3. DISCRETE LIMIT CYCLE CONTROL

This chapter describes our basic control approach and its application to the generation of bipedal walking motions. Section 3.1 begins by describing the notion of limit cycles, on which our control formulation is based. Next, Section 3.2 presents the overall control strategy and develops a discrete system model to be used with a number of user-specified control elements to stabilize periodic open-loop motions. Section 3.3 discusses the application of this control system to bipedal walking. The underlying open-loop control, which serves as the basis for a desired gait, is discussed in Section 3.4. Sections 3.5 and 3.6 then go on to describe various possible observed and controlled variables for walking. Section 3.7 provides details on the application of the control elements introduced in earlier sections to the generation of balanced walks. Finally, minor variations on the basic control which are of particular use in improving the aesthetics of the human model's motion are described in Section 3.8. While the basic formulation is applied to bipedal walking, it is not inherently tied to any particular model or motion and could potentially be applied to the generation of limit cycles in other types of animated figures. Some of the terms we define in this and subsequent chapters have different meanings in the context of control system theory. While we make efforts to avoid such conflicts, we have chosen to sometimes give preference to the colloquial usage of terms. Our relaxed definition of a limit cycle is one example of such usage.

3.1 Limit Cycles

One common characteristic of many non-linear dynamical systems is the existence of system-wide *limit cycles*. A limit cycle is a periodic, cyclic trajectory through the state-space of a system. Strictly speaking, a limit cycle involves the full state of the system. However, within the context of this thesis we will use a relaxed definition in which only part of the full system state must cycle periodically. This definition is one of convinience since it allows periodic locomotion such as walking or running to be discussed in terms limit cycles.

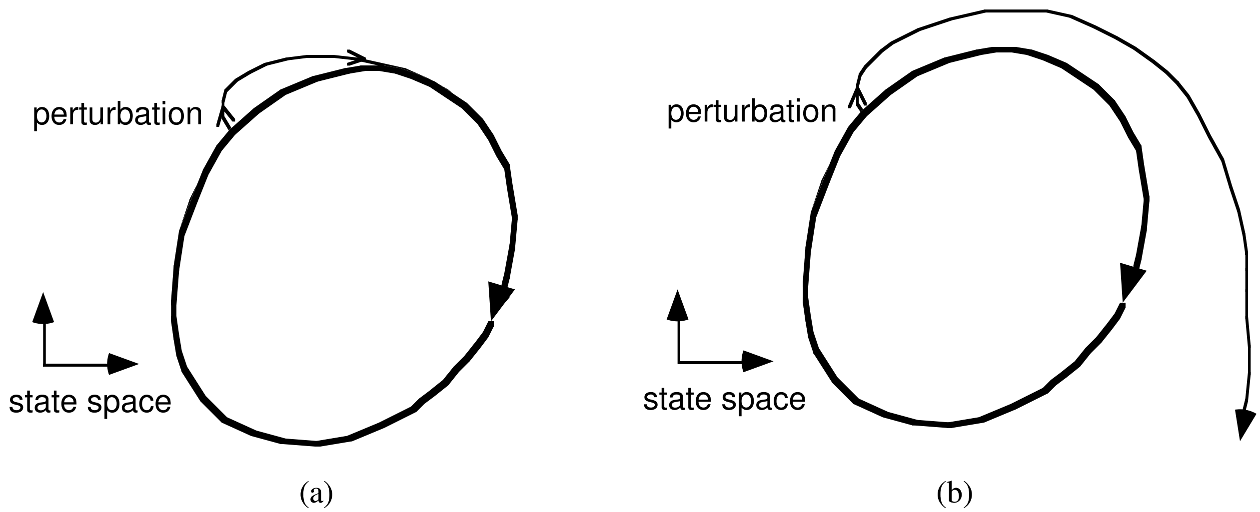


Figure 3.1 - Passive Limit Cycle Stability
(a) Passively stable
(b) Passively unstable

Limit cycles may be *stable* or *unstable*. A stable limit cycle is one in which slight perturbations to the state-space trajectory are driven back into the limit cycle as indicated in Figure 3.1 (a). An unstable limit cycle is one in which slight perturbations to the trajectory result in the system deviating further from the limit cycle as shown in Figure 3.1 (b). We will call limit cycles that do not require explicit control forces to maintain them *passive* limit cycles. Note that this definition does not preclude a system with active components (motors etc.) from exhibiting passive limit cycles. A motorized or windup toy is an example of such a system.

