Self-Reconfigurable Transformer Robot

Project report submitted to
Visvesvaraya National Institute of Technology, Nagpur
in partial fulfillment of the requirements for the award of the degree

Bachelor of Technology
in
Mechanical Engineering

by

Sapan Santosh Agrawal **(BT14MEC003)**

Vinit Chunilal Sarode **(BT14MEC093)**

under the guidance of

Dr. Shital Chiddarwar

Department of Mechanical Engineering
Visvesvaraya National Institute of Technology
Nagpur 440 010 (India)

April 2018

Department of Mechanical Engineering

Visvesvaraya National Institute of Technology, Nagpur

Declaration

We, **Sapan Santosh Agrawal** and **Vinit Chunilal Sarode**, hereby declare that this project work titled "**Self-Reconfigurable Transformer Robot**" is carried out by us in the Department of Mechanical Engineering of Visvesvaraya National Institute of Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Sapan Santosh Agrawal

Vinit Chunilal Sarode

Date:

Certificate

This to certify that the project titled "**Self-Reconfigurable Transformer Robot**", submitted by **Sapan Santosh Agrawal** and **Vinit Chunilal Sarode** in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Mechanical Engineering, VNIT Nagpur. The work is comprehensive, complete and fit for final evaluation.

Dr. Shital Chiddarwar,

Associate Professor,

Dept. of Mechanical Engineering,

VNIT, Nagpur

Head, Department of Mechanical Engineering

VNIT, Nagpur

Date:

Acknowledgement

With immense pleasure and great respect, we express our deep sense of gratitude to Dr. Shital Chiddarwar of Department of Mechanical Engineering VNIT, Nagpur for her invaluable guidance, inspiration and constant encouragement throughout the project work.

We also express our sincere gratitude towards Rohan Thakker, Alumni VNIT, Nagpur for generous guidance and motivation from the beginning to the completion of the project.

Last but not the least, we express our thanks to our parents, friends Shalini Addepalli, , Rajeshree Deotalu, Yogesh Phalak and all members of IvLabs, Robotics Club of VNIT for their constant support which helped us complete this work.

ABSTRACT

Operations in search and rescue requires self-reconfigurable robotic systems to overcome unforeseen locomotion hindrances. Hence, an individual robotic system excellent in accomplishing a specific task fails to complete other operations. Hence, such unpredictable and unstructured scenarios urge the deployment of modular system which could reconfigure itself in accordance with the task at hand. In this thesis, we present a modular self-reconfigurable robotic system which extends the traversing versatility of snake robots to legged locomotion. The robotic system can easily rearrange to various linear, quadrupedal and biped morphologies through magnetic connections as per circumstances in situ. This thesis discusses the design aspect of the snake robot, localization of robots, transformations and locomotion gaits of various morphologies.

s

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ACM – Active cord Mechanism

API – Application Program Interface

BSD – Berkeley Software Distribution

BLE – Bluetooth Low Energy

CMU – Carnegie Mellon University

CoG – Centre of Gravity

CoM – Centre of Mass

CSAIL - Computer Science & Artificial Intelligence Lab

DARPA – Defense Advanced Research Projects Agency

DoF – Degrees of freedom

DH – Denavit Hartenberg

GPIO – General Purpose Input/ Output

GPS – Global Positioning System

HSV – Hue Saturation Value

I2C – Inter-Integrated Circuit

LAN – Local Area Network

LiDAR - Light Detection and Ranging

LIP – Linear Inverted Pendulum

LISP – List Processing

MIT – Massachusetts Institute of Technology

PC – Personal Computer

PWM – Pulse Width Modulation

RAM – Random Access Memory

RGB – Red Blue Green

nD – n-Dimension

ROS – Robotic Operating System

SLAM – Simulteneous Localization and Mapping

UART- universal asynchronous receiver-transmitter

USB – Universal Serial Bus

ZMP – Zero Moment Point

INDEX

1. INTRODUCTION

First responder teams in urban search and rescue or disaster scenarios could benefit from a rapidly deployable robot with multiple configurations embodied within a compact, easily-transportable, package. Disaster environments often contain unstructured, diverse, and challenging terrains and differ by case, exploiting the advantages of modular, reconfigurable robots to meet requirements of the unforeseen tasks at hand. The goal of these robots is to provide first responders with the ability to remotely access and survey the disaster zone, locate victims, and possibly manipulate the environment.

Reconfigurable robots can bypass the complexity of specialized behaviors exhibited in animals competent of locomotion over varying terrain by allowing the robot operator to intelligently choose a task-specific configuration. In this way a single set of robot with varying configurations, morphologies, and capabilities can be created using a modular approach. We believe this modular approach has many advantages over highly specialized robots that are harder to adapt to a wide range of different environments and tasks.

For instance, a hexapod configuration can be used for collapsed, rubble-filled, environments where stability and navigability of terrain is prioritized. Alternatively, a multimodal quadruped with the ability to walk, or roll on wheels, offers both high mobility on rugged terrains in walking mode, as well as a higher efficiency for flat surfaces in rolling mode. Multi-modal platforms are advantageous over long distances over varying terrain. In man-made environments designed for humans, the larger form factor of hexapods and quadrupeds may inhibit their mobility (stairs, narrow corridors, etc.). In these cases a biped configuration may be used. For very confined spaces or areas that have extremely limited access a snake robot could be the most useful configuration. In this chapter, we discuss the locomotion capabilities of various robots like mobile robots, chain/snake robots and legged robots.

1.1. Locomotion of Robot

In this sections, the evolution of three types of robots, mobile, apodal, pedal and modular will be studied. Emphasis will be placed on recently created prototypes and it will be seen, from a general perspective. In the first place the problem of locomotion will be introduced. This is followed by the evolution of self-propelled apodal state of art robots. Then the progress in a popular branch of investigation among most prestigious research laboratories, what is known as modular robotics, will be presented. Along with this, study of locomotion capabilities of podal robots is also discussed.

a) <u>Mobile Robots</u>

In nature, the movement of animals is adapted to environment in which they live. This gives them flexibility to perform various tasks which are impossible of a fixed robot. Mobile robots are robots that employ mechanisms based on wheels, conveyors and treads for motion. However, recently, there is development in the field of bio-inspired mobile robots. These robots employ novel strategies for locomotion based on the study of biological organisms. These robots can generally be classified as podal or apodal robots. Podal robots include hexapod, humanoids, bipeds whereas apodal may include snake robots, lizard robots. Study of locomotion is performed at two levels. Investigations of the superior level start with the supposition that robots can move, without taking into account the mechanisms that make it possible (feet, wheels…) and concentrates on the task of the superior level such as path planning, vision, collaboration and cooperation.

b) <u>Snake Robots</u>

In contrast to terrestrial movement by means of feet, are the living beings that use corporal movements. The robots that use this kind of movement are known as apodal robots. The word apodal means "lacking feet". Snake robots are one such kind of robots. Theses robots possess characteristics that make them unique. On one hand is the ability to change their form. Compared with rigid structures of the rest of the robots, the apodal can bend and adapt to the form of the terrain on which they move. On the other hand, their section is very small compared to their size, which permits them to enter small tubes or orifices and get to places inaccessible to other robots. This section analyses the snake robots created in the most important research centers and their evolution up to now.

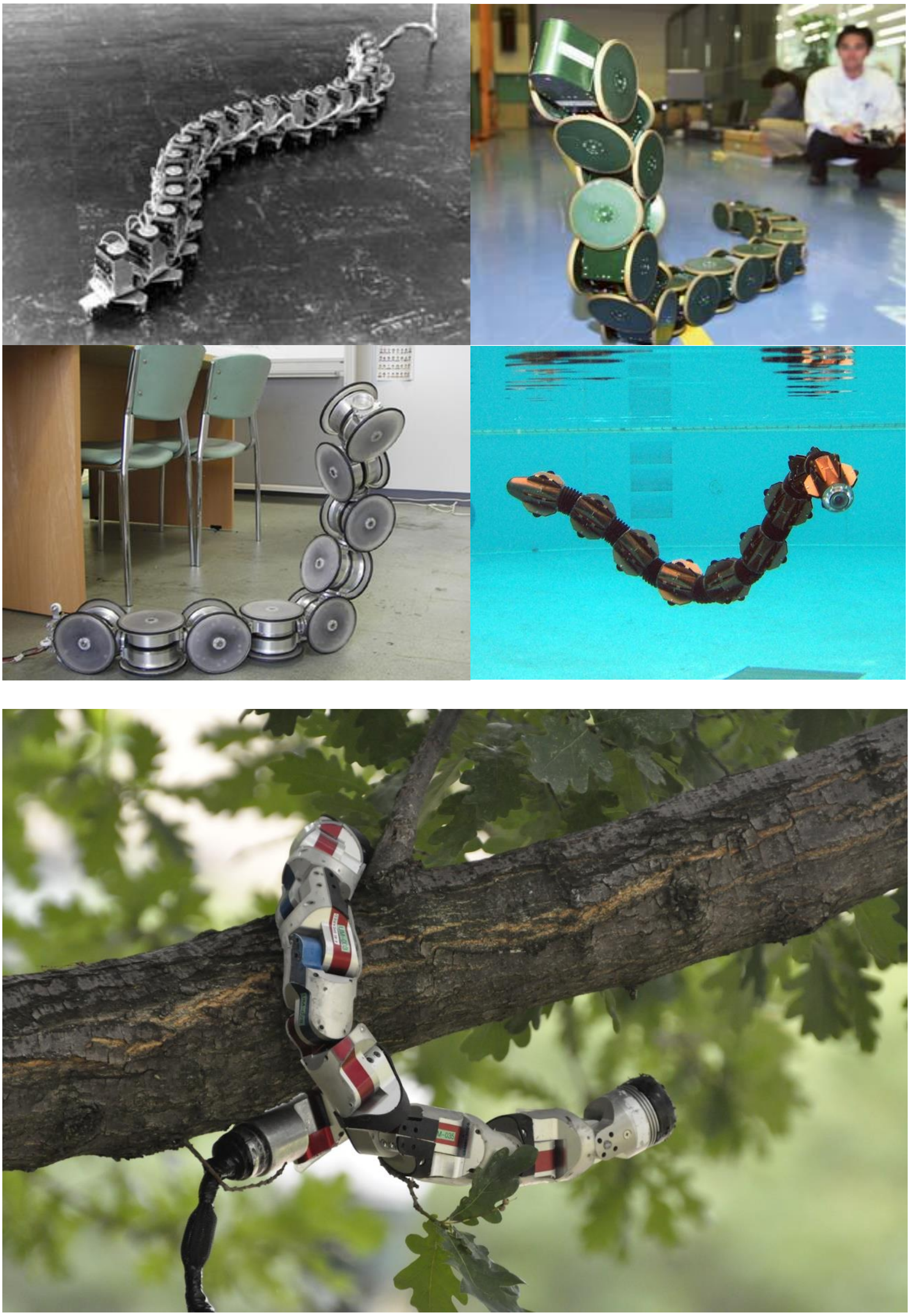


Figure 1-1: ACM Family of- Snake Robots and CMU Snake Robot

Hirose, of the Tokyo Institute of Technology pioneered studies of snake's bio-mechanics for its application to robotics. The first snake robot called ACM-III (Active Cord Mechanism) was implemented in 1976. This prototype was 20 years ahead of it time. ACM - III measures 2 meters long and is made up of 20 articulation that move parallel to the ground (yaw), capable of moving at a speed of 40cm/s. Each module has some passive wheels that allow the robot to crawl along the ground. These wheels have the effect that the friction coefficient in a tangential direction is very low. ACM - R3 has an alternating pitch and yaw freedom at the joints. ACM - R4 was designed for industrial applications and uses and active wheel for motion. The latest ACM - R5 was designed in 2006 and is an amphibious snake robot.

Research at Carnegie Mellon University, headed by Prof. Howie Choset has produced a number of snake robots which have the capabilities to crawl, roll and climb a tree. Along with it his lab has developed a modular snake robot and a Snake Monster robot.

c) <u>Legged Robots</u>

Research in legged robots is very much ahead as compared with snake robots. A list of various humanoid robots developed by various labs further illustrates this point.

