Nature does nothing in vain.

— Aristotle

7. Modeling and Simulation Results

A good, complete, crawling vehicle solution for the all-terrain vehicle problem requires a capable robot design together with efficient gaits and leg step trajectories. The previous chapters have described the creation of simulation tools for modeling crawling vehicles and simulating their motion. These tools facilitate both hardware design of the physical robot and software design of the robot's motion programs.

In this chapter, the results of a series of test cases are presented to demonstrate the usefulness of the tools created as part of this dissertation work, and to explore some of the design possibilities for the vehicle and its motion programs. An exhaustive search of the design space for the vehicle and its motion programs is not included here. (Such a study could easily be the subject of another dissertation.) However, the majority of examples presented here go beyond simply showing that the simulation works. They also have design significance.

First, the parametric modeling capability of the simulation code is illustrated with rendered examples of several robots that were created using the parametric geometry scheme described in Section 4.1.

Second, the stability analysis functions of the simulation code are used to determine the number of leg pairs required on a robot design to be able to effectively use two

significant wave gaits, namely, the single-leg-pair-per-wave gait and the double-leg-pair-per-wave gait that were discussed in earlier chapters.

Finally, the chapter presents a series of motion programming simulation test cases that compare three different motion programs on a single robot design. These examples will focus on using the simulation tools to explore the benefits of caterpillar-like motion programs in improving locomotion.

7.1 Example Robot Models

As explained in Section 4.1, the simulation programs developed as part of this work allow a wide variety of possible crawling robot designs to be modeled by varying one or more of the 31 geometric parameters that specify each design. Figures 7.1 and 7.2 show four different crawling robots that were modeled and rendered using the simulation code. These test cases demonstrate a few of the design possibilities for the multibody passive-legged class of crawling vehicles. The design shown in Fig. 7.1 will also be used in the motion programming test cases later in this chapter.

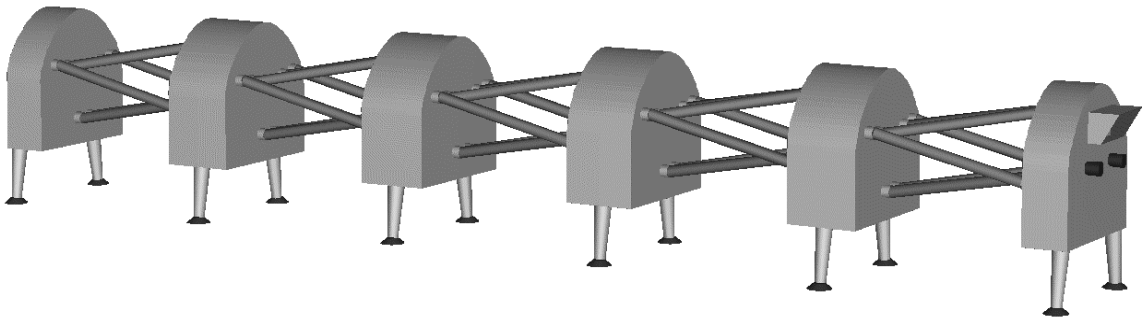


Figure 7.1 — A Crawling Robot Design Rendered with the Simulation Programs.

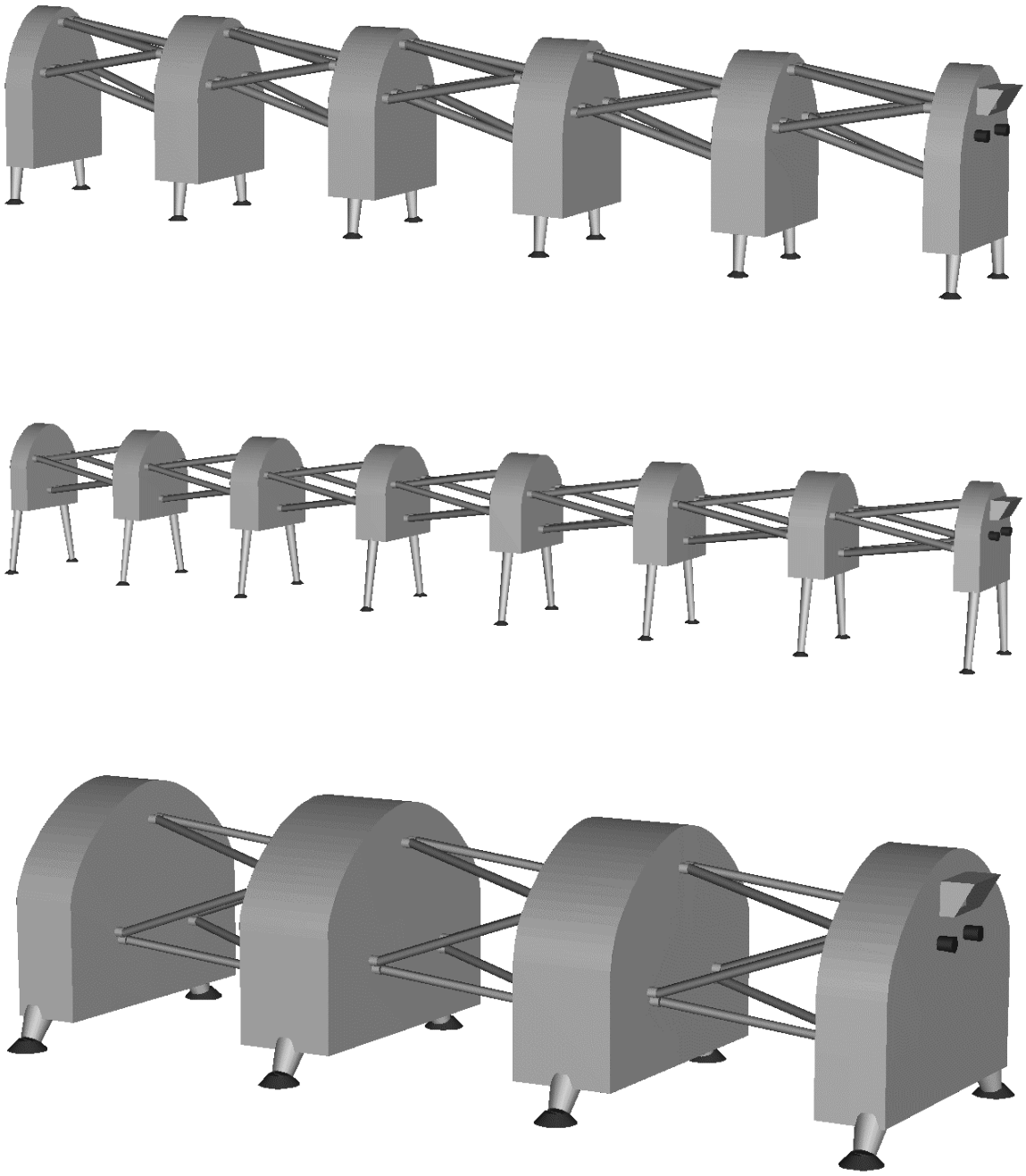


Figure 7.2 — More Example Crawling Robot Designs Rendered with the Simulation Programs. (Note the differing actuation unit arrangements and geometries.)

7.2 The Design Implications of Stability

This section demonstrates the stability margin calculation capabilities of the simulation code and its application to vehicle design. Specifically, the tool is used to compare the stability of two gaits and to determine the minimum number of leg pairs required for a crawling vehicle design to safely use each gait.

Song and Waldron (1989) define the "gait stability margin" for a periodic gait as the minimum longitudinal stability margin over an entire locomotion cycle as the robot performs the gait. They go on to explain that the longitudinal stability margins are minimized at "critical events" during the locomotion cycle – when feet are lifted or placed at either end of the robot, causing the support polygon to abruptly change shape. These "critical events" can be recognized using the simulation code by displaying the stability plots in single-step mode, and then watching the resulting slow-motion animation for the shifts in the shape and size of the support polygon.

The two gaits compared here are the step interval 1.0 wave gait and the step interval 0.52 wave gait. These are essentially what were referred to in Chapter 3 as the "single-leg-pair-per-wave" gait and the "double-leg-pair-per-wave" gait, respectively. Thus, the $\sigma = 0.52$ gait is similar to the gait used by caterpillars. A value of 0.52 was chosen rather than the singular gait value of 0.50 as a safety factor to ensure that, with a real robot and its controller errors, the *n*th leg pair would truly be back on the ground *before* leg pair *n* -1 lifted off of the ground.

Tables 7.1 and 7.2 were created by running the simulation code several times using motion programs that embody the two gaits and using several robot designs that vary only in the number of leg pairs they possess. For these tests, the program was run at a high time resolution so that each time-slice was small, and thus, the snapshot of the worst case longitudinal stability margin would be accurate.

Comparing the two tables it can be seen that, for a given number of leg pairs, the $\sigma = 1.00$ gait has greater longitudinal stability than the $\sigma = 0.52$ gait.

Looking at Table 7.1 we see that a minimum of 4 leg pairs are required to obtain stable locomotion using the $\sigma = 1.00$ gait on flat, level terrain. By adding more leg pairs, the longitudinal stability margin increases, enabling the robot to climb steeper slopes without tipping over. The stability plot diagrams in Table 7.2 demonstrate that the $\sigma = 0.52$ gait requires at least 6 leg pairs to be comfortably stable even on flat, level ground.

Publications by Mahalingham (1988) and Dwivedi and Mahalingham (1990) point out that the load-bearing capacity of the terrain under each foot of a legged vehicle cannot be reliably determined until the foot actually contacts the ground. Shifting rocks, sinkage of soft soils, foot slippage, and structural damage can all cause support failure of a leg. Therefore they stress the importance of redundant stability for legged vehicles. They extend the support polygon concept in the following way:

> It is proposed that the point of intersection of the weight vector with the support polygon be kept within a sub-region of the support pattern such that stability survives the loss of any of the supported legs. This region is termed the conservative support polygon... (Mahalingham, 1988).

Looking at the $\sigma = 1.00$ gait in Table 7.1, we see that, with 4 leg pairs, it is stable, but not redundantly stable. With 5 leg pairs, the gait is redundantly stable. And with 6 leg pairs, it appears that it is redundantly stable to the point that *any* two footholds could be lost at *any time* during its locomotion cycle and the robot would still remain standing. In Table 7.2, we see that the $\sigma = 0.52$ gait requires at least 6 leg pairs to achieve redundant stability by a small margin.

