of the system.

Several snapshots of the motion of the system corresponding to the obtained solution are given in Figure 7.9.

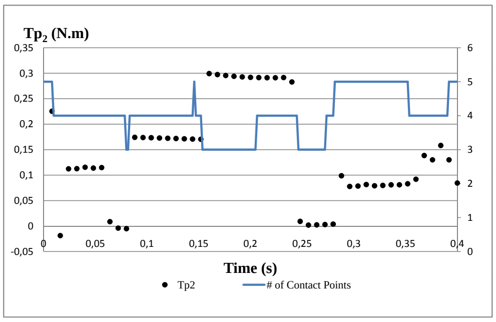
In order to perform dynamic analysis for this locomotion pattern, solution of the dynamic equations for underconstrained cases have been performed. As an example, consider the system at time $t = 0.22s$ where the number of contact points is 4. The snapshot of the position of the system is shown in Figure 7.10. This system position is considered to be the combination of a slider-crank mechanism and a four bar mechanism. The mechanism model and related joint variables are shown in



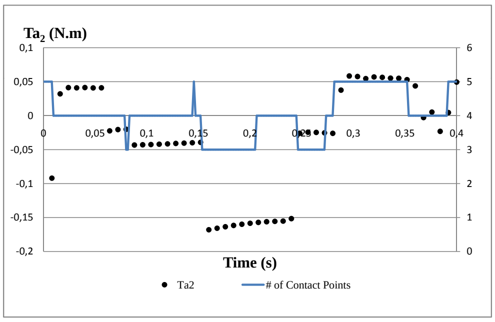
(a)



(b)



(c)



(d)

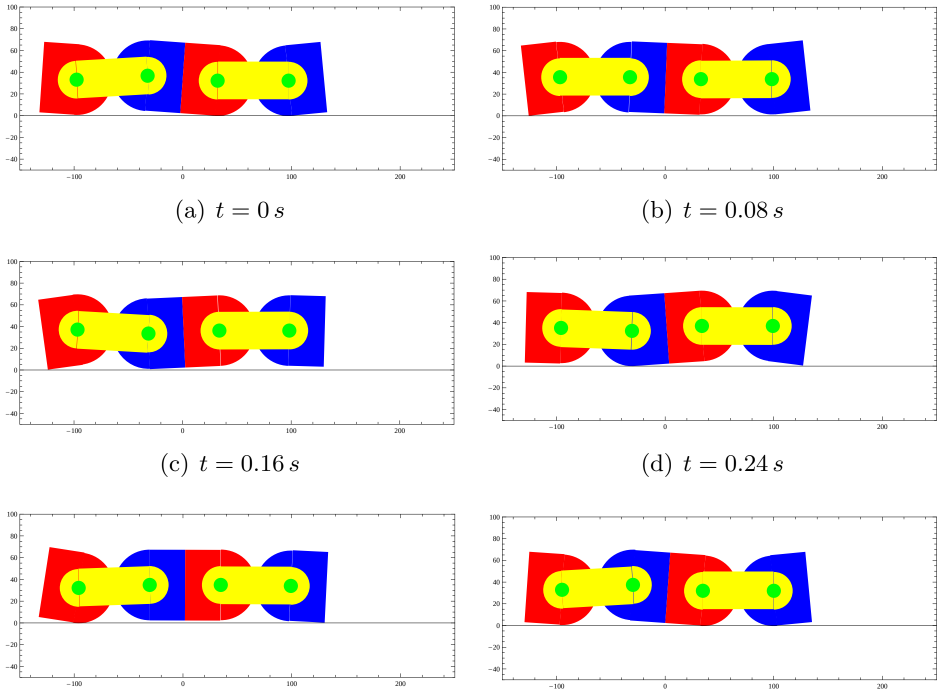
Figure 7.8: Variations of motor torques with respect to time

Figure 7.11. Performing the analysis using one free parameter gives the results in Table 7.3. In the deformation analysis, the Young's Modulus has been taken to be $E = 3.1GPa$ for the material delrin, which is a type of an engineering plastic.

The results show that the joint variables after the loading are very close to the joint variables before the loading, since the axial deflections are very small with respect to link length. According to this analysis, motor torques are calculated as $T_{p1} = 0.1704\,Nm$, $T_{a1} = -0.2974\,Nm$, $T_{p2} = 0.2914\,Nm$ and $T_{a2} = -0.1553\,Nm$.

7.4 Case 4 : Minimize CE

It is indeed possible to minimize CE subject to the constraints $RC = 0$, $RV = 0$, $RN = 0$ and $RF = 0$. However, due to the large number of design parameters, complicated constraints and objective function; the various attempts that have been made using NMINIMIZE have failed to yield a satisfactory result. Hence, in this case study, the number of design variables is reduced to two (namely, the horizontal velocity V_x and the period, T). The remaining design variables are fixed at their



(a) $t = 0\,s$ (b) $t = 0.08\,s$

(c) $t = 0.16\,s$ (d) $t = 0.24\,s$

(e) $t = 0.32\,s$ (f) $t = 0.40\,s$

Figure 7.9: Some snapshots of the visual simulation of the motion of the system corresponding to the solution which satisfies all dynamic analysis constraints

values obtained in case study 3. Recall that case study 3 has yielded a solution which satisfies all constraints (i.e., $RC = 0$, $RV = 0$,$RN = 0$ and $RF = 0$) leading to $CE = 0.202J$.

The design parameter space considered in this case study is given by $5mm/s \leq V_x \leq 10mm/s$ and $0.2s \leq T \leq 2s$. A grid is formed in this space by taking $\Delta V_x = 0.1mm/s$ and $\Delta T = 0.01s$. CE is evaluated at each node of the grid provided that all constraints are satisfied. As a result, a suboptimal solution, yielding $CE = 0.192J$ (i.e. a %5 decrease in consumed energy), is obtained. The resulting locomotion parameters are given below.

- $\theta p_1(t) = -0.202 \sin(5.609 + 13.369t)$
- $\theta a_1(t) = \pi + 0.129 \sin(4.699 + 13.369t)$
- $\theta p_2(t) = 0.074 \sin(1.244 + 13.369t)$
- $\theta a_2(t) = \pi - 0.124 \sin(4.022 + 13.369t)$

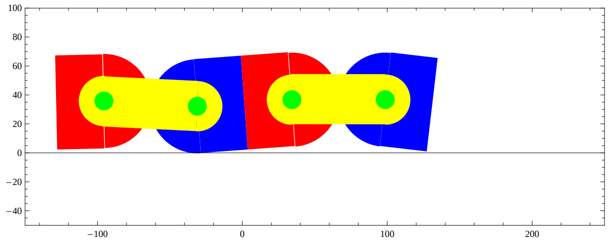


Figure 7.10: System position at $t = 0.22s$

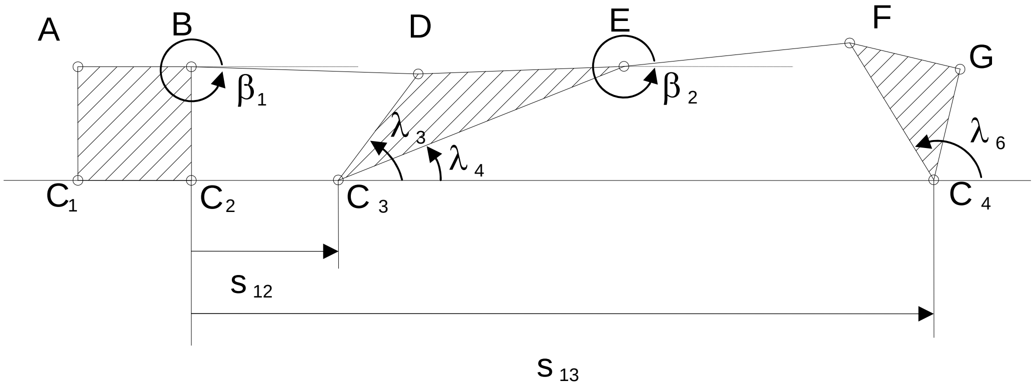


Figure 7.11: Mechanism model of the system and related joint variables at $t = 0.22s$

Table 7.3: Results of the underconstrained dynamic analysis for $t = 0.22s$

Parameter	Before the Loading	After the Loading
β_1 (rad)	-0.04939	-0.00836
β_2 (rad)	-0.00084	-0.02730
δ_1 (mm)	0	-1.17004
δ_2 (mm)	0	0.07546
s_{12} (mm)	58.0683	58.0683
s_{13} (mm)	164.6208	164.6208
λ_3 (rad)	1.3760	1.3892
λ_4 (rad)	0.4759	0.4811
λ_6 (rad)	2.2398	2.2719

- $x_s(t) = 5.4t\, mm$
- $y_s(t) = 34.730 - 0.011 \sin(4.100 + 13.369t)\ mm$
- $\gamma_s^3(t) = 0.074 \sin(1.841 - 13.369t)$
- $T = 0.47\, s$

According to these results, the variations of the minimum distance and number of contact points are shown in Figure 7.12. Furthermore, the variations of the motor torques are presented in Figure 7.13 and several snapshots of the motion are shown in Figure 7.14.

