Carnegie Mellon University
Research Showcase

Robotics Institute | School of Computer Science

1-1-2006

Controlling Velocity in Bipedal Walking: A Dynamic Programming Approach

Thijs Mandersloot
Delft University of Technology

Martijn Wisse
Delft University of Technology

Christopher G. Atkeson
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/robotics

Part of the Robotics Commons

Recommended Citation
Mandersloot, Thijs; Wisse, Martijn; and Atkeson, Christopher G., "Controlling Velocity in Bipedal Walking: A Dynamic Programming Approach" (2006). *Robotics Institute.* Paper 70.
http://repository.cmu.edu/robotics/70

This Conference Proceeding is brought to you for free and open access by the School of Computer Science at Research Showcase. It has been accepted for inclusion in Robotics Institute by an authorized administrator of Research Showcase. For more information, please contact research-showcase@andrew.cmu.edu.

Controlling Velocity In Bipedal Walking: A Dynamic Programming Approach

Thijs Mandersloot and Martijn Wisse
Delft University of Technology
Delft, Netherlands
thijs.mandersloot@gmail.com, m.wisse@tudelft.nl

Christopher G. Atkeson
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA
cga@cmu.edu

***Abstract*— We are interested in adding actuation to passive dynamic walkers to enable them to control their velocity. We control velocity by using dynamic programming to design control laws for each desired velocity. We consider three cases: a simulated planar compass gait walker, a simulated 3D compass gait walker with roll dynamics, and a simulated planar compass gait walker with a torso. Each of the walkers have massless legs. The actions include foot placement, ankle torque, and desired torso orientation. We use Poincaré sections to define the state of the model, and thus choose a new action once per footstep. The optimization criterion is based on the effort of swinging the limbs, applying torques, and maintaining the desired velocity. By generating control laws at different desired velocities and then selecting the appropriate control law we are able to control velocity in each of these walkers, and smoothly transition between different velocities. Our results also indicate how complex nonlinear control laws can be approximated by gain-scheduled linear control laws.**

I. Introduction

We are interested in adding actuation to passive dynamic walkers to enable them to control their velocity, place their feet at desired locations, and handle rough terrain [1]. In this paper we control velocity in a simulated planar compass gait walker (Fig. 1) [2], in a simulated 3D compass gait walker with roll dynamics, and in a simulated planar compass gait walker with a torso (Fig. 1). Control actions include step length, step width, ankle torque, and desired torso orientation. Building on our previous work, we replace manually designed controllers with control laws developed using dynamic programming [3]. By choosing the criterion optimized by dynamic programming appropriately, we can design control laws that minimally interfere with the natural dynamics of the biped.

We also build on our previous work in dynamic programming applied to gait control [4]. We further simplify the model to have massless legs, and use the location of the system in a Poincaré section as the state for dynamic programming purposes (Fig. 2) [5]. These simplifications help us avoid the "curse of dimensionality" commonly faced by dynamic programming approaches [6], [7], [8]. For example, for the compass gait walker, the dimensionality of the state is reduced from 4 (stance leg angle and angular velocity and swing leg angle and angular velocity) to 1 (stance leg angular velocity). Repercussions of these simplifications are that we do not take into account the dynamic effect of the swinging leg and we can only choose new control actions once per footstep.

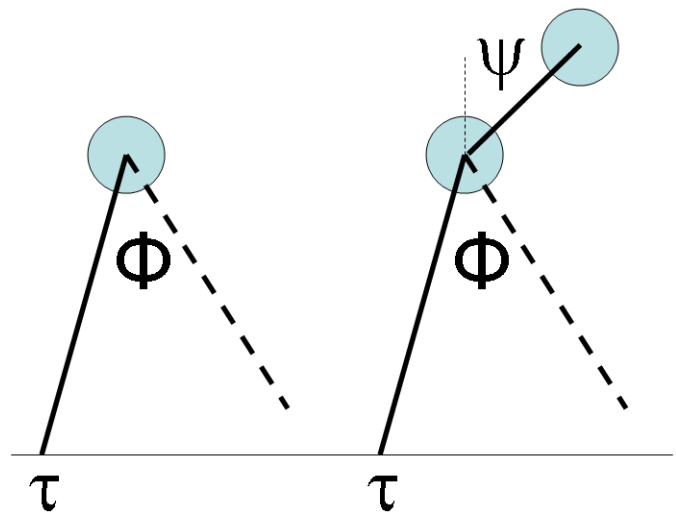

Fig. 1. Planar compass gait walker and planar compass gait walker with torso.

The success of optimal control approaches depends on the choice of optimization criteria. Our optimization criteria are based on a weighted sum of the effort of swinging the limbs, applying torques, and maintaining the desired velocity. Even though the legs are massless, we can estimate the effort to swing legs with mass by taking into account leg accelerations. By generating control laws at different desired velocities and then selecting the appropriate control law we are able to control velocity in each of these walkers, and smoothly transition between different velocities. Our results also indicate how complex nonlinear control laws can be approximated by gain scheduled linear control laws.