Thus, for terrains with nearly certain load-bearing capacity, such as man-made roads and floors, the $\sigma = 1.00$ gait requires a robot with 4 leg pairs, and the $\sigma = 0.52$ gait requires a robot with at least 5 leg pairs, preferably 6.

Table 7.1 — Worst Case Stability Plots for $\sigma = 1.0, \omega = n$ Gait and Varying Numbers of Leg Pairs

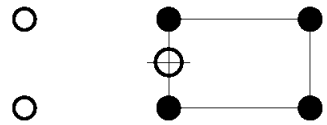
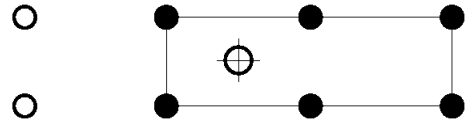
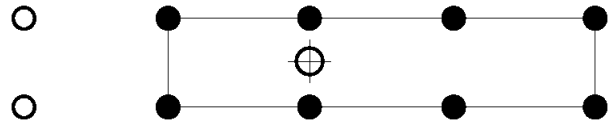
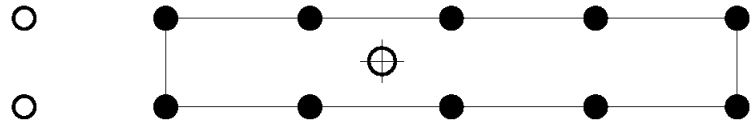
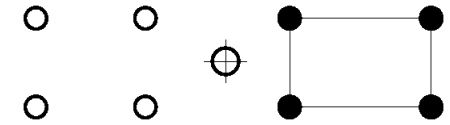
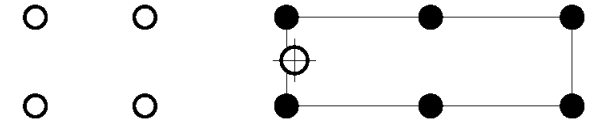
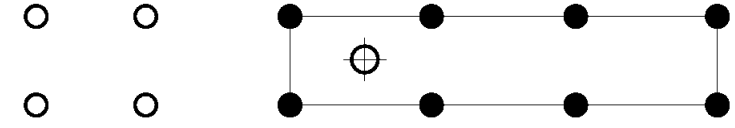
3 Leg Pairs Borderline *UNSTABLE*	
4 Leg Pairs Stable	
5 Leg Pairs Stable	
6 Leg Pairs Stable	

Table 7.2 — Worst Case Stability Plots for $\sigma = 0.52, \omega = n$ Gait and Varying Numbers of Leg Pairs

4 Leg Pairs *UNSTABLE*	
5 Leg Pairs Borderline *UNSTABLE*	
6 Leg Pairs Stable	

For terrains of uncertain load-bearing support, such as most natural surfaces, the $\sigma = 1.00$ and the $\sigma = 0.52$ gaits will require robots having a *minimum* of 5 and 7 leg pairs, respectively, to avoid tip-over failures that could damage the vehicle.

Note that robots having enough leg pairs to perform the $\sigma = 0.52$ gait can also temporarily switch to the $\sigma = 1.00$ gait, or some other gait, to adapt their locomotion and resulting stability as needed to the terrain.

Adding still more leg pairs gives a robot greater longitudinal stability, better climbing ability, better obstacle crossing ability, and the capacity to do multiple wave ($\omega < n$) locomotion, albeit at the expense of greater complexity and financial cost. For locomotion over very rough terrain with steep grades and with soft or rocky soils, it may be desirable to design a crawling vehicle with 8 or more leg pairs. (Recall from Ch. 3 that caterpillars actually have 12 segments with 16 legs.)

7.3 Overview of the Motion Program Test Cases

In addition to the visualization and stability analysis discussed earlier, the kinematics model embodied in the simulation aids the designer in two other significant ways: synthesizing the robot's link lengths for a given robot design and motion program, and providing comparative data for judging between competing robot/motion program design combinations.

The most fundamental use of the simulation is to synthesize the maximum and minimum lengths of all the prismatic links required for a robot to track a specified motion program. If the synthesized link dimensions are possible, then the desired motions are within the workspace of the robot, and the robot geometry and link length information constitute a feasible configuration design for performing the given motion. Hence, this "synthesis approach" fits the robot to the application rather than

fitting the application to the robot. This approach was taken because the ultimate goal is not to develop a robot of particular dimensions, but rather, to develop a locomotion system capable of certain motions.

At this stage in the development of the crawling vehicle, it is desirable to test and compare numerous design possibilities in order to select a high performance configuration design before proceeding to the time consuming and expensive detailed design stage.

Because a complete, detailed design does not yet exist for the crawling vehicle, its mass properties and its actuator force/speed characteristics are not known. Without this information, it is not possible to predict the absolute motions of the actuators or the absolute speed of the vehicle. But while the kinematics-based analysis used here does not yield accurate *absolute* quantitative results, it does yield accurate *relative* qualitative results for comparing competing configuration designs.

The remainder of this chapter will demonstrate this comparative function of the tools by presenting the simulation results of several different gaits and leg step trajectories run on a single crawling vehicle design, and will evaluate and draw conclusions from these examples. Specifically, the gaits used in the motion programs can then be compared based on attributes such as the resulting peak actuator velocities, the resulting peak actuator accelerations, and the margin of static stability. The goal of these tests is twofold: first, to demonstrate the utility of the simulation tools by using them to compare three different motion programs, and second, to determine whether imitating caterpillar gait and leg step trajectories will improve locomotion for the multibody passive-legged crawling vehicle.

The three motion programming test cases are as follows: the $\sigma = 1.00$ gait, the $\sigma = 0.52$ gait, and the $\sigma = 0.52$ gait including pitching motions of the leg pairs. The resulting simulation output includes graphs of link lengths, velocity, and acceleration versus time, and sequences of renderings and stability plots.

7.3.1 Creating a Uniform Basis for Comparing the Motion Programs

In order to accurately compare the motion programs, it was necessary to hold all the other test case variables constant. To accomplish this, the test cases all used the following conditions:

- the same flat, smooth, level terrain
- the same parametric geometry robot design
- the same neutral stance longitudinal leg pair spacing (1.27 meters)
- the same maximum and minimum link lengths
- the same leg-step position curve tangent vectors, which produce the same maximum step height (0.10 meters)
- stride lengths scaled to fully utilize the available workspace
- the same wave interval ($\omega = 6$, for one wave at a time)
- and gaits scaled to propel the robot forward at the same average velocity

7.3.2 The Robot Design for the Test Cases

As mentioned in the previous section, each of the sample runs presented in the remainder of this chapter will be based on a single robot design. As we saw in Section 7.2, a gait with a step interval of 0.52 requires at least six leg pairs to maintain a satisfactory margin of stability. Therefore, the robot used for the motion program test cases was designed to have six leg pairs. A rendering of this robot design was shown in Fig. 7.1. The geometric parameters that define its geometry are listed in Table 7.3.

Table 7.3 — Geometric Parameters for the Tested Robot

Number of leg pairs, N	6
Payload Box Height, H (m)	0.7620
Payload Box Width, W (m)	0.7620
Middle Payload Box Length, L (m)	0.3810
Lateral Space Between Feet, F (m)	0.7620
Ground Clearance, G (m)	0.3048
Joint Offset Length, O (m)	0.0444

Universal Joint Number	Y Location (m)	Z Location (m)
1 (Front Side)	0.152400	0.076200
2 (Front Side)	0.254000	0.091440
3 (Front Side)	-0.050800	-0.228600
4 (Front Side)	0.050800	-0.228600
5 (Front Side)	-0.254000	0.091440
6 (Front Side)	-0.152400	0.076200
7 (Back Side)	0.050800	0.228600
8 (Back Side)	0.254000	-0.091440
9 (Back Side)	-0.152400	-0.076200
10 (Back Side)	0.152400	-0.076200
11 (Back Side)	-0.254000	-0.091440
12 (Back Side)	-0.050800	0.228600

Note: The Y and Z locations of the universal joints connecting the actuated links to the payload boxes are expressed relative to the centroid of each leg pair's payload box. The universal joint X values are derived using the method described in Section 4.2.2.

Note that with the exception of the number of leg pairs and the neutral stance longitudinal foot spacing, this design is the same as that described by Virginia Tech student Paul Mele in his Master's Thesis (1991). Most significantly, the "Mele configuration" Stewart-Gough platform has equal link lengths at any arbitrary neutral stance spacing. That is, whenever any adjacent leg pairs have their **X**-axes aligned to be coincident and have no relative roll angle, then all six links of the connecting Stewart-Gough Platform will have the same length, regardless of the distance between the leg pairs. This property encourages the use of identically sized linear actuators for all the links of each Stewart-Gough platform mechanism. This gives the advantage of interchangeable parts, and may also result in simpler low-level controls and lower cost than having to use two or three different linear actuator sizes.

7.3.3 Placement of the Local Coordinate Frame

Recall from Chapter 4 that the model specifies the positions of the leg pairs relative to the "local" coordinate frame of reference. Since we are dealing with a mobile robot, this is not a fixed relationship. Thus, for the test runs, we can choose any arbitrary position relative to the robot for the local frame. The relationship selected for these test cases is shown in Fig. 7.3.

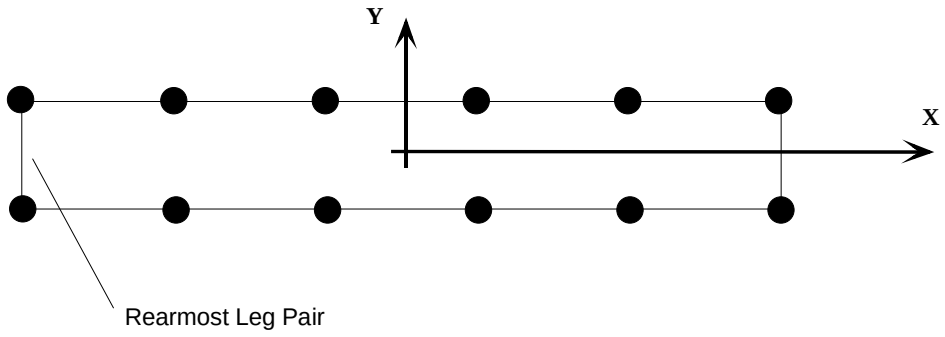


Figure 7.3 — Coordinate System Used for Specifying Leg Pair Trajectory

To keep things simple, the origin of the local frame was placed at the center of gravity of the robot's initial stance. The **X**-axis was aligned with the robot's longitudinal axis, pointing positive to the front of the robot, and the **Z**-axis points up, out of the page.

