Approximate Policy Transfer applied to Simulated Bongo Board Balance

Stuart O. Anderson, Jessica K. Hodgins, Christopher G. Atkeson
Robotics Institute
Carnegie Mellon University
soa,jkh,cga@ri.cmu.edu

***Abstract*— Developing global policies for humanoid robots using dynamic programming is difficult because they have many degrees of freedom. We present a formalism whereby a value function for a humanoid robot can be approximated using the known value functions of similar systems. These similar systems can include approximate models of the robot with reduced dimensionality or trajectories derived from human motion capture data. Once an approximate value function is known, a local controller is used to compute control signals. The approximate value function provides information about the global strategies that should be used to solve the task. The local controller provides complementary information about the robot's dynamics. We present an implementation of this strategy and simulation results generated by this implementation.**

I. Introduction

We are interested in humanoid robots that use human-like motions to perform difficult balance tasks. Using human-like motions allows robots to appear more natural and may improve their social acceptance. Additionally, a robot that uses human-like motions may be better able to model a human subject in experiments or tasks where a human subject cannot be used or is undesirable. Finally, controllers that are intended to produce human-like motions are well suited to learning from observing humans completing similar tasks.

We created a controller that allows a simulation of the Sarcos Primus humanoid biped (Fig. 1) to balance on a Bongo Board in the presence of random perturbations. A Bongo Board is a balance toy consisting of a plank resting on a cylinder that is in rolling contact with both the ground and the board (Fig. 2). Additionally, we describe a general feature-based framework for transferring the global structure of value functions between systems and for combining multiple value function approximations.

Motion capture data and dynamic programming have both been used successfully to generate motion for animated human figures [1], [2], [3] and to control robots [4]. However, each of these tools is typically limited to solving a specific type of problem or producing a specific type of solution. Dynamic programming produces global policies but is, in general, limited to problems with a small number of degrees of freedom. Motion capture-based methods are able to produce motion for systems with many degrees of freedom, but are typically unable to control the system when the state is far from any of the observed states. Our approach combines motion capture from a human subject balancing on the Bongo Board with a dynamic programming solution for a reduced representation of the robot to produce a controller in the full state space.

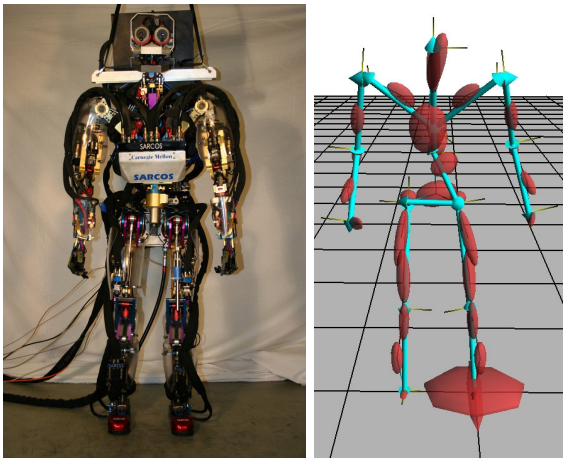

Fig. 1. Sarcos Primus hydraulic biped and Sarcos Primus simulation model

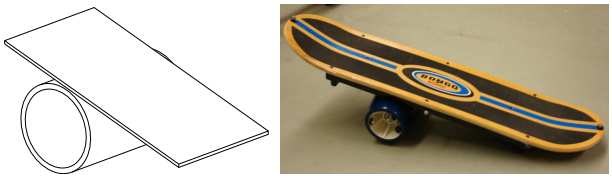

Fig. 2. Bongo Board balance toy

The intent of our approach is to capture information about the global strategies used by these similar systems and then apply that information to solving the global control problem for the humanoid robot. If the systems are defined such that an equivalence between the dynamics can be shown, it is possible to project the value function from one system onto a manifold embedded in the state space of the other without approximation. In the more general case where the similarity between the systems is based only on intuition, as in the case of the similarity between a human and a humanoid robot, an approximate value function can still be defined. However, no guarantees can be made about the validity of this approximation. Intermediate cases between these extremes may exist where limited bounds on the approximated value function can be given.

