Stepping in Human Motion

David Brown

Abstract

Stepping is the basis of a large portion of human motion. Everything from locomotion to simply having a conversation with someone usually involves some stepping. As such, there is a large body of work examining stepping in animation and robotics. However, existing research has focused on stepping in the context of locomotion and/or fall prevention. These are certainly important aspects of animation and robot control, but they only cover a small portion of the situations in which humans perform stepping. In this report we will look at how and when humans step in different situations, examine some previous work involving stepping, and introduce a simple system for generating potential step locations given a character's root trajectory.

1 Introduction

The Oxford English Dictionary defines a step as "An act of bodily motion consisting in raising the foot from the ground and bringing it down again in a fresh position" [7]. The goal of stepping is usually to move the body to a new location. As such, locomotion is one of the most common reasons for stepping (and one of the most studied). However, once we reach our destination we may want our feet in some particular configuration to perform whatever task we set out to do. For example to open a door we need to position our feet so we have enough leverage to pull on the door but also keep our feet clear of the path the door will take when we open it. Many other tasks that involve interaction with environment such as writing on a whiteboard, arranging objects on a shelf, or cooking all involve many small steps in which we are not only trying to move to a specific position but also get our body into a configuration where performing some desired task is possible. We will look at some of these examples more closely to try and come up with some insight into how, when, and why humans step.

2 Previous Work

There is a very large body of research in robotics, animation, and biomechanics on stepping and tasks involving stepping. We will be focusing on robotics and animation publications that look at and when a person or robot should step. A common theme among all the papers examined is whether where the character steps depends on the trajectory of the character (it's current and predicted position and velocity) or whether the trajectory of the character is determined by where it plans to step.

2.1 Inverted Pendulum Control

The inverted pendulum model and its derivatives are very commonly used in step planning for robotics and animation. The idea is to model a stepping character or robot as a point mass on an inverted pendulum. The base of the pendulum is placed at the support position and the point mass at the character's center of mass. It is then possible to examine the energy of this model to determine where the character should step next so that the pendulum will end up with zero (or some fixed amount) of kinetic energy. This technique has been used for both balance [5, 8, 10] and locomotion [2, 4, 6, 9]. Most techniques use the inverted pendulum to decide where to step based on the character's center of mass position and velocity and thus fall into the category of determining step position based on the character's trajectory.

2.2 Locomotion

Kajita et al. use a modified inverted pendulum where the height of the center of mass is fixed point mass is fixed [4]. This constraint simplifies the equations of motion and enables them to generate walking patterns. Using this model they find the trajectory of the center of mass is a hyperbolic curve which they then use to create a control law that computes where the next step position should be to best achieve a desired center of mass position and velocity.

The SIMBICON locmotion control method [11] is a popular method of controlling walking and running characters due to its simplicity and robustness. They specify target poses for different phases of the walk and adjust the swing leg hip joint's target angle using an intuitive feedback strategy. The angle is adjusted based on the distance from the stance foot to the center of mass and the velocity of the center of mass. So the character will step farther when the character is moving quickly (high center of mass velocity) or when character is leaning very far forwards (center of mass far from the stance foot). This makes for an effective stepping strategy even though there is no explicit target step location. The speed of the character is implicitly controlled by the target poses and gain parameters and the controller determines where it should step based on the character's speed. This places it in the category of choosing where to step based on the character's trajectory.

Coros et al. combined a variety of existing techniques to create a robust walking controller [2]. They use a fixed length inverted pendulum to decide where to step based on the character's center of mass position and velocity. They do include the ability for the character to interact with the environment by reaching for, pulling, and lifting objects. These interactions do not directly affect planned stepping locations, however the parameters of the inverted pendulum model are adapted in the case of lifting and pulling objects.

de Lasa et al. take the opposite approach and specify step trajectory parameters such as step length, step duration, and foot spread, and then uses a center of mass trajectory that should be compatible. The downside of this is that the resulting controller does not adapt steps to external disturbances or unexpected terrain.

These works adapt their steps to the state of the character, but none of them adapt to the state of the environment. Wu and Popović's locomotion controller is one example of work that does adapt to the environment. They use linear feedback to adjust the step position similar to [11], but in addition they also modify their default step trajectory so the swing foot will clear any obstacles in the environment.

Another work that adapts stepping is Igor et al.'s work on locomotion with low dimensional planning [6]. Their technique not only adapts the step position but also the center of mass trajectory to the environment. They do so by mapping the character to a spring loaded inverted pendulum (SLIP) model with a small set of control parameters. This model can be forward simulated to determine how it will move with a given set of parameters. An optimization is used to find the parameters that best achieve the character's goal (including step time, step distance, and facing direction). Once the best parameters are found the full character attempts to track the resulting center of mass position. The benifit of this technique is that the step locations and center of mass trajectory are determined together instead of one being a function of the other. This allows the character to successfully navigate terrain such as stepping stones which the other techniques mentioned would fail.

