Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2008
August 3-6, 2008, Brooklyn, New York, USA

DETC2008-49615

WORKSPACE ANALYSIS FOR THE LIMBS OF A HEXAPEDAL ROBOT WALKING GAIT GENERATION ALGORITHM DEVELOPMENT

Mark Showalter
Robotics and Mechanisms Laboratory
Department of Mechanical Engineering
Virginia Tech
Blacksburg, Virginia 24061
Email: mshowalt@vt.edu

Dennis Hong
Robotics and Mechanisms Laboratory
Department of Mechanical Engineering
Virginia Tech
Blacksburg, Virginia 24061
Email: dhong@vt.edu

ABSTRACT

The Multi-Appendage Robotic System (MARS) is a hexapedal robotic platform capable of walking and of performing manipulation tasks. Each of the six limbs of MARS incorporates a three-degree of freedom (DOF), kinematically spherical proximal joint, similar to a shoulder or hip joint, and a 1-DOF distal joint, similar to an elbow or knee joint. The generation of walking gaits for such robots with multiple limbs requires a thorough understanding of the kinematics of the limbs, including their workspace. Since the entire limb workspace cannot be used in a statically stable alternating tripedal gait for such a robot, a subset of the general limb workspace is defined to be used for walking gait generation algorithms. The specific abilities of a walking algorithm dictate the usable workspace for the limbs. Generally speaking, the more general the walking algorithm is, the less constricted the workspace becomes. In this paper we develop the workspaces for the limb of MARS in the knee up configuration, which range from simple 2D geometry to complex 3D volume, and analyze its limitations for use in walking on flat level surfaces. Next we discuss the case when the robot body is not parallel to the ground. The results from this paper can be applied to the development of walking gait generation algorithms.

NOMENCLATURE

workspace	a defined area or volume within which the limb tip can reach all points
gereral workspace	the broadest workspace in which the limb tip is guaranteed to be able to travel in a continuous straight line
adaptive gait	a gait which is continuously updated according to the input
stride lines	momentary straight line path of the robot limb tip while walking
non-contact	refers to limbs or limb tips which do not touch the walking surface
contact	refers to limbs or limb tips which touch the walking surface
limb switch	the process of taking a step: the contact limbs become non-contact and the non-contact limbs become contact
knee-up	refers to a limb configuration in which the distal joint of the limb makes an angle of 0 to -90 degrees about the z_4 axis; figures in this paper only depict limbs in a knee-up position
distal joint	the 1DOF joint farthest out the limb
proximal joint	the 3DOF joint which connects the limb to the robot body
distal section	the limb section between the distal joint and limb tip
proximal section	the limb structure between the proximal and distal joints
buffer cylinder	the cylindrical workspace boundary surrounding the z_2 axis
L_1	the length of the proximal limb segment: 12.7 cm (5 in)
L_2	the length of the distal limb segment: 15.24 cm (6 in)
r_B	the radius of the buffer cylinder
r_c	the radius of the spherical workspace

Figure 1: MARS (MULTI-APPENDAGE ROBOTIC SYSTEM), DEVELOPED AT ROMELA (ROBOTICS AND MECHANISMS LABORATORY).

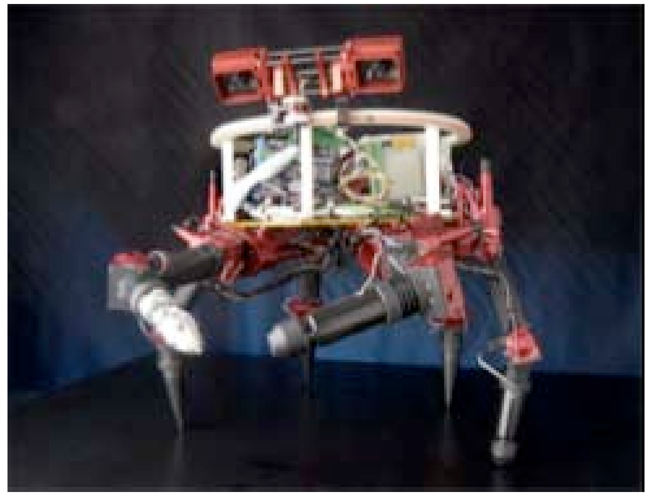

Figure 2: LEMUR IIB (LEGGED EXCURSION MECHANICAL UTILITY ROVER) FROM NASA JPL.

INTRODUCTION

This paper presents limb workspaces for a mobile robot, MARS (Multi-Appendage Robotic System) shown in Fig. 1. MARS is a hexapedal mobile robotic research platform patterned after the LEMUR IIb (Legged Excursion Mechanical Utility Rover) [1, 2, 3]. The LEMUR IIb, shown in Fig. 2, is the latest in a series of hexapedal robots developed at JPL for autonomous inspection and maintenance tasks on the exterior of space structures and vehicles in near zero gravity. The robot performs maintenance tasks by exchanging, via a quick connect, a foot for a tool. After positioning with the remaining limbs, the robot would then perform necessary repairs. These scenarios evoke many research possibilities including: wrench space analysis, robot and work object coordination, hull navigation, and walking algorithms. The focus of this paper is to develop limb workspaces for walking algorithms applicable to robots kinematically similar to LEMUR IIb.