Once an approximate value function has been defined,

the control signals that should be sent to the robot must be determined. If sufficient guarantees are provided for the approximation bounds on the value function, a local solution to the Bellman equation can be used, as when deriving a policy directly from the true value function. In the more general case where the approximated value function represents only an educated guess at the true value function, less local control methods, such as model predictive or receding horizon control, can be used to avoid instability resulting from local errors in the value function.

More generally, we define a formalism describing a method for transferring global information about solution strategies between similar systems. This transformation depends on the identification of key features shared by both systems, and on estimates of how salient those features are in determining the value function of the controlled system. We also describe how this transformation can be implemented for real systems with many degrees of freedom.

There is a large body of work similar in spirit and methodology to our controller design process. Like our work, the templates and anchors framework used in the RHex project [5] builds controllers for complex systems based on simplified representations. The SLIP model of running gait [6] and its various robotic embodiments [7] as well as virtual model control [8] are also examples of this approach. Even Zero Moment Point control [9] can be viewed as a method for representing the balance of a complex system in a simplified representation (the ZMP location). The operational space control formulation and hierarchical control methodologies [10] combine tasks of differing priority. This approach is similar to the approach our system uses maintains balance while preferring human-like movements. Physically based motion transforms [11] are also similar in many ways to our work, in that they allow motions observed in motion capture data to be transformed into a reduced representation, then reconstructed on the full model. The trajectory generation work in [12] uses PCA to reduce human motion to a simplified representation, and showed that optimizations within a behavior specific simplified representation could produce natural motions. Finally, we are able to solve the dynamic programming problem in the reduced representation using the stochastic methods described in [13], [14].

Section II of this paper describes the framework we use for transferring value functions between similar systems. Section III describes the simulations we conducted to demonstrate this framework. Section IV describes the results of these simulations. Section V discusses these results and presents directions for future applications of this framework.

II. General Value Function Transfer

The definitions provided in this section present a framework for using the known value functions of one or more example systems to compute an approximation of the unknown value function for a similar system. In this paper we investigated two the pairs of similar systems. The first was pair was two inverted pendulums with slightly different kinematic and dynamic parameters. The second pair was a two link inverted pendulum on a Bongo Board and the sarcos biped on the Bongo Board. The notion of similarity between systems is intentionally left vague in the general case, although additional constraints on the quality of the value function approximation can be given if the similarity between the systems is defined more strictly. In general, it remains the responsibility of the control engineer to identify features shared by the example and target systems and to provide estimates of the salience of those features in estimating the value function.

A. Defining the Approximate Value Function

We consider a system S and a similar system R. Let $\mathbf{x}_s$ be a state of S and $\mathbf{x}_r$ be a state of R. A value function defined over the states of R, $V_r(\mathbf{x}_r)$ is known. We wish to define an approximation of the value function of S, $\hat{V}_s(\mathbf{x}_s) \approx V_s(\mathbf{x}_s)$, based on V_r. To do this we define a pair of functions, f_s and f_r, that map from states in S and R to a common feature space. These features describe shared attributes of S and R that are believed to be related to V_r. For example, in the case where S is a humanoid robot and R is a two link pendulum, relevant features could include the total kinetic energy, location of the system's center of mass, and the angular and linear momenta of the system.

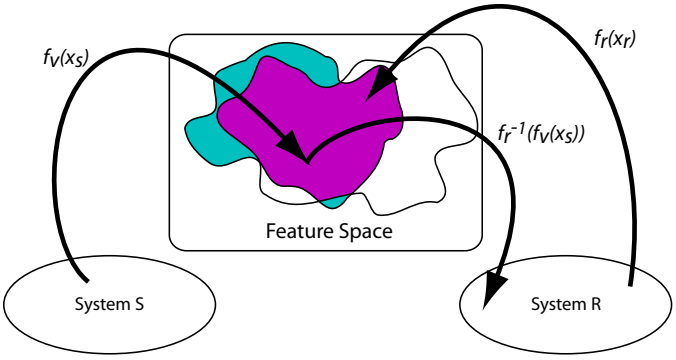

Fig. 3. Illustration of the relationship between the two similar systems, R and S, and feature space. The region of feature space to which states from both systems map is shaded magenta, while the regions containing states from only one system are shaded cyan and white. Arrows indicate the functions mapping between state and feature spaces.

