USER INTERFACE WITH MULTISENSORY FEEDBACK FOR FLUID POWERED RESCUE ROBOT

Hannes DAEPP[1] and Wayne BOOK[2]
Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
801 Ferst Drive
Atlanta, GA 30332-0405
USA
Email: hdaepp@gatech.edu

A semi-autonomously controlled fluid-powered legged search and rescue robot is proposed as a solution to the ever-increasing demand for more versatile rescue robot technology. The success of such a robot is dependent on the existence of a user interface that optimizes the balance between user and machine decisions and provides the operator with the appropriate amount of information to soundly make such decisions. The system design, consisting of a simulation/physical robot, on-board computer, and operator interface are described. Aspects of the relation between operator input and robot motion are discussed, such as the influence of feedback on operator actions or the integration of input into semi-autonomous gaits. The basis for tests seeking to optimize the interface design is established.

Keywords: fluid power, rescue robot, haptic feedback, user interface

1 INTRODUCTION

Figure 1: Destruction following the 2010 Haiti earthquake – a typical environment for a search and rescue robot.

The recent 7.0 magnitude earthquake that rocked Haiti left in its wake a path of rubble, agony, and devastation (Figure 1). Rescue teams were forced to deal with harsh terrain, limited resources, and minimal time for action. This is the type of scenario in which rescue robots aim to prove themselves. There are many research centers that are actively working to enhance the role of robots in disaster recovery, such as the Center for Robotic-Assisted Search and Rescue (CRASAR) in the United States or the International Rescue Systems Institute in Japan. Past and current research efforts have focused largely on the endurance and search abilities (Messina et al., 2005), rather than the need to actually rescue the victims or provide victim assistance (Schneider, 2009). A recent survey (Figure 2) of actual fire brigades and search teams found manipulative abilities in rescue

[1] PhD candidate
[2] Supervisor

robots to be highly desirable (Driewer; Baier; Schilling, 2005). There is a need for more versatile rescue robots that are able to handle manipulation tasks as well as the ability to effectively navigate challenging and unpredictable terrain.

Legged locomotion has been studied for years in biology and engineering alike, and has been shown to provide an excellent solution to the challenges of varying landscapes (Song and Waldron, 1988). The demand for rescue robots to be able to manipulate potentially heavy objects as well as traverse tough terrain can be met through fluid powered robots. Fluid-powered legged robots can provide steady control of large external loads and higher power density than their electrically actuated equivalents. The Compact Rescue Robot (CRR), a project of the NSF Center for Compact and Efficient Fluid Power (CCEFP), seeks to demonstrate the advantages of fluid power. To counteract difficulties encountered in past control efforts of fluid-powered walking machines, the CRR places large emphasis on operator interface design.

