Quadruped Simulation

Project Report

EKLAVYA MENTORSHIP PROGRAMME

At

SOCIETY OF ROBOTICS AND AUTOMATION,
VEERMATA JIJABAI TECHNOLOGICAL INSTITUTE
MUMBAI

JUNE 2020

ACKNOWLEDGMENT:

Foremost, We would like to express our gratitude to SRA VJTI – Eklavya 2020 for motivating us to undertake a project and provide us support throughout the course of time.

Our sincere thanks to all our mentors :

Dhruva Gole
Jitesh Sonkusare
Shamit Savant
Omkar Bhilare
Lukesh Ankamvar

Thank you for constantly helping us.

1. Chaitravi Chalke
   chaitravi007@gmail.com
   9137824336
2. Sakshi Chikshe
   sakshi.chikshe.1@gmail.com
   8262962662

TABLE OF CONTENTS

1. Project Overview:

The project is about modelling a 4-legged robot which can move forward. The project uses A Dog-Shaped Quadruped Model. Quadrupeds are considered stable and comparatively easier to program than hexapod, They are more stable than biped robot. In this project, we have programmed a quadruped using the creep gait algorithm to move it in the forward direction. Creep Gait Algorithm keeps the centre of gravity (C.O.G) inside the triangular base formed by three legs when one leg is moving. CoppeliaSim software is used to simulate this model with Lua Programming. The project uses Coppeliasim (V-rep) Software for simulation of motion of robot.

2. Introduction:

Domains:
Forward and Inverse Kinematics
Gait Analysis and Generation
Coding

2.1 Introduction

Nature is made up of wonderful creatures and human beings have always been curious, interested or excited about their behavior and have tried to understand, enjoy or imitate them. Emulating live creature performances is an attractive idea but extremely difficult to accomplish. Generally, we have to settle for building some simple apparatus that can imitate only minute aspects of what we ordinarily sense from our surroundings: creatures that can see, smell, manipulate, and . . . walk.

2.2 Walking Mechanisms

The first walking mechanism, as mentioned above, was built around 1870 by Chebyshev. It was based on a system he had devised about two decades earlier. It consisted of a device based on four bars, placed in such a way that when Link 1 rotates around axis A1, perpendicular to the sheet, the foot (and also point P1) follows a quasi straight-line trajectory, T1, at certain times during the cycle, and it moves off the ground during the rest of the cycle, T2. The shape of the trajectory and the quality of the straight line, T1, depend on the link lengths. The trajectory has been obtained for A1A2 = 0.15 m, A2A4 = 0.41 m, A3A4 = 0.4 m, A3P1 = 0.9 m and distance A1A2 = 0.3 m.

2.3 Gait Generation

In the English language, gait is defined as a way or manner of moving on foot. In the field of legged locomotion, a gait is defined as a repetitive pattern of foot placements (Todd, 1985). A more precise description was made by Song and Waldron (1989), as follows.

Definition 3.1. A gait is defined by the time and the location of the placing and lifting of each foot, coordinated with the motion of the body in its six degrees of freedom, in order to move the body from one place to another.

The first attempts to define mathematical models for gaits were carried out by McGhee and Frank.

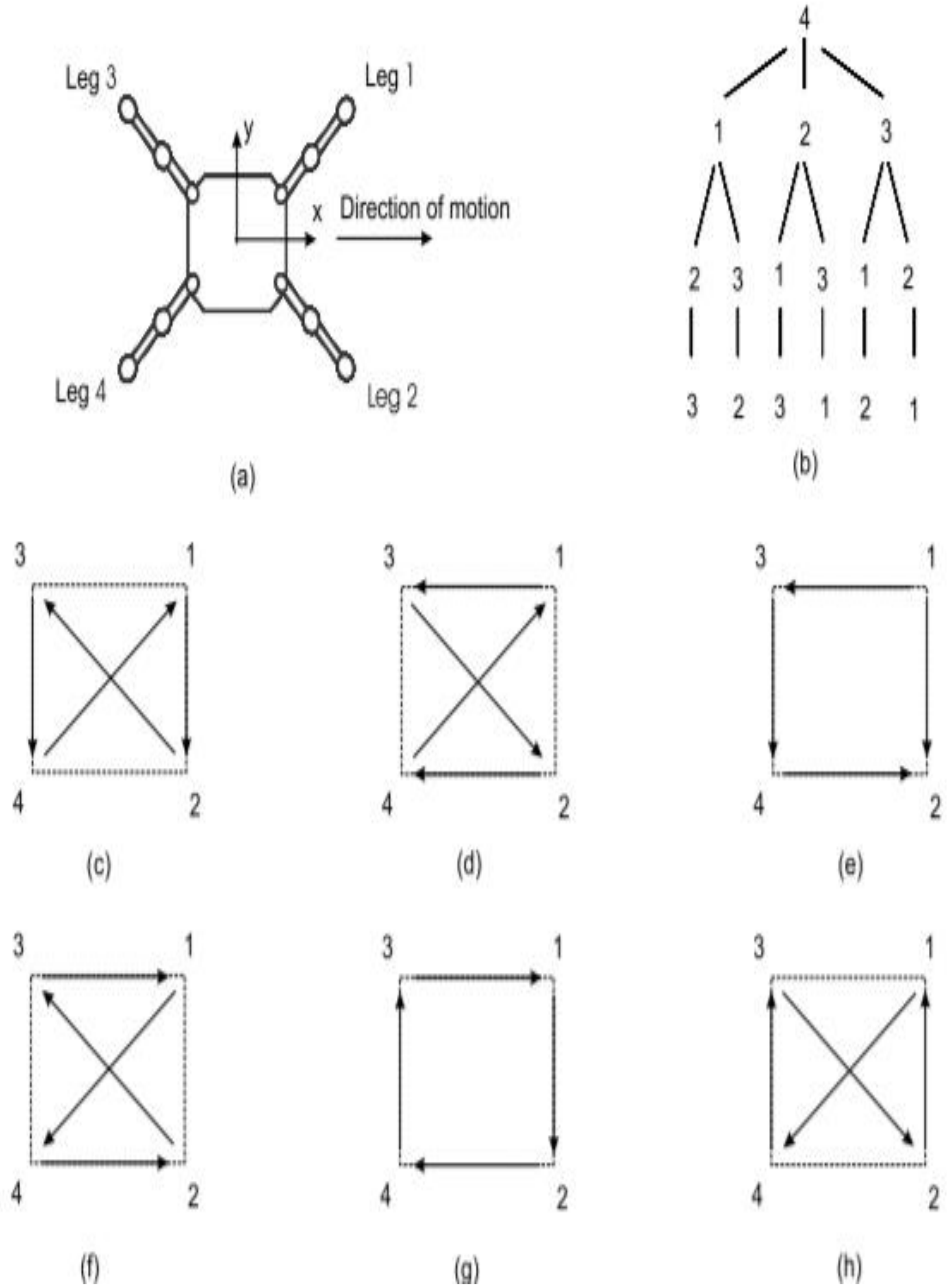


Fig. 3.1. Quadruped gaits: (a) top view of a robot; (b) graph of leg sequences; (c)– (h) sequences