II. What Is Dynamic Programming?

Dynamic programming (in discrete time) is a method to solve optimal control problems of the following form: find a control law $\mathbf{u}_k = \mathbf{u}(\mathbf{x}_k)$ for a system with discrete time dynamics $\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ that minimizes $V(\mathbf{x}_0) = \sum_{k=0} c(\mathbf{x}_k, \mathbf{u}_k)$ [6], [7], [8]. $c(\mathbf{x}, \mathbf{u})$ is the one step cost function. In this work we use a straightforward way to solve the dynamic programming problem. We discretize the states and actions, and represent the control laws and value function in tables. We use a procedure called value iteration to refine the control laws and value

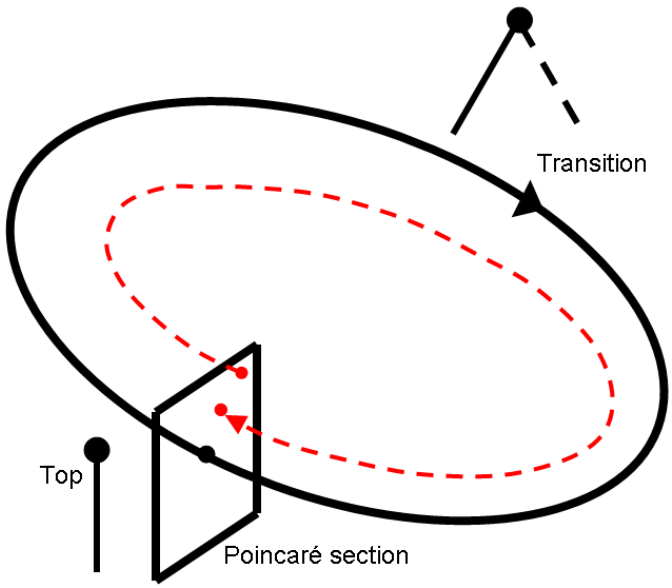

Fig. 2. Gait cycle and Poincaré section. The red dashed line shows how a return map is computed. A trajectory is started from a point on the Poincaré section at TOP, and the intersection of the trajectory with the Poincaré section at the next TOP is recorded.

function, after generating an initial guess $V_0(\mathbf{x})$. Value iteration involves looping through all the states updating the tabulated control law $\mathbf{u}(\mathbf{x})$:

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^* = \min_{\mathbf{u}}(c(\mathbf{x},\mathbf{u}) + V(\mathbf{f}(\mathbf{x},\mathbf{u}))) \quad (1)$$

The minimization in Equation 1 is performed by searching over a discretized set of actions. In this paper we explore all actions. We are developing more sophisticated search algorithms for this minimization. Lookups in the tabulated value function are interpolated using multilinear interpolation. The tabulated value function is updated by substituting in the optimal $\mathbf{u}^*$ into the sum of the one step cost and the value function applied to the optimal next state $\mathbf{x}^* = \mathbf{f}(\mathbf{x},\mathbf{u}^*)$.

$$V(\mathbf{x}) = c(\mathbf{x},\mathbf{u}^*) + V(\mathbf{x}^*) \quad (2)$$

III. Planar Compass Gait Walker

In this section we describe the approach applied to a compass gait walker. The walker has massless straight legs, with all mass located at the hips. The trajectory of the robot is given by the dynamics of a simple inverted pendulum, with a mass of $1kg$, leg length of $1m$, and moment of inertia about the ankle of $1kgm^2$. Gravity is $9.81m/s^2$, and the integration time step for simulation is 1 millisecond.

We use a Poincaré section to define the state of our model (Fig. 2). We place the section when the stance leg is vertical, so the hip is at its highest point (TOP). The state at TOP is just the angular velocity of the stance leg. We will consider angular velocities at TOP between 0 and 3 radians/second, as above 3 radians/second the foot lifts off the ground.

We define two actions that are taken at TOP and held constant through the step cycle until the next TOP. The first action is to choose a hip angle in the forward (sagittal) direction (ϕ in Fig. 1), and thus a landing angle of the swing leg and corresponding step length. The swing leg pitch angle is measured relative to the stance leg. We will refer to this action as the "step angle". The second action is a constant pitching torque at whatever ankle is the stance ankle (the same torque is used before and after the foot transition at impact) (τ in Fig. 1). The torque is necessary to make up for energy losses due to impacts, and is also useful to accelerate and decelerate. We will consider step angles less than one radian and thus leg angles with respect to vertical of less than half a radian, to remain within the friction cone and avoid slipping on landing. We will limit ankle torques to less than 1 Newton-meter in magnitude so that the center of pressure remains within a foot of length 20cm.

The one step cost function penalizes the square of the torques, swing leg acceleration, and deviation from the desired velocity. Because we are using Poincaré states and actions that occur only once per footstep, some of these quantities are approximated. The torque component of the one step cost function is proportional to the square of the ankle torque chosen on each step: τ^2. The swing leg acceleration is approximated by considering the swing angle, ϕ, divided by the step time, T (from TOP to TOP), squared. The one step cost function thus has a component proportional to swing leg acceleration squared: $(\phi/T^2)^2$. The body translational velocity is approximated using $2*l*sin(\phi/2)/T$, with $l = 1m$, so the velocity component of the one step cost function is $(2*l*sin(\phi/2)/T - v)^2$ where v is used to design control laws with different steady state velocities. These penalties are evaluated at each occurrence of TOP. The swing penalty is weighted by 0.1 relative to the other penalties:

$$c(\mathbf{x},\mathbf{u}) = 10*\tau^2 + (\phi/T^2)^2 + 10*(2*l*sin(\phi/2)/T - v)^2 \quad (3)$$

There is a discount factor of 0.99 per step, to allow the value iteration computation to converge.