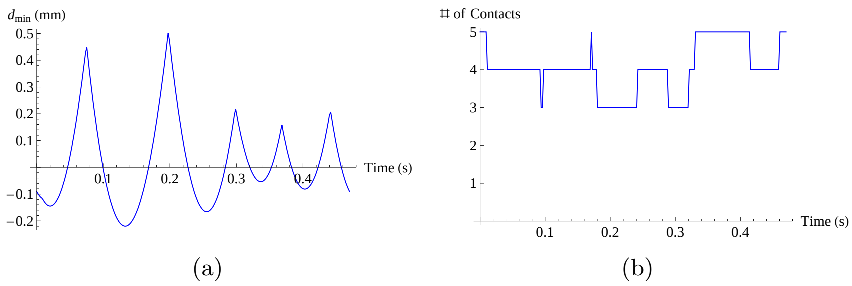
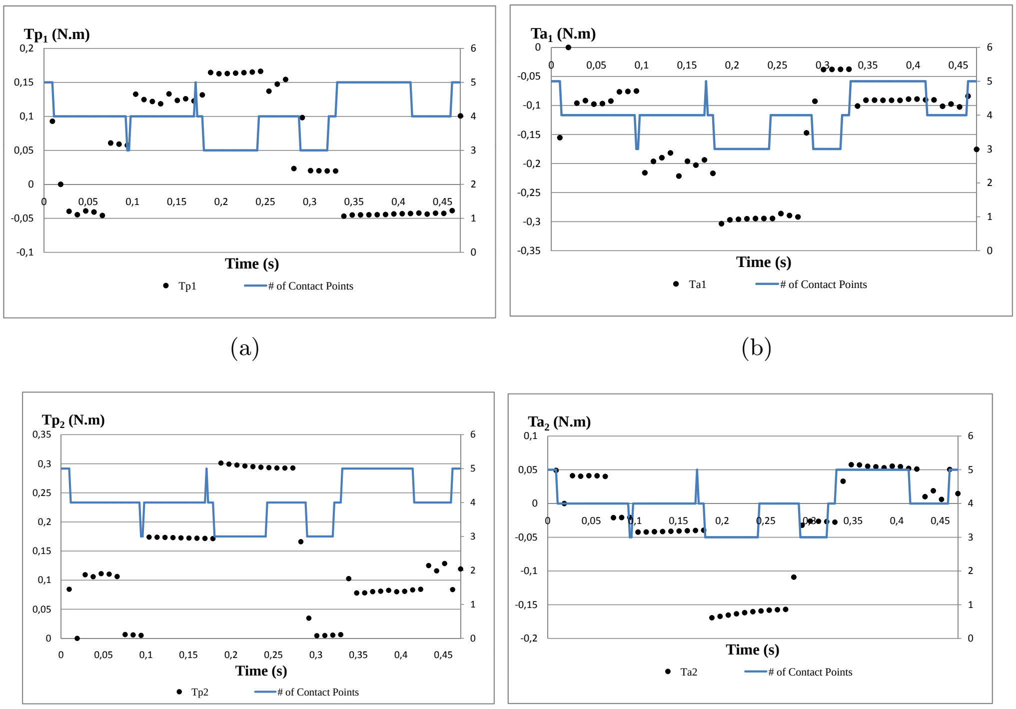
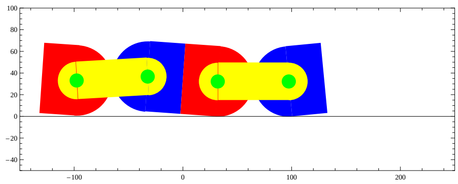


Figure 7.12: Minimum distance (d_{min}) and the number of contact points (♯ of Contacts) with respect to time for the sub-optimal solution

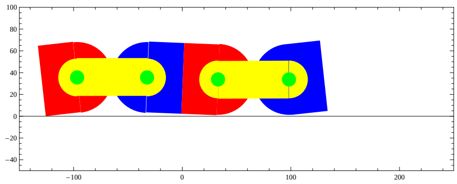


(a) (b) (c) (d)

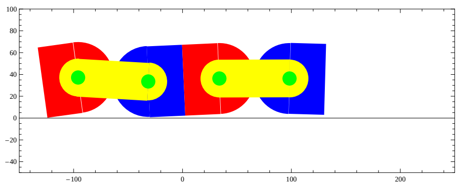
Figure 7.13: Variations of motor torques with respect to time for the sub-optimal solution



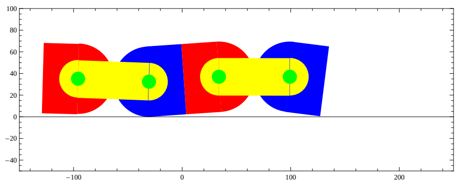
(a) $t = 0\,s$



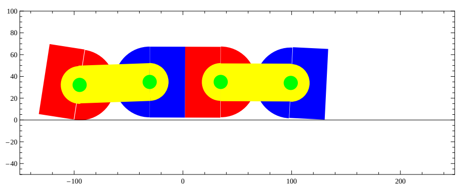
(b) $t = 0.094\,s$



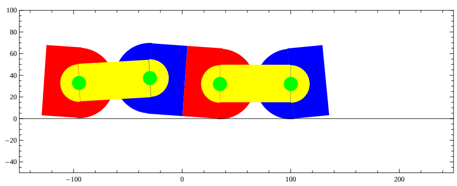
(c) $t = 0.188\,s$



(d) $t = 0.282\,s$



(e) $t = 0.376\,s$



(f) $t = 0.470\,s$

Figure 7.14: Some snapshots of the visual simulation of the motion of the system corresponding to the sub-optimal solution

CHAPTER 8

CONCLUSION

In this study, a novel algorithm is developed for the optimization of the locomotion of modular robots. The method is illustrated via a modular robot called MTRAN, however; it may be extended to other modular robots in the literature by adjusting the module parameters accordingly. The optimization procedure is separated into in four parts, namely, kinematic analysis, contact point determination, dynamic analysis and optimization.

In the kinematic analysis, the position, velocity and acceleration of the links in the modules are calculated in terms of the joint variables and the global position variables. The number of modules is not constant for a modular robot. Hence, the kinematic analysis is performed recursively. The recursive equations are derived for a 3D MTRAN system composed of n modules. However, in proceeding calculations, the locomotion of a 2D system is studied as a special case.

It is necessary to determine the contact points of the system and their coordinates in the global reference frame. In order to determine the contact points; the critical points, at which a contact between a link and the ground may occur, on the links are considered. The vertical distance of all the critical points in the system are calculated and the minimum value is taken to identify the status of the system with respect to the ground. If the minimum distance of the robot to the ground is less than a user defined tolerance, the robot is considered to be in contact with the ground.

In order to find the actuator torques corresponding to the specified system position, dynamic analysis of the system is performed. The normal and frictional reaction forces applied to the system at the contact points are included in the dynamic analysis. Unfortunately, the number of contact points and the locations of the contact points vary with respect to time. Hence, the number of equations and the number of unknowns in the dynamic analysis change between overconstrained, regular and underconstrained cases. In the regular case, the number of equations is equal to the number of unknowns. If the determinant of the coefficient matrix is non-zero, the solution is easily found. However, if the number of unknowns is higher than the number of equations, the system is underconstrained. In order to determine the unknowns, additional relations are required. These relations are obtained by considering the axial deflections of the middle links.

The final part of the optimization procedure is the selection of the objective function, design parameters and the constraints. The optimization problem is solved by using the NMinimize function of the Mathematica software. In the optimization, MD and CE are considered as the objective functions. MD expression guarantees that the robot is in contact with the ground. Furthermore, it also ensures that the contact points do not penetrate into the ground more than the user defined tolerance. Therefore, the normal velocities at the contact points cannot be negative. In short, the $MD = 0$ guarantees that the normal velocities at the contact points are non-negative so that the links do not hit to the ground. On the other hand, the CE expression is used for the minimization of the work done by the actuators.

