Kinematic Simulator for a Walking Hexapod Robot

Joel Feske and Athma Narayanan

Abstract: Walking robots are more versatile than wheeled or tracked robots, as they can easily traverse rough terrain. In this paper, we describe the development of a kinematic model for use in the simulation of three different walking gaits for a hexapod robot. We discuss the criteria for static stability using the concept of support polygons, and analyze the walking speed of three statically stable gaits as a function of the duty factor β. This simulator allows for better intuition about the performance of the hexapod using various gaits, and will facilitate future work, which will include a dynamical model, and the implementation of a simple control system to handle disturbances to the system.

Introduction

Legged robots have many advantages over wheeled or tracked robots, since they are not limited to roads or level terrain. Additionally, much of the world is already set up for bipeds, and so legged robots would be more suited to daily use as they would not be confounded by existing infrastructure, such as stairs [5]. Drawbacks of legged robots include increased complexity of the kinematic models and control systems needed to maintain stability and execute walking gaits [1]. Many legged animals walk with both statically and dynamically stable gaits depending on the speed at which they are walking; quadrupeds, for instance, are statically stable while walking, but dynamically stable while trotting [3].

Despite the added complexities of legged locomotion, much research is being done into solving the associated problems because of the versatility such robots would have. Applications could be as light hearted as robocup soccer [6], or as serious as search and rescue missions, where robots could be required to traverse rubble or other unpredictable terrain. Carrying shifting loads over such terrain and handling slippage due to imperfect grip will require complex, adaptive controllers.

Our project focuses on building a kinematic simulator for a six-legged robot that will facilitate future work in modelling the dynamics and developing control systems that will allow it to walk on rough terrain. Three statically stable gaits are implemented and compared using the simulator.

Model

We began building our kinematic simulator by modeling a single leg as a three degree of freedom manipulator, shown in Figure 1. Each leg has three segments anatomically analogous to the leg of an insect or spider, referred to hereafter as the coxa, femur, and tibia. The lengths in centimeters of the segments are 1, 3, and 4 respectively, and the body radius is 4 cm.

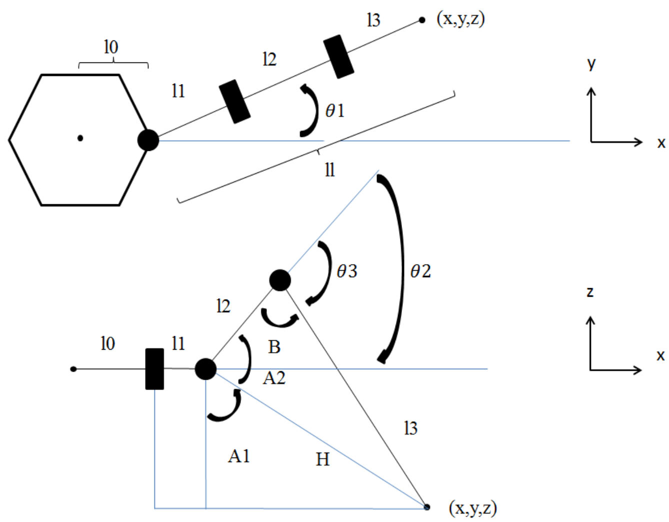

Figure 1: ***Top*** *- view of a single leg from the top, showing the reference position of the coxa joint.* ***Bottom*** *- view of a single leg from the back, showing the relative positions of the coxa, femur, and tibia (l_1, l_2, and l_3 respectively. l_0 is the body radius).*

The inverse kinematics (IK) were initially solved analytically using Paden-Kahan (PK) subproblems. This was relatively trivial, since the leg only has three degrees of freedom, however we found that the IK solution derived from the PK subproblems was overly complex. It required as input not only a position, but an orientation for the foot at each point in the leg trajectory, but since we had only a three DoF leg the joint angles were fully defined after specifying only a foot position. Having a manipulator with fewer than six DoF meant that any deviation from the exactly corresponding foot orientation meant the input position and orientation could not be physically realized, and the IK algorithm would return imaginary or complex leg angles. To avoid this issue, we replaced the PK subproblem solution with a purely geometric one. The equations are as follows:

$$Leg\ Length = ll = \sqrt{y^2 + (x - l_0)^2}$$

$$H = \sqrt{(ll - l_1)^2 + z^2}$$

$$A_1 = \mathrm{atan}(\frac{ll - l_1}{-z})$$

$$A_2 = \mathrm{acos}(\frac{{l_3}^2 - H^2 - {l_2}^2}{-2Hl_2})$$

$$B = \mathrm{acos}(\frac{H^2 - {l_2}^2 - {l_3}^2}{-2l_2 l_3})$$

$$\theta_1 = \mathrm{atan}(\frac{y}{x})$$

$$\theta_2 = \frac{\pi}{2} - A_1 - A_2$$

$$\theta_3 = \pi - B$$

These equations are quite similar to those used by Shahriari and Kambiz [4], and were compared with their equations for accuracy. With the inverse kinematics in place, the forward kinematics were solved using the product of exponential formula so that we could draw the legs in the simulator given the foot positions.

