Design of Fast Walking with One- Versus Two-at-a-Time Swing Leg Motions for RoboSimian

Marten Byl and Katie Byl
UCSB Robotics Lab, Dept. of Electrical and Computer Engineering
University of California at Santa Barbara
Santa Barbara, CA 93106
Email: {marten.byl, katiebyl}@gmail.com

Abstract—This paper presents two prototype fast walking gaits for the quadruped robot RoboSimian, along with the experimental results for each. The first gait uses a statically stable one-at-a-time swing leg crawl. The second gait uses a two-at-a-time swing leg motion, which requires deliberate planning of zero-moment point (ZMP) to balance the robot on a narrow support base. Of particular focus are the development of practical means to exploit the fact that RoboSimian has high-dimensionality, with seven actuators per limb, as a means of partially overcoming low joint velocity limits at each joint. For both gaits, we use an inverse kinematics (IK) table that has been designed to maximize the reachable workspace of each limb while minimizing joint velocities during end effector motions. Even with the simplification provided by use of IK solutions, there are still a wide range of variables left open in the design of each gait. We discuss these and present practical methodologies for parameterizing and subsequently deriving approximate time-optimal solutions for each gait type, subject to joint velocity limits of the robot and to real-world requirements for safety margins in maintaining adequately balanced over time. Results show careful choice of parameters for each of the gaits improves their respective walking speeds significantly. Finally, we compare achievable walking speed of each gait and find they are nearly equivalent, given current performance limitations of the robot.

I. INTRODUCTION

RoboSimian, shown in Figure 1(b), is a highly dexterous, four-limbed robot, designed and built by Jet Propulsion Labs (JPL) to compete in the DARPA Robotics Challenge (DRC). This work studies the problem of increasing gait speed for RoboSimian on flat terrain. RoboSimian has four identical limbs that are designed for both locomotion and manipulation. During locomotion, the robot chassis is nominally parallel to ground allowing all four 7-DOF limbs to be used for locomotion [1]. Two practical challenges we focus on in this work are (1) that joint actuators are designed to favor high torque, resulting in low-speed (1.4 rad/s) joints, and (2) that there are seven actuated degrees-of-freedom (DOFs) per limb, allowing for much greater freedom in gait design, compared with typical quadrupeds with either 3- or 4-DOF legs. Of particular interest, we wish to explore the trade-offs in using static walking gaits, in which there are always either three or four feet in contact with the ground to provide a large base of support, versus using dynamic trot-walking gaits, in which some portion of the gait cycle uses only two feet to provide a very narrow base of support, requiring compatibly-planned accelerations of the body to remain balanced.

One gait we study is a classic quadruped walking gait where the robot’s center of mass (COM) remains within the polygon defined by either three or four limbs in contact with the ground while at most one leg is swung forward as shown in Figure 1(a). Our second gait attempts to increase walking speed by using a dynamic gait that moves two legs at once, toward reducing the time needed to complete a gait cycle. More specifically, we use a “trot-walk” that involves moving limbs in diagonal pairs. The greatest challenge with this gait is to balance in the narrow “double support” polygon, shown in Figure 1(c), in which COM (red dot) may deviate significantly from ZMP (in yellow). A ZMP preview control method has successfully been adapted to execute such a gait with high repeatability, as demonstrated by Byl et al. [2] in work with the LittleDog quadruped during DARPA’s Learning Locomotion program. Figure 1(d) shows LittleDog mid-stance in this trot-walk, with only two feet in contact with the ground. We describe a new, more direct planning approach for such a gait in the present work.

In the crawl gait, each leg is swung through free space while the body either remains stationary or moves relatively slowly,
compared with the end effector. The three stance legs therefore require lower joint velocities than the swing leg, and forward speed is limited principally by motion of the swing leg relative to the body frame. Thus, reorienting the body may assist the swing limb, much as a pitcher can use whole-body motions to maximize velocity of a thrown ball.