In the case studies, different optimization problems with various objective functions and constraints are studied. First of all, the minimization of MD, subjected to only the physical constraints, is successfully performed via different optimization algorithms. However, the results are not sufficient for a stable locomotion because the number of contact points are not always greater than or equal to two. Hence, an additional constraint, called the number of contact points ratio (RC), is defined in order to ensure two or more contact points during the motion. Executing the optimization algorithm to minimize MD subjected to $RC = 0$ yields stable solutions, which have two or more contact points throughout the motion.

In order to perform dynamic analysis, on the other hand, the contact point velocities ratio, normal reaction forces ratio and friction force ratios are defined. These ratios must be equal to zero to obtain singularity-free and physically feasible solutions at the end of the dynamic analysis. In order to determine solutions which satisfy the $RC = RV = RN = RF = 0$ constraints, firstly, the solutions with $RC = 0$ are considered. By changing suitable design parameters (such as the horizontal velocity of the system, and the locomotion period) while keeping the remaining parameters to be the same, these constraints are satisfied. Then using the design parameters of the dynamically feasible solutions, minimization of the consumed energy is performed.

In conclusion, the following problems are solved in this thesis.

- Solution of position, velocity and acceleration analyses for a 3D modular robot
- Determination of the contact points of a modular robot on a flat surface
- The dynamic analysis for a 2D modular robot system
- Determination of the necessary conditions on the tangential velocities at the contact points necessary such that the equations in the dynamic analysis are solvable (i.e. singularity free)
- Solution of the underconstrained system of equations in dynamic analysis
- Determination of optimal locomotion patterns with respect to minimum energy consumption
- Visualization of the locomotion of modular robots

The algorithm can be further improved in the future by considering the following issues.

- Deformation model used for the flexible links may be extended to include continuous, rather than lumped, inertial effects.
- The objective function used in the solution of underconstrained case may be modified to reduce the sudden changes in the actuator torques.
- Optimization of the consumed energy can be studied via more efficient numerical optimization methods.
- Locomotion patterns corresponding to different periodic functions instead of the sine function used in this thesis, can be studied. Furthermore, locomotion for different horizontal velocity profiles can be studied.
- The algorithm can be extended to the 3D motion of modular robots.
- Locomotion in a non-flat environment can be studied.
- Optimal reconfiguration of modular robots can be studied and the locomotion and the reconfiguration algorithms can be combined in order to have a complete motion planning algorithm for modular robots.

- A graphical user interface can be developed for the kinematic and dynamic analysis of modular robots.

REFERENCES

[1] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. S. Chirikjian. Modular Self-Reconfigurable Robot Systems: Challenges and Opportunities for the Future. *IEEE Robotics and Automation Magazine*, March 2007.

[2] Oxford University Contributors. Askoxford: Brownian motion. Retrieved 12.07.2007, from <`http://www.askoxford.com/concise_oed/brownianmotion?view=uk`>.

[3] P. J. White, K. Kopanski, and H. Lipson. Stochastic Self-Reconfigurable Cellular Robotics. Proc. of *IEEE International Conference on Robotics and Automation*, pages 2888-2893, 2004.

[4] M. W. Jørgensen, E. H. Østergaard, and H. H. Lund. Modular ATRON: Modules for a self-reconfigurable robot. Proc. of *IEEE International Conference on Robotics and Automation*, pages 2068-2073, 2004.

[5] W.-M. Shen, B. Salemi, and P. Will. Hormones for self-reconfigurable robots. In *International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 918–925. IOS Press, 2000.

[6] K. Gilpin, K. Kotay, and D. Rus. Miche: Modular shape formation by self-disassembly. In *IEEE International Conference on Robotics and Automation*, pages 2241–2247, April 2007.

[7] V. Zykov, E. Mytilinaios, M. Desnoyer, and H. Lipson. Evolved and designed self-reproducing modular robotics. In *IEEE Transactions on Robotics*, volume 23, pages 308–319, April 2007.

[8] D. G. Duff, M. Yim, and K. Roufas. Evolution of polybot: A modular reconfigurable robot. Proc. of *the Harmonic Drive Intl. Symposium* 2001 and Proc. of *COE/Super-Mechano-Systems Workshop*, 2001.

[9] B. Kirby, J. Campbell, B. Aksak, P. Pillai, J. F. Hoburg, T. C. Mowry, and S. C. Goldstein. Catoms: Moving robots without moving parts. In *AAAI (Robot Exhibition)*, pages 1730–1, July 2005.

[10] M. Koseki, K. Minami, and N. Inou. Cellular robots forming a mechanical structure. Proc. of *Distributed Autonomous Robotic Systems*, 2004.

[11] D. Rus and M. Vona. Self-reconfiguration with compressible unit modules. In *Autonomous Robots*, volume 10, pages 107–124, January 2001.

[12] S. Murata, H. Kurokawa, and S. Kokaji. Self assembling machine. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 441–448, May 1994.

[13] C. Unsal and P. K. Khosla. Mechatronic design of a modular self-reconfiguring robotic system. Proc. of *IEEE International Conference on Robotics and Automation*, pages 1743-1747, 2000.

[14] G. S. Chirikjian. Kinematics of a metamorphic robotic system. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 449–455, May 1994.

[15] S.Murata, E. Yoshida, K. Tomita, H. Kurokawa, A. Kamimura, and S. Kokaji. Hardware design of modular robotic system. Proc. of *IEEE International Conference on Intelligent Robots and Systems*, 2000.

[16] N. Inou, H. Kobayashi, and M. Koseki. Development of pneumatic cellular robots forming a mechanical structure. Proc. of *Seventh International Conference on Control, Automation, Robotics And Vision*, pages 63-68, 2002.

[17] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Self-organizing programmable parts. In *International Conference on Intelligent Robots and Systems*. IEEE/RSJ Robotics and Automation Society, 2005.

[18] W.-M. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh. Multimode locomotion via SuperBot robots. In *International Conference on Robotics and Automation*, pages 2552–2557, 2006.

[19] J. W. Suh, S. B. Homans, and M. Yim. Telecubes: Mechanical design of a module for self-reconfigurable robotics. Proc. of *IEEE International Conference on Robotics and Automation*, pages 4095-4101, 2002.

[20] G. J. Hamlin and A. C. Sanderson. Tetrobot modular robotics: Prototype and experiments. Proc. of *IEEE International Conference on Intelligent Robots and Systems*, pages 390-395, 1996.

[21] K. Hosokawa, T. Tsujimori, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, and I. Endo. Self-organizing collective robots with morphogenesis in a vertical plane. Proc. of *IEEE International Conference on Robotics and Automation*, pages 2858-2863, 1998.

[22] K. D. Kotay and D. L. Rus. Algorithms for self-reconfiguring molecule motion planning. Proc. of *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2184-2193, 2000.

[23] K. C. Prevas, C. Ünsal, M. Ö. Efe, and P. K. Khosla. A hierarchical motion planning strategy for a uniform self-reconfigurable modular robotic system. Proc. of *IEEE International Conference on Robotics and Automation*, pages 787-792, 2002.

[24] E. Yoshida, S. Murata, A. Kamimura, K. Tomita, H. Kurokawa, and S. Kokaji. Evolutionary synthesis of dynamic motion and reconfiguration process for a modular robot M-TRAN. Proc. of *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 1004-1010, 2003.

[25] E. Mytilinaios, D. Marcus, M. Desnoyer, and H. Lipson. Designed and evolved blueprints for physical self-replicating machines. Proc. of *Ninth International Conference on Artificial Life*, pages 15-20, 2004.

[26] C. J. Chiang and G. S. Chirikjian. Connectivity planning for closed-chain reconfiguration. Proc. of *SPIE, Sensor Fusion and Decentralized Control in Robotic Systems III*, 2000.

[27] C. J. Chiang and G. S. Chirikjian. Modular robot motion planning using similarity metrics. Proc. of *Autonomous Robots 10*, pages 91-106, 2001.

[28] A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji. Automatic locomotion design and experiments for a modular robotic system. In *IEEE/ASME Transactions on Mechatronics*, volume 10, pages 314–325, June 2005.

[29] E. H. Østergaard and H. H. Lund. Evolving control for modular robotic units. Proc. of *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 886-892, 2003.

[30] D. Marbach and A. J. Ijspeert. Online optimization of modular robot locomotion. Proc. of *IEEE International Conference on Mechatronics and Automation*, pages 248-253, 2005.

