Proceedings of 2010 ISFA
2010 International Symposium on Flexible Automation
Tokyo, Japan July 12-14, 2010

UPS-2740

AN INTERACTIVE SIMULATION FOR A FLUID-POWERED LEGGED SEARCH AND RESCUE ROBOT

Wayne J. Book
Georgia Institute of Technology
801 Ferst Drive
Atlanta, GA 30332
wayne.book@me.gatech.edu

Hannes G. Daepp
Georgia Institute of Technology
801 Ferst Drive
Atlanta, GA 30332
hdaepp@gatech.edu

Ta Y. Kim
Georgia Institute of Technology
801 Ferst Drive
Atlanta, GA 30332
ta.y.kim@ gatech.edu

Peter P. Radecki
Michigan Technological University
1400 Townsend Drive
Houghton, MI 49931
pradecki@mtu.edu

ABSTRACT

A pneumatically actuated search and rescue quadrapedal robot is presented as a system with potentially enhanced versatility relative to existing rescue robots. The usage of fluid powered actuation, combined with tele-operation of the robot via an operator workstation, enables the 12 degree of freedom robot to better manipulate large objects and provide on-site victim assistance than existing rescue robots, which are often limited solely to assisting in search functions. To better examine the system's capabilities, a simple model of a pneumatic actuator is created and then integrated into a simulation that allows the user to manipulate a model of the robot in a virtual environment. Constraints on simulation design and control for optimal performance are discussed and implementation and potential further impact are presented.

NOMENCLATURE

A	Area	m^2
C_1	Constant 1	
C_2	Constant 2	
C_d	Discharge coefficient	.016
CoG	Center of Gravity	
DoF	Degrees of Freedom	
F	Force exerted by actuator	N
P	Pressure	Pa
R	Universal gas constant (air)	287 J/kg K
T	Instantaneous internal cylinder temperature	ºK
b	Viscous damping coefficient	kg/s
k	Ratio of specific heats (air)	1.4
m	Mass of piston and load	kg
$\dot{m}$	Mass flow rate	kg/m^3
$x, \dot{x}, \ddot{x}$	Actuator position, velocity, acceleration	m,m/s,m^2/s
$\theta_1, \theta_2, \theta_3$	Rotation angles of robot leg for joints 1,2, and 3, respectively	degrees

Subscripts

atm	Atmospheric
d	Downstream
p	Piston-side
r	Rod-side
O	Orifice
u	upstream

INTRODUCTION

In the wake of catastrophic disasters, rescue teams are often 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 and the International Rescue Systems Institute in Japan. Yet the actual state of such technologies in existence is less than ideal: the current focus in most rescue robots is placed on endurance and search [1] rather than actual rescue [2] and victim assistance ability. This state of affairs affirms the need for more versatile robots that are able to handle manipulation tasks as well as the ability to effectively navigate challenging and unpredictable terrain [3].

By providing steady control of large external loads and higher power density than their electrically actuated equivalents, fluid-powered legged robots can provide a solution to this demand. 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 various landscapes [4]. The Compact Rescue Robot (CRR), a testbed of the NSF Center for Compact and Efficient Fluid Power (CCEFP), seeks to demonstrate the advantages that fluid power brings to the rescue robot field.

The CRR further enhances its effectiveness through the use of haptic feedback to the operator. Improved haptics has been shown to have a more substantial effect on proper operator tele-presence [5] than the enhancement of its visual counterpart. Haptics is also efficient, providing signals that concisely provide comprehensive, intuitive directional and magnitude related information through direct interaction with the user [6], providing less ambiguous feedback than auditory or visual warning signals.

The CRR simulation, described below, presents a comprehensive basis for evaluation of fluid power in robotics. The simulation couples modeling of pneumatic actuation with a dynamics and environment simulation. The simulation effectively provides a prototype of the physical hardware and allows researchers to view the effects of system designs and control techniques that have not previously been studied with relative ease. Integration of the dynamic simulation into the real-time environment also enables researches to fully examine altered dynamics and effectiveness of control schemes following system modification. Additionally, the simulation provides flexibility in design of the operator interface: The outputs are related to the operator via graphic output and haptic feedback, providing user insight to the effects of fluid power on legged rescue robot versatility. The ease of modification and testing on the simulation make it possible to safely alter the system design and operator interface hardware and software in parallel, quickly viewing the change in overall performance, and determining the right combination of parameters to encourage the best operator performance.