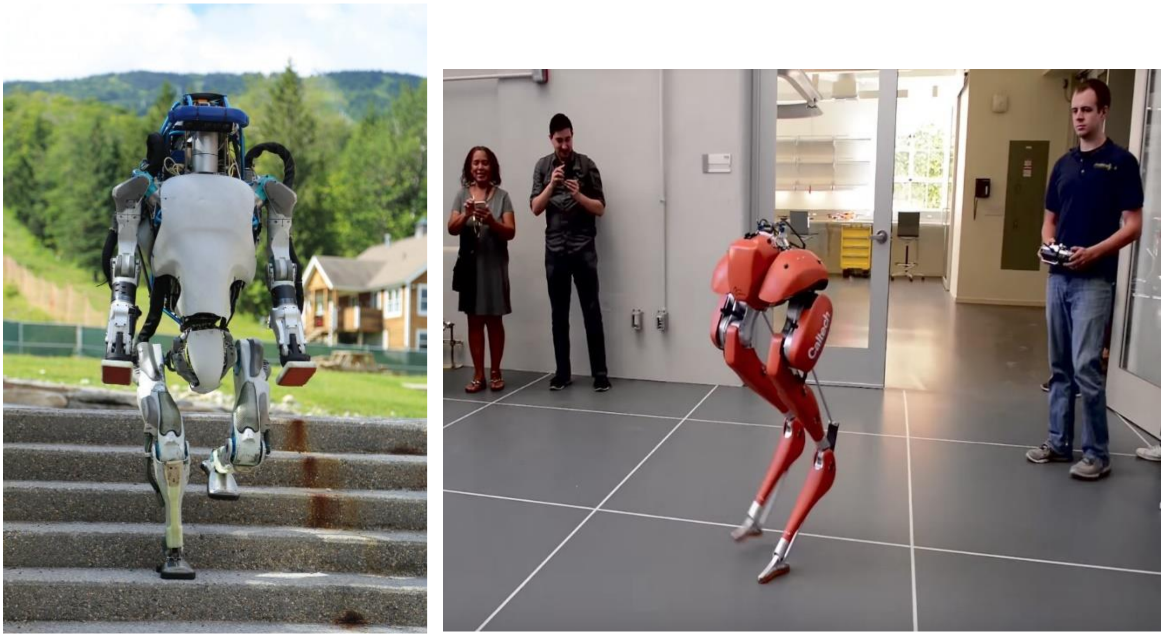


Figure 1-2: Atlas Robot by Boston Dynamics and Cassie Robot by Agility Robotics

Table 1-1 shows various legged robots developed by various research labs

Lab	Robot Name	Website	Reference
Caltech	Cassie	http://www.agilityrobotics.com/	-
Boston Dynamics	Atlas	https://www.bostondynamics.com/atlas	-
Honda	ASIMO	www.honda.co.jp/ASIMO/	-
Sony	SDR-3X	-	Kuroki et al. (2001)
Fujitsu	HOAP-1	pr.fujitsu.com/en/news/2001/09/10.html	-
MIT	Cog	www.ai.mit.edu/projects/humanoid-robotics-group/cog/cog.html	Brooks et al. (1998)
Waseda University	iSHA	www.phys.waseda.ac.jp/shalab~kenji/iSHA/index.html	Suzuki and Hashimoto (2001)
Bundeswehr Univ. Munich	HERMES	www.unibw-muenchen.de/hermes/index.htm	Bischoff (1997)

1.2 Modular Robots

Self-assembling and self-reconfiguring modular robot systems can achieve varied complex tasks. Having the abilities of coordinated self-assembly and self-reconfiguration could allow a robotic system to adapt to different or changing environments on-the-fly. These robotic systems have the potential to exploit self-healing abilities with a reserve supply of low cost robot modules for increased system robustness. They are particularly well suited to situations in which they must adapt to tasks not known a priori such as search and rescue applications in unstructured environments, planetary exploration and deep space exploration.

Modular self-reconfigurable robotic systems can be generally classified into several architectural groups by the geometric arrangement of their units. Several systems exhibit hybrid properties.

Table 1-2 shows list of various modular robots

System	Class	DOF	Author	Affiliation	Year
CEBOT	mobile	various	Fukuda et al.	Nagoya	1988
Polypod	chain	2 3-D	Yim	Stanford	1993
Metamorphic	lattice	3 2-D	Chirikjian	JHU	1993
Fracta	lattice	3 2-D	Murata	MEL	1994
Tetrobot	chain	1 3-D	Hamlin et al.	RPI	1996
3D Fracta	lattice	6 3-D	Murata et al.	MEL	1998
Molecule	lattice	4 3-D	Kotay & Rus	Dartmouth	1998
CONRO	chain	2 3-D	Will & Shen	USC/ISI	1998
PolyBot	chain	1 3-D	Yim et al.	PARC	1998
TeleCube	lattice	6 3-D	Suh et al.	PARC	1998
Vertical	lattice	2-D	Hosakawa et al.	Riken	1998
Crystal	lattice	4 2-D	Vona & Rus	Dartmouth	1999
I-Cube	lattice	3-D	Unsal	CMU	1999
Pneumatic	lattice	2-D	Inoue et al.	TiTech	2002
Uni Rover	mobile	2 2-D	Hirose et al.	TiTech	2002
MTRAN II	hybrid	2 3-D	Murata et al.	AIST	2002
Atron	lattice	1 3-D	Stoy et al.	U.S Denmark	2003
Swarm-bot	mobile	3 2-D	Mondada et al.	EPFL	2003
Stochastic 2D	stochastic	0 2-D	White et al.	Cornell U.	2004
Superbot	hybrid	3 3-D	Shen et al.	USC/ISI	2005
Stochastic 3D	stochastic	0 3-D	White et al.	Cornell U.	2005
Catom	lattice	0 2-D	Goldstein et al.	CMU	2005
Prog. parts	stochastic	0 2-D	Klavins	U. Washington	2005
Molecube	chain	1 3-D	Zykov et al.	Cornell U.	2005
YaMoR	chain	1 2-D	Ijspeert et al.	EPFL	2005
Miche	lattice	0 3-D	Rus et al.	MIT	2006

1.2.1. Lattice Architecture

Lattice architectures have units that are arranged and connected in some regular, three-dimensional pattern, such as a simple cubic or hexagonal grid. Control and motion can be executed in parallel. Lattice architectures usually offer simpler reconfiguration, as modules move to a discrete set of neighboring locations in which motions can be made open-loop. The computational representation can also be more easily scaled to more complex systems.

Figure 1-3: M-Blocks (CSAIL, Massachusetts Institute of Technology, USA)

1.2.2. Chain/Tree Architecture

Chain/tree architectures have units that are connected in a string or tree topology. This chain or tree can fold up to become space filling, but the underlying architecture is serial. Through articulation, chain architectures can potentially reach any point or orientation in space, and are therefore more versatile but computationally more difficult to represent and analyze and more difficult to control.

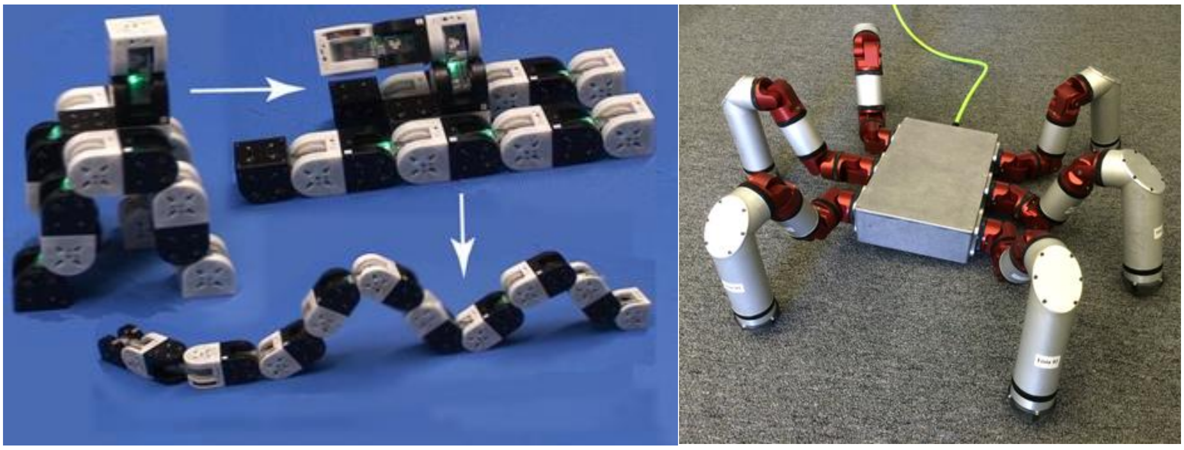


Figure 1-4 MTRAN3 Robot, AIST & Tokyo Tech and Snake Monster, CMU

1.2.3. Mobile Architecture

Mobile architectures have units that use the environment to maneuver around and can either hook up to form complex chains or lattices or form several smaller robots that execute coordinated movements and together form a larger "virtual" network.

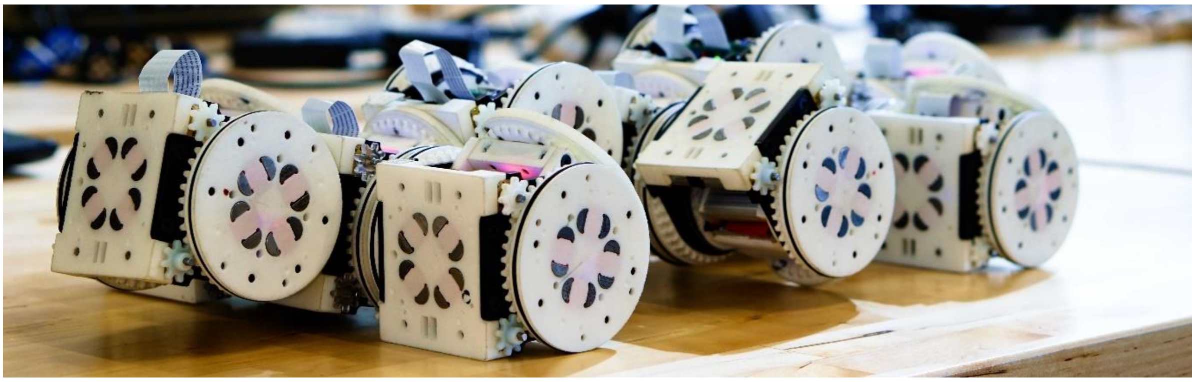


Figure 1-5. Smores Robot, GRASP Lab, University of Pennsylvania

1.3. Motivation

Unpredictable and unstructured scenarios in space exploration, surveillance tasks and search and rescue missions urge the deployment of modular systems which could reconfigure itself to meet a specific task at hand. The Snake Monster allows rapid and robust prototyping of reconfigurable robots but fails at instances where robot operator cannot reach the system to change the module positions. Therefore, a self-reconfigurable robotic system is required. Another work from GRASP Lab – Smores Robot, discusses development of a *Universal Robot* capable of emulating movements abilities of other robots. But the basic building units are incapable of navigating on rough terrains, making the system difficult to change morphology in rugged terrain.

Various researches have shown the versatility of snake robots to navigate on rough terrains and climb stairs. Also, legged robots play vital role when accurate foot placement is required on extreme rough terrains where stability is prioritized and better locomotion is required in terms of speed. Hence, we present a Self-Reconfigurable Transformer Robot capable of changing its morphology by extending the capabilities of snake robots to legged locomotion which is as versatile as a snake robot and dexterous at Little Dog or SpotMini.

1.4. <u>Design Statement</u>

The design goals of the robot can be broken into three categories:

a) System Design Goals:
   - The robot should be polymorphic, it should be able to attain various morphologies and shapes.
   - Self-reconfigurable - it should be able to reconfigure itself on its own without any human physical interaction.
   - Inexpensive – the system must be as cost effective as possible.
   - Wireless system – wires used for powering and communicating with the robot hinders its motion and movements in the workspace.
   - On-Board Processing – it enables robot to take decisions on its own.

b) Module Design Goals:
   - Modularity – the system must have a balance between flexibility and dexterity as highly flexible systems are difficult to control.
   - The robot design must allow the robot to achieve various morphologies.
   - The degrees of freedom (DoF), number of docking ports, geometric shape should allow largest (or atleast useful) range of required motions and configurations with minimum number of motors.

c) Docking Design Goals
   - The docking system should enable modules to connect in many useful arrangements.
   - The docking and undocking must be performed with minimal energy consumption.
   - The docking and undocking mechanism must not use any extra actuator.

1.5. <u>Thesis outline</u>

This thesis deals with the complete prototyping of modular robotic system. The first chapter discusses the design of individual snake module. It also describes the fabrication of snake module using affordable and light-weight 3D printing technology. Gait generation and kinematic study of various morphologies is discussed in subsequent chapter. The generated mathematical gait is verified on real robot and simulated model. The next chapter discusses the transformations of snake robots into various attainable morphologies. Localization and State Estimation approach is discussed in the subsequent chapter along with computer vision techniques.

2. BIOLOGICAL STUDY & MATHEMATICAL MODELLING

2.1 Snake Robots

Snakes are diverse creatures that occupy a wide range of habitats. They also have a wide range of locomotive capabilities, ranging from crawling and burrowing to climbing and even swimming. While snakes all have a similar structure, they do exist in a variety of sizes and aspect ratios. For example, snakes such as the Boidae family (Boas and Pythons) tend to have thicker, heavier bodies, while snakes in families such as Leptotyphlopidea family (Thread snakes and Worm snakes) tend to have thinner body types. Snakes also range in length from more than 20 feet for reticulated pythons and anacondas, to substantially less than 1 foot long for many of the smaller varieties.

2.1.1 Skeletal Structure

The design of a snake is a simple structure that is repeated many times. Snake bodies are elongated forms that consist of a long backbone made of many vertebrae. In fact, there are only three different kinds of bones in the entire snake skeleton: the skull, the vertebrae, and the ribs. Snake backbones consist of 100-400 vertebrae and the design of each vertebra allows small motions in both the lateral and vertical directions. They do not allow any twisting, however, and thus act as compliant universal joints. Each vertebra itself only allows a very small amount of angular motion, but the motions of many vertebrae allow snakes to drastically curve their bodies. Each vertebra allows rotation of 10-20 degrees in the horizontal plane, and between 2-3 degrees in the vertical plane.

2.1.2 Locomotion

Snake-inspired locomotion provides the following advantages over traditional forms of locomotion in both animals and machines.

- Due to their elongated form and lack of legs, snakes have compact cross-sections and thus can move through very thing holes and gaps. In addition to the thinner cross section, snakes also have the ability to climb up and over obstacles that are much taller than their body height. This is done by lifting the front half of their long bodies. Similarly, a snake-inspired robot can lift its body up and over obstacles much larger than most legged or wheeled devices. These properties are very desirable when moving through complex and cluttered environments.

- Gaits used by snakes for locomotion are very stable. Because their bodies are constantly in contact with the ground at many different points, it is difficult to knock them over, especially since they have a low COM and do not lift their bodies off the ground much during locomotion. The form of locomotion that snakes use also relies on a large amount of contact between the ground and the posterior. This large surface area gives the snake good traction characteristics in variable environments. Whereas one wheel or leg in a traditional kind of robot may slip, the large contact surface of a snake-inspired robot would make this occurrence less likely.
- Snakes have redundant designs that rely on the same kind of joint that is repeated many times. This means that if one joint fails, the snake can continue to locomote. The simplicity of the design also means that the snake does not have any fragile appendages that can easily break.
- snakes are very versatile and can act as both locomotors and manipulators, as they can use their bodies to wrap around objects to grasp them. This can be seen in the climbing action across tree branches, or when a constrictor is clenching its prey. Since one structure can do both things, the need for different mechanisms to achieve different tasks is eliminated.
- Despite frictional opposition to their locomotion, snakes have been shown to consume a comparable amount of energy to other biological forms with similar sizes, weights and speeds. This can be explained by the fact that snakes do not perform a significant amount of lifting of their body in their motion, and they also do not consume as much energy by moving different appendages like legged animals.