7.4 Motion Program Test Case Procedure

Creating each test case and running the simulation was a four-step procedure. These steps ensure the validity of the motion program comparison by creating test cases that comply with the list in Section 7.3.1.

Recall from Section 6.4.3 that an assembly constraint, a workspace uniformity constraint, and a workspace lateral symmetry constraint apply to virtually all crawling vehicle designs and motion programs, and thus will apply *individually* to each of these test cases. In order to create motion program test cases that are valid for comparison, three new *collective* constraints are also required. These new constraints apply to all the test cases as a group, and include a workspace equality constraint, a workspace full-range constraint, and an average velocity equality constraint.

7.4.1 The Workspace Equality Constraint

The workspace equality constraint ensures that all the test cases are simulating an essentially identical robot. That is, the motion programs for all the test cases are made so that the resulting range of motion of the robot's actuation units are the same from test case to test case. Thus, the workspace equality constraint imposes an additional restriction that the equivalent links of the robot in **all** test cases must have a maximum length less than or equal to the overall maximum, and a minimum length

greater than or equal to the overall minimum. By "overall" it is meant that the same numerical values for the minimum and maximum lengths for each equivalent link group apply to of all the test cases as a set.

7.4.2 The Workspace Full-Range Constraint

It is possible for two test cases to simulate the same robot with the same maximum potential workspace, but with one of the tests using a subset of that workspace. For example, the same robot could be programmed to walk with strides as long as its workspace would allow, or with short steps. However, using a workspace subset will **not** provide a trustworthy comparison of the motion programs, especially if they use different gaits. A comparison of motion programs is not valid unless each test consistently uses the **full** range of the available workspace.

Therefore a workspace "full-range" constraint must be added to the workspace equality constraint of the previous section. Specifically, in order to obtain valid motion program comparisons, each of the different test case motion programs must result in having the maximum link length of at least one equivalent link group **equal** to the overall maximum for that group **and** having the minimum link length of at least one equivalent link group **equal** to the overall minimum for that group.

Of course in the real world, true "equality" of continuous scalars is virtually impossible. Therefore, when the length values are within a small tolerance value of the ideal, they will be considered "equal". In the four motion program test cases presented later in this chapter, the worst case difference in the minimum link lengths when comparing all of the test cases was less than 0.0007 meters, or 0.14%. In the same manner, the worst case difference for the maximum lengths was 0.0004 meters, or 0.04%.

7.4.3 The Average Velocity Equality Constraint

Two of the main outcomes to be compared in these test cases are the actuator velocities and accelerations required by each motion program to produce the *same* forward velocity of the vehicle. Thus, the average velocity equality constraint applies to all the test cases as a set and ensures that the crawling vehicle moves with the same average velocity in each test case.

The average velocity of the robot is considered to be the average velocity of its center of gravity. Recall from Ch. 5 that equations were presented for determining the vehicle's velocity for both the discrete case of a single locomotion cycle (each leg pair taking one step) and for the continuous case of each locomotion cycle being immediately followed by another. Since the continuous case average velocity is a more realistic measure of crawling vehicle traveling speed, it will be the measure used when imposing the average velocity equality constraint of the test cases.

7.4.4 The Four Step Procedure for Each Test Case

1. Determine the maximum stride length, λ.

   Specifically, write a motion program that specifies the range of motion for a single actuation unit. (Generally, this requires programming the stride of one leg pair for $\sigma \geq 1$ or two leg pairs for $\sigma < 1$.) Recall from Ch. 5 that a motion program for a leg pair trajectory consists of path information (two parametric cubic coefficient matrices) and timing information (the start and stop time for each cubic curve). At this point we are only concerned with the path, so any arbitrary consecutive times can be used for the start and stop times. Use the simulation code to run this short program on the given robot parametric geometry and evaluate the resulting minimum and maximum link lengths to see if they violate the assembly constraint, the workspace symmetry constraint, the workspace

equality constraint, or the workspace full-range constraint. If any of these constraints are broken, rewrite the motion program with a different stride length to compensate, and run it again. By iterating in this manner, a maximum stride length can be found that satisfies the rigorous workspace full-range constraint within a small tolerance value.

2. Compute the stride time, τ, so as to conform to the average velocity equality constraint.

   Equation 5.15 is used to calculate the continuous case average velocity of a crawling vehicle's center of gravity. Rearranging this equation and using the stride length determined in step 1, the step interval, the wave interval, and the specified velocity of the test cases, we can solve for the stride time.

$$\tau = \frac{\lambda}{\sigma \omega s_{cg}} \tag{7.1}$$

3. Write the motion program for a full locomotion cycle.

   The path information for the program comes from using foothold positions (determined by the initial neutral stance spacing and the stride length), the given *xyz* tangent vectors for the cubic curves, and the coordinate system shown in Fig. 7.3. The timing information for the program consists of start times and stop times for each cubic that are determined from the stride time and the gait (step interval). (Note that if ω had been less than n we would have needed to use it as well, but for these test cases it was not necessary.)

4. Run the simulation code.

   Use the simulation code to run the motion program with the given robot geometry and use it to output the data for graphing the link length, velocity, and acceleration, as well as stability plots and rendered image data.

7.5 Test Case #1: Step Interval (σ) = 1.0

For this test case, our 6 leg pair crawling vehicle was programmed so that only one of its leg pairs moved at a time. The gait diagram for this motion program is the same as that shown in Fig. 5.3, except that the number of leg pairs and time scale are different.

This initial test case differed slightly from the later test cases in that the overall constraints for link lengths were not yet defined. Therefore, its stride length was determined by the assembly constraint to be 0.336 meters. The link lengths resulting from this stride were then used to define the overall workspace equality and workspace full-range constraints for the remainder of the test cases.

Again, because this was the first, benchmark test case, there was no specified vehicle velocity. For the sake of simplicity, a stride time of 1.0 second was selected. The vehicle velocity resulting from this choice was later matched by the other test cases using the method described in Step #2 of the previous section. The motion parameters unique to this test case are summarized in Table 7.4.

Table 7.4 — Test Case #1 Motion Program Parameters

Step interval, σ	1.0
Stride length, λ	0.336 m
Stride time, τ	1.0 s

For the case of $\sigma = 1.0, \omega = 6.0$ without any leg pair pitching motion, the parametric cubic coefficient matrices for the rearmost leg pair are shown below. The motion program for leg pair 6 specified that it would follow the path described by Eq. 7.2 and Eq. 7.3, starting at $t = 0.1\text{s}$ and finishing at $t = 1.1\text{s}$.

$$\mathbf{B}_{XYZ} = \begin{bmatrix} -3.175 & 0.000 & 0.000 \\ -2.839 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.400 \\ 0.000 & 0.000 & -0.400 \end{bmatrix} \tag{7.2}$$

$$\mathbf{B}_{\alpha\beta\gamma} = \begin{bmatrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.3}$$

Note that the foothold positions of leg pair 6 are both negative in Eq. 7.2 because the coordinate frame origin is located at the robot's centroid, as was shown in Fig. 7.3. The leg step trajectories defined for leg pairs 1 through 5 have similar coefficient matrices, with the only differences being different starting and ending foothold positions, and that the "lift-off" and "touch-down" times are offset an amount of time corresponding to the step interval of 1.0. The coefficient matrix for the position curve (Eq. 7.2) is sparsely populated because this test case is for straight-line locomotion along the **X**-axis, over flat terrain. Likewise, since the leg pair was not rolling, pitching, or yawing (as it would if the vehicle were climbing or turning), the geometric coefficient matrix for the orientation curve was all zeros (Eq. 7.3). Although not needed here, these unused array elements will be essential for specifying turns and the more complicated leg step trajectories required for crossing rugged 3-D terrain.

The three graphs shown in Fig. 7.4 display the results of the inverse kinematics and its derivatives for the rearmost actuation unit as it executed this motion program.