A semicircular step trajectory was chosen for simplicity, and the step length was set at 1 cm.

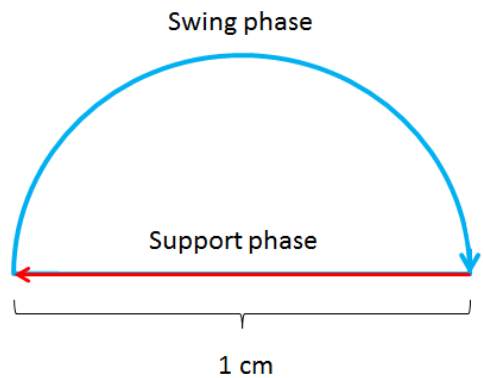

Figure 2: The semicircular trajectory used for a single step. During the swing phase, the leg is in the air moving forward, and during the support phase the leg is on the ground.

In general for a gait to be statically stable, the projection of the center of mass onto the ground must lie within the support polygon formed by connecting all the feet currently in contact with the ground [2,5,6]. This can be achieved with a minimum of three legs touching the ground at any given time. We focus on three gaits, dubbed "tripod," "wave," and "ripple." Figure 3 demonstrates the gait timings.

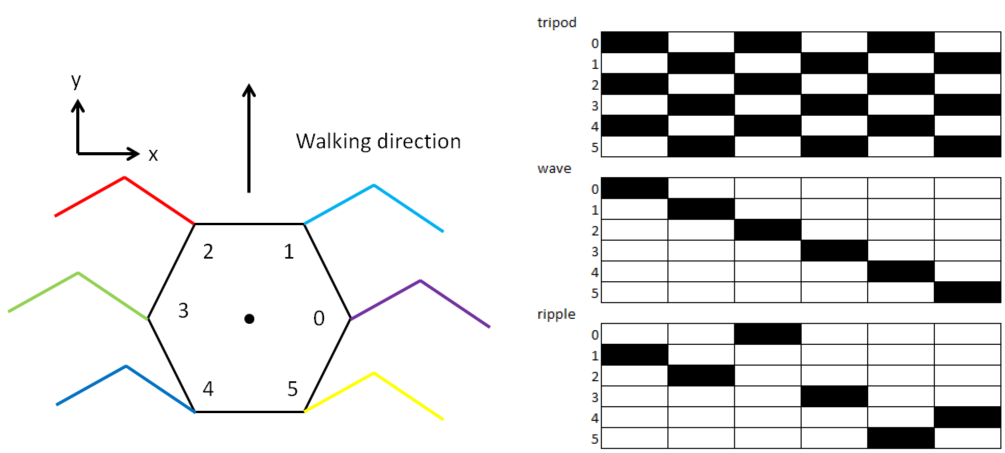

Figure 3: ***Left*** *- The hexapod's legs are numbered counterclockwise from zero, starting with the right, middle leg. This was done so that the zeroth leg would line up with the x-axis when at rest. The zero-based indexing simplified many calculations.* ***Right*** *- A visual representation of the step timings for each gait. Black represents a let in the swing phase, while white represents a leg in the support phase. Note that the apparently erratic pattern in the ripple gait is due only to the leg numbering scheme.*

Results

Each gait has an associated duty factor β which is equal to the ratio of the sides on the support polygon to the total number of legs [6]. The duty factor for each gait can be derived from the chart in Figure 3, since the number of supporting legs for a gait is equal to the number of white cells in each column. Thus, the tripod gait has $\beta = 1/2$, while both the wave and ripple gaits have $\beta =$ □. The duty factor determines the walking velocity resulting from a gait according to the formula:

$$V = \frac{L_s}{t_T}\left(\frac{1-\beta}{\beta}\right)$$

where L_s is the step length (1 cm), T is the cycle time, and the step time $t_T = T(1-\beta)$ is the time for which each leg is in the swing phase [6]. This results in a walking speed for the tripod ($\beta = ½$) that is five times faster than the wave or ripple gaits ($\beta =$ □).