The feature vector, F, is the conduit through which information can move between the two systems. Because there is no other source of information the problem of finding $\hat{V}_s(\mathbf{x}_s)$ is equivalent to finding the value function over features, $V_f(f_s(\mathbf{x}_s))$. To construct V_f, V_r can be mapped into feature space by using the inverse of f, $V_{f_r}(\mathbf{F}) = V_r(f_r^{-1}(\mathbf{F}))$. Note that because $f(\mathbf{x}_r)$ may be a many to one mapping V_{f_r} may be a one to many mapping.

Furthermore, because f is not an "onto" mapping there may be areas of feature space over which V_{f_r} is not defined. In the example given above, it's possible that the humanoid robot can move its center of mass further than the two link pendulum. When the robot is in such a state the corresponding feature vector does not correspond to any state of the two link pendulum.

The value of V_f at points outside the domain of V_{f_r} should reflect both an extrapolation of V_{f_r} and a penalty that increases with distance from the domain of V_{f_r}. One possible choice would be to use a zeroth order extrapolation of V_{f_r} at the point on the domain nearest $\mathbf{F}$, $nn_d(\mathbf{F})$, and to use a penalty that is linear in $(\mathbf{F} - nn_d(\mathbf{F}))$. In our example, this would apply a penalty to states of the robot with center of mass locations not achievable by the two link pendulum that was linear in the distance between the robot's center of mass and the closest center of mass location the two link pendulum could achieve.

Inside the domain of V_{f_r} multiple values can be associated with a single $\mathbf{F}$. The maximum value in the set $V_{f_r}(\mathbf{F})$ is a pessimistic choice consistent with minimax dynamic programming[4]. In our example, many configurations of the humanoid robot correspond to a single center of mass location, in which case the function mapping from robot states to a feature vector consisting only of the center of mass location is non-invertible.

It is often the case that the salience of some features or groups of features will be known to vary predictably with the state of R. For example, kinetic energy goes to zero at stable states of the Bongo Board balance problem, but is not particularly helpful in determining the relative values of states with kinetic energies in the range typically encountered during the balance task. The group of features consisting of the Bongo Board's state, the kinetic energy of the system, and the lateral center of mass displacement constitute a sufficient set of features for determining whether a state of S or R is stable.

In this case, any state $\mathbf{x}_s$ for which these elements of $f(\mathbf{x}_s)$ match the features of a stable state of R has a known value of zero, because $\mathbf{x}_s$ is guaranteed to be a stable goal state. In this case the other elements of the feature vector have no salience, in particular, any penalty associated with $f(\mathbf{x}_s)$ being outside the domain of V_{f_r} should go to zero at the goal states. We define a scalar function $W(\mathbf{F})$ that goes to zero when $\mathbf{F}$ is sufficient to determine V_f exactly. In the example given, W would go to zero any time the salient group of features indicated a stable state. When computing V_f the penalty associated with feature vectors outside the domain of V_{f_r} is scaled by W. Choosing the definition of W carefully is necessary to ensure that stable states of S are local minima of $V_f(f(\mathbf{x}_s))$.

B. Computing the Approximate Value Function

Computing V_f either implicitly or explicitly is prohibitively difficult in general. An implicit computation of V_f requires computing the inverse of the multivalued function f_r, then finding the nearest point in feature space for which that inverse was defined. An explicit table-based computation of V_f would require a prohibitively large amount of storage space. An explicit second order approximation of V_f depends on the composition of f_r and V_r. Even if a good sampling point for approximating V_r is known, as in the case when V_r is the result of our approximate dynamic programming procedure, the choice of good sampling points in feature space depends on the structure of f_r.