We wish to attain similar stable limit cycles with passively unstable bipedal systems by applying suitable control forces to periodically drive the system back into an *active* limit cycle. We define an *active* limit cycle as one that requires corrective control forces to be applied to the system for the explicit purpose of maintaining the cyclic trajectory. Figure 3.2 illustrates this idea.

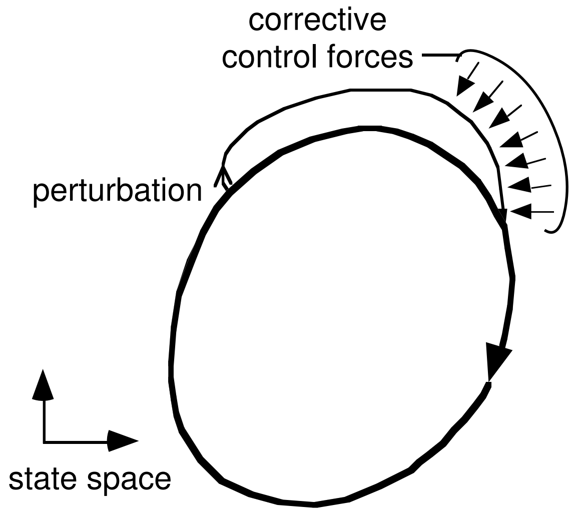


Figure 3.2 - An Active Limit Cycle

3.2 Control Formulation

The problem of choosing appropriate control perturbations to drive the *entire state* of a system to a desired value is a difficult one. Assuming a solution does exist, the number of parameters to be determined is large for all but very simple systems (a few DOFs or less). Non-linearities in a system mean that over the course of a full cycle even small perturbations of certain state variables can cause large changes in final state and/or result in almost no cycle at all. For example, a small change in the roll angle of the ankle in a walk might cause the next foot to miss the ground completely.

The essence of our control approach is to begin with a passively unstable system, discretize it into individual cycles and stabilize each cycle in turn. Each cycle is stabilized by applying control perturbations which drive its final state to a suitable state from which to begin the next cycle. The motivation for using a discrete version of the system is that the discrete dynamics are much simpler to model than the continuous system and therefore, simpler to control, as we shall see shortly.

The continuous dynamic system which we eventually wish to control can be modelled by the following system state equation:

$$x(t+\Delta t) = V(x(t), U). \tag{3.1}$$

where x is the system state and U is a set of applied control forces defined over $[t, t+\Delta t]$.

V is a highly complex function which involves the integration of the forward dynamics of the animated model over time and includes the effect of both internal and external applied forces such as gravity, ground collisions, and muscular control forces. Instead of working directly with this complex continuous system, we assume that a strictly cyclic motion is desired and discretize Eq. 3.1 into individual motion cycles to obtain

$$x_{i+1} = g(x_i, U_i). \tag{3.2}$$

Here, the subscript i denotes the cycle number. U_i is the set of time-varying control forces applied over the ith cycle. The function g is a special case of V in which the sample times are not necessarily regular[1], depending on the definition of a motion cycle. For example, the end of a motion cycle could be defined as the time of a particular transition in a state machine. We further assume that a user-supplied open loop controller, U^{nom}, produces a near-cyclic motion when applied to the system being controlled. To drive the final motion into a cycle, additional control forces are required. We denote these forces as ΔU_i^*, which are the control perturbations required to drive each cycle of the nominal motion, toward a limit cycle. The discrete system then becomes

$$x_{i+1} = g\left(x_i, U^{nom} + \Delta U_i^*\right). \tag{3.3}$$

where ΔU_i^* are still to be determined. Figure 3.3 illustrates this discrete dynamical system.

---

[1] Note that strictly speaking, g is a different function for each cycle since the size of the interval over which it is defined may vary from one cycle to the next.