ROBOT CONFIGURATION

	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)

The general robot design consists of a long spine with four 3 degree of freedom legs that use pneumatic cylinders to actuate each joint. A local coordinate frame on the spine orients the robot in space. End effector positions are specified using Denavit-Hartenberg parameters (Table 1) to determine positions with respect to local frames at each shoulder mount (Figure 1).

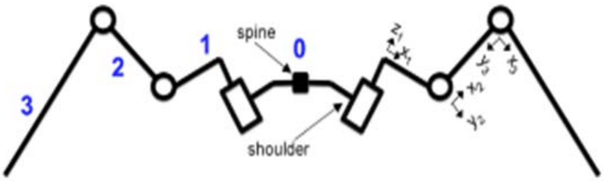

Figure 1: Front view of robot - joint labelling

The shoulders are then mapped to the base frame to return the global location of the end effectors. Additional inverse kinematic algorithms have been generated to calculate the joint angles corresponding to a particular end effector position.

SIMULATION ARCHITECTURE

The system is made up of three components: the operator interface, a PC104 target xPC, and the robot dynamics engine.

The operator interface consists of a chair and two SensAble Phantom controllers: 3 degree of freedom joysticks with haptic feedback. A computer performs all necessary data conversions and communications operations. The phantoms are used to allow the operator to place the legs of the robot. To encourage optimal leg placement, most of the gait combinations used are based on a Follow-the-Leader gait, in which the user places the front two legs, and the rear legs are determined based on knowledge of the robot's balance and local environment.

The dynamics engine is the computational equivalent of the robot, which communicates with the target and operator interface in a fashion almost identical to that of SrLib, making them interchangeable in the system design.

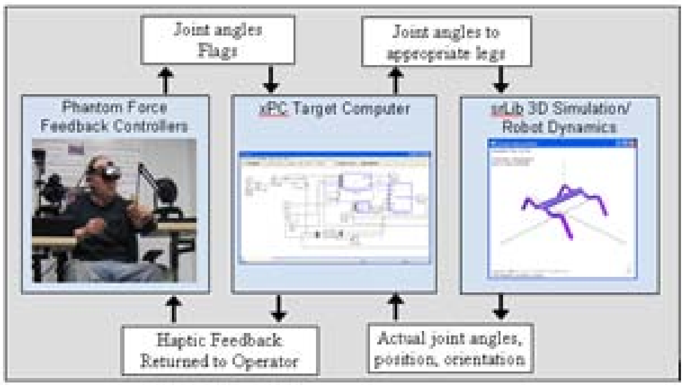

Figure 2: Interaction of Simulation Components

Physical verification of the simulated behavior can be attained through a two-legged hardware model, as well as a four-legged physical model in development at Vanderbilt University, a fellow CCEFP member.

Integration of these components into the system is illustrated in Figure 2. 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. The geometries are identical for each leg.

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 trajectories are transformed 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.

DYNAMICS ENGINE

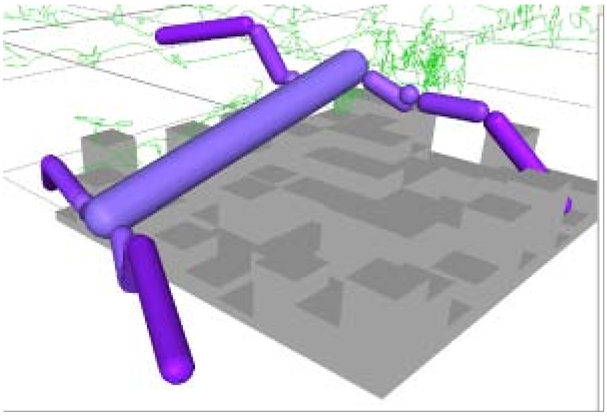

Figure 3: Graphic output of SrLib dynamic simulation

The dynamics engine 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, 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 3 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 University (Figure 4), possessing the kinematic design discussed in section 2.1 (Figure 1). 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.