[31] S. Murata, K. Kakomura, and H. Kurokawa. Docking experiments of a modular robot by visual feedback. Proc. of *IEEE International Conference on Intelligent Robots and Systems*, pages 625-630, 2006.

[32] H. Kurokawa. M-TRAN (modular transformer). Retrieved 15.07.2007, from <`http://unit.aist.go.jp/is/dsysd/mtran3/`>.

[33] H. Kurokawa, A. Kamimura, E. Yoshida, K. Tomita, S. Murata, and S. Kokaji. Self-reconfigurable modular robot (m-tran) and its motion design. Proc. of *Seventh International Conference on Control, Automation, Robotics and Vision*, pages 51-56, 2002.

[34] H. Kurokawa, E. Yoshida, K. Tomita, A. Kamimura, S. Murata, and S. Kokaji. Self-reconfigurable m-tran structures and walker generation. In *Robotics and Autonomous Systems*, volume 54, pages 142–149, 2006.

[35] M. K. Özgören. *ME 522 Principles of Robotics, Lecture Notes, METU*. 2005.

[36] R. Soylu. On the dynamic stability of bodies that are in planar contact with coulomb friction. In preparation.

[37] E. Yılmaz. Suboptimal control of dynamic systems using polynomial parametrization and symbolic manipulation. Master's thesis, Graduate School of Natural and Applied Sciences, Middle East Technical University, December 1997.

[38] Wolfram Research Center Contributors. *Mathematica Technical Documentation*, 2007.

[39] H. Baruh. *Analytical Dynamics*. WCB McGraw-Hill, 1999.

APPENDIX A

BASIC CALCULATION

A.1 Calculation of Angle Between Two Vectors

The angle measured from $\vec{a}$ to $\vec{b}$ around $\vec{c}$ can be calculated by means of dot and cross product operations. Note that, $\vec{c}$ is perpendicular to both of the vectors $\vec{a}$ and $\vec{b}$. Dot product of $\vec{a}$ and $\vec{b}$ can be written as

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}| \cos\theta \tag{A.1}$$

Here, θ is the angle between $\vec{a}$ and $\vec{b}$. The magnitude of the cross product vector of $\vec{a}$ and $\vec{b}$ is

$$|\vec{a} \times \vec{b}| = |\vec{a}||\vec{b}| \sin\theta \tag{A.2}$$

If $\vec{a}$ and $\vec{b}$ are unit vectors, their magnitudes are unity. Substituting $|\vec{a}| = |\vec{b}| = 1$ into Equations (A.1) and (A.2) yields

$$\vec{a} \cdot \vec{b} = \cos\theta \tag{A.3}$$

$$|\vec{a} \times \vec{b}| = \sin\theta \tag{A.4}$$

Therefore, θ can be found to be

$$\theta = \arctan(\cos\theta, \sin\theta) = \arctan\left(\vec{a} \cdot \vec{b}, |\vec{a} \times \vec{b}|\right) \tag{A.5}$$

There exists two solutions, namely, $\theta_1 = \theta$ and $\theta_2 = 2\pi - \theta$ for the angle measured from $\vec{a}$ to $\vec{b}$. The solution is the angle measured around $\vec{c}$ in a right hand sense. Therefore, the direction of the cross product vector and $\vec{c}$ is compared. If they are in the same direction, $\theta_1 = \theta$ is taken as the result. If they are in opposite directions, $\theta_2 = 2\pi - \theta$ is the angle measured around $\vec{c}$ in a right hand sense.

A.2 Calculation of Mass Moment of Inertias

The mass moment of inertias of the passive and active links can be calculated by separating each link into two parts, namely, a semi-cylinder and rectangular prism as shown in Figure A.1. The mass moment of inertias of the passive and active links are assumed to be the same since they share the same geometry. Therefore, calculation of the mass moment of inertia of a passive link is sufficient.

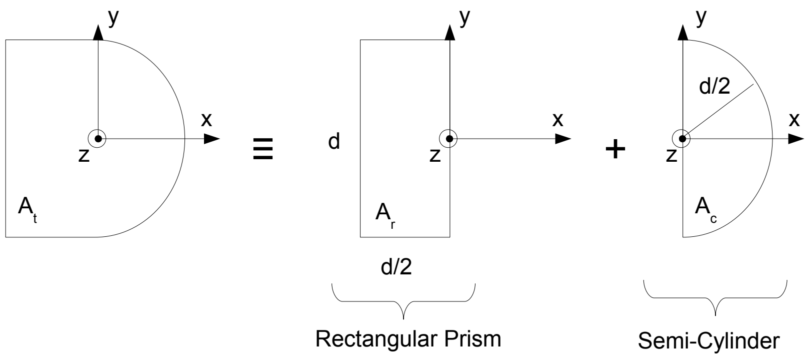


Figure A.1: Separation of the link shape into semi-cylinder and rectangular prism

The masses of the semi-cylinder and the rectangular prism are not the same. Since the densities are the same, these masses are proportional to their cross-sectional areas. The cross-sectional area of the rectangular prism, in terms of the unit length d, is

$$A_r = \frac{d^2}{2} \tag{A.6}$$

whereas, the cross-sectional area of the semi-cylinder is given by

$$A_c = \frac{d^2}{4}\frac{\pi}{2} = \frac{\pi d^2}{8} \tag{A.7}$$

Hence, the total cross-sectional area of the passive link becomes

$$A_t = \frac{d^2}{2} + \frac{\pi d^2}{8} = d^2\left(\frac{1}{2} + \frac{\pi}{8}\right) = d^2\left(\frac{\pi + 4}{8}\right) \tag{A.8}$$

The masses of the semi-cylinder and the rectangular prism can be calculated as

$$m_r = \frac{A_r}{A_t} m_p \tag{A.9}$$

$$m_c = \frac{A_c}{A_t} m_p \tag{A.10}$$

Substituting A_r and A_c from Equations (A.6) and (A.7), one obtains

$$m_r = \frac{4}{\pi + 4} m_p \tag{A.11}$$

$$m_c = \frac{\pi}{\pi + 4} m_p \tag{A.12}$$

Now, the mass moment of inertias of the semi-cylinder and the rectangular prism can be calculated with respect to the reference frame shown in Figure A.1. The mass moment of inertia expressions for semi-cylinder and rectangular prism are shown in Figure A.2.

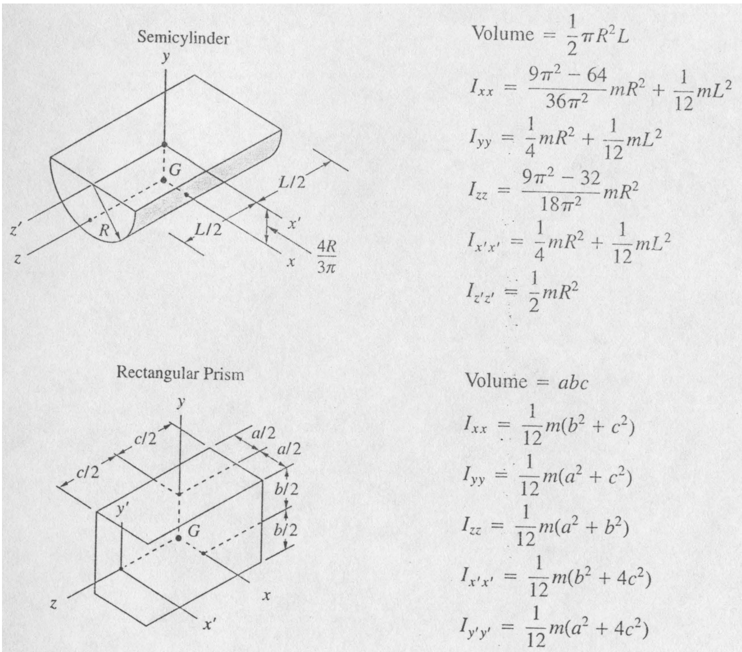


Figure A.2: Mass moment of inertias of semi-cylinder and rectangular prism [39]

Using these expressions, the mass moment of inertia of the semi-cylinder about the z axis can be written as

$$I = \frac{1}{2}mr^2 \quad \Rightarrow \quad I_c = \frac{1}{2}m_c\left(\frac{d}{2}\right)^2 = \frac{m_c d^2}{8} \tag{A.13}$$

Similarly, the mass moment of inertia of the rectangular prism is

$$I = \frac{1}{12}m\left(a^2 + 4b^2\right) \quad \Rightarrow \quad I_r = \frac{1}{12}m_r\left(d^2 + 4\left(\frac{d}{2}\right)^2\right) = \frac{m_r d^2}{6} \tag{A.14}$$

Thus, the mass moment of inertia of the passive link becomes

$$I_p = I_r + I_c = \frac{m_r d^2}{6} + \frac{m_c d^2}{8} \tag{A.15}$$

Substituting the expressions for m_c and m_r in Equations (A.12) and (A.12), one obtains

$$I_p = \frac{m_p d^2\left(3\pi + 16\right)}{24\left(\pi + 4\right)} \tag{A.16}$$

For $d = 65mm$ and $m_p = 140g$, the mass moment of inertia of the passive link becomes $I_p = 8.774 \times 10^4\, g.mm^2 = 8.774 \times 10^{-5} kg.m^2$. Since the mass moment of inertias are assumed to be the same for the passive and active links, $I_a = I_p = 8.774 \times 10^{-5} kg.m^2$. Similarly, calculating the mass moment of inertia of the middle link, one obtains $I_l = 8.580 \times 10^{-5} kg.m^2$

APPENDIX B

ZERO DETERMINANT CONDITIONS

In this appendix, the conditions, which make the determinant of the coefficient matrix of dynamic analysis equations are presented by means of MATHEMATICA outputs.