In contrast to the crawl gait, the dynamic trot-walk provides little opportunity to exploit such full-body maneuvers for two key reasons. First, order of the legs in this gait cycle eliminates most of the advantage of twisting the body. For a crawl, rear and front legs on the same side of robot (i.e., left vs right) move sequentially, so as the right-side legs swing forward, yawing the body counterclockwise (from above) assists motions for both legs, as does pitching the body forward. In contrast, diagonal pairs of legs move together in the trot-walk, effectively canceling out any potential utility of body yaw and pitch, by symmetry. Second, required joint velocities of all limbs are more equally matched in the trot-walk, so that motions can not effectively be optimized for one (or perhaps two) of the limbs, as they can for the “pitcher-like” swing leg in the crawl. For example, the swing leg joints require peak velocities roughly an order of magnitude greater than the three remaining stance legs during our crawl, while swing legs “outpace” stance legs by only roughly 2:1 in the trot-walk.

As noted in [3], the combination of possible end effector trajectories and kinematic redundancies makes it a challenge to plan – let alone optimize – RoboSimian’s limb motions when traversing rough terrain. In this work, we simplify the planning problem by focusing specifically on optimizing motions of the limbs and body for fast-walking in a straight line on flat terrain with minimal obstacles, noting that much of the DRC course involves similarly mild terrain.

II. IK Tables

There are 7 actuated joints that determine the 6-DOF pose of the end effector on each of RoboSimian’s limbs, yielding redundant kinematic solutions. Solutions for redundant kinematics have been studied as both a generalized problem [4], [5], [6], [7], [8], and for robot walking [9] and manipulation [10], [11]. For RoboSimian, we have developed a specialized set of inverse kinematics tables (IK) which take advantage of both RoboSimian’s kinematics and dynamics (i.e., exploiting configuration-dependency of Jacobians). Development of these tables is described fully in [12] and briefly summarized here.

In describing our the IK tables, we note that although there are 7 DOFs per limb, only the first 6 DOFs determine the position and orientation of the last L-shaped segment of the limb, with the seventh DOF controlling the “wrist” angle which simply prevents yaw slippage during locomotion. For RoboSimian, there happen to be eight, distinct, 6-to-6 IK solution families for the first 6 DOFs. Each family has a unique collision free workspace and unique “jumps” in solutions. For our work on rough terrain locomotion, we deliberately use a family of solutions with a particularly large workspace. Figure 2(d) depicts this. In general the end effector should be nearly perpendicular to the ground, but we allow the end effector to pitch as it moved radially outward from the body to increase reach of the limbs. (See Fig. 2(a,c).)

Note that while IK tables simplify creation of “good” fast walking solutions, they also reduce the problem to one of finding only approximately optimal solutions.

III. Strategies for Fast Walking

Many high speed dynamic robots such as Boston Dynamic’s Cheetah and WildCat [13] exist today, and they rely on constant movement when walking / running to ensure stability. Depending on the capabilities and design of a robot, there are many different approaches that have been used for maintaining stability while walking quickly. The most popular approaches for planning humanoid locomotion focus on regulation of the center of pressure, referred to as the so-called Zero Moment Point (ZMP) which was introduced by Miomir Vukobratović in 1968 [14]. The ZMP is the point on the ground about which horizontal moments are zero. In traditional ZMP methods, motions for a robot are planned such that the resulting ZMP lies strictly within the convex hull of the robot’s contacts with the ground, called the support polygon, thereby ensuring there are no toppling moments about any edge of the support polygon for the planned motion. This approach is used widely for bipedal robots [15], [16], [17]. ZMP planning can also be used to allow for fast reconfigurations of robot inertia using a very limited base of support [18]. The ZMP trajectory can be designed to jump instantaneously from one point to another. This is possible because the ZMP is a function of both the positions and accelerations of the distributed masses.