These researchers tried to discover which quadruped gaits were able to maintain static stability. For this study, McGhee introduced a notation called the event sequence. An event is defined as a foot placement or a foot lifting. For an n-legged robot, the placement of foot i is denoted by event i, while the lifting of foot i is denoted by event i + n. Thus, a gait is expressed as a sequence of events such as 2-4-5-7-3-1-8-6, creating 2n different events. If two events occur at the same instant, the gait is termed singular gait, as opposed to totally ordered gaits, or non-singular gaits. The number of possible non-singular quadruped gaits is the permutation of 2n events; that is 2n!. Considering the number of event sequences starting from a given one, the result is N = (2n – 1)!; for a quadruped N = 5040 (McGhee,1968). N represents the number of possible permutations of foot events. However, for a quadruped to maintain static stability, it must keep at least three legs in support, i.e. just one leg in transfer. This means that after the lifting of leg i (event i + n) the placement of leg i must occur (event i). This feature drastically reduces the number of stable combinations to N = (n – 1)!, which for a quadruped results in only six event sequences, demonstrated in Fig. 3.1(b) and illustrated in Fig. 3.1(c)–(h). In this example, the locomotion cycle starts with the movement of foot 4. Leg numbers are indicated in Fig. 3.1(a). The practice of numbering legs front to rear, using even numbers for right legs and odd numbers for left legs, is largely accepted.

Tomovic (1961) defined the gait of an n-legged robot as a creeping gait when every support pattern involves at least n–1 contact points. Hence, gaits in Fig. 3.1 are creeping gaits: there are always three feet in support. Creeping gaits may be either singular or non-singular. Notice that a singular gait can be obtained as the limit of a non-singular gait. Therefore, the event sequences in Fig. 3.1 are applicable to both singular and non-singular gaits. In this case, a singular gait means that the placement of a foot and the lifting of the next leg in the sequence occur at the same time.

Some years after this study, Hirose et al. (1986) found that these six event sequences could be applied to the formulation of quadruped turning gaits. They were classified as ±x type, ±y type (for motion along the x and y axes, respectively) and ±o type (for gaits that rotate the body around the z axis), where '+' means forward motion and '–' means reverse motion (see Fig. 3.1(c)– (h)).

McGhee and Frank (1968) studied the static stability of the six defined creeping gaits, and showed that the optimum static stability margin is achieved by a regular (every foot supported along the same fraction of the locomotion cycle), singular, +x creeping gait (see Fig. 3.1(f)). This creeping gait is sometimes termed a crawl gait. Surprisingly, it is the unique gait used by quadruped animals at low velocities, thus it is known as the standard gait. Notice that there is no generally accepted definition for crawl gait, but crawl and creeping gaits are synonyms for quadrupeds

2.3.1 Continuous Gaits

The continuous gait that is the most widely used by natural and artificial quadrupeds. In this gait formulation, an ideal machine that assumes massless legs is used.The following definitions are also required for gait formulation.

Definition 2 . The duty factor, βi, of leg i is the fraction of the cycle for which it is on the ground. If βi is the same for all legs, the gait is regular.

Definition 3. The leg phase of leg i, Φi, is the normalized time by which the placement of leg i on the ground lags behind the placement of leg 1 (leg 1 is normally considered the reference leg).

Definition 4. The leg stroke, R, is the distance which a foot is moved relative to the body during the support phase. R must be within the leg workspace defined by Rx and Ry.

Definition 5. The stroke pitch, P, is the distance between stroke centers of the adjacent legs. Px is the distance between stroke centers of collateral legs and Py is the distance between stroke centers of contra-lateral legs.

Definition 6. The stride length, λ, of a gait is the distance travelled by the center of gravity COG of the body along a locomotion cycle. If the gait is periodic then

$$\lambda = R/\beta$$

With these definitions, the +x type wave gait is defined by the following leg phases, assuming that the leg workspaces do not overlap, i.e. $R \leq P$

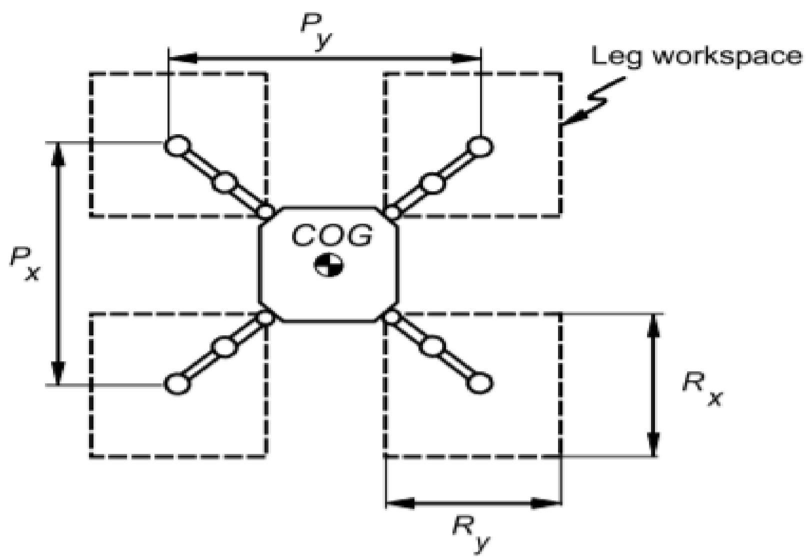


Fig. 3.2. Geometric definitions

$\varphi 1 = 0$
$\varphi 2 = 1/2$
$\varphi 3 = \beta$
$\varphi 4 = F(\beta - 1/2)$
where F is the fractional function defined as
Definition 7. A fractional function Y = F(X) of a real number X is

defined as $Y = \begin{cases} \text{the fractional part of } X \text{ if } X \geq 0 \\ 1 - \text{the fractional part of } |X| \text{ if } X < 0. \end{cases}$

2.3.2 Two-phase Discontinuous Gaits

In generating discontinuous-periodic gaits for quadrupeds, certain aspects should be considered:

1. If one leg in its support phase reaches the rear limit of its workspace (kinematic limit), this leg should change to the transfer phase to be placed at its front kinematic limit.

2. The body is propelled forward with all legs on the ground. After a body motion, at least one leg should stay in its rear kinematic limit to perform a transfer phase into the next leg motion.

3. The leg that is contralateral and non-adjacent (CNA) to the present transfer leg should be placed at such a point that after the placement of the transferred leg, the COG stays on the other side of the line connecting the CNA leg with the transfer leg (see Fig. 3.4). In this way, it will be possible to lift another leg while maintaining the machine's stability.

4. The sequence of legs should be periodic; this will allow several locomotion cycles to be joined to follow a path.

In this section, a gait to move the machine straight forward along the longitudinal x-axis of the machine and under static stability will be considered. That means that the vertical projection of the COG is always inside the support polygon. The longitudinal stability margin, SLSM, will be used as a stability measure.