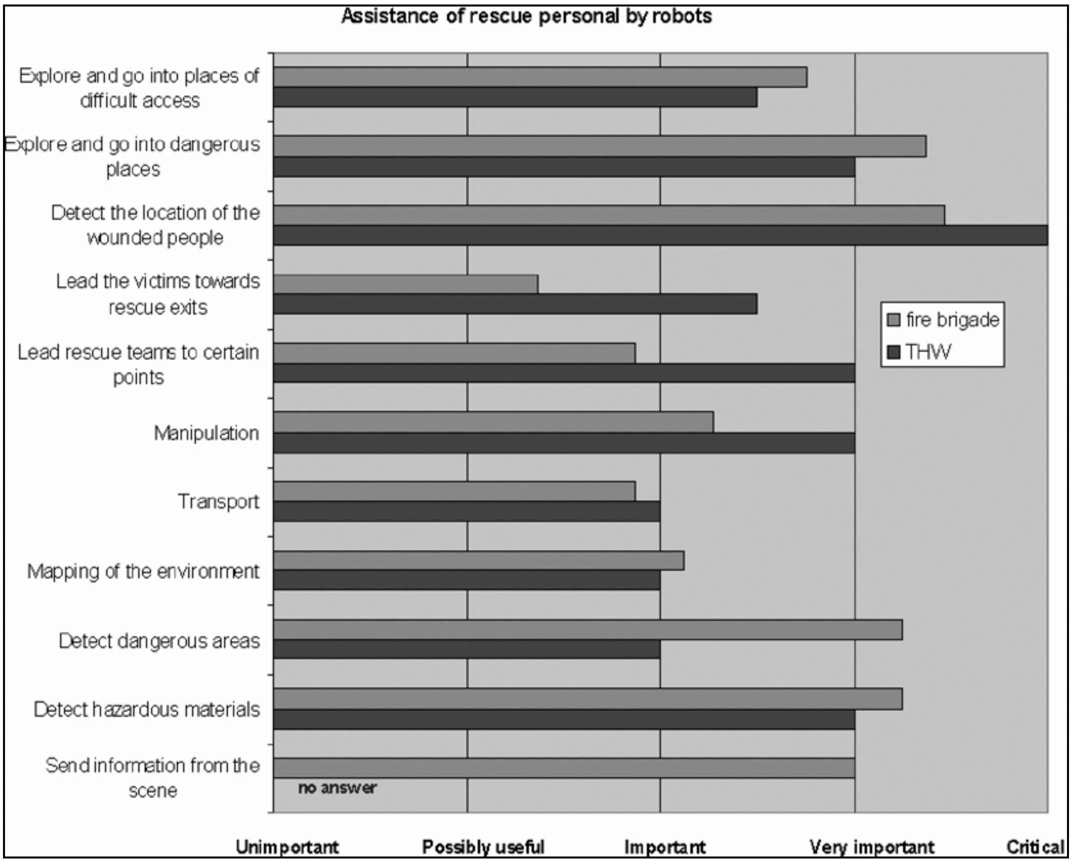

Figure 2: Desired assistance of rescue personnel by robots

One key to optimal performance lies in the design of the operator interface: using a semi-autonomous control scheme to effectively delegate tasks to the intuition and skill of the operator or computational effectiveness of the computer. This principle provides the cornerstone for the CRR interface, which allows a human, equipped with multimodal feedback, to tele-operatively manipulate the front legs, while the system determines the appropriate position for the rear legs. To ensure that the user is comfortable performing such tasks, an excellent sense of embodiment must be attained. The CRR achieves this effect through audio and visual feedback, but places particular emphasis on the study of haptic interfaces. Improved haptics has been shown to have a more substantial effect on proper operator tele-presence than the enhancement of its visual counterpart (Lee, 2008). Haptics is also efficient, providing signals that concisely provide comprehensive, intuitive directional and magnitude related information through direct interaction with the user

(Gentry; Wall; Oakley; Murray-Smith, 2003), providing less ambiguous feedback than auditory or visual warning signals.

The operator workstation used for the CRR uses an approach to gait control known as a Follow-the-Leader gait. The gait, well-known for its usage on rough terrain, allows the operator to place the front two legs, while the rear legs are placed autonomously. The operator interface combines this gait with the touted advantages of multimodal feedback, with particular emphasis on haptics, to establish a fluid-powered robot that will have improved versatility in motion with respect to previous similar machines. This paper presents a basis for interfacing an operator with a quadruped search and rescue robot. This interface will be adjusted to obtain the optimal balance of user control and computational effort for effective performance of a fluid-powered rescue robot.

2 MODELS AND METHODS

The system is made up of three components: the operator interface, a PC104 target xPC, and the simulation host computer/robot. As this project is part of a collaborative CCEFP effort, the focus at Georgia Tech is placed on operator interface design. It is validated with a four-legged robot dynamics and environment simulation as well as a two-legged robot for verification of physical constraints. A full quadruped version of the hardware is in development at Vanderbilt.

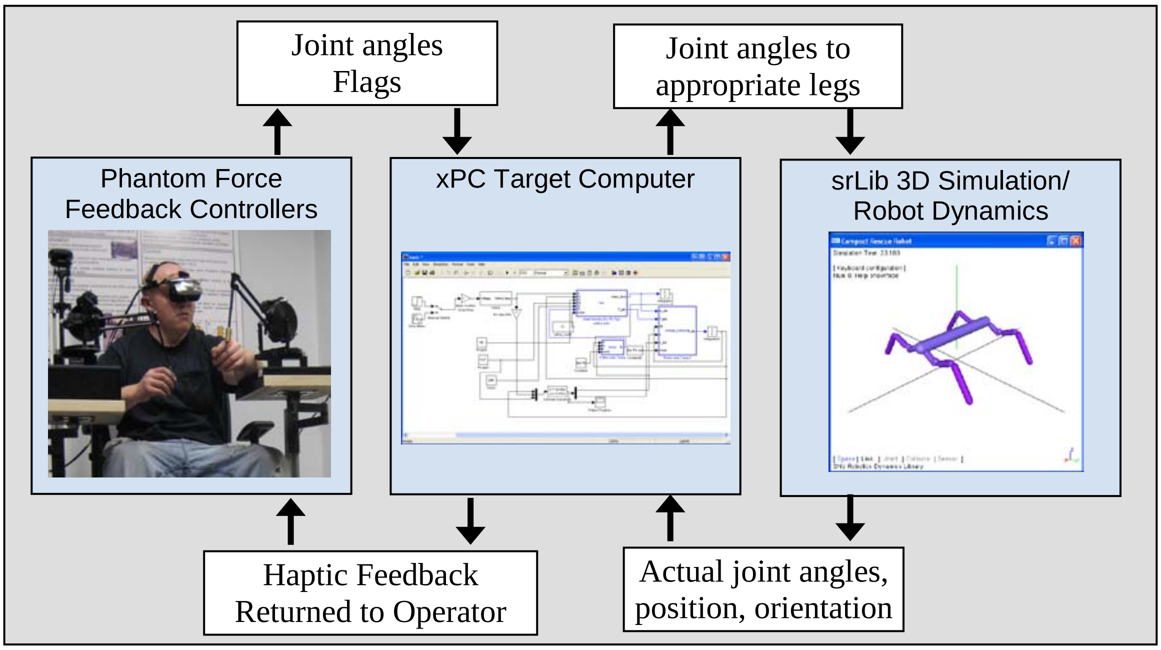

Figure 3: Compact Rescue Robot System Layout

Integration of these components into the system is illustrated in Figure 3. Communication between the three parts is as follows:

1. Operator moves the endpoint of the Phantom. The endpoint coordinates of the Phantom are sampled by the operator workstation, converted to endpoints in the local robot leg space, and transformed to leg joint angles.
2. Each set of three joint angles is transmitted, via wireless network, to the xPC target, along with flags from the operator workstation that specify the leg to which the joint angles should be routed.
3. Real-time software on the target PC routs these joint angles according to the supplied flags, sending the appropriate joint angle commands via User Datagram Protocol (UDP) to the dynamics engine—either the robot or its equivalent simulation.

4. The actual motions are translated to joint angles and sent back to the xPC target, along with matrices representing the change in the robot's global position and orientation.
5. The xPC target calculates center of gravity (CoG), end effector locations, and stability parameters, and sends the joint angles back to the operator workstation.
6. Joint angles are converted to phantom endpoint locations and used to provide haptic feedback to the operator.

2.1 Hardware

The operator workstation and the robot provide the physical hardware for the CRR. The PC104 computer, used as the computational engine for the robot, doubles as the real-time target PC for the simulation, ensuring a duality between simulation and hardware.

2.1.1 Robot Design

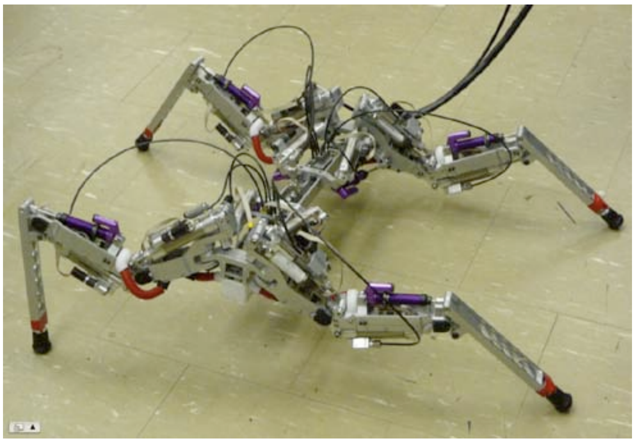

Figure 4: Four-legged version of robot in development at Vanderbilt University

The general robot design is best described as a long spine with four 3 degree of freedom (DoF) legs that use pneumatic cylinders to actuate each joint (Figure 4). This straightforward construction allows simple mounting of the PC104 computer, as well as a camera for Visual feedback, pressurized air tanks, and power supply on the central axis. Legs are further equipped with pressure sensors and potentiometers at each actuating cylinder. The actuation and signal measurement design of a single leg is depicted in Figure 5.

As previously mentioned, the robot used by Georgia Tech is currently simulated. An additional physical version with just front legs and a cart for rear support provides an efficient means of simulation validation, as well as a basis for examining real-life motion and design constraints.

A local coordinate frame at the base of the robot's spine is used as the basis for robot leg motions. Each end effector is mapped to the main robot frame by way of a leg frame, as shown in Figure 5.