Snake-inspired robots were introduced in the early 1970’s by Shigeo Hirose. Since that time, numerous snake-inspired robot designs have been conceived and prototyped. Although the various robot designs follow the common theme of mimicking snake locomotion, they may differ greatly in physical configuration and purpose. For example, some robots are redundant; while others are hyper-redundant and others still may have no redundancy at all. Some robots use powered wheels or treads, while others may use passive wheels or no wheels at all. Some designs are even amphibious, travelling effortlessly between ground and water environments. Snake-inspired robots have been proposed for missions ranging from exploration to search and rescue to military reconnaissance and surveillance. There are four major snake locomotion gaits: (1) lateral undulatory, (2) concertina, (3) sidewinding, and (4) rectilinear progression. Most snake-inspired robot designs use either lateral undulation or rectilinear progression.

2.2. Legged Robot

2.2.1. Skeleton Structure

a) Bipedalism

Bipedalism is a form of terrestrial locomotion where an organism moves by means of its two rear limbs or legs. Habitual Bipedalism is observed in mammals like kangaroo and humans. Many mammals are engaged in limited, non-locomotory, bipedalism. Several mammals adopt a bipedal stance in specific situations such as for feeding or fighting. Bears fight in a bipedal stance to use their forelegs as weapons. Ground squirrels and meerkats will stand on hind legs to survey their surroundings, but will not walk bipedally. The gerenuk antelope stands on its hind legs while eating from trees. Several lizard species move bipedally when running, usually to escape from threats.

Figure 2-1: Bipedalism observed in lizard, kangaroo and deer

Bipedalism raises the head; this allows a greater field of vision with improved detection of distant dangers or resources. While upright, non-locomotory limbs become free for other uses, including manipulation (in primates and rodents), flight (in birds), digging (in giant pangolin), combat (in bears, great apes and the large monitor lizard) or camouflage (in certain species of octopus). Maximum bipedal speed appears less fast than maximum speed of quadruped with a flexible backbone but over longer distances bipeds like human outrun other animals per endurance running hypothesis.

For nearly the whole of the 20th century, bipedal robots were very difficult to construct and robot locomotion involved only wheels, treads, or multiple legs. Recent cheap and compact computing power has made two-legged robots more feasible. Some notable biped robots are ASIMO, HUBO, MABEL and Cassie and Atlas.

Figure 2-2: Bipedal robots Atlas, Boston Dynamics; Asimo, Honda; Cassie Robot, Agility Robotics

b) **<u>Quadrupedalism</u>**

Quadrupedalism or pronograde posture is a form of terrestrial locomotion in animals using four limbs or legs. Due to extra limbs, quadrupeds' robots are more stable than bipedal robot. Most quadrupeds are vertebrate animals, including mammals such as cattle, dogs and cats, and reptiles such as lizards.

Figure 2-3: Cheetah chasing for the hunt

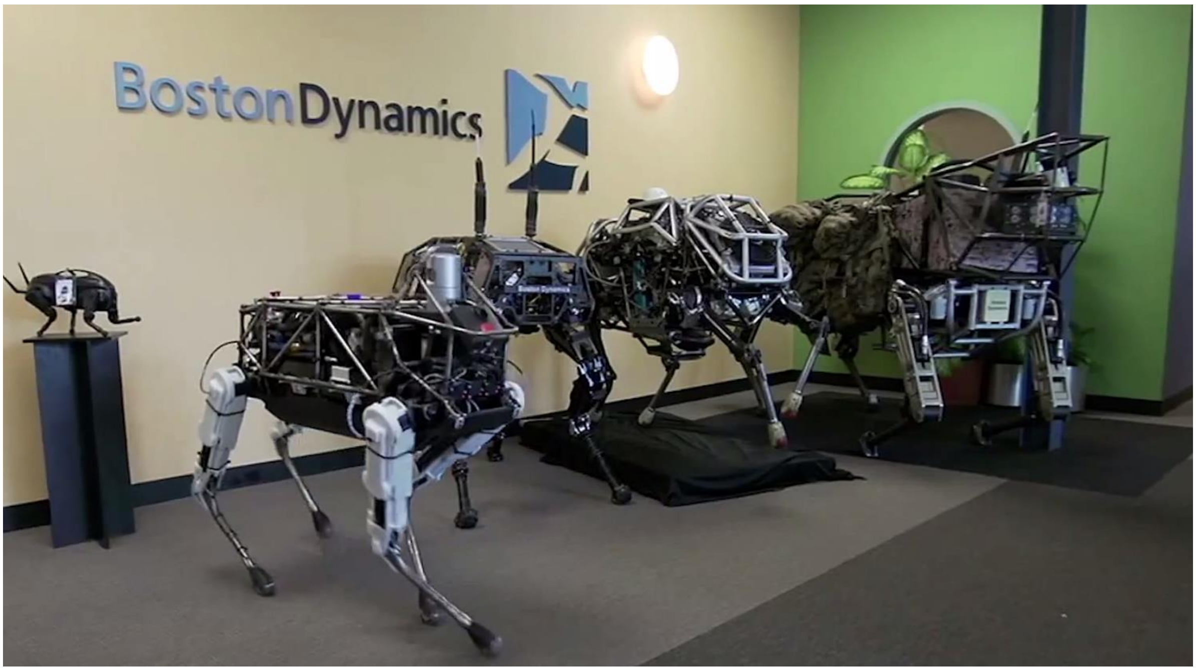


Figure 2-4: Quadruped Robots from Boston Dynamics: LittleDog, Spotmini, Cheetah robot and BigDog

Boston Dynamics is an American engineering and robotics design company that is best known for the development of BigDog, a quadruped robot designed for the U.S. military with funding from Defense Advanced Research Projects Agency (DARPA). Marc Raibert is the company's president and project manager. He spun the company off from the Massachusetts Institute of Technology in 1992.

The Cheetah is a four-footed robot that gallops at 28 miles per hour (45 km/h; 13 m/s), which as of August, 2012 is a land speed record for legged robots. The previous record was 13.1 miles per hour (21.1 km/h; 5.9 m/s), set in 1989 at MIT. This robot has an articulated back that flexes back and forth on each step, thereby increasing its stride and running speed, much like the animal does. LittleDog has four legs, each powered by three electric motors. The legs have a large range of motion. The robot is strong enough for climbing and dynamic locomotion gaits. The onboard PC-level computer does sensing, actuator control and communications. LittleDog's sensors measure joint angles, motor currents, body orientation and foot/ground contact. Control programs access the robot through the Boston Dynamics Robot API.

2.2.2. Locomotion in legged Robots

a) Static Walking

Static balance or static walking refers to a system which stays balanced by always keeping the center of mass (COM) of the system vertically projected over the polygon of support formed by the feet. While this is the case there can be no horizontal acceleration due to tipping moments caused by gravity. Therefore, whenever a foot or leg is moved, the COM must not leave the area of support formed by the feet still in contact with the ground. This is illustrated by Figure. 2-5.

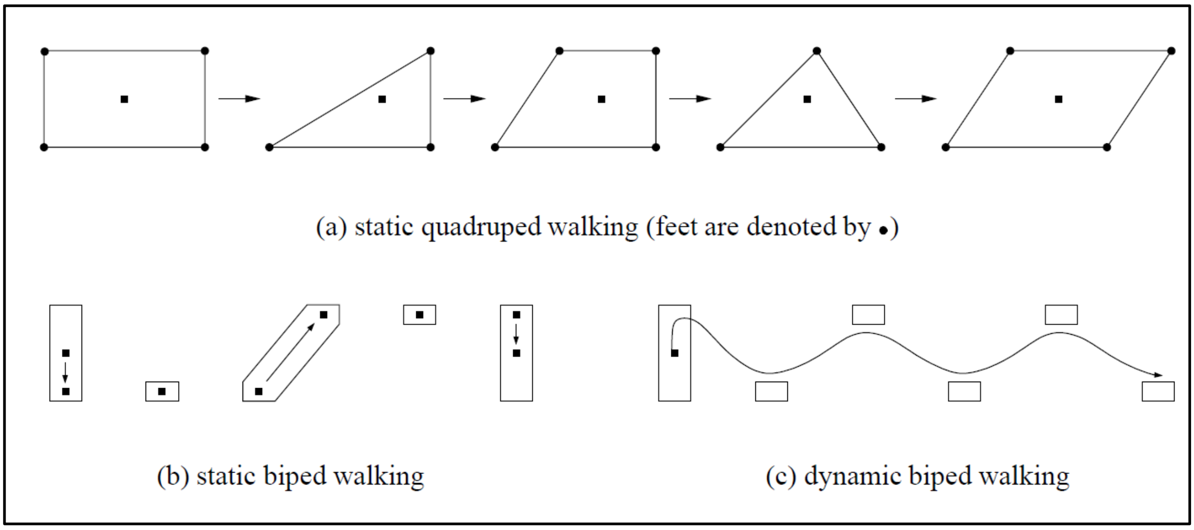
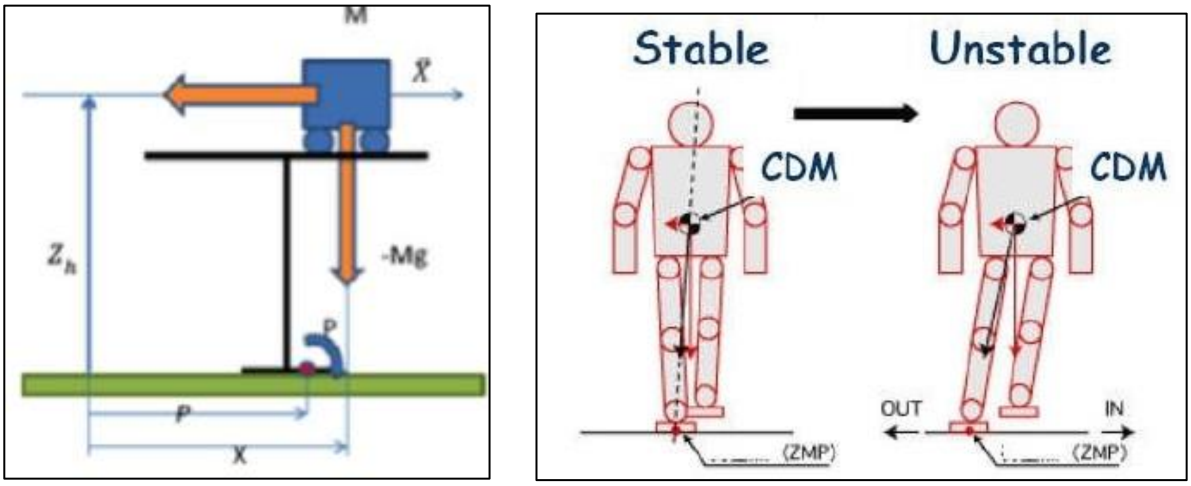


Figure 2-5: Support polygon (a) static quadruped walking (b) static biped walking (c) dynamic biped walking



(a) **(b)**

Figure 2-6: (a) Cart Table Model (b) Location of CoG and ZMP points during static and dynamic walking

When we compare the two methods of balance, we see that the static method is highly restrictive and results in movement which is slow. Very rarely do animals and humans exhibit such behavior for this reason—the velocity achievable is very low and the motion is not efficient. However, we can see that by removing the constraining nature of the rule for static balance that the mobility of the system is increased. This is due to the increased flexibility of the movement of the legs and placement of the feet. The accelerating tipping moments can be used to achieve higher speeds, move all legs at once or to utilize footholds which are far apart.

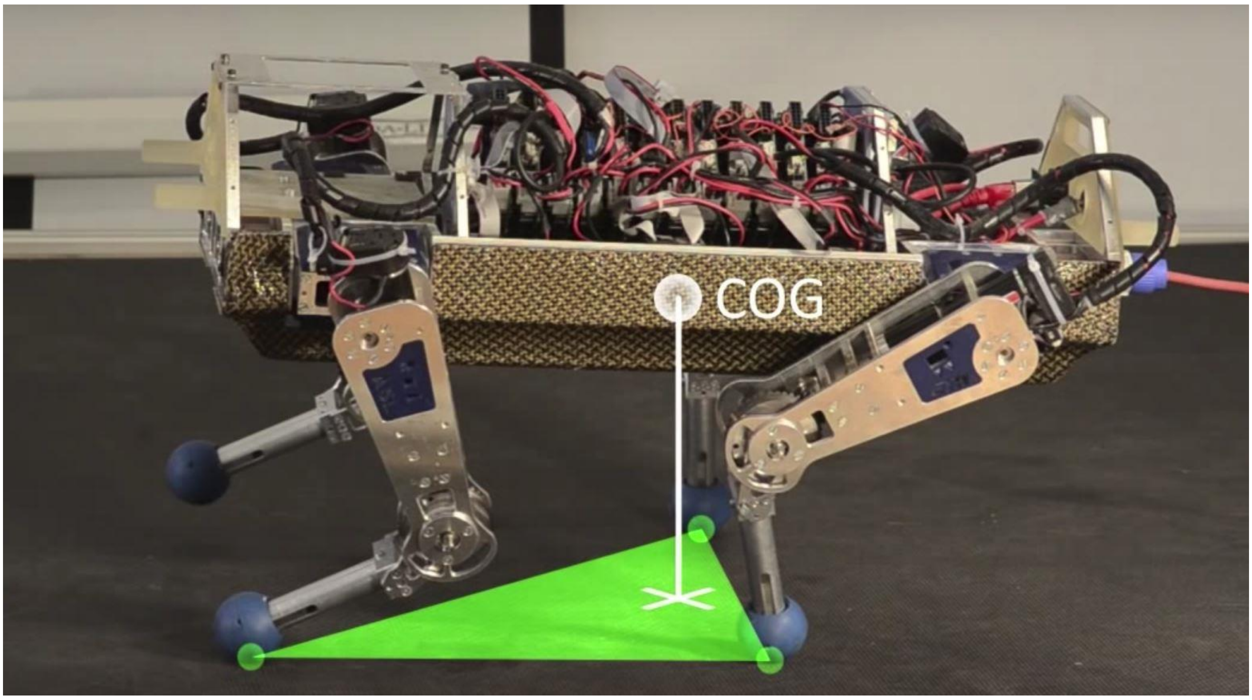


Figure 2-7: CoG projection lying in support polygon during stable static walking of quadruped

b) <u>Dynamic Walking</u>

For a bipedal robot to gain efficiency and speed, it will require dynamic balance. Much of this dissertation will be concerned with analyzing the forces on the system, which result from the COM being outside the base of support of the robot. The foot cannot be controlled directly but in an indirect way, by ensuring the appropriate dynamics of the mechanism above the foot.

