# Improving SIMBICON for Biped Animation

Zhiguo Zhang

School of Computer Science,
McGill University,
Montreal, Canada
zhiguo.zhang@mail.mcgill.ca

***Abstract*—Biped balance control strategy is difficult to design as biped locomotion suffers from high-dimensional action spaces, underactuated dynamics, and unstable joint controls. We propose a generalized balance control strategy that can synthesize robust biped walking locomotion. The generalization consists of two parts, a coordinated joint control and generalized ground height feedback. We also conduct principle component analysis on generated locomotion to further improve the state used in finite state machine. Systems employing proportional-derivative controllers face the trade-off between simulation efficiency and effectiveness. We explore and exploit a stable variant of proportional-derivative controller to enhance the simulation process. Our main contributions are improving the robustness of biped control against upward steps in real time, and generating more torque-efficient and smoother simulations.**

## I. Introduction

Locomotion is an essential part of character animations and movie creation. This is often achieved by using motion captured data or key frame animators. However, it does not scale well in more realistic situations. For instance, it is hardly seen that two people walk in the exactly same way on an unexpected terrain environment. As a result, algorithmic approaches is more capable of generating a set of motions instead of individual motions. One way to solve this problem is to develop a physical-based control strategy that generates different poses and gaits in real time. In this project, we focus on biped locomotion as it is useful to animate human-like creatures in either video games or animated movies. More specifically, biped locomotion can be difficult to produce due to its instability, large number of degree of freedom, and high-dimensional dynamic system.

The general goal of this project is to explore some modifications to the previous SIMBICON controller [1]. Unlike passive physical-based simulation of cloth or deformable objects, character control is active. SIMBICON actively controls the swing-hip angle to maintain balance. We generalize the linear feedback strategy to all joints instead of only swing hip, as human beings actively control all joints to walk. Although SIMBICON is very robust, it fails when the terrain contains unanticipated upward steps. We address this issue by introducing ground height feedback to the generalized linear feedback system. This improvement demonstrates the usefulness of external information in motion control. The controller no longer passively adapts to environment changes, but actively reacts to sensory information.

Besides the enhancement of controller performance, we investigate the characteristics of the motion generated by SIMBICON controller. Principle component analysis (PCA) is a well-known technique for data processing, model analysis and dimensionality reduction. We apply PCA on data simulated by SIMBICON. The eigenvector basis of PCA shows some interesting poses that can be used in kinematic animation or FSM state design.

Lastly, we explores the possibility of replacing conventional proportional-derivative (PD) controllers with stable proportional-derivative (SPD) controllers [2]. Stable proportional-derivative controllers can provide stable simulation in the case of large controller gains and large simulation time steps. We also show that SPD can produce smooth and torque efficient simulations.

## II. Related Work

Many methods have been proposed to create realistic and physically valid motion in Computer Graphics literatures. Optimization methods are an appealing way to synthesize realistic motion. However, the dimensionality of search space makes optimizers hard to solve the problem for complex characters such as humans. Dimensionality reduction based on existing motion capture database was introduced by Safonova et al. [3]. Typical dimension reduction technique such as PCA was employed in their experiments. They showed that representations with five to ten dimensions can create animated motion with very small errors.

Later, Yin et al. developed SIMple BIped CONtrol (SIMBICON) strategy to animate biped characters [1]. The controller consists of four parts. The first is a finite state machine with states of poses driven by a PD controller. The second part is virtual PD control on torso and swing hip. The third part is balance feedback to maintain balance. The last is feedback error learning that reduces the controller gains in simulation. SIMBICON shows its robustness in terms of unexpected downward steps, terrain slope changes and push forces in all direction. Coros et al. developed generalized biped walking control inspired by SIMBICON and other related work such as Jacobian transpose [4] and inverted pendulum [5]. They demonstrated the generalization across different gait parameters, style, characters, and tasks. In the vein of feedback policies, Ding et al. proposed a method for searching low-dimensional linear feedback policies in replacement of manually designed feedback strategies [6]. In contrast to online methods, sample based control strategy shows success in offline biped control [7]. Sample based optimization does not require an explicit derivative computation comparing with gradient based approaches.

Modal analysis can be useful in character animation. Kry et al. applied modal analysis based on the physical information of virtual animals [8]. Modal vibrations are obtained by solving the generalized eigenvalue problem. They showed that a locomotion controller can be constructed from a small number of modal vibrations with low frequencies either manually or automatically with a set of heuristics. Using modal vibrations alone can not generate robust kinematic locomotion due to lack of balance control. Jain and Liu further explored the problem in modal space and formulated the problem into long-horizon planning [9]. They identified the two main advantages of modal analysis: Independent control and Model reduction. Nunes et al. demonstrated that robust locomotion based on modal vibrations can be achieved by an addition of optimization routine [10].