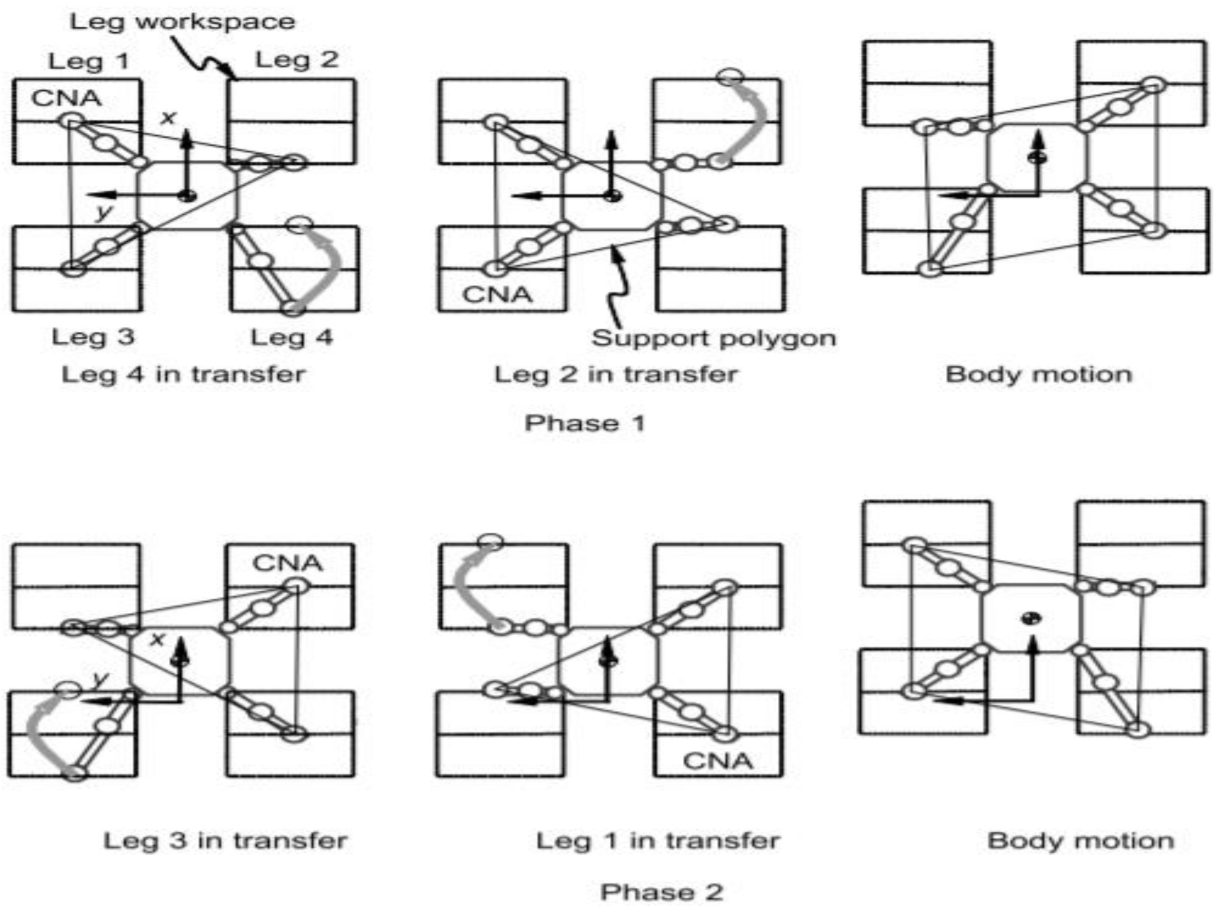


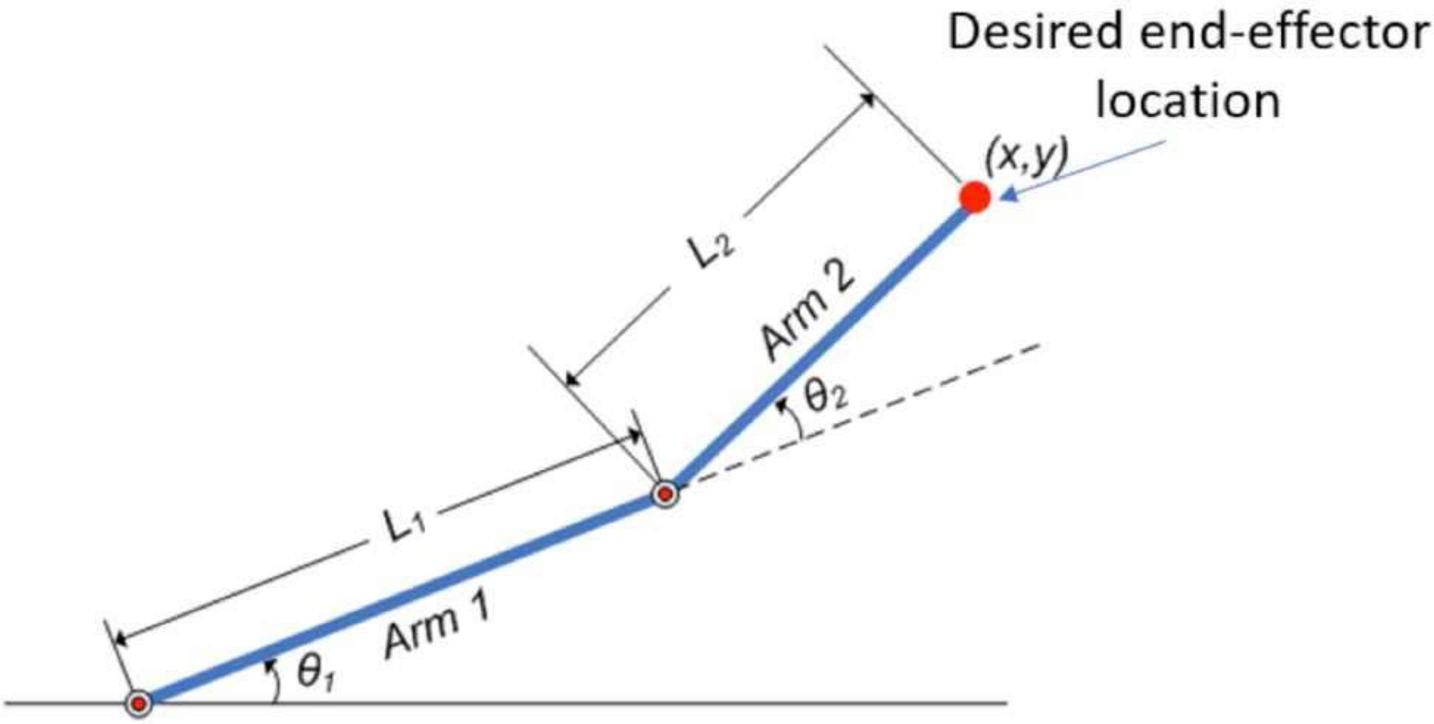
Fig. 3.4. Successive gait pattern of a two-phase discontinuous gait

2.4 Kinematics and Dynamics

Walking robots are very complex mechanical systems, featuring a variable structure defined by its number of degrees of freedom (DOF). For designing algorithms for the control of walking robots it is important to have good models describing the kinematic and dynamic behaviour of the robot. The most popular convention was defined by Denavit and Hartenberg1 in 1955, which is a helpful, systematic way of choosing the reference frame associated with joints and links to derive the kinematic model of an open-loop articulated chain. This method is especially useful when the number of DOF is high. For a 3-DOF mechanism such as a leg, the kinematic model could be easily derived by using trigonometric relationship. However specifying the kinematic model in terms of the D-H method also helps in deriving the dynamic model. The D-H convention and procedure to derive the kinematic model of an open-loop articulated mechanism can be found in books on robot manipulators (Fu et al., 1987; Paul, 1981; Spong and Vidyasagar, 1989; Craig, 1989). In this chapter we will just mention the basic procedure of the D-H method.