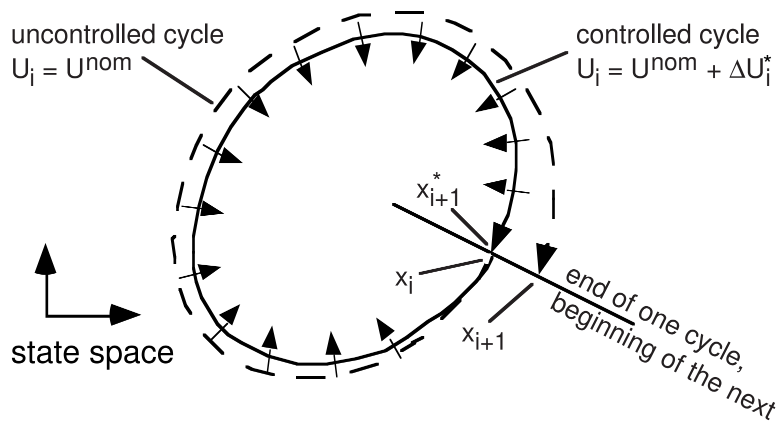


Figure 3.3 - Discrete System

x_i	initial state of the ith cycle/final state of the (i-1)th cycle
x_{i+1}	final state of the uncontrolled ith cycle
x_{i+1}^*	final state of the controlled ith cycle/initial state of the (i+1)th cycle
U^{nom}	basic cyclic control
ΔU_i^*	computed control perturbation for ith cycle

We will furthermore choose to express the corrective control forces, ΔU_i^*, as the linear sum of several "basis" corrective actions:

$$U_i \quad = \quad U^{nom} + \sum_{j=1}^{N} k_{ij} \Delta U_j \quad = \quad U^{nom} + K_i \bullet \Delta U \tag{3.4}$$

where ΔU_j are fixed control perturbations which are defined over a cycle and k_{ij} are linear scaling factors applied to ΔU_j. K_i is the vector of perturbation scaling factors and ΔU is a vector whose elements are the fixed control perturbations, ΔU_j, which remain the same from one cycle to the next. *N* is the dimensionality of our control system and is equal to the number of state variables which we wish to observe (and control).

Rather than using the complete system state, we choose to work with a small number of *regulation variables* (RVs). Regulation variables are a projection of the system state and are the observed part of state in our control system. Using a reduced set of state variables greatly reduces the computational effort required to construct a model of the discrete system. We will use Q to denote the vector of RVs and define $\gamma(x)$ to be the projection function such that

$$Q = \gamma(x). \tag{3.5}$$

Note that choosing to work strictly in the reduced state space carries the imlicit assumption that controlling the reduced state is sufficient to control the complete system state in a desireable way. For this to hold, $\gamma(x)$ and ΔU must be chosen appropriately. However, there is no guarantee that such an appropriate choice exists.

Replacing the full state in Eq. 3.3 with the reduced state, Q, and substituting the control formulation of Eq. 3.4 yields the reduced-order system which we will control directly:

$$Q_{i+1} = h\left(Q_i, U^{nom} + K_i \bullet \Delta U\right) \tag{3.6}$$

For a given cycle, Q_i, U^{nom}, and ΔU are treated as apriori information. We will use $Q_{i+1} = h(K)$ as a convinient short form of Eq. 3.6 when we are interested in discussing only the effect of K on the system.

Eq. 3.6 can be restated as

$$Q_{i+1} = Q^{nom} + \Delta Q_{i+1} = h\left(Q_i, U^{nom} + K_i \bullet \Delta U\right) \tag{3.7}$$

where
$$Q^{nom} = h\left(Q_i, U^{nom}\right) \tag{3.8}$$

We choose to approximate the response of this system about the nominal operating point Q^{nom} (where K=0) using the following linear predictive model:

$$\Delta Q_{i+1} = J\, K_i \tag{3.9}$$

where J is the NxN discrete system Jacobian, defined as $\left[\frac{\partial q_i}{\partial k_j}\ i, j \in [1, N]\right]$:

$$J = \begin{bmatrix} \frac{\partial q_1}{\partial k_1} & \cdots & \frac{\partial q_1}{\partial k_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial q_N}{\partial k_1} & \cdots & \frac{\partial q_N}{\partial k_N} \end{bmatrix}. \tag{3.10}$$

J relates the change in Q_{i+1} over a single cycle to the applied perturbation scaling, K_i.

Figure 3.4 illustrates the relationship between the parameters of the linear model and the perturbed system's cyclic motion in state space for a 1D system. In this example, the linear predictive model is constructed using two sample points on $Q_{i+1} = h(k)$, corresponding to applied perturbation scalings of k_1 and k_2. The figure illustrates that we can predict (and hence control) the value of Q_{i+1}.

