Stability Control in an Actively-Balanced Bipedal Robot

GrahamMann
School of Computer & Information Science
Edith Cowan University
2 Bradford Street
Mt. Lawley WA, 6050, Australia
g.mann@cowan.edu.au

Abstract

The problem of actively stable bipedal walking on small feet continues to challenge robotic engineers, controller designers, and machine learning theorists. Preliminary results with a real 10-link biped, TarBaby, are presented. This machine is designed to test new ideas in bipedal locomotion, including on-line tuning of key settings in a dynamic motion generator and ultimately automatic expansion of the machine's envelope of stability by systematic motion exercises.

1 Introduction

A good solution to the problem of active-balanced walking on two legs with small feet might eventually lead to robots capable of greater speed, agility and energy efficiency than are available today. Machines with legs could potentially access locations beyond the reach of any machine depending on wheels for locomotion – including buildings designed for humans. Dynamic bipedal walking is also an interesting problem from a purely control engineering point of view. For these and other reasons, research on bipedal robots continues. An ideal solution to bipedal walking should satisfy the following criteria:

Simplicity. It should not unduly complicate to the mechanical or computational complexity of the whole machine.

Efficiency. Animal locomotion systems use legs for efficient energy usage. A solution that optimised power consumption would be attractive for self-contained robots.

Robustness. Not only should falls be minimised, but the solution should afford the best possible performance on uneven slopes, irregular floor surfaces and carrying different loads.

Parameterisability. Offered as a service to higher level control functions, the solution should offer a range of speeds, directions and possibly gait styles. Walking should be able to be stopped and started at arbitrary times without compromising stability.

So far, bipedal robots have only partially achieved these goals. Fair control of some active-balance bipeds has been accomplished by building a detailed dynamic model of the moving linkages, then designing a controller that generates the joint angle vector sequences needed to regulate stability and optimise forward velocity as derived indirectly from inertial sensors in the torso and contact and/or orientation sensors on the feet. For example, Raibert tackled the problem by first developing a single-legged hopper [Raibert, 1986], then generalising that to an agile hopping biped [Playter & Raibert, 1992].

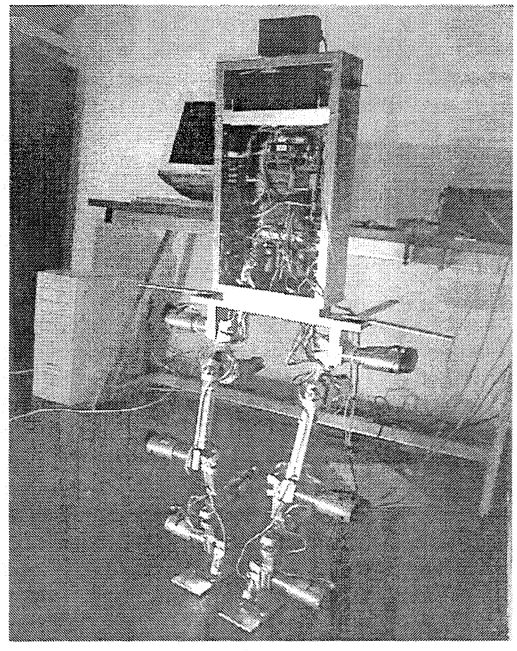

Figure 1. The TarBaby Biped

Furushu and Sano used two independent dynamic models for lateral and sagittal plane motion control in their BLR-G2 and -G3 bipeds [Furushu & Sano, 1990; Sano et.al.,

The author gratefully acknowledges the help of Phil Preston and Barry Drake of the University of New South Wales, and Bruce Armstrong of Electronic & Software Systems P/L.

1992]. [Yamaguchi et.al., 1996] designed specially-instrumented feet with clinometers, gyroscopes and linear potentiometers for their WL-12RVII biped, which enabled walking speeds of up to 0.38m/sec on floor surfaces with vertical irregularities of ±16mm and foot tilt disturbances of ±3°. Honda's P2 and P3 humanoids, which are claimed to walk at speeds of up to 2km/hr, carry loads of 2 to 5kg and use staircases, also gather information from a head-mounted vision system [Honda Motor Corp, 1996].