3. Kinematics of Walking Robots

The kinematic model describes the relationship between the joint variables of the leg, (q1 , ..., qn)T , and foot position and orientation, (x, y, z, r, p, y)T . In the case of a rotary joint, the joint variable is the angle between the links joined in that joint. In the case of a prismatic or sliding joints, the joint variable is the link extension. The kinematic model of an open-loop articulated chain can be divided into two problems: forward kinematic problem and inverse kinematic problem. The forward kinematic problem gives the position and orientation of the foot, (x, y, z, r, p, y)T , in terms of the joint variables, (q1, ..., qn)T . In contrast, the inverse problem gives the joint variables in terms of the position and orientation of the foot.



3.1 Forward Kinematics: The Denavit-Heartenberg Convention

This section derives the forward kinematic model by using the D-H convention. In fact, this is the description of the procedure following Spong and Vidyasagar (1989).

Let us assume an n-DOF leg. Then:

- The leg has n joints.
- The leg has $n+1$ links numbered from 0 (hip) to n (foot).
- The i-th joint connects link $i-1$ and link i.
- The joint variable of the i-th joint is denoted by q_i. In a rotary joint q_i is the angle between link $i-1$ and link i. In a prismatic joint q_i is the displacement between link $i-1$ and link i.
- Every link i has attached rigidly a reference frame i, which is associated to the joint $i-1$. It is assumed that the hip has attached the inertial frame 0.

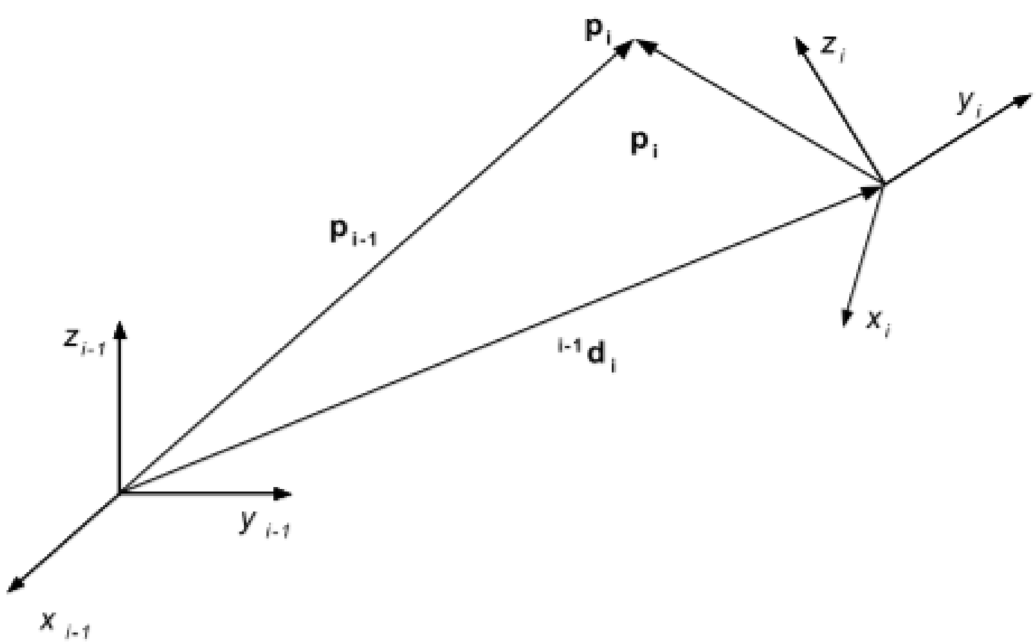
With these assumptions, we have that the coordinates of a point $\mathbf{p}_i$ in the link i are constant with respect to the reference frame i and they do not depend on the leg motion. However, if a rotation occurs between link i and $i-1$, point $\mathbf{p}_i$ is expressed in the reference frame $i-1$ as

$$\mathbf{p_{i-1}} = {}^{\mathbf{i-1}}\mathbf{R_i}\mathbf{p_i} \tag{6.1}$$

where ${}^{\mathbf{i-1}}\mathbf{R_i}$ is the rotation matrix (3×3) that transform a point in the i-th reference frame into the *(i-1)*th reference frame when origins on both frames coincide.

In a more general case, if there is a rotation and translation between reference frames, then (6.1) becomes (see Fig. 6.1)

$$\mathbf{p_{i-1}} = {}^{\mathbf{i-1}}\mathbf{R_i}\mathbf{p_i} + {}^{\mathbf{i-1}}\mathbf{d_i} \tag{6.2}$$



where ${}^{\mathbf{i-1}}\mathbf{d_i}$ is the vector (3×1) that represents the origin of the reference frame i into the reference frame $i-1$. Equation (6.2) can be expressed as a matrix product as

$$\begin{pmatrix} \mathbf{p_{i-1}} \\ \mathbf{1} \end{pmatrix} = \begin{pmatrix} {}^{\mathbf{i-1}}\mathbf{R_i} & {}^{\mathbf{i-1}}\mathbf{d_i} \\ \mathbf{0}_{(1\times3)} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{p_i} \\ \mathbf{1} \end{pmatrix}. \tag{6.3}$$

The matrix

$${}^{\mathbf{i-1}}\mathbf{A_i} = \begin{pmatrix} {}^{\mathbf{i-1}}\mathbf{R_i} & {}^{\mathbf{i-1}}\mathbf{d_i} \\ \mathbf{0}_{(1\times3)} & \mathbf{1} \end{pmatrix} \tag{6.4}$$

is termed the homogeneous matrix. Vectors with the form $(p_x\, p_y\, p_z\, \mathbf{1})^T$ are termed homogeneous vectors. Matrix ${}^{\mathbf{i-1}}\mathbf{A_i}$ transforms the coordinates of a point from reference frame i into reference frame $i-1$. This matrix changes with the configuration of the leg but it only depends on the joint variable q_i, *i.e.*

$${}^{\mathbf{i-1}}\mathbf{A_i} = {}^{\mathbf{i-1}}\mathbf{A_i}(q_i). \tag{6.5}$$

The homogeneous transformation matrix associated to the joint i in the most general case can be expressed by (Fu *et al.*, 1987; Paul, 1981; Spong and Vidyasagar, 1989; Craig, 1989)

$${}^{\mathbf{i-1}}\mathbf{A_i} = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{6.6}$$

where ai, αi, θi, and di are the link parameters. Parameters ai and αi are constant for every link, θi is the joint variable for a rotary joint and di is the joint variable for a prismatic or sliding joint, otherwise they are also constants. These joint parameters are defined below. Finally, the homogeneous matrix that transforms the coordinates of a point from reference frame n (foot) into the reference frame 0 (hip) is given by 0An = 0A1 1A2....n−1An.