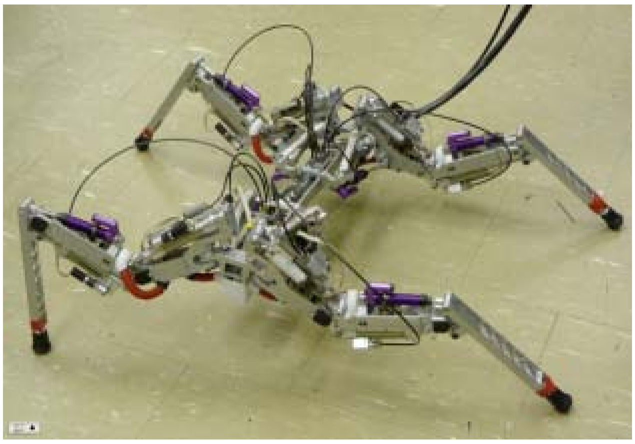

Figure 4: Physical robot in development at Vanderbilt University

FLUID POWERED ACTUATOR MODEL

To achieve 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. Since the purpose of the actuator simulation was to imitate the most basic actuator dynamics, a simple scenario was modeled, consisting of a light mass at the end of the rod. The model was validated against open-loop and closed-loop comparisons with the actual hardware. This basic model was later integrated into simulation to include the true system (leg) dynamics in the actuation model.

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.

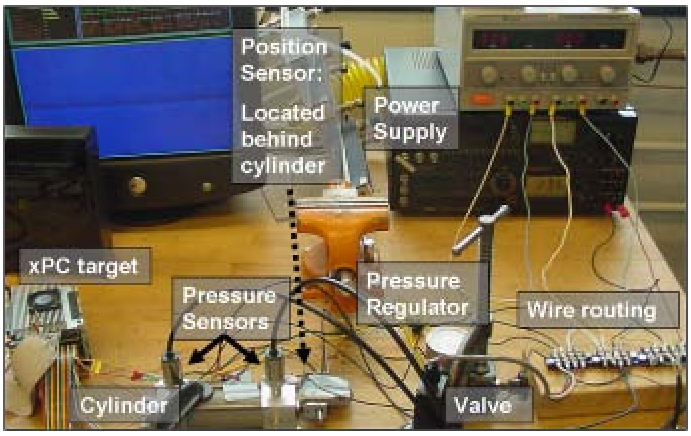

Figure 5: Physical hardware experimental setup

Open Loop Model

Modeling the cylinder is achieved by inspecting each side of the cylinder independently and coupling the two sides into a single dynamics equation. A control volume is drawn about 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 [7]. The flow rate through each side of the valve is independently calculated based on Equation (2), 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.

$$\dot{m} = C_d A_0 f\left(P_u, T_u, {P_d}/{P_u} \right) \qquad (1)$$

Critical Pressure Ratio for air P_d / P_u =.528

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

$$f\left(P_u, T_u, {P_u}/{P_d} \right) = C_1 \frac{P_u}{\sqrt{T_u}} \left(\frac{P_d}{P_u} \right)^{1/k} \sqrt{1 - \left(\frac{P_d}{P_u} \right)^{(k-1)/k}} \qquad (1a)$$

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

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

$$f\left(P_u, T_u, {P_u}/{P_d} \right) = C_2 \frac{P_u}{\sqrt{T_u}} \qquad (2b)$$

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

An energy balance, shown in Eq. (3), assumes the compressed gas obeys the ideal gas law and that the system is adiabatic—there is negligible heat transfer between the cylinder chambers and external atmosphere. This adiabatic assumption is generally acceptable for fast acting systems such as a walking robot.

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

The dynamics of the cylinder are represented by Eq. (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.

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

This output force can be converted to a joint torque or resulting joint angle based on instantaneous part geometry and mass. Following implementation and tuning of the model in SIMULINK, an actual valve-cylinder testbed (Figure 5), complete with potentiometer and pressure sensors, was set up to compare the behavior of the simulated and physical models.