However, this dynamic modelling approach has disadvantages. First, detailed dynamic models are generally complicated to design, build and tune (violates simplicity). Specifying optimal leg vectors for all possible combinations of commanded input parameters, current postures and desired motions is certainly non-trivial. Second, if something in the physical system changes, the model may no longer generate suitable movement. Linkages and motors deteriorate, circuits behave differently with variations in temperature or battery voltage, and environmental conditions of a robot could be subject to uncontrollable change (violates robustness). Third, a solution developed for a given machine could be difficult to adapt to a biped of a new and different design.

Adaptive control methods, in particular those that can rapidly learn control functions online from error signals fed back during practice offer hope of dealing with these problems. In the past decade, reinforcement learning paradigms such as actual return [Jaakola, et.al, 1995], temporal differences [Sutton, 1988] and model-based learning [Zazala, 1996] have been combined with such adaptive mechanisms as memory-based lookup [Schaal & Atkeson, 1994], neural networks [Warwick, 1996] and fuzzy logic controllers [Verbrugen, et. al., 1999] to tackle robot control problems characterised by noisy or incomplete sensor information, non-Markovian state dependencies and high-speed learning/performance requirements.

Neural network controllers, in particular, have contributed to robust dynamic walking in bipedal robots. Salatian and Zheng simulated off-line and on-line neural network adaptation of bipedal gaits used by their experimental SD-2 8-link biped [Zheng & Shen, 1990] on flat surfaces into gaits suitable for slopes [Salatian & Zheng 1992]. Inspired by that work, Kun and Miller achieved respectable parameterisability and robustness at walking speeds of up to 10cm/s on flat surfaces with a small biped that used three Cerebellar Model Articulation Controllers (CMAC) to learn key parameters of a shuffling gait. Importantly, no detailed engineering models of the machine was needed and the training times required for stable walking were short, in the order of 60 minutes [Miller, 1994; Kun & Miller, 1996].

This paper presents preliminary results with an experimental biped, TarBaby, which is in turn inspired by the work of Kun and Miller. This machine is being developed to help test new ideas in bipedal locomotion, especially automatic expansion of the envelope of dynamic stability.

2 System Description

Physically, the robot consists of a pair of jointed legs linked by a hip bar supporting a torso consisting of metal case (Figure 1). The assembly is 107cm in height and weighs 12kg. The legs are scaled-down, simplified versions of human legs. Each leg possesses all but one of the 6 degrees of freedom in a human leg: that enabling yawing motion of the foot. The upper member of each joint supports a gearhead-motor/shaft-encoder assembly, with the output shaft coupled to the lower member. All gear ratios are 180:1, except the knees, which have a 256:1 ratio for extra torque. Each foot is a 12 x 7 cm plate carrying four force sensing resistor disks coupled to the floor by four square neoprene rubber pads at the corners.

The torso contains a 3AH lead-acid battery, which powers all sensing and control circuitry. The system is designed around a 40MHz 16-bit 80C167-based industrial microcontroller evaluation board manufactured by Phytec. A CyberTrack II 3D motion tracker, which supplies pitch, roll and yaw data to the microcontroller via a 9600 bps serial link, is mounted atop the torso. The force sensing resistors on the feet are individually monitored by eight 10-bit A/D converter channels on the microcontroller board. The ten 12V DC motors are powered via cable from a 30Ah laboratory power supply. Command and control signals from an off-board Windows NT computer are supplied via a 9600 bps serial link cable.