Rather than compute V_f explicitly we define a projection $P(\mathbf{x}_s) = \mathbf{x}_r$ and a complimentary scalar valued loss function $P_e(\mathbf{x}_s)$. For a properly defined P and P_e,

$$V_f(f_s(\mathbf{x}_s)) \approx V_r(P(\mathbf{x}_s)) + P_e(\mathbf{x_s})$$

Because the projection function has no information about the value of the state it projects to, it cannot select the maximum valued state of all possible projections with the same loss error. Additionally, P_e must perform the role of the W salience function, in that its value is determined not solely by the magnitude of the projection error, but by the estimated salience of that error to the value function estimation.

C. Using the Approximate Value Function for Control

Although the approximated value function is believed to be valid at larger scales, it is generally a poor local model of the true value function of S. However, there are cases where the approximate value function is known to be exactly equal to the true value function, as it is at stable points. If R is the same dynamic system as S with additional constraints applied to reduce the available degrees of freedom, then the value function of R will be exactly equal to the value function of S if optimal trajectories of S that enter the subspace defined by those constraints never leave that subspace. Additionally, if optimal trajectories of S make only small deviations from the subspace it is reasonable to assume that the value function of R along the subspace is a good approximation of the value function of S along that subspace. If the approximated value function is equal to the true value function then the one step Bellman equation can be used to determine the correct control signals to apply to the system.

In the general case, however, it is necessary to use a local controller with a planning horizon that extends to a range at which the approximate value function becomes useful.

III. Examples

We provide examples using this framework to create controllers for various models of the Bongo Board problem. We initially identify the models used, then describe the features of each system we considered when implementing the projection and loss functions between systems. Finally, we describe the local controllers used to track the global value function approximations.

A. Models

Our experiments used three dynamic models, a model of the Bongo Board, a full model of the Sarcos Primus system, and a reduced dimensionality double pendulum approximation of the Sarcos Primus system. This section describes the design of these models.

1) Bongo Board: The Bongo Board is modeled without a rolling contact between the board and wheel. The model used is a kinematic chain consisting of a rotational joint followed by a translational joint. The axes of these joints intersect at a single point. The axes are also perpendicular to each other and the gravity vector. Even for small displacements this model fails to reproduce the translation of the support point due to the rolling translation of the wheel. Additionally, our physical Bongo Board has significant rolling friction that is not captured in our model.

2) Primus Biped: The Sarcos Primus System is a human scale hydraulic biped. The Primus System includes 34 actuated degrees of freedom, not including the hands and face. Because of the high maximum torques and large range of motion at the joints, the system is able to reproduce many motions observed in humans accurately. However, the distribution of mass in the Primus System is not similar to human mass distribution [15]. In particular, the lower half of the legs have proportionately greater mass than a human, a difference that significantly influences the dynamics of walking gaits.

We modeled the Primus System using information from the CAD model of the robot used for its construction. There are significant differences between this model and the built robot, such as the presence of hydraulic fluid and hoses. These additions increase the mass, change its distribution, and exert unmodeled torques on joints. The model used in our simulations omitted the finger and face degrees of freedom for computational efficiency and because our motion capture data does not include any information about these degrees of freedom. We consider the model to be a precise kinematic match to the physical robot, but only a rough match in terms of dynamics.

3) Reduced Dynamic Model: The reduced model of the Bongo Board balance task uses a two link inverted pendulum to represent the body of the balancer. The two link inverted pendulum model has been used often to represent the human body in balance tasks [16]. The mass, moment of inertia, and position of the center of mass of the upper and lower links of the pendulum are set equal to those of the upper and lower body of the robot in its reference pose. The length of the lower link is set to the height of the robot's torso abduction joint. These features make it a good example system for determining the general form of the value function for the full robot. The pendulum does not represent the closed loop kinematic constraints present when two feet are placed on the Bongo Board.

