System Design of a Quadrupedal Galloping Machine

J. Gordon Nichol, Luther R. Palmer III, Surya P.N. Singh, David E. Orin and Kenneth J. Waldron

***Abstract*—Ohio State and Stanford Universities are cooperating to construct and artificial quadruped. This quadruped has a rest leg length of 0.68m and weighs 70kg. The body is made of four single-leg modules. Each module houses an articulated 3-dof leg. The articulated leg structure uses a set of mechanical springs to store energy. The stiffness of the leg, from hip to foot, is highly nonlinear. The leg initially stiffens as the leg is compressed the first 1/3 of its range, then remains roughly constant for the remainder. This stiffening allows the leg to maintain a relatively constant working length. A cable is used to flex the knee of the quadruped. As the cable is pulled, the knee bends, tensioning the energy storage springs. This cable makes about a ¼ turn over a pulley which is concentric with the hip axis, ½ turn around a pulley at the ankle and then ¾ turn back around a second hip pulley in a direction opposite the first hip pulley. This arrangement decouples cable tension and hip torque. The non-anchored end of the cable is tensioned by a capstan. This capstan uses unique geometry to decrease holding torque and to release the cable instantly. The current top speed of this leg in a 2-dimensional test is 4.15m/s. This leg has been controlled using two control systems, both implemented on an embedded controller attached to the leg. An articulated leg presents control challenges not seen when trying to control a prismatic leg. The first controller tested is a heuristic algorithm whose parameters are updated in real time by the Levenberg-Marquardt learning method. This controller is similar to the controllers used by Raibert[24], with modifications to allow for leg asymmetry. The second controller tested is a direct adaptive fuzzy controller. The fuzzy controller consists of a rule base, inference mechanism, fuzzification interface, and defuzzification interface. Fuzzification starts by mapping an input (body velocity, desired change in velocity, height) into one or more membership functions. The inference mechanism then determines the applicability of each rule to the current inputs. Defuzzification combines the recommendations of each rule in an output based upon rule certainties. The adaptation mechanism modifies the rule output centers to correct velocity errors. The adaptation mechanism gains are experimentally tuned. Once tuned, the controller quickly adapts to leg changes. Both control systems were tested with and without adaptation. Both systems more accurately tracked desired velocity with adaptation. Accurate, high resolution, high speed body attitude sensing is essential for successful quadruped operation. No existing solutions meet the project requirements. An adequate sensor system is being developed in cooperation with a commercial vendor. Initial results show that accurate position tracking is possible with currently available MEMS inertial sensors.**

I. INTRODUCTION

IN a cooperative effort to better understand the quadruped gallop gait, Ohio State and Stanford Universities are constructing an artificial quadruped. Construction of the quadruped has been preceded by several single-leg and full-quadruped simulations [1][2] and three single-leg mechanical prototypes. It is planned that this project will produce the first autonomous artificial galloping quadruped.

Galloping presents a different challenge than that of other dynamic gaits. In contrast to the trot, bound and pace [3], which are paired foot gaits that have some form of symmetry, the gallop is a highly asymmetric single foot gait [4].

The Stanford group, under the direction of Dr. Kenneth Waldron, has responsibility for continuing the mechanical design work begun at Ohio State University (OSU), and designing a suitable sensor system. The OSU group, under Dr. David Orin, is working to develop a control system for this quadruped.

II. MECHANICAL DESIGN

The quadruped body and legs must be designed to meet several criteria. A first reaction to designing such a quadruped would lead the designer to mimic, as closely as possible, biological quadrupeds. Although extensive data on biological quadrupeds is available [5][6], a purely biomimetic approach would impose such stringent requirements on actuator and structure design as to be impracticable. Instead, the authors have attempted to identify characteristics of biological quadrupeds essential to the gallop. The design we have developed incorporates these characteristics.

A. Quadruped

In designing a quadruped, one of the first design decisions is scale [7]. As a machine increases in size with respect to some linear dimension, area, which is related to actuator authority for most actuation technologies, increases as the square of the length, while volume, which affects mass and inertia, increases as the cube of the length. In other words, as scale increases, actuator authority decreases relative to mass. On the other hand, reducing scale tends to result in poor power system efficiency, and runs against the lower limits of available component sizes.

A complete discussion of the design decisions made in designing the OSU-Stanford Quadruped (OSQ) is beyond the scope of this paper. Tradeoffs lead us to design a quadruped

J. G. Nichol, S. P. N. Singh, K. J. Waldron are with Stanford University. 424 Panama Mall, Bldg 560, Stanford, CA 94305 USA (email-jgnichol@stanford.edu)

L. R. Palmer and D. E. Orin are with the Ohio State University

which is roughly the size of a large goat or dog. Energy storage and actuation are electrical. The total mass budget for the quadruped is 70 kg. At rest, leg length from foot to hip is 0.68 m.

The quadruped, seen in Fig. 1 in a partially-assembled state, has four legs. Each leg is mounted in a module which adds an abduction-adduction axis, bringing the total to three axes per leg. The modules can be connected in a variety of configurations. This will allow future tests of leg location, orientation, and body flexion. The top surface of each module serves as a mounting platform for motor amplifiers, power supplies and control computers. This platform permits easy assembly, diagnosis, repair and reconfiguration of the electrical subsystem.

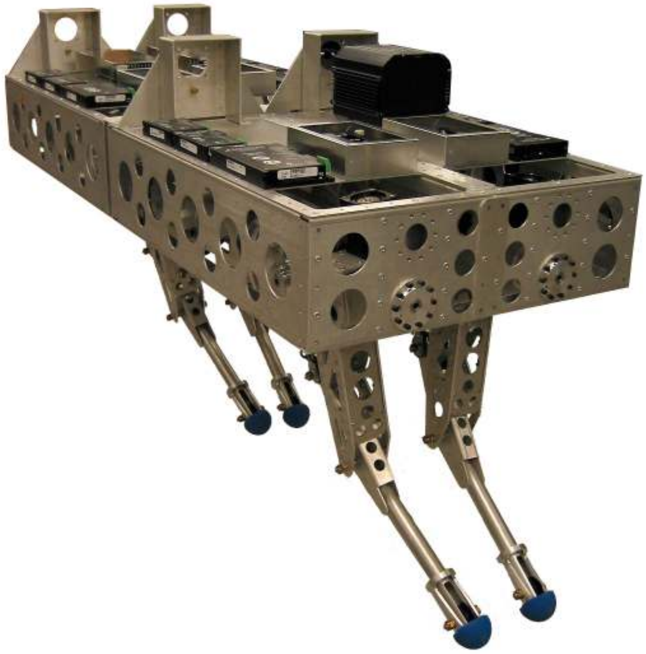

Figure 1 OSU-Stanford Quadruped (OSQ)

B. Construction Methods

A galloping machine must be able to withstand the constant foot-ground-impact loads imposed by galloping. It must also be robust to dynamic mishaps. The selection of materials and construction methods becomes a very important early step in the design process. After considering many options, we chose methods and materials employed by the light aircraft industry in constructing stressed-skin aluminum airframes. Using these methods [8][9][10], sheet aluminum can be cut, bent and riveted into structures.

These methods were chosen for several reasons. First, they are capable of creating structures of high stiffness and strength relative to weight. Because these structures are fabricated from cold-rolled or strain-hardened sheet aluminum, they have a higher stiffness than a machined part of the same geometry. Additionally, thin sections can be easily fabricated. These thin sections can be located far from a neutral axis to provide high strength for a given amount of material.

Second, structures correctly designed with these methods are highly vibration resistant. Riveted joints are assembled by plastically deforming the fasteners. This plastic deformation applies a pre-load to the parts being joined. A correctly designed riveted joint will not loosen under vibration. Additionally, rivets are often made of aluminum; threaded fasteners are usually made of steel. Rivets provide a weight savings when used as a substitute.

Third, parts which have been punched or laser-cut and subsequently folded on computer controlled equipment can be assembled with little loss in accuracy. Assembly is relatively simple, requiring only the tools necessary for riveting. Correctly designed parts are self-fixturing.

Fourth, these methods are adapted to producing small quantities of parts economically. Materials and fabrication are relatively inexpensive. Assembly is rapid and requires no tooling. Rivet removal is nearly as fast as removing a threaded fastener.