Fig. 2: Inverse Kinematics (IK) for RoboSimian. At left, (a) and (c) show how yaw and pitch of the end effector pose allows limbs to unfurl as they move. At right, (b) shows a particular IK table solution, and (d) illustrates its underlying geometry. IK solutions are found by solving the intersection of a sphere centered at the armpit (in blue) with a (partial) sphere centered at the left elbow joint (shown in red). The elbow position is simply set by the 6-DOF pose of the end effector, while the armpit is set by the 6-DOF pose of the robot body.
in the robot, and because we assume we can instantaneously apply torques to set accelerations at the joints. For quadruped robots, ZMP has also been used to perform maneuvers such as lunging and walking [19, 2] which requires two limbs swing at once (see Figure 1(d)). However, ZMP walking is not practical for robots with ”low” joint velocity limits because dynamic balance requires increasingly more precise feedback control as motions become slower: e.g., passive walking devices with stable limit cycles exist, but balancing an underactuated pendulum requires active feedback.

For robots with low joint velocity limits, a more practical method of fast walking may be to optimize trajectories while moving only on limb at a time, as shown in Figure 1(a) and (b). One method is to use third-order spline interpolation to define the swing leg end effector trajectory [20], [21], [22].

For balance during quadruped crawl gaits that move only one leg at a time, the locations \((x, y, z)\) of the feet on the ground is important, while their orientation (roll, pitch, and yaw) is much more flexible, since multiple feet will be used to ground is important, while their orientation (roll, pitch, and yaw) is much more flexible, since multiple feet will be used to support the robot. ZMP walking is less flexible, since multiple feet will be used on the same leg at a time, the locations \((x, y, z)\) of the feet on the ground. Exploiting a large reach in turn increases the potential for yaw and pitch motions of the body to amplify velocity of the end effector, increasing the radial distance of the feet with respect to the robot center of body.

Thus, briefly stated, the goal in designing a crawl gait is to define trajectories for both the body pose and end effector pose throughout the gait to exploit high-DOF limbs to improve both stride length and speed, and intuitively, this is somewhat analogous to an animal exploiting additional degrees of freedom of a flexible spine (for example) during locomotion to achieve longer, faster strides.

IV. DYNAMIC WALKING GAIT

Intuitively, a quadruped should be able to walk faster by using gaits that move two legs forward together, using the other two legs for support, than by swinging the legs forward one-at-a-time in a crawl gait. Such a two-at-a-time gait requires more carefully planned balance, however. In this section, we outline a general approach for achieving such balanced gaits in theory, and we then highlight key practical issues in implementing such gaits on real-world robots. This work is a direct extension of our earlier design of such gaits for LittleDog, presented in [19] and [2].

As shown in Figure 3, the ground reaction force, \(F_g\), points from the ZMP on the ground (e.g., \(p_1\)) to the COM. The geometry shown describes the relative magnitude of the x component of \(F_g\), \(F_x = m\ddot{x}_{com}\), compared with the vertical “length” of \(F_g\), \(F_z = mg\), i.e., the weight under gravity, since \(\ddot{z} = 0\) for a constant-height COM path.

The two-at-a-time trot-walk described in [2] is based directly on a preview control approach for planning ZMP by Kajita et al. [16]. More generally, ZMP planning simply requires generating center of mass (COM) trajectories such that ZMP falls on desired locations within the support polygon throughout the gait. We plan such motions assuming (1) there is no rotation of the body for the dynamic walk (much unlike the static crawl we describe later), and (2) the COM remains at a constant height, so \(\ddot{z}_{com} = 0\), as depicted in Fig. 3.