With tremendous development of machine learning techniques in recent years, complex locomotion problems can be solved with reinforcement learning on action space. Peng et al. employed reinforcement learning [11] and deep reinforcement learning [12] to create 2D locomotion that is robust against gaps, walls, and steps. Later, two level hierarchical deep reinforcement learning was adopted to train dynamic locomotion skills including following trails, navigation through various obstacles, and dribbling a soccer ball [13].

Simulation stability and efficiency is another important topic in character animations. In the context of forward simulation, penalty methods can enforce constraints by virtual penalty forces based on the deviation from constraints [14]. Similar to penalty methods, PD controllers use a virtual force to reduce the deviation of current state from target state. However, penalty springs using high gains to precisely maintain the constraints introduces numerical instability [15]. High-gain PD controllers suffer from numerical instability as well. Several techniques have been proposed to improve PD controllers. Yin et al. [16] modeled muscle forces as a combination of feed forward forces based on motion captured data and low gain feedback control. Weinstein et al. [17] proposed a novel PD variation that decouples each degree of freedom and calculates analytical solutions regardless of global effects. Tan et al. [2] presented a stable PD formulation that uses deviations in the next time step.

## III. METHODS

### A. SIMBICON Review

As SIMBICON is the cornerstone of this project, we review its methods and components first. The main control strategy consists of three parts: a finite state machine, a control of torso and swing hip, and a balance feedback term.

SIMBICON starts with a finite state machine or pose graph, where the poses are represented by the target angles with respect to parent joins. Transitions happen after a fixed duration of time or after a new foot contact. PD controllers, $\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}$, drive each joint toward to their target angles.

Without notion of balance, the pose graph cannot produce a robust locomotion. Some modifications are added to ensure robustness in terms of unexpected changes in terrains and perturbations by forward and backward push on torso. The torso and swing hip have targeted angles expressed with respect to world frame instead of their parent joints. This helps control the orientation of torso in world frame. A virtual PD controller working in world coordinates computes the net torso torque $\tau_{torso}$. Moreover, the swing foot positioning is decoupled from the current torso pitch angle. This is achieved by a swing-hip controller works in world frame to calculate swing-hip torque $\tau_B$. Since only internal torques are allowed, the stance-hip torque, $\tau_A$, is treated as free variable, i.e. $\tau_A = -\tau_{torso} - \tau_B$.

In addition, a balance feedback term continuously modifies the swing hip target angle as a linear function of the center of mass position and velocity. The balance feedback law is described as the following:

$$\Delta\theta_d = c_d d + c_v v, \tag{1}$$

where $\Delta\theta_d = \theta_d - \theta_{d0}$ is the difference between the target angle, $\theta_d$, used by PD controllers and the fixed target angle, $\theta_{d0}$, specified by the state machine, $d$ is the horizontal distance from the stance ankle to the center of mass (COM) and $v$ is the velocity of the center of mass. The coefficient $c_d$ and $c_v$ are called "balance feedback gain" parameters, which are essential to maintain balance during low-speed gaits or in-place stepping. The balance feedback gain parameters are manually designed based on observations (details can be found in Section 4 of the SIMBICON paper [1]).

The balance feedback in SIMBICON is an example of general linear feedback described by Ding et al. [6], and can be written

$$\delta a = M_F \cdot \delta s, \tag{2}$$

where $\delta a = a - \hat{a}$, $\delta s = s - \hat{s}$, and $M_F$ is a $m \times n$ full-order feedback matrix. This assumes the existence of reference control actions, $\hat{a}$, and reference sensory observations, $\hat{s}$. In SIMBICON example, controlling the joint angle is the action, while the position of stance foot is reference sensory observation and the COM position is current sensory observation. The reference for velocity is 0 for a balanced biped. The balance feedback gains form the feedback matrix, $M_F = [\, c_d \quad c_v \,]$.