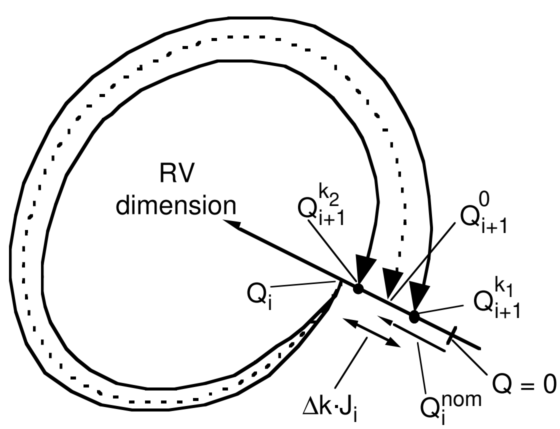


Figure 3.4 - Linear parameters of a 1D discrete system in state space.
Here, $\Delta k = k_2 - k_1$

Evidence supporting the use of a linear discrete system model for bipedal control and the details of model construction are presented later in the thesis (Section 3.6), after the particular choices of RVs and LPPs have been described.

The key to using the linear predictive model for determining the correct control action is the following. Once the linear discrete system model has been constructed, the control perturbation scalings required to drive Q_{i+1} to a desired value, can be computed using the inverse model:

$$K_i = J^{-1}\Delta Q^d_{i+1} \qquad (3.11)$$

where $\Delta Q^d_{i+1} = Q^{nom} - Q^d_{i+1}$, the desired change in the RVs with respect to Q^{nom} for the current cycle.

Figure 3.5 shows the limit cycle viewed as state trajectory with respect to time. Three consecutive controlled cycles of a 1D discrete system are shown in this diagram, T is the cycle period, k_i is the perturbation scaling for cycle i, and V_i is the resulting state trajectory for cycle i. In this example, Q^d_{i+1}, the desired RV value for each cycle, is held constant for all three cycles.

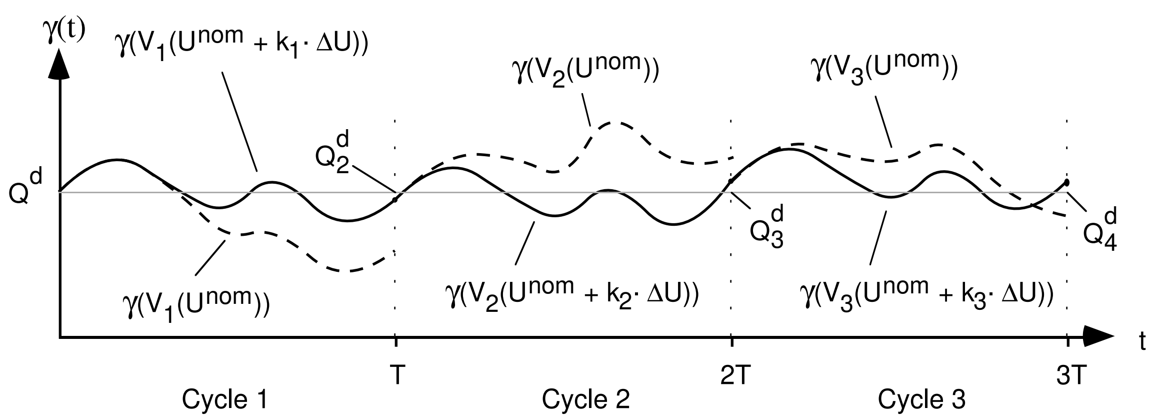


Figure 3.5 - Three cycles of a typical 1-dimensional system. In this case, each cycle has the same period, T.

Our control formulation assumes the following apriori information, supplied by the user:

1. the open-loop control, U^{nom}.

2. the RV projection function, γ.
3. the fixed control perturbations, ΔU.
4. the desired (target) values for the RVs, Q_{i+1}^{d}, for each cycle, i.

The parameters of the discrete system model (Q^{nom} and J) and the required perturbation scalings, K_i^*, are calculated automatically. These computations will be discussed in Section 3.7 after the various elements of the bipedal control system have been described.

3.3 Application to Bipedal Locomotion

In attempting to generate balanced locomotion for a biped, we must first select the number of control dimensions to be used. For successful balance, the base of support must, on average, remain under the centre of mass and the torso should remain generally upright. Only two control dimensions are required to achieve this, one in each dimension of the horizontal plane. Thus to balance each step, our bipedal control system will use two RV dimensions and require two LPPs which span the RV space. Since we use pose control exclusively with our biped we will use B and ΔP to represent the pose control equivalents of U^{nom} and ΔU respectively.