Referring to Fig. 3, let us consider the case in which we wish the plan for the ZMP to be at point \(p_1\), and define a coordinate system with respect to this desired point, so that \(p_1 = 0\). By Newton’s law, note that \(F_z = mg\) and \(F_x = m\ddot{x}\). From the geometry then,

\[
F_x = \frac{x_{com}}{z_{com}} F_z, \tag{1}
\]

so a constant ZMP location (e.g., remaining at \(p_1\)) requires

\[
\ddot{x} = \frac{g}{z} x. \tag{2}
\]

Equation 2 is simply a linear, second-order equation of motion. It’s familiar solution can be written as:

\[
x(t) = A_1 e^{\lambda_1 t} + A_2 e^{\lambda_2 t}, \tag{3}
\]

where

\[
\lambda_1 = -\lambda_2 = \sqrt{\frac{g}{z}}. \tag{4}
\]

Defining \(t = 0\) when \(x_{com} = 0\) simplifies intuition for the equation. In this case, \(A_1 = -A_2\), and the velocity when \(x_{com} = x_{zmp} = 0\) (i.e., at \(p_1\)) is:

\[
\dot{x}_{com}(t) = 0 = A_1 \lambda_1 + A_2 \lambda_2 = 2A_1 \sqrt{\frac{g}{z}}, \tag{5}
\]

which is a local minimum in its value; i.e., velocity is slowest just as the COM passes over the ZMP.

Figure 4 shows an overhead view of footholds during the dynamic gait. Diagonal pairs of limbs (e.g., left front and right rear) move ahead a distance \(dx_{step}\) and body shifts (dynamically) forward a distance \(\frac{1}{2} \frac{dx_{step}}{dt}\), while the remaining diagonal pair forms a very narrow support polygon. For
RoboSimian, the feet are essentially cylinders, approximately 10 cm in diameter, and we plan with a safety margin of 2.5 cm, leaving only a region 5 cm wide in the $x$ direction in which to place the ZMP. For an aggressive trot-walk, swing legs can theoretically move exactly in sync. A more conservative option is to lift the front leg of the swing-pair first, beginning the rear swing leg motion while the front leg is partway through its travel. Figure 5 depicts this.

Figure 6 shows example COM trajectories for ZMP-based trot-walks for RoboSimian. 6(a) shows the aggressive case, with swing legs in sync. Here, trajectories are piecewise solutions to Equation 3. The transition from solid to dashed line occurs when the ZMP instantaneously jumps to the next diagonal of support, e.g., from $p_1$ to $p_2$ in Fig. 3. Figure 6(b) illustrates a more conservative gait, broken into three phases. (See Fig. 5.) The front foot begins moving near the end of (time) region $T_A$ and finishes exactly at the end of $T_B$. The partnering rear foot travels during the full range of $T_B$ and $T_C$. Trajectories during $T_{A2}$, $T_B$, and $T_C$ all simply use solutions to Equation 3. Note by symmetry, velocities at the border between $T_B$ and $T_C$ match, as desired. Region $T_{A1}$ depicts the special case of starting from rest. Here, we simply use a constant acceleration to match the velocity at the start of $T_{A2}$, resulting in a constant offset between $x_{com}$ and $x_{zmp}$, as shown in Figure 6(c), which also shows the available support polygon (shaded). This subplot also illustrates the timing of swing leg motions more clearly than 6(b).

![Fig. 4: ZMP gait cycle. (A) As feet 2 and 4 swing forward, COM moves from A and B, with $F_y$ acting at ZMP point $p_1$. (B) COM continues forward past B, with ZMP now at $p_2$.](image)

![Fig. 5: ZMP gait support polygons (green shading, or line) during Fig. 4(A). Front left leg begins motion first, with partial overlap of swing phase with rear right foot.](image)

We tune the gait for speed under the constraint that no joint velocity can ever exceed the actuator limit: $q_i \leq 1.4$ (rad/s). The key tuning parameters are: (1) a desired initial offset in the center mass, offset by $dz_{start}$ from a symmetric, centered position within the initial four-foot stance in Figure 6(c), (2) step length, $dx_{step}$, and (3) half-swing time, $T_{half} = T_B = T_C$. Other nominal geometry parameters are set to be near the optimal values determined for the crawl gait, described next, except that the swing legs do not arc outward nearly as much. Given current actuator limits, the largest practical $dx_{step}$ is only 0.25 m, with $dz_{start} = -3$ cm and $T_{half} = 0.4$ s. Minimum velocity (see Equation 5), where the + marks occur, is about 0.06 (m/s), for an overall average forward speed of 0.087 (m/s) [17 (ft/min)].