### B. Ground Height Feedback

We observe that with only COM and velocity feedback terms, SIMBICON can fail with unexpected upward steps. Biped characters usually increase the swing hip angle if an upward step is anticipated. This requires ground height changes as an extra sensory feedback. Therefore, we define the ground height change, $h$, as the maximum ground height change from the stance foot to the position of expected swing foot placement. The reason to take maximum instead of direct

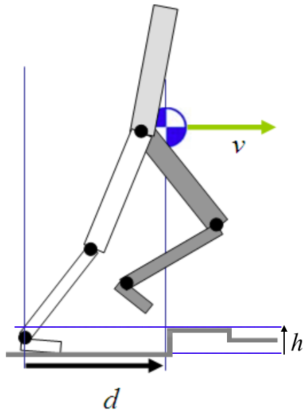


Fig. 1. The three components in generalized balance feedback law in the Equation 4. This figure is a modified version of Figure 3(b) in SIMBICON paper [1]

difference is to avoid small walls in between stance foot and swing foot, in this case the direct difference is 0. Suppose that the stance foot is at position $(x_0, y_0)$, and the swing foot is expected to be placed at $(x_1, y_1)$, then $h$ is obtained by

$$h = \max_{t\in[0,1]}(H(x_0 + \delta xt, y_0 + \delta yt) - H(x_0, y_0)), \quad (3)$$

where $H(x, y)$ is the ground height function, $\delta x = x_1 - x_0$, and $\delta y = y_1 - y_0$. The feedback law in the Equation 1 therefore becomes

$$\Delta\theta_d = c_d d + c_v v + c_h h, \quad (4)$$

where $c_h$ is the gain on ground height feedback. Notice that $h$ satisfies the sensory observation definition, $\delta s$, in the Equation 2 as it simplifies to $h = \max H(t) - H(x_0, y_0)$.

In general, we observe that the step sizes of SIMBICON controllers with the same set of gain parameters are very similar. Hence, the distance from $(x_0, y_0)$ to $(x_1, y_1)$ can be estimated by averaging over all step sizes generated by the same controller on a flat terrain. The direction from $(x_0, y_0)$ to $(x_1, y_1)$ is the same as the direction of current velocity.

### *C. Principle Component Analysis*

Principle component analysis (PCA) is a statistical technique to find a set of linearly independent variables called principle components from a set of observations of variables that are possibly correlated. The first component in PCA has the largest variance in the data set. The second component shows the second largest variance and so on. The principle components can be derived by singular value decomposition on centralized data matrix, eigenvalue decomposition on sample covariance matrix, or iterative methods (normally used for very high-dimensional sample space).

For high-dimensional joint control, PCA is a practical procedure to reduce dimensionality by projection onto optimal subspace, which is spanned by selected principle components based on their eigenvalues. Since principle components show the variance difference in their directions, each principle components can be visualized and used for FSM target angle construction.

Given a motion $M$, we can apply PCA to find the basis of this motion. The motion $M$ has a representation in the form of $M = M(t) = \{p(t), Q(t)\}$, where $p(t)$ is the root position, and $Q(t) = \{q_1(t) \ldots q_n(t)\}$ denotes all the joint angles at time $t$. Hence, a motion $M$ is a set of points in an $(n+1)$-dimensional space. Since the position is not correlated to any joint angles, we only apply PCA on the joint angles. PCA can be computed either by iterative methods or direct covariance eigenvalue decomposition. We use the later as the multiplication in covariance computation does not generate numerical instability based on our experiments.

Consider $m$ sample frames of a motion $M$, the first step is to centralize the sample data by the sample mean $Q_m = \sum_{i=1}^{m} Q_s(t_i)/m$. Therefore, for each $Q_s(t)$, we have $Q_c(t) = Q_s(t) - Q_m$. Then, we apply eigenvalue decomposition on the sample covariance matrix of $Q_c(t)$. Since covariance matrix is symmetric, the eigenvector matrix is orthogonal. As a result, we can construct a $d$-dimensional subspace spanned by the eigenvectors, $\{B_1 \ldots B_d\}$. In general, $d$ is estimated by a standard heuristic [18], in which for eigenvalues, $\lambda_1 > \lambda_2 > \ldots > \lambda_n$, $d$ is the smallest number such that

$$\frac{\sum_{i=1}^{d}\lambda_i}{\sum_{i=1}^{n}\lambda_i} \geq 0.9. \quad (5)$$

The approximation of $Q(t)$ is a linear combination of eigenvectors,

$$\hat{Q}(t) = Q_m + A_1(t)B_1 + \ldots + A_d(t)B_d, \quad (6)$$

where $\{A_1(t) \ldots A_d(t)\}$ are scalar coefficients. This approximation also minimize the total squared error, $\sum_{i=1}^{m}(Q_s(t_i) - \hat{Q}(t_i))^2$ [19].

The coefficients in the Equation 6 can be found by solving the optimization problem with an objective function,

$$G(M) = w_T G_T(M) + w_A G_A(M) + w_P G_P(M), \quad (7)$$

which is a weighted sum of three components [3]. The $G_T(M)$ part minimizes the sum of squared torques. The $G_A(M)$ minimizes the sum of squared joint accelerations and sum of squared root accelerations. Minimization of $G_A(M)$ enhances the smoothness of motion trajectory. The $G_P(M)$ component penalizes the deviation of coefficient $A_i(t)$ from zero in inverse proportion to its eigenvalue.