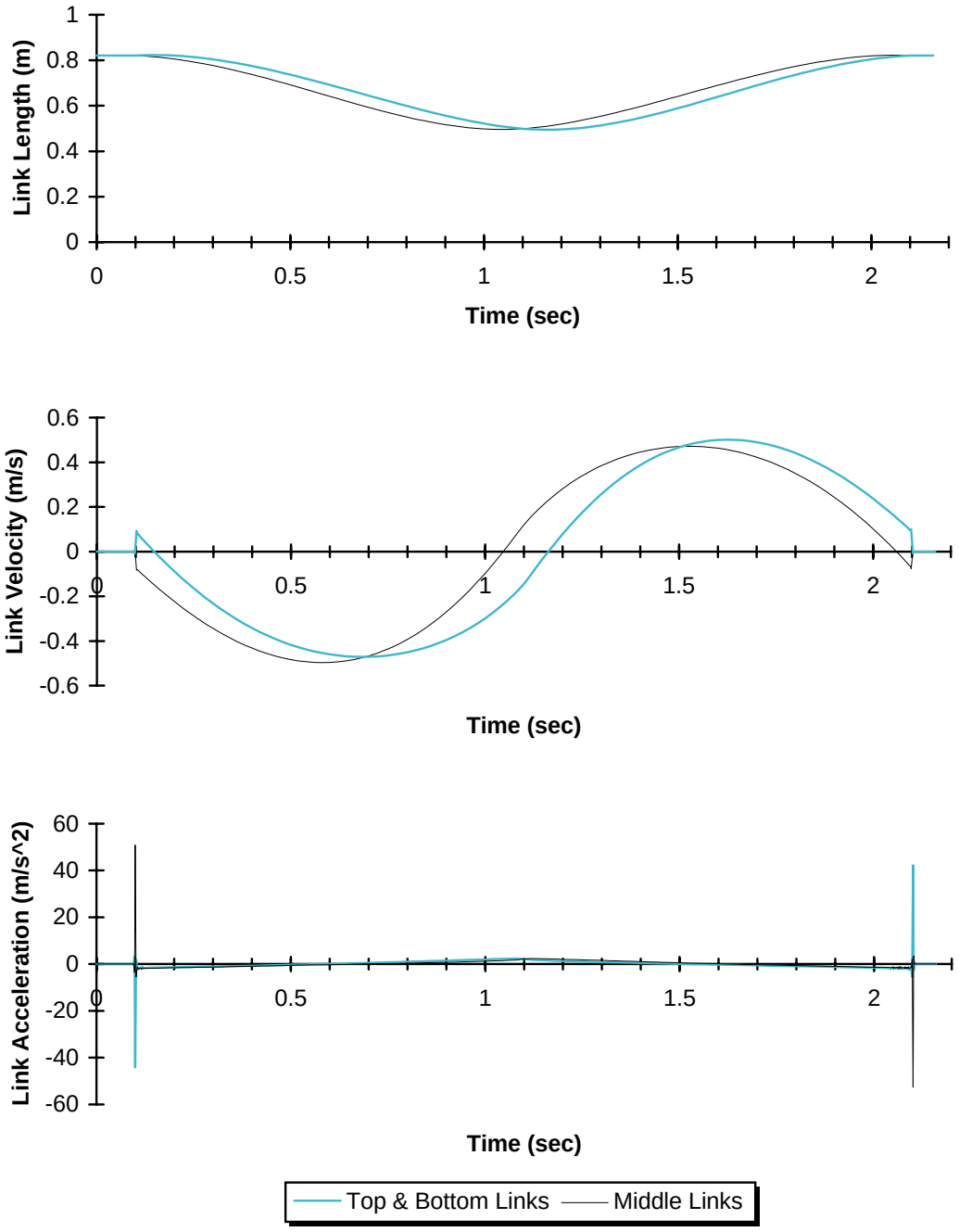


Figure 7.4 — Link Length, Velocity, and Acceleration Graphs for an Actuation Unit Performing the $\sigma = 1.0, \omega = 6.0$ Gait, with $\lambda = 0.336\,\mathrm{m}$, No Pitching, and Constant Parametric Variable Spacing.

Specifically, the top graph shows the variation of length of all six of the prismatic links making up the rearmost Stewart-Gough platform mechanism. However, because the locomotion is straight ahead and does not involve turning, all the corresponding left and right actuators have identical motions.

Also, the "Mele" configuration of the Stewart-Gough Platform, with its equal length links, has an additional symmetry of its top and bottom links. Thus, rather than seeing six independent curves in each of the three graphs, only two independent curves are needed to display the motion of all six links of the actuation unit.

In the top graph we see a smooth variation of the prismatic joints between their maximum and their minimum lengths. First, the actuation unit contracts as leg pair 6 strides forward. Then it re-extends as leg pair 5 performs its own leg step.

In the link velocity graph everything appears satisfactory except for an abrupt change in velocity at the beginning and end of the actuation unit motion, $t = 0.1\text{s}$ and $t = 2.1\text{s}$, respectively.

In the acceleration graph at the bottom of Fig. 7.4, we see that these rapid velocity changes differentiate to become large spikes that dominate the graph. On a real robot, it is likely that the actuators would saturate and be unable to produce these accelerations. The next section describes a modification to the simulation code that solves this problem.

7.6 Cycloidal Transformation of the Parametric Variable

In the top graph in Fig. 7.4, the link length variation versus time curve looks smooth to the naked eye. But after two differentiations, its acceleration graph had large spikes at the beginning and end of leg steps. At first it appeared that these spikes

were the result of some numerical or round-off error. However, after carefully re-running this test case at a variety of time resolutions (as explained in Section 4.6) it was found that the spikes were not caused by round-off or truncation errors. They were simply the result of impulse moves to or from a stationary position. These unintended impulse moves were caused by the simple method used by the simulation code to interpret the motion programs. Thus, a modification of the simulation code was necessary.

A solution was reached by first noting the similarity of this problem with the design of high-speed cams. Scaling the rate of motion along the cubic curve paths would be more realistic and would, in turn, yield more meaningful velocity and acceleration graphs. In the simulation, the parametric variable is assumed to be time or proportional to time. Therefore, this motion scaling could be accomplished by transforming the parametric variable, u, of the leg step trajectory so that its value was no longer linearly varying with time, but had an appearance like that shown in Fig. 7.5.

The equation for a cycloidal motion profile was selected because it provides zero acceleration at both the beginning and end of the motion. Adapting the equations from Kloomok and Muffley (1955) and simplifying, we get:

$$u^* = \left(u - \frac{\sin 2\pi u}{2\pi} \right) \tag{7.4}$$

Applying the cycloidal function in Eq. 7.4 as a transformation on the parametric variable for each cubic curve results in the desired variation in point spacing.

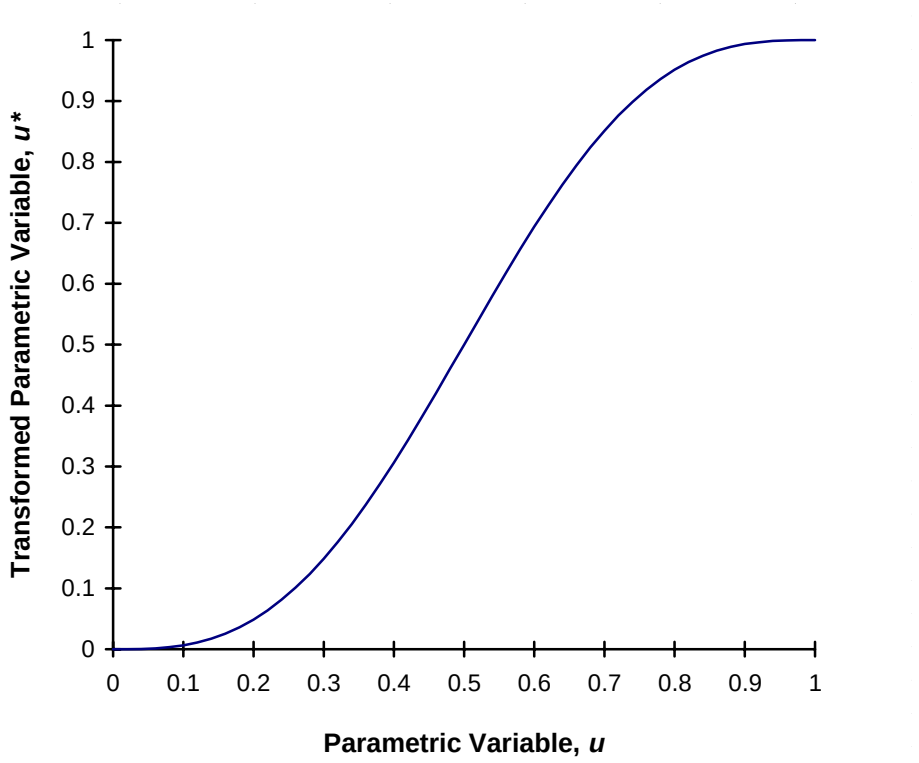


Figure 7.5 — The Cycloidal Transformation of the Parametric Variable.

Figures 7.6 and 7.7 demonstrate the situation before and after using the cycloidal transformation. Both figures show a side view of the same leg step path and indicate points along the trajectory for the same values of the parametric variable.

In the first graph, which uses the untransformed, linear parametric variable spacing, the data points are approximately evenly spaced in terms of arc length along the cubic curve. This results in large accelerations at the beginning and end of the stride. In the second graph, which uses the cycloidal parametric variable spacing, notice that the data points are closer together at the beginning and end of the trajectory, and further apart in the middle, just as would be expected with an accelerating and decelerating body.

In the next section, we see that this scaling of the leg pairs' rate of motion yields more practical answers for the accelerations of the links.

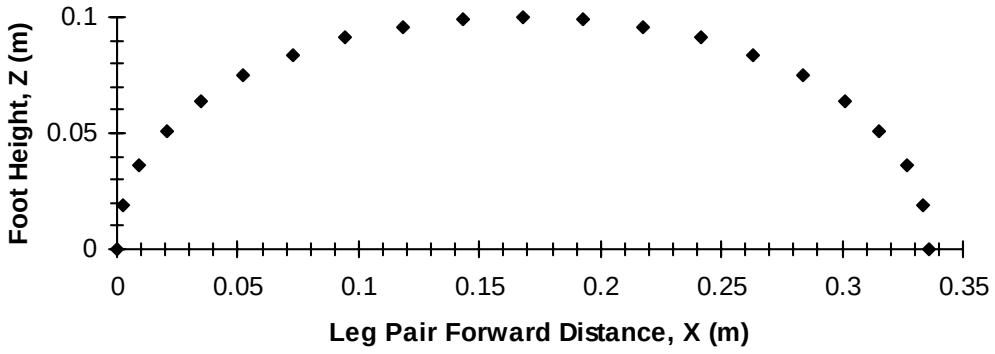


Figure 7.6 — Cubic Curve X-Z Foot Trajectory Using *Linear* Spacing of the Parametric Variable, u.

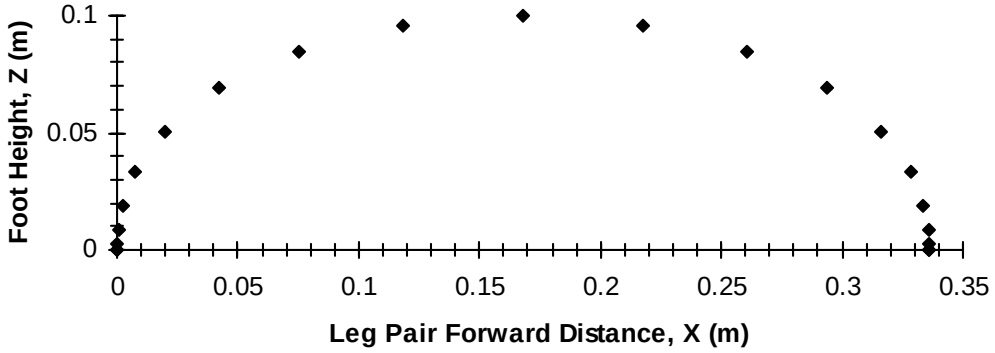


Figure 7.7 — Cubic Curve X-Z Foot Trajectory Using *Cycloidal* Spacing of the Parametric Variable, u^*.