Except for the microcontroller and the motion tracker, all circuits are duplicated bilaterally. The principal processing load on the microcontroller is continuous monitoring of the ten shaft encoders, which might generate pulse rates as high as 6kHz. In to avoid the possibility of miscounting pulses during peak loads, and to simplify the problem of keeping track of shaft positions, a special subprocessor was added, consisting of a pair of 8-bit flash-programmable AT90S8515 RISC microcontrollers, each responsible monitoring for the shaft encoders of one leg. When the 80C167 microcontroller supplies a 5-bit address specifying one encoder, the divider board returns a 16-bit number representing the current position of the corresponding shaft. These numbers can also be reset by a command from the main controller.

3 Learning to Control a Walking Program

From a machine learning perspective, the problem requires a vector of 10 2000-level motor position updates approximately every 20 milliseconds as a nearly continuous learned function of 10 counts from the shaft encoders, 8 force sensors in the feet, and 3 attitude values from the motion tracker. A problem space of that magnitude is probably beyond the capability of known reinforcement learning algorithms, because i) training visits to the problem space are necessarily sparsely distributed, so generalisation is difficult ii) memory demands suffer a combinatorial explosion [Sehad, 1998]

and iii) on-line learning imposes serious computational demands on computation per update cycle (but see [Schaal & Atkeson, 1994]).

Another challenge is thrown down by the passive dynamic systems engineers, who argue that much of the power efficiency and adeptness of natural bipedal walking can be explained in terms of tuned mechanical oscillators formed by the limbs and their springy joints. To support this contention, they have designed mechanical models which achieve limited forms of bipedal motion down slopes with no sensors and actuators and no electronic control [McGeer, 1990; Coleman & Ruina, 1998]. This suggests that with better mechanical design, powered, controlled bipeds could be simplified. How can these developments be accommodated in TarBaby's design?

Rather than trying to develop a control algorithm that learns the entire transformation from raw sensory data to motor commands, the approach taken here aims to use machine learning processes to modulate a simple gait generator. In the TarBaby design, a basic generator cycles through eight phases of movement (Figure 2) while respecting the natural inverted-pendulum dynamics of the robot. The transitions from one phase to the next in sequence are to be triggered when learned sensory thresholds are exceeded. Instead of learning to control the legs, then, the function approximators will learn how to control a program which controls the legs.

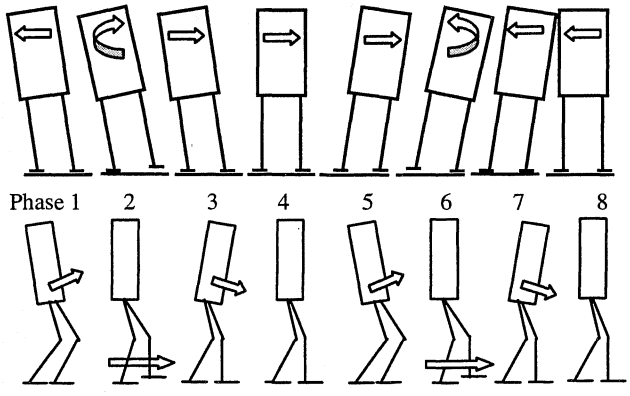

Figure 2. Lateral and sagittal plane motions of the biped during eight phases of the approximate walking algorithm. Arrows show the direction of momentum of the body in each plane.

The natural dynamics of moving TarBaby machine may be characterised as follows: the robot is top heavy and cannot stand on one leg without falling. Therefore, to achieve a forward leg swing, it must sway laterally toward the stance leg side, causing the swing leg foot to lift from the floor. This breaks the angular momentum of the sway, causing the robot to fall back toward the central position (Phases 2 and 6). In this short single-support period, the swing leg must smoothly translate to the desired forward position, then the foot must make a firm, shock-free landing on the floor. Failure to do these smoothly would result in destabilising vibrations in the link structure.

Forward and backward pendulum oscillations probably need to be of smaller magnitude. The joints must operate to ratchet the hips forward with respect to the new swing leg in Phase 1 and the new stance leg in Phase 3. When both feet are on the ground, the linkages become a closed chain system. At those times, it is important to keep the joint angles consistent so that both feet are flat on the floor.