Defination:

As the load has the same sign all over the surface, it can be reduced to the resultant force $\boldsymbol{F}_P$*, the point of attack of which will be in the boundaries of the foot. Let the point on the surface of the foot, where the resultant* $\boldsymbol{F}_P$ *passed, be denoted as the Zero-Moment Point.*

To clarify this statement, consider a rigid foot with a flat sole which is fully contacting and supported by the floor, as depicted in figure 3.1. For simplicity, the influence of the biped is replaced with the force $\mathbf{F}_A$ and the moment $\mathbf{M}_A$ acting on a point A on the floor. The gravitational acceleration is **g**, acting in the negative z direction. To keep the whole biped in balance: in point P the reaction force $\mathbf{F}_P = (F_{PX}, F_{PY}, F_{PZ})$ and the moment $\mathbf{M}_P = (M_{PX}, M_{PY}, M_{PZ})$ are acting. The horizontal reaction force (FPX, FPY) is the friction force that is compensating for the horizontal components of force $\mathbf{F}_A$. The vertical component of the reaction moment, being M_{PZ}, is balancing the vertical component of moment $\mathbf{M}_A$ and the moment induced by the force $\mathbf{F}_A$. Assuming there is no slip, the static friction is represented with (F_{PX}, F_{PY}) and M_{PZ}. Before deriving the equilibrium equations, i.e. the static balance equations, a few remarks about the point P. First of all: to compensate for the horizontal components of $\mathbf{M}_A$, being (M_{AX}, M_{AY}), the point P is shifted in such a way that F_{PZ} is fully compensating for them (obviously, with the 'arm' $\mathbf{d} = \mathbf{p}_{OP} - \mathbf{p}_{OA}$, lying on the floor plane). This implies that the horizontal components of $\mathbf{M}_P$ are reduced to zero. Hence:

$$M_{PX} = M_{PY} = 0$$

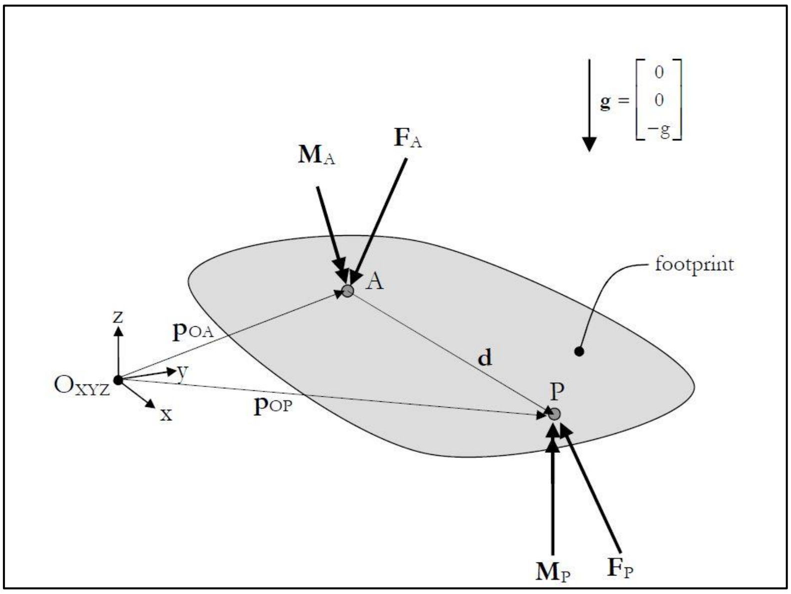


Figure 2-8: Forces and Moments acting on a rigid foot with a flat sole; fully supported with the floor.

In order to achieve a dynamically stable gait the ZMP should be within the support polygon at every instance.

To calculate the point P, there are several assumptions that have to be made:
a) The biped robot consists of n rigid links.
b) All kinematic information, such as position of CoM, link orientation, velocities, etc. are known and calculated by *forward kinematics*.
c) The floor is rigid and motionless.
d) The feet cannot slide over the floor surface.
e) All joints are actively actuated.

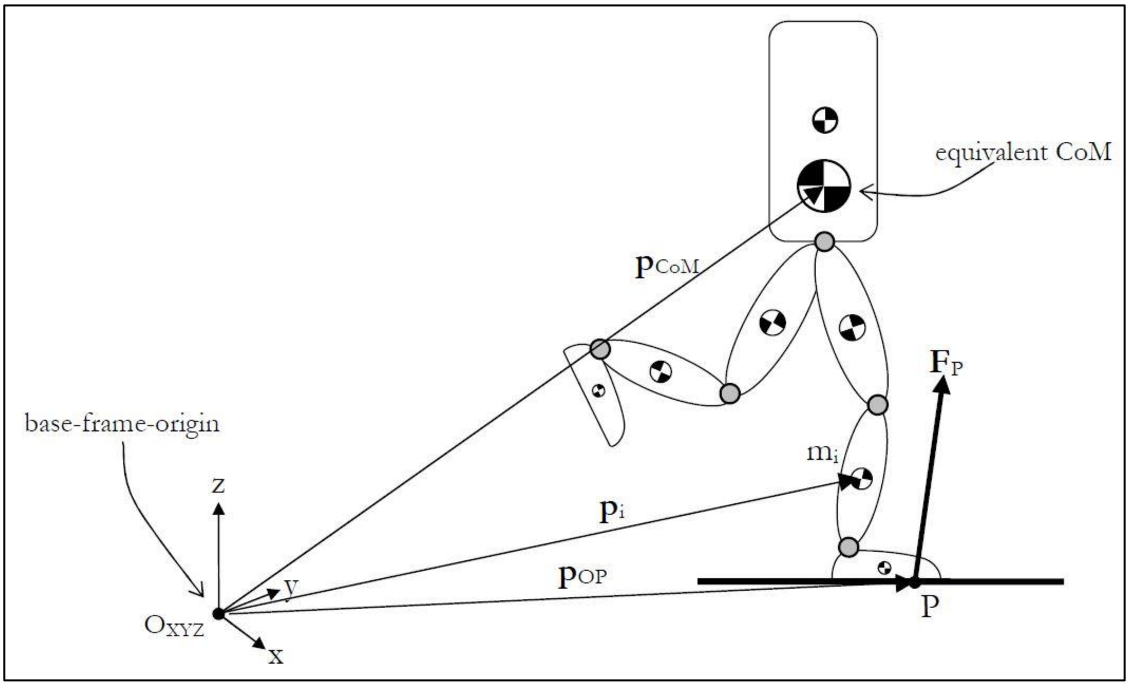


Figure 2-9: Schematic 3D Biped model and point P.

With this the total *linear momentum P* and, respectively, the total *angular momentum* **H** with respect of the base-frame-origin can be stated as:

$$\mathbf{P} = \sum_{i=1}^{n} m_i \dot{\mathbf{p}}_i$$

$$\mathbf{H} = \sum_{i=1}^{n} \{ \mathbf{p}_i \times m_i \dot{\mathbf{p}}_i + \mathbf{I}_i \boldsymbol{\omega}_i \}$$

From the basic knowledge of the ZMP several equations for the calculation of it were derived. Components of ZMP position can be calculated from following equation.

$$x_{ZMP} = \frac{m_{tot} g_z p_{CoMx} + z_{ZMP} P'_x - H'_y}{m_{tot} g_z + P'_z}$$

$$y_{ZMP} = \frac{m_{tot} g_z p_{CoMy} + z_{ZMP} P'_y - H'_x}{m_{tot} g_z + P'_z}$$

3. Self-Reconfigurable Transformer Robot

3.1. Mechanical Design and Fabrication

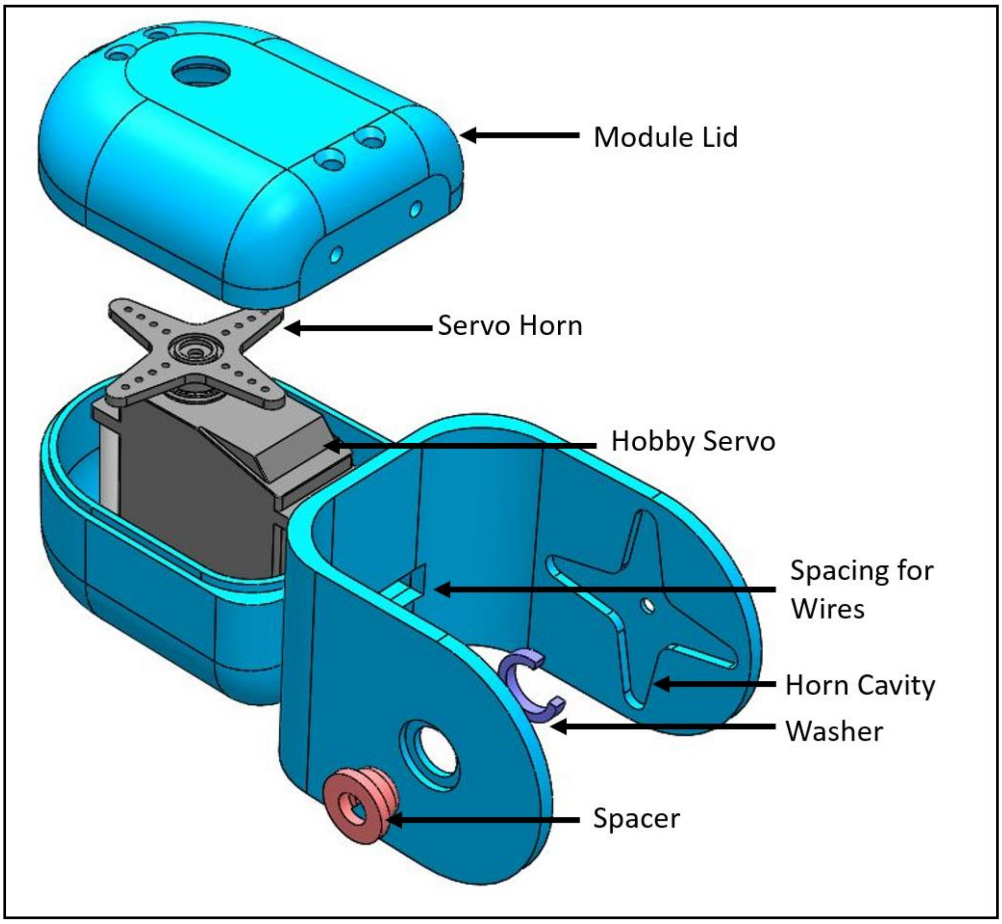


Figure 3-1: Design of single module of snake robot

Figure 3-1 shows the basic module of snake robot. Each module consists of a single DoF (± 90 degrees). The module has three main parts: 1) Module Case 2) Hobby Servo motor 3) Module Lid. The Servo motor is supported using the spacers. Servo Horn is fixed such that the module lid encloses the motor in the case. The horn of servo motor extrudes into subsequent module in the horn cavity provided; smartly transferring the power with no screw attachment required. Spacer and washer are attached for free motion of the joint. Hollow spacer allows room for the wires to exchange between modules. The head module is designed so as to accommodate microprocessor, servo controller board and camera. While the tail module contains Battery and power distribution circuit. The 4 DoF assembled snake is shown in Figure 3-2.