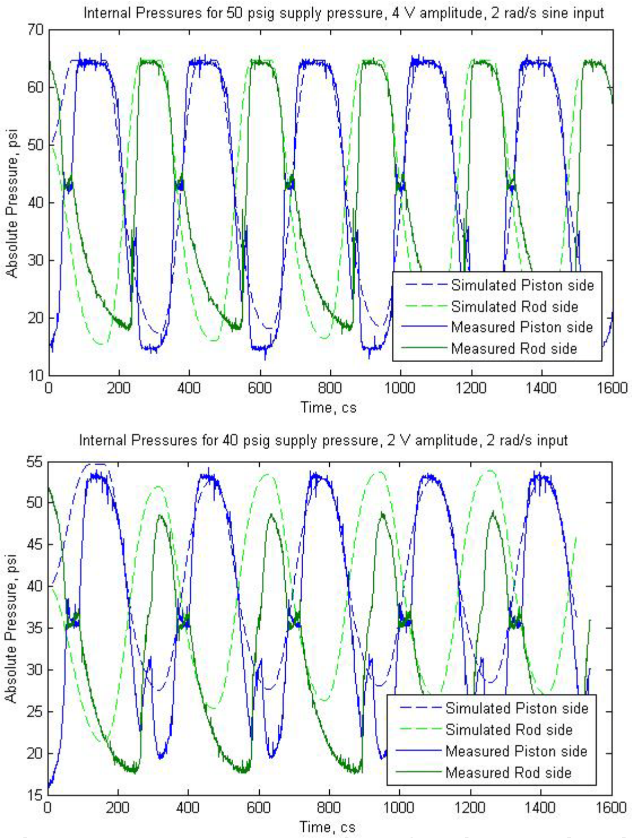

Figure 6: Pressure comparison for simulated and physical open loop actuator

Because the open loop behavior resulted from the cylinder's rapid response to practically instantaneous extensions or retractions of the stroke length, it was found preferable to compare internal pressures of the two models. This was done in several cases, as shown in Figure 6. As can be seen, the measured pressures closely match the behavior of the simulated pressures, with magnitudes achieving reasonable similarity as well.

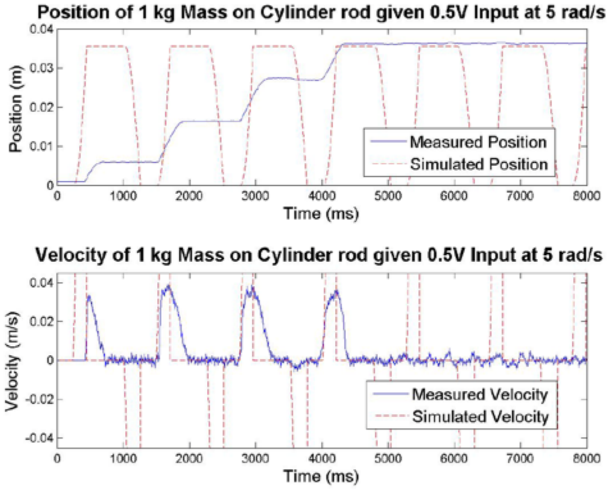

Figure 7: A comparison of measured and simulated system behaviour for a low input voltage shows evidence of stiction in physical model

One likely reason for a difference in behavior is due to the fact that static friction, or stiction, was ignored in the open-loop model. This decision was based on the need for optimum simulation performance. The effects of stiction are most apparent in a system with a low input voltage (Figure 7). Techniques such as dither are commonly used to greatly reduce the effects of stiction when the final feedback control is implemented. Closed loop behavior of the physical system compared to the simulation supports that conclusion for this system.

Closed Loop Model

A PID control approach based on approaches that had successfully controlled the two-legged physical model in the past was applied to the model. The control scheme was tuned so that the output would follow a sine wave, representative of a continuously changing leg motion, within 5% error. Tests were performed on the actual hardware that compensated for the effects of stiction by taking the algorithm and replacing the derivative terms with transfer functions that instead sampled over several periods. The results (Figure 8) showed a near identical tracking response, requiring only slight tuning to achieve the exact desired performance.

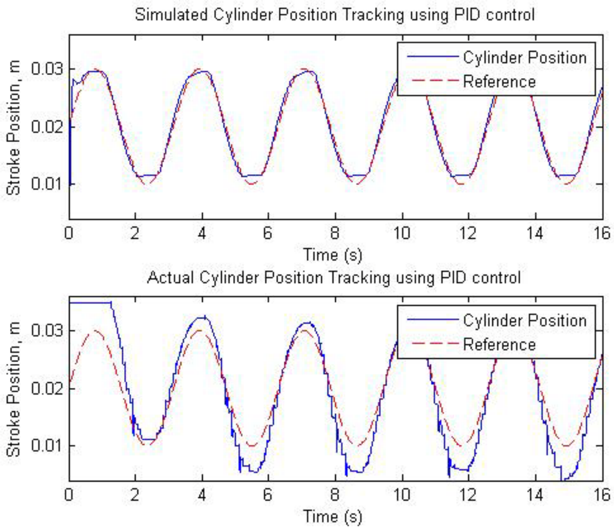

Figure 8: Position tracking of the simulated and physical actuators with PID control

INTEGRATION OF MODEL INTO SIMULATION

The addition of pneumatic actuator models to the simulation provides the key component that allows the user to experience fluid powered rescue robotics. A simple first approach was used to approximate the effects of the dynamics on the actuator by assuming constant mass. A more complete method that included actual joint dynamics – a rod rotating about a joint actuated by a torque from the pneumatic cylinder (Figure 9) – was then analyzed based on simulation performance and constraints so that optimal simulation of fluid-powered actuation could be achieved.