B.1 Notation

Configuration of the contact points in a modular robot is specified via the following notation.

$$\text{Contact Point Data} = \{\underbrace{\{N_{mod}, N_{part}, \{\overbrace{\{x1, y1\}, \{x2, y2\}}^{\text{Coordinates}}\}}_{\text{Data of Single Module}}\}$$

where, N_{mod} is the module number, N_{part} is the part number (1: passive, 2: active) and $x1, y1$ are the coordinates of the contact point in global coordinate system. Following cases are presented as examples.

- ContactData $= \{\{1, 1, \{\{x1, y1\}\}\}, \{2, 2, \{\{x2, y2\}\}\}\}$ represents two contact points. One of them is on the passive part of the 1^{st} module with coordinates $(x1, y1)$ and the other one is on the active part of the 2^{nd} module with coordinates $(x2, y2)$.
- ContactData $= \{\{1, 1, \{\{x1, y1\}, \{x2, y2\}\}\}\}$ represents line contact on the passive part of the 1^{st} module. The coordinates of the endpoints of the contact line segment is $(x1, y1)$ and $(x2, y2)$.

The tangential velocity of a contact point defines whether the static or dynamic Coulomb friction model is used. Hence, different velocity conditions for the same contact point configuration are considered. In order to define non-zero tangential velocity V_i symbol is used. In this expression i indicates the contact point number. On the other hand, if there is a line contact, the tangential velocities at both ends of the line segment are taken as equal.

B.2 Program Outputs

In the following pages, Mathematica output for different contact point and corresponding tangential velocity configurations are presented.

- **Two Contact Points**

```
Contact Point Data : {{1, 1, {{x1, y1}}}, {1, 2, {{x2, y2}}}}
Tangential contact point velocities : {V1, 0}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V1] == x2
Tangential contact point velocities : {0, V2}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V2] == x2

Contact Point Data : {{1, 1, {{x1, y1}}}, {2, 1, {{x2, y2}}}}
Tangential contact point velocities : {V1, 0}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V1] == x2
Tangential contact point velocities : {0, V2}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V2] == x2

Contact Point Data : {{1, 1, {{x1, y1}}}, {2, 2, {{x2, y2}}}}
Tangential contact point velocities : {V1, 0}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V1] == x2
Tangential contact point velocities : {0, V2}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V2] == x2

Contact Point Data : {{1, 2, {{x1, y1}}}, {2, 2, {{x2, y2}}}}
Tangential contact point velocities : {V1, 0}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V1] == x2
Tangential contact point velocities : {0, V2}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V2] == x2

Contact Point Data : {{2, 1, {{x1, y1}}}, {2, 2, {{x2, y2}}}}
Tangential contact point velocities : {V1, 0}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V1] == x2
Tangential contact point velocities : {0, V2}
Condition of zero determinant for coefficient matrix : x1 + (y1 - y2) μ Sign[V2] == x2

Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}}
Tangential contact point velocities : {0, 0}
Condition of zero determinant for coefficient matrix : x1 == x2

Contact Point Data : {{1, 2, {{x1, y1}}}, {2, 1, {{x2, y2}}}}
Tangential contact point velocities : {0, 0}
Condition of zero determinant for coefficient matrix : x1 == x2

Contact Point Data : {{2, 2, {{x1, y1}, {x2, y2}}}}
Tangential contact point velocities : {0, 0}
Condition of zero determinant for coefficient matrix : x1 == x2
```

Figure B.1: Mathematica output for different contact point and velocity configurations

Three Contact Points

```
Contact Point Data : {{1, 1, {{x1, y1}}}, {1, 2, {{x2, y2}}}, {2, 2, {{x3, y3}}}}
Tangential contact point velocities : {V1, V2, V3}
Condition of zero determinant for coefficient matrix :
 (Sign[V2] == Sign[V3] && (Sign[V1] == Sign[V3] || x2 + (y2 - y3) μ Sign[V3] == x3)) ||
 (1/(Sign[V2] - Sign[V3])
    (-x2 Sign[V3] + Sign[V1] (x2 - x3 + (-y1 + y2) μ Sign[V2] + (y1 - y3) μ Sign[V3]) +
      Sign[V2] (x3 + (-y2 + y3) μ Sign[V3])) == x1 && Sign[V2] ≠ Sign[V3]) || μ == 0
Tangential contact point velocities : {V1, V2, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3
Tangential contact point velocities : {V1, 0, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V3] == x3
Tangential contact point velocities : {0, V2, V3}
Condition of zero determinant for coefficient matrix : (Sign[V2] == 0 && Sign[V3] == 0) ||
 ((-x2 Sign[V3] + Sign[V2] (x3 + (-y2 + y3) μ Sign[V3]))/(Sign[V2] - Sign[V3]) == x1 && Sign[V2] ≠ Sign[V3]) ||
 (Sign[V2] == Sign[V3] && x2 + (y2 - y3) μ Sign[V3] == x3 && Sign[V3] ≠ 0) || μ == 0
Tangential contact point velocities : {V1, 0, 0}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, V2, 0}
Condition of zero determinant for coefficient matrix : x1 == x3 || μ == 0 || Sign[V2] == 0
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 1, {{x1, y1}}}, {2, 1, {{x2, y2}}}, {2, 2, {{x3, y3}}}}
Tangential contact point velocities : {V1, V2, V3}
Condition of zero determinant for coefficient matrix :
 (Sign[V2] == Sign[V3] && (Sign[V1] == Sign[V3] || x2 + (y2 - y3) μ Sign[V3] == x3)) ||
 (1/(Sign[V2] - Sign[V3])
    (-x2 Sign[V3] + Sign[V1] (x2 - x3 + (-y1 + y2) μ Sign[V2] + (y1 - y3) μ Sign[V3]) +
      Sign[V2] (x3 + (-y2 + y3) μ Sign[V3])) == x1 && Sign[V2] ≠ Sign[V3]) || μ == 0
Tangential contact point velocities : {V1, V2, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3
Tangential contact point velocities : {V1, 0, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V3] == x3
Tangential contact point velocities : {0, V2, V3}
```

Figure B.2: Mathematica output for different contact point and velocity configurations continued

```
Condition of zero determinant for coefficient matrix : (Sign[V2] == 0 && Sign[V3] == 0) ||
  ((-x2 Sign[V3] + Sign[V2] (x3 + (-y2 + y3) μ Sign[V3])) / (Sign[V2] - Sign[V3]) == x1 && Sign[V2] ≠ Sign[V3]) ||
  (Sign[V2] == Sign[V3] && x2 + (y2 - y3) μ Sign[V3] == x3 && Sign[V3] ≠ 0) || μ == 0
Tangential contact point velocities : {V1, 0, 0}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, V2, 0}
Condition of zero determinant for coefficient matrix : x1 == x3 || μ == 0 || Sign[V2] == 0
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {1, 2, {{x3, y3}}}}
Tangential contact point velocities : {V1, V1, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || x1 + (y1 - y2) μ Sign[V1] == x2 || Sign[V1] == Sign[V3]
Tangential contact point velocities : {V1, V1, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V1] == x3
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {2, 1, {{x3, y3}}}}
Tangential contact point velocities : {V1, V1, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || x1 + (y1 - y2) μ Sign[V1] == x2 || Sign[V1] == Sign[V3]
Tangential contact point velocities : {V1, V1, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V1] == x3
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {2, 2, {{x3, y3}}}}
Tangential contact point velocities : {V1, V1, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || x1 + (y1 - y2) μ Sign[V1] == x2 || Sign[V1] == Sign[V3]
Tangential contact point velocities : {V1, V1, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V1] == x3
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 2, {{x2, y2}}}, {2, 1, {{x3, y3}}}, {1, 1, {{x1, y1}}}}
Tangential contact point velocities : {V1, V1, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || x2 + (y2 - y3) μ Sign[V1] == x3 || Sign[V1] == Sign[V3]
Tangential contact point velocities : {V1, V1, 0}
```