3.1.1 Denavit–Hartenberg Procedure

The D-H procedure can be stated in the following steps.

Step 1: Locate the joint axes $z_0, ..., z_{n-1}$ along the joint shaft.
- If joint i is a rotary joint, z_i lies along the joint revolution axis.
- If joint i is a prismatic one, z_i lies along the joint translation axis.

Step 2: Set up the hip reference frame. Locate the origin anywhere in the z_0 axis. The x_0 and y_0 axes must be chosen to form a right-hand frame.

Step 3: Perform Steps 4 to 6 for $i= 1, . . . , n-1$

Step 4: Locate the origin o_i.
- If z_i intersects z_{i-1}, then locate o_i at this axis intersection.
- If z_i is parallel to z_{i-1}, then locate o_i at joint $i + 1$.
- If z_i and z_{i-1} are not in the same plane, then locate o_i where the common normal to z_i and to z_{i-1} intersects z_i.

Step 5: Locate x_i-axis.
- If z_i intersects z_{i-1}, then locate x_i in the direction normal to $z_i - z_{i-1}$ plane.
- If z_i does not intersect z_{i-1}, then locate x_i along the common normal between z_i and z_{i-1}through o_i.

Step 6: Define y_i to form a right-hand frame.

Step 7: Establish the foot reference frame (x_n, y_n, z_n)
- z_n lies along z_{n-1}.
- x_n must be normal to z_{n-1} and z_n (x_n intersects z_{n-1}).
- y_i must form a right-hand frame.

Step 8: Get the link parameters a_i, α_i, θ_i, and d_i for every i.
- a_i is the link length or distance along x_i from o_i to the intersection of x_i and z_{i-1}.
- α_i is the link rotation or the angle that z_{i-1} must rotate about x_i to coincide with z_i.
- d_i is the distance between adjacent links or distance along z_{i-1} from o_{i-1} to the intersection of x_i and z_{i-1}. For a prismatic joint, d_i is the joint variable.
- θ_i is the angle between adjacent links or the angle that x_{i-1} must rotate about z_{i-1} to coincide with x_i. For a rotary joint, θ_i is the joint variable

Step 9: Form the matrices ${}^{i-1}A_i$ for $i =1, ..., n$.

Step 10: Form the homogeneous matrix where vector ${}^{i-1}d_i$ gives the foot position in the hip reference frame and ${}^{i-1}R_i$ gives the orientation of the foot reference frame in the hip reference frame.

3.2 Inverse Kinematics

The inverse kinematics consists in determining the joint variables (q1, ...,qn) T in terms of the foot position and orientation. For a 3-DOF leg the problems can be stated as

$$ {}^{0}\mathbf{A}_{n} = {}^{0}\mathbf{A}_{1}{}^{1}\mathbf{A}_{2}\ldots{}^{n-1}\mathbf{A}_{n} = \begin{pmatrix} {}^{i-1}\mathbf{R}_{i} & {}^{i-1}\mathbf{d}_{i} \\ \mathbf{0}_{(1\times 3)} & \mathbf{1} \end{pmatrix} $$

where (x, y, z)T is the foot position and (a11, a21, a31)T , (a12, a22, a32)T and (a13, a23, a33)T are the orientation vectors of the foot. Equation (6.16) represents a system of 12 equations in 3 unknowns, which is difficult to solve
directly in closed form and, when the direct kinematic has a unique solution,
the inverse problem may or may not have a solution. Furthermore, if a solution exits it may or may not be unique.

Closed form solutions rather than numerical solutions are preferable for two reasons: first, they can be solved at a quicker rate and, second, it is easier to choose a particular solution among several possible solutions. There exist several methods to solve the inverse kinematics of an open-loop chain: geometric approach, algebraic method, inverse transform technique. The following section solves the inverse kinematics of the SILO4 leg by using an algebraic approach.

$$ \begin{pmatrix} a_{11} & a_{12} & a_{13} & x \\ a_{21} & a_{22} & a_{23} & y \\ a_{31} & a_{32} & a_{33} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^{0}A_{1}(q_1){}^{1}A_{2}(q_2){}^{2}A_{3}(q_3) $$

An Algebraic Approach to the Solution of the Inverse Equations (6.13)–(6.15) relate
foot positions and joint variables. From (6.13) and (6.14) we obtain
xS1 − yC1 = 0
that is

$$ \tan\theta_1 = \frac{S_1}{C_1} = \frac{y}{x}. \tag{6.18} $$

Therefore if $x \neq 0$ and $y \neq 0$ a solution for θ_1 is

$$ \theta_1 = \arctan 2(y, x)\,. \tag{6.19} $$

If x=0 and y=0, an infinite number of solutions exits for θ_1. In such a case we say the leg is in a singular configuration.

From (6.13) and (6.14) and using the trigonometric relationship

$$ S_j^2 + C_j^2 = 1 \tag{6.20} $$

we can obtain

$$ C_{23} = \frac{xC_1 + yS_1 - a_2C_2 - a_1}{a_3}. \tag{6.21} $$

From (6.15) we get

$$ S_{23} = \frac{z - a_2S_2}{a_3} \tag{6.22} $$

and substituting for C_{23} and S_{23} from (6.21) and (6.22), respectively, into (6.20) we obtain

$$AS_2 + BC_2 = D \tag{6.23}$$

where

$$\begin{aligned} A &= -z \\ B &= a_1 - (xC_1 + yS_1) \\ D &= \frac{2a_1(xC_1 + yS_1) + a_3^2 - a_2^2 - a_1^2 - z^2 - (xC_1 + yS_1)^2}{2a_2}. \end{aligned} \tag{6.24}$$

The equation form at (6.23) is traditionally solved by performing the following change of variables (Craig, 1989):

$$\begin{aligned} A &= r\cos\phi \\ B &= r\sin\phi \end{aligned} \tag{6.25}$$

then

$$\begin{aligned} \phi &= \arctan 2(B, A) \\ r &= +\sqrt{A^2 + B^2} \end{aligned}. \tag{6.26}$$

Equation (6.23) can now be written as

$$\cos\phi\sin\theta_2 + \sin\phi\cos\theta_2 = \frac{D}{r} \tag{6.27}$$

or

$$\sin(\phi + \theta_2) = \frac{D}{r}. \tag{6.28}$$

Using the trigonometric relationship (6.20), we get

$$\cos(\phi + \theta_2) = \pm\sqrt{1 - \sin^2(\phi + \theta_2)} = \frac{\pm\sqrt{r^2 - D^2}}{r}. \tag{6.29}$$

Thus,

$$\tan(\phi + \theta_2) = \frac{D}{\pm\sqrt{r^2 - D^2}} \tag{6.30}$$

and so

$$\theta_2 = -\phi + \arctan 2(D, \pm\sqrt{r^2 - D^2}) \tag{6.31}$$

and substituting the values of ϕ and r defined in (6.26) we obtain

$$\theta_2 = -\arctan 2(B, A) + \arctan 2(D, \pm\sqrt{A^2 + B^2 - D^2}). \tag{6.32}$$

Notice that (6.32) has two solutions. The right solution must be analyzed in terms of mechanical constraints.