The design of LEMUR IIb and MARS differ from biologicaly-inspired hexapedal robots [4-7] in symmetry. Quinn, Espenchied, et al., have developed highly mobile hexapedal robots patterned after the stick insect and cockroach. These robots employ two rows of three limbs bilaterally symmetric. However, by employing radial symmetry the LEMUR IIb and MARS platforms do not possess a set front and back, and are therefore capable of walking in any direction without turning. Rather than varying stride length or frequency to turn, MARS moves each limb tip in stride lines parallel to the current direction of motion.

To generate such movement of the limb tip in 3-space, a mathematical definition of the limb workspace is required. Knowing the boundaries of limb tip position is a basis for planning stride lines through the workspace as well as for planning for the transition from contact to non-contact. Fortunately, as all limbs have identical workspaces for MARS, a walking algorithm can be made generic to all the limbs. This walking algorithm would use distributed higher-level control planning. Each limb would be controlled separately based on body velocity and limb position on the body. Walking algorithms are covered in more detail in the companion paper [8].

The mathematical definitions of the 3D-limb workspace defined in this paper will serve as an integral component to such walking algorithms. Using a classical robotics approach, limb tip paths would be generated to form a walking motion within this defined workspace.

LIMB DESCRIPTION

MARS Limb Design

MARS is kinematically and dimensionally similar to JPL's LEMUR IIb robot [2]. Each of the six limbs has four revolute actuators and therefore four degrees of freedom (DOF). The limb attaches to the body of the robot with a 3-DOF proximal joint. In this joint the axes of three revolute actuators intersect orthogonally at a single point. The result is a kinematically spherical joint which can be equated with a ball and socket joint. The remaining 1-DOF distal joint uses a single revolute actuator located between

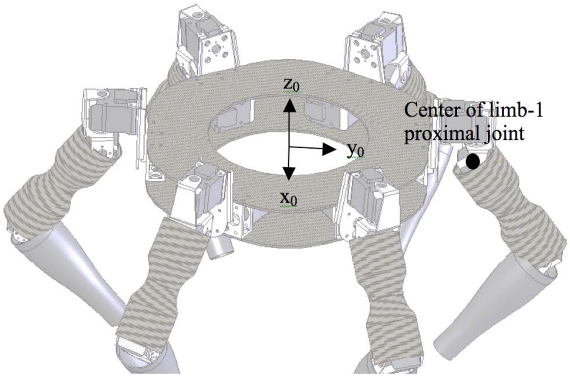

Figure 3: THE Y-AXIS OF THE BODY COORDINATE FRAME POINTS TOWARD THE CENTER OF THE PROXIMAL JOINT OF LIMB-1.

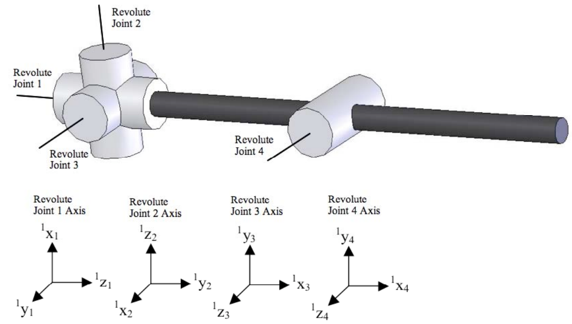

Figure 4: COORDINATE FRAMES ARE ASSIGNED TO EACH REVOLUTE JOINT ON A LIMB.

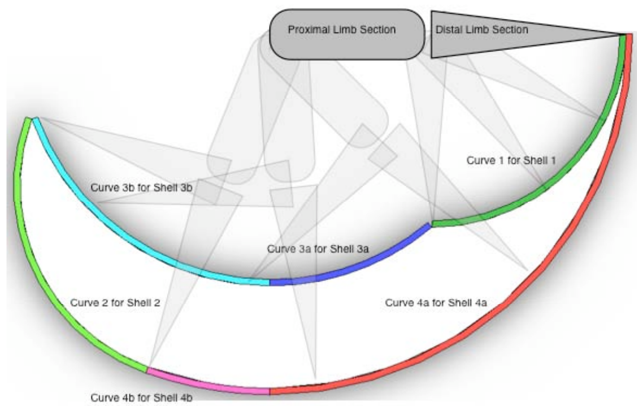

Figure 5: IN THE KNEE-UP CONFIGURATION THE 2D WORKSPACE IN THE Z_2-Y_2 PLANE IS THE AREA CONTAINED WITHIN FOUR CURVES.

the inner and outer limb sections.

As with the LEMUR class robots, the MARS limb design simplifies the kinematics, resulting in a large workspace [1]. The use of the spherical proximal joint simplifies the limb kinematics. Further, combining three of the four degrees of freedom at the base of the limb rather than distributing them along the limb increases the total workspace volume.

Carbon-fiber composite, aluminum, and polystyrene were used to form the structural limb and body components of MARS in order to reduce weight and maximize stiffness. This lightweight design allowed for the use of compact Dynamixel DX-117 actuators for all 24 revolute joints. The actuators provide sufficient torque for the robot to be fully supported by three limbs in any statically stable position. While these actuators are capable of 300 degrees of rotation, the joint rotations are structurally limited for three of the four degrees of freedom. The remaining degree of freedom, revolute joint 1, is not used for walking and therefore not considered in this workspace analysis.