Fig. 3 shows the step angle control laws for different desired velocities, Fig. 4 shows the ankle torque control laws, and Fig. 5 shows the return maps. Let's first focus on the red dashed curves in each of these figures, which were generated with $v = 2.5m/s$. The red dashed curve in Fig. 3 shows the step angle chosen by the control law at each possible stance leg pitch angular velocity at TOP. The red dashed curve in Fig. 4 shows the ankle torque chosen by the control law at each angular velocity as well. Given these two control laws, we can ask what stance leg pitch angular velocity results at TOP on the next step, given a current stance leg pitch angular velocity at TOP on this step. This is known as the return map, and for $v = 2.5m/s$ is shown by the red dashed line in Fig. 5. To find out what velocity results from the control laws and the resulting return map, we look for a current stance leg pitch angular velocity that results in the same pitch angular velocity on the next step. This can be found by seeing where the return map intersects the diagonal line $x = y$, which in this case occurs at a pitch angular velocity of 1.05 at TOP. If the red dashed control laws in Fig. 3 and 4 are used continuously, the pitch

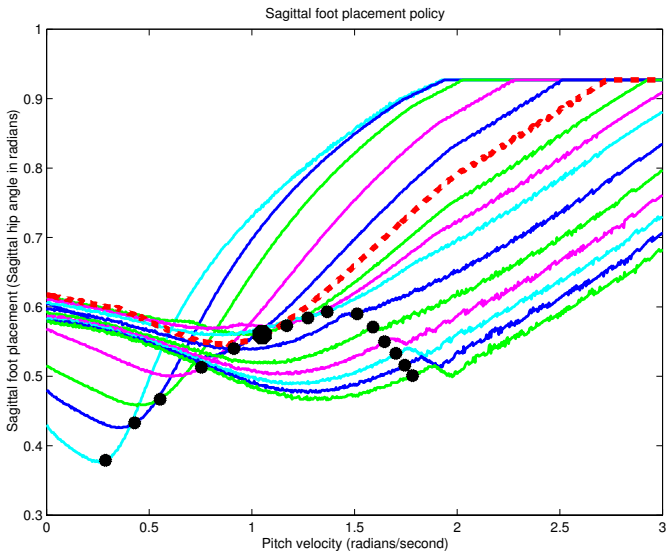

Fig. 3. Step angle for the planar compass gait walker for several desired pitch angular velocities at TOP.

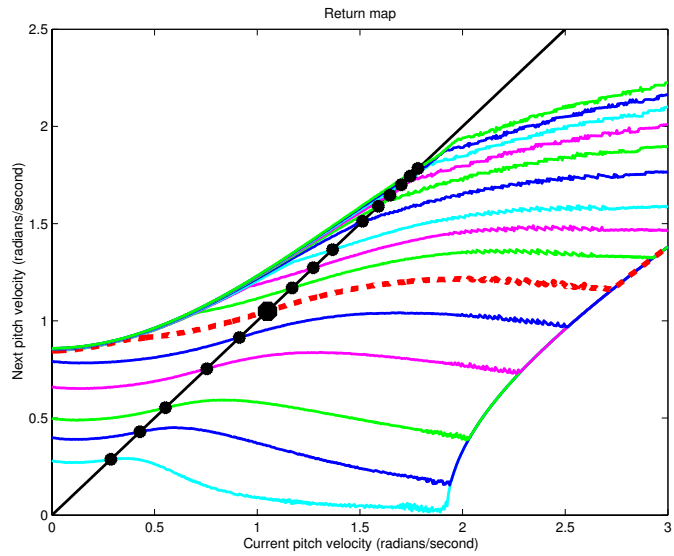

Fig. 5. Return maps for the planar compass gait walker for several desired pitch angular velocities at TOP.

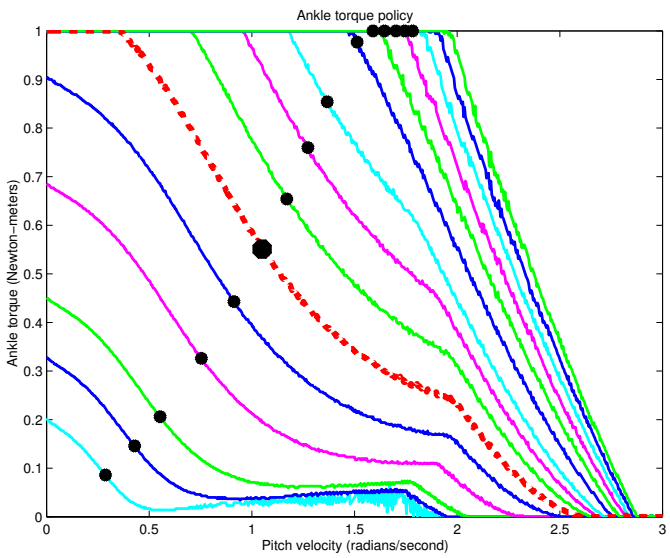

Fig. 4. Ankle torque for the planar compass gait walker for several desired pitch angular velocities at TOP.

angular velocity at TOP equilibrates to this value (shown in the first and last 5 seconds of Fig. 6).