2.3 Balance and Recovery

Another heavily studied type of stepping is stepping to maintain balance. Much of the work in this area has been in robotics since when a robot falls it is likely to damage its self and/or those around it. Most techniques use some variation of the inverted pendulum to determine where to step.

Pratt et al. use a modified inverted pendulum with a telescoping leg and a flywheel instead of a point mass [8]. This lets them take into account the angular momentum of the robot as well as changes in the leg length. Instead of just finding a single point where the robot must step to maintain balance, they find a region of points in which the robot must step to maintain balance.

Komura et al. use a standard inverted pendulum, but then use human push recovery data to determine the parameters of the model [5]. They put subjects in a motion capture system with force plates on the ground and pushed a subject with varying force to see under what conditions they took a step and how far they stepped. Using this data they parameterised both when their model should step and how far it should step based on the push impulse.

Wu and Zordan designed a controller that can take arbitrary directed steps, and then used it with a supervisor system to automatically take reactive steps to maintain balance [10]. The stepping controller allows a user to specify a location and duration for a step and the controller will compute a center of mass trajectory and swing foot trajectory to take that step. Their supervisor system looks at the character's linear and angular momentum to determine if and where the character needs to step to maintain balance. They also show that if they exclude angular momentum, their technique is the same as using an inverted pendulum model.

2.4 Discussion

Nearly all these techniques either choose step locations based on the center of mass trajectory or determine the center of mass trajectory based on step positions. The one exception is [6]. This limits their ability to work for situations with complex interactions or constrained environments. Most also do little to no step position planning to make future tasks easier. For example to step over some obstacle like a curb most people will take a few longer or shorter steps leading up to the curb so that when they step over the curb their stance foot is close to the curb but does not run into it. This kind of short to medium range planning is vital for successfully interacting with the environment.

3 Step Generation

Most of the papers looked at in the previous section decided where to step using some kind of model or simulation. However another alternative is to use example motion capture data of steps to decide when and where to step. To demonstrate this idea we created a system where the user draws a spline and then the system attempts to fit some existing steps from motion capture data to this spline.

3.1 Implementation

To start we need to extract steps from motion capture data. To do this we look at the velocity of the feet. When a foot's velocity is below a certain threshold we assume it is in contact with the ground. Since motion capture data can be a bit noisy we look at the average velocity over several frames. Additionally if a foot has two intervals of low velocity with a small interval of high velocity in between we assume the small high velocity interval is some bad data and merge the two intervals. Once we have all the intervals where the foot is in contact, we can assume that all the intervals where it is not in contact the actor is taking a step.

All the steps we extracted from the motion capture data are in different world positions and orientations. To make them easier to work with we transform each step so that the support foot for that step is facing in the positive x direction.

Once we have this normalized step data we can fit it to the spline. To do this we greedily choose steps to add to our list of steps that follow the spline. At any given point in the spline we look at each possible step, translate it so its support position is the same as the end of the last steps swing foot position and rotate it so the angle of the vector from the support position to the initial root position is the same as the angle from the previous step's final swing foot position and the position of the spline for the current time. We then calculate the error of the step as

$$E = E_{path} + E_{final} + \frac{E_{time}}{10}$$

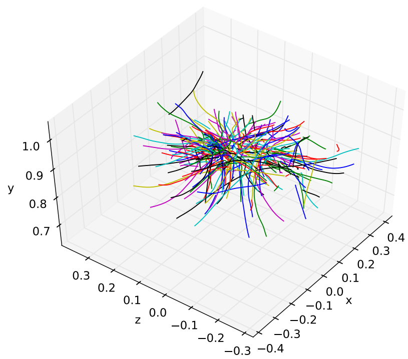


Figure 1: Trajectories of the root position relative to the stance foot for the brownie motion set. Note how the steps are relatively short in X direction (parallel to the stance foot) and somewhat longer in the Z direction (perpendicular to the stance foot).

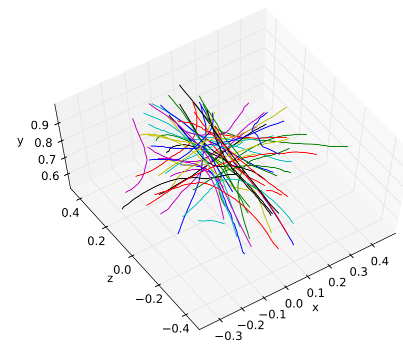


Figure 2: Trajectories of the root position relative to the stance foot for Subject 03's motion set. Note the many 90 degree turns.