Coordinate Frame Definitions

The workspace analysis covered here is based on Schmledeler, Bradley, and Kennedy [9]. For that reason their coordinate frame definitions will be followed. The coordinate frame of the body of the robot (x_0, y_0, z_0) is positioned at the center of the body with the y-axis pointing directly at the center of limb-1's proximal joint and the z-axis extending upward away from the body, as shown in Fig. 3. The revolute joints are then assigned coordinate frames in accordance with the Denavit-Hartenburg convention as shown in Fig. 4.

THE GENERAL KNEE UP WORKSPACE

The limb design of MARS is such that in walking stance the limb tip is generally pointed downwards. For this reason the proposed walking algorithm does not specify the orientation of the limb tip. As only the limb tip position in 3-space and not the orientation is specified, only three rather than six constraints are available for the inverse kinematics. For this reason, while each limb has 4-DOF, the first revolute joint is not used for the walking motion. By constraining revolute joint-1, only two configurations are available for a given tip position; "knee-up" and "knee-down" configurations.

Geometric Delineation of the General Knee Up Workspace

For the walking algorithm only the knee-up workspace is used. The geometry of the general knee-up workspace can be completely defined mathematically. The 2D workspace is examined in the z_2-y_2 plane as shown in Fig. 5. Sweeping revolute joints 3 and 4 through their respective ranges causes the limb tip to reach all points within the area bounded by curves 1 through 4. Curves 1 and 2 are the result of sweeping revolute joint 4 though its range while revolute joint 3 is at the two extremes of its range. Curves 3 and 4 are the result of sweeping revolute joint 4 through its range while revolute joint 3 is at the two extremes of its range. All four curves are mathematically defined in Table I.

Table 1. MATHEMATICAL 2D WORKSPACE DEFINITIONS

Curve Number/Shell Type	Curve Equation
1/torus	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{L_2^{\,2}}\cos\theta + 5 \\ \sqrt{L_2^{\,2}}\sin\theta \end{bmatrix}, \theta = \frac{-90\pi}{180}...0$
2/torus	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{L_2^{\,2}}\cos\theta - 5\sin\frac{20\pi}{180} \\ \sqrt{L_2^{\,2}}\sin\theta - 5\cos\frac{20\pi}{180} \end{bmatrix}, \theta = \frac{-200\pi}{180}...\frac{-110\pi}{180}$
3/sphere	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sqrt{L_1^{\,2} + L_2^{\,2}}\cos\theta \\ \sqrt{L_1^{\,2} + L_2^{\,2}}\sin\theta \end{bmatrix}, \theta = \left(\frac{-110\pi}{180} - a\tan\frac{6}{5}\right)...-a\tan\frac{6}{5}$
4/sphere	$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (L_1 + L_2)\cos\theta \\ (L_1 + L_2)\sin\theta \end{bmatrix}, \theta = \frac{-110\pi}{180}...0$

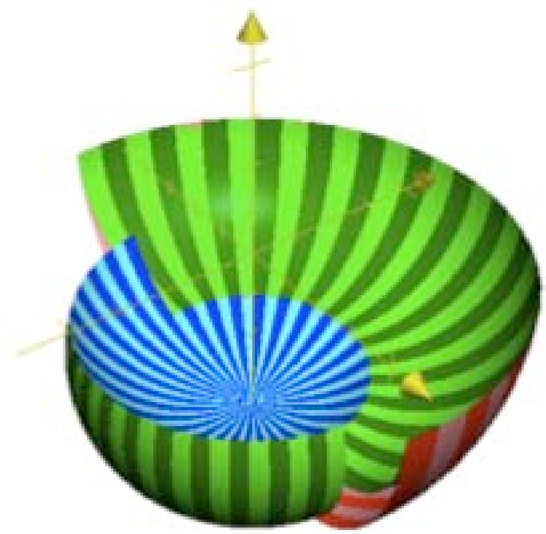

Figure 6: SIX SHELLS FORM THE INITIAL 3D KNEE-UP WORKSPACE.

Rotating the 2D workspace about the z_2 axis through the range of revolute joint 2 generates the 3D workspace shown in Fig. 6. The shells which comprise the 3D workspace are formed by sweeping the 2D curves about the z_2 axis through the range of revolute joint 2. However, as 2D curves 3 and 4 cross the z_2 axis, they are separated into shells 3a, 3b, 4a, and 4b. All six shells are mathematically defined in Table II. However, the shells in Fig. 6 do not fully contain the workspace. In addition to these six shells, two planar section also bound the workspace. These planar sections lie on two planes, both of which contain the z_2-axis. These two planes intersect the x_2 y_2 plane at -10 degrees and 190 degrees respectively. As the workspace is symmetric on either side of the y_2 z_2 plane, the two planar sections are identical. The form of these sections is shown in Fig. 7. Each planar section is formed from two areas bounded by arcs. The equations of the circles which form these arcs are also shown in Fig. 7. It should be noted that the equations for the arcs are based on a separate coordinate system specific to the respective planes