Figures 3 and 4 show control laws generated with many values of v, and Fig. 5 shows the corresponding return maps. The set $v = (0.5,\ 0.75,\ 1,\ 1.5,\ 2,\ 2.5,\ 3,\ 3.5,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10m/s)$ results in pitch angular velocities at TOP of (radians/second): $(0.29, 0.43, 0.55, 0.75, 0.91, 1.05, 1.17, 1.27, 1.37, 1.51, 1.59, 1.65, 1.7, 1.74, 1.78)$. The black dots indicate the equilibrium velocities for each set of control laws and corresponding return maps. We see that the equilibrium step angle increases and then decreases with steady state speed, while the equilibrium ankle torque steadily increases until a limit is reached. These control laws can either be used directly, or fit with parameterized curves. The control laws for step angle are nonlinear, especially at small v. The control laws for ankle torque could be fit by straight lines near the equilibrium ankle torques for each v, if the saturation at $1Nm$ is taken into account. This is an example of how the form of the nonlinear policies created by dynamic programming can be used to suggest appropriate simple parametric control laws to apply.

Fig. 6 shows a simulation where the desired stance leg pitch angular velocity at TOP is changed periodically by selecting appropriate control laws. In each case the desired velocity at TOP is attained within a few steps.

IV. Compass Gait Walker With Roll

We study the compass gait walker with roll to get some insight into control laws for lateral foot placement. This also demonstrates the approach applied to a higher dimensional problem. The planar compass gait walker only rotates in pitch. The compass gait walker with roll is a 3D inverted pendulum and rotates in both pitch (positive is forward) and roll (positive is to the right side). Since the body mass is concentrated at a point, yaw movement has no effect and is ignored. In this version of the compass gait walker we represent body configuration as if there was a universal joint at the ankle with pitch first (inner gimbal) and roll second (outer gimbal). The swing leg has a similar arrangement at the hip, with pitch as the inner gimbal and roll as the outer gimbal.

The parameters of the model are the same as the planar compass gait walker. The walker has massless legs, with all mass located at a point at the hips. The trajectory of the robot is given by the dynamics of a simple inverted pendulum, with a mass, length, and moment of inertia of 1, as before. Gravity is 9.81, and the integration time step is 1 millisecond.

We use a similar Poincaré section to define the state of our model. We place the Poincaré section when the stance leg has zero pitch. The state at TOP is given by the stance leg pitch

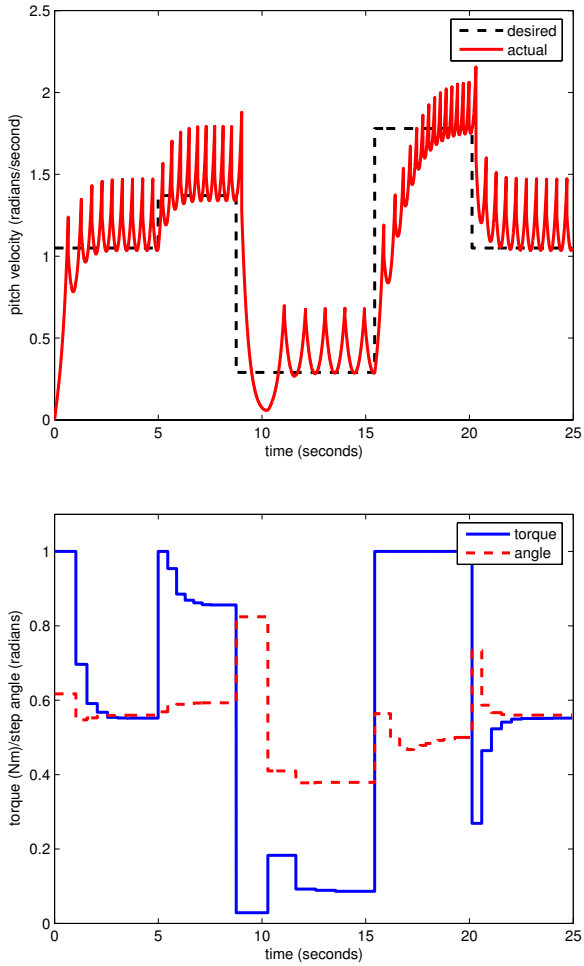

Fig. 6. Simulation results for the planar compass gait walker. Actual (red solid) and desired (black dashed) stance leg pitch angular velocity are plotted in the top graph, and the control signals (step angle (red dashed) and ankle torque (blue solid)) are plotted in the lower graph. Note that in this simulation the compass gait walker started from standing with an initial stance leg pitch angular velocity of zero.

angular velocity, the stance leg roll angle, and the stance leg roll angular velocity. We will consider pitch velocities at TOP between 0 and 3 radians/second, as above 3 the foot lifts off the ground. We consider roll angles less than 0.25 radians, and roll velocities less than 0.5 radians/second, since we expect much less movement in the roll direction.