C. Single Leg

Once overall size was established, the quadruped was designed literally from the ground up. James Schmiedeler designed the OSU DASH (Dynamic Articulated Structure for High performance) leg [11][12], based on work done by Brown and Zeglin [13] and others [14][15][16]. The Stanford DASH leg, the second prototype, was developed at Stanford University (Fig. 3) [17]. The final leg, a slightly modified version of the Stanford DASH leg, also developed at Stanford University, can be seen installed in the body in Fig. 1.

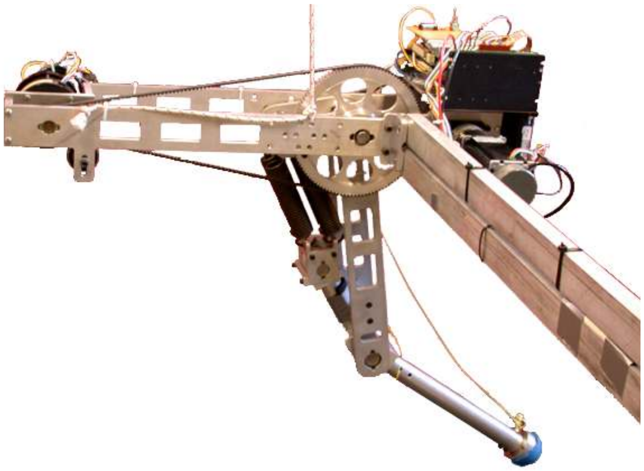

Figure 2 The OSU Dash leg, attached to test boom

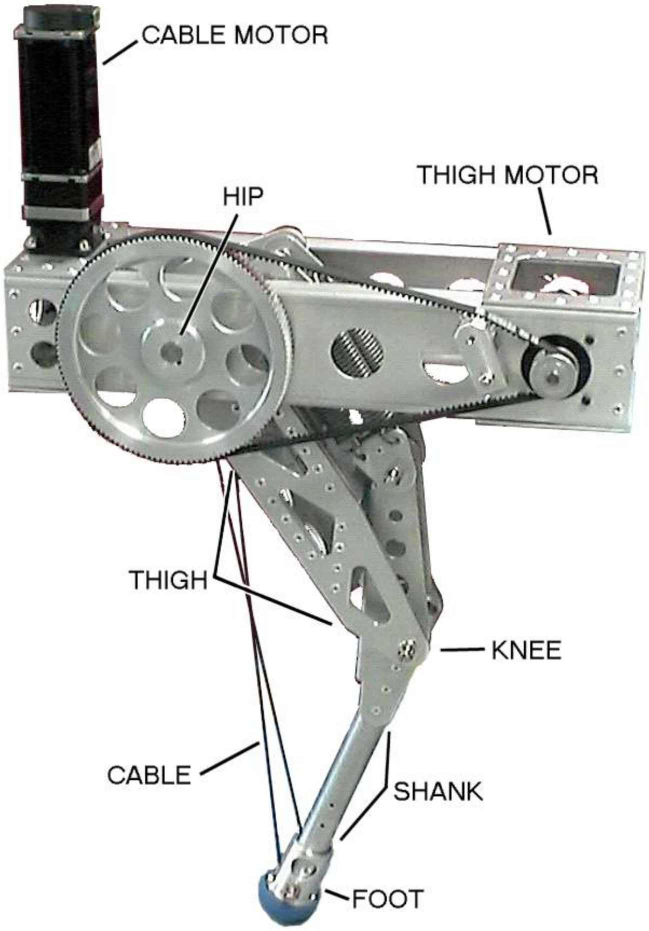

Figure 3 The Stanford Dash leg with nomenclature

One large divergence from biomimesis is the choice, made early in the project, to design only one leg to serve as both fore and hind leg. In natural quadrupeds, the roles of the fore and hind legs are quite different [5]. In spite of this difference, a single leg has been designed which will fill both roles. Using a single leg reduces the number of different parts in the final machine, and reduces single leg testing by half.

The names chosen to identify parts of the single leg (Fig. 3) were drawn from animal anatomy and represent a mix of fore and hind leg nomenclature. This mix is appropriate, as the same leg serves for both.

With each leg prototype, features were incorporated and refined. Three of the most important design features found in the Stanford DASH leg are: nonlinear effective stiffness, the over-under cable linkage and the quick-release capstan.

1) *Nonlinear Stiffness*

The functional leg geometry has nonlinear virtual stiffness. The leg mechanism (Fig. 4, left) can be modeled as a prismatic leg (Fig. 4, right) with nonlinear stiffness (Fig. 5). As the leg is compressed, the leg stiffness initially increases sharply until approximately 47 mm (15.9 kN/m), and then remains constant within 5%. This variable stiffness allows us to keep the leg operating length relatively short.

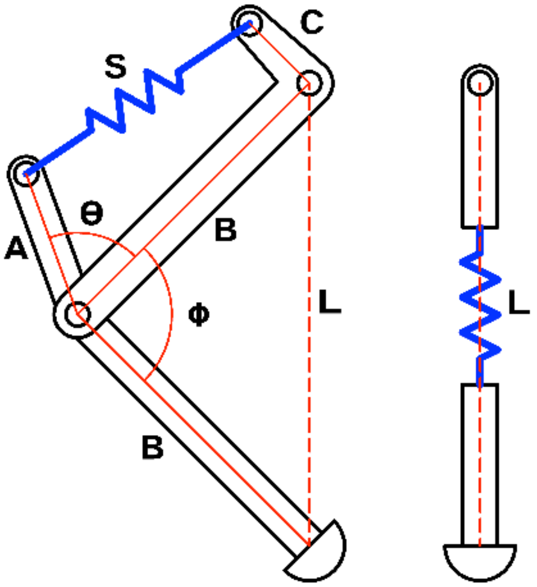

Figure 4 Dash leg (left) and comparable virtual leg (right)

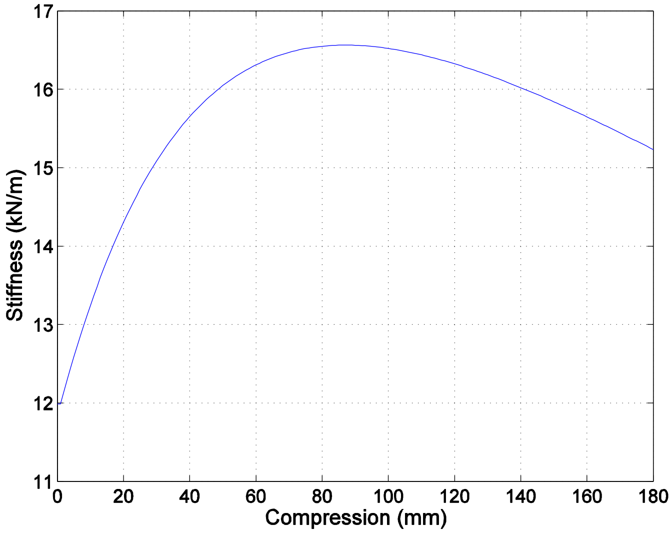

Figure 5 Virtual leg compression vs. stiffness (after Schmiedeler [9])

2) *Over-Under Cable Linkage*

In order to avoid coupling the thigh and cable axes, a cable linkage was developed which separates the two axes by applying equal and opposite moments about the hip (fig. 6). One end of the cable is attached to a fixed anchor. The cable passes under, then around the first of two pulleys at the hip. The cable is then looped around the ankle pulley, and over the second hip pulley. Finally the cable is attached to the capstan.

This cable drive gives a 2:1 force reduction with relatively little loss. Cable tension is half as large as the virtual spring force.