### *D. Stable Proportional-Derivative Controller*

The SIMBICON uses conventional proportional-derivative controller of the form,

$$\tau = k_p(\theta_d - \theta) - k_d\dot{\theta}, \quad (8)$$

for torso pitch and joint angles. If the controller needs to drive the joint towards the target angle rapidly, it must use a large $k_p$. In this case, numerical instability issue arise if the time step is not small enough. This traditional proportional-derivative controller has to consider the trade-off between large proportional gains and simulation efficiency. An improved proportional-derivative controller formulation, called stable proportional-derivative (SPD) controllers, was proposed by Tan et al. [2]. The idea is inspired by fully implicit integrator described by Baraff and Witkin [20]. The SPD controllers compute the current torque using the position and velocity in the next state. Suppose that $\theta^n$ is the joint angle in current state at time $t = n$, the Equation 8 can be written as:

$$\tau^n = k_p(\theta_d^n - \theta^n) - k_d\dot{\theta}^n.$$

Then, SPD formulation is expressed as

$$\tau^n = k_p(\theta_d^{n+1} - \theta^{n+1}) - k_d\dot{\theta}^{n+1}. \tag{9}$$

Since the position and velocity for next state is unknown, $\theta^{n+1}$ and $\dot{\theta}^{n+1}$ are approximated by the first-order Taylor expansion around $\theta^n$ and $\dot{\theta}^n$:

$$\begin{bmatrix} \theta^{n+1} \\ \dot{\theta}^{n+1} \end{bmatrix} = \begin{bmatrix} \theta^{n} \\ \dot{\theta}^{n} \end{bmatrix} + \Delta t \begin{bmatrix} \dot{\theta}^{n} \\ \ddot{\theta}^{n} \end{bmatrix}.$$

By substituting $\theta^{n+1}$ and $\dot{\theta}^{n+1}$, the Equation 9 becomes

$$\tau^n = k_p(\theta_d^{n+1} - \theta^n - \Delta t\dot{\theta}^n) - k_d(\dot{\theta}^n + \Delta t\ddot{\theta}^n),$$

where the target angle for next state is stored in FSM. SPD shows more stability and robustness when the simulation requires large controller gains or large time steps. More details are shown in next section.

## IV. Experimental Results

In this section, some modifications are added to the SIMBICON controller. The experiments start with the 2D biped controller (detailed specification of the model can be found in section 7 of the SIMBICON paper [1]).

### A. Generalized Balance Feedback

The balance feedback law in the Equation 1 can be generalized to all joints instead of only swing hip. Yin et al. [1] mention that balance feedback strategy on multiple joints has the form of

$$\boldsymbol{\theta_d} = \boldsymbol{\theta_{d0}} + \mathbf{F} \begin{bmatrix} d \\ v \end{bmatrix}, \tag{10}$$

where $\mathbf{F}$ is an $n \times 2$ matrix with balance feedback gain coefficients for $n$ joints. However, in the their implementation and provided parameters, only swing-hip balance feedback is employed. Therefore, generalized balance feedback on all joints except stance hip (as it is considered to be a free variable in torso and swing-hip control) is implemented and simulated.

The stable range on each joint is quite different. The balance feedback gains on both ankles do not influence the result of simulation significantly. Large positive or negative gains, for example, in range of $[-20, 20]$, fail to cause biped to lose control, instead only make the feet movement unstable. Nevertheless, the stance knee is very sensitive to the balance feedback gains. Adding a $[-0.5, 0.5]$ gain on either COM or velocity can easily cause loss of control. The reason is that in FSM, the stance knee is driven to be straight, i.e., stance knee has a $-0.05$ rad angle with its parent joint. With the positive additional feedback onto stance knee, the biped starts accumulate velocity until it rolls over. Lastly, the most interesting joint to have a balance feedback is the swing knee. The stable range of gains is proportional to the magnitude of target angle in FSM.

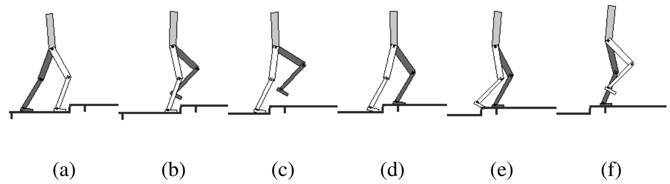

Fig. 2. An example of generalized balance feedback that can provide robustness against unanticipated upward steps. The height of upward step is 10 cm.