We add a roll hip angle (lateral foot placement) to the pitch hip angle (sagittal foot placement) and ankle torque actions. We will consider roll hip angles less than 0.5 radians, to avoid slipping on landing. We add a swing penalty to the one step cost function of $(\phi_{roll}/T^2)^2$ on the touchdown hip roll angle that matches the swing penalty on the pitch hip angle to prevent high lateral leg accelerations. Otherwise the one step cost function is the same as the previous case:

$$c(\mathbf{x},\mathbf{u}) = 10*\tau^2+(\phi/T^2)^2+10*(2*l*sin(\phi/2)/T-v)^2 +(\phi_{roll}/T^2)^2 \quad (4)$$

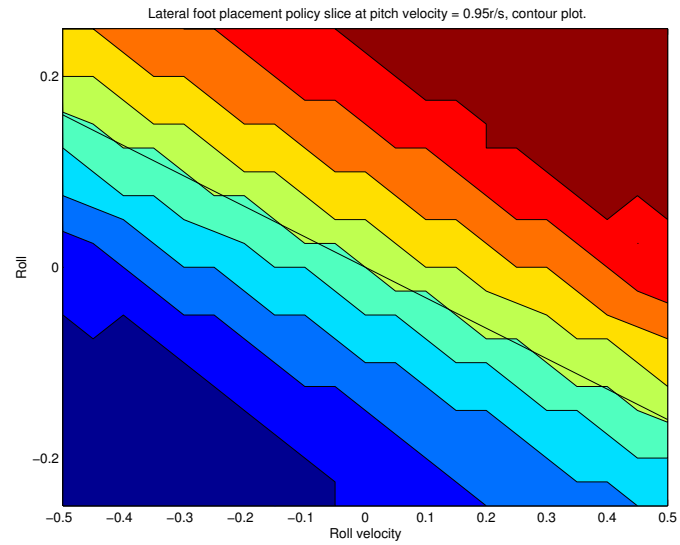

Fig. 7. Contour map of lateral foot placement (hip angle in roll direction) control law slice at pitch velocity = 0.95r/s. The diagonal line is the alpha direction in the bottom plots.

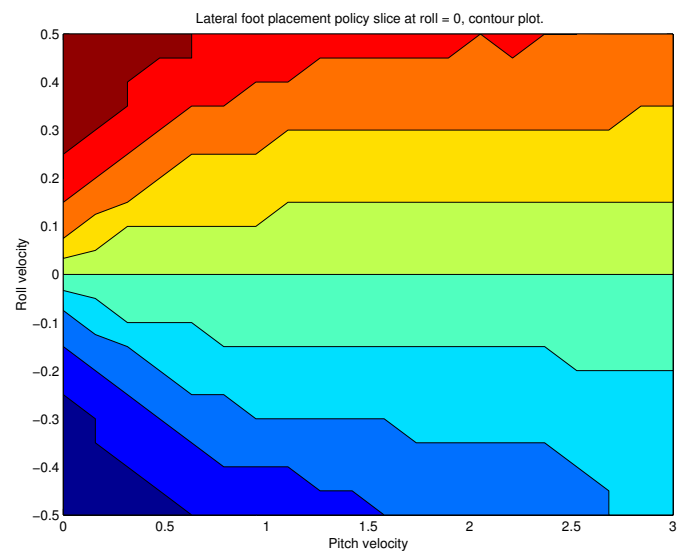

Fig. 8. Contour map of lateral foot placement control law slice at roll = 0.

A useful direction for thinking about roll is to consider roll angles and angular velocities that will result in the inverted pendulum returning to vertical with no actuation. These are given by $velocity = -\alpha * position$ near the equilibrium where the natural behavior of the compass gait is an unstable exponential divergence from the equilibrium. $\alpha = mass * length * g/I$ where I is the moment of inertia about the ankle. In (roll, roll velocity) coordinates the direction $(1,-\alpha)$ takes no effort to return to the equilibrium, and the direction $(\alpha,1)$ requires maximal cost. We will refer to the no effort direction $(1,-\alpha)$ as the α direction.

Figures 7 and 8 show some examples of control laws for lateral foot placement. Since the state vector is three dimensional, (pitch velocity, roll, roll velocity), we cannot plot

the complete control law. Instead we plot a slice by fixing one of the state variables as indicated in the figure captions. In Fig. 7 we see that the lateral foot placement dependence on body roll and roll velocity is almost linear, at a fixed body pitch velocity. A simple linear control law would probably work well in this case. In Fig. 8, we see that the lateral foot placement dependence on body roll velocity is also almost linear, at a fixed body pitch velocity. However, the "gain" of the lateral foot placement dependence on roll velocity increases for pitch velocities below 1r/s. A family of linear control laws selected by pitch velocity would probably work well in this case. Again, the form of the nonlinear policies created by dynamic programming can be used to suggest appropriate simple parametric control laws to apply.