B. Controllers

We implemented LQR controllers for both the reduced and full models. These controllers required that the system be linearized about a point at which all steady-state accelerations could be compensated by static torques. We chose to linearize the system with torques applied to cancel all accelerations of the body. These torques decouple the dynamics of the body so that each degree of freedom couples only with the tilt and offset of the Bongo Board. In the case of the full model, the kinematic loop introduced by both feet being in contact with the board reduced the number of degrees of freedom in the system by twelve. We identified the linear subspace of possible time derivatives of the current state, S, at a pose q using the null space of the Jacobian of the foot constraint with respect to changes in position:

$$\mathbf{N} = null(\mathbf{J}(\mathbf{q})) \quad (1)$$

$$\mathbf{S} = \begin{pmatrix} \mathbf{N} & 0 \\ 0 & \mathbf{N} \end{pmatrix} \quad (2)$$

We solved the control problem for the reduced system using dynamic programming. The particular method we used is based on the stochastic sampling and update method described in [14]. Additionally, the Q matrix defining the component of the objective function computed from the system's state was chosen to be the inverse of the PCA basis of the example motions. The intent of this cost function is to have the dynamic programming solution mimic the motions made by the human when possible by applying a large cost to states that are unlikely to appear in the example data.

IV. Results

We present two simulation results demonstrating our framework. The first result shows a simulation in which a value function for the reduced model was transferred to a second version of the reduced model in which the length of the base link was increased by three percent. We show that the system was stable under a combination of local receding horizon control and the approximated value function. The system was not stable when either component was used in isolation. The failure of the receding horizon control alone indicates that the disturbance was sufficiently large that the extended horizon provided by the receding horizon control was unable to stabilize the system without the additional information from the approximate global value function. The failure of the policy derived directly from the approximated value function without receding horizon control indicates the the differences between the two systems are significant enough that the receding horizon control is necessary. We show both the cumulative cost of each policy (Fig.4) and the estimated future cost as a function of time (Fig.5).

The second result shows the Sarcos Primus model simulation balancing using a policy derived for the reduced model. This system is very sensitive to disturbances that move it away from the manifold on which the approximation is dynamically equivalent to the full model. We use a computed torque constraint to keep the system on the manifold where the reduced system's dynamics are identical to the full system. We show the estimated value of the system's state versus time as it recovers from a small disturbance (Figs6 and 7). Note that the system finds a stable state that does not have zero value under the approximated value function.

V. Discussion

A complex system can be "factored" into many constrained nonlinear subspaces. These subspaces can be solved by dynamic programming exponentially faster than the full system.

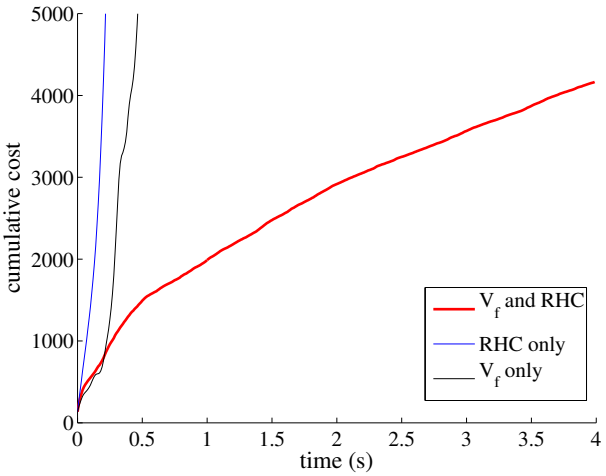

Fig. 4. Cumulative cost of policies generated from receding horizon control only, the transferred value function only, and the combination of receding horizon control with the transferred value function. Note that only the combination of both produces a stabilizing policy for these initial conditions.