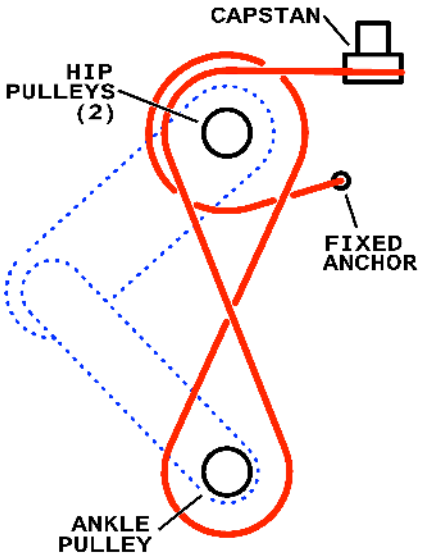

Figure 6 Cable Linkage

We are currently using ultra-high-molecular-weight polyethylene (UHMWPE) (Spectra® 2000) cable. This cable is constructed with a unidirectional UHMWPE core covered by tightly braided jacket of the same material. The line is of the type used with kite boards. It has an ultimate tensile strength of 2.7 kN and a diameter of 1.8 mm.

Spectra 2000 has a very high stiffness (113 GPa) and high strength (3.25 GPa) [18]. This is somewhat better than typical UHMWPE which has a tensile strength of about 3 GPa and a stiffness of approximately 90 GPa [19 pp. 225]. UHMWPE is comparable to aramid fiber in strength and stiffness. Aramid fibers typically have an ultimate tensile strength of approximately 3 GPa and a stiffness of about 100 GPa [19 pp. 308-309]. UHMWPE is comparable in strength to carbon fiber, but is inferior in stiffness. Carbon fiber has a stiffness of approximately 300 GPa, and a tensile strength of around 3 GPa [19 pp. 170]. Additionally, UHMWPE does not self abrade, making it the best choice for this cable drive.

Because UHMWPE is very lubricious, terminating either end of the cable is nontrivial. Experience has shown that the "Anglers Loop" [20, knot number 1017] works well for affixing the static end of the cable to the anchor, and for attaching to the swivel (see next section). The "Figure-Eight," [20, knot number 520] backed by an overhand knot [18, knot number 514] holds the cable in the capstan anchor hole. Prior to tying a knot, each end of the cable is singed by holding it near a flame. The knots are then tied near the singed end. Failure at the knots is relatively rare. Some cable elongation occurs as the knots tighten during the first few seconds of operation. No additional elongation occurs until failure. A cable will typically last 2000 cycles.

3) *Cable Drive Capstan*

Initial attempts to employ an electromagnetic clutch to release the cable showed that the majority of clutches cause large energy losses and are quite slow. Electromagnetic brakes suffer from similar disadvantages. Initially, the cable actuator was held in a stall after cable windup, then reversed as ground contact was detected. This method was effective, however, the tradeoff between winding torque, reversal speed and stall current lead to a very inefficient actuator. The quick-release capstan drive was designed to overcome this problem.

The cable drive capstan attaches to the end of the cable actuator (Fig. 7). The hole through the center of the capstan, with the rectangular groove, fits over the output shaft of the actuator. The holes surrounding, and parallel to, the grooved hole reduce cam weight and allow access to the back of the anchor hole. The cylindrical surface, small-radius surface, flat surface and oblique flat surface (release bevel) on the outside of the cam are the four functional surfaces.

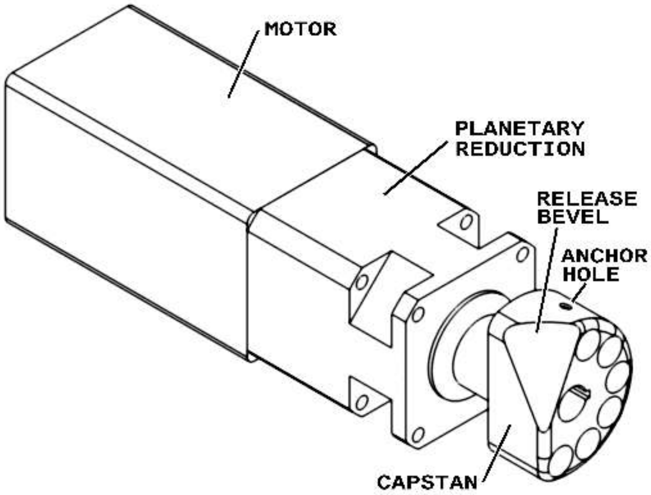

Figure 7 Cable Actuator with capstan

In operation, the capstan rotates counter-clockwise as seen in the figure, so that the cable winds behind the capstan (as pictured). At rest, the cable extends radially from the anchor hole. As the capstan begins to rotate, cable is drawn from the leg, causing the leg to shorten, and the springs to store energy. After the first 90°of capstan rotation the cable is tangent to the cylindrical surface of the cam. Another 180° and the cable has wrapped completely around the large cylindrical surface and begins to wrap around the small-radius surface which joins the cylindrical face with the flat face. Another 70° brings the cable parallel to the flat face of the capstan. In this position, the cable is relatively close to the center of rotation, decreasing holding torque. When the foot contacts the ground, the capstan rotates an additional 20-40°, forcing the cable down the release bevel and off the cam.

The capstan adds one full twist to the cable at each release. This twisting, if left unchecked, would result in premature cable failure. A swivel was added between the hip and ankle pulleys near the capstan. This swivel prevents twist accumulation.

One of the significant trade-offs in designing such a capstan is the compromise between holding torque and controllability. A small holding radius (holding radius being the length of the common normal between the cable centerline and the cable actuator output shaft centerline) requires a small torque to hold a given cable tension. A large holding radius requires a larger torque to hold the same cable tension. The current required to maintain this torque is wasted, as it does nothing to add energy to each stride. Conversely, a small holding radius makes the cable length relatively insensitive to capstan rotation. A large holding radius makes the cable more sensitive to capstan rotation. The capstan is currently designed to favor efficiency.

The cable actuator was chosen so that it can rotate only as far as necessary to release the cable during the short (50-80 msec) ground contact. This means, as stated above, that the cable actuator cannot take in significantly less than the capstan design cable length of 15 cm. Strategies to control energy addition are being investigated.

Fig. 8 shows the amount of energy which can be stored in the leg as a function of length L. Maximum energy storage (290J) is set by strain limits on the coil springs being used.

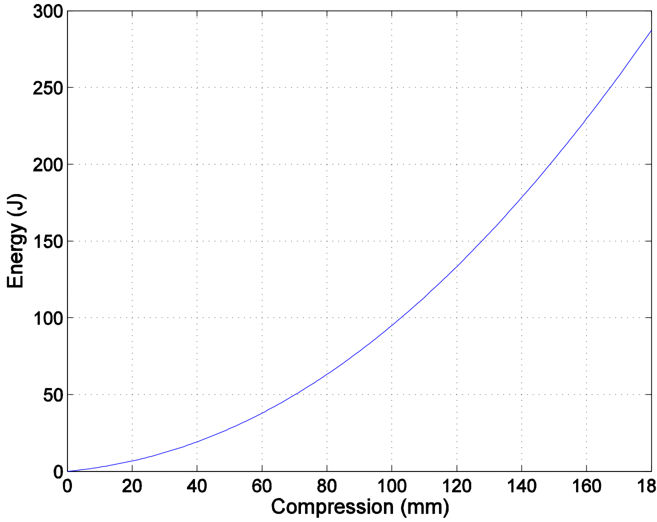

Figure 8 Leg spring energy storage as a function of leg compression

The cable lengthens slightly during the first few cycles after it is installed. This lengthening comes from the tightening of the knots which attach the cable. Because the capstan takes in the same amount of cable each cycle, slack cable length has a large effect on total energy addition. Fig. 9 shows that, as the length of slack cable increases, the energy which can be stored in the springs decreases. The total energy which can be stored in the springs is most sensitive to changes in cable length when the cable has zero slack. One strategy for controlling energy addition is to control cable slack.