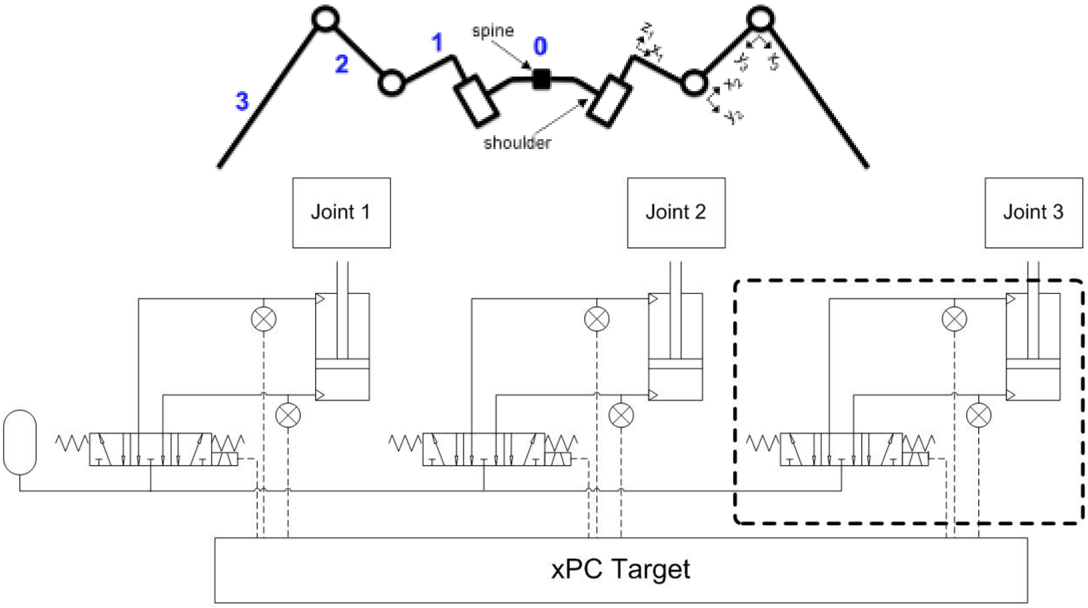

Figure 5: ***Top*** **- Robot leg space (corresponds to front robot view) with respect to spine (at point 0).** ***Bottom*** **- Schematic for a single leg (there are four of these on the robot). The dashed box shows the single actuator, which excludes robot joint dynamics, that was modelled for use with simulated dynamics in section 2.2.**

The position of the end effectors in leg space was determined using the Denavit-Hartenberg method to define the forward kinematics of the robot:

	θ	d	α	a
0	--	--	0	0
1	θ_1	1.608	-90	5.750
2	θ_2	0	0	6.828
3	θ_3	0	--	12.00
4	0	0	--	--

Table 1: Denavit-Hartenberg Parameters (units in inches)

An algorithm defining the leg inverse kinematics has been established to calculate joint angles from any given end effector position.

2.1.2 Operator Interface

Figure 6 depicts the operator interface, consisting of a chair with two 3 DoF haptically enabled Phantom joysticks. The user is further provided with feedback through the use of a headset with audio/visual output. The headset includes a rotation sensor, such that when the operator turns their head to move about, the on-board camera moves accordingly. Data to and from the interface is processed and directed by an attached PC workstation.

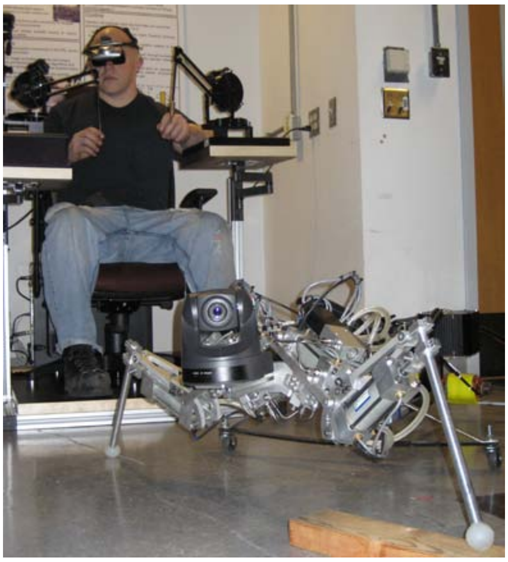

Figure 6: Operator interface and Two-legged Robot model before design revision

A major focus within the operator interface is the use of haptic response to guide operator decisions. The robots' four-legged structure allows for simpler gait motions and better power-weight ratio than a robot with additional legs, but incurs stability constraints on the motion that must be related back to a controlling operator. One prospective approach is to actively calculate the level of static stability, using techniques such as the stability margin (Frank and McGhee, 1968) or the Stability Approximating Lines (Pack and Kak, 1995). The resulting quantity would be supplied to the user via non-traditional haptic cues, such as resistance fields or vibrations. Such an approach would make use of the noted efficacy of haptics over other modalities without inhibiting its more general application as an operator sense for terrain and dynamics characteristics and constraints.

To steer the robot, the joysticks provide an end effector location to a C++ script located on the PC workstation, which is then translated to joint angles that are sent to the xPC target. Inputs from switches on the joysticks and from the keyboard are used to set flags that correspond to various states of motion in the outgoing commands. It is advantageous to perform all possible computation here, as the workstation is very powerful in comparison with the xPC target. For example, one proposed method of achieving successful Follow-the-Leader gait implementation involves recording joint trajectories of the front leg, manipulating the data, and then replaying the altered data back at a specified later time. This is achieved in an efficient manner by recording the joint angles to a file on board the workstation, saving this file, and then later reopening the [altered] file, reading the data, and sending the joint angles to the xPC target along with an appropriate set of flags. Thus, through efficient usage of computational resources, it is possible to greatly increase the complexity of the system design while maintaining effectiveness.