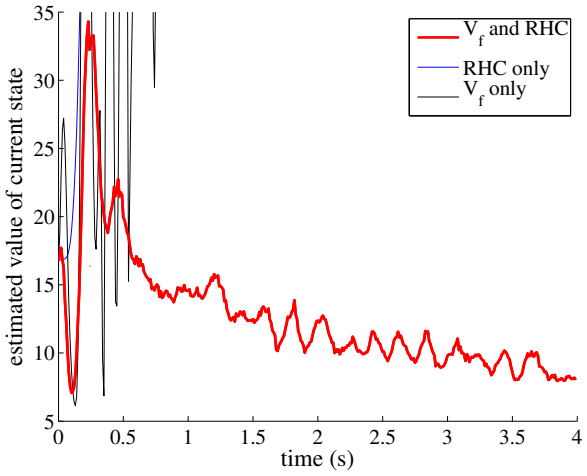

Fig. 5. Estimated value of state as a function of time for the three policies. Note that only the policy that combines receding horizon control with the transferred value function is stable for these initial conditions.

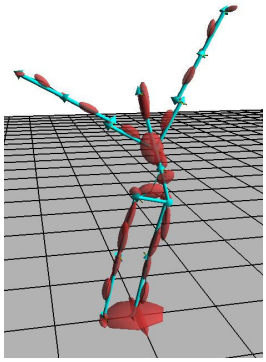

Fig. 6. Primus simulation balancing on a Bongo Board model with one limb in contact

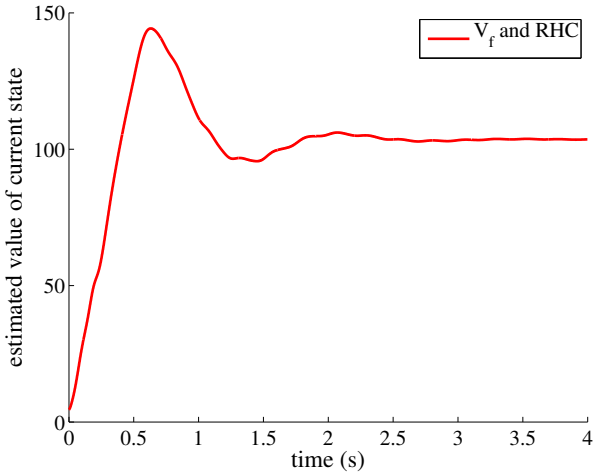

Fig. 7. Estimated state value versus time for Primus balancing on Bongo Board.

When the subspace is a holonomic constraint that permits only a single degree of freedom, the value function over the subspace can be found quickly with dynamic programming methods. The value functions found for these subspaces will approximate the true value function of the full system well if optimal trajectories of the full system tend not to diverge from that subspace [17]. The feature-based value function approximation described in this paper allows the solutions to these reduced systems to be combined into an approximate solution for the entire system.

The performance of the current implementation is severely limited by our implementation of receding horizon control. We use Nelder-Mead nonlinear optimization to determine the torques applied at 0.01 second intervals. In practice, this method limits the viable lookahead horizon to 0.1 seconds or less. The system must stay close to regions where the value function is accurate and is unstable for many disturbances.

VI. Future Work

We described how the objective function in our dynamic programming solution is based on the principal component analysis of the human motion capture data projected onto the reduced representation. This choice of objective function was intended to create solutions that used motions similar to those observed in the motion capture data. A more robust approach to achieving this goal may be to compute the cost functions for both state offset and applied torques using the motion of the full model. However, this would require a mapping from the reduced model to the full model to be defined.

Trajectories from motion capture data can be used to define nonlinear subspaces that can be treated as reduced models of the full system. By identifying trajectories toward which other trajectories tend to converge, we can find trajectories that are probably optimal trajectories of the full system. Finding the value function along one of these trajectories is computation-

ally inexpensive and should provide a good approximation of the value function of the full system along the same trajectory. We intend to experiment with this procedure for developing control policies directly from human motion capture data.

Ultimately, we intend to use this methodology to control the physical Sarcos Primus system. The poor dynamic match between our simulation and the hardware will be a significant obstacle. We intend to automatically identify the dynamics of the robot along the two dimensional nonlinear subspaces defined by motion capture trajectories, then solve these identified dynamics using a DDP trajectory optimizer. This will provide a policy and value function along the subspace that reflects the true dynamics of the hardware without exhaustively identifying large regions of the robot's state space.