With the addition of swing-knee balance feedback, more gaits and locomotion can be created. One example is walking with slightly high step but still maintaining balance without bending the swing knee too much to counter the torque from high step. Yin et al. show so-called "*highstep walk*" [1]. The locomotion is unnatural in the sense that the swing knee bends backward too much. This new walking gaits can be robust against unexpected upward steps of 10 cm, which all the original SIMBICON controllers in 2D fail to be (based on our tests using data provided in paper [1]). Fig. 2 shows a robust walk on an upward step.

Based on our experiments, we noticed that the balance feedback coefficients cannot be designed alone without the consideration of target angles in FSM and the purpose of some angles in reality. For instance, slightly high-step gait is useful for climbing upward steps, it also raises the COM and results in less robustness against forward or backward pushes. Comparing with the normal *walk* locomotion, the threshold for forward pushes decreases from 600 N to around 200 N, while for backward pushes it is reduced from 500 N to around 100 N.

### B. Ground Height Feedback

With the addition of ground height feedback, the generalized balance feedback in the Equation 10 is updated to

$$\boldsymbol{\theta_d} = \boldsymbol{\theta_{d0}} + \mathbf{F} \begin{bmatrix} d \\ v \\ h \end{bmatrix}.$$

To demonstrate the robustness with ground height feedback, we implement the ground height feedback and retain the *walk* controller gains in SIMBICON [1]. The average step size of *walk* controller is $0.55$ m. That is the controller can detect ground height changes $0.55$ m in front of its stance foot (or behind, depending on the direction of current velocity). In our experiment, swing hip, swing knee, and stance knee have non-zero ground height feedback gains. The new locomotion is similar with SIMBICON walk controller on flat terrain as they use the same $c_v$ and $c_d$. Moreover, the new controller maintains the same robustness against push forces. However,

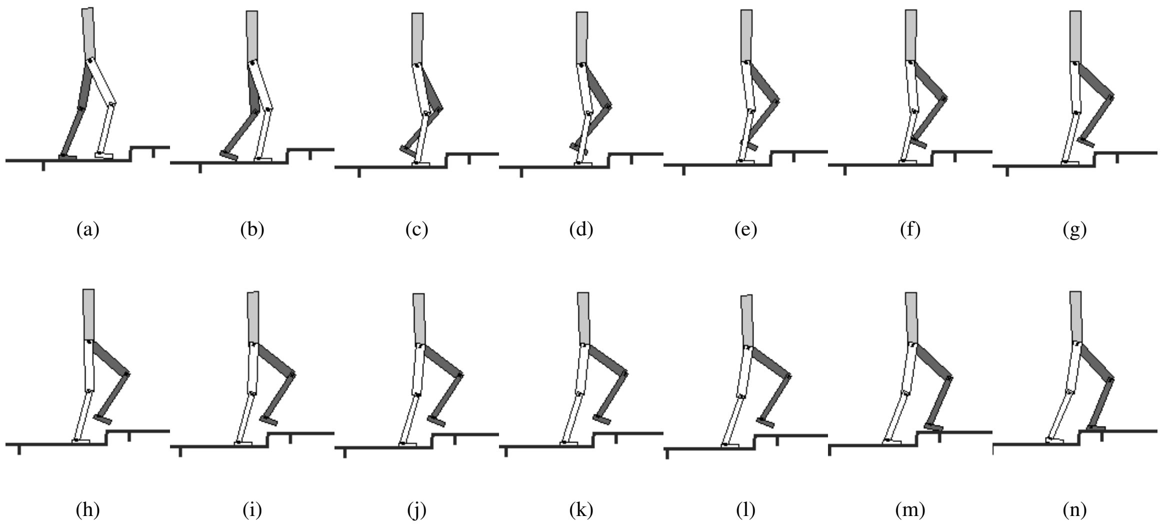


Fig. 3. An example of generalized balance feedback with ground height. The height of upward step is 15 cm.

controller with ground height feedback can robustly walk on a terrain with random steps of $\pm 15$ cm. This result is better than any generalized balance feedback controller in previous section. Fig. 3 shows an example of controller with ground height feedback. We can see that, in comparison with Fig. 2, stance knee keeps straight when swing hip increases its torque. The maximum swing hip torque is also reduced, which causes less torso pitch angle. As a result, the new locomotion looks more natural.