2.2 Dynamic Simulation

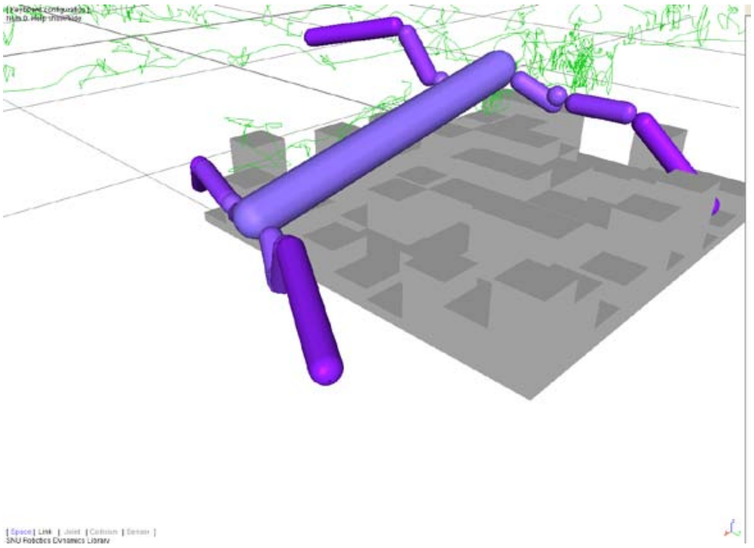
[Space] Link | Joint | Collision | Sensor]
SNU Robotics Dynamics Library

Figure 7: Graphic output of the SrLIB dynamic simulation

Premium operator performance is achieved through a parallel modification of operator workstation and robot hardware designs, control schemes, and interfacing software. The simulation plays a key role in facilitating such modifications by providing a safe and efficient means of testing changes in design and control of both the operator interface and the robot itself on the operational ability of the system on a selection of virtual terrains.

The simulation provides a computational equivalent to the actual robot dynamics and sensor output. The Seoul National University's Robotics Library (SrLIB) was used to model the legged robot. SrLIB is an open source library for multi-body dynamics and simulation in real-time. SrLIB's library is composed of simple rigid body shapes, joint types, actuation methods, and sensors. The libraries are built upon and modified for more accurate simulations, such as through the inclusion of joint limits and definition of inertial and friction coefficients for each rigid body. The simulation also establishes a method to test the robot's versatility on assorted terrain types. Using shapes in SrLIB, obstacle fields are constructed for the robot to interact with. Figure 7 shows the graphical output of the simulated robot crossing an obstacle.

Library links and joints are used to construct a robot representative of the four-legged version in development at Vanderbilt, possessing the kinematic design discussed in section 2.1 (see figure 7). SrLIB provides a real-time displacement vector and direction cosine matrix corresponding to the position and orientation, respectively, of the local robot coordinate frame, equivalent to sensors placed on the actual robot.

To maintain the equivalence between simulated and physical robot, an accurate representation of fluid power actuation is critical. A pneumatic actuator simulation, consisting of a valve and cylinder model, was developed in SIMULINK for use on the xPC target. This first edition simply models the actuator itself, validating results against open-loop and closed-loop comparisons with the actual hardware. Future directions will take the simple actuator and incorporate it into the full dynamic simulation by providing an output torque in place of the output force, based on the instantaneous joint geometry. The actuator will receive position feedback based on the outputs of the comprehensive dynamic simulation, srLib.

The valve model is based on the Festo MPYE-5-M5 proportional directional control valve used on the robot. Voltage spanning a 10 V input range is zeroed, fed through discontinuities such as a dead zone and saturation block, and then multiplied by an appropriate gain to provide a proportional positive or negative orifice area output. The valve block was verified by comparing input voltage versus measured flow rates to manufacturer's data, which it matched closely.

Modeling the cylinder is done by inspecting each side of the cylinder independently and coupling the two sides into a single dynamics equation. A control volume must be drawn around each side and an energy balance written for that control volume based on the mass flow calculated by the valve model and the volume change calculated by the dynamics equation and pressure equilibrium (Al-Dakkan; Barth; Goldfarb, 2006). The flow rate through each side of the valve can be independently calculated based on Equation (1), where $\dot{m}$ is mass flow, C_d is the discharge coefficient, A_0 is the orifice area, P_u and T_u are the upstream pressure and temperature, respectively, and P_d and T_d are the downstream pressure and temperature, respectively. Temperature is calculated with the ideal gas law, using the instantaneous total mass and pressure in the cylinder.

$$\dot{m} = C_d A_0 f\left(P_u, T_u, {}^{P_d}\!\big/_{P_u}\right) \tag{1}$$

Critical Pressure Ratio for air P_d / P_u =.528

If P_d / P_u > Critical (Un-Choked Flow):

$$f\left(P_u, T_u, {}^{P_u}\!\big/_{P_d}\right) = C_1 \frac{P_u}{\sqrt{T_u}} \left(\frac{P_d}{P_u}\right)^{1/k} \sqrt{1 - \left(\frac{P_d}{P_u}\right)^{(k-1)/k}} \tag{2a}$$

$$C_1 = \sqrt{\frac{2k}{R(k-1)}}$$

If $P_d / P_u \le$ Critical (Choked Flow):

$$f\left(P_u, T_u, {}^{P_u}\!\big/_{P_d}\right) = C_2 \frac{P_u}{\sqrt{T_u}} \tag{2b}$$

$$C_2 = \sqrt{\frac{k}{R\left(\frac{k+1}{2}\right)^{(k+1)/(k-1)}}}$$

Assuming that the system is adiabatic, an energy balance can be constructed as shown in Eq. (3)

$$\dot{P} = \frac{kRT\dot{m}}{xA} - \frac{P\dot{x}}{x}\left(\frac{kR}{c_p} + 1\right) \tag{3}$$