An additional area of future research is identifying more widely applicable formal measures of the similarity between systems. The feature sets and similarity metrics described in this paper either guarantee that the value functions of the reduced and full system are precisely equal or provide no bounds on approximation error. It is desirable to identify similarity metrics and features that allow some bounds to be given over large regions of state space. We are also interested in applying machine learning techniques to feature selection and salience determination.

ACKNOWLEDGMENT

This material is based upon work supported in part by the National Science Foundation under NSF Grants CNS-0224419, DGE-0333420, ECS-0325383, and EEC-0540865.

REFERENCES

[1] P. S. A. Reitsma and N. S. Pollard, "Evaluating motion graphs for character navigation," *ACM Transactions on Graphics*, vol. 26, no. 4, Oct. 2007.

[2] D. Wiley and J. Hahn, "Interpolation synthesis of articulated figure motion," in *Computer Graphics and Applications*, vol. 17, no. 6. IEEE, Nov/Dec 1997, pp. 39–45.

[3] V. B. Zordan and J. K. Hodgins, "Motion capture-driven simulations that hit and react," in *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA: ACM Press, 2002, pp. 89–96.

[4] J. Morimoto, G. J. Zeglin, and C. G. Atkeson, "Minimax differential dynamic programming: application to a biped walking robot," in *Proc., Int. Conf. on Intelligent Robots and Systems*, vol. 2. IEEE, 2003, pp. 1927 – 1932.

[5] U. Saranli, M. Buehler, and D. E. Kodi1tschek, "Rhex: A simple and highly mobile hexapod robot," *Int. J. Robotics Research*, vol. 20, no. 7, pp. 616–631, 2001.

[6] R. Blickhan and R. J. Full, "Similarity in multilegged locomotion: Bouncing like a monopode," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 173, no. 5, pp. 509–517, November 1993.

[7] M. H. Raibert, *Legged robots that balance*. Cambridge, Massachusetts: The MIT Press, 1986, iSBN 0-262-18117-7.

[8] J. E. Pratt, C.-M. Chew, A. Torres, P. Dilworth, and G. Pratt, "Virtual model control: An intuitive approach for bipedal locomotion," *Int. J. of Robotics Research*, vol. 20, no. 2, pp. 129–143, 2001.

[9] M. Vukobratovic, "How to control the artificial anthropomorphic systems," *IEEE Trans. System, Man and Cybernetics SMC-3*, pp. 497–507, 1973.

[10] L. Sentis and O. Khatib, "A whole-body control framework for humanoids operating in human environments," *2006. ICRA 2006. Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 2641–2648, May 2006.

[11] Z. Popović and A. Witkin, "Physically based motion transformation," in *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM, 1999, pp. 11–20.

[12] A. Safonova, J. K. Hodgins, and N. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *ACM Transactions on Graphics (SIGGRAPH 2004)*, vol. 23, no. 3, August 2004.

[13] C. G. Atkeson and J. Morimoto, "Non-parametric representation of a policies and value functions: A trajectory based approach," in *Advances in Neural Information Processing Systems*, vol. 15, 2003.

[14] C. G. Atkeson and B. Stephens, "Random sampling of states in dynamic programming," in *Advances in Neural Information Processing Systems*, 2007.

[15] S. O. Anderson, M. Wisse, C. G. Atkeson, J. K. Hodgins, G. J. Zeglin, and B. Moyer, "Powered bipeds based on passive dynamic principles," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, Dec. 2005, pp. 110–116.

[16] R. J. Peterka, "Sensorimotor integration in human postural control," *Journal of Neurophysiology*, vol. 88, pp. 1097–1118, 2002.

[17] S. O. Anderson and S. S. Srinivasa, "Identifying trajectory classes in dynamic tasks," in *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, Apr. 2007, pp. 172–177.