Assuming that allowing the height, length and rate of stepping to be controlled within limits and that allowing a stop to be requested at any point (but not necessarily immediately honoured) counts as reasonable parameterisability, a suitable algorithm is described as shown in Figure 3.

Walk (Step_Length, Step_Height, Step_Rate)

Loop Until (RequestToStop)

1. Extend left knee, bring hip forward to point over left foot. Terminate on left knee position passing offset threshold θ_F. This motion also generates right tilt to maximum θ_R
2. Lift left foot and translate to new forward position using step parameters by open-loop control. Terminate on floor contact.
3. Shorten left leg to move hips to an offset point midway between the feet. Terminate after a fixed time interval T_F.
4. Pause. Wait for integrated AC signal from feet to decay. Continue when a stability measure falls below threshold θ_P.

5-8. Repeat steps 1- 4 for right leg.

End Loop

RequestToStop:
Continue to Phase 4. Return control to command level.

Figure 3. Approximate Walking Algorithm

Of the several thresholds and limits in this scheme, which can be set as constants, and which need to be learned? If we wish to avoid detailed modeling, it will be necessary to regulate the difficult-to-predict parameters in movement components which are most likely to result in toppling. According to [Miller, 1994] these are the side-to-side motions, the backward and forward motions, and the foot landings. Lateral rocking depends on relative leg length (knee adduction), and may be measurable by distribution of forces on the feet, but only when the foot is on the ground. Front-back rocking depends on the combination of the relative position of each hip to its corresponding foot (hip-foot offsets), which may also be measurable by foot force sensing.

The limit for stability is the maximum amplitude of these motions for a given input stepping triple. The thresholds for these (θ_R , θ_L , θ_F and θ_B) would be learned during an initial standing balance exercise in which the machine explores the limits of its stability in these directions. Error feedback would be the difference between the desired Step_Length and observed step length. If the foot touched the floor before Step_Length had been achieved, for example, the parameters in Step 1 would be adjusted to allow it more time in the air. The other parameters, T_F and θ_P, should be able to be preset as empirical constants.
Foot placement may be more complicated. Without

knowing exactly how the floor surface ahead will interact with the foot, it will be difficult to compute how to set the ankles. Perhaps the feet can be made to passively or actively comply with the surface as it descends. Only for a short time, however, since the ankles' contribution to overall balance must recommence in the early stages of Phases 3 and 7. On flat floors, it may be possible to compute a placement prediction function based on the commanded swing leg kinematics. The error term is clearly the observed deviation from a perfectly even distribution of forces on the landing foot.

If this tuning is to be carried out by some kind of adaptive function approximator, questions arise. What kind of approximator will be needed? How will the sensors on TarBaby be arranged to serve as input and training error signals?

5 Results

Details of a simple experiment designed to arrive at a suitable measure of the biped's stability during walking movement are now presented. [Sano et.al., 1992] emphasised the value of angular momentum around the ankle as a measure of quality of balance. [Kun & Miller, 1996] integrated a measure the distribution of forces between the heels and toes of their biped's feet to produce a "quality of balance" signal.

Since at least one foot is always on the ground, TarBaby's feet sensors can be used for this purpose. The total force on foot i is simply the sum of values for all four corner sensors:

$$\text{ForceF}_i = \sum_{j=1}^{4} F_{i,j}$$

Figure 4 shows ForceF values collected from the right foot during a 5-step test walk generated by a rudimentary form of the approximate walking algorithm. This was a

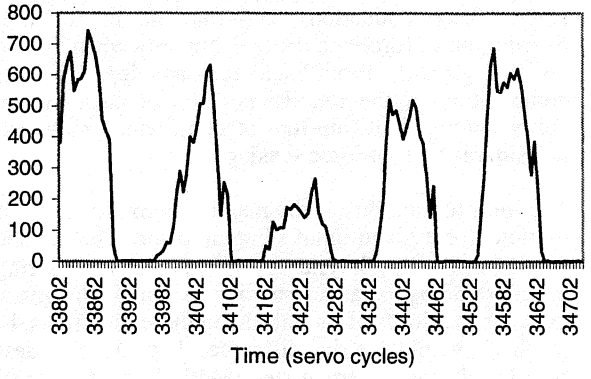

Figure 4. Force acting on one foot during 5 steps generated by the approximate walking algorithm. Servo cycle labels are 1.08 seconds apart.