The dynamics of the cylinder are represented by equation 4, where *F* is the output force, P_p is the piston-side pressure, A_p is the piston-side area, P_r is the rod-side pressure, A_r is the rod-side area, P_{atm} is the atmospheric pressure, A_s is the rod shaft area, *b* is the viscous damping coefficient between the piston and cylinder wall and *m* is the mass of the piston and rod. The coefficient *b* is approximated by matching the simulation output with the physical output, meaning that it likely also encompasses some other performance-affecting friction terms. This output force is converted to an output torque based on the instantaneous specific joint geometry.

$$F = P_p A_p - P_r A_r - P_{atm} A_s - \dot{x}b - m\ddot{x} \tag{4}$$

The actuation model was compared with a test setup of an actual cylinder and valve and adjusted to match realistic open-loop dynamics. Based on past control approaches to control of these pneumatic actuators, PID control was chosen to be applied to the model. It was tuned so that the output would follow a sine wave, representative of a continuously changing leg motion, within 5% error. Tests on the actual hardware proved that by taking the algorithm and replacing the derivative terms with transfer functions that instead sampled over several periods, a response in the actual hardware (Figure 8) was shown to be near identical, with small deviations due to simplifications in the model. As mentioned previously, the next iteration of the simulation will be include the controlled pneumatic actuator in real time on the MATLAB host to ensure that the user is made fully aware of the effect that fluid power has on situational dynamics.

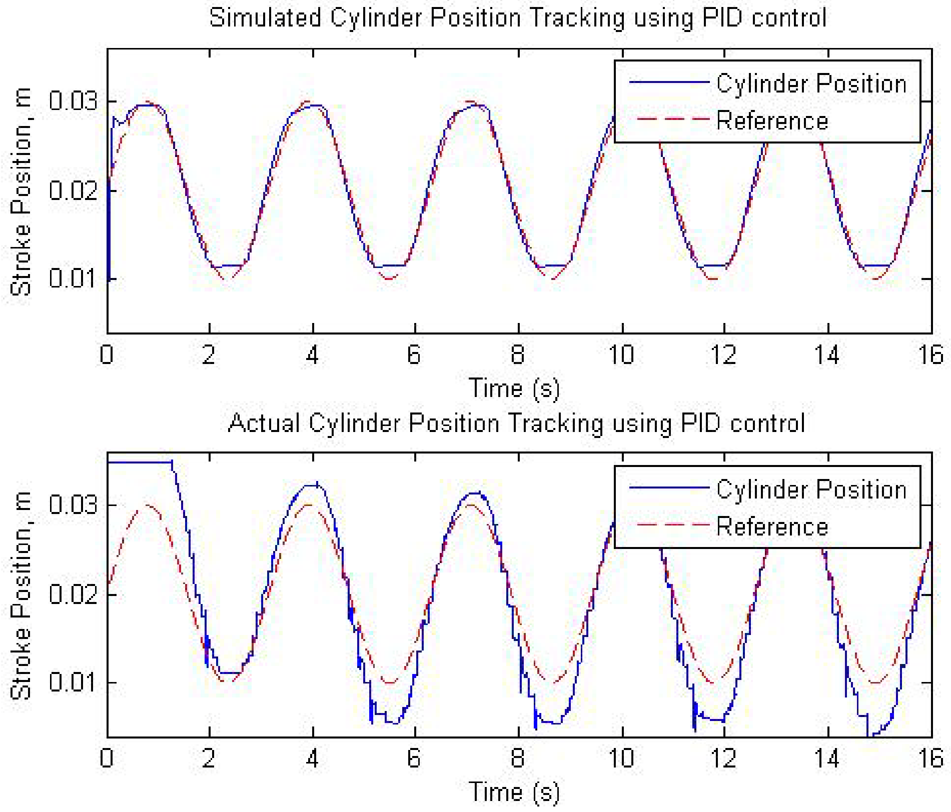

Figure 8: ***Left:*** **Comparison of closed loop behavior in pneumatic actuator simulation (top) and in practice (bottom).**

3 GAIT CONTROL

An important aspect of the operator interface is the ability of the software to map the actions of the user, operating two haptically enabled joysticks, to the motion of the four-legged robot. The general form of gaits chosen by the CRR is the Follow-the-Leader (FTL) gait, a semi-autonomous type of control in which a user controls the front two legs of a robot while the rest are machine activated. FTL optimizes decision delegation, ensuring that the human makes decisions subject to external constraints, while the machine makes unconditional ones. Within this class of gaits, there are variations, such as leg placement order, that can result in very different motion.