Developed Dynamics

To include leg dynamics in the actuator model behavior, it was necessary to modify the dynamics simulation's configuration, as well as the data transferred between the xPC target and the dynamics simulation (SrLib). The revised setup places the actuator model, without any dynamics component, in the Simulink file on the xPC. The actuator outputs a force, which was converted to a torque based on the instantaneous robot geometry. This torque is sent to the dynamics simulation, which calculated the resulting joint position, velocity, and acceleration, while also providing effects of gravity, joint (stroke) limits, and environmental interaction. These values are sent back to the xPC target, where they are used for feedback to the actuator controller and internal cylinder dynamics. To test the combination of the actuator simulation from the simple mass on rod model with actual system dynamics, only one joint, shown in figure 9, was actuated, and the robot was elevated above the plane such that no environmental interference could occur. Simple sinusoidal and step functions were sent to the

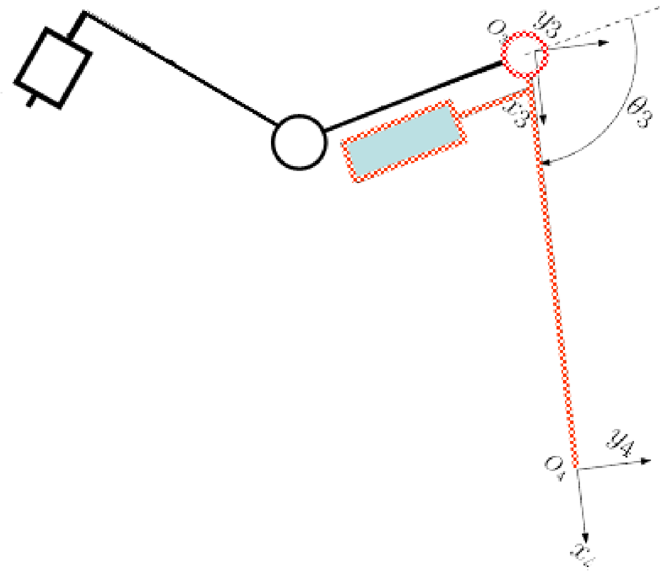

Figure 9: Diagram of cylinder – joint dynamics used for actuator model integrated with simulation

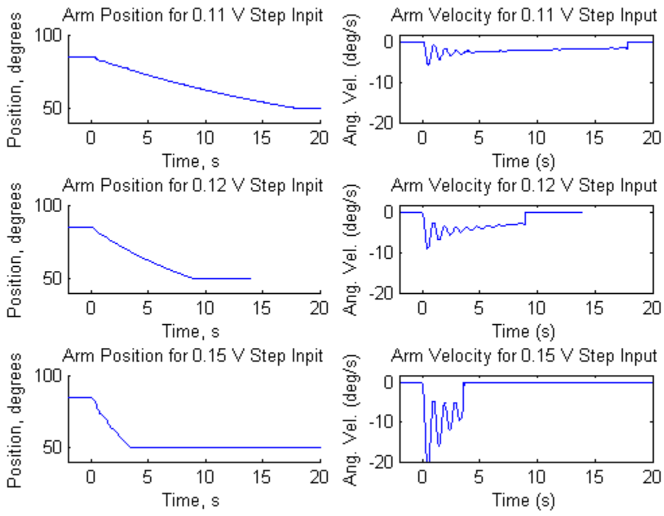

Figure 10: Step responses for open loop model of arm actuated by joint 3

joint to simulate actual motion, while the other joints on the sample leg were held fixed. In the full robot implementation, each of these joints would receive motion commands based on operator placement of the Phantom joysticks.

Before applying a control algorithm to the revised configuration, several open loop responses were found, as shown in figure 10.

As can be seen from the step response, the system is stable, though underdamped and with increasing settling time as the input voltage is increased. The minimal voltage required for large actuator responses can be explained by the lack of stiction in the model. Additionally, it can be seen that the velocity response drifts slightly with time. This is most likely due to the effect of gravity on the system dynamics, which will be shown to require more complex control than was needed with the simple mass approach.