7.7 Test Case #2: Step Interval (σ) = 1.0, with Cycloidal Spacing

The robot design and motion program for this test case are identical to Test Case #1 with the exception that the cycloidal function was used to scale the parametric variable of the *xyz* and $\gamma\beta\alpha$ cubic curves that define the motion of each leg pair.

Figure 7.8 shows the kinematic data for the last actuation unit that results from running this motion program. Looking at the link length graph, we see once again that between $t = 0.1$ and $t = 1.1$ seconds the prismatic links of the Stewart-Gough Platform mechanism are compressing as leg pair 6 strides forward. The links then re-extend again from $t = 1.1$ to $t = 2.1$ seconds as leg pair 5 steps forward.

Comparing the link length graph of Fig. 7.8 with that of Fig. 7.4, we see that, while the links go through the same range of motion (from 0.82292 to 0.49439 meters and back), the curve shapes are different. Looking at the velocity and acceleration graphs of Fig. 7.8 we see that they are smooth and have none of the discontinuities that were present in Fig. 7.4. Also, note that the velocity and acceleration graphs of Fig. 7.8 do *not* have the same magnitude scale as in Fig. 7.4.

Figure 7.4 shows that for a robot to accurately run the motion program with linear parametric variable spacing would require a peak link acceleration of 52.64583 m/s^2. But as Fig. 7.8 reveals, following the same leg step path using cycloidal parametric variable spacing would require a peak link acceleration of only 5.14500 m/s^2.

This factor of ten reduction is impressive considering that the robot was moving at the same average velocity along the same exact path as in the previous example. In view of this improvement, cycloidal parametric variable spacing will also be used with the remaining examples of this chapter.

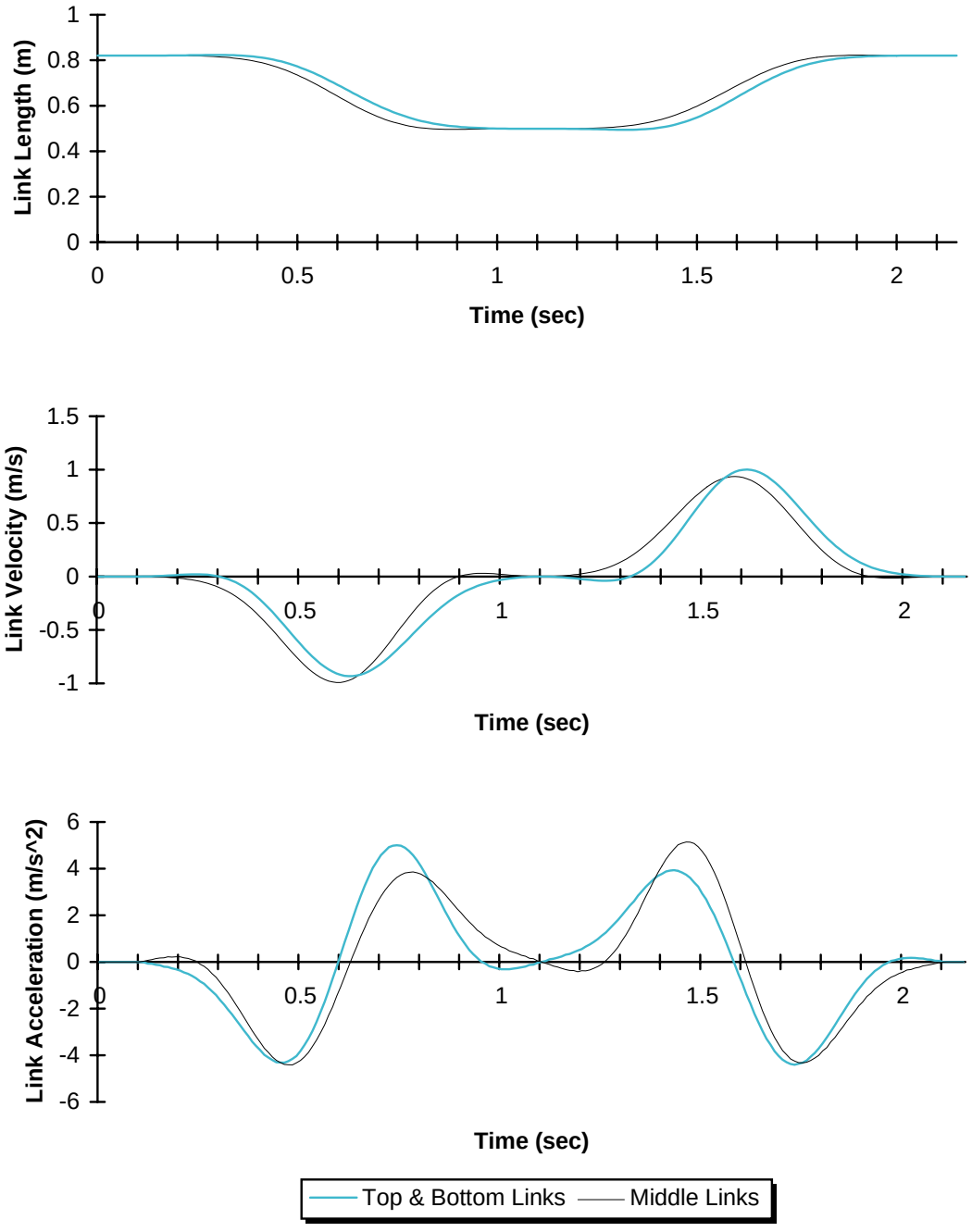


Figure 7.8 — Link Length, Velocity, and Acceleration Graphs for an Actuation Unit Performing the $\sigma = 1.00$, $\omega = 6.0$ Gait, with $\lambda = 0.336\,\mathrm{m}$, No Pitching, and Cycloidal Parametric Variable Spacing.

Table 7.5 — A Sequence of Side Views and Stability Plots of a Robot Performing the $\sigma = 1.00, \omega = 6.0$ Gait.

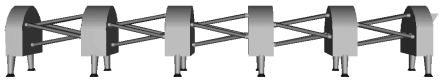
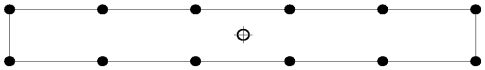
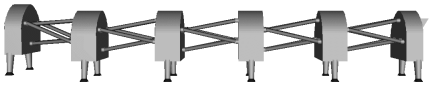
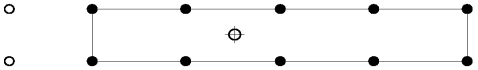
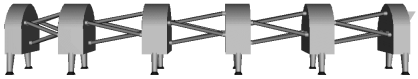
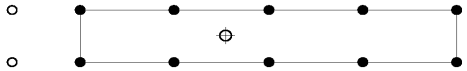
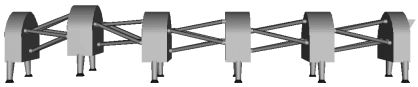
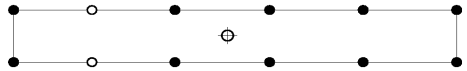
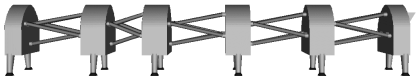
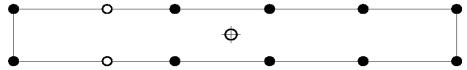
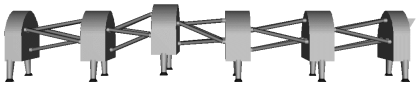
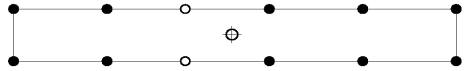
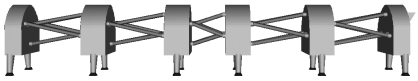
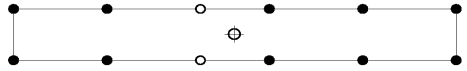
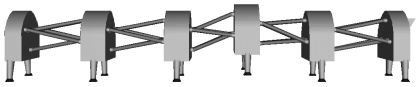
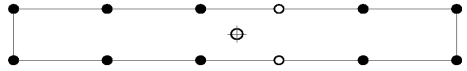
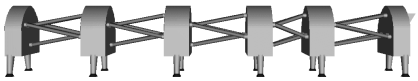
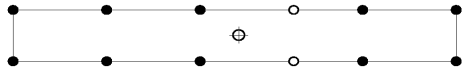
t = 0.071 seconds		
t = 0.571		
t = 1.071		
t = 1.571		
t = 2.071		
t = 2.571		
t = 3.071		
t = 3.571		
t = 4.071		

Table 7.5 presents a sequence of rendered images and stability plots of the vehicle created with the simulation code. Since these images were made from the same robot design and motion program as the graphs presented in Fig. 7.8, the time scales are the same in both Table 7.5 and Fig. 7.8. While Table 7.5 covers a longer duration than Fig. 7.8, so as to show more of the gait, the first five rows of Table 7.5 are "snapshots" corresponding to the data graphed in Fig. 7.8.

Due to the small size of the rendered images and the moderate step height, it is helpful to place a straightedge along the robot's feet to better see which leg pair is off the ground and moving forward. The rear-to-front, wave-like motion of the gait can be seen by looking at this sequence. It is much more obvious when viewing the animation produced by the simulation tools on a computer monitor.

7.8 Test Case #3: Step Interval (σ) = 0.52

This test case differs from the previous one by using a different gait, namely the step interval = 0.52, wave interval = 6 gait. While the gait used in the first two tests was quite simple, the gait used here is similar to the one the caterpillars were observed to use (see Ch. 3). The gait diagram for this test case is similar to that of Fig. 5.4. Figure 5.4 depicted the $\sigma = 0.50$ gait, while here the $\sigma = 0.52$ gait was used as a safety factor to ensure stability (as discussed in Section 7.2).