Finally, θ_3 can be obtained from (6.21) and (6.22) as

$$\theta_3 = \arctan 2(z - a_2S_2, xC_1 + yS_1 - a_2C_2 - a_1) - \theta_2. \tag{6.33}$$

That equations (6.19), (6.32) and (6.33) provide the inverse kinematic model of the SILO4 leg.

3.3 A Geometric Approach to Solve Kinematics

As it was mentioned above, D-H convention is just a method to solve the forward kinematics in a systematic manner. However, especially for legs or manipulators with few DOF, it could be straightforward to use traditional geometric methods. In this section, the pantographic mechanism is used to illustrate this method although the author's main aim is to provide the kinematic relationships of the pantograph, a device broadly used as a leg in a large number of walking robots (see Figs. 1.5 and 1.8).
The pantograph is a four-bar mechanism with four passive joints (pj) as indicated in Fig. 6.3. Point A is moved along the z axis by using a prismatic device.

Point B is also moved along the x axis by using an additional prismatic device. Motions of points A and B produce the motion of point C, which is linked with the foot when the device is used as a leg. This mechanism is known as planar pantograph and provides 2 DOF. There are two manners of providing the third DOF. The first and maybe the most broadly used pantographic configuration exhibits a rotary joint which shaft coincides with the z0 axis. In this way, the planar pantograph rotates about the z0 axis. The ASV used this mechanism with the z-axis parallel to the longitudinal axis of the body (see Fig. 1.4). The second configuration provides in point B one more prismatic device that moves point B parallel to the y axis (perpendicularly to the sheet). This mechanism is called Cartesian pantograph and provides three independent linear motions in the foot (see Figs. 1.5 and 1.8).

From Fig. 6.3, we can compute the x-foot component as

$$x = L_1 \cos\alpha + L_2 \cos\beta \tag{6.34}$$

and

$$\cos\alpha = \frac{d_x - a_2 \cos\beta}{a_1} \tag{6.35}$$

and thus

$$x = L_1 \frac{d_x - a_2 \cos\beta}{a_1} + L_2 \cos\beta = \frac{L_1}{a_1} d_x + (L_2 - \frac{L_1}{a_1} a_2) \cos\beta. \tag{6.36}$$

If the four main bars satisfy

$$\begin{aligned} L_1 = L_2 = L \\ a_1 = a_2 = a \end{aligned} \tag{6.37}$$

then (6.36) yields

$$x = \frac{L}{a} d_x. \tag{6.38}$$

From Fig. 6.3, the z_0 foot component is[3]

$$z_0 = L_1 \sin\alpha + d_{z0} - L_2 \sin\beta \tag{6.39}$$

and

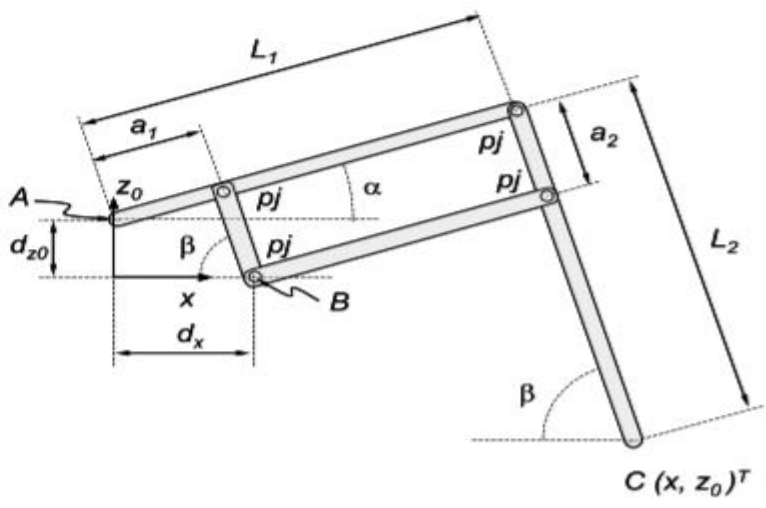


Fig. 6.3. Planar pantographic mechanism

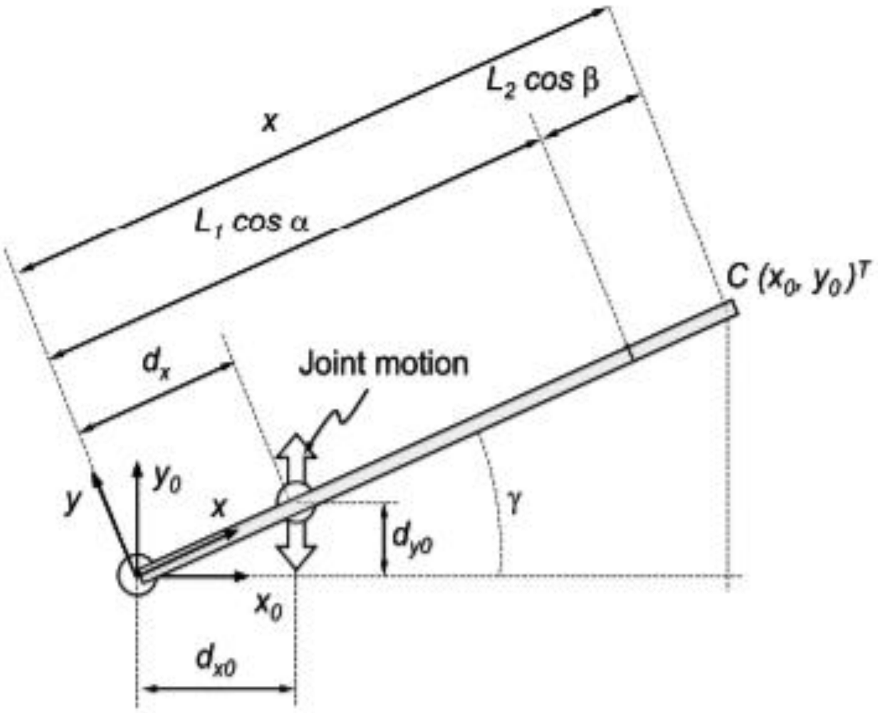


Fig. 6.4. Top view of the Cartesian pantographic mechanism

$$\sin \alpha = \frac{a_2 \sin \beta - d_{z0}}{a_1} \tag{6.40}$$

and thus

$$z_0 = (1 - \frac{L_1}{a_1})d_{z0} + (\frac{L_1}{a_1}a_2 - L_2)\sin \beta \tag{6.41}$$