V. Compass Gait Walker With Torso

To explore the effect of a torso on control, we added a torso to the planar compass gait walker (Fig. 1). This adds torso angle and angular velocity to the state, both of which are measured with respect to vertical. The total mass of $1kg$ of the compass gait walker is split into a hip point mass of $0.5kg$ and a torso point mass of $0.5kg$ at height $0.5m$ above the hip. We continue to use the Poincaré section when the hip is at its highest point (TOP). In addition to step angle and ankle torque, hip torque acting between the stance leg and the torso is available. However, choosing a new hip torque once per footstep is not an adequate bandwidth to control the inverted pendulum formed by the torso. Therefore, the control action for the torso is a desired torso angle, and a PD servo controls the torso to that angle with respect to vertical using hip torque (P gain $= 190$, D gain $= 20$). The elements of the control vector are step angle, ankle torque, and desired torso angle (ψ_d). This modification also allows us to reduce the dimensionality of the state. Assuming accurate tracking of desired torso angle, torso angular velocity can be dropped from the state vector, since at TOP the torso angle would have reached its fixed target.

In addition to optimizing ankle torque, swing leg acceleration, and deviation from a desired velocity, the one step cost function includes a penalty on the desired torso angle squared, ψ_d^2. The modified one step cost function is:

$$c(\mathbf{x},\mathbf{u}) = 10*\tau^2+(\phi/T^2)^2+10*(2*l*sin(\phi/2)/T-v)^2 +100*\psi_d^2 \quad (5)$$

Fig. 9 shows a simulation where the desired stance leg pitch angular velocity at TOP is changed periodically by selecting the appropriate control laws. The pattern of desired velocities is similar to that for the compass gait walker with no torso in Fig. 6 and the pattern of commands in Fig. 9 should be compared to those of Fig. 6. In each case the desired velocity at TOP is attained within a few steps. We see that in addition to the control actions used in the previous case, the torso angle is also used. The torso walker takes longer to make a step, so the time window of Fig. 9 is increased. The torso walker also takes longer steps.

Fig. 10 shows the commands used for steady state walking at a variety of speeds under two conditions: using the torso and holding the torso vertical. The case with the torso held vertical is similar to the previous compass gait walker with no torso: ankle torque increases with steady state stance leg pitch angular velocity (black dots in Fig. 4) until it saturates, and step angle first increases and decreases. (black dots in Fig. 3). When the torso angle is used as part of the control, the decrease in step angle at higher values is reduced. The steady state desired torso angle leans more forward with

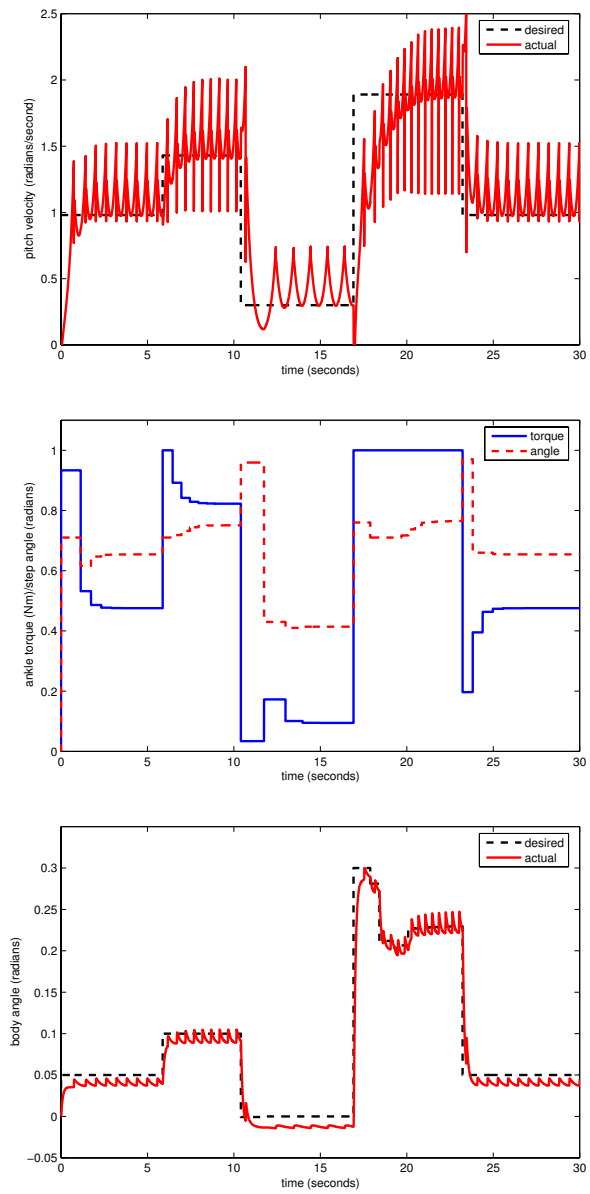

Fig. 9. Simulation results for the planar compass gait walker with a torso. Actual (red solid) and desired (black dashed) stance leg pitch angular velocity are plotted in the top graph, and the control signals (step angle (red dashed) and ankle torque (blue solid)) are plotted in the middle graph. The desired (black dashed) and actual (red solid) torso angles are plotted in the bottom graph. Note that the robot leans forward when it goes faster.