V. CRAWL GAIT

We have explored a number of different strategies for determining end-effector trajectories and body poses while walking with a crawl gait, including Rapidly-exploring Random Tree (RRT) search algorithms [3]. While RRT’s work well for finding solutions on rough terrain or in confined spaces, they are relatively computationally expensive, and, more importantly, the resulting collision free trajectories are not optimized for walking speed.

To design fast steady state walking gaits, we previously applied formal methods such as gradient descent and value iteration to this problem [23], but these earlier solutions involved an overly restrictive gait parameterization – most particularly, by not allowing body rotations – and we are able to find solutions over twice as fast in this work, through iteratively sweeping across a larger parameter space.

Figure 7 shows a full gait cycle for the crawl walk, which includes six “phases”. In four of the phases (A, B, D, and E), a swing leg moves, assisted by body motion, and the remaining two (C and F) move the body while all four feet are in stance, to transfer the COM cleanly into the next upcoming support polygon. The gait exploits the symmetries (left-right and forward-back) of the robot to simplify parameterization.

In designing our gait, we define a number of gait characteristics which can be tuned with the goal of maximizing...
0.6 1.5 0.8 4 0.6 0.2 3 2 7 5 0.5 0.4 0.8 0.4
0
0.05
y (m)
x (m)
Perspective View
z (m)
0 0.2 0.4 0.6 0.8
−0.1
−0.05
0
0.05
x (m)
Overhead View
y (m)
which was used to produce our fastest crawl walk to date. The end effector trajectory is defined in the world coordinate system. The forward velocity of the robot. The coordinate convention here defines the x-axis as the direction of the walk, the y-axis perpendicular to x along the ground plane (to the robot’s left), and the z-axis pointed up from the ground plane. The gait variables include:

The end effector trajectory is defined in the world coordinate frame, using the gait variables and a cubic spline with seven waypoints. Figure 8 shows an end effector trajectory for a

dx_{step} = The closest in x that any two feet will get.
dy_{step} = The forward distance of a step.
dz_{swing} = Max outward arc of swing foot, in y.
dz_{foot} = How high to raise the foot to clear obstacles.
gbody = Height of center of body, above ground.
pitchbody = Body pitch amplitude during swing.
rrollbody = Body roll amplitude during swing.
dxbody = Body advance during swing.

TABLE I: Crawl gait – parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dx_{step}</td>
<td>The closest in x that any two feet will get.</td>
</tr>
<tr>
<td>dy_{step}</td>
<td>The forward distance of a step.</td>
</tr>
<tr>
<td>dz_{swing}</td>
<td>Max outward arc of swing foot, in y.</td>
</tr>
<tr>
<td>dz_{foot}</td>
<td>How high to raise the foot to clear obstacles.</td>
</tr>
<tr>
<td>gbody</td>
<td>Height of center of body, above ground.</td>
</tr>
<tr>
<td>pitchbody</td>
<td>Body pitch amplitude during swing.</td>
</tr>
<tr>
<td>rrollbody</td>
<td>Body roll amplitude during swing.</td>
</tr>
<tr>
<td>dxbody</td>
<td>Body advance during swing.</td>
</tr>
</tbody>
</table>

Fig. 6: Center of mass (com) trajectories of robot body for example ZMP gaits, with double-support phases. Note inflection points in the slope. By design, where com slope is at a local minimum (shown as + symbols in (a) and (c)), com passes directly over ZMP. At local maxima (shown as o symbols in (a) and (c)), ZMP experiences a discrete jump.

dx_{step} = 0.76 m, dz_{swing} = 0.1 m, and a dz_{foot} = 0.07 m, which was used to produce our fastest crawl walk to date.