Assuming a general gait, the states of the leg can be broken down into three categories: *stance*, *swing*, and *shift*. These correspond, respectively, to standing still, moving through the air, and moving joint angles to affect the CoG without changing the end effector positions. As a rule for statically stable motion, *swing* motions are only possible if they can be performed without causing the robot to tip. There are several known methods of measuring the stability of the robot, most notably via stability margins within a stability polygon, the shape generated by the horizontal projection of the lines connecting the anchored end effectors (Frank and McGhee). The stability margin represents the shortest distance to a potential tipping axis on the polygon edge. A more computationally efficient alternative is Hirose's Stability Approximating Lines, or SALs. These lines, which connect two diagonally opposing end effectors, divide the robot into four quadrants. Depending on the location of the CoG within any of these quadrants, static stability may or may not be achieved.

Using these methods as a basis for judging leg motion, several preliminary gaits were developed. These were divided into two primary categories: *complete control* and *guided gait control*. In the first case, an operator controls all legs individually, whereas in the latter, the operator controls just the front two legs, while the rear legs follow with a predetermined gait. For *complete control*, the leg states are split into two categories: *manipulate*, which encompasses both *stance* and *swing*, and *shift.* The gait is then outlined in the following steps:

1. Operator indicates the leg he/she wants to manipulate
2. Rescue robot shifts its center of mass to a stable location
3. Phantom manipulators adjusts its position to reflect the foot he/she wants to control
4. Operator can manipulate the leg
5. Operator can repeat these steps to manipulate a different leg

Originally, *complete control* was intended to maneuver across large obstacles and extreme terrain. However, simulation results showed that this control is not capable of such tasks because it does not produce forward motion. Instead, the robot shifts back and forth until it reaches its original center of mass with some of its joints at their limits.

There are three general types of gait sequences that are considered for *guided gait control*: moving the front legs first and then back legs, moving the legs of one side first then the other, and moving the legs in a zig-zag order. In addition, the order in which the leg and body moves can vary for some of the gaits. Of these options, the zig-zag order (Figure 9) has the largest tripod area (stability polygon), which makes it the most stable for walking. Notice that the robot takes half steps in order to avoid leg collisions.

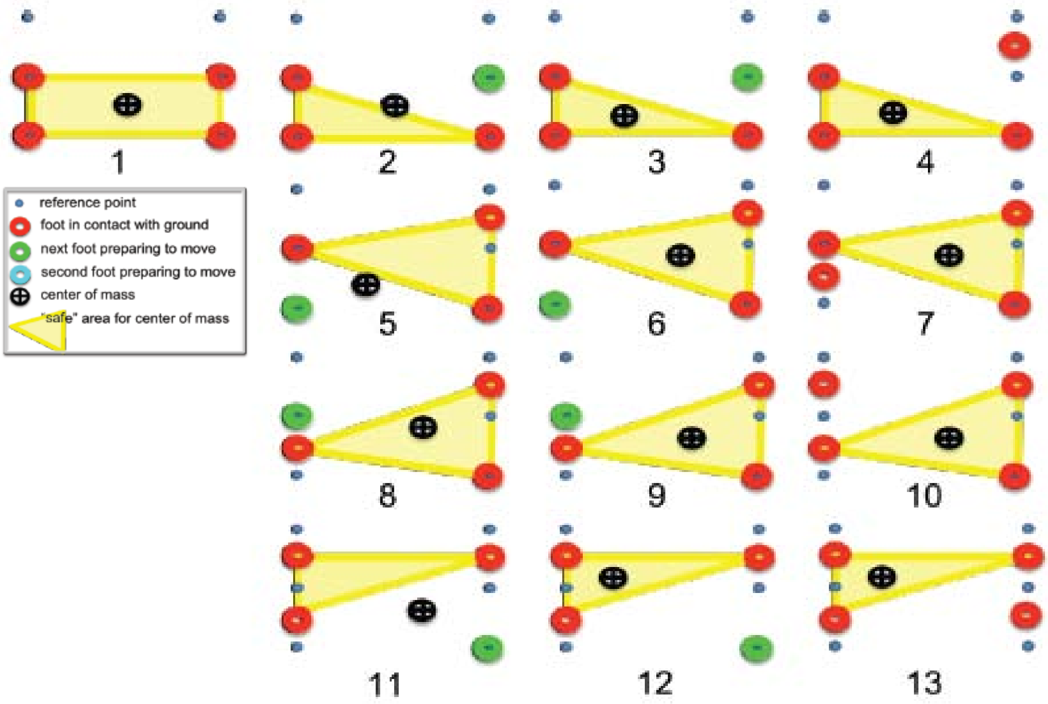
1
2
3
4
reference point
foot in contact with ground
next foot preparing to move
second foot preparing to move
center of mass
"safe" area for center of mass
5
6
7
8
9
10
11
12
13

Figure 9: End Effector motions used in zig-zag gait approach. The goal is to maintain static stability by ensuring that whenever less than four legs are on the ground, the center of mass is within the "safe" area. As shown, Zig-zag order has the largest possible stability polygon of those examined.