This effect can be further observed from the response to a ramp (Figure 11), which demonstrates that the velocity does not maintain a constant upward slope with input voltage, and instead curves gently, again most likely nonlinearities resulting from dynamic effects.

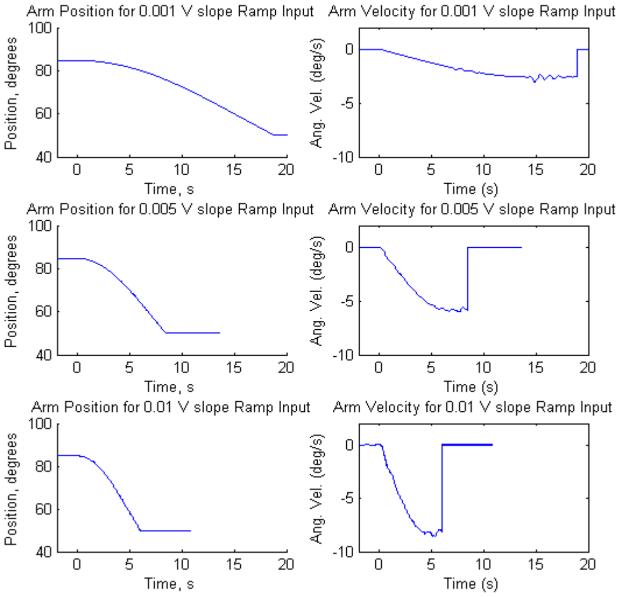

Figure 11: Ramp responses for open loop model of arm actuated by joint 3

Another concern in using SrLib for dynamics feedback was its ability to provide feedback at a constant rate of 1000 Hz, equivalent to the computational speed of the xPC target. A joint torque was sent to a simple joint in SrLib that rotated one member. The packets containing the joint position, velocity, and acceleration were time-stamped and monitored to see when they were received by the xPC target. The output of this test was plotted and can be seen in Figure 12.

Several aspects of this plot raise concern. First, there is a clear drop in the position curve located at approximately 5000 ms. This drop corresponds to toggling a window on the host computer while the simulation is running, thus pointing out the disadvantages of using a non real-time operating system (RTOS) as a platform.

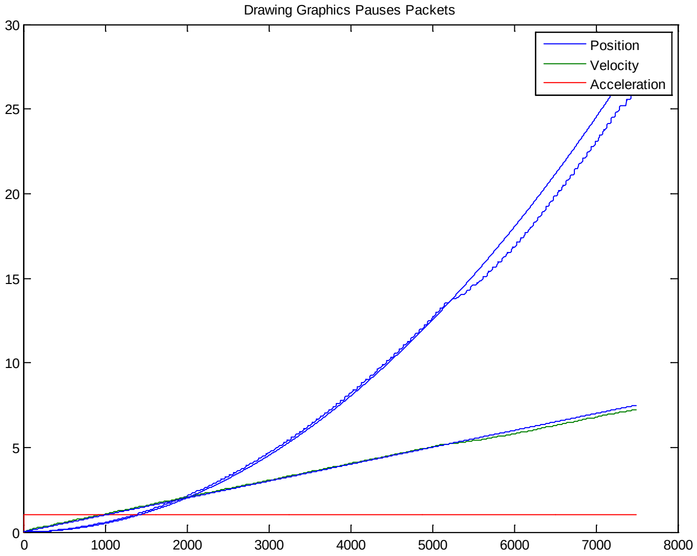

Figure 12: Position, velocity, and acceleration curves from SrLib performance test.

Additionally, the position and velocity plots appear jagged. This effect is due to the fact that every time a frame is redrawn on screen, all UDP packets are paused for approximately 20 ms. Once the frame is drawn, the computer sends all the back-logged packets available in the network buffer. It takes almost 10 ms for the network device to synchronize again. In some cases, the network device may not be able to synchronize as desired, resulting in a gradually increasing lag in system performance. These effects are illustrated in Figure 13.