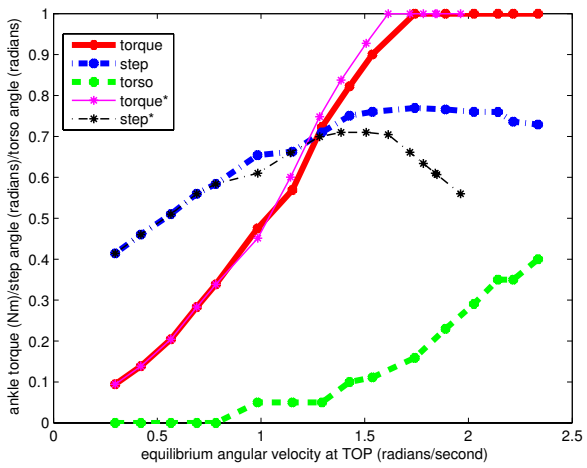

Fig. 10. The commands used for steady state walking at a variety of speeds while controlling the torso (thick lines: ankle torque (red solid), step angle (blue dash-dot), and torso angle (green dashed)) and holding the torso vertical (thin lines: ankle torque (magenta solid), step angle (black dash-dot))

higher velocity. In general the controller for the walker with an adjustable torso can walk faster (maximum stance leg pitch angular velocity of 2.3r/s vs 2r/s for the fixed torso case). Fig. 11 shows an example of how torso angle is used transiently during velocity changes.

VI. Discussion

This paper has presented a method to control velocity by selecting control laws designed by dynamic programming. The approach can handle starting from zero velocity, and can achieve a range of velocities. However, because of the use of Poincaré states, it is not possible to design a control law for stopping. The occurrence of TOP states while stopping can be very intermittent or non-existent, and the ankle torque needs to be controlled on a continuous basis with high bandwidth. This problem can be solved by using states with high temporal resolution. In the case of the compass gait walker the state could be pitch angle and angular velocity, and the dynamics used in dynamic programming could have a time step of milliseconds, rather than the duration of a footstep used in this paper. We are currently developing such a stopping controller.

This work takes the point of view that gait control is based on selecting an appropriate controller, such as the stopping controller, the standing controller, the gait initiation controller, or constant velocity controllers with different equilibrium velocities. Generalizing this to 3D might require a set of turning controllers indexed by turn radius and velocity magnitude. This work also takes the point of view that some variables are controlled on the millisecond time scale (hip torque) and some are controlled on the footstep time scale (footstep location, ankle torque, and desired torso angle). For most variables this makes sense. However, we see that for stopping and standing, ankle torque needs to be controlled on the millisecond time scale. It may be that there is a variable such as *energy-to-be-added* that is controlled on the footstep time scale, and drives lower level controllers that control ankle torque and push off.

This paper has applied dynamic programming to very simplified models with states of up to 3 dimensions. This has allowed us to explore lateral control and the use of a torso, in addition to forward velocity. We were forced to ignore leg dynamics in order to keep the dimensionality low. With the exponential improvement in current computers, it is not unrealistic to apply dynamic programming to higher dimensional problems. However, other local optimization techniques can be used which can handle high dimensionality, instead of a global optimization technique such as dynamic programming. An alternative approach to controlling steady state gait is to linearize the dynamics of the Poincaré state and apply Linear Quadratic Regulator (LQR) design techniques. This approach would allow very high dimensional control laws to be designed, and is currently under development by the authors. Another approach is to use a trajectory optimization approach, such as DIRCOL [9] or Differential Dynamic Programming [10]. Local optimization approaches find a local optimum, rather than a global optimum. The quality of the local optimum typically depends on the starting point for the search. This is not the case for dynamic programming.

There are several features of human walking that we do not address. One is another way to add energy to the gait, such as "push off" at the end of each stance. This could be addressed by adding stance leg length and leg extension velocity to the state. As discussed previously, this would make the dynamic programming solution much more expensive. Another feature of human walking we are missing is double support. This could also be addressed by including stance leg lengths and extension velocities in the state. Low level control or passive properties that make the legs appear to be compliant may allow the robot to achieve natural double support without explicit control [11]. We could also make these additional degrees of freedom functions of other aspects of the state, and thus not have to consider them during control law design [12].

Our implementation of the compass gait biped with roll dynamics is not fully three dimensional. It does not include turning and yaw dynamics, hip spacing, or a distinction between left and right legs. Future work will include these aspects.

The control law design done in this paper is deterministic. We assume that knowledge of imperfect sensors and dynamic models will not change the form of the control laws. It is possible, however, that explicit consideration of uncertainty will cause different control strategies to be adopted. Future work includes considering variations in ground height, limitations in where footfalls can occur, slipping, and tripping, and their effect on control law design. We will also explicitly consider robustness in terms of the volume of possible initial states, the size of possible perturbations, and the variation in model parameters such as robot mass that the control laws can handle.

In work on optimal control, we should always be skeptical of the optimization criterion used. It is merely an imperfect mathematical statement of what is desired. If a simulation

showed walk features that were undesirable, we changed the weights in the optimization criterion. Thus, the optimization criteria presented in this paper are the result of some thought and also some experience and trial and error. In the future, a more exhaustive search of the space of possible optimization criteria may provide insight into what the "right" optimization criteria is.