E_{path} is the average distance between the transformed root position projected onto the ground and the spline position for each frame of the step. E_{final} is the distance between the root position at the end of the step and the desired root position at that time computed from the spline. And E_{time} is the negative of the duration of the step to prefer longer steps over shorter ones.

We start at the beginning of the spline finding the best step and the advancing the duration of the step up the spline until the end of the spline is reached.

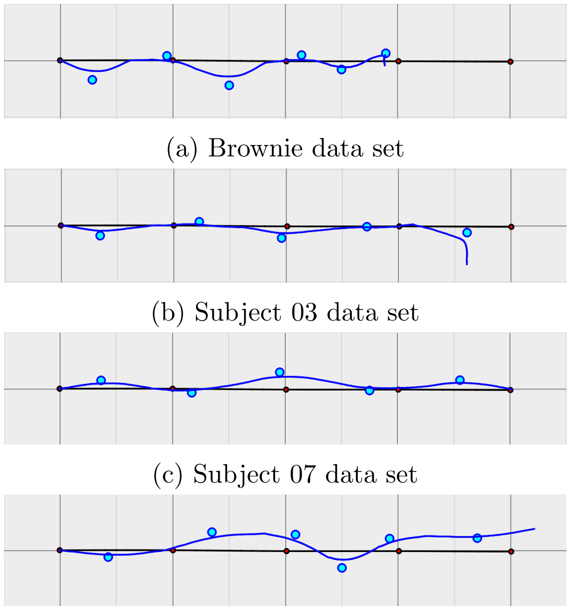
3.2 Results

We tested this system using steps from four different sets of motion capture data. The first we'll call the brownie set contains 252 steps and is a recording of someone making brownies obtained from the CMU Multi-Modal Activity Database [3]. Since the person is making brownies many of the steps are sideways and most are relatively short. Figure 1 shows the trajectories of all the steps in this set.

The other three sets of motions were taken from the CMU Motion Capture database [1]. The second motion set is all the motions for Subject 03 which contains 116 steps and is a recording of someone navigating a set of different height boxes. This includes many large forwards steps as well as turning 90 degrees in either direction. Figure 2 shows the trajectories of all the steps in this motion set. The third motion set contains all the motions for subject 07 which has 47 steps and contains all walking motions of various speeds with no turns. The final motion set is all the motions of subject 35 which has 263 steps. This set is something of a combination of subjects 3 and 7, containing a variety of forward walking examples as well as many turns.

To demonstrate our system we show the result of trying to fit each motion set to different curves. Figure 3 shows each data set fitted to a straight line with a speed of 1 meter per second. The subject 03, 05, and 35 data sets all do reasonably well. However the brownie data set fails because all its steps are all too small to walk at 1 meter per second.

Figure 4 shows trying to fit each data set to a spiraled, lower speed curve. With the lower speed, the brownie data set is able to fit it very well since it has so many steps in a wide variety of directions. The subject 07 data set does not fit as well since it only has 90 degree turns and larger steps. The subject 07 data set only has straight line walks so it is unable to turn with the curve and fits very poorly. The subject 35 set fits relatively well since it has both fast and



(a) Brownie data set

(b) Subject 03 data set

(c) Subject 07 data set

(d) Subject 35 data set

Figure 3: Fitting each set to a straight line with a 1 meter per second speed. The black line is the spline and the red dots on it are control points spaced at 1 second intervals. The blue line is the fitted step path and the blue circles are the step positions

slow walks plus turns. However since most of its turns are 90 degrees like subject 03 it has to make some very abrupt turns to stay on track.

Finally, Figure 5 shows fitting a longer spline using the subject 35 data set. Although the stepping path deviates from desires spline a fair bit at times, it eventually gets back to it and overall fits relatively well.

4 Conclusion

Although there has be a large number of publications examining human and robot stepping, they have been largely confined to looking at locomotion and balance recovery related stepping and there is a relative lack of work looking at stepping for other tasks where interaction with the environment is involved. In addition, nearly all of the papers reviewed had either step locations entirely dependant on the center of mass trajectory or had the center of mass trajectory dependant on step locations. Ideally both step locations and center of mass trajectory would be selected together to satisfy the constraints of where you can step for a task while still being able to position the body where it is needed to carry out the desired task.