One unintuitive point of note is that joint 7, which is the most distal (“wrist”) joint, was most typically the joint requiring the greatest total motion during swing leg trajectories, essentially acting as the bottleneck to the entire gait. During stance, joint 7 is rotates to prevent the foot slipping against the ground as the limb reconfigures to produce body motions. While the amount of rotation required varies with gait specifics, it often approach 180°. This rotation needs to be unwound during swing because each joint can only rotate a limited number of rotations before damaging the robot, and we plan this unwinding trajectory only after all other joint trajectories are planned, rather than pinning it to the IK table solution, to allow it to travel as quickly as possible to its final angle. As a comparison, the next largest displacement was at joint 1 and was on the order of 90°. Thus, while joint 7 seems of minor importance in moving a limb forward in space, it is actually a significant bottleneck in the crawl gait. This impact could possibly be reduced somewhat with future modifications to the IK tables.

Implementation on RoboSimian

For reference, RoboSimian (Fig. 1(b)) is a roughly human-scale robot, with a mass of around 90 kg and a wingspan of over 2 meters. 40% of its mass is in its body, with 15% in each limb, meaning limb motions affect COM significantly and cannot be ignored during planning. Also, the height of COM is roughly 0.5 m during walking, compared with LittleDog, with a COM around 12 cm high.

As discussed in detail in [1], RoboSimian has a distributed planning and control architecture with two quad-core I7 computers on the robot which process all of the onboard sensor data and control the limb motions. Motion planning can be done either on-board the robot or via an external computer connected to the robot through an Ethernet connection. For our testing, we performed all of the planning off-line on an external computer. Desired joint trajectories were then sent to the “control” computer on the robot for execution as a series of waypoints. The on-board control computer linearly interpolates between the waypoints and sends a higher resolution
joint trajectory to each of the microcontrollers controlling each joint. We separate each gait cycle into 6 sets of waypoints: one corresponding to each of the six phases in Figure 7.

VI. RESULTS

Both gaits were tested for a variety of parameterizations of the actual RoboSimian platform. Trials were recorded with video, to verify speed, and actual speeds during experiments on the robot were very close to our predicted values. (Specifically, rolling of the feet increases forward speed, but only by less than 5% of predicted values.)

For the crawl gait, initial testing required that all joint trajectories be slowed somewhat, for an overall speed of about 18 ft/min, due to joint tracking faults. With hardware improvements, we anticipate a forward walking speed of up to 25 ft/min [i.e., 0.29 (mph); 0.13 (m/s)], given a joint velocity limit of 1.4 (rad/s).

For the dynamic two-at-a-time gait, testing to date has used low swing leg clearance, for safety, resulting in some inelegant scraping of the feet. Despite this, however, the robot travels at the predicted speed of about 17 ft/min; i.e., most of the weight of the robot is supported by the stance feet, as planned. Recalling that this gait is designed conservatively, with brief pauses after diagonal pairs of legs moves and with only partial overlapping of front and rear leg motions, we anticipate that the fastest practical dynamic walking speed is, theoretically, approximately 1.5x this tested value, also very close to 25 ft/min. However, future testing is definitely needed to verify this can be achieved. In particular, the body currently resets any accumulated pitch error during a deliberate pause included in region $T_A$ of Figure 6(b).

In practice, the dynamic trot-walk seems to provide no speed advantage over a crawl, given current joint speed limits. Therefore, we focus only on the crawl gait in highlighting specific results below, due to space constraints.