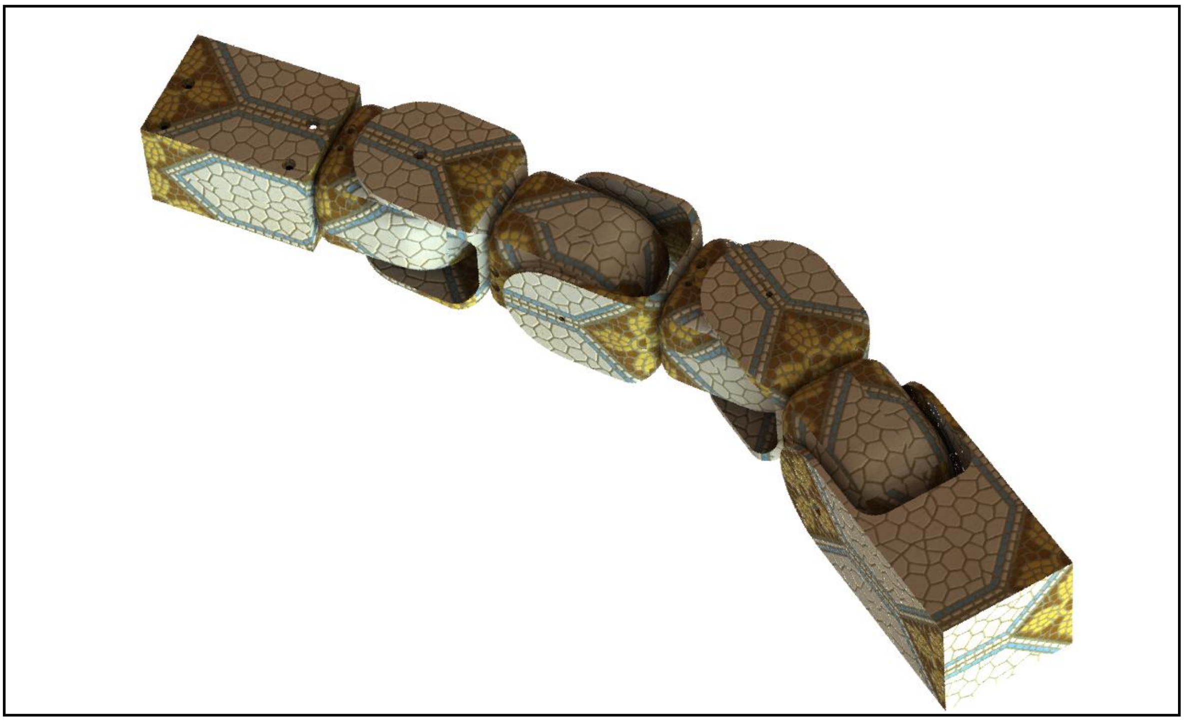
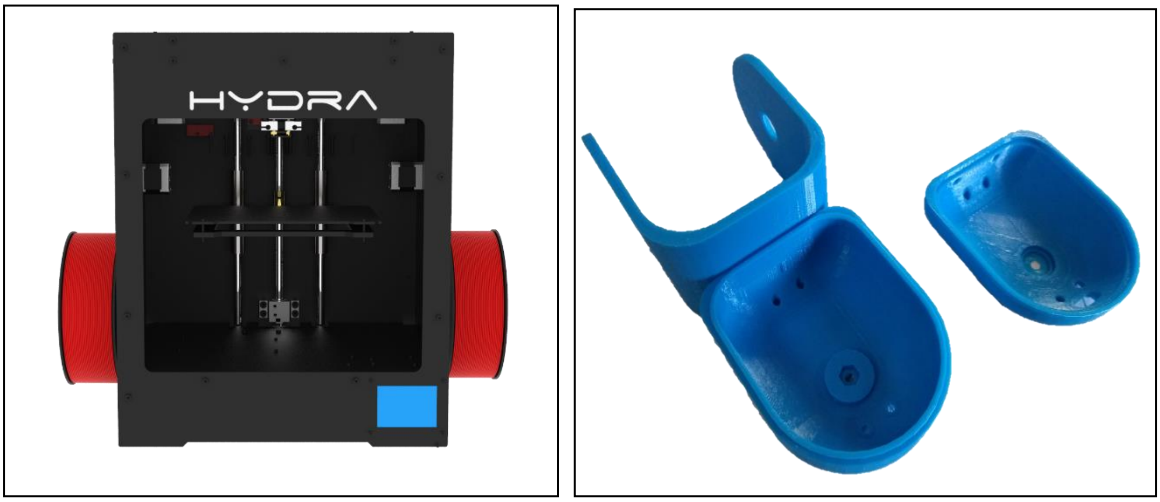


Figure 3-2: Design of assembled Snake Robot.

The fabrication was done using 3D Printing Technology on Hydra Pro 200 3D Printer. Poly-Lactic Acid (PLA) plastic material was used as the filament.



(a) (b)

Figure 3-3 shows (a) Hydra Pro 200 3D Printer (b) Printed snake module

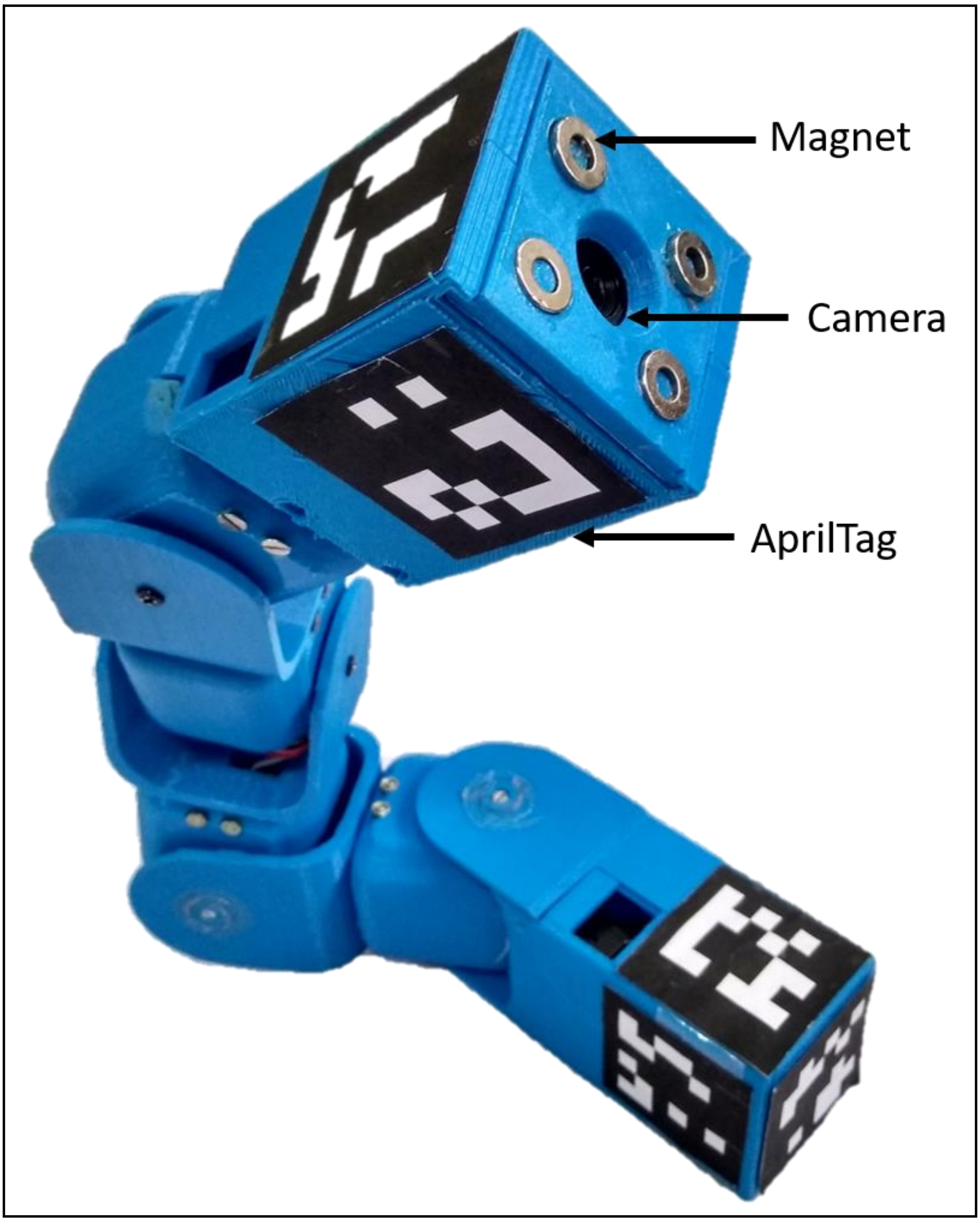
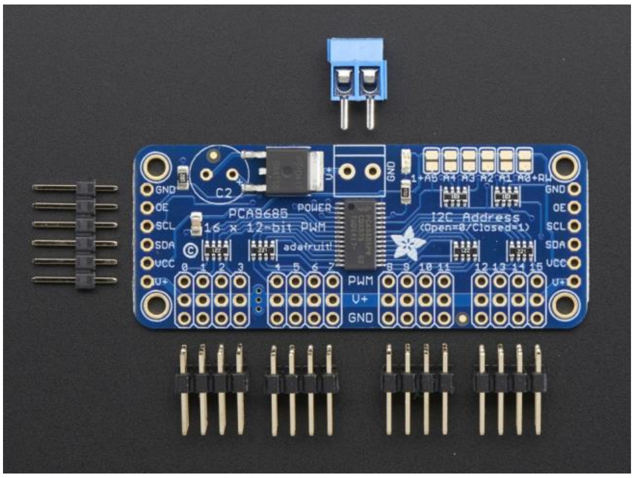


Figure 3-4: Fabricated wireless snake robot with magnets, AprilTag and Camera.

3.2) <u>Electronic Interface</u>

3.2.1. <u>Servo Interface</u>

Servo motor (as shown in Figure 3-5 a) can be easily controlled via microprocessor Raspberry Pi Zero W through Pulse Width Modulation (PWM) signal. Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. Continuous rotation servos allow the rotation of the shaft to be set to various speeds. As only three PWM pins are available on the microprocessor, Adafruit 16 - channel PWM/ Servo HAT (as shown in Figure 3-5 b) is used to control four servo motors.



(a) (b)

Figure 3-5: (a) Hobby Servo Motor (b) Adafruit 16 - channel PWM/ Servo HAT

3.2.2. <u>Raspberry Pi Zero W –Processor</u>

Raspberry Pi Zero W (as shown in Figure 3-6) is the affordable development board from Raspberry Pi Foundation acting as a tiny computer in compact size. It comes with 1GHz, single Core BCM2835 ARMv6 SoC Microprocessor chip with 512 MB RAM. 40 GPIO pins enables it to connect several sensors and peripherals along with UART, I2C and SPI communication. Monitor can be connected at Mini- USB Port. Micro-USB On-The-Go port can be used to connect USB peripherals such as camera, mouse, keyboard, etc. WiFi 802.11 n wireless LAN and Bluetooth 4.1(BLE) enable wireless communication with the processor. All these features packed in 65mm x 30 mm board, making it perfect for the application.

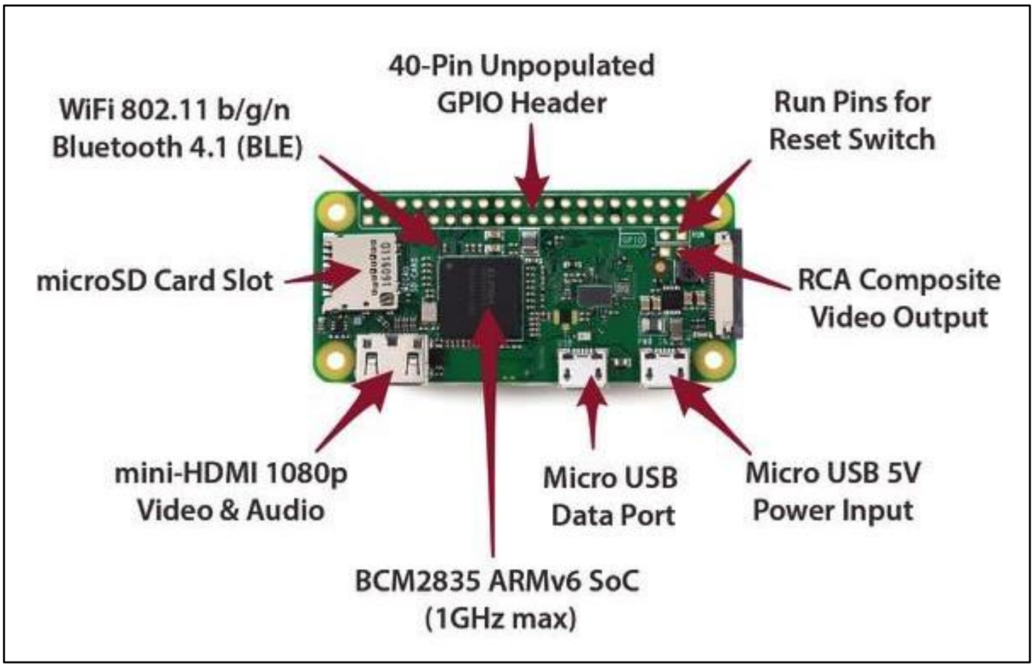


Figure 3-6: Raspberry Pi Zero W Microprocessor Board

3.3. Software Overview

3.3.1. Cyberphysical Architecture

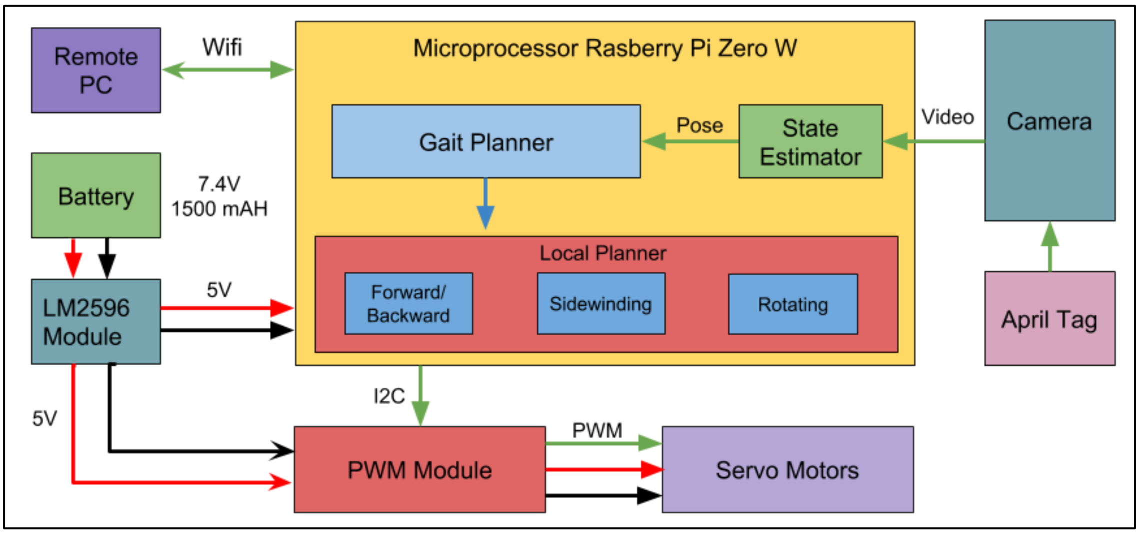


Figure 3-7: Cyberphysical Architecture

Figure 3-7 shows the Cyberphysical architecture of the snake robot. The snake can be controlled remotely over wifi. The camera detects the marker (AprilTag) and the feed is received by the State Estimator node which computes the state (pose: x, y, z, roll, pitch, yaw) of the robot with respect to the marker. According to the robot position and orientation the Gait Planner node takes the decision of activating suitable locomotion gait available in Local Planner and sends the motor angle data to the PWM Module through I2C communication. Accordingly, motors are position controlled through PWM signals from PWM module.