We have presented a basic technique for choosing step locations to track a desired root trajectory using steps extracted from motion capture data. This system tries to fit the desired trajectory as best as possible while being constrained by the steps available in the motion capture data. Although it works relatively well there are many possible improvements that could be made. For one we do not take into account the root body orientation or the foot orientation. For more accurate and robust stepping this would undoubtedly be required. We also do no adaptation or blending of the steps from the motion capture data. With even just a little bit of blending we would likely be able to fit the desired trajectory much more closely. Our step fitting algorithm could also be improved since our greedy method is certainly not optimal. Finally the error measure we used for determining the best step for any given situation could likely be improved by including things like how smooth the transition from the previous step is or other terms.

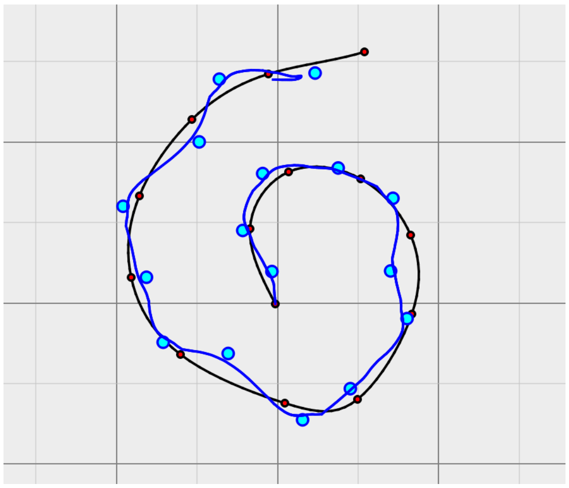
Despite our naïve solution we are still able to produce reasonable looking step locations for a wide variety of trajectories. Additionally, since our step extraction technique is entirely automatic it is easy to improve the solution our system find by simply supplying it with more motion capture data with more steps.

References

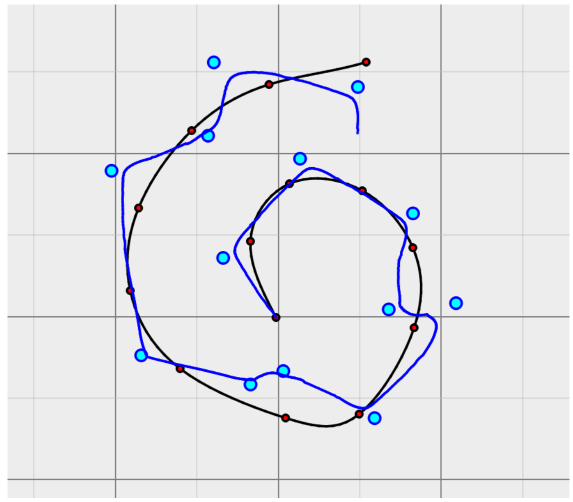
[1] Cmu graphics lab motion capture database. `http://mocap.cs.cmu.edu/`.

[2] Stelian Coros, Philippe Beaudoin, and Michiel van de Panne. Generalized biped walking control. *ACM SIGGRAPH 2010 papers on - SIGGRAPH '10*, page 1, 2010.

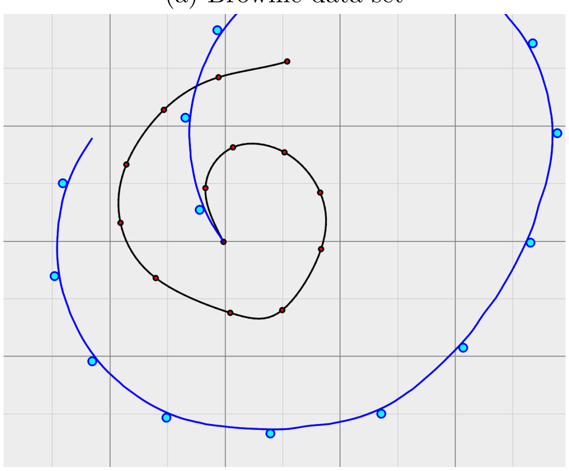
[3] F. De la Torre, J. Hodgins, J. Montano, S. Valcarcel, Forcada R., and