inverse kinematic model of walking that was not phase synchronised to the natural dynamics of the robot and was not tuned by adaptive learning. (This model will be superceded by the fully-implemented, adaptively tuned algorithm of Figure 3.) Frequent human assistance was required to stop the machine from toppling. A clear periodic force signal appears for each step, but the magnitude varies considerably, so any measure of differential distribution of force on the foot will need to be a proportion of total force. If the force on the single right-front sensor of foot i is represented by the expression $F_{i,\,rf}$ then a convenient lateral center of force in the range [-32,32] may be defined as

$$\text{LatCoF}_i = \frac{F_{i.\,rf} + F_{i.\,rr} - F_{i.\,lf} - F_{i.\,lr}}{(\text{ForceF}_i + 32)/32}$$

This will be a maximum when most of the force is on the right side of the foot and minimum when it is on the left side. Another measure, SagCoF, may be similarly defined that reaches a maximum when most of the force is on the front of the foot.

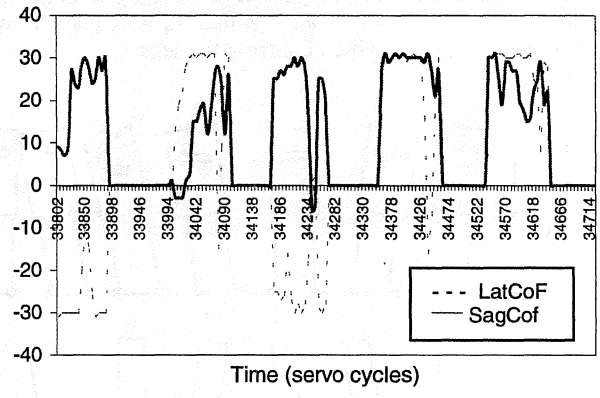

Figure 5. Lateral and sagittal plane forces on one foot during 5 steps generated by the approximate walking algorithm (without adaptive tuning).

LatCoF and SagCoF measures taken during the same 5-step test walk as above are shown in Figure 5. The values periodically fall to zero while the right foot is in the swing phase. The need for sensory triggers to phase-synchronise stepping to the natural oscillations of the body is shown clearly, as the phase of LatCoF shifts with respect to SagCoF. At small phase differences, the machine tended to slew diagonally, requiring intervention to avoid a fall.

6 Discussion

The derived measures of side-to-side and back-and-forward motion, LatCof and SagCof appear to be usable as indicators of stability, provided they are taken from a

support foot. The eight-phase approximate walking algorithm must respect the natural dynamics of the oscillating robot by monitoring these measures such that the lateral and sagittal plane motions of the robot are properly synchronised. The exact sychronisation pattern is determined by the leg vectors and should appear as a natural outcome of the triggered sequence of phases once a stable walk has been learned.

Some kind of function approximator must regulate each at-risk motion during walking. These would ideally be able to avoid becoming trapped in local minima, deal with the reduced problem space presented by the algorithm, quickly converge on a solution during training and approximate the learned functions at over 50Hz once learning was complete. CMACs [Glanz, et.al., 1991] and their self-organising, fuzzy logic descendents, FCMACs [Nie & Linkens, 1992] have these properties and are currently being studied by the author for use in this application.

References

[Coleman & Ruina, 1998] M.J. Coleman and A. Ruina. An uncontrolled walking toy that cannot stand still. *Physical Review Letters*, 80:3658--3661.

[Furushu & Sano, 1990] Junji Furusho and Akihito Sano. Sensor-based control of a nine-link biped. *International Journal of Robot Research*, 9(2):83--98, April, 1990.