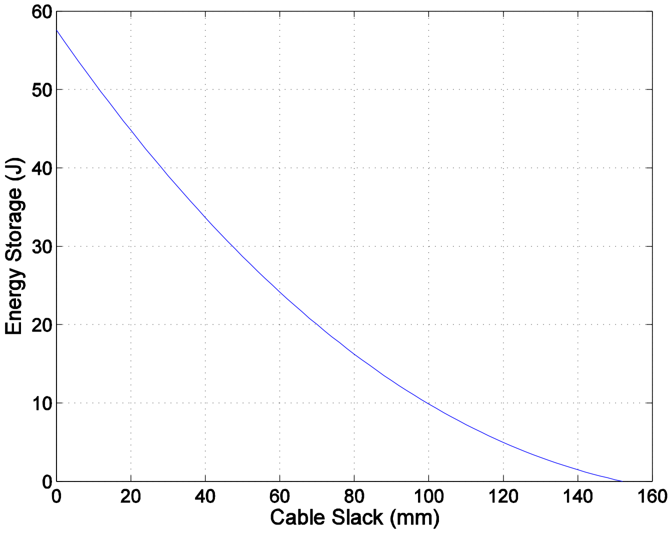

Figure 9 Cable slack length and its effect on the per-stride energy addition

D. Results

Single leg testing was accomplished differently at Stanford and OSU. The Stanford DASH leg and its successor were tested as they ran in a circle. The OSU DASH leg was tested on a treadmill. Both legs were constrained by test booms to be quasi-planar. All leg tests were accomplished by running the leg knee-backward, measuring forward velocity to the left in Fig. 3. The quadruped may also use the knee-forward configuration, but the results of only one knee direction are presented here.

Current top speed of a single leg is 4.15 m/s. Different gaits are more efficient at different speeds. According to Hoyt and Taylor, [21] horses transition between trot and gallop at about 4.5 m/s. Using the dynamic similarity hypothesis of Alexander and Jayes [22], and an estimate (Alexander and Jayes include mass ranges for their test subjects, but no measurements) of the relative size of the horse and the OSQ, 4.15 m/s is slightly above the OSQ trot-gallop transition speed. Another result, from Heglund and Taylor [23] predicts a trot-gallop transition speed of 3.9 m/s. The current single leg top speed of 4.15 m/s will allow the quadruped to gallop efficiently.

III. CONTROL SYSTEM

Raibert [24] showed that the airborne trajectory of a single-leg machine is determined by the horizontal placement of the foot relative to the hip and the energy in the springs at touchdown. The thigh angle and length of cable drawn in before ground contact dictate these two parameters on the prototype leg, so cable draw and hip angle are the two outputs of each controller developed. Achieving good performance here is more difficult, though, because of the articulated leg structure as opposed to the prismatic structure used by Raibert.

The leg cycle begins when the controller is called at the top of flight (TOF), the highest vertical point in the leg's ballistic flight trajectory. The controller computes new outputs based upon present and desired body states, and experimentally tuned PID controllers drive the joints of the leg to the desired positions. The joint controllers continue actuation until the foot detects ground contact. At this point, the actuator at the hip is turned off to allow the thigh to undergo its natural response and the cable motor is driven forward to a position allowing the cable to slip off of the specially designed cam. When the foot no longer detects ground contact, the joint actuators use the previous TOF outputs as setpoints and wait for new inputs after the next top of flight.

A. *Control System Design*

Legged machines are quickly growing in complexity to accomplish more demanding tasks, which in this case is galloping at high speeds. Accurately modeling these complex machines operating with asymmetric footfalls in dynamic trajectories becomes difficult, and control techniques based on the system model can yield poor results. Intelligent controllers do not require system identification and can incorporate user heuristics to successfully control this type of system. Intelligent controllers can require more computational power, which has previously limited their use in real time. In this work, two intelligent methods are implemented in real time on the prototype leg and compared with each other and against a more traditional control algorithm. The first controller tested is a heuristic algorithm whose parameters are updated in real time by the Levenberg-Marquardt (LM) learning method, and the second is a direct adaptive fuzzy controller.

1) *Heuristic Control with LM Learning*

Raibert's original controller computed the forward touchdown position of the foot by

$$x_f = \frac{vT_s}{2} + k_v\left(v - v_d\right), \qquad (1)$$

where T_s is the time of the previous stance period, v is the TOF body velocity, and v_d is the desired TOF velocity. The gain on velocity error, k_v, is tuned experimentally for desired performance. The first term of this equation estimates the foot placement required for running at constant speed and the second term corrects velocity errors. This equation is the result of some dynamic modeling done by Raibert. By observation, stance time does not vary much so $T_s /2$ can be included in one coefficient for v.

The nonsymmetrical leg requires an offset term added by Marhefka [25] to maintain zero velocity. The new heuristic control equation, with a change in coefficient names, is

$$x_f = \alpha_1 v + \alpha_2\left(v - v_d\right) + \alpha_3 . \qquad (2)$$

The offset term α_3 is experimentally tuned. The Levenberg-Marquardt online learning method is used to train the remaining two parameters, α_1 and α_2.

The error signal, ε, to be minimized is:

$$\varepsilon = y - F(\mathbf{p}, \boldsymbol{\alpha}), \qquad (3)$$

where y is the unknown best forward foot touchdown position for the present and desired system states, $\mathbf{p}$. The function $F(\mathbf{p}, \boldsymbol{\alpha})$ represents Eq. 2 as the output of the heuristic controller dependent on α_1 and α_2. The LM algorithm is a derivative of the Gauss-Newton learning method used to solve least squares problems [26]. The derivation of this algorithm for a one-leg machine is outlined in [27]. The resulting update formula for our system is then:

$$\alpha_{j+1,m} = \alpha_{j,m} + \frac{p_m}{p_m^2 + \lambda_m} e_j \qquad m = 1,2 \qquad (4)$$

where

e_j = system error,
λ_m = step size control variables,
$p_1 = v$, and
$p_2 = (v - v_d)$.

λ_1 and λ_2 are components of the LM algorithm that do not exist in the Gauss-Newton method. These added parameters correct the ill-posed case of very small state parameters, $\mathbf{p}$, and are also used to control the adaptation step size. The error, ε, in Eq. 3 is not available for computing updates. Because of their monotonic relationship, the system error, $e = v_d - v$, is used instead of ε with good results. This is verified later in the results. The update is computed immediately before the controller is called at the beginning of the next cycle. The new coefficients, $\boldsymbol{\alpha}_{j+1}$, are then used in Eq. 2 to compute the setpoints for the following touchdown.

2) *Fuzzy Control with Direct Adaptive Learning*

The fuzzy controller consists of a rule base, inference mechanism, fuzzification interface, and defuzzification interface. Figure 10 is a block diagram of the control process with an adaptation mechanism which will be described later. The fuzzy control process starts with fuzzification by mapping an input into one or more membership functions. Example triangular input membership functions used to characterize the desired change in body velocity, Δv_d, are shown in Fig. 11. One membership function is centered at 0.0 m/s and will have a certainty, $\mu_{0.0}^{\Delta v_d}$, of 1.0 if the input lies at the center. If the desired change in body velocity is 0.125 m/s, then $\mu_{0.0}^{\Delta v_d} = \mu_{0.25}^{\Delta v_d} = 0.5$, and all other membership functions for that input become zero. The membership functions at both ends are saturated to include the entire range of input values. Using triangular membership functions without center overlap limits the number of nonzero membership certainties, for each input, to two. This reduces the computational complexity of the fuzzy controller.

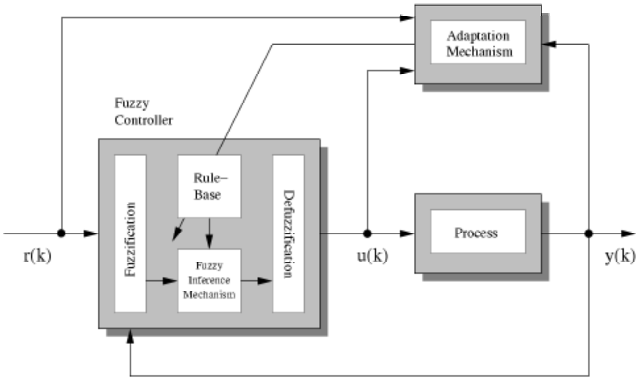

Figure 10. Structure of the direct adaptive fuzzy control system. (Marhefka and Orin 2000)

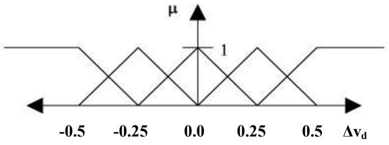

Figure 11. Example membership functions for the change in desired leg velocity.

The actual membership function centers used in the control system are shown in Table 1. The body velocity, *v*, the desired change in velocity, Δ*v_d*, and height, *h*, make up the three inputs to the controller.