The stride length was determined and the stride time was scaled by explicitly following the procedure described in Section 7.4.4, so that this test case satisfies all the constraints, including the overall constraints specified by Test Case #1. The resulting parameters are shown in Table 7.6.

Table 7.6 — Test Case #3 Motion Program Parameters

Step interval, σ	0.52
Stride length, λ	0.349 m
Stride time, τ	1.997482 s

Notice that the use of the $\sigma = 0.52$ gait allows a slightly longer stride length and a stride time almost double that of the $\sigma = 1.00$ gait (see Table 7.4).

The motion program for the rearmost leg pair was defined as follows. At $t = 0.1$ seconds, leg pair 6 began lifting off the ground. It followed the cubic path defined by Eq. 7.5 until it completed its leg step at $t = 2.097482\,\text{s}$.

$$\mathbf{B}_{XYZ} = \begin{bmatrix} -3.175 & 0.000 & 0.000 \\ -2.826 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.400 \\ 0.000 & 0.000 & -0.400 \end{bmatrix} \tag{7.5}$$

As in Test Cases #1 and #2, since the robot is not turning, rolling, or climbing, the orientation geometric coefficient matrix, $\mathbf{B}_{\alpha\beta\gamma}$, is made up of all zeros.

Unlike the gait used in the first two test cases, in this case, leg pair 5 began to move while leg pair 6 was still only about halfway through its own leg step. Specifically, leg pair 5 lifted off the ground at $t = 1.13869\,\text{s}$, while leg pair 6 did not finish its leg step until $t = 2.097482\,\text{s}$. The simultaneous movement of leg pairs can be seen in the last four rows of Table 7.7.

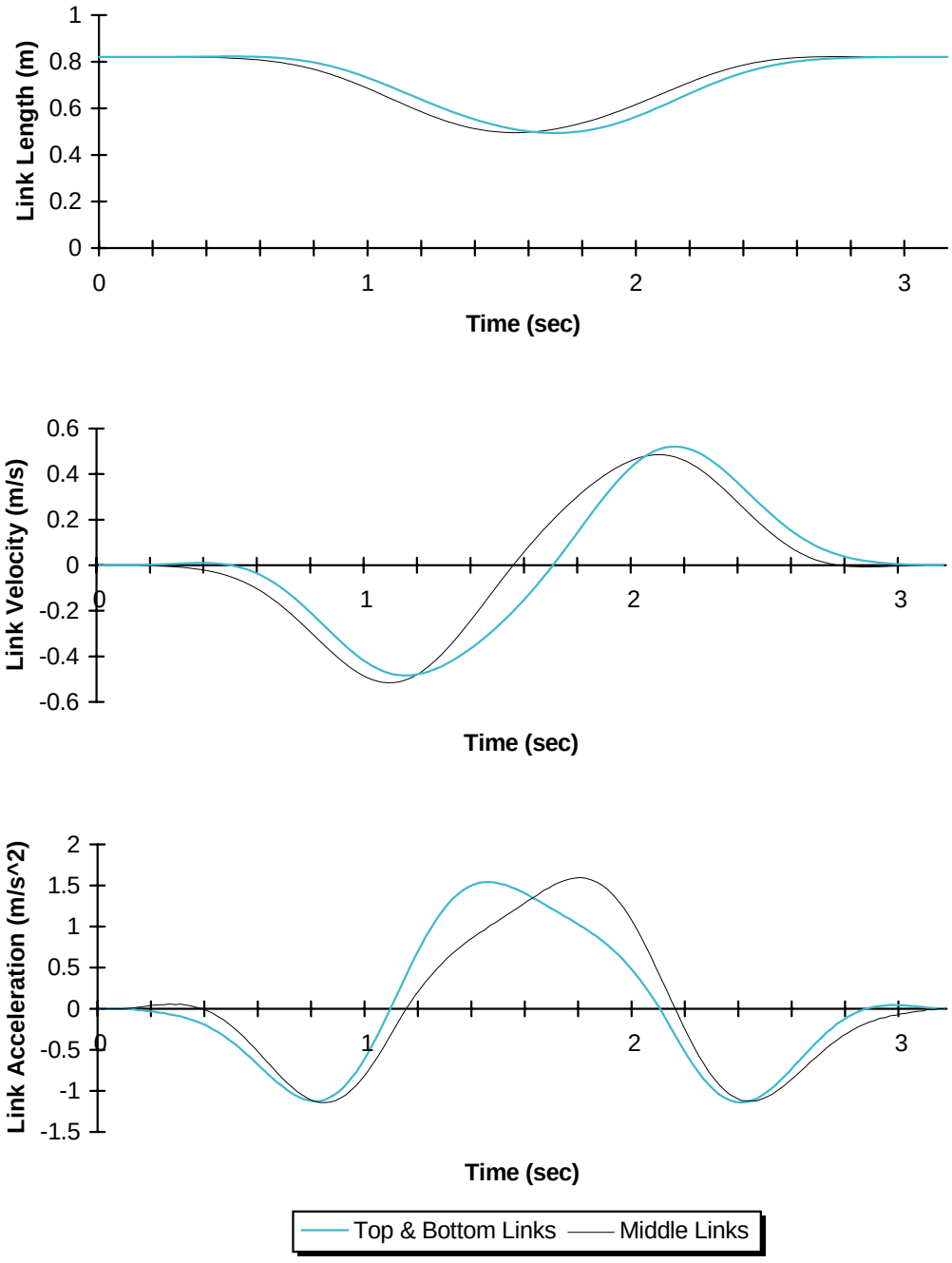
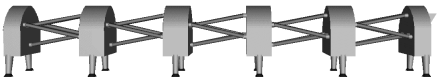
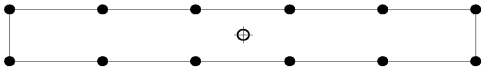
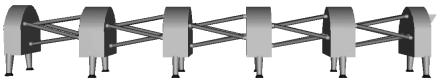
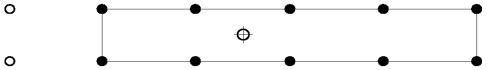
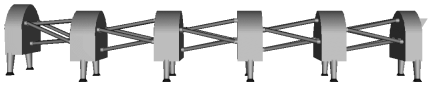
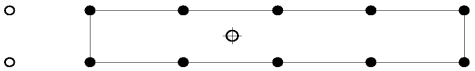
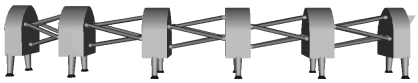
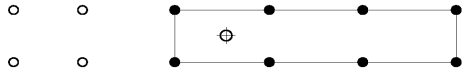
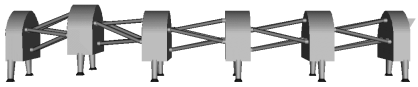
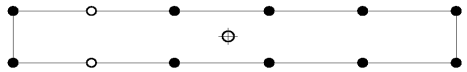
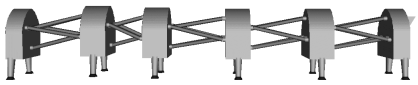
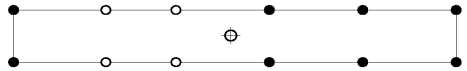
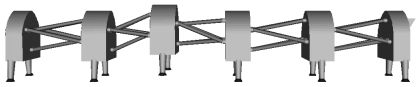
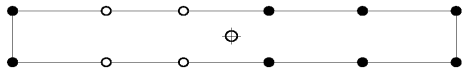
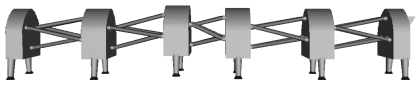
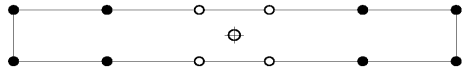
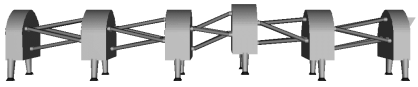
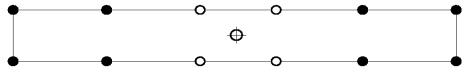


Figure 7.9 — Link Length, Velocity, and Acceleration Graphs for an Actuation Unit Performing the $\sigma = 0.52$, $\omega = 6.0$ Gait, with $\lambda = 0.349\,\mathrm{m}$, No Pitching, and Cycloidal Parametric Variable Spacing.

Table 7.7 — A Sequence of Side Views and Stability Plots of a Robot Performing the $\sigma = 0.52$, $\omega = 6.0$ Gait, without Pitching.

t = 0.071 seconds		
t = 0.571		
t = 1.071		
t = 1.571		
t = 2.071		
t = 2.571		
t = 3.071		
t = 3.571		
t = 4.071		

Looking at the graphs in Fig. 7.9, we see that this simultaneous movement of leg pairs causes no discontinuities. While the links go through the same range of motion as in the earlier test cases, the peak actuator velocities with the $\sigma = 0.52$ gait are significantly lower for the same average speed of the crawler. More importantly, the peak acceleration of 1.59625 m/s^2 for the $\sigma = 0.52$ gait is 69% lower than the peak acceleration for the $\sigma = 1.00$ gait.

7.9 Test Case #4: Step Interval (σ) = 0.52, with Pitching

Like the example presented in the previous section, the gait used in this test case uses a step interval of 0.52 and a wave interval of 6. But it differs from the earlier examples by adding a pitching motion to the trajectory of each leg pair. The pitching makes this motion program resemble the caterpillars' method of locomotion even more closely.

As in the previous example, the gait diagram for this test case is similar to that shown in Fig 5.4. The additional pitching motions of the leg steps do not alter the gait because, as each leg pair tilts, its feet continue to support the weight of the robot.

Specifically, each leg pair starts pitching forward and rotating about its ankles *before* it lifts off the ground. This pitching motion begins before the previous leg pair is even halfway through its own leg step trajectory. Then the leg pair lifts off the ground and tilts backward as it moves forward and returns to the ground. Once back on the ground, it pitches forward again, rotating about its ankles until it has returned to an upright position.

Recall from the motion programming chapter (Ch. 5) that leg steps with pitching are described using three pairs of *xyz* and $\gamma\beta\alpha$ parametric cubic curves. One pair defines the initial pitch forward. The next pair defines the simultaneous stride forward and

pitch backwards. And the last pair defines the final pitching motion, which brings the leg pair to a vertical, upright position again.