In our experiments, we notice that linear ground height feedback only works for a small range of height changes ( that is $\pm 15$ cm). In order to successfully walk on a higher step, linear model is not sufficient. We propose and implement an non-linear feedback strategy that allows our biped to walk on a $+20$ cm step. The new feedback law is as following:

$$\Delta\theta_d = c_d d + c_v v + c_h f(h) h,$$

where $f(h)$ is defined as 1.0 for $h \leq 0.15$, and 1.5 for $h > 0.15$. For a even bigger upward step, due to the stiffness limit of PD controller and joint limits of biped model, a physically viable locomotion cannot be produced. Although the function $f(h)$ is designed manually and empirically without a mathematical proof, it shows a new way to design feedback strategies that are not linear.

One limitation of ground height feedback is when swing foot is placed on the edge of an upward step. It results in unstable simulation, and biped may or may not recover from the unstable state. One way to address this issue is first computing the foot placement by forward kinematics, and then using inverse kinematics to adjust the swing-hip and swing-knee angles to avoid step edges. We can simplify the swing leg into a model of two joints, that is ignoring the ankle. Then, we can form an analytical solution using simple trigonometry. However, for a complex model without this simplification, an numerical solution is needed.

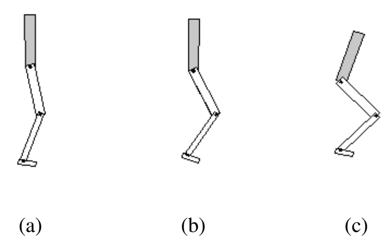


Fig. 4. Average poses for (a) *walk*, (b) *fast walk*, and (c) *crouch walk* controllers.

### C. Principle Component Analysis

For principle component analysis, we generate $10\,000$ sample frames from the *walk* SIMBICON controller [1]. Then, we computed the eigenvalue decomposition of the sample covariance matrix. The eigenvectors are ranked based on their eigenvalues. The data centering step in covariance calculation also shows the average pose for each controller. As shown in Fig. 4, the average poses for *walk*, *crouch walk*, and *fast walk* controllers appear very similarly.

We notice that first three eigenvalues are significantly larger than the other eigenvalues. However, with only the first two eigenvalues, the heuristic in the Equation 5 is satisfied. Eigenvectors can be visualized by computing $\hat{Q}_i = Q_m + B_i$. Eigenvectors with large eigenvalues show some plausible frames of motion. For example, as shown in Fig. 5, (a) can represent a stance state of walking, (b) can be a frame of running, and (c) shows a frame of jumping. Nevertheless, the other eigenvectors do not generate anything meaningful. The detailed PCA results can be found online[1]. Similar to the ideas in the work of Kry et al. [8] for kinematic animation, interesting locomotion can be created based on selected eigenvectors in a sinusoidal fashion.

[1] https://github.com/FrankZhang427/FrankProjectResults

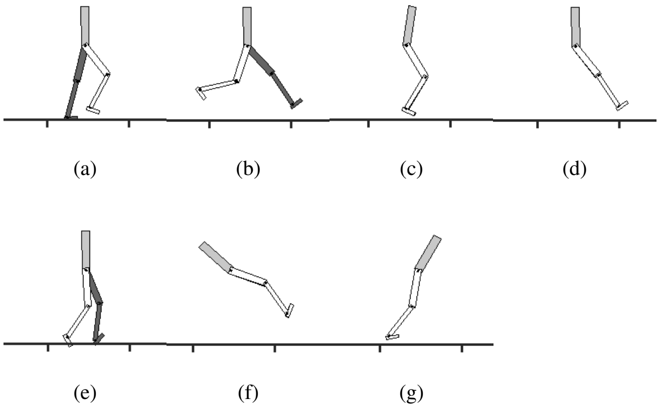


Fig. 5. Visualized eigenvectors for *walk* controller. (a) to (g) are ordered from the largest eigenvalue to the smallest.

For all SIMBICON controllers, we observe that the first principle component is very similar to the target pose of stance phase, that is, the state 0 and 2. Therefore, we replace the state 0 and 2 with the visualized joint angles of the first eigenvector, $\hat{Q}_1$, for instance, Fig. 5 (a) is used for *walk* controller. Notice that for each principle component, there is a complement vector that represents the symmetry of the visualized eigenvector. Generally, the eigenvector corresponds to the state 0 and the complement vector corresponds to the state 2. The modified *walk* and *fast walk* controller perform very similar to the original *walk* and *fast walk* controller, while the modified *crouch walk* controller fails catastrophically at start. The reason of this failure is due to the absence of specifically-designed transition motions in SIMBICON.