Figure B.3: Mathematica output for different contact point and velocity configurations continued

```
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x1 + (y1 - y3) μ Sign[V1] == x3
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 2, {{x1, y1}}}, {2, 1, {{x2, y2}}}, {2, 2, {{x3, y3}}}}
Tangential contact point velocities : {V1, V1, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || x1 + (y1 - y2) μ Sign[V1] == x2 || Sign[V1] == Sign[V3]
Tangential contact point velocities : {V1, V1, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V1] == x3
Tangential contact point velocities : {0, 0, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0

Contact Point Data : {{1, 1, {{x1, y1}}}, {2, 2, {{x2, y2}, {x3, y3}}}}
Tangential contact point velocities : {V1, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == Sign[V2] || x2 + (y2 - y3) μ Sign[V2] == x3
Tangential contact point velocities : {V1, 0, 0}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V2] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3

Contact Point Data : {{1, 2, {{x1, y1}}}, {2, 2, {{x2, y2}, {x3, y3}}}}
Tangential contact point velocities : {V1, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == Sign[V2] || x2 + (y2 - y3) μ Sign[V2] == x3
Tangential contact point velocities : {V1, 0, 0}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V2] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3

Contact Point Data : {{2, 1, {{x1, y1}}}, {2, 2, {{x2, y2}, {x3, y3}}}}
Tangential contact point velocities : {V1, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == Sign[V2] || x2 + (y2 - y3) μ Sign[V2] == x3
Tangential contact point velocities : {V1, 0, 0}
Condition of zero determinant for coefficient matrix : x2 == x3 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, V2, V2}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V2] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3
```

Figure B.4: Mathematica output for different contact point and velocity configurations continued

Four Contact Points

```
Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {1, 2, {{x3, y3}}}, {2, 2, {{x4, y4}}}}
Tangential contact point velocities : {V1, V1, V3, V4}
Condition of zero determinant for coefficient matrix :
 (Sign[V3] == Sign[V4] && (Sign[V1] == Sign[V4] || x3 + (y3 - y4) μ Sign[V4] == x4)) ||
 (            1
  ─────────────────────
  Sign[V3] - Sign[V4]
    (-x3 Sign[V4] + Sign[V1] (x3 - x4 + (-y2 + y3) μ Sign[V3] + (y2 - y4) μ Sign[V4]) + Sign[V3]
       (x4 + (-y3 + y4) μ Sign[V4])) == x2 && Sign[V3] ≠ Sign[V4]) || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {V1, V1, V3, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {V1, V1, 0, V4}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x3 + (y3 - y4) μ Sign[V4] == x4
Tangential contact point velocities : {0, 0, V3, V4}
Condition of zero determinant for coefficient matrix :
 x1 == x2 || μ == 0 || Sign[V3] == 0 || Sign[V4] == 0
Tangential contact point velocities : {0, 0, 0, V4}
Condition of zero determinant for coefficient matrix : True
Tangential contact point velocities : {0, 0, V3, 0}
Condition of zero determinant for coefficient matrix : True
Tangential contact point velocities : {V1, V1, 0, 0}
Condition of zero determinant for coefficient matrix : x3 == x4 || μ == 0 || Sign[V1] == 0
Contact Point Data : {{1, 1, {{x1, y1}}}, {1, 2, {{x2, y2}}}, {2, 2, {{x3, y3}, {x4, y4}}}}
Tangential contact point velocities : {V1, V2, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V2] == Sign[V3] || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {V1, V2, 0, 0}
Condition of zero determinant for coefficient matrix :
 x3 == x4 || μ == 0 || Sign[V1] == 0 || Sign[V2] == 0
Tangential contact point velocities : {V1, 0, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V3] == 0 || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {0, V2, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V2] == 0 || Sign[V3] == 0 || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {0, 0, V3, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0
Tangential contact point velocities : {V1, V2, 0, 0}
Condition of zero determinant for coefficient matrix :
 x3 == x4 || μ == 0 || Sign[V1] == 0 || Sign[V2] == 0
Contact Point Data :
 {{1, 1, {{x1, y1}}}, {1, 2, {{x2, y2}}}, {2, 1, {{x3, y3}}}, {2, 2, {{x4, y4}}}}
```

Figure B.5: Mathematica output for different contact point and velocity configurations continued

```
Tangential contact point velocities : {V1, V2, V2, V4}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || x2 + (y2 - y3) μ Sign[V2] == x3 || Sign[V2] == Sign[V4]
Tangential contact point velocities : {V1, V2, V2, 0}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V2] == 0 || x3 + (y3 - y4) μ Sign[V2] == x4
Tangential contact point velocities : {V1, 0, 0, V4}
Condition of zero determinant for coefficient matrix :
 x2 == x3 || μ == 0 || Sign[V1] == 0 || Sign[V4] == 0
Tangential contact point velocities : {0, V2, V2, V4}
Condition of zero determinant for coefficient matrix :
```

$$\left(\frac{-x3\,\mathrm{Sign}[V4] + \mathrm{Sign}[V2]\,(x4 + (-y3 + y4)\,\mu\,\mathrm{Sign}[V4])}{\mathrm{Sign}[V2] - \mathrm{Sign}[V4]} == x1\ \&\&\ \mathrm{Sign}[V2] \neq \mathrm{Sign}[V4]\right) \ ||$$

```
  (Sign[V2] == Sign[V4] && x3 + (y3 - y4) μ Sign[V4] == x4 && Sign[V4] ≠ 0) || μ == 0 || Sign[V2] == 0
Tangential contact point velocities : {0, V2, V2, 0}
Condition of zero determinant for coefficient matrix : x1 == x4 || μ == 0 || Sign[V2] == 0
Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {1, 2, {{x3, y3}}}, {2, 1, {{x4, y4}}}}
Tangential contact point velocities : {V1, V1, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V1] == Sign[V3] || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {V1, V1, 0, 0}
Condition of zero determinant for coefficient matrix : x3 == x4 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, 0, V3, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0
Contact Point Data : {{1, 1, {{x1, y1}, {x2, y2}}}, {2, 2, {{x3, y3}, {x4, y4}}}}
Tangential contact point velocities : {V1, V1, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V1] == Sign[V3] || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {V1, V1, 0, 0}
Condition of zero determinant for coefficient matrix : x3 == x4 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, 0, V3, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0
Contact Point Data : {{2, 2, {{x1, y1}, {x2, y2}}}, {1, 2, {{x3, y3}}}, {2, 1, {{x4, y4}}}}
Tangential contact point velocities : {V1, V1, V3, V3}
Condition of zero determinant for coefficient matrix :
 μ == 0 || Sign[V1] == 0 || Sign[V1] == Sign[V3] || x3 + (y3 - y4) μ Sign[V3] == x4
Tangential contact point velocities : {V1, V1, 0, 0}
Condition of zero determinant for coefficient matrix : x3 == x4 || μ == 0 || Sign[V1] == 0
Tangential contact point velocities : {0, 0, V3, V3}
Condition of zero determinant for coefficient matrix : x1 == x2 || μ == 0 || Sign[V3] == 0
```

Figure B.6: Mathematica output for different contact point and velocity configurations continued

APPENDIX C

MECHANISM MODELS

The modeling of the different contact point configurations by basic mechanisms are presented in the following tables for a two module system. In these tables, P indicates point contact, whereas L indicates line contact.