The base PCG, B, describes one complete cycle of motion, or in the case of bipedal walking, two steps. For bipedal walking we choose to split the cycle into two symmetric halves and apply our control formulation to each step. The PCG control perturbations, ΔP, affect the motion throughout the cycle, rather than at a single point in the cycle.

Figure 3.6 illustrates the overall pose control structure used for our bipedal systems. The structure is an expansion of Eq. 3.4, for N=2, with the left step and right step perturbations specified explicitly. In summary, each step is balanced by the choice of two scalar parameters, for example, k_{10} and k_{20} for step 0. These parameters are calculated automatically using the discrete system model.

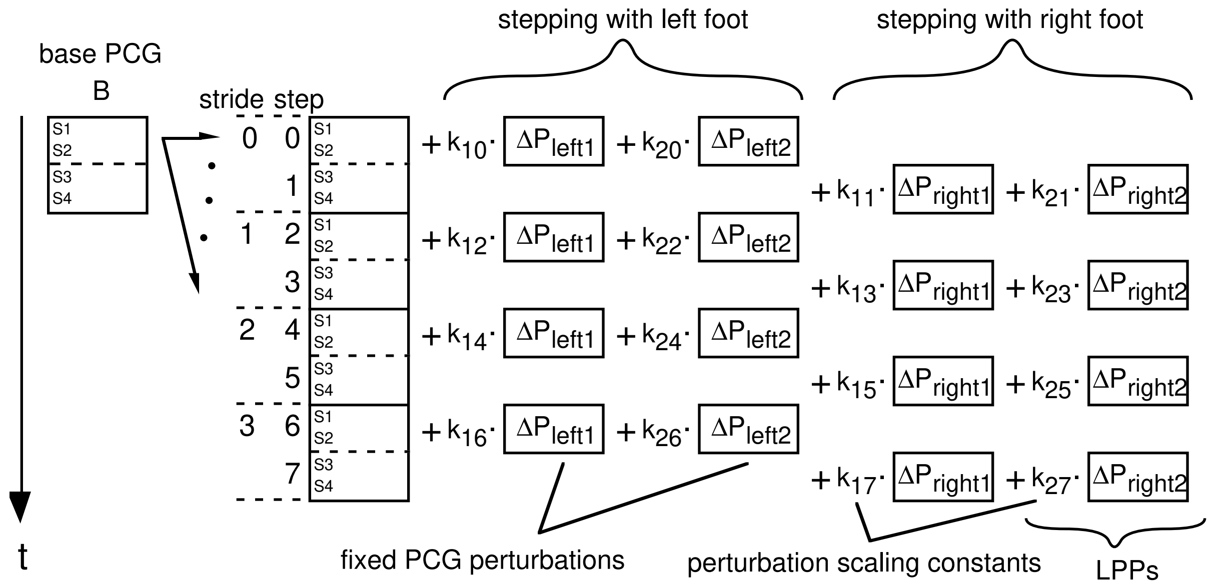


Figure 3.6 - Overall limit cycle control structure for a walking biped. The walk begins with the left leg as the stance leg. B is the open-loop base PCG. C is the overall applied control for each step, consisting of the base PCG B and additional (left or right) stance leg perturbations to balance the step. The Si are the states of B (four in this example).

With suitable parameter choices, the control algorithm is able to generate stable walking limit cycles whose open loop motion is described by the base PCG.

Appropriate parameter choices are crucial for successful control. The next few sections describe each component of the control system for a walking biped. The process of choosing various parameters will be described as they are introduced.