and for the conditions at equation (6.37) we get

$$z_0 = (1 - \frac{L}{a})d_{z0}. \tag{6.42}$$

Equations (6.38) and (6.42) give the kinematic relationships of the planar pantograph. For a Cartesian pantograph, the y component is actuated by one more prismatic joint. In this case, the actuator moves the point B (passive joint) parallel to the y_0 axis (no matter what the x component is). Figure 6.4 shows a top view of the Cartesian pantograph mechanism. In this figure, we can compute that

$$\sin \gamma = \frac{d_{y0}}{d_x} \tag{6.43}$$

and

$$y_0 = x \sin \gamma = x\frac{d_{y0}}{d_x} \tag{6.44}$$

and using (6.38) we obtain

$$y_0 = \frac{L}{a}d_{y0}. \tag{6.45}$$

Foot components are $(x_0, y_0, z_0)^T$ and joint variables are $(d_{x_0}, d_{y_0}, d_{z_0})^T$; however, (6.38) gives the x-component of the system (x, z_0) contained in the plane of the planar pantograph, where d_x is the joint variable. To compute x_0 as a function of d_{x0}, we have

$$x_0 = x\cos\gamma = \frac{L}{a}d_x\cos\gamma. \tag{6.46}$$

From Fig. 6.4 we obtain

$$d_x = \sqrt{d_{x0} + d_{y0}} \tag{6.47}$$

and

$$\cos\gamma = \frac{d_{x0}}{\sqrt{d_{x0} + d_{y0}}} \tag{6.48}$$

and then

$$x_0 = \frac{L}{a}d_{x0}. \tag{6.49}$$

Equations (6.49), (6.45) and (6.42) can be written in matrix form as

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} = \begin{pmatrix} \frac{L}{a} & 0 & 0 \\ 0 & \frac{L}{a} & 0 \\ 0 & 0 & 1 - \frac{L}{a} \end{pmatrix} \begin{pmatrix} d_{x0} \\ d_{y0} \\ d_{z0} \end{pmatrix}. \tag{6.50}$$

Thus, the components of a Cartesian pantograph are decoupled and each external component only depends on its internal variable. This is one of the advantages of the pantograph mechanism. Another advantage is that the motion of a foot component is the motion of its joint (d_{x0} or d_{y0}) times the factor L/a for x_0 and y_0 components, and d_{z0} times the factor $(1 - L/a)$ for z_0 component.

As the components are decoupled the inverse kinematic can be easily computed, yielding

$$\begin{pmatrix} d_{x0} \\ d_{y0} \\ d_{z0} \end{pmatrix} = \begin{pmatrix} \frac{a}{L} & 0 & 0 \\ 0 & \frac{a}{L} & 0 \\ 0 & 0 & \frac{a}{a-L} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}. \tag{6.51}$$

Interested readers should try to compute the kinematic relationships of a pantograph leg with a rotary joint as the third DOF.

Forward kinematics

The forward kinematic equations of the SILO4 robot, derived in Sect. 6.2, are

$$x = C_1(a_3C_{23} + a_2C_2 + a_1) \tag{A.1}$$

$$y = S_1(a_3C_{23} + a_2C_2 + a_1) \tag{A.2}$$

$$z = a_3S_{23} + a_2S_2. \tag{A.3}$$

where link parameters, a_i, and joint variable, θ_i, are defined in Fig. 6.2 and their values are given in Table (6.1). Let us remember that $C_i = \cos(\theta_i)$, $S_i = \sin(\theta_i)$, $C_{ij} = \cos(\theta_i + \theta_j)$ and $S_{ij} = \sin(\theta_i + \theta_j)$.

Inverse kinematics

The inverse kinematic equations of the SILO4 robot, also derived in Sect. 6.2, are

$$\theta_1 = \arctan 2\,(y, x) \tag{A.4}$$

$$\theta_2 = -\arctan 2\,(B, A) + \arctan 2\,(D, \pm\sqrt{A^2 + B^2 - D^2}) \tag{A.5}$$

$$\theta_3 = \arctan 2\,(z - a_2S_2, xC_1 + yS_1 - a_2C_2 - a_1) - \theta_2. \tag{A.6}$$

Jacobian matrix

The Jacobian matrix of the SILO4 leg used in Chap. 5 is

$$\mathbf{J} = \begin{pmatrix} -S_1(a_3C_{23} + a_2C_2 + a_1) & -C_1(a_3S_{23} + a_2S_2) & -a_3C_1S_{23} \\ C_1(a_3C_{23} + a_2C_2 + a_1) & -S_1(a_3S_{23} + a_2S_2) & -a_3S_1S_{23} \\ 0 & a_3C_{23} + a_2C_2 & a_3C_{23} \end{pmatrix}. \tag{A.7}$$

The derivation of the Jacobian matrix is beyond the scope of this book.

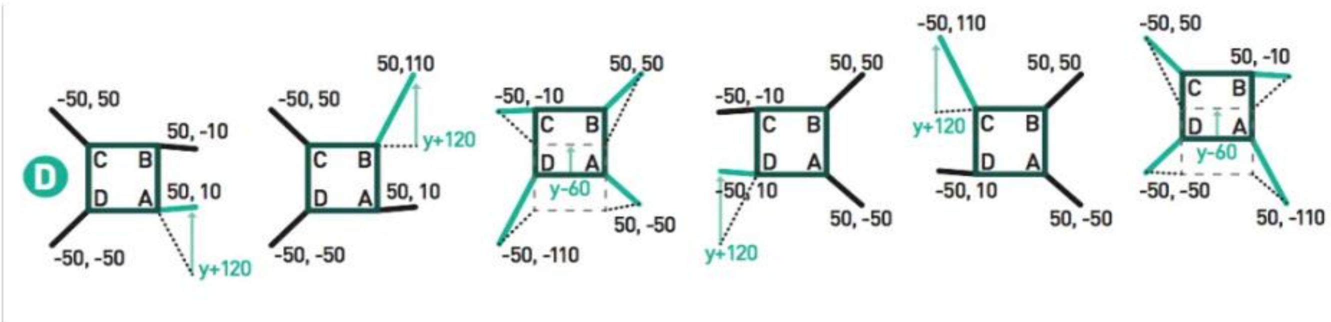
4. Creep Gait Algorithm

4.1 Creep Stability:

The creep gait is “potentially” very stable, since 3 legs form a stable support tripod whenever any one leg is suspended. The simple idea behind the generation of this gait is tp keep the Center of Gravity(CoG) of quadruped inside the triangle formed by the three supporting legs at any given time.

4.2 Creep Gait Algorithm Steps:

1. The starting position, with two legs extended out on one side, and the other two legs pulled inward.
2. The top-right leg lifts up and reaches out, far ahead of the robot.
3. All the legs shift backward, moving the body forward.
4. The back-left leg lifts and steps forward alongside the body. This position is the mirror image of the starting position.
5. The top-left leg lifts and reaches out, far ahead of the robot.
6. Again, all the legs shift backward, moving the body forward.
7. The back-right leg lifts and steps back into the body, bringing us back to the starting position.