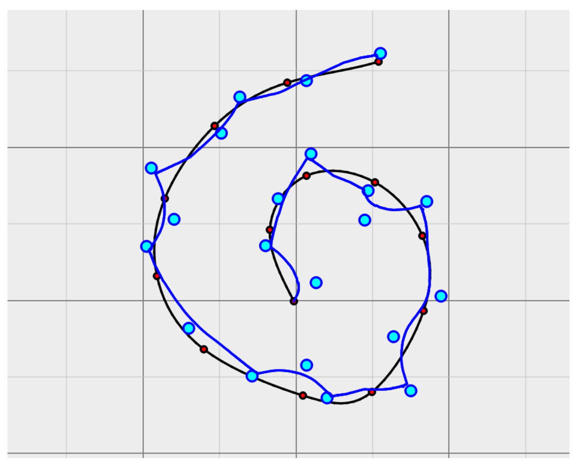
(a) Brownie data set



(b) Subject 03 data set



(c) Subject 07 data set



(d) Subject 35 data set

Figure 4: Fitting each set to a spiraled line. Black line is the spline and the red dots are the control points which are 1 second apart. The blue line is the fitted step path and the blue circles are the step positions

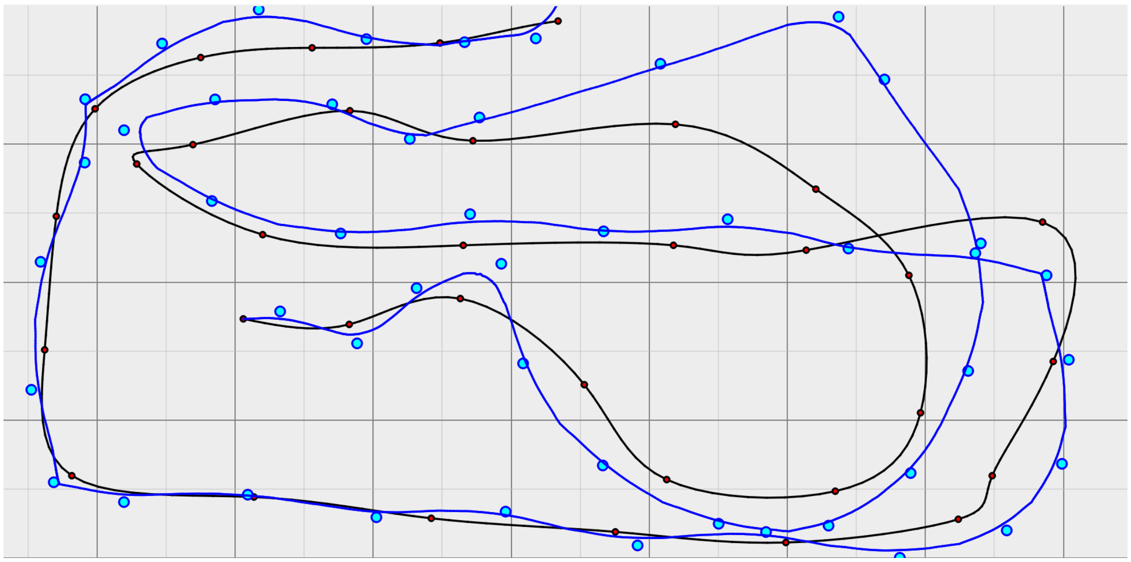


Figure 5: Fitting a long path with the subject 35 data set

J. Macey. Guide to the carnegie mellon university multimodal activity (cmu-mmac) database. Technical Report CMU-RI-TR-08-22, Robotics Institute, Carnegie Mellon University, July 2009.

[4] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, K. Yokoi, and H. Hirukawa. The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246. IEEE, 2001.

[5] T. Komura and K. Ikeuchi. Stepping motion for a human-like character to maintain balance against large perturbations. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, (May):2661–2666, 2006.

[6] Igor Mordatch, Martin De Lasa, and A Hertzmann. Robust physics-based locomotion using low-dimensional planning. *ACM Transactions on Graphics . . .*, 2010.

[7] OED Online. “step, n.1”. `http://www.oed.com/view/Entry/189836?rskey=nejbXT&result=2&isAdvanced=false`, March 2013.

[8] Jerry Pratt, John Carff, Sergey Drakunov, and Ambarish Goswami. Capture Point: A Step toward Humanoid Push Recovery. *2006 6th IEEE-RAS International Conference on Humanoid Robots*, pages 200–207, December 2006.

[9] Yao-Yang Tsai, Wen-Chieh Lin, Kuangyou B Cheng, Jehee Lee, and Tong-Yee Lee. Real-time physics-based 3D biped character animation using an inverted pendulum model. *IEEE transactions on visualization and computer graphics*, 16(2):325–37, 2010.

[10] Chun-Chih Wu and Victor B Zordan. Goal-directed stepping with momentum control. *Proceedings of the 2010 ACM SIGGRAPH/*, 2010.

[11] Kangkang Yin, Kevin Loken, and Michiel van de Panne. SIMBICON: Simple Biped Locomotion Control. *ACM Transactions on Graphics*, 26(3):105, July 2007.