### D. Stable Proportional-Derivative Controller

PD controllers in SIMBICON can be replaced by SPD controllers without changing the underlying control strategies or simulation mechanisms [2]. We replace all the PD controllers including virtual controller for torso and swing hip and joint controllers with SPD version. Since SIMBICON uses the penalty method to enforce joint torque constraints, the joint torque limits are implemented in SPD fashion. That is, if $k_{pL}$ and $k_{dL}$ are the maximal allowed $k_p$ and $k_d$ respectively, we have

$$\tau = \begin{cases} k_{pL}(\theta_{\min} - \theta^{n+1}) - k_{dL}\dot{\theta}^{n+1}, & \text{for } \theta^{n+1} < \theta_{\min} \\ k_{pL}(\theta_{\max} - \theta^{n+1}) - k_{dL}\dot{\theta}^{n+1}, & \text{for } \theta^{n+1} > \theta_{\max} \end{cases},$$

where $\theta^{n+1} = \theta^n + \Delta t\dot{\theta}^n$ and $\dot{\theta}^{n+1} = \dot{\theta}^n + \Delta t\ddot{\theta}^n$. This modification mostly prevents each joint from reaching its minimal or maximal limits before the joint actually reaches the limits.

At the original controller gains, $k_p = 300$ and $k_d = 30$, the time step can increase from $0.00027$ s to $0.00038$ s without crashing. In general, SPD is stable around $0.00035$ s for reasonable gain parameters, where $k_p$ is in the range of $[200, 600]$ and $k_d$ is in the range of $[20, 60]$. In contrast, conventional PD requires a smaller time step of $0.0002$ s for large gains. We apply binary search on different time step, $\Delta t$, to find the maximal time step for a fixed proportional gain, $k_p$. Fig. 6 shows the maximal stable time steps versus proportional gains. In this comparison, we set $k_d = 0.1k_p$. Nonetheless, as long as we ensure $k_d \geq k_p\Delta t$, the simulation is stable [2].

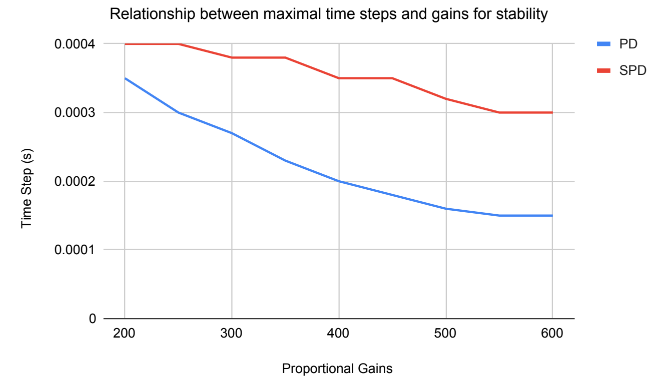


Fig. 6. The relationship between maximal time steps and proportional gains.

TABLE I
JOINT TORQUE NORM SQUARED AVERAGED OVER TIME FOR 100 THOUSAND TIME STEPS, WITH $k_p = 350$ AND $k_d = 35$.

| terrain type | flat | | downward | |
|---|---|---|---|---|
| gaits | SPD | PD | SPD | PD |
| *walk* | 19211.824 | 19257.502 | 28891.479 | 28956.936 |
| *fast walk* | 36158.094 | 36257.850 | 38877.630 | 38981.566 |
| *crouch walk* | 65352.285 | 65434.848 | 74499.780 | 74638.500 |

Inspired by the objective function in the Equation 7 of the optimization in the work of Safonova et al. [3], we can compute the joint torque norm squared and joint acceleration norm squared averaged over time to compare SPD and PD in terms of motion efficiency and smoothness. We sample $100\,000$ iterations for each style and each terrain type from SPD and PD controllers. The parallel comparison shows that SPD controllers have slightly lower averaged joint torques and accelerations, as shown in Table I and Table II. That is SPD controllers drive each joint to its target angle a little more efficiently and smoothly than conventional PD controllers.

## V. IMPLEMENTATION

The implementation in this project is developed on 2D SIMBICON code in Java. The original code can be found at SIMBICON project webpage[2]. The graphical interface is implemented in Java Applet. Some modifications are added for easier usage, including extra buttons for recording and running PCA, and extra keybindings for new terrains and new controllers. Additional gaits described in SIMBICON paper [1] are also implemented. We use JAMA[3], a linear algebra pack-

[2]https://www.cs.ubc.ca/~van/papers/Simbicon.htm
[3]https://math.nist.gov/javanumerics/jama/

TABLE II
JOINT ACCELERATION NORM SQUARED AVERAGED OVER TIME FOR 100 THOUSAND TIME STEPS, WITH $k_p = 350$ AND $k_d = 35$.

| terrain type | flat | | downward | |
|---|---|---|---|---|
| gaits | SPD | PD | SPD | PD |
| *walk* | 933125.6 | 994656.6 | 2499578.8 | 2724502.5 |
| *fast walk* | 1910214.4 | 2019986.5 | 1954188.0 | 2069604.6 |
| *crouch walk* | 10424819 | 11157453 | 7612047.0 | 8264317.5 |

age for Java [21], to facilitate matrix decomposition in this project. The project source code can be found online[1].