Table 1. Fuzzy controller input membership functions

Input	Membership Function Centers	Units
v	-1, 0, 1, 2, 3, 4, 5	m/s
Δv_d	-0.5, -0.25, 0, 0.25, 0.5	m/s
h	70, 75, 80, 85, 90	cm

The fuzzy rule base is a table of controller outputs for every combination of input membership functions. The number of rules is then equal to the product of the number of membership functions for each input. For our controller there are 7x5x5 = 175 rules, each with a corresponding output for touchdown thigh angle. For the results given in the paper, the cable length was not varied so energy added to the system each cycle remained constant.

The inference mechanism is the next step in the fuzzy controller. This mechanism determines the applicability of each rule to the current inputs. The product is used to determine the certainty, μ_i, that the premise of rule *i* is currently applicable. The certainty of rule *i* whose premise is:

IF VELOCITY IS `0.0 M/S' AND DESIRED CHANGE IN VELOCITY IS `0.25 M/S' AND HEIGHT IS `75 CM'

would be:

$$\mu_i = \mu^{v}_{0.0} \times \mu^{\Delta v_d}_{0.25} \times \mu^{h}_{75}. \qquad (5)$$

As mentioned earlier, each input has a maximum of two nonzero membership certainties so the number of rules with nonzero premise certainties becomes 2^n, where n is the number of inputs. Adding membership functions to an input will not affect the amount of computation because only two membership functions are on in each input. Adding inputs, however, will significantly increase computation.

The last element of fuzzy control is defuzzification. This process combines the recommendations of each rule in an output based upon rule certainties. Center average defuzzification is used and the output *y* is given by

$$y = \frac{\sum_i c_i \mu_i}{\sum_i \mu_i}, \qquad (6)$$

where μ_i is the premise certainty of rule *i*, and c_i is the output of rule *i*. This equation shows a summation over all rules. Each rule output center is multiplied by its certainty, which weights the controller output toward the rule most applicable. This controller uses three inputs, meaning that only the eight rules with nonzero premise certainties need to be included in Eq. 6.

3) Adaptation mechanism

The adaptation mechanism modifies the rule output centers to correct velocity errors. Immediately before the controller is called, the current system state is compared to the desired state calculated at the previous TOF. The output for rule *i*, c_i,is updated as a factor of this error by

$$c_{i,j+1} = c_{i,j} + K_c \mu_{i,j} e_j, \qquad (7)$$

where

K_c = adaptation gain,
$\mu_{i,j}$ = certainty of rule *i* at the jth cycle, and
$e_j = v_d - v$ = system error of cycle *j*.

K_c is tuned experimentally. Note that the certainty of rule *i* is used to scale its update. This applies more change to the rule outputs that were more applicable. Again, this premise certainty is nonzero for only 2^n rules meaning that only the rules that applied to the previous controller output are updated by the present error. In this adaptation mechanism, the foot is placed further forward if the velocity was too high. This method is a direct result of the user's heuristic knowledge of the system.

The direct adaptive approach is computationally simple enough to run in real time. This method also utilizes heuristics to eliminate the need for difficult system identification and added complexity in the algorithm equations. The controller can also adapt to changes in leg configuration without the need for major restructuring.

4) Results

The real-time control is implemented on a Kameleon board from K-Team, which has a Motorola MC68376 microcontroller running at 20 MHz and without hardware support of floating point operations. Body velocity is measured by applying a simple Butterworth filter to the Euler derivative of incremental encoder counts on the boom and vertical axis.

Figure 12 shows the leg response to heuristic control without learning compared to the response of heuristic control with online LM learning. The initial values for α_1 and α_2 were hand tuned for reasonable performance. Without LM learning, the leg slowly responds to changes in the desired velocity, and steady state error is observed. The large velocity errors exhibited with learning turned off are quite similar in size to those reported by Raibert [28]. With learning on, the steady state error is decreased and a faster response to desired velocity changes is noticed.

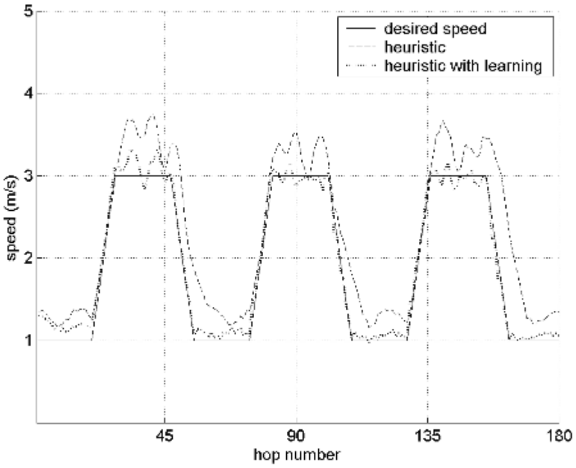

Figure 12. Heuristic control compared with learning on and off

Figure 13 shows the response of the leg using a fuzzy controller with and without direct adaptive learning. The rules shown in Table 1 were used in both cases shown. The fuzzy controller with learning outperforms the fuzzy controller without learning and the heuristic controller with learning. This is expected because the fuzzy controller stores more parameters. The precision control of the fuzzy algorithm is a trade-off to the computational simplicity and small memory footprint of the heuristic controller.

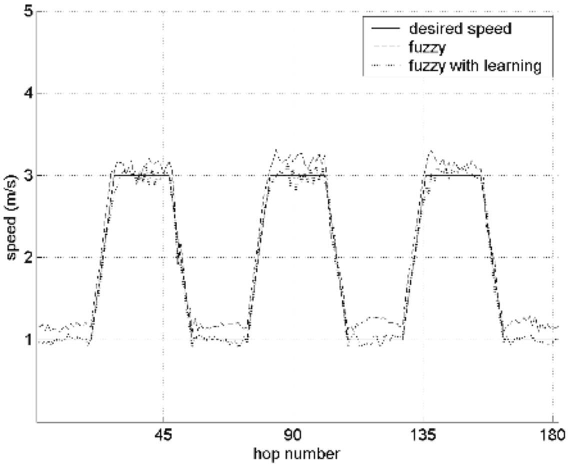

Figure 13. Fuzzy control compared with learning

As described earlier, eight rules have nonzero premise certainties in the case of three inputs. The nonzero certainties are used to calculate a fuzzy output and also update individual rule outputs. In order to test the performance of the leg when the number of inputs and rules are reduced, the height input in Table 1 was eliminated. The new controller has 7x5=35 rules which occupy less space in memory and only four rules have nonzero premise certainties for a given set of inputs. The response of the leg using the simplified fuzzy controller is shown in Fig. 14 with the response using the extended-rule set. With fewer rules, the performance is degraded but may still be acceptable. Although not shown, the body height varied more during tests when height was not used as an input to control the system. The performance of the reduced-rule fuzzy controller closely matches that of the heuristic controller with LM learning although it is expected to do better. During these tests of mainly two speeds, only a limited number of fuzzy rules are activated. The fuzzy controller is expected to show superior performance during a more extensive test covering more speeds, which also degrades the performance of the heuristic controller.

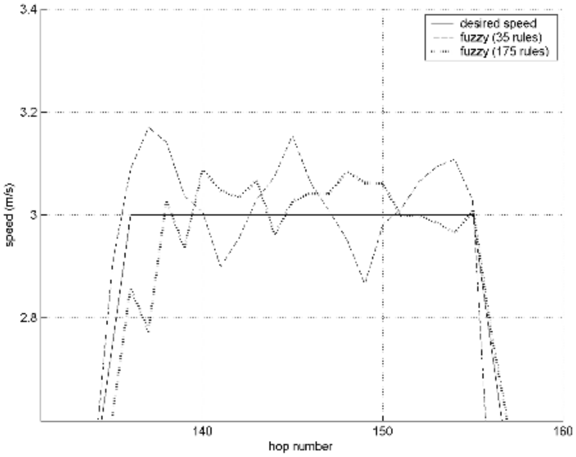

Figure 14. Original fuzzy controller compared with a smaller rule base controller

The Levenberg-Marquardt algorithm significantly improved the performance of the heuristic controller. The update laws for the LM algorithm were derived from the heuristic control equation, which is based on Raibert's original equation for the forward touchdown position of the foot. Although simplified, the heuristic algorithm was based on the true dynamics of a one-leg system. At the time of writing, there was no corresponding set of equations modeling a 3-dimensional quadruped. It is planned to produce an adequate controller without needing one, allowing the intelligent controller to create its own mapping.