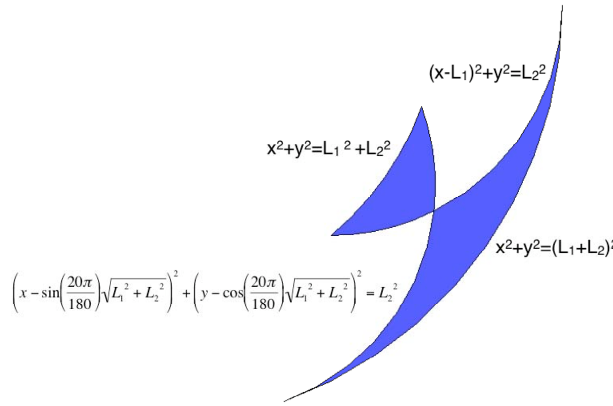

Figure 7: PLANAR SECTIONS WHICH FORM PART OF THE BOUNDARY FOR THE WORKSPACE ARE ENCOMPASSED BY ARC SECTIONS OF CIRCLES..

Table 2. 3D WORKSPACE MATHEMATICAL DEFINITIONS

Shell Number/Shell Type	**Shell Equation**
1/torus	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (L_1 + L_2\cos v)\cos u \\ (L_1 + L_2\cos v)\sin u \\ L_2\sin v \end{bmatrix}, u = \frac{-10\pi}{180}...\frac{190\pi}{180}, v = \frac{-90\pi}{180}...\frac{0\pi}{180}$
2/torus	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \left(L_1\sin\left(\frac{20\pi}{180}\right) + L_2\cos v\right)\cos u \\ \left(L_1\sin\left(\frac{20\pi}{180}\right) + L_2\cos v\right)\sin u \\ L_2\sin v - 5\cos\left(\frac{20\pi}{180}\right) \end{bmatrix}, u = \frac{190\pi}{180}...\frac{350\pi}{180}, v = \frac{20\pi}{180}...\frac{-70\pi}{180}$
3a/sphere	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sqrt{L_1^{\,2} - L_2^{\,2} - u^2}\cos\theta \\ \sqrt{L_1^{\,2} - L_2^{\,2} - u^2}\sin\theta \\ u \end{bmatrix}, \theta = \frac{-10\pi}{180}...\frac{190\pi}{180}, u = 6...-\sqrt{L_1^{\,2} - L_2^{\,2}}$
3b/sphere	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sqrt{L_1^{\,2} - L_2^{\,2} - u^2}\cos\theta \\ \sqrt{L_1^{\,2} - L_2^{\,2} - u^2}\sin\theta \\ u \end{bmatrix}, \theta = \frac{190\pi}{180}...\frac{350\pi}{180}, u = \sqrt{L_1 + L_2}\cos\left(\frac{20\pi}{180} + a\tan\frac{L_2}{L_1}\right)...-\sqrt{L_1^{\,2} - L_2^{\,2}}$
4a/sphere	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sqrt{(L_1 + L_2)^2 - u^2}\cos\theta \\ \sqrt{(L_1 + L_2)^2 - u^2}\sin\theta \\ u \end{bmatrix}, \theta = \frac{-10\pi}{180}...\frac{190\pi}{180}, u = 0...-(L_1 + L_2)$
4b/sphere	$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \sqrt{(L_1 + L_2)^2 - u^2}\cos\theta \\ \sqrt{(L_1 + L_2)^2 - u^2}\sin\theta \\ u \end{bmatrix}, \theta = \frac{190\pi}{180}...\frac{350\pi}{180}, u = -11\cos\left(\frac{20\pi}{180}\right)...-(L_1 + L_2)$

containing the sections.

This mathematical definition of the workspace boundary is necessary for the walking algorithm. The walking algorithm operates in the space domain rather than the actuator domain. For this reason, defining the boundaries of the workspace in the space domain is a necessary preliminary to formulating the walking algorithm.

General Knee Up Workspace Limitations

While the limb tip can reach any point within the workspace in the knee-up configuration, it is not necessarily possible to travel continuously from one point in the workspace to another. Fig. 8 shows a top view of the workspace. The limb tip can travel continuously within the

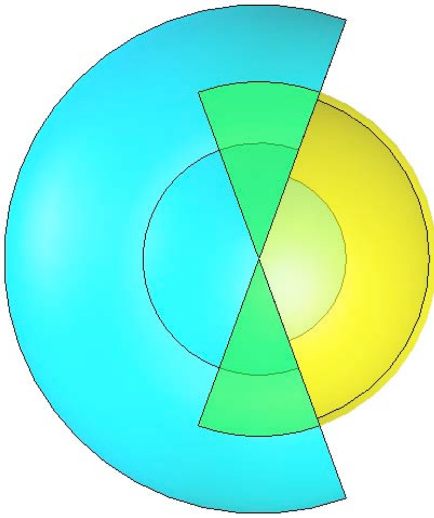

Figure 8: MOVEMENT OF THE LIMB TIP FROM THE BLUE REGION OF THE WORKSPACE TO THE YELLOW REGION REQUIRES AN INSTANTANEOUS 180 DEGREE ROTATION OF REVOLUTE JOINT-2.