[Glanz, et.al., 1991] Filson H. Glanz, W. Thomas Miller and L. Gordon Kraft. An overview of the CMAC neural network. *IEEE Conference on Neural Networks for Ocean Engineering*. pages 301—308. Washington D.C., 1991.

[Honda Motor Corp., 1997] Unpublished notes from *15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997 and Honda website http://www.honda.co.jp/english/technology/robot/index.html

[Jaakola, et.al, 1995] T. Jaakola, S.P. Singh and M.I. Jordan. Reinforcement learning algorithm for partially observable Markov decision processes. In *Advances in Neural Information Processing Systems 7*. Morgan Kaufmann, 1995.

[Kun & Miller, 1996] Andrija Kun and W. Thomas Miller III. Adaptive balance of a biped robot using neural networks. *Proceedings of the 1996 IEEE International Conference on Robotics & Automation*, pages 240--245, Minneapolis, Minnesota, April, 1996.

[McGeer, 1990] Tad McGeer. Passive Dynamic Walking. *International Journal of Robot Research*, 9(2):62--82, April, 1990.

[Miller, 1994] W. Thomas Miller III. Real-time neural network control of a biped walking robot. *IEEE Control Systems*. 40(1):41--48, February, 1994.

[Nie & Linkens, 1994] Junhong Nie and D.A. Linkens. FCMAC: A fuzzified Cerebellar Model Articulation Controller with self-organising capacity. *Automatica*, 30(4):655--664 ,1994.

[Playter & Raibert, 1992] Robert R. Playter and Marc Raibert. Control of a biped somersault in 3D. *International Symposium on Theory of Machines and Mechanisms*, pages 669--674, Nagoya, Japan, September, 1992.

[Raibert, 1986] Marc H. Raibert *Legged Robots that Balance*. Cambridge, MA. MIT Press, 1986.

[Salatian & Zheng, 1992] A. W. Salatian and Y.F. Zheng Gait synthesis for a biped climbing sloping surfaces using neural networks -II. Dynamic Learning. *Proceedings of the 1992 IEEE Conference on Robotics and Automation*, pages 2607-2611, Nice, France, 1992.

[Sano et.al., 1992] Akihito Sano, Junji Furusho and Wataru Yabuki. Biped walking on floor and stairs based on torque distribution. *International Symposium on Theory of Machines and Mechanisms*, pages 663--668, Nagoya, Japan, September, 1992.

[Schaal & Atkeson, 1994] Stephan Schaal and Christopher Atkeson. Robot juggling: An implementation of memory-based learning. *Control Systems Magazine*, 14(1):57--71,1994.

[Sehad, 1998] S. Sehad. Neural reinforcement learning for robot navigation. In Lakhmi. C. Jain & Toshio Fukuda (Eds.) *Soft Computing for Intelligent Robotic Systems*. Physica-Verlag, Heidelberg, 1998, pages 159--184.

[Sutton, 1988] Richard. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9--44, 1988.

[Warwick, 1996] Kevin Warwick. An overview of neural networks in control applications. In A.M.S. Zalzala and A.S. Morris.(Eds.) *Neural Networks for Robotic Control*. Ellis-Horwood, London, 1996, pages 1--

[Verbruggen, et.al., 1999] Henk B. Verbruggen and Robert Babuška (Eds.) *Fuzzy logic control : advances in applications*. River Edge, New Jersey: World Scientific, 1999.

[Yamaguchi, et.al., 1996] Jin'ichi Yamaguchi, Noboru Kinoshita, Atsuo Takanishi and Ichiro Kato. Development of a biped walking robot adapting to the human's living floor. *Proceedings of the 1996 IEEE International Conference on Robotics & Automation*, pages 232--239, Minneapolis, Minnesota, April, 1996.

[Zalzala, 1996] Ali. M. S.Zazala. Model based adaptive neural structures for robotic control. In A.M.S. Zalzala and A.S. Morris.(Eds.) *Neural Networks for Robotic Control*. Ellis-Horwood, London, 1996, pages 80--105.