3.3.2 <u>ROS Framework</u>

Robot Operating System (ROS) is a collection of software frameworks for robot software development, providing operating system-like functionality on a heterogeneous computer cluster. ROS provides standard operating system services such as hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management. Running sets of ROS-based processes are represented in a graph architecture where processing takes place in nodes that may receive, post and multiplex sensor, control, state, planning, actuator and other messages. Despite the importance of reactivity and low latency in robot control, ROS, itself, is not a real-time OS, though it is possible to integrate ROS with real-time code.

Software in the ROS Ecosystem can be separated into three groups: (1) language- and platform- independent tools used for building and distributing ROS-based software; (2) ROS client library implementations such as roscpp, rospy and roslisp; and (3) packages containing application- related code which uses one or more ROS client libraries. Both the language-independent tools and the main client libraries (C++, Python and LISP) are released under the terms of the BSD license, and as such are open source software and free for both commercial and research use. The majority of other packages are licensed under a variety of open source licenses. These other packages implement commonly used functionality and applications such as hardware drivers, robot models, datatypes, planning, perception, simultaneous localization and mapping, simulation tools, and other algorithms.

3.3.3 <u>Gait Generator Node</u>

Gait Generator node is the one which calculates the set of angles for each snake robot. As per the input from communicator node, this node generates angles for each motor as the gait cycle. This node uses the Adafruit_PCA library to command the servo motors using the

PWM pulses through GPIO pins on the processor. This node consists of different functions for each gait cycle. The communicator node gives the information about which gait cycle and how many such cycles.

Table 3-1 shows Gait Generator node with their performing gaits

Gait Generator Node	Gait
snake_forward.py	Rectilinear motion in Snake configuration
snake_backward.py	Backward motion in Snake configuration
snake_rolling.py	Rolling motion in Snake configuration
snake_final.py	Snake to Biped transformation
biped.py	Static walking in Biped configuration

3.3.4 Communicator Node

This node enables hardware abstraction capabilities to the remaining nodes. The main function of this node is to take the inputs from camera and do the processing required for computer vision algorithms to find the position of another snake in the environment. This node communicates with the gait generator node for the movement of snake robot. The decision of planar motion is taken on the basis of two gaits: rectilinear motion and backward motion and rolling motion.

3.4 Kinematics

Self-Reconfigurable Transformer is capable of snake as well as bipedal gaits. Also a special gait needs to be implemented to transform the snake robots into a bipedal robot. This section describes the gaits implemented on the robot.

3.4.1. Snake gaits

To mimic snake motion, we implemented gaits based on sinusoidal curves. The gaits implemented on snake robots are summarized here. The snake gaits consist of two sinusoidal waves; one in each horizontal and vertical plane.

$$angle(n,t) = \begin{cases} A_x \sin(\omega_x t + n\delta_x), where \quad n = even \\ A_y sin(\omega_y t + n\delta_y + \phi), where \quad n = odd \end{cases}$$

$A_x, A_y : Amplitudes$

$\delta_x, \delta_y : SpatialFrequency(Decides \quad frequency \quad based \quad on \quad modules)$

$\omega_x, \omega_y : Temporal \quad Frequency(Decides \quad frequency \quad of \quad sine \quad wave \quad to \quad be \quad propagated \quad w.r.t. \quad time)$

$\phi : Phase \quad difference \quad between \quad sine \quad waves \quad in \quad horizontal \quad and \quad vertical \quad plane.$

Table 3-2 : Shows parameters for snake gait

Gait	**Parameters**			
	Amplitude	Frequency	Phase Difference	
Sidewinding	Ax=300 Ay=300	x=5/6 y=5/6	x=2/3 y=2/3	0
Rolling	Ax=300 Ay=300	x=5/6 y=5/6	x=2/3 y=2/3	6

As the snake robot, have only 4 modules, so the equation of rectilinear motion slightly changes as follows:

$$\theta_1 = 70 * (i/100)$$
$$\theta_2 = -\theta_1 - \arcsin(\frac{L_1}{L_2} * \sin(theta_1))$$

These equations are obtained from the geometry of snake robots. L1=166.26 & L2=188.23. Here L1is the link length of snake robot between the head and the first servo motor whose axis is horizontal and L2is the length of the link in between the first and second servo motor whose axes are horizontal. Let these two motors be m1 and m3. As per the gait equations, 1is the angle calculated for motor m1and 2for motor m3. Motors m2 & m4will always be in home position (i.e. at zero degrees).

3.4.2 <u>Transforming Gaits</u>

We have designed the transforming gaits using key-frame interpolation based approach. This approach is commonly used by animation community. Key-frame consists of a set of joint angle data at a timestamp. Figure shows key-frames of the transforming gait which reconfigures the two snake robots into walking configuration. To ensure the stability of the robot, the ground project of center of mass must lie within the support polygon of the robot.

Hence every key-frame must satisfy the criteria. Further, the criteria should also be satisfied while transitioning between key-frames.

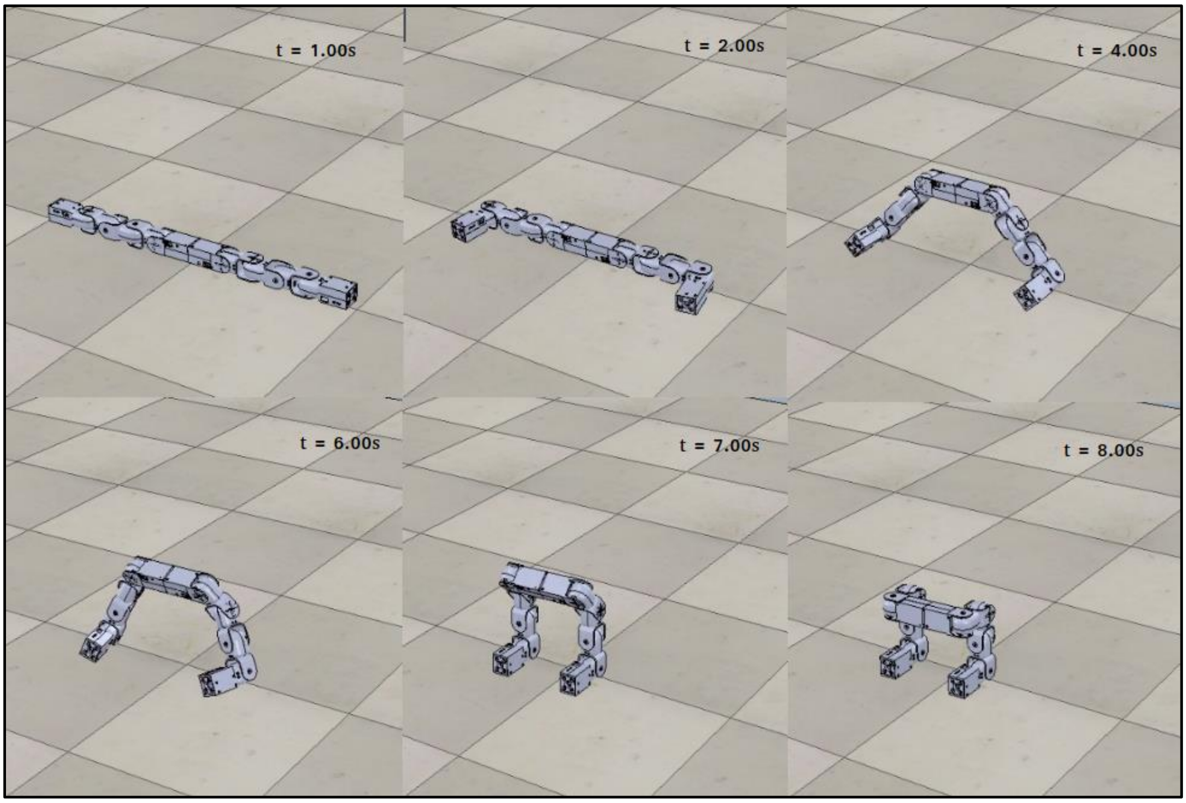
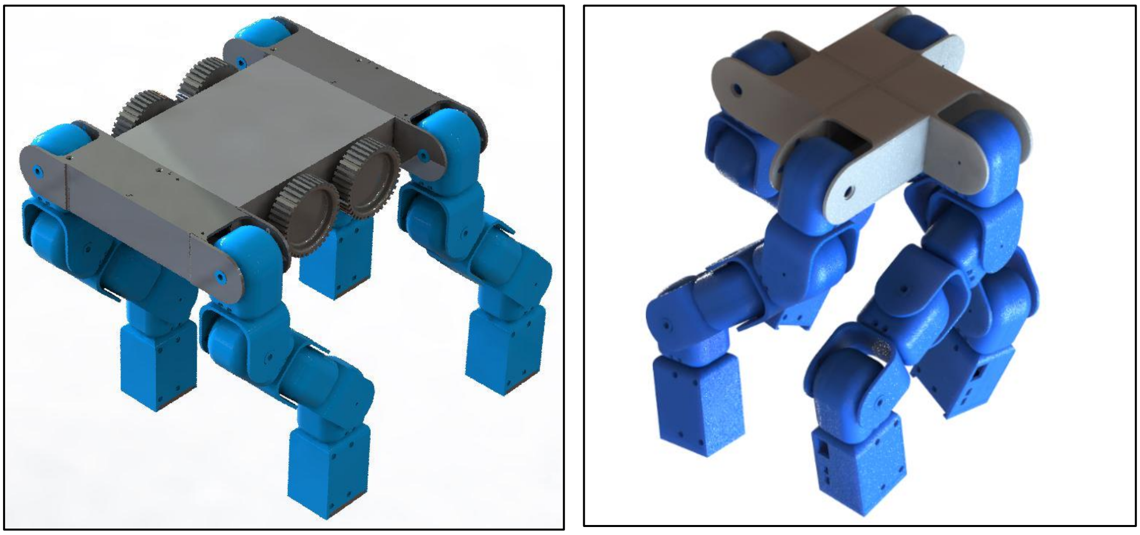


Figure 3-8: Snake robots to Bipedal transformation key-frames.

In key-frame 1, the two robots are attached together and have formed a straight open chain by setting all joint angles to zero. In key-frame 2, the robot reconfigures the tails of both snake robots. In key-frame 3, the robot achieves almost a U shape by lifting its center part from the ground using the support of the tails of the two snake robots. In key-frame 4, the robot is completely transformed into a U shape configuration with its COM within the support polygon. In key-frame 5, the robot reconfigures into the walking configuration.

Figure shows the quadrupedal morphologies attainable by the system: (a) Quadruped robot (b) QuadMonster. Snakes in Quadruped Robot are attached to the four-wheeled robot. This gives the system capabilities of wheeled locomotion for fast traversing on smooth surface. QuadMonster is a unique design – first of its kind. QuadMonster due to its design is able to traverse through confined spaces and is capable to rotate about its central axis. As no wheeled robot is attached to the robot, the snake robot themselves are enough for the transformations.



(a) **(b)**

Figure 3-9: (a) Quadruped Robot (b) QuadMonster robot

3.4.3 Bipedal Walking Gaits

Most of the bipedal walking gaits are designed using zero-moment-point (ZMP) or linear inverted pendulum (LIP) approach. In these methods CoM/ZMP is shifted within the support polygon of one of the legs and then the other leg is moved forward. The configuration of our robot is designed in such a way that it inherently satisfies the stability criteria without lateral shifting of the ground projection of CoM.

To generate gait trajectory, we assumed the foot in contact with the ground as the base and the other foot as the manipulator. DH convention is used for forward kinematics. The equation for forward kinematics is given by:

$$^{n-1}T_n = \left[\begin{array}{ccc|c} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ \hline 0 & 0 & 0 & 1 \end{array}\right] = \left[\begin{array}{ccc|c} & R & & T \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

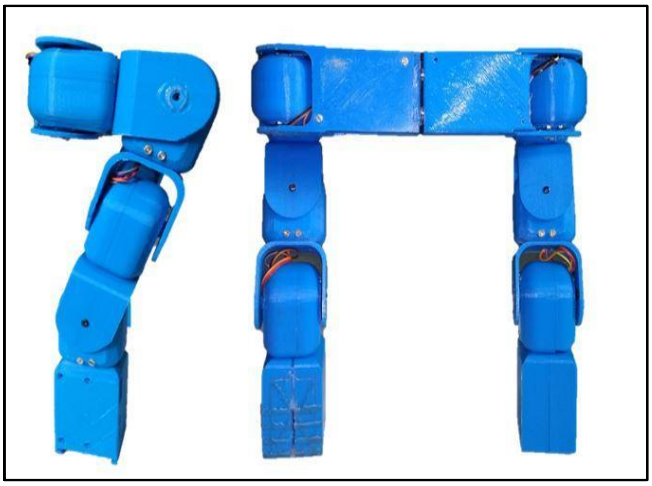


Figure 3-10: Snake robots in Walking configuration. Side view (left) and Front view (right)

3.4.4. <u>Quadruped Walking Gait</u>

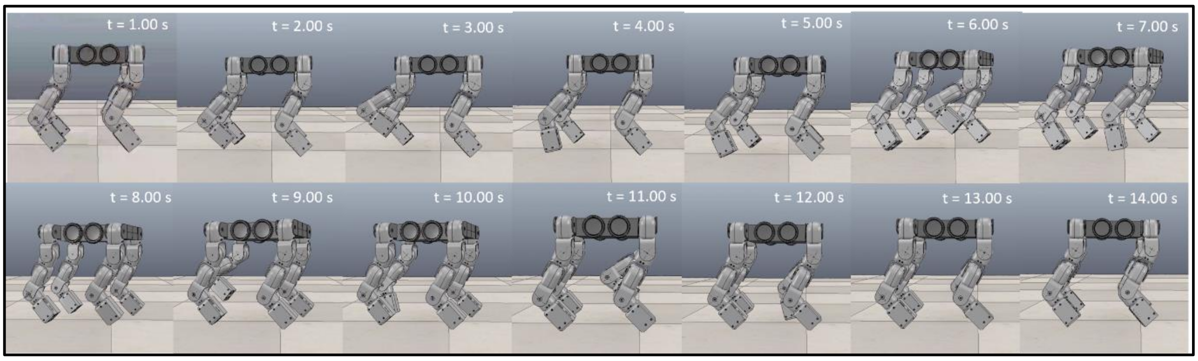


Figure 3-11: Time lapsed key frames of complete gait cycle of quadruped robot.