It was surprising that the control discontinuities due to physical limits did not cause more effects on the control laws. These effects included limited foot size limiting ankle torque and the coefficient of friction limiting step angle. When ankle torque saturates, step size is decreasing (no torso) or staying the same (torso). To some extent torso angle compensates for saturation of ankle torque. Step angle did not saturate for the optimization criteria we used here.

VII. Conclusion

These results show that dynamic programming can be applied to dynamic walking, despite the curse of dimensionality, by considering simple models that include the dynamics of interest, and ignoring other aspects of the dynamics. The form of the computed control laws can be used to suggest appropriate parametric control laws. The next step is to apply these control laws to an actual robot.

Acknowledgment

This material is based upon work supported in part by the DARPA Learning Locomotion Program and the National Science Foundation under NSF Grants CNS-0224419, DGE-0333420, and ECS-0325383. M. Wisse was supported by the NWO, the Netherlands Organization for Scientific Research.

References

[1] S. H. Collins, A. L. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, pp. 1082–1085, 2005.

[2] T. McGeer, "Passive dynamic walking," *International Journal of Robotic Research*, vol. 9, no. 2, pp. 62–82, 1990.

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

[4] M. Stilman, C. G. Atkeson, J. Kuffner, and G. Zeglin, "Dynamic programming in reduced dimensional spaces: Dynamic planning for robust biped locomotion," in *ICRA 2005*, 2005, pp. 2399–2404.

[5] J. Morimoto, J. Nakanishi, G. Endo, G. Cheng, C. G. Atkeson, and G. Zeglin, "Poincare-map-based reinforcement learning for biped walking," in *ICRA 2005*, 2005, pp. 2381–2386.

[6] R. Bellman, *Dynamic Programming*. Dover, 2003.

[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[8] J. Si, A. Barto, W. B. Powell, and D. W. II, *Handbook of Learning and Approximate Dynamic Programming*. IEEE, 2004.

[9] O. von Stryk, "Dircol," 2006, www.sim.informatik.tu-darmstadt.de/sw/dircol/.

[10] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*. Elsevier, New York, NY, 1970.

[11] A. Seyfarth, H. Geyer, S. Lipfert, J. Rummel, Y. Minekawa, and F. Iida, "Running and walking with compliant legs (submitted)," 2006.

[12] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. C. de Wit, and J. W. Grizzle, "Rabbit: A testbed for advanced control theory," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 57–79, 2003.

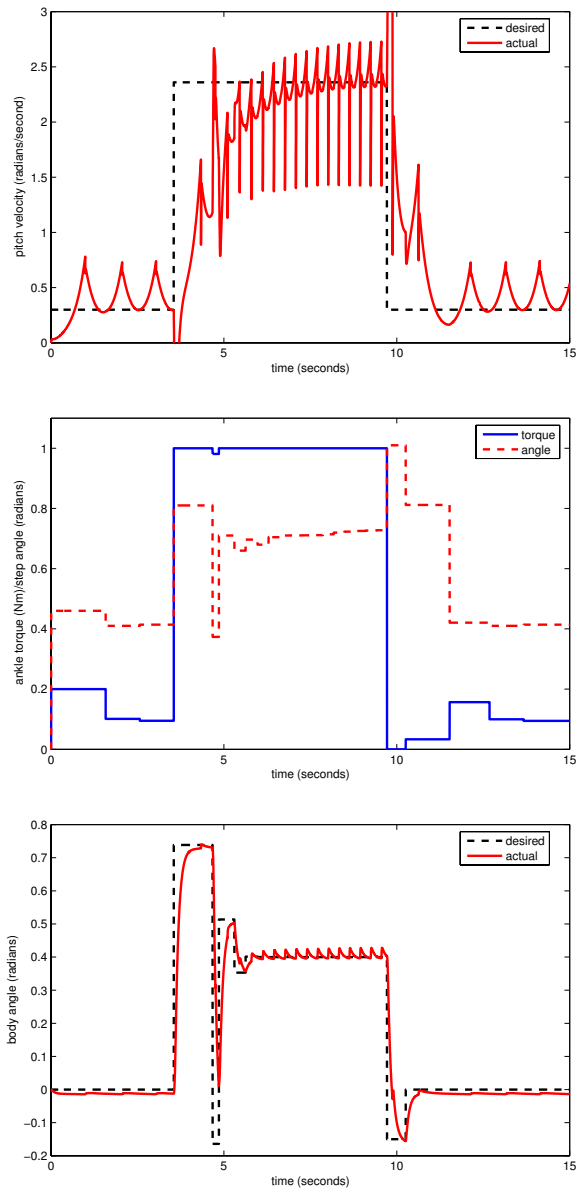

Fig. 11. Simulation results for the planar compass gait walker with a torso for large velocity changes. Actual (red solid) and desired (black dashed) stance leg pitch angular velocity are plotted in the top graph, and the control signals (step angle (red dashed) and ankle torque (blue solid)) are plotted in the middle graph. The desired (black dashed) and actual (red solid) torso angles are plotted in the bottom graph. Note the greater use of torso movement during large velocity changes, as compared to Fig. 9.