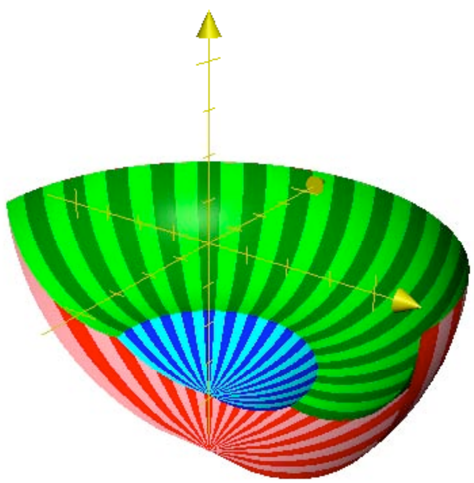

Figure 9: THE WORKSPACE IS LIMITED DUE TO CONSTRAINTS ON CONTINUOUS MOVEMENT OF THE LIMB TIP.

blue and green regions or the limb tip can travel continuously within the yellow and green regions. However, it is not possible for the limb tip to travel continuously from the blue region to the yellow region. Such a motion would require a rotation of 180 degrees of revolute joint-2 in the green region. Due to the complexity of incorporating this requirement into a continuously updating walking algorithm, the yellow region was removed from the workspace. As a result, only shells 1, 3a, and 4a are used to bound the workspace. The workspace assumes the form in Fig. 9, and the planar sections assume the form in Fig. 10.

The Buffer Cylinder

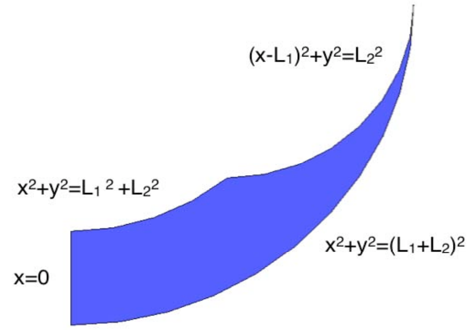

Figure 10: THE PLANAR SECTIONS ALSO CHANGES DUE TO LIMITING THE WORKSPACE.

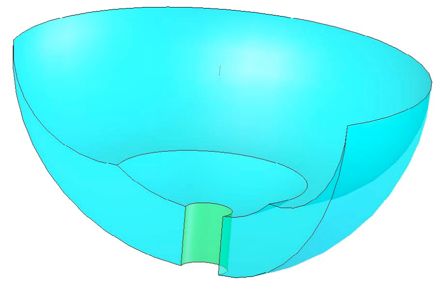

Figure 11: THE BUFFER CYLINDER, ABOUT HE Z_2-AXIS, FURTHER LIMITS THE WORKSPACE.

While it is possible for the limb tip to reach a range of points on the z_2-axis, it is not necessarily possible for the limb tip to move near the z_2-axis. Limb tip motions near the z_2-axis which do not require the movement of revolute joint 2 are possible. For example: by freezing revolute joint 2, revolute joints 3 and 4 can still be used to trace a line from the z_2-axis radially out. Limb tip motions near the z_2-axis which require the movement of revolute joint 2 may not be possible do to the velocity limit of revolute joint 2. For example: tracing a line, at a finite speed, which passes infinitesimally close to the z_2-axis would require infinite rotational velocity of revolute joint 2 as it rotates nearly 180 degrees. The farther this traced line is from the z_2-axis, the slower the required rotational velocity of revolute joint 2 for a given tracing speed. Therefore, for a given walking speed and maximum rotational velocity of revolute joint 2, the required minimum distance a stride line must pass from the z_2-axis, can be calculated. A cylinder about the z_2-axis with this distance for its radius can be removed from the workspace. This cylinder is referred to as the buffer cylinder. With the buffer cylinder the workspace changes, as shown in Fig. 11. Also the vertical boundary of the planar sections is moved away from the y-axis by the radius of the buffer cylinder as shown in Fig. 12.

Discussion of the General Workspace

Walking algorithms based on the gereral workspace

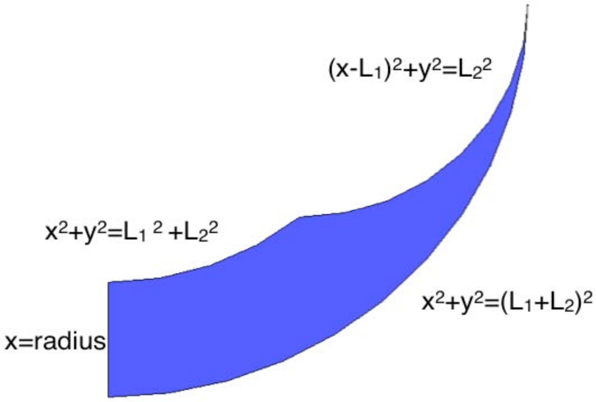

Figure 12: THE PLANAR SECTIONS CHANGE SLIGHTLY DUE TO THE BUFFER CYLINDER.