5 CONCLUSION

The operator workstation and accompanying simulation provide an excellent means for demonstrating the value of fluid power when combined with a user interface that optimizes the user's ability to control a fluid-powered legged robot. By filtering operator motions with a computationally versatile workstation executable and intelligent gait design, the operator will be allowed to make simple decisions based on skill and intuition to provide the best results. The operator's ability to make these decisions is enhanced through good tele-presence, achieved through application of several forms of feedback, with particular emphasis on haptics. Force and position feedback, provided by the Phantom joysticks, allow the operator to experience first hand the advantages and constraints that fluid power has on the system. Additionally, creative application of distinct haptic cues, such as force fields or vibrations, can be used to quantify the robot's stability and map this sensation back to the operator, thus guiding the user in effective foot placement. Because the simulation guarantees that each situation is identical, this setup can be used to isolate individual modalities and test their effectiveness in improving speed and motion control of the robot. Future studies will be conducted to determine how feedback and gait design can be matched to achieve an optimal balance of user and computer input, minimize stress on the operator, and culminates in an interface that allows the advantages of fluid-powered actuation to be fully realized in the legged rescue robot scenario.

6 ACKNOWLEDGEMENTS

The authors would like to thank Ta Kim and Peter Radecki for previous work on the SrLIB adaptation, hardware design, and actuator model development. Additionally, this work benefitted from hardware/software support from J.D. Huggins and collaborations in robot development at Vanderbilt and human factors at North Carolina A&T.

7 LIST OF NOTATIONS

CoG	Center of Gravity	[]
DoF	Degrees of Freedom	[]
$\theta_1, \theta_2, \theta_3$	Rotation angles of robot leg for joints 1, 2, and 3, respectively	radians
$\dot{m}$	Mass flow rate	kg/m^3
C_d	Discharge coefficient	[]
A_0	Orifice area	m^2
P_u	Upstream pressure	Pa
T_u	Upstream temperature	ºK
P_d	Downstream pressure	Pa
T_d	Downstream temperature	ºK
k	Ration of specific heats for air	1.4
R	Universal gas constant for air	287 J/(kg*K)
C_1	Constant 1	.1562
C_2	Constant 2	.0404
$x, \dot{x}, \ddot{x}$	Actuator position, velocity, acceleration	m, m/s, m^2/s
T	Instantaneous internal cylinder temperature	ºK
F	Actuator force	N
P_p	Piston-side pressure	Pa
A_p	Piston-side area	m^2
P_r	Rod-side pressure	Pa
A_r	Rod-side area	m^2
P_{atm}	Atmospheric pressure	Pa
b	Viscous damping coefficient	kg/s
m	Mass of the piston and rod	kg

8 REFERENCES

Al-Dakkan, K.A.; Barth, E.J.; Goldfarb, M. (2006). Dynamic constraint-based energy-saving control of pneumatic servo systems. *Transactions of the ASME Journal of Dynamic Systems, Measurement and Control, Vol 128, n 3, pp.655-62.*

Driewer F.; Baier, H.;Schilling, K. (2005). Robot–human rescue teams: a user requirements analysis. *Advanced Robotics, Vol. 19, no. 8, pp. 819-838.*

Frank, A.A; McGhee, R.B. (1968). On the stability properties of quadruped creeping gaits. *Mathematical Biosciences, Vol. 3, No. 3-4, pp. 331-52.*

Gentry, S; Wall, S.; Oakley, I.; Murray-Smith, R. (July 2003). Got rhythm? Haptic-only lead and follow dancing. *Proceedings of Eurohaptics Conference, Dublin, pp. 481-488.*

Lee, S. *et al* (2008). Effects of haptic feedback, stereoscopy, and image resolution on performance & presence in remote navigation. *Int. J. Human-Computer Studies, Vol. 66, pp. 701-17.*

Messina, E. *et al*. (May 2005). Statement of Requirements for Urban Search and Rescue Robot Performance Standards. *NIST. Preliminary Rep.*

Pack, D.J.; Kak, A.C. (1995). A simplified forward gait control for a quadruped walking robot. *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems. Advanced Robotic Systems and the Real World, Vol.2, pp. 1011-18.*

Schneider, D. (February 2009). Robin Murphy Roboticist to the Rescue. *Spectrum, IEEE , Vol.46, no.2, pp.36-37.*

Song, S.;Waldron, K.J. (1988). *Machines That Walk: The Adaptive Suspension Vehicle*. New York: The MIT Press.