There are two main quadruped robot gaits: Creep Gait and Trot Gait. Fig shows static walking creep gait of quadruped implemented in open source simulation software VRep. The walk has been described as the least tiring and most efficient form of locomotion of the quadruped. While walking quadruped always has atleast three legs touching the ground. The basic principle of forward shift of Centre of Mass (CoM) along with stabilizing Centre of Gravity (CoG) is implemented. In summary, each leg picks up and moves forward during its own quarter-phase, and then moves backwards during the other 3 quarter-phases. The overall action results in very smooth and even forward movement, since all legs are in constant motion here. The body remains nice and level. The whole gait cycle can be divided into two phases; stance phase and swing phase. Stance phase can be further divide into two phases:

one in which CoG is shifted in the support polygon and other when leg is in none of the motion state (when any of the other leg is swinging). The CoG is shifted by actuating the Roll and Knee pitch motors. Once the CoG is shifted in the support polygon, the swing leg follows the Gaussian trajectory as shown in Fig. with defined step length and step size. The Figure shows time lapsed key frames of complete gait cycle in time interval of 1s.

Step Length = 160 mm

Step Height = 50 mm

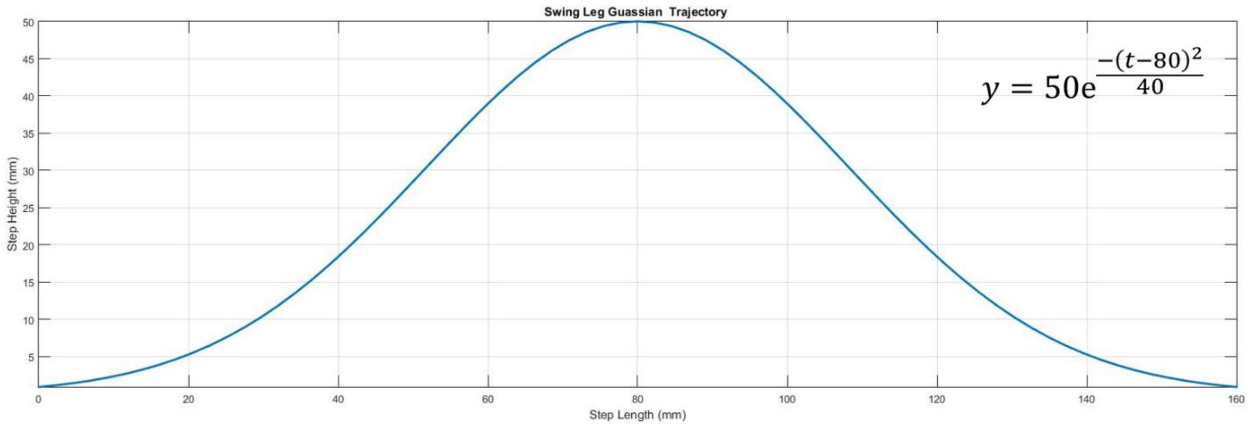


Figure 3-12: Swing Leg Gaussian Trajectory

Lifting only 1 leg at a time sounds nice, but in the real world, this doesn't always work as predicted – for a quadruped, at least. It turns out, if the quad's legs are too short with respect to its body length, or they don't travel far enough (front-to-back) towards the mid-line of the body, or they are not coordinated well, then the 3 down legs may not form a stable tripod when the fourth is in the air. The down leg on the same side as the lifted leg, especially, must have its foot positioned far enough back, else the COG may not be contained within the stability triangle formed by the 3 down legs. Overall, creep stability relates to: body length, body width, leg length, leg angles, foot positions, and general distribution of weight on the body.

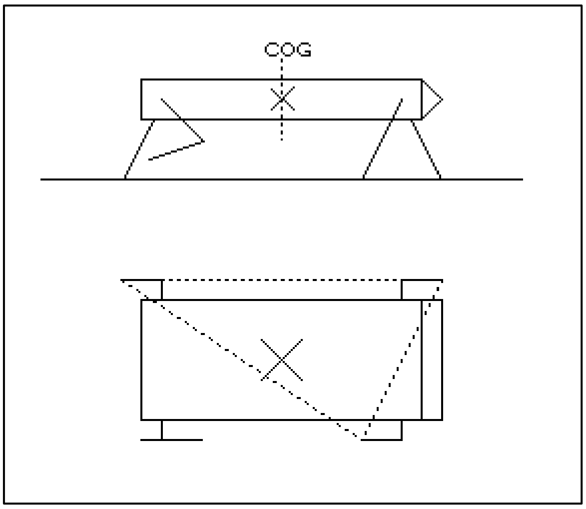


Figure 3- 13: CoG projection during Creep Gait

The diagram above illustrates this. Given the position of the right front leg relative to the left rear, the associated edge of the stability triangle falls very close to the COG at this point. If those 2 legs are not coordinated correctly, a point of instability may occur nearby in the stride. To improve stability here, the right front foot would have to touch down further back.

4. COMPUTER VISION

Navigation and path planning could not be achieved unless distance information is obtained. For other activities of human such as object recognition, shape extraction, etc., some other cues such as silhouette, shading, texture, etc. could also be used but for navigation and localization depth information is crucial. Binocular disparity refers to a small positional difference between corresponding images features in the two eyes, and arises because the two eyes are separated horizontally. Depth perception based on binocular disparities is known a stereopsis.

4.1 Camera Calibration

Geometric camera calibration, also referred to as *camera resectioning* estimates the parameters of a lens and image sensor of an image or video camera. These parameters can be used to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in scene. Camera parameters include intrinsics, extrinsics, and distortion coefficients. To estimate camera parameters, there is a need of 3D world points and their corresponding 2D image points. These correspondences can be obtained using multiple images of a calibration pattern, such as a checkerboard.

The calibration algorithm calculates the camera matrix using extrinsic and intrinsic parameters. The extrinsic parameters represent a rigid transformation from 3D world coordinate system to the 3D world cameras coordinate system. The intrinsic parameters represent a projective transformation from the 3D cameras coordinates into the 2D image coordinates.

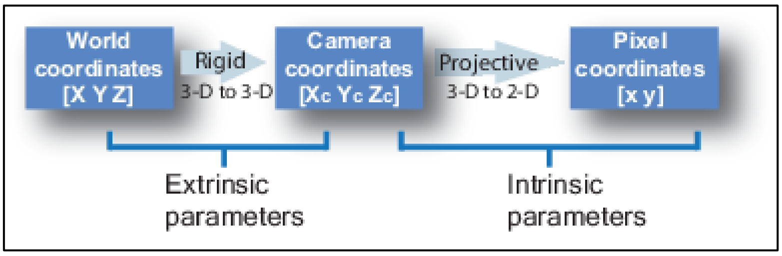


Figure 4-1: Conversion of world coordinates to pixel coordinates

Extrinsic parameters consist of a rotation, R and a translation t. The origin of the cameras coordinate system is a its optical center and its x- and y- axis define the image plane.

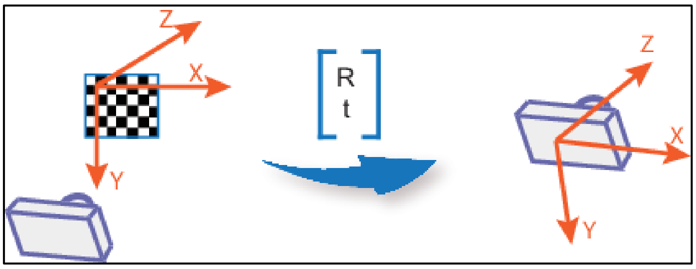


Figure 4-2: Camera Calibration

Intrinsic parameters include focal length, the optical center, also known as principal point and the skew coefficient. The camera intrinsic matrix, K is defined as:

$$\begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix}$$

$[c_x, \ c_y]$ – *Optical center in pixels*

$(f_x, \ f_y)$ – *Focal length in pixels*

s – *Skew coefficient, which is non-zero if the image axes are not perpendicular.*

The camera matrix does not account for lens distortion because an ideal pinhole camera does not have a lens. To accurately represent a real camera, the camera model includes the radial and tangential lens distortion.

(a) Radial Distortion

Radial distortion occurs when light rays bend more near the edges of a lens than they do at its optical center. The smaller the lens, the greater the distortion.

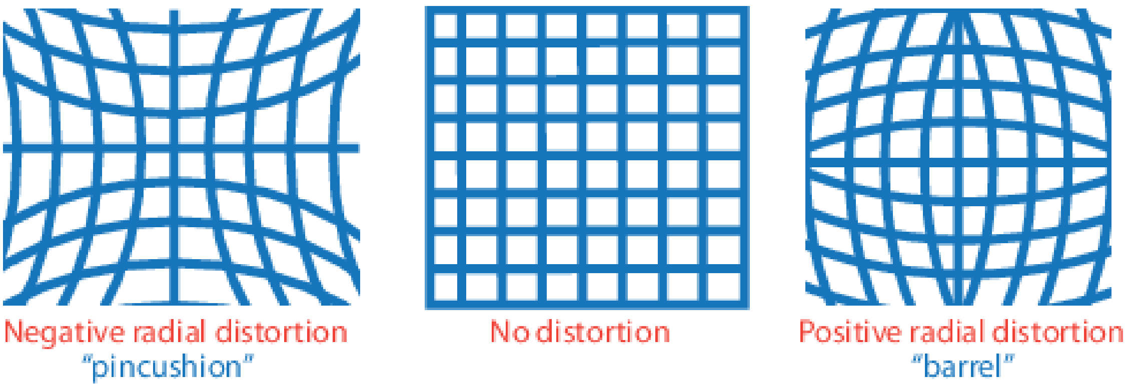


Figure 4-3: Different types of distortion in images

The radial distortion coefficients model this type of distortion. The distorted points are denoted as ($x_{distorted}$, $y_{distorted}$):

$$x_{distorted} = x\,(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

$$y_{distorted} = y\,(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

x, *y* — Undistorted pixel locations. *x* and *y* are in normalized image coordinates. Normalized image coordinates are calculated from pixel coordinates by translating to the optical center and dividing by the focal length in pixels. Thus, *x* and *y* are dimensionless.

*k*1, *k*2, and *k*3 — Radial distortion coefficients of the lens.

$r^2 : x^2+y^2$

Typically, two coefficients are sufficient for calibration. For severe distortion, such as in wide-angle lenses, you can select 3 coefficients to include *k*3.

(b) <u>Tangential Distortion</u>

Tangential distortion occurs when the lens and the image plane are not parallel. The tangential distortion coefficients model this type of distortion.

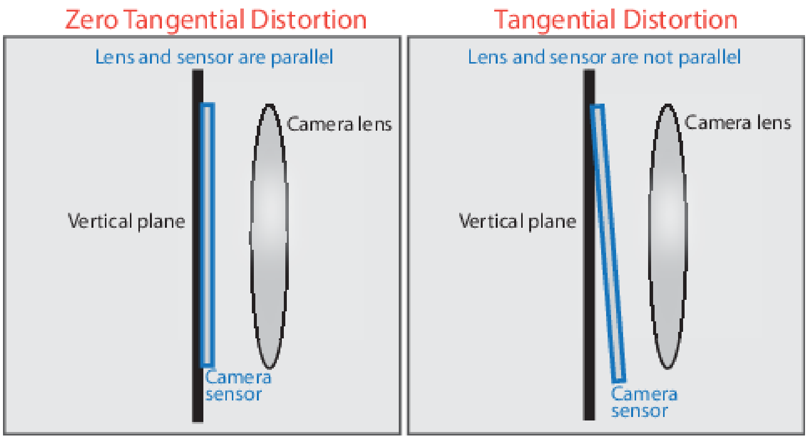


Figure 4-4: Types of Tangential Distortion

The distorted points are denoted as ($x_{distorted}$, $y_{distorted}$):

$$x_{distorted} = x + [2 * p_1 * x * y + p_2 * (r^2+2 * x^2)]$$

$$y_{distorted} = y + [p_1 * (r^2 + 2 * y^2 + 2 * p_2 * x * y)]$$

x, *y* — Undistorted pixel locations. *x* and *y* are in normalized image coordinates. Normalized image coordinates are calculated from pixel coordinates by translating to the optical center and dividing by the focal length in pixels. Thus, *x* and *y* are dimensionless.

*p*1 and *p*2 — Tangential distortion coefficients of the lens.

$r^2 : x^2 + y^2$

4.2 Algorithms in Computer Vision

This section enlists the algorithms which are used in the thesis. A short description of each algorithm has been discussed below.

4.2.1 Image Segmentation

Thresholding is used for one of the most basic techniques of image processing i.e. Image Segmentation. The thresholded image gives segments inside the image, each representing some data associated with the environment which is called as feature and this process is called as feature extraction. The algorithm works in such a fashion that as per the threshold value the part of interest has been assigned value 1 and remaining part gets 0 value.

4.2.2 Color Detection

Detection of colors is very easy for human beings, but for robots? In computer vision, to extract a certain color from the environment color detection algorithms are used on the videos captured by the camera sensors.

Differential RGB is one such algorithm which can be used for the efficient detection of red, blue and green colors. The algorithm works in below fashion.