## VI. Conclusion

The SIMBICON paper presents a simple yet robust way of biped simulation. The use of FSM provides a good platform for developing more sophisticated strategies as shown in many other related works. In this project, we extend some of the ideas in the SIMBICON paper. The generalization of linear balance feedback makes it possible to create locomotion of more gaits for a variety of tasks. We generalize not only the control to all joints, but linear balance feedback to utilize more sensory information. Furthermore, stable proportional-derivative controller improves the simulation in terms of both simulation efficiency and appearance. We also apply PCA on SIMBICON locomotion. The results of PCA can be useful to generate more realistic states for FSM, or even animations with proper animators.


## Acknowledgements

I am very grateful to Professor Paul G. Kry, my research supervisor, for patient guidance, invaluable critiques, and professional insight during the course of this project. I would also thank Kaixiang Xie for discussions in the early stage of this project.


## References


[1] KangKang Yin, Kevin Loken, and Michiel van de Panne. Simbicon: Simple biped locomotion control. *ACM Trans. Graph.*, 26(3), July 2007.

[2] J. Tan, K. Liu, and G. Turk. Stable proportional-derivative controllers. *IEEE Computer Graphics and Applications*, 31(4):34–44, July 2011.

[3] Alla Safonova, Jessica K. Hodgins, and Nancy S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *ACM Trans. Graph.*, 23(3):514–521, August 2004.

[4] C. Sunada, D. Argaez, S. Dubowsky, and C. Mavroidis. A coordinated jacobian transpose control for mobile multi-limbed robotic systems. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 1910–1915 vol.3, May 1994.

[5] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Generalized biped walking control. *ACM Trans. Graph.*, 29(4):130:1–130:9, July 2010.

[6] Kai Ding, Libin Liu, Michiel van de Panne, and KangKang Yin. Learning reduced-order feedback policies for motion skills. In *Proceedings of the 14th ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '15, pages 83–92, New York, NY, USA, 2015. ACM.

[7] Libin Liu, KangKang Yin, Michiel van de Panne, Tianjia Shao, and Weiwei Xu. Sampling-based contact-rich motion control. *ACM Trans. Graph.*, 29(4):128:1–128:10, July 2010.

[8] P.G. Kry, L. Reveret, F. Faure, and M.-P. Cani. Modal locomotion: Animating virtual characters with natural vibrations. *Computer Graphics Forum*, 28(2):289–298, 2009.

[9] Sumit Jain and C. Karen Liu. Modal-space control for articulated characters. *ACM Trans. Graph.*, 30(5):118:1–118:12, October 2011.

[10] Rubens F. Nunes, Joaquim B. Cavalcante-Neto, Creto A. Vidal, Paul G. Kry, and Victor B. Zordan. Using natural vibrations to guide control for locomotion. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pages 87–94, New York, NY, USA, 2012. ACM.

[11] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. Dynamic terrain traversal skills using reinforcement learning. *ACM Trans. Graph.*, 34(4):80:1–80:11, July 2015.

[12] Xue Bin Peng, Glen Berseth, and Michiel van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Trans. Graph.*, 35(4):81:1–81:12, July 2016.

[13] Xue Bin Peng, Glen Berseth, Kangkang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.*, 36(4):41:1–41:13, July 2017.

[14] Matthew Moore and Jane Wilhelms. Collision detection and response for computer animation. *SIGGRAPH Comput. Graph.*, 22(4):289–298, June 1988.

[15] Andrew Witkin, Michael Gleicher, and William Welch. Interactive dynamics. *SIGGRAPH Comput. Graph.*, 24(2):11–21, February 1990.

[16] KangKang Yin, M. B. Cline, and D. K. Pai. Motion perturbation based on simple neuromotor control models. In *11th Pacific Conference onComputer Graphics and Applications, 2003. Proceedings.*, pages 445–449, Oct 2003.

[17] R. Weinstein, E. Guendelman, and R. Fedkiw. Impulse-based control of joints and muscles. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):37–46, Jan 2008.

[18] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition (2Nd Ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[19] I. T. Jolliffe. *Principal component analysis*. Springer-Verlag, 2002.

[20] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 43–54, New York, NY, USA, 1998. ACM.

[21] Joe Hicklin, Cleve Moler, Peter Webb, Ronald F. Boisvert, Bruce Miller, Roldan Pozo, and Karin Remington. Jama : A java matrix package.