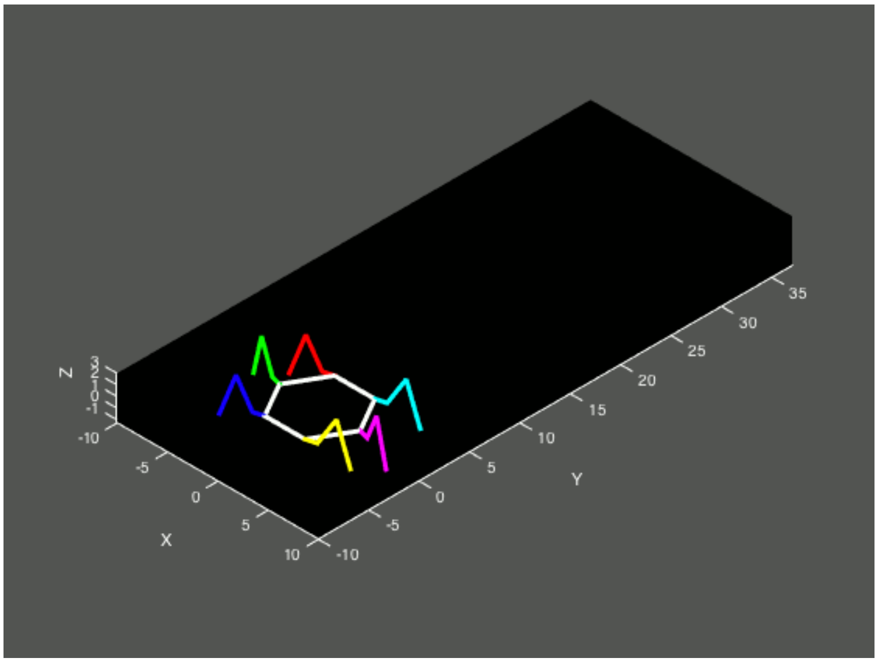

Figure 4: A screenshot of the animation window of the simulator

The final product of this project is a kinematic simulator for a hexapod robot that allows the user to choose a gait, walking distance, and walking direction for the robot, and view an animation of the robot as it walks, as shown in Figure 4.

Discussion

As the duty factor decreases, the walking speed increases, as can be seen in the equation above. Since statically stable gaits require at least three legs on the ground at any given time in order to form a support polygon, the fastest statically stable gaits for an n-legged robot will be the ones that have n-3 legs in the air at all times. Since the duty factor for these gaits is 3/n, it stands to reason that as n is increased, the robot will be able to walk faster while maintaining static stability. Thus, hexapods will be faster than quadrupeds but slower than octopods, etcetera [6].

The immediate next step for this project will be to add a dynamical model of the hexapod to the simulator so that the effects of gravity on the robot can be visualized. This will also allow us to analyze the stability of the robot on slopes, as is done by Mănoiu-Olaru and Niţulescu [5]. A simple controller could then be implemented that would allow the study of disturbances to the hexapod, and its responses while walking with different gaits. Lastly, it would not be hard to randomly generate a “ground” with varying heights at each point that could be used to simulate uneven terrain. At this point, the simulator could be used to tune the controller to perform well on terrain with differing levels of roughness.

References

[1] García-López, M. C., et al. "Kinematic analysis for trajectory generation in one leg of a hexapod robot." Procedia Technology 3 (2012): 342-350.

[2] Kecskés, István, and Péter Odry. "Walk Optimization for Hexapod Walking Robot." Proceedings of 10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics, Budapest, Hungary, November. 2009.

[3] Marhefka, Duane W., et al. "Intelligent control of quadruped gallops."*Mechatronics, IEEE/ASME Transactions on* 8.4 (2003): 446-456.

[4] Shahriari, Mohammadali, and Ghaemi Kambiz. "Kinematic and Gait Analysis Implementation of an Experimental Radially Symmetric Six-Legged Walking Robot." The 6th RoboCup IranOpen International Symposium and the 4th joint conference of AI & Robotics. 2014.

[5] Mănoiu-Olaru, Sorin, and Mircea Niţulescu. "Basic Walking Simulations and Gravitational Stability Analysis for a Hexapod Robot Using Matlab."

[6] Woering, R. "Simulating the “first steps” of a walking hexapod robot." University of Technology Eindhoven (2011).