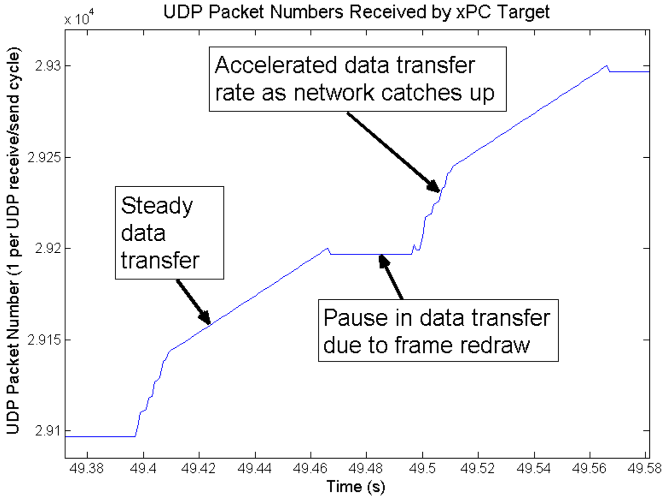

Figure 13: Pausing of packet output due to frame drawing

Since it was apparent that the graphics rendering would have a significant effect on the actuator model, a parameter was set in the dynamics simulation to ensure that frames were only redrawn once per second. While this approach is hardly an ideal goal for the final product, it provides a way to first validate the actuator model with comprehensive dynamics simulation and then adapt SrLib to work with the system.

Using this approach, the closed loop system of an arm actuated by joint 3 (Figure 9) was modeled and control was achieved for a limited input range using a simple PID controller (Figure 14). When the input frequency was changed, however, the controller quickly lost effectiveness. Control was recovered by varying a gain at the output of the control signal. This effect was likely due to the nonlinear actuator dynamics observed earlier; the varying torque and effects of gravity on the actuator require a more advanced controller than a PID algorithm, which effectively assumes linearity about a point.

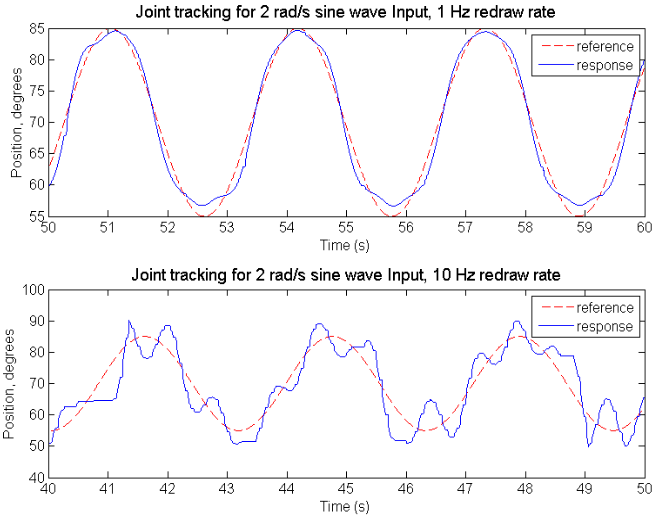

Figure 14: Tracking response to sine wave of 2 rad/s frequency.

Despite the limited effectiveness of the controller, the closed loop signal could be used within its accurate range to view the effects that constant frame updating (10 Hz frame redraw rate instead of 1 Hz used above) would have on the simulation. The result, shown in figure 14, demonstrates the dangers of constantly pausing the simulation. As viewed by the inconsistent response, the controller cannot get accurate dynamics feedback, and is thus unable to provide an appropriate signal. Following achievement of complete system control, then, the redesign of the dynamics simulation to allow constant rendering without interfering with network actions is of utmost priority.

PERFORMANCE OPTIMIZATION

As seen in the SrLib performance analysis, computational architecture can play a key role in the effectiveness of a simulation. To fix the afore-mentioned issues with delays and optimize simulation performance on the dynamics calculation side, several changes are possible. One option would be to migrate the simulation to an RTOS, though this would require major modifications to the SrLib engine, which is built for usage on Windows. An alternative would be to modify the general dynamics simulation architecture. Currently, the system runs sequentially, first building the models, then constructing the parts, running the dynamics simulation/transferring data, and finally rendering frames for the graphical output. A modified version would assign threads of variable priority to SrLib. A thread with high priority would perform network operations and dynamics, while a secondary thread with lower priority performs graphic drawing operations. A third option is to replace the current host machine with one that features a dual core processor. The SrLib dynamics simulation could then be modified to use both processors, running graphics operations on one and dynamics and network operations on the other.

Performance constraints are not limited solely to the dynamics simulation calculation. They also have considerable weight in design of the Simulink file in use on the xPC target. Because the target is running at a fixed rate, its computational ability is limited. Thus, complex algebraic or trigonometric operations, such as the derivation of joint angles from end effector positions, are better performed on a machine with comparably greater processing power. This effect was noted during a simulation in which multiple processes involving the calculation of joint angles and center of mass were performed, resulting in a failure of the target to perform as desired. Shifting the more complex trigonometric angle conversions to the PC attached to the operator workstation resulted in the desired enhanced performance.