Table C.1: Models for two contact points

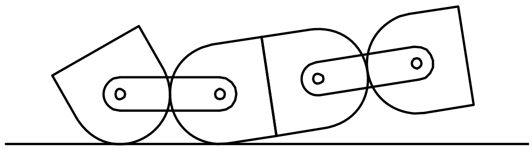
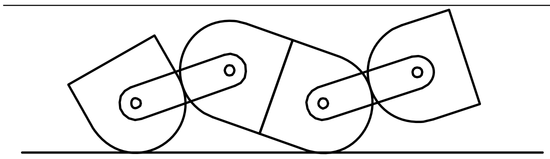
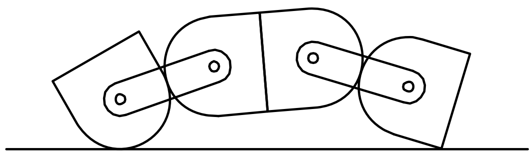
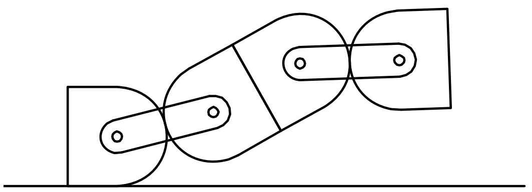
Configuration	Type	Model
	2P	1x Four bar
	2P	1x Four bar
	2P	1x 6-link
	1L	Not Applicable

Table C.2: Models for three contact points

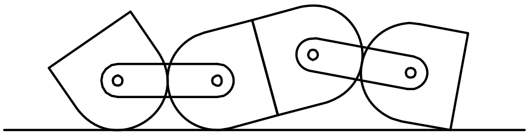
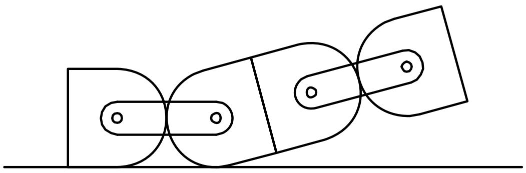
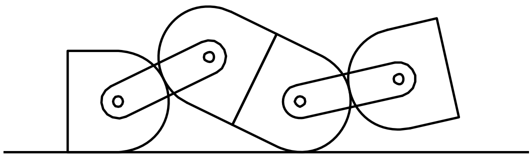
Configuration	Type	Model
	3P	2x Four bar
	L+P	1x Slider-crank
	L+P	1x Slider-crank

Table C.3: Models for four contact points

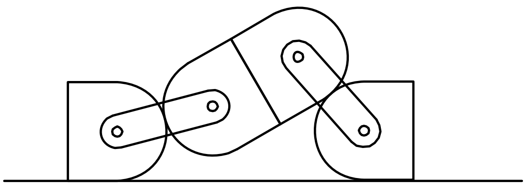
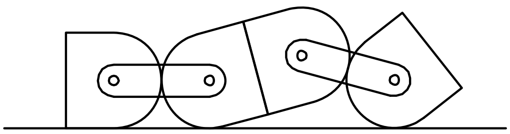
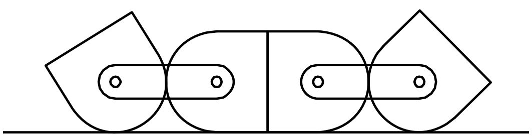
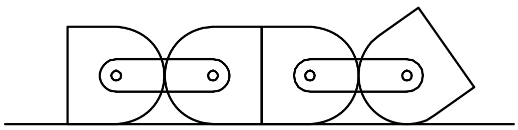
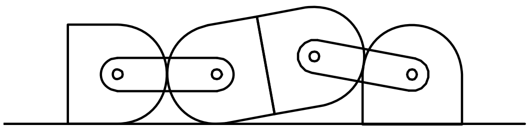
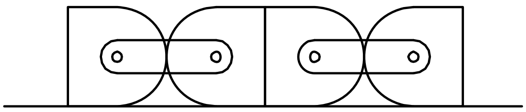
Configuration	Type	Model
	2P	1x Four bar
	L+P	Slider-crank + Four bar
	2L	2x Slider-crank

Table C.4: Models for five and six contact points

Configuration	Type	Model
	2L+P	Double slider + slider crank
	2L+P	2x Slider crank
	3L	2x Double slider

APPENDIX D

VISUAL SIMULATION

In order to visualize the locomotion pattern, a visual simulation algorithm is developed. In this appendix, the details of the simulation algorithm are discussed and various snapshots of the simulations are presented.

The visual simulation is developed by means of the MATHEMATICA software. In order to model the links, the module design of the MTRAN system is considered. The developed simulation algorithm is able to perform both 2D and 3D simulations. In the 2D visualization all links are modeled by means of rectangles and circles. However, in the 3D visualization, the links are modeled via 3D geometric objects (like rectangular prisms and cylinders). 2D simulation is mainly used for the examination of the motion of the links with respect to each other during the system locomotion. On the contrary, it is not possible to investigate the relative motion of the middle links (with respect to the passive and active links) in the 3D simulation. This is because, the motion of the middle links cannot be distinguished clearly. Hence, the 3D simulation is solely used for the demonstration of the motion in a working environment.

The basic principal of the simulation is plotting the positions of all links at different time instants through the discretization of time. Firstly, the required time range for the simulation is divided into a number of equally spaced segments. Secondly, the physical shapes of the links are modeled and their positions and orientations are plotted for each point of the discretized time range. Finally, the plots are played in sequence by means of the LISTANIMATE function of the MATHEMATICA software. The flowchart of the simulation algorithm is presented in Figure D.1.

In order to perform visual simulation of an n module MTRAN system, the following parameters are needed as inputs.

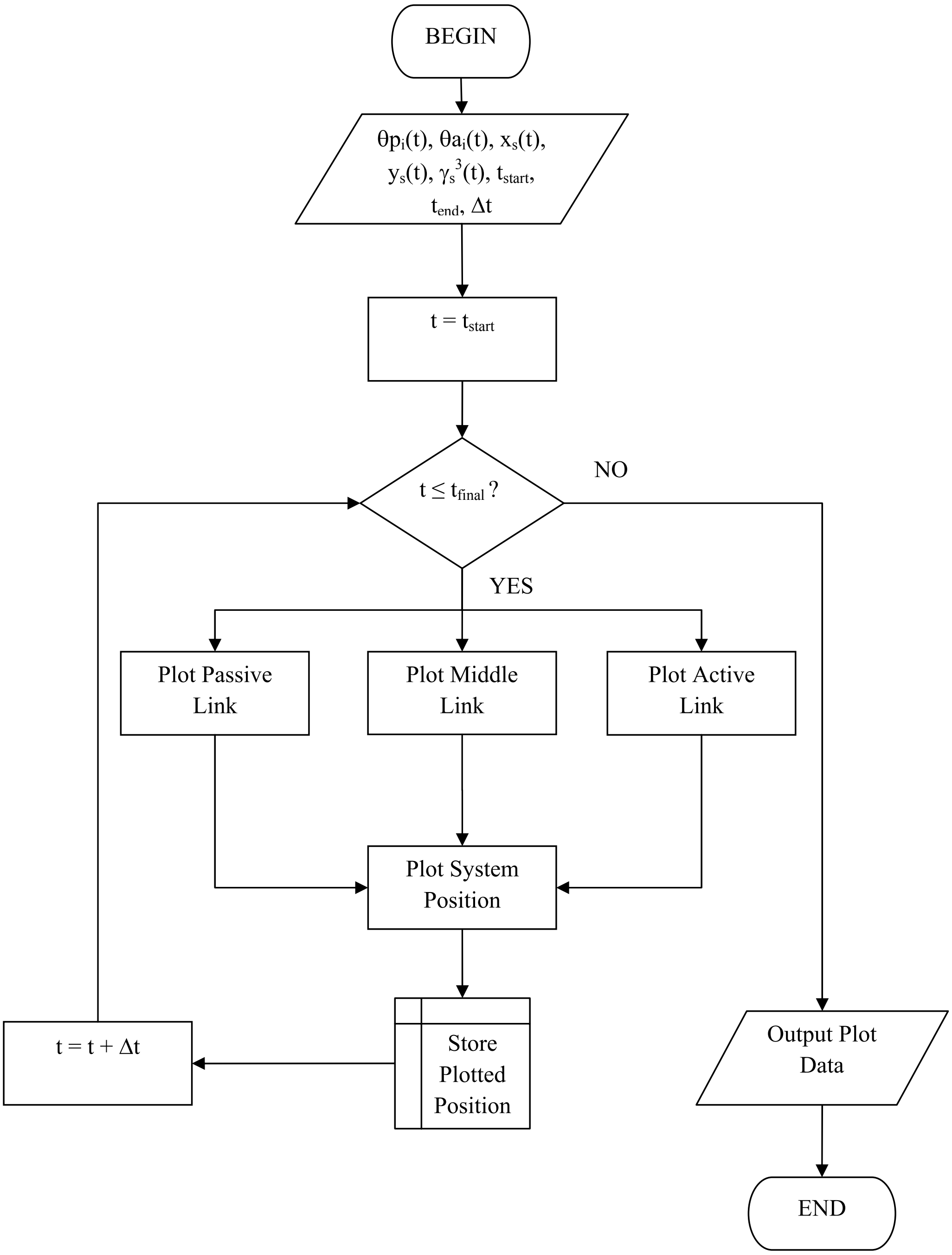


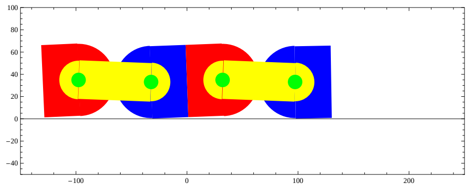
Figure D.1: Flowchart of the visual simulation algorithm

- Actuator rotation variables $(\theta p_i(t), \theta a_i(t))$
- Global position variables $(x_s(t), y_s(t), z_s(t), \gamma_s^1, \gamma_s^2$ and γ_s^3
- Start and stop times of the simulation
- Time increment for the discretization of the time range of the simulation

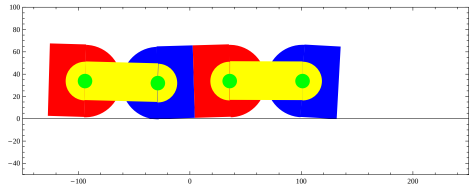
Consider the solution, which is calculated in the case studies which leads to the locomotion parameters given below.