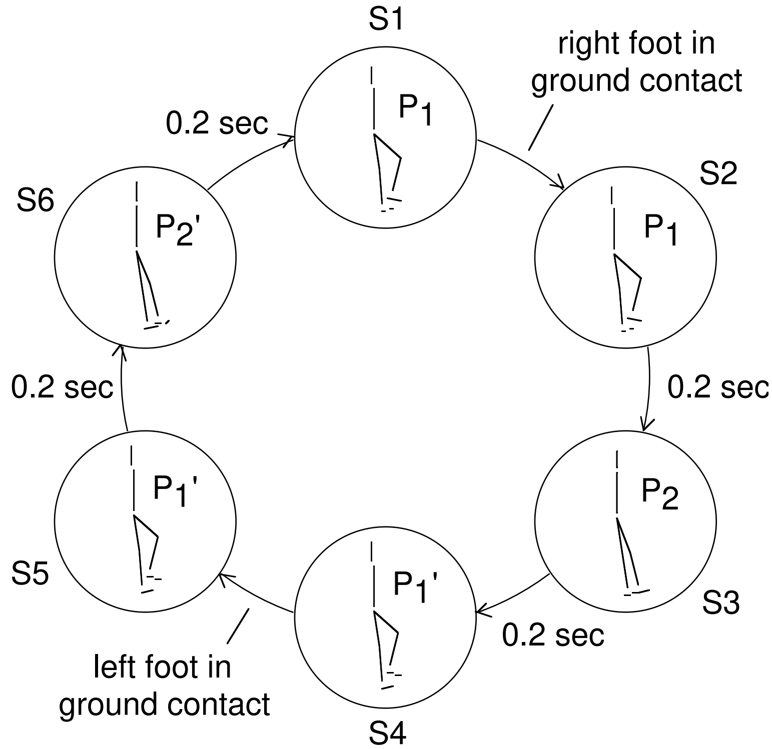
3.4 Nominal Open-loop Control, U^{nom}

A base PCG provides the nominal open-loop control for the periodic motion of the biped. In the case of a forward walk, the base PCG consists of a sequence of poses to generate a forward step with one leg followed by a forward step with the other leg. The base PCG for walking must generate motion that is initially close to attaining a balanced walk when used without additional compensating perturbations. A base PCG producing an open-loop walk of a few steps is more likely to be successfully stabilized than one that falls in the first step.

Figure 3.7 shows a base PCG used to generate a forward walking motion for the simplest human model along with its equivalent graphical representation. The base PCG has right-left symmetry, although PCG asymmetry can also be quite useful as will be demonstrated by the turning perturbations presented in Section 6.2.

The base PCG consists of six states. The poses in states S1 and S2 are identical. The poses in states S4, S5, and S6 are identical to those of states S1, S2 and S3 respectively, with the left and right sides exchanging roles. Having a total of only four unique poses, the controller describes a simpler motion than a typical human walk. This relatively small number of poses simplifies the specification of a base PCG as well as the creation of suitable parametric perturbations.

Sensor-based transitions are necessary to synchronize the biped's motion to the external world. The base PCG of Figure 3.7 uses such transitions to move from S1 to S2 and from S4 to S5. A transition occurs when the swing foot for the current step (i.e. the next stance foot) contacts the ground. If this has already happened before the state is entered, the transition occurs immediately. Actively ensuring ground contact at fixed points in the walk cycle is important in certain situations such as when starting from an initial resting state. While the sensors are typically no longer necessary once a walking limit cycle is reached, changing from one limit cycle to another can require sensor-based synchronization again.



(a)

DOF:		1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info	
	S1	0	0	0	-50	60	-5	0	0	-10	0	0	0	0	R
	S2	0	0	0	-50	60	-5	0	0	-10	0	0	0	.2	
state	S3	0	0	0	-20	0	5	0	0	-10	0	0	0	.2	
	S4	0	0	0	-10	0	0	0	0	-50	60	-5	0	0	L
	S5	0	0	0	-10	0	0	0	0	-50	60	-5	0	.2	
	S6	0	0	0	-10	0	0	0	0	-20	0	5	0	.2	

(b)

Figure 3.7 - Forward walking base PCG for a simple human model

(a) State diagram (right foot dashed). Poses P_i and P_i' are left-right symmetric.

(b) Pose table. All DOFs are in degrees relative to reference position. Transition information is given as time (seconds) followed by an optional foot sensor type (left or right) for sensor-based transitions.

State	Description
S1	Right foot placed on ground Left leg begins forward swing (knee bent for ground clearance)
S2	Right leg propels body forward Left leg continues forward swing (knee bent for ground clearance)
S3	Left leg extends in anticipation of ground contact
S4	Left foot placed on ground Right leg begins forward swing (knee bent for ground clearance)
S5	Left leg propels body forward Right leg continues forward swing (knee bent for ground clearance)
S6	Right leg extends in anticipation of ground contact

Figure 3.8 shows the open loop motion resulting when the base PCG alone is used for control. As expected, the biped falls over after several steps.