FUTURE WORK

The first step in achieving a complete simulation is to obtain dependable control across a spectrum of inputs. This requires a more complex controller that accounts for the effects of complex dynamics across a wide range of motion. A likely requirement will be a better examination of current system behavior, followed by implementation of techniques such as a gain scheduler for differential pressures or feedback linearization.

Next, validation of the closed loop model can be achieved by testing the algorithm, with dither, on an arm of the two-legged robot available at Georgia Tech.

Following validation of the individual joint, actuation will be extended to the other two joints as well. This will likely require further revision of applied control techniques to accommodate the more complicated resulting dynamics.

Finally, the dynamics simulation will be modified, as discussed previously, to achieve a result that is capable of simultaneously providing accurate simulation of the robot and providing graphic output to the user.

CONCLUSIONS

By combining modeling and control of pneumatic actuators with a haptically enabled operator interface, the CRR simulation will relate the effects of fluid powered actuation on a high degree of freedom system to the operator. The simulation models the impact of fluid power actuation on a legged robot and provides a basis for further research related to user control of legged robotics and application of fluid power to the demanding field of search and rescue. An actuator model with simple dynamics demonstrated similarity to a physical model in both the open and closed loop case. A comparison of the closed loop model with the actual hardware also validated a decision to limit computational requirements by excluding stiction and a counteracting control technique, such as dither, from the simulation.

This model was then combined with a dynamics simulation capable of providing more complex dynamics feedback and found to have stable, though nonlinear open-loop behavior. While control was achievable in distinct input ranges, it was apparent that a more complex controller would be required to achieve the versatility in motion control required of the system.

Integration of the modeled actuator into the simulation was shown to require careful consideration of limitations on the performance of the dynamics simulation computer. A future revision will achieve updated rendering capabilities while maintaining accuracy in simulation. It will thereby also compensate for the slowing effect of the delays to ensure that the simulation can be run for long hours without a decrease in performance.

The completion of the CRR simulation would provide an excellent tool for fluid power studies, especially in combination with haptic control. Because of its simulated nature, robot designs can be easily modified and tested with realistic actuation without large monetary or time expenditures for manufacturing. This resulting design flexibility could be used to allow combined modification of the robot, operator interface, and control design to accurately relate the effects of actuation to the operator. This simulation would be a driving point in the development of an intuitive user interface design that maximizes operator understanding and optimally balances user control with system versatility to develop a more capable robotic system capable of saving lives.

ACKNOWLEDGMENTS

The authors would like to thank JD Huggins for his assistance with hardware and software, as well as Seoul National University's Robotics Lab for its development of and assistance with SrLib. Additionally, this research has benefited from collaborators at Vanderbilt University, North Carolina Agriculture and Technological University, and University of Minnesota.

REFERENCES

[1] Messina, E., Jacoff, A., Scholtz, J., Schlenoff, C., Huang, H., Lytle, A. and Blitch, J., 2005, "Statement of Requirements for Urban Search and Rescue Robot Performance Standards," Technical Report, Preliminary Report, National Institute of Standards and Technology.

[2] Schneider, D., "Robin Murphy: Robotiscist to the Rescue," Spectrum, IEEE, 46(2), pp. 36-37.

[3] Driewer, F., Baier, H., and Schilling, K., 2005, "Robot-Human Rescue Teams: A User Requirements Analysis," Advanced Robotics, 19(8), pp. 819-838.

[4] Song, S. and Waldron, K.J., 1988, Machines that Walk: The Adaptive Suspension Vehicle, The MIT Press, New York, NY, Chap. 1.

[5] Lee, S. and Kim, G.J., 2008, "Effects of Haptic Feedback, Stereoscopy, and Image Resolution on Performance and Presence in Remote Navigation," Int. J. Human-Computer Studies, 66, pp.701-717.

[6] Gentry, S., Wall, S., Oakley, I., Murray-Smith, R., 2003, "Got Rhythm? Haptic-Only Lead and Follow Dancing," Proc. Eurohaptics Conference, Dublin, pp. 481-488.

[7] Al-Dakkan, K.A., Barth, E.J., and Goldfarb, M., 2006, "Dynamic Constraint-Based Energy Saving Control of Pneumatic Servo Systems," Transactions of the ASME Journal of Dynamic Measurement and Control, 128(3), pp. 655-662.