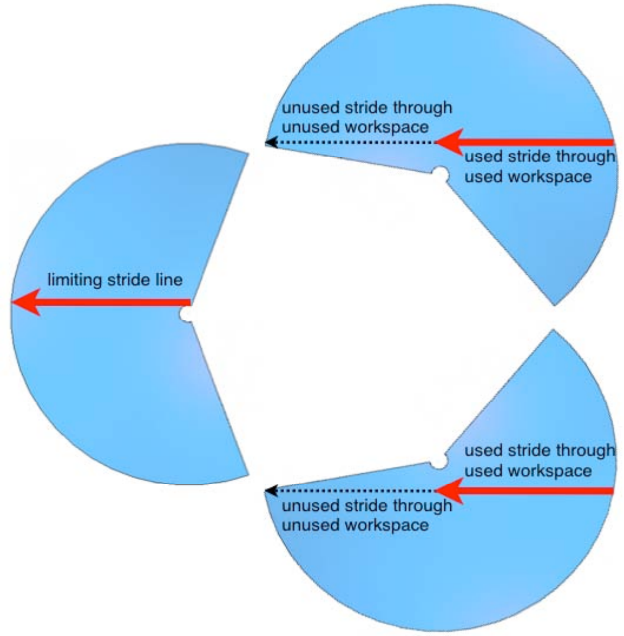

Figure 13: ALL THREE CONTACT LIMB STRIDE LINES ARE LIMITED BY THE SHORTEST STRIDE LINE.

have both advantages and disadvantages. The general workspace is a complex 3D volume bounded by sections of two spheres, a torus, 2 planes, and a cylinder. An adaptive walking algorithm would require line intersection calculations with each of these sections for each limb at each iteration. For this reason the algorithm would be quite complex and require adequate computing power. Another draw-back to the use of the general workspace is that the walking algorithm for a robot dimensioned similarly to MARS would also need to prevent the limbs from colliding with each other while walking. However, the general workspace is large, providing for lengthy stride lines. Further, a walking algorithm designed for the general workspace need not be limited to dimensions similar to MARS. In fact, such an algorithm could work with any robot with six or more limbs. The limbs could be attached to the body at any orientation and height, so long as they could reach the walking surface. Also, the body of the robot could be any shape, so long as static stability was achievable through all gait arrangements.

MARS SPECIFIC WORKSPACE LIMITATIONS

Limb arrangement on the body determine the further limitations on the work space. Because the walking algorithm requires a limb switch when the first contact limb reaches its workspace boundary, the work space of the other two contact limbs is essentially limited by the workspace of the one contact limb. In other words, for a given stride, the longest stride all three contact limbs can make is limited to the shortest of the three individual stride lines. This concept is illustrated in Fig. 13. Notice that though limbs 3 and 5 have not reached the boundary of their respective workspace, they are cut short by limb 1, which has reached its workspace boundary. Because all strides are limited to the shortest stride, the usable workspace of all three contact limbs is limited. The resulting workspace is found by overlaying the three contact limb workspaces as shown in Fig. 14 for 2D. The workspaces are overlain so that the largest common workspace will result. The largest common workspace roughly resembles a circle. For ease of programming the common workspace was limited to the circle shown in Fig. 14.

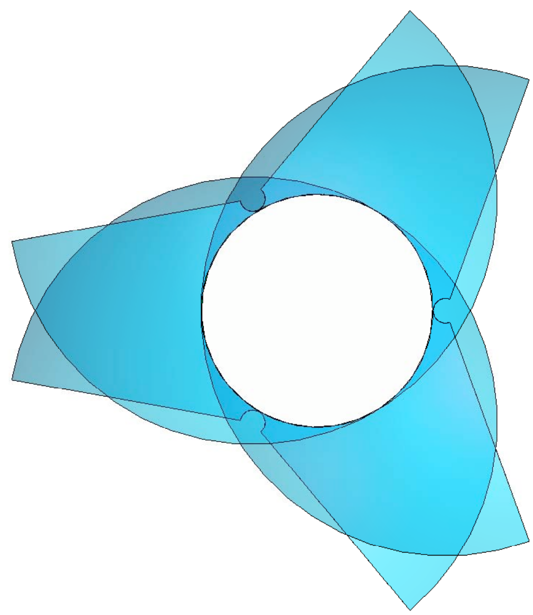

Figure 14: OVERLAYING THE WORKSPACES OF THE THREE CONTACT LIMBS REVEALS THE COMMON WORKSPACE IN THE STYLE OF A VENN DIAGRAM. THE CIRCLE IS A SIMPLIFICATION OF THE COMMON WORKSPACE.