5. Coding

5.1 CoppeliaSim

The robot simulator CoppeliaSim, with integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, a ROS or BlueZero node, a remote API client, or a custom solution. This makes CoppeliaSim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab or Octave.

CoppeliaSim is used for fast algorithm development, factory automation simulations, fast prototyping and verification, robotics related education, remote monitoring, safety double-checking, as digital twin, and much more.

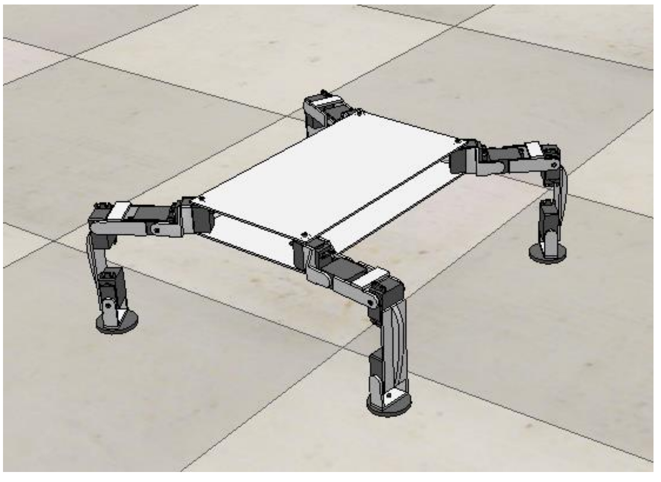
5.2 Lua:

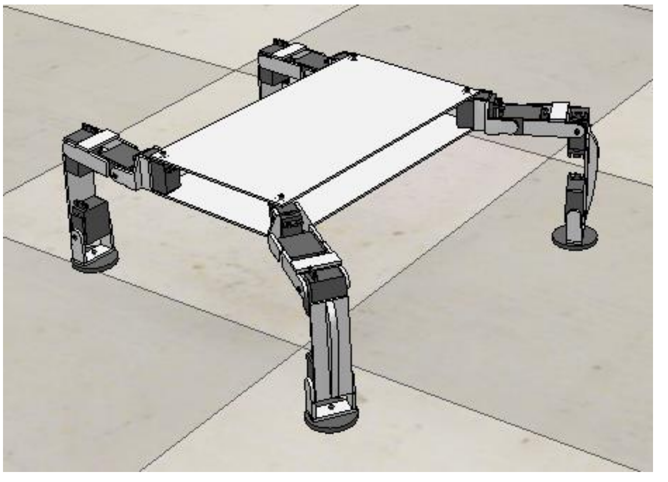
Lua is an open source language built on top of C programming language. Lua has its value across multiple platforms ranging from large server systems to small mobile applications.

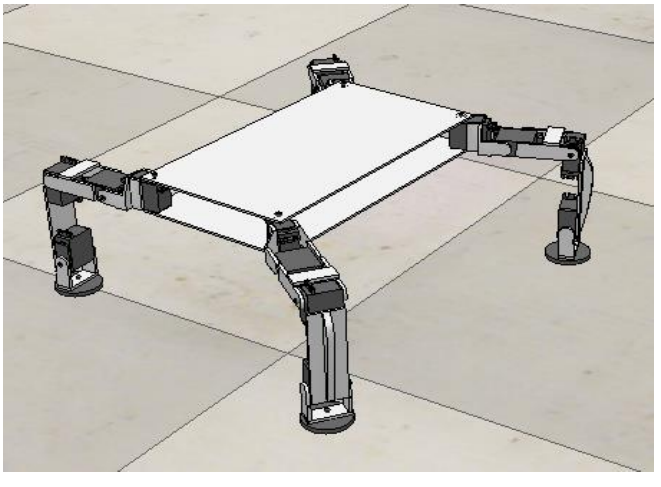
6. Experiments and Results:

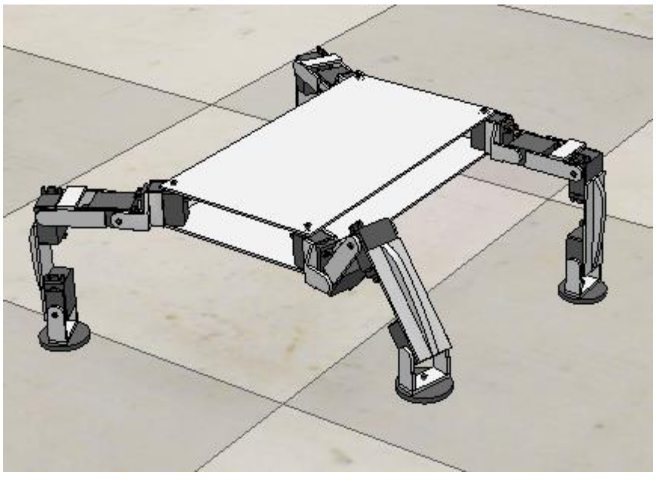
Quadrupeds can be useful in a variety of situations. Main advantage of quadruped over wheeled robots is, the quadrupeds can move on rough terrains whereas wheeled robots cannot. Quadrupeds can also be used on slopes and stairs, where wheeled robots cannot be used. In the project, a model is simulated using CoppeliaSim software and its motion follows creep gait algorithm.

Video of the Simulation:
https://drive.google.com/file/d/14wP4J3AhH0u-MoTKX1THPP_viay8aTmF/view









7. Future Work:

- Obstacle Avoidance
- Reverse Movement of Quadruped
- Left and Right Turns

8. References:

Forward Kinematics

- https://www.youtube.com/watch?v=EzNAs2w1cS0
- https://www.youtube.com/watch?v=x9AR3diUw0A&t=39s
- https://www.researchgate.net/publication/245573479_Kinematics_Analysis_and_Simulation_of_a_Quadruped_Robot

Inverse Kinematics

- https://www.youtube.com/watch?v=VjsuBT4Npvk
- https://www.youtube.com/watch?v=3ZcYSKVDlOc
- https://www.youtube.com/watch?v=JSEPoylypHQ&t=196s
- https://www.youtube.com/watch?v=FiKEAdednuo
- https://www.youtube.com/watch?v=vEpxPfCTwe4&list=PLTOyzR-4RRZmf3qgkdOl_mYa6FV_x5dvF
- https://www.youtube.com/watch?v=f9kxhj5bR6w&t=165s
- https://www.youtube.com/watch?v=P4dGqe-qB6c&t=607s
- https://www.youtube.com/watch?v=Ad5DLd8vrbQ
- https://www.youtube.com/watch?v=MCgMIcolkBk
- https://www.youtube.com/watch?v=O8nzaDcjTiY
- Book by Pablo Gonzalez - Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots

Gait

- https://cjme.springeropen.com/articles/10.1186/s10033-019-0321-2
- https://www.researchgate.net/publication/258177249_The_simplest_creeping_gait_for_a_quadruped_robot
- https://makezine.com/2016/11/22/robot-quadruped-arduino-program/

Coding

- https://www.coppeliarobotics.com/helpFiles/index.html
- http://www.lua.org/pil/
- https://en.wikipedia.org/wiki/Lua_(programming_language)#Syntax
- http://lua-users.org/wiki/