Although more cubic curves were used to describe each leg step of this motion program compared to the earlier examples, this test case conformed to all the constraints described in Section 7.4, resulting in the parameters listed in Table 7.8.

Table 7.8 — Test Case #4 Motion Program Parameters

Step interval, σ	0.52
Stride length, λ	0.653 m
Stride time, τ	3.737408 s

A maximum pitch angle of 15.5 degrees and a pitch duration of 1.868704 seconds were also used to create this motion program. The pitch duration was selected to be half of the stride duration, τ, in order to maximize the time available for pitching and thus reduce accelerations. For simplicity, the same maximum pitch angle and pitch duration were used at both the beginning and end of each step.

Since the stride time, τ, listed in Table 7.8 does not include the pitch durations on either end of the stride, the total time that the leg pair is in motion is actually 2τ, or 7.47 s. Also, notice that the stride length in Table 7.8 is considerably longer than that of the $\sigma = 0.52$ gait without pitching.

The motion program for the 6th (rearmost) leg pair was as follows. Between $t = 0.1$ seconds and $t = 1.968704$ seconds, leg pair 6 follows the trajectory described in Eq. 7.6 and 7.7. This causes the leg pair to lean forward, rotating its center of mass 15.5 degrees around its ankles in a circular arc, while its feet remain on the ground supporting it.

$$\mathbf{B}_{XYZ} = \begin{bmatrix} -3.175 & 0.000 & 0.000 \\ -3.175 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.6}$$

$$\mathbf{B}_{\alpha\beta\gamma} = \begin{bmatrix} 0.000 & 0.000 & 0.000 \\ 0.000 & 15.500 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.7}$$

Between $t = 1.968704\,\text{s}$ and $t = 5.706113\,\text{s}$ the leg pair lifts off the ground and steps forward 0.653 m from its starting foothold (Eq. 7.8). Meanwhile, it rotates backward 31.0 degrees so that it is at a -15.5 degree angle when its feet touch down on the ground again (Eq. 7.9).

$$\mathbf{B}_{XYZ} = \begin{bmatrix} -3.175 & 0.000 & 0.000 \\ -2.522 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.400 \\ 0.000 & 0.000 & -0.400 \end{bmatrix} \tag{7.8}$$

$$\mathbf{B}_{\alpha\beta\gamma} = \begin{bmatrix} 0.000 & 15.500 & 0.000 \\ 0.000 & -15.500 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.9}$$

Finally, between $t = 5.706113\,\text{s}$ and $t = 7.574817\,\text{s}$, the leg pair's feet remain in stationary contact with the ground (Eq. 7.10). At the same time, the leg pair pitches forward, rotating 15.5 degrees around its ankles, until it has returned to a vertical orientation (see Eq. 7.11), thus completing its leg step.

$$\mathbf{B}_{XYZ} = \begin{bmatrix} -2.522 & 0.000 & 0.000 \\ -2.522 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.10}$$

$$\mathbf{B}_{\alpha\beta\gamma} = \begin{bmatrix} 0.000 & -15.500 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{7.11}$$

As with the previous examples, motions of the last actuation unit are also dependent on the programmed motion of leg pair 5. Note that leg pair 5 began pitching forward at $t = 2.043452\,\text{s}$, just after leg pair 6 lifted off. And leg pair 5 lifted off the ground at $t = 3.912157\,\text{s}$, when leg pair 6 was near the peak of its stride. Leg pair 4 lifted off at $t = 5.855609\,\text{s}$, which was just after leg pair 6 touched down, but a relatively long time before leg pair 6 had finished returning to a vertical orientation.

Figure 7.10 shows graphs of the length, velocity, and acceleration of the six links making up the rearmost actuation unit as it performed the programmed motions. As in the graphs for the earlier test cases, the straight-line, forward locomotion again results in lateral symmetry so that the left side and right side links have identical motions. However, unlike the earlier cases, the pitching motion of this test case prevents symmetric motion of the top and bottom links. Thus, we have three curves per graph in Fig. 7.10 rather than only two.

Note that the cycloidal parametric variable spacing was applied individually to all three of the trajectory curve segments for each leg pair.

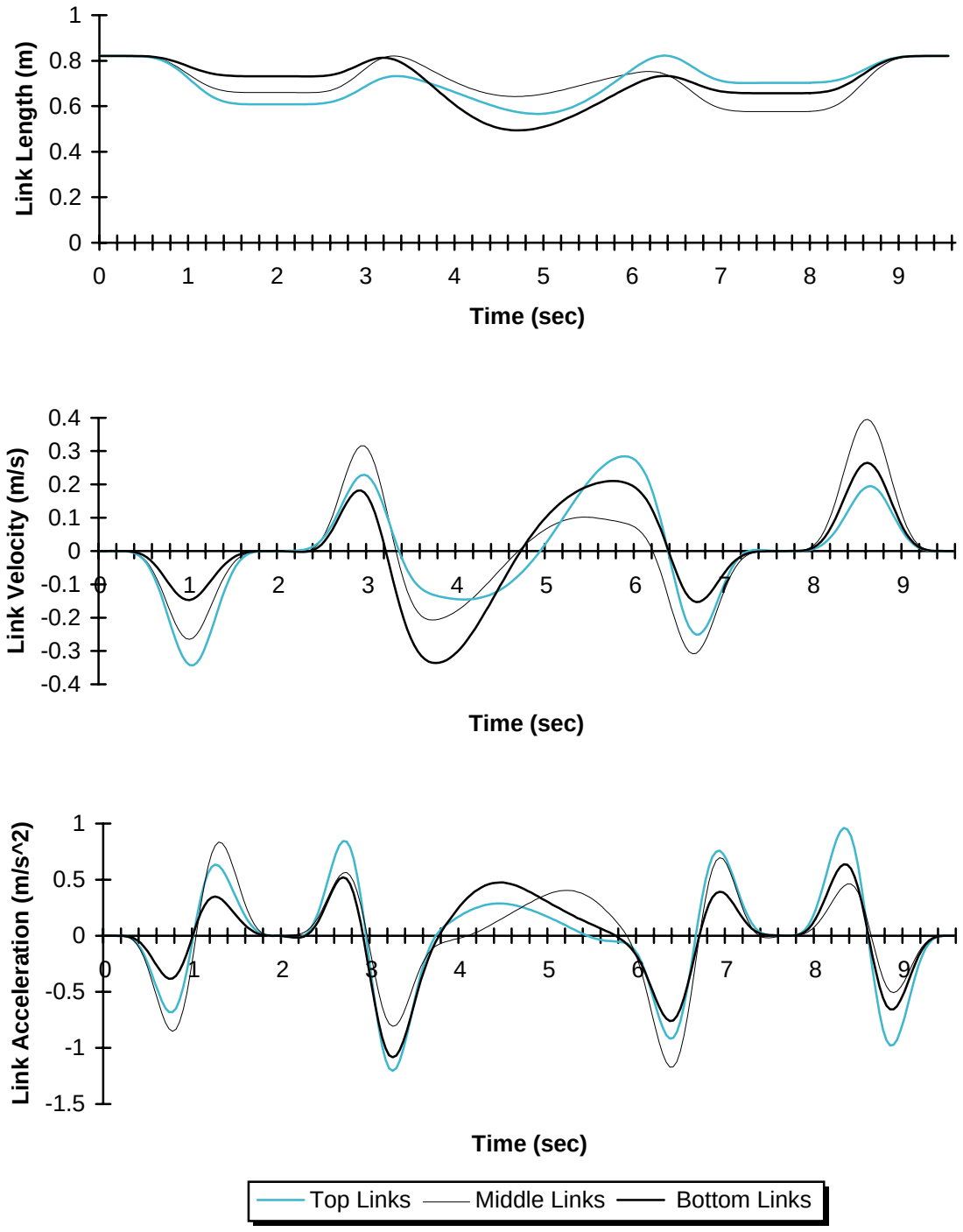
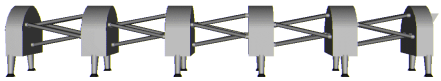
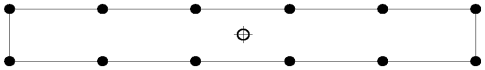
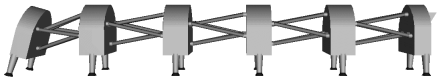
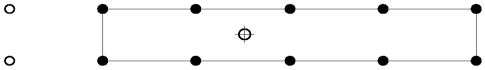
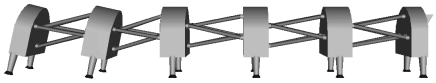
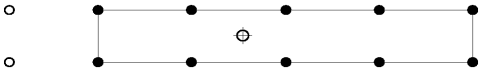
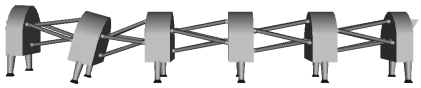
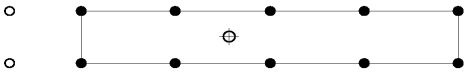
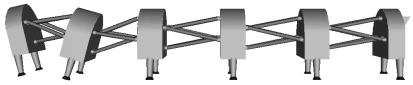
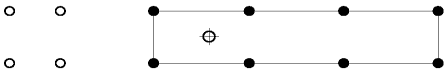
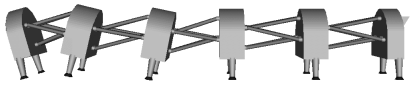
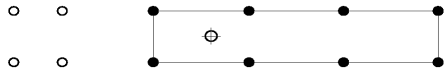
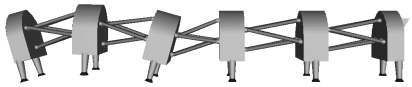
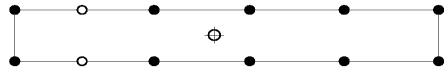
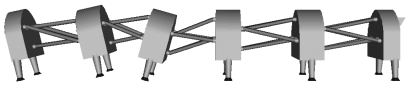
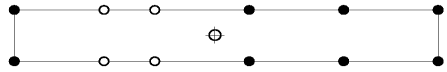
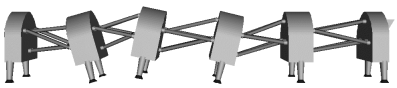
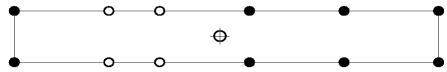


Figure 7.10 — Link Length, Velocity, and Acceleration Graphs for an Actuation Unit Performing the $\sigma = 0.52$, $\omega = 6.0$ Gait, with $\lambda = 0.653\,\mathrm{m}$, 15.5° Pitching, and Cycloidal Parametric Variable Spacing.