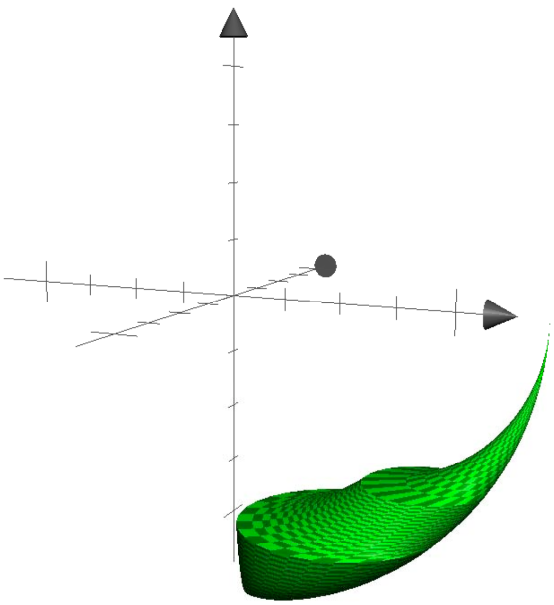

Figure 15: A 3D VOLUME CONTAINS THE SET OF 2D CIRCULAR WORKSPACES. THIS VOLUMES CENTER AND DIAMETER ARE GIVEN BY THESE TWO LINES. THESE LINES WERE GENERATED FOR A PROXIMAL SECTION LENGTH OF 5, A DISTAL SECTION LENGTH OF 6, AND A BOUNDARY CYLINDER DIAMETER OF 0.5.

Walking with the Robot Body Parallel to the Walking Surface

A 2D representation of the workspace is sufficient for walking level. When the body of MARS is parallel to the walking surface then the walking surface is parallel to the x_2-y_2 plane of each limb. While the x_2-y_2 planes for the limbs are parallel to the walking surface, only a 2D slice of the workspace needs to be considered for the walking algorithm. For this condition the workspace for each limb will be a circle. The piecewise Eqn. (1) defines the circle diameters:

$$D = \begin{cases} \sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - \left(\sqrt{\left(L_2^2 - z_2^2\right)} + L\right) & if \quad 0 \geq z_2 > -L_2 \\ \sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - \sqrt{L_1^2 + L_2^2 - z^2} & if \quad -L_2 \geq z_2 > -\sqrt{L_1^2 + L_2^2 - r_B} \\ \sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - r_B^2 & if \quad -\sqrt{L_1^2 + L_2^2 - r_B} \geq z_2 \geq -\sqrt{\left(L_1+L_2\right)^2 - r_B^2} \end{cases} \tag{1}$$

where L_1 and L_2 are the lengths of the proximal and distal limb sections, respectively; D is diameter of the circular workspace; and r_B is the radius of the buffer cylinder. The location of the center of the circle is given by the piecewise Eqn. (2):

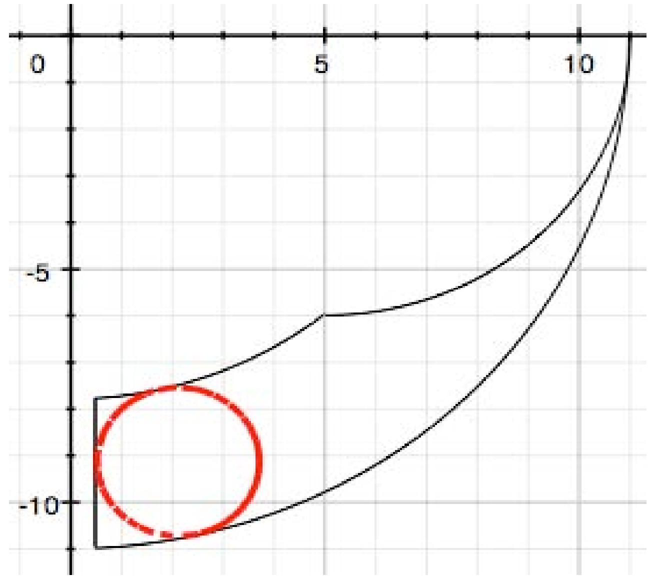

Figure 16: A SPHERICAL WORKSPACE SIMPLIFIES THE WALKING ALGORITHM WHILE ALLOWING ROBOT BODY ROLL AND PITCH.

$$y_2 = \begin{cases} \dfrac{\sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - \left(\sqrt{\left(L_2^2 - z_2^2\right)} + L\right)}{2} & if \quad 0 \geq z_2 > -L_2 \\ \dfrac{\sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - \sqrt{L_1^2 + L_2^2 - z^2}}{2} & if \quad -L_2 \geq z_2 > -\sqrt{L_1^2 + L_2^2 - r_B} \\ \dfrac{\sqrt{\left(\left(L_1+L_2\right)^2 - z_2^2\right)} - r_B^2}{2} & if \quad -\sqrt{L_1^2 + L_2^2 - r_B} \geq z_2 \geq -\sqrt{\left(L_1+L_2\right)^2 - r_B^2} \end{cases} \tag{2}$$

where the center point is located on the y_2-z_2 plane. The 3D workspace for each limb, provided the robot is walking parallel to the walking surface, is shown in Fig. 15.

At this point it is possible to generate a walking algorithm which uses this workspace to walk parallel to the walking surface. Due to the limited size of the workspace, there is no risk of the limbs colliding with each other while walking. It should be noticed that the robot can theoretically walk at any height between 0 and 11 inches. However, the dimensions of the limbs limit this to roughly 2 to 11 inches.