A. Approximately Optimal Crawl Gait Parameters

As described in Section V, there are 15 parameters (10 spatial parameters and 5 temporal) over which we optimize the gait. Since the search space is already relatively large, we fix the temporal locations of trajectory waypoints, $dx_{fb} = 0.2$ m, $y_{step} = 0.6$ m, $dz_{foot} = 0.7$ m, and $b_z = 0.67$ m to simplify the analysis. Table II lists the remaining variables and the range over which we optimized them. We parameterize the fraction of overall forward body motions that happen simultaneously with swing leg motions as $(Rx_{body})$. As this approaches 1, there is no margin of safety for the COM within the support polygon during part of the gait. We test $Rx_{body}$ values between 0 and 0.8, but for our approximate optimal solution, we limit this to 0.6, for safety.

Figure 9(a) shows the effects of step length and body advance on the resulting gait speed for a fixed roll, pitch, yaw, and foot offset. Empty elements in the surface mesh indicate regions where the combination of parameters was not feasible. We could not reach further than 0.78 m, and a body shift is required to reach further than 0.74 m. While this Figure represents only a small subset of the search space (2% of the total mesh points), we can see two trends which are consistent through the search space:

1) The walk speed increases as $Rx_{body}$ increases.
2) The step length has a relatively modest effect on the walk speed, near its optimal value.

Figure 9(b) shows the effects of body yaw and $y$ trajectory offset of the swing leg. Increasing the body yaw amplitude generally increases feasible walking speed, but certain combinations reduce velocity. Similarly, the velocity initially increases with increasing $y$-offset (see Fig. 8), to arc the swing leg outward, but drops beyond an optimal offset of 0.1 m, used in our solution. Arcing the swing leg outward improves overall speed by approximately 20%, compared with using a constant $y$ value of the end effector (with all parameters held constant). Using $Rx_{body} = 0.6$ yields a 35% faster gait than $Rx_{body} = 0$, and $Rx_{body} = 0.8$ would push the gait speed to 28.5 (ft/min), all else remaining unchanged, while using the optimal yaw (-8°) and pitch (3°) instead of keeping body orientation constant improves speed by 25%.

VII. CONCLUSIONS AND FUTURE WORK

RoboSimian is designed to enable dexterous mobility in complex environments. High-torque, low-speed actuation facilitates climbing but also makes steady-state walking gaits quite slow, compared with (for example) human walking. Unlike most quadrupeds, RoboSimian also has 7-DOF limbs, opening the possibility of exploiting full-body motions in optimizing gait speed for a crawl. To achieve a more dynamic, two-at-a-time gait, we outline a basic ZMP-planning strategy that is both relatively simple and practical.

Based on earlier work with the smaller LittleDog robot, which has only three actuators per leg, we anticipated two-at-a-time walking would intuitively be significantly faster than a standard crawl gait, if feasible. For current velocity limits, however, both gaits are instead surprising similar in speed, both in theoretical limits (25 ft/min) and actual speed experimentally tested to date (17 ft/min) with hardware. Our future goal is to incorporate the fast crawl gait into our walk planner for RoboSimian, for use in the DRC.

ACKNOWLEDGEMENT

This work is supported by JPL NASA subcontract #1471138, for the DARPA Robotics Challenge (DRC).

| $dx_{step}$ | 0.4-0.8 m |
| $dy_{swing}$ | 0-0.15 m |
| $yaw_{body}$ | 0-10° |
| $pitch_{body}$ | -90° |
| $roll_{body}$ | -5-5° |

$Rx_{body} = dx_{body}/dx_{step}$; 0.6-0.8

TABLE II: Crawl gait: range of parameter search.
Fig. 9: (a) A subset of the analysis showing the effect of step length and body advance. (b) Walk velocity as a function of body yaw and (outward) y offset magnitude of swing leg, for a fixed roll, pitch, and step length. (c) Walk velocity as a function of body pitch and roll for a fixed step length, body yaw, body advance, and leg swing.

REFERENCES