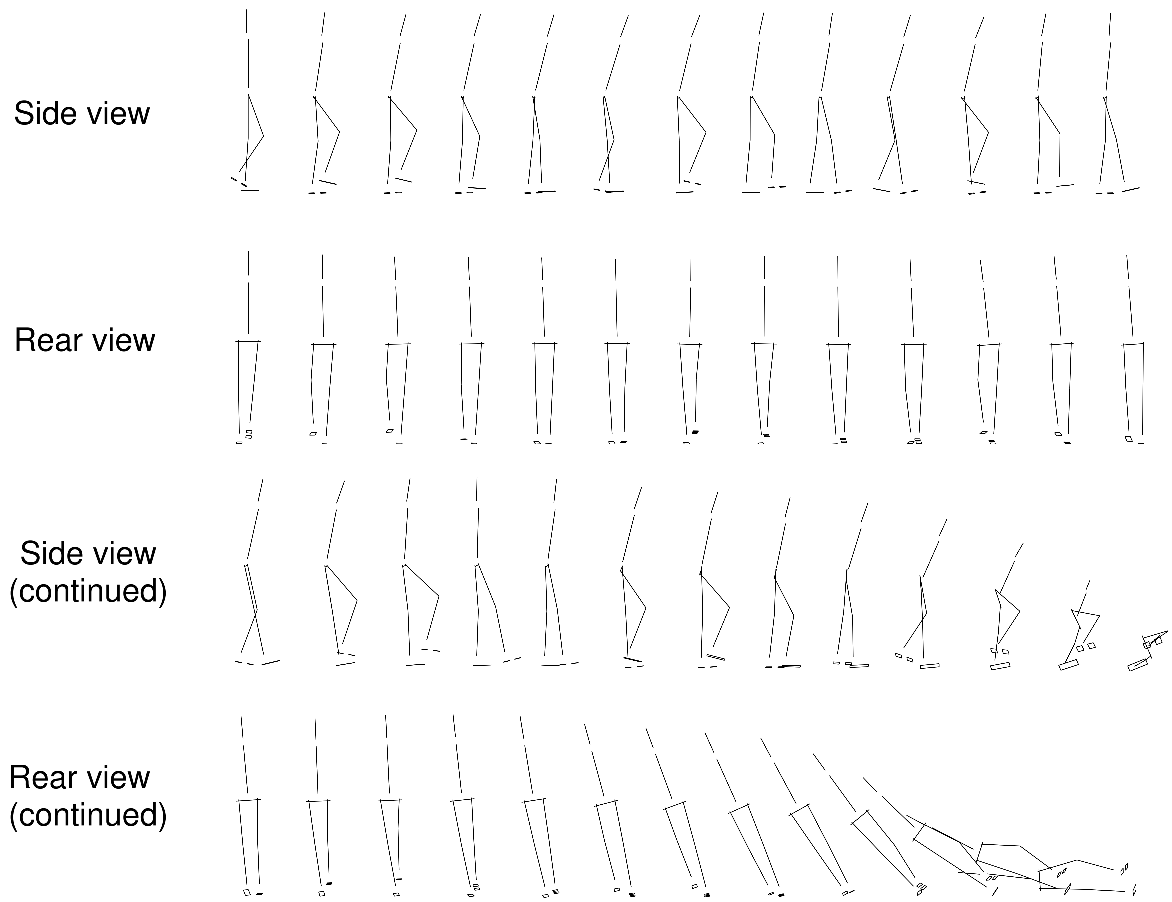


Figure 3.8 - Unbalanced motion of human model using base PCG shown in Figure 3.7.

Surprisingly, the small number of poses is sufficient not only for walking, but also for a limited form of running. Running can be achieved simply by replacing the 0.2 second time-based transitions of the walking base PCG with 0.1 second transitions. The flexibility of the pose controller to accommodate the different motions is due to the ground sensors which are used in state transitions. The sensor-based transitions exist to ensure that the next stance foot is in contact with the ground before proceeding into its stance phase. However, they do not constrain the initial time of ground contact. In the case of running, ground contact typically occurs shortly after entering state S1 or S4 as the new stance leg moves to propel the biped forward. In the case of walking, ground contact typically occurs in the previous state (S6 or S3) toward the end of the swing leg extension. These two scenarios are illustrated in Figure 3.9.