The fuzzy controller outperformed the heuristic controller without relying on a model of the system. The direct adaptive fuzzy controller uses heuristic update laws to create its own mapping of input leg configuration to output body state. A reduced-rule base fuzzy controller also performed well on the prototype leg.

The fuzzy controller can be extended for use on the quadruped by applying similar heuristics. Marhefka simulated a planar quadruped with prismatic legs galloping close to 7 m/s using the direct adaptive fuzzy approach [2]. Intelligent controllers and learning algorithms will continue to prove themselves a valuable asset towards the development of a galloping quadruped.

IV. Sensors

In order to propel itself and execute the maneuvers necessary for galloping, the OSU-Stanford Quadruped must accurately perform a complex and involved series of motions. To realize this, the robot uses high-performance actuators and a high-frequency controller. This, in turn, dictates that the on-board instrumentation needs to have a faster response and be robust or, controllability will be lost.

Traditionally, such demanding localization applications use an inertial navigation system (INS). An INS is a self-contained navigation device that operates on the principle of dead reckoning [28]. Typically implemented using a triaxial accelerometer and a gyro triad, data from these sensors are referenced to the body of the vehicle and integrated with an absolute localization source (e.g., global positioning systems or GPS) to provide tracking with respect to an Earth frame. This reset process addresses the major drawback of INS systems, namely, the unbounded divergence of the position estimate due to integration of errors (noise) in the sensors and drift from numerical integration techniques. [29].

Barshan and Durrant-Whyte [30] have shown that with the widespread availability of compact (and low-cost) solid-state inertial sensors, an INS is a promising approach for mobile robot applications. Further, Kalman filters, and other novel algorithms, allow for limited robotic operations without resets from an absolute source [31][32]. This is of importance as common navigation aids (e.g., GPS) can be occluded and may not provide sufficient spatial resolution to fully disambiguate drift accumulated over the relatively short time intervals during which the OSQ is tested.

However, the direct application of these sensors to dynamic locomotion is limited by the scale, nature, and parameters encountered. The locomotion domain allows for alternative correlative solutions including magnetometers, vision systems, and odometery via kinematic joints (proprioception); but it also requires sensors with good sensitivity and dynamic range [29]. This variation is especially evident in vehicles where dynamic forces are dominant and that are characterized by rapid state changes and large acceleration impulses, such as autonomous helicopters and unmanned ground vehicles [33].

1) *Position Sensing*

Robust position and motion sensing with respect to the ground is needed for the control and tracking of OSQ. As discussed in Sections II and III, the criteria for this sensor are stringent as the sensor system must operate such that it provides:

- Robustness to large cyclic shock and vibration loads (withstand >20g shock [power on], and a linear acceleration range of ±10g on all axes).
- High-resolution attitude estimation (<½° and >360°/sec)
- Fast update of final position/velocity estimates (>30 Hz)
- Computational efficiency (for integration with embedded microprocessor)

Further design objectives include self-contained operation, a low noise floor, low mass (i.e., <100g), low power consumption (<5 W), and low cost (<$1500).

At first glance, an inertial navigation system (INS) appears to be the ideal (if not the only) sensing methodology for this robot [30]. The rigid mechanical design of the OSQ improves INS applicability by simplifying the calculations and allowing measurements about any single point to be extrapolated for the body. This provides flexibility in sensor placement as components need not be collocated. Further, recent developments in inertial micromachined sensors have improved the sensitivity and reduced the drift of these devices such that an Inertial Measurement Unit (IMU) can be constructed at significantly reduced cost and size compared to traditional versions [34].

An extensive survey of the commercial market for these sensors did not reveal an equally robust alternative to an IMU for meeting the aforementioned requirements. Furthermore, integrated attitude sensing and INS packages (often employing micromachined sensors) are available from a variety of commercial vendors such as: Crossbow Technologies, InterSense, McKinney Technologies, Microstrain, and Xsense. In general, these units are designed for less demanding applications (e.g., body tracking) and need to be modified for the severe dynamic loads present during full-speed galloping. These units combine core sensors with proprietary software to yield a final sensed value that is communicated to a PC. This process adds significant delay and complicates the integration of these devices [32].

2) *Foot Contact Sensing*

It is important to know when the foot has contacted the ground so as to release the cable and to potentially update the position estimate. This is achieved by placing a Force-Sensing Resistor™ (FSR) (Interlink Electronics) between the shank and the upper section of the foot assembly (see also Fig. 14). To reduce the forces on the FSR a rubber pad is placed above it. On contact with the ground, a moment is generated about the pin holding the foot and ankle pulley. This, in turn, causes the FSR to be compressed changing its resistance significantly. This change is detected by a comparator circuit which sends the final signal to the controller.

3) Sensor System Design

The OSU-Stanford Quadruped is equipped with both basic and integrated sensors in a layered architecture. This uses the redundancy present to provide quick initial responses and a means of addressing drift. This is compatible with the structured testing of the DASH and OSQ and allows for testing of parts of the system independently.

In addition to the inertial sensors, the motors on the robot are instrumented with precision 1000-count encoders. 6000-count encoders were also added to a stiff boom arm used during single-leg testing. Due to the geometry of the boom, this method gives a spatial resolution of 5 mm. Data from the encoders are used to calibrate and assess the quality of the integral routines.

As shown in Table 2, position sensing was performed and tested by one or more of the following sensors: encoders on a test boom, high shock survivability thermal-based micromachined accelerometers, an inertial attitude sensor, and a modified commercial IMU. It is envisioned that the OSQ will use the integrated unit for position sensing, as the sensor needs to be fully self-contained.

Table 2: Sensors on Robot

Sensor	Measurement	Notes
Accu-Coder Model 755 (size 15)	Boom and joint angles	Precision encoder. Used to verify the position estimate based on the kinematics of the boom. The joint positions are also used to extrapolate the robot's configuration during ground contact.
MEMSIC MXA2050AL	Acceleration	Dual-axis, thermal ±10g accelerometer. Has exceptionally high shock survival (50,000g).
InterSense InertiaCube2 Microstrain 3DM-G-ADI	Orientation sensing	Combines magnetometers and angular rate sensors to give orientation data in the Azimuth-Elevation-Roll frame. The sensor could not be fully modified for dynamic tests, so the data could not be fully integrated.
McKinney Technologies MARG-III	IMU	Not installed on the leg. Will be evaluated and installed on the quadruped body.

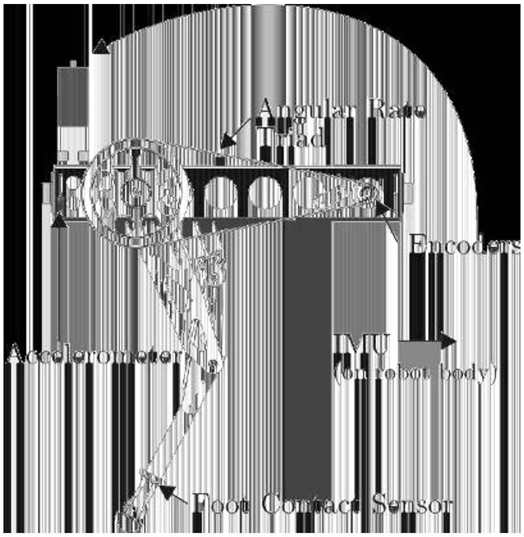

Figure 15: Sensor Layout

4) Calibration and Drift Cancellation

Calibration of the inertial components is especially important as the micromachined sensors that make up the INS have significant temperature variance and part-to-part sensitivities. While it is possible to calibrate these sensors off-board (i.e. using a voice-coil shaker table and/or a turntable), it is preferable to calibrate these sensors while they are mounted on the robot, as variations in temperature and supply voltage need not be manually corrected.