Table 7.9 — A Sequence of Side Views and Stability Plots of a Robot Performing the $\sigma = 0.52, \omega = 6.0$ Gait with Pitching.

t = 0.000 seconds		
t = 2.286		
t = 3.143		
t = 3.786		
t = 4.500		
t = 5.071		
t = 5.714		
t = 6.429		
t = 7.143		

Thus we see that, when leg pair 6 tilts forward between $t = 0.1\text{s}$ and 1.968704 s, the acceleration graph shows the links both accelerating and decelerating back to zero.

The graphs are considerably more complex here than in the earlier test cases. Yet each graph is continuous, and the peak acceleration required by this motion is only 1.20526 m/s^2, a 24% reduction from the $\sigma = 0.52$ test case without pitching.

Table 7.9 shows the same motion as a sequence of side view renderings and stability plots. The pitching motion of the leg pairs is clearly evident in these renderings.

Once again, since the simulation code made the graphs of Fig. 7.10 and images of Table 7.9 from the same robot design and motion program, the time scales are identical. The nine "snapshots", pairs of images and stability plots, in Table 7.9 cover almost the entire leg step motion of the 6th leg pair, from $t = 0$ until $t = 7.143$ seconds. The graphs of Fig. 7.10 show that the rearmost actuation unit continues to move longer, until $t = 9.518269\,\text{s}$, in order to impart motion to the 5th leg pair.

It can be seen from the simulation output that programming the leg pairs to simultaneously pitch forward while supporting weight prolongs the available time for each leg pair to move its center of gravity forward. It also elongates the stride of the leg pairs because the pitching leaves more room for the rear leg pairs to move forward without reaching their minimum link lengths.

7.10 Chapter Summary

The results presented in this chapter demonstrate the utility of the simulation tools for the analysis and design of multibody passive-legged crawling vehicles. The simulation tools aid in visualizing crawling vehicle designs and their motions,

analyzing stability, synthesizing link lengths, and selecting motion programs. The results of three different types of tests were described in this chapter.

First, several diverse examples of robot designs were presented to illustrate the parametric geometry modeling and rendering capabilities of the simulation.

Next, an analysis of two gaits with regard to stability versus number of leg pairs yielded results that a step interval = 1.00 gait requires a robot with at least 4 leg pairs and that a step interval = 0.52 gait requires at least 6 leg pairs to remain statically stable on flat, level terrain with consistent load-bearing capacity. For terrains of unknown load-bearing capacity, the σ = 1.00 and σ = 0.52 gaits require at least 5 and 7 leg pairs respectively. Of course, the longer robots can use either gait and thus trade-off speed/efficiency for stability as appropriate for the terrain.

Finally, the chapter presented the procedure for using the simulation tools to compare motion programs, with an extensive example containing three notable test cases. The most basic use of the simulation code is the task of programming a desired locomotion for a robot and then synthesizing the required link lengths. While this most basic function of the simulation was only briefly mentioned here, its effectiveness has been demonstrated in each of the three test cases.

The motion programming test cases also showed that the cycloidal transformation of the parametric variable was useful for scaling the rate of movement of the robot in a realistic manner. While other rate scaling methods are also possible, the cycloidal parametric variable spacing was definitely superior to the linear parametric variable spacing.

The results of the three motion programming test cases are summarized in the charts of Fig. 7.11, Fig. 7.12, and Fig. 7.13. We can clearly see from the charts that imitating the caterpillars' gait and pitching leg step trajectory allows longer strides, lower peak link velocities, and lower peak accelerations for a given vehicle speed.

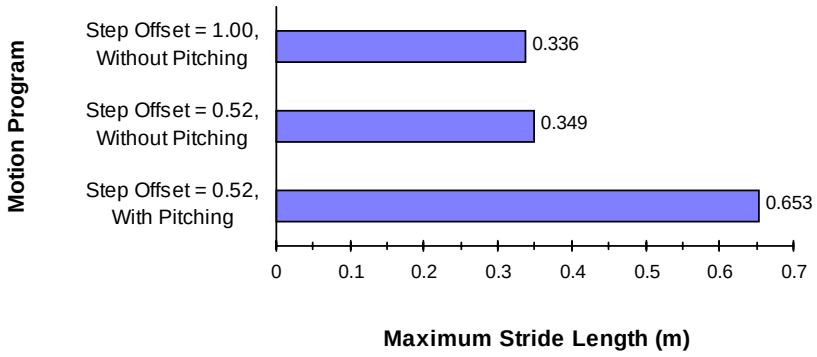


Figure 7.11 — Comparison of Motion Program Leg Step Maximum Stride Lengths

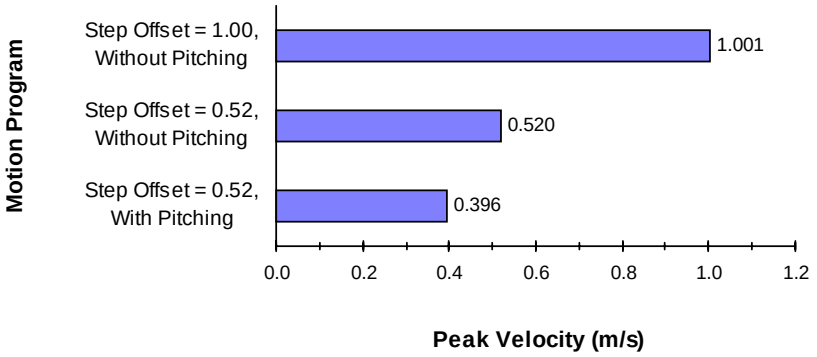


Figure 7.12 — Comparison of Motion Program Peak Velocities

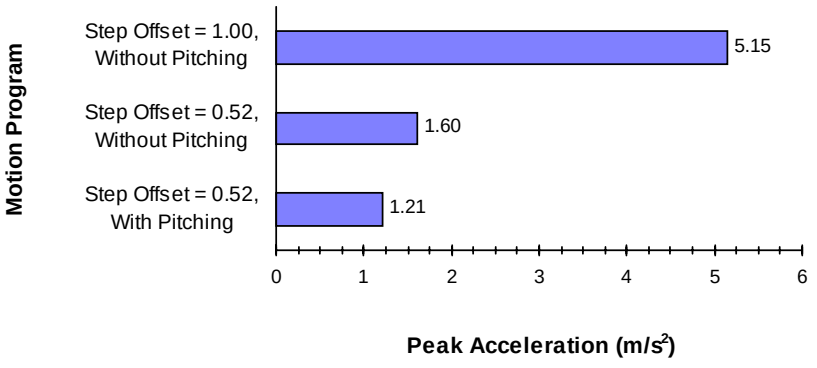


Figure 7.13 — Comparison of Motion Program Peak Accelerations

In some cases, depending on the power transmission used, a robot may be limited by peak joint velocity rather than torque/acceleration. For example, if an actuation unit uses lead screws for the prismatic links, peak speed may be a limitation. Comparing the peak velocities of the links during the three motion programs (Fig. 7.12), we see that the motion programs with the lowest peak accelerations also had the lowest peak velocity. This is surprising until we recall that the caterpillar-like gait and leg step trajectories used in the later test cases both tended to prolong the *duration* of each stride. As shown in Fig. 7.12 and Fig 7.13 above, the motion program using the step interval (σ) 0.52 gait and pitching trajectories produces a 60.4% lower peak link velocity and a 76.5% lower peak link acceleration than the $\sigma = 1.00$ motion program.

After considering the test results and viewing the animations produced by the simulation, the superior maximum link velocities and accelerations of the caterpillar-like gaits appear to be brought about by three factors.

- Longer strides were enabled by moving the leg pair at the front of the "wave" out of the way of the following leg pair.
- The duration of each leg pair's stride was lengthened.
- More leg pairs and actuation units were moved simultaneously, see Table 7.10 below.

Thus, comparing the caterpillar-like gaits with the σ = 1.00 gait, since more leg pairs were moved simultaneously to achieve the same overall speed, they did not need to be, individually, moved as quickly.

Table 7.10 — Comparison of Simultaneous Motions of the Motion Test Cases

Motion Program	Number of Simultaneously Moving Leg Pairs	Number of Simultaneously Moving Actuation Units
$\sigma = 1.0$	1	2
$\sigma = 0.52$	2	3
$\sigma = 0.52$ with pitching	4	5

The kinematics model and simulation developed in this work can provide the information needed for configuration design by allowing various robot configurations and motion programs to be compared to determine their *relative* performance. That is, the simulation tools can be used to determine which robot design/motion program combination is best for a particular criterion. For example, when testing two gaits with a robot design, if Gait A has lower actuator accelerations than Gait B for the same robot speed, then the robot will definitely have a *higher* speed using Gait A when the two gaits have the same actuator accelerations. The benefits of a more efficient gait can be thought of as either a faster robot with the same size actuators, or an equally fast robot using less powerful, lighter weight actuators.

Thus, we can say from the motion programming tests in this chapter, that for this particular robot design, the $\sigma = 0.52$ with pitching motion program is the preferred method of locomotion in situations where stability is not a concern. When extra stability is needed, the $\sigma = 1.00$ gait is the best of the three examples.

While the motion programing tests presented here have only scratched the surface of possible uses of the simulation, they do demonstrate the usefulness of the tools and the underlying model for making configuration design decisions. Beyond design selection, by aiding in the visualization of the motion, the tools also helped answer *why* the caterpillar-like motion programs were effective.

7.11 Conclusion

The simulation has shown that the conceptual design described in Ch. 2 and the motion programming technique presented in Ch. 5 yield reasonable results. The renderings of parametric geometry models, stability analysis, and motion program comparisons demonstrate that the mathematical model and simulation created as part of this research are indeed useful tools for designing both crawling vehicle physical hardware and motion programs.