Walking with Robot Body Roll and Pitch

Walking with the robot body not parallel to the walking surface complicates the use of the workspace. If 2D slices of the workspace are used, as with level walking, the slices will be of different shapes and sizes for each of the three contact limbs. With this approach the three 2D shapes would need to be calculated for each limb, for each iteration of the walking algorithm. However, the same results can be achieved by finding points of intersection of the stride line with the workspace boundary. For this reason, this approach is not examined in this paper. However, two other approaches are examined:

- Mathematically define the workspace, as with the shell method, and find the intersection points of stride lines with the workspace boundary.

- Select a spherical workspace which fits within an overlay of the 3D workspaces of all three contact limbs.

Using the spherical workspace as the basis for a walking algorithm requires that the sphere be mathematically defined. The sphere is defined, as shown in Fig. 16, as tangent to shell-3a, shell-4a, and centered on the y_2-z_2 plane. These constraints for the spherical workspace define its size, but there is still a range of locations for the workspace: tangent to shell-1 (farthest possible from the body), tangent to the buffer cylinder (closest to the body), and on a continuum between these two extremes. The condition, tangent to the buffer cylinder, was selected to reduce motor torques. A sphere in this location is mathematically defined by Eqn. (3,4):

$$x_2^2 + \left(y_2 - \left(r_B + r_c\right)\right)^2 + \left(z_2 + \sqrt{\left(\left(L_1 + L_2\right) - r_c\right)^2 - \left(r_B + r_c\right)^2}\right)^2 = r_c^2 \quad (3)$$

where:

$$r_c = \frac{\left(L_1 + L_2\right) - \sqrt{L_1^2 + L_2^2}}{2} \quad (4)$$

The small size of this spherical workspace would require more frequent limb switching. However, the geometric simplicity of the sphere, as compared to a set of shells, would reduce computing time and increase the algorithms iteration rate.

CONCLUSION

A general 3D workspace for walking can be described for knee up adaptive walking algorithms for robots with six or more limbs kinematically similar to MARS' limbs. This general workspace accommodates any or all of the following physical modifications:

- the robot could be a range of sizes and shapes
- the limbs could be attached at different heights and angles
- the limbs could be of different dimensions

By applying further constraints specific to the MARS platform, the workspace can be redefined depending on the desired functionality of the walking algorithm. These further constrained workspaces fit within the general workspace. A 2D workspace can be used if robot body roll and pitch are not required during walking. Inclusion of roll and pitch while walking necessitates a 3D workspace. Two such 3D workspaces were addressed:

- a simple sphere which simplifies walking algorithm development and programming, but decreases average stride length
- a complex shape formed from three shells, which complicates walking algorithm development and programming, but allows for longer stride length

Development of walking algorithms for MARS-specific 3D workspaces will enable the robot to walk on level terrain while changing direction, height, roll, pitch, and yaw. Later work will focus on walking on uneven terrain with the same functionality.

ACKNOWLEDGMENTS

Thanks go to Daniel Larimer of Open Tech Inc. for work with the MARS-specific workspace limitations and development of a 2D walking algorithm.

REFERENCES

[1] Kennedy, B., Agrazarian, H., Cheng, Y., Garrett, M., Hickey, G., Huntsberger, T., Magnone, L., Mahoney, C., Meyer, A., and Knight, J., 2001, "LEMUR: Limbed Excursion Mechanical Utility Rover," *Autonomous Robots*, 11(3), pp. 201-205.

[2] Kennedy, B., Agrazarian, H., Cheng, Y., Garrett, M., Huntsberger, T., Magnone, L., Okon, A., and Robinson, M., 2002, "Limbed Excursion Mechanical Utility Rover- LEMUR II," 53rd International Astronautical Congress, Houston, Texas, October 10-19.

[3] Wagner, R., Hobson, L., and Kennedy, B., 2005, "Autonomous Walking Inspection and Maintenance Robot," Advances in the Astronautical Sciences, Vol. 121, pp. 235-249.

[4] Espenchied, K. S., Quinn. R. D., Chiel, H. J., Beer, R. D., 1993,"Leg Coordination Mechanisms in the Stick Insect Applied to Hexapod Robot Locomotion," The Massachusetts Institute of Technology, Adaptive Behavior Vol. 1. No. 4. pp. 455-468.

[5] Espenchied, K. S., Quinn. R. D., Beer, R. D., Chiel, H. J., 1996, "Biologically based distributed control and local reflexes improve rough terrain locomotion in a hexapod robot," Robotics and Autonomous Systems 18, pp. 59-64.

[6] Quinn, R. D. and Ritzmann, R. E., ""Construction of a Hexapod Robot with Cockroach Kinematics Benefits both Robotics and Biology," Connection Science, 10(3-4), December 1998.

[7] Allen, T.J, Quinn, R.D., Bachmann, R.J., and Ritzmann, R.E., 2003. "Abstracted Biological Principles Applied with Reduced Actuation Improve Mobility of Legged Vehicles," IEEE International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas.

[8] Showalter, M., Hong, D., "Development and Comparison of Gait Generation Algorithms for Hexapedal Robots Based on Kinematics with Considerations for Workspace", DETC2008.

[9] Schmiedleler, James P., Bradley, Nathan J., and Kennedy, Brett, 2004. "Maximizing Walking Step Length for a Near Omni-Directional Hexapod Robot," Proceeding of the 2004 AMSE International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Salt Lake City, Utah, September 28 – October 2.