As the sensor output (in Volts) is assumed to be a linear function of the acceleration (in m/s^2), there are five values that need to be determined: sensitivity (s) and offset (k) for both the x and y-axes and the orientation (θ) of the accelerometer with respect to the fixed frame. Static calibration uses the known gravitational acceleration at rest (i.e., 9.8067 m/s^2) [35]. This can be done dynamically via the common signal approximation which assumes that for normal operations the only dominant DC (i.e. zero hertz) acceleration present is gravity. These can be combined as shown.

$$a_x = sV_x + k, a_y = sV_y + k \quad (8)$$

$$a_x^2 + a_y^2 = g^2 = \left(9.8067 \tfrac{m}{s^2}\right)^2$$

$$\theta = atan2\left(a_x, a_y\right)$$

Common (DC) Signal Approximation:

$$lowpass_{0.01Hz}\left(a_y\right) \approx 9.8$$

Since inertial navigation is fundamentally based on dead reckoning, it is highly susceptible to sensor noise and integration drift, especially for extended operations. This is typically addressed through updates against an absolute source. For the Stanford DASH traditional sources are not applicable: GPS is not available for indoor operations; magnetometers are affected by the large, non-deterministic magnetic fields from the robot's motors and amplifiers in close proximity to the senor; and tilt sensors are gravity referenced and thus can not be used in a greater than 1g environment such as the one encountered on the robot. Another alternative to remove noise and drift is to carefully filter the data, often via a Kalman filter. However, Kalman filtering requires *a priori* knowledge of the nature of these errors and assumes that they are deterministic. Filtering also adds delay, may result in the masking of absolute extremes, and does not address numerical integration errors [36].

The OSQ sensor system addresses drift and sensor non-linearity by comparing the position estimated by filtering and integrating sensor data against any independent estimate of position. A best-available methodology is employed such that an estimate is presented and updated as further information becomes available. As shown in the composite flowchart (see also figure 15), this can be performed via three distinct modes:

- **Short operation assumption mode**: This is the simplest and fasted method and estimates the position by simply integrating the signal from the accelerometer. It is based on the assumptions that drift and sensitivity changes are negligible over the short operation periods for this sensor. While helpful for relative changes, absolute acceleration/position changes require very precise calibration of the device.

- **Boom encoder drift estimation/correction mode**: This method (used during initial testing) takes advantage of connection to a rigid boom arm. The difference in position estimation is used to update the model and cancel drift (see also equation 9).

$$sV_j\left(t_i\right)+k+drift=\left[\frac{p_j\left(t_{i-\delta}\right)-2p_j\left(t_i\right)+p_j\left(t_{i+\delta}\right)}{2\delta}\right] \quad (9)$$

  where: i is a given time step, j is either axis (x or y), δ is the difference in time steps. The drift values from several experiments are averaged and stored as a lookup table as a function of the uncorrected acceleration value. Successive experiments iterate and correct the drift.

- **Proprioceptive estimation/updates mode**: Future versions of the quadruped will not be able to make use of the boom for calibration. Because the robot makes cyclical contact with the ground, it is possible to use the kinematics of the robot joint to cancel the drift error. This quadrupedal proprioceptive or "pedometric" filter is currently being evaluated. In particular, it operates by assuming the full kinematics of the robot are known. An advanced form of drift correction is possible by using the values of the joint angles (as measured by precision encoders on the motors) during foot contact with the ground. For this very short amount of time the leg can be viewed as a chain tied to ground and the position can be estimated via kinematic techniques. This method is complicated, however, due to the highly non-linear nature of the spring.

This process allows for updates from an inertial system to be used to update a rapidly determined estimate based on basic sensors and for the estimate to be obtained solely using the accelerometer (and angular rate sensor based attitude estimator if angles body orientation is also desired). As a commercial IMU is planned for the final quadruped, this method allows the robot to work around noticeable IMU delays which may affect the stability of the control system.

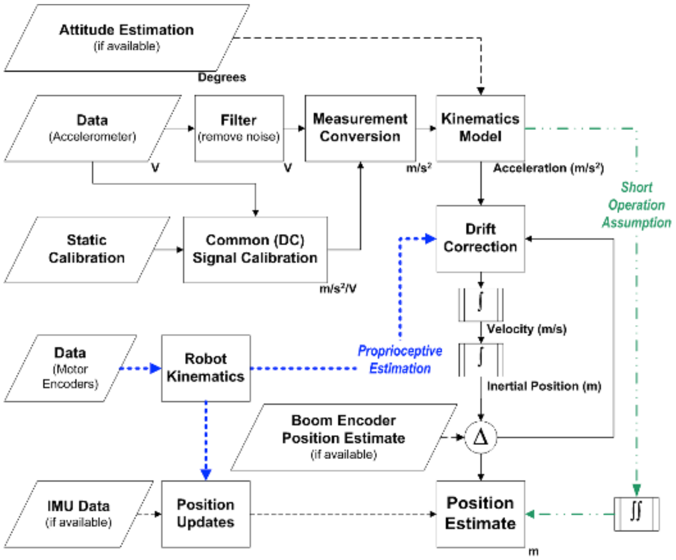

Figure 16: Flowchart of the various position estimation methods used by the Stanford Robot (Dashed line represents optional sensor data, dashed-dotted line is the short operation model, thick-dashed line represents additional data used by proprioceptive method. As updates come from the IMU the position estimate is updated).

5) *Experimental Validation*

The OSQ sensor design combines multiple sensors to provide a fast estimate of the absolute position of the robot where by faster sensors such as the accelerometer and encoders will be used to provide an initial estimate of position that would be updated by a filtering algorithm or an IMU.

Testing confirmed the assumption of the presence of high-shock loads present during full speed bounding. Acceleration loads present depend on the length of the cable and can be as high as on the system can be as 9g (88 m/s^2). This is shown in Fig. 17.

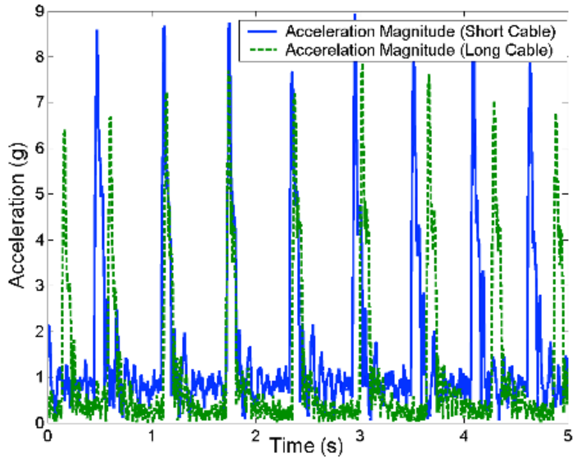

Figure 17: Stanford-OSU DASH leg accelerations (in g's) as a function of time (in seconds). As the cable is shortened, more energy is stored and the accelerations increase.

The OSQ sensor system has been experimentally tested using the short operation assumption and boom encoder update modes. Fig. 18 shows the height estimate without correction (i.e., short operation assumption mode). This mode has a mean error (from the value measured by the encoders) of 3.8%. The boom encoder update mode was retrained for different days and robot configurations. Using this method, drift was attenuated and height estimates had a mean error of less than 1%.

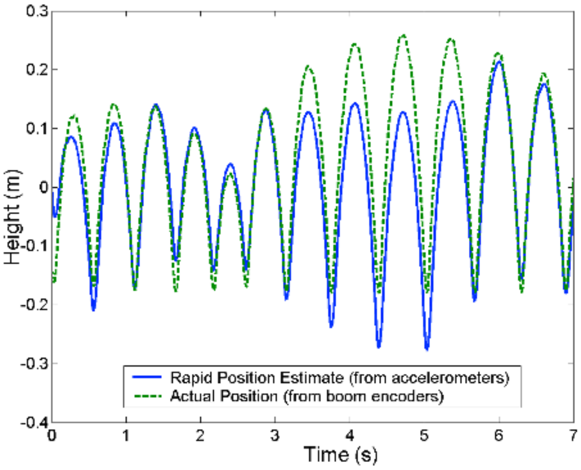

Figure 18. Performance of Rapid Position Estimation. Position estimate (in meters as a function of time in seconds) obtained *without* using boom encoder data for drift correction shows that that drift is present, but small. (Data from 0-2 mps Modified Raibert experiment).