- $\theta p_1(t) = 0.088 \sin(4.363 - 14.758t)$
- $\theta a_1(t) = \pi + 0.083 \sin(1.975 - 14.758t)$
- $\theta p_2(t) = 0.077 \sin(4.815 - 14.758t)$
- $\theta a_2(t) = \pi + 0.092 \sin(2.489 + 14.758t)$
- $x_s(t) = 40.992t\, mm$
- $y_s(t) = 33.790 - 0.580 \sin(3.5270 - 14.758t)\; mm$
- $\gamma_s^3(t) = 0.0421 \sin(1.999 - 14.758t)$
- $T = 0.426\, s$

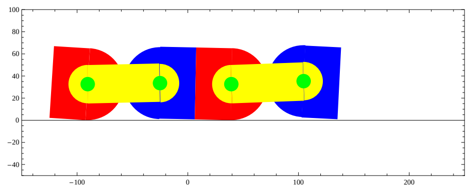
The snapshots of the 2D visual simulation for this solution are presented in Figure D.2. As the snapshots indicate, the motion of the middle links are clearly distinguishable. However, it is not possible to show the locomotion at different perspectives in the 2D simulation. By means of the 3D visual simulation, the motion of the system can be seen at any perspective (i.e., front view, top view, isometric view, etc,...). Executing the 3D visual simulation algorithm according to the given locomotion data, the snapshots in Figure D.3 are obtained. Furthermore, the snapshots displayed at different perspectives are presented in Figure D.4.



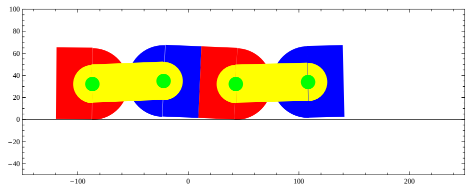
(a) $t = 0\,s$



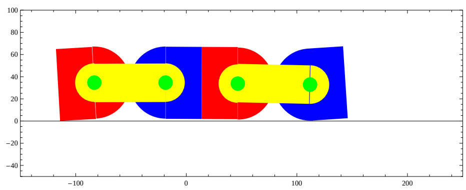
(b) $t = 0.085\,s$



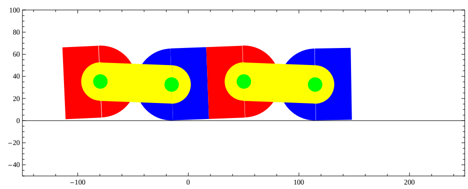
(c) $t = 0.170\,s$



(d) $t = 0.255\,s$

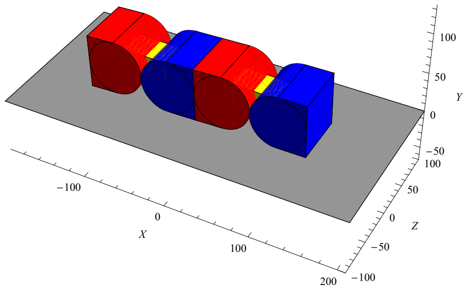


(e) $t = 0.341\,s$

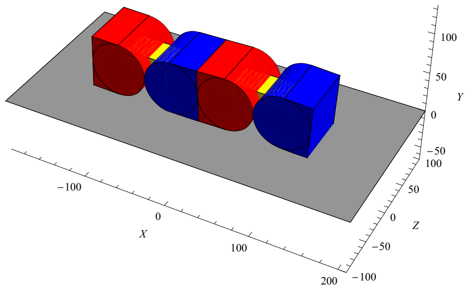


(f) $t = 0.426\,s$

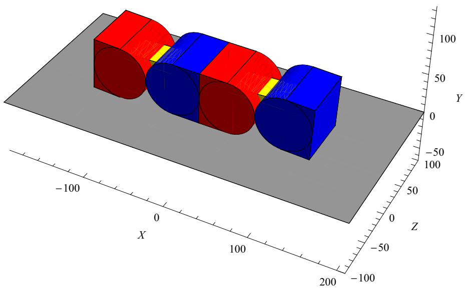
Figure D.2: Some snapshots of the visual simulation of the system locomotion from 2D visual simulation



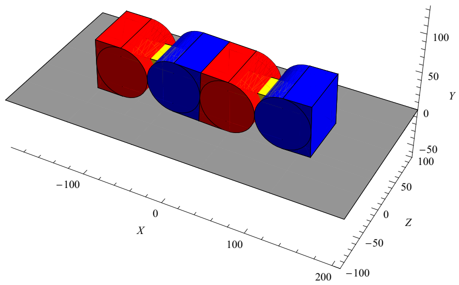
(a) $t = 0\,s$



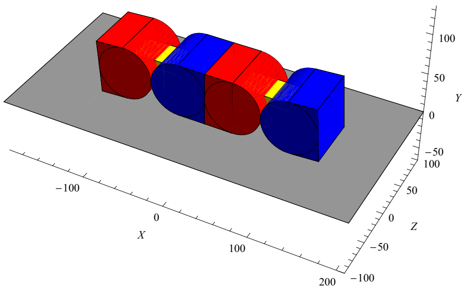
(b) $t = 0.085\,s$



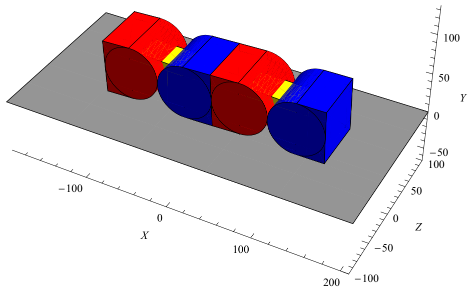
(c) $t = 0.170\,s$



(d) $t = 0.255\,s$

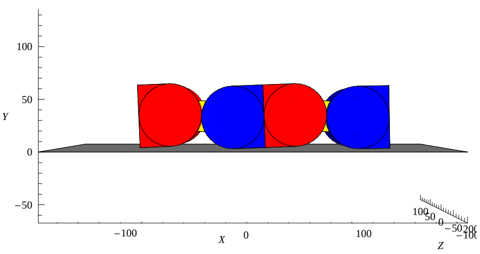


(e) $t = 0.341\,s$

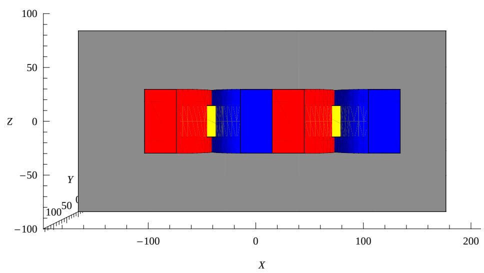


(f) $t = 0.426\,s$

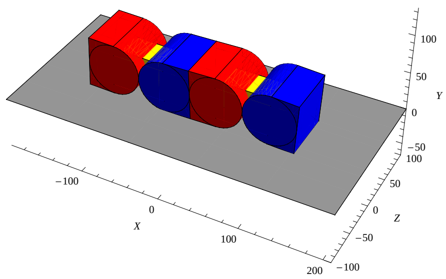
Figure D.3: Some snapshots of the visual simulation of the system locomotion from 3D visual simulation



(a) Front view



(b) Top view



(c) Isometric view

Figure D.4: Some snapshots of the visual simulation of the system locomotion from 3D visual simulation at different view angles