1. First split the RGB image into three individual channels.
2. Perform following operations using these three channels:
   - R’ = R – G
   - G’ = G – B
   - B’ = B - R
3. Now, it’s time to merge the new (R’, G’, B’) channels.

Variation with light intensity is very less compared to HSV. This algorithm is 5 time faster than normalized RGB algorithm. Color object tracking using this colorspace is accurate.

4.2.3 Harris Corner Detection

This algorithm as the name suggests, is used to detect the corner points in the image. A corner can be defined as the intersection of two edges. A corner can also be defined as a point for which there are two dominant and different edge directions in a local neighborhood of the point. The algorithm works as follows.

$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2$$

Here, I is the intensity of pixels. E is the intensity variation and (x, y) is the position of pixels. As we are searching for that point which have corners, we have to maximize the above equation. Using Taylor expansion,

$$E(u,v) = \sum_{x,y}[I(x,y) + uIx + vIy - I(x,y)]^2$$

$$E(u,v) = \sum_{x,y} u^2I_x^2+2uvI_xI_y+v^2I_y^2$$

$$E(u, v) = [u\ v]\ M\ [u\ v]^T$$

$$R = det(M) - k(trace(M))^2$$

A window with a score R greater than a certain threshold value will be considered as a corner.

5. State Estimation

Robotics inherently deals with things that move in world. The state of a robot is a set of quantities such as position, orientation and velocity, that, if known, fully describe the robot's motion over time. Here in this section the focus is entirely on the problem of estimating the state of a robot.

5.1 Localization Techniques

Navigation is one of the most challenging competencies required for any robot. Success in navigation requires success at the four building blocks of navigation.

1. Perception- the robot must interpret its sensors to extract meaningful data;
2. Localization- the robot must determine its position in the environment;
3. Cognition- the robot must decide how to act to achieve its goals;
4. Motion control- the robot must modulate its motor outputs to achieve the desired trajectory.

In this section, localization techniques have been discussed in a general perspective.

5.1.1. GPS based localization

GPS is a global positioning system. It is a satellite-based radio navigation system. It is a global navigation satellite system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. Position w.r.t. General global reference frame can be obtained using this technique. This becomes the advantage along with the need of processor having moderate processing power. But any robotic system need a precise and accurate location to work in proper fashion. GPS gives and accuracy within 10m that too with a precision of 90%. GPS system is thus very useful to track vehicles as it is powerful and cheap for longer travelling. But in case of robots even a small movement is also of utter importance. So, this technique is not very reliable.

5.1.2 Odometry based localization

Odometry is the use of data from motion sensors to estimate change in position over time. It is used in robotics by some legged or wheeled robots to estimate their position relative to a starting location. Generally, wheel encoders are used to gather the data for the Odometry based localization. This method is sensitive to errors due to the integration of velocity measurements over time to give position estimates. Rapid and accurate data collection, instrument calibration and processing are required in most cases for odometry to be used effectively.

5.1.3 Vision based localization

Vision based localization uses computer vision algorithms for state estimation and cameras for the vision purpose. The most efficient way is marker based vision systems. A marker is a visual tag which have variety of features which can be extracted using computer vision algorithms. These features are further used for state estimation with reference to the camera frame. April robotics laboratory in University of Michigan had developed one such family of markers (named "AprilTag") which has become very popular in recent years due to its vast applications. Ava group of University of Cordoba in Spain has developed ArUco markers.

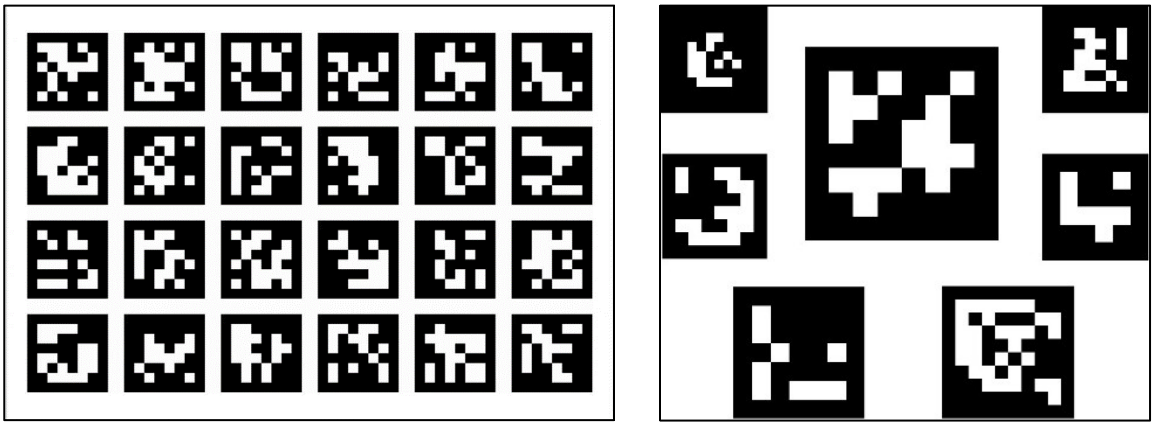


Figure 5-1: Left side - AprilTags & right side - ArUco markers

5.2 Marker Detection

This section will discuss more details about the detection of markers using computer vision techniques. Any marker has a shape and some features inside that shape. Just like AprilTags or ArUco markers have a square shape and some pattern of white blocks inside the square. Color of marker acts like bottom level of hierarchy, shape of marker is the middle level and the pattern on the marker as the top level. The information is first extracted from the bottom level, then from the middle level and at the end from the top level which results into the localization of marker with respect to camera frame. The algorithm works in the following way.

1. Color detection algorithms are used to find possible positions of the marker in image.
2. Corner and edge detection algorithms are used to find the corners and edges in the image obtained in step 1.
3. These corners and edges are used to find the exact position of the marker from the image. (For ex. Square marker will always have four edges and four corners)
4. The pattern on the marker has two functions.
    a) After step 3, if two locations are found in the image for marker then this design is used to verify position of marker.
    b) The orientation of marker in the image has been decided by the this design.
5. Using the information about marker position (in pixels) and orientation in image and the parameters of camera, we can find the world coordinates of that marker with respect to frame of reference associated with marker.

5.3 State Estimation of Snake Robots

Vision based localization techniques are used for the localization and state estimation of snake robots. Four different markers are placed on the skeleton of snake robot as shown in figure below.

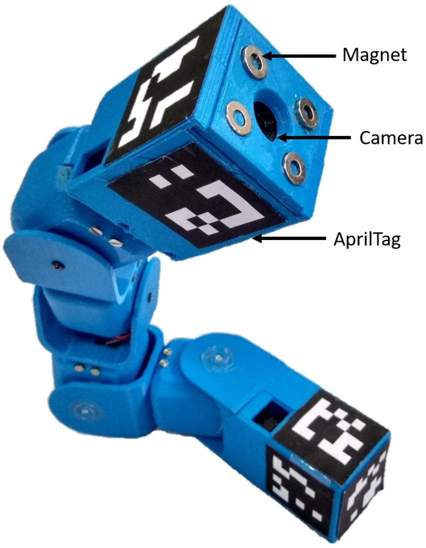


Figure 5-2: Position of AprilTags on snake robot

Also, a camera is placed in the head section of snake robot. So, this is the state of art for the localization of multiple snake robots. Each snake robot has a camera and markers mounted on it. A continuous stream of video will be provided to the processor of snake robot by its own camera. The processor uses the marker detection algorithm (discussed above) for the detection of marker. As soon as the marker is detected, processor can find the position of the marker w.r.t camera frame. So, the snake robot now has the information about the position of other snake robot from its own frame of reference in head section at camera.

5.4 Results

Marker detection algorithm is tested with the colored marker. OpenCV library is used to implement the basic functions of computer vision. A blue colored square marker is used with a red colored L shaped design in it. The following image shows the extracted position of marker from the input image.

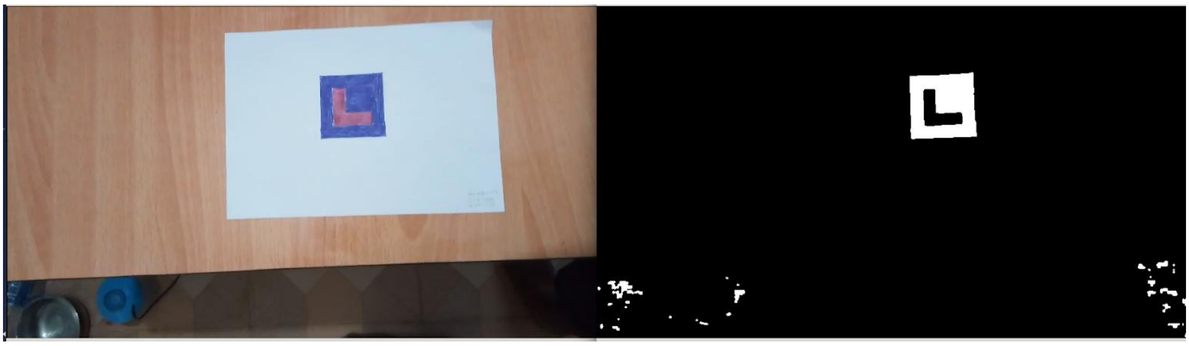


Figure5-3: Left side - Input image. Right side - Marker detected image

6. RESULTS

Self-Reconfigurable Transformer Robot was successfully designed and fabricated. The kinematic model of the robot was also implemented. The team was able to demonstrate various attainable morphological transformations and their respective locomotion gaits. Results were confirmed through implementation on real hardware and simulations. The team was also able to successfully solve the localization problem using computer vision techniques.

Thus, a modular self-reconfigurable transformer robot is designed; capable of self-reconfiguring itself into various morphologies in accordance with the tasks at hand in situ. Section 3.4.2. shows various attainable morphologies and scope of the system. Results of different gaits are shown in section 3.4.4. The results of localization are shown in section 5.4.

7. FUTURE SCOPE

The modularity in Self-Reconfigurable Transformer Robot platform enables wide scope of future research in the system.

The future scope of this project involves training the robot to distinguish various terrains and to select right locomotive transformation through Deep Learning.

Other research areas include integration of Kinect or LiDAR sensors for 3D Vision and Simultaneous Localization And Mapping (SLAM).

Work on swarm and distributed control can also be explored using the current system.

On the industrial application sides, the Self-Reconfigurable Transformer Robot can be made waterproof.

8. BIBLIOGRAPHY

1. S. Kalouche, D. Rollinson and H. Choset, "Modularity for maximum mobility and manipulation: Control of a reconfigurable legged robot with series-elastic actuators," 2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), West Lafayette, IN, 2015, pp. 1-8.
2. B. Shirmohammadi, C. J. Taylor, M. Yim, J. Sastra and M. Park, "Using Smart Cameras to Localize Self-Assembling Modular Robots," 2007 First ACM/IEEE International Conference on Distributed Smart Cameras, Vienna, 2007, pp. 76-80.
3. R. Thakker, A. Kamat, S. Bharambe, S. Chiddarwar and K. M. Bhurchandi, "ReBiS - Reconfigurable Bipedal Snake robot," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 309-314.
4. D. Pongas, M. Mistry and S. Schaal, "A Robust Quadruped Walking Gait for Traversing Rough Terrain," Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, 2007, pp. 1474-1479.
5. J. Davey, N. Kwok and M. Yim, "Emulating self-reconfigurable robots - design of the SMORES system," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 4464-4469.
6. M. Mutlu, S. Bonardi, M. Vespignani, S. Hauser, A. Bernardino and A. J. Ijspeert, "Natural user interface for lighting control: Case study on desktop lighting using modular robots," *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, New York, NY, 2016, pp. 288-293.
7. L. Barrios, T. Collins, R. Kovac and W. M. Shen, "Autonomous 6D-docking and manipulation with non-stationary-base using self-reconfigurable modular robots," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, 2016, pp. 2913-2919.
8. C. Gehring, S. Coros, M. Hutter, M. Bloesch, M. A. Hoepflinger and R. Siegwart, "Control of dynamic gaits for a quadrupedal robot," *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, 2013, pp. 3287-3292.
9. M. Vespignani, E. Senft, S. Bonardi, R. Moeckel and A. J. Ijspeert, "An experimental study on the role of compliant elements on the locomotion of the self-reconfigurable

modular robots Roombots," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, 2013, pp. 4308-4313.

10. T. Tosun, J. Davey, C. Liu and M. Yim, "Design and characterization of the EP-Face connector," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, 2016, pp.45-51.
11. P. Liljebäck, Ø. Stavdahl, K. Y. Pettersen and J. T. Gravdahl, "A modular and waterproof snake robot joint mechanism with a novel force/torque sensor," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, 2012, pp. 4898-4905.
12. B. Shirmohammadi, C. J. Taylor, M. Yim, J. Sastra and M. Park, "Using Smart Cameras to Localize Self-Assembling Modular Robots," *2007 First ACM/IEEE International Conference on Distributed Smart Cameras*, Vienna, 2007, pp. 76-80.
13. V. Vonásek and J. Faigl, "Evolution of multiple gaits for modular robots," *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens, 2016, pp. 1-8.
14. C. S. S. Reddy, S. patlolla, A. Agrawal and K. R. Anupama, "SQ-BOT - a modular robot prototype for self-reconfiguring structures," *2016 International Conference on Robotics: Current Trends and Future Challenges (RCTFC)*, Thanjavur, 2016, pp. 1-6.
15. M. Yim *et al*., "Modular Self-Reconfigurable Robot Systems [Grand Challenges of Robotics]," in *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 43-52, March 2007.
16. E. Olson, "AprilTag: A robust and flexible visual fiducial system," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 3400-3407.
17. Shengyong Chen, Xiuli Li, Ke Lu, Yinfeng Fang, and Wanliang Wang, “Gait, Stability and Movement of Snake-Like Robots” in *International Journal of Advanced Robotic Systems, vol.* 9, no. 5, pp. 103-108, April 2018