V. Discussion

The focus of this paper is the design of a leg for a quadrupedal galloping machine. At this point the design of the leg is considered to be complete, including the design of suitable control algorithms, and the testing process is well advanced. The integration of four identical legs into the complete machine is also in progress.

The use of electric actuation is attractive because of the simplicity of using batteries for a fully autonomous power supply for the complete machine, and the ease of control that is offered by electric actuation. However, the shortcomings of electric actuators for actively controlled oscillatory systems like this have also been very evident in this project. It is simply not possible to provide the intense power pulses needed from rotary electric motors without resorting to mechanical energy storage that entails considerable complexity, and is very challenging to design. It is noteworthy that Raibert's machines [Raibert book], which were of similar scale, used hydraulic, or hybrid actuation systems to bypass this issue. The problem then becomes one of living with a hydraulic hose tether, or providing an onboard hydraulic power supply, which poses challenges in dealing with weight, bulk and heat dissipation.

The dynamic environment of such a system also poses instrumentation challenges. These challenges range from simple ruggedness problems in an environment experiencing repetitive shock loading of the order of 10 *g*'s, to update rate, interfacing and drift problems. Some of the solutions that have been tested on the leg require further development for the fully three dimensional requirements of the integrated quadruped. Obviously, the boom encoder cannot be used for drift correction if there is no boom!

VI. Acknowledgement

The authors would like to acknowledge the support of the National Science Foundation Grant No. IIS-0208664, and of a National Defense Science and Engineering Graduate (NDSEG) program fellowship held by Surya P.N. Singh, in the conduct of this work.

VII. Refrences

[1] Marhefka, D. W. and Orin, D. E. (2000), "Fuzzy control of quadrupedal running," in Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, pp. 3063–3069.

[2] Marhefka, D. W., Orin, D. E., Schmiedeler, J. P.,and Waldron, K. J. (2003), "Intelligent Control of Quadruped Gallops," To appear in *IEEE/ASME Transactions on Mechatronics*.

[3] M. H. Raibert. Running with symmetry. *International Journal of Robotics Research*, 5(4):3–19, Winter 1986.

[4] J.P.Schmiedeler and K. J. Waldron, "The mechanics of quadrupedal galloping and the future of legged vehicles", *International Journal of Robotics Research*, vol. 18, pp. 1224-1234, 1999

[5] M. G. Pandy, V. Kumar, N. Berme, and K. J. Waldron, The Dynamics of Quadrupedal Locomotion. ASME Journal of Biomechanical Engineering. 110:230-237, August 1988.

[6] P. P. Gambaryan. *How Mammals Run: Anatomical Adaptations*. JohnWiley & Sons, New York, 1974.

[7] R. McN. Alexander, "Body Support, Scaling, and Allometry," in *Functional Vertebrate Morphology*, M. Hildebrand et al., Ed., Cambridge : Belknap Press, 1985, pp. 26-37.

[8] Federal Aviation Administration*, Acceptable Methods,*

Techniques, and Practices – Aircraft Inspection and Repair, AC43.13-1B.

[9] Department of Defense, *Metallic Materials and Elements for Aerospace Vehicle Structures*, MIL-HDBK-5H.

[10] A. Bingelis, *Sportplane Construction Techniques.* Austin, TX: Sportplane Builder Publications, 1986.

[11] J. P. Schmiedeler, "The Mechanics of and Robotic Design for Quadrupedal Galloping," Ph.D. dissertation, Dept. Mech. Eng., The Ohio State Univ., Columbus, OH, 2001.

[12] J. P. Schmiedeler and K. J. Waldron, Impact analysis as a design tool for the legs of mobile robots. In J. Lenarcic and M. M. Stanisic, editors, Advances in Robot Kinematics. pp. 129-136, Dortrecht, 2000. Kluwer Academic Publishers.

[13] H. B. Brown, Jr. and G. Zeglin. The bow leg hopping robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 781–786, Piscataway, New Jersey, 1998. IEEE.

[14] H. Kimura, S. Akiyama, and K. Sakurama. Realization of dynamic walking and running of the quadruped using neural oscillator. *Autonomous Robots*, 7:247–258, 1999.

[15] G. Mennitto and M. Buehler. CARL: A compliant articulated robotic leg for dynamic locomotion. *Robotics and Autonomous Systems*, 18:337–344, 1996.

[16] H. DeMan, D. Lefeber, and J. Vermeulen. Design and control of a robot with one articulated leg for locomotion on irregular terrain. In *Proceedings of the Twelfth CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, pages 417–424, Vienna, 1998. Springer Verlag.

[17] J. G. Nichol and K. J. Waldron, Biomimetic leg design for untethered quadruped gallop. *Proceedings of the 5th International Conference on Climbing and Walking Robots*. Ed. Philippe Bidaud, Faiz Ben Amar. pp. 49-54

[18] *Spectra fiber 2000*, Honeywell International Inc., 2002. Available: http://www.spectrafiber.com/pdfs/hon-pf-ps10-sp2000.pdf

[19] *Handbook of Composites,* 2nd ed., S. T. Peters, Ed. London: Chapman and Hall, 1988.

[20] C. W. Ashley, *The Ashley Book of Knots*. Garden City, NY: Doubleday and Company, Inc., 1953.

[21] D. F. Hoyt and C. R. Taylor. Gait and the energetics of locomotion in horses. *Nature*, 292:239–240, 1981.

[22] R. M. Alexander and A. S. Jayes. A dynamic similarity hypothesis for the gaits of quadrupedal mammals. *Journal of Zoology, London*, 201:135–152, 1983.

[23] N. C. Heglund and C. R. Taylor. Speed, stride frequency and energy cost per stride: How do they change with body size and gait? *Journal of Experimental Biology*, 138:301–318, 1988.

[24] Raibert, M. H. (1986). Legged Robots that Balance, MIT Press, Cambridge, Massachusetts.

[25] Marhefka, D. W. (2000), Fuzzy Control and Dynamic Simulation of a Quadruped Galloping Machine, Ph.D. Thesis, The Ohio State University, Columbus, Ohio.

[26] Jang, J.S.R.; Mizutani, E. (1996). "Levenberg Marquardt Learning for ANFIS Learning", 1996 Biennial Conference of the North American Fuzzy Information Processing Society, Berkeley, CA, pp. 87-91.

[27] Palmer, L., Orin, D., Marhefka, D., Schmiedeler, J., and Waldron, K. (2003). "Intelligent Control of an Experimental Articulated Leg for a Galloping Machine," in Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan.

[28] Raibert, M. H. (1990). Trotting, pacing, and bounding by a quadruped robot. *Journal of Biomechanics*, vol. 23, suppl. 1, pp. 79-98.

[29] Necsulescu, D., Sasiadek, J., Kim, B., and Green, D. (1993). Fusion of inertial and kinematic navigation systems for autonomous vehicles. In *Proc. IEEE-IEE Conf. on Vehicle Navigation and Information Systems*, pp. 462-465.

[30] Barshan, B., and Durrant-Whyte, H.F. (1995). Inertial navigation systems for mobile robots. *IEEE Trans. Robotics Automation* 11(3):328-342.

[31] Roumeliotis, S., Sukhatme, G., and Bekey, G. (1999). Smoother based 3D attitude estimation for mobile robot localization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation* 3:1979-1986.

[32] Bachman, E., Yun, X., and McGhee, R. (2003). Sourceless tracking of human posture using small inertial/magnetic sensors. In *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*.

[33] Waldron, K., Arkin, R., Bakkum, D., Merril, E., and Abdallah, M. (2003). Proprioceptive control for a robotic vehicle over geometric obstacles. In *Proc. of the IEEE Int. Conf. on Robotics and Automation.*

[34] Belloy, E., Sayah, A., and Gijs, M. (2002). Micromachining of glass inertial sensors. *J. of Microelectromechanical Systems* 11(1):85-90.

[35] Roberts, C., and Jachens, R. (1993). Isostatic residual gravity map of the San Francisco bay area. *Map GP-1006.* California: U.S. Geological Survey Geophysical Investigations.

[36] Rogers, R. (2000). Applied Mathematics in Integrated Navigation Systems. Reston, VA: American Institute of Aeronautics and Astronautics.