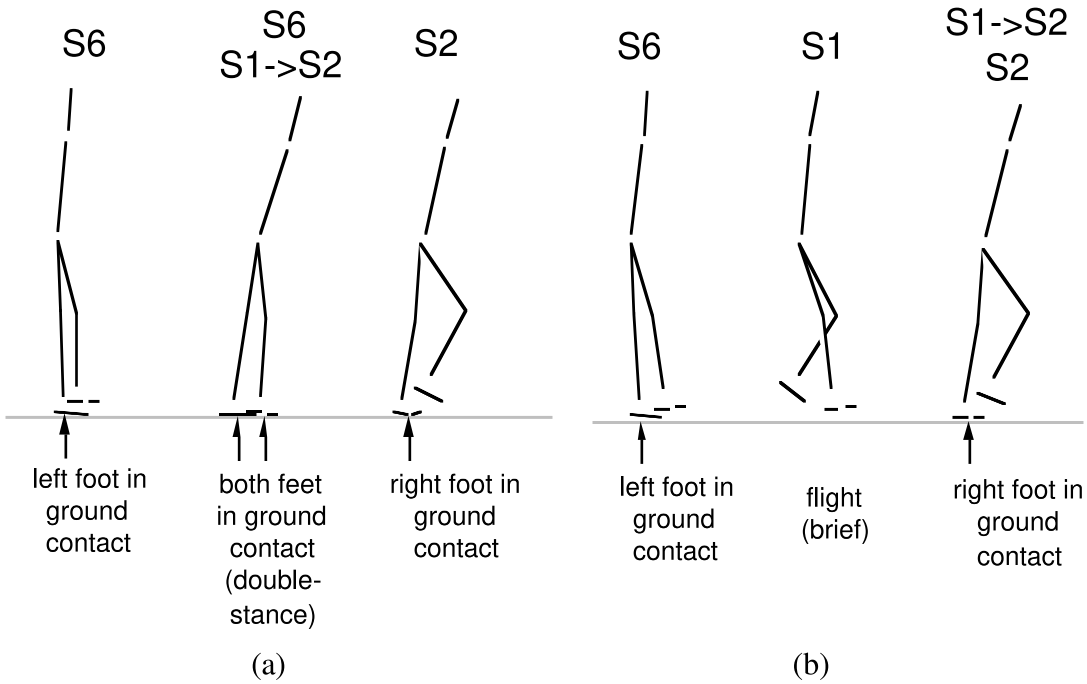
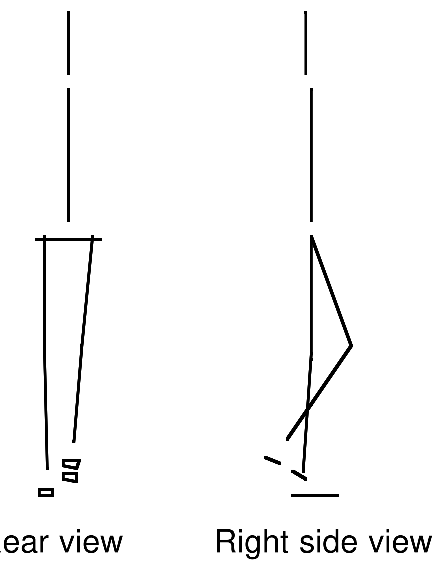


Figure 3.9 - Possible walking and running steps described by base PCG of Figure 3.7. Related states and transitions are indicated.
(a) walking
(b) running

An initial state must also be chosen for the walk. The initial state provides a starting position from which a successful walking motion can begin. While the simplest initial state to use would have the creature at rest in the reference position, it is easier to achieve a limit cycle when the initial state more closely resembles a state on a balanced walk cycle. Finding an appropriate initial state currently requires trial-and-error on the part of the animator. Figure 3.10 shows the initial state which was generated manually for the simple human model controlled by the base PCG of Figure 3.7.



DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1
Value:	0	0	0	0	0	0	0	-5.1	-20	50	0	0

Figure 3.10 - Initial configuration for simple human model. Right foot shown as two segments for clarity. All velocities are 0.

3.5 Choice of Regulation Variables, *Q*

As discussed earlier, the discrete balance control of Eq. 3.6 is applied only once per step. This implies that each regulation variable must represent the behaviour of some part of the system state over an entire step as a single scalar value. It is important to choose suitable functions of state and sampling times which give a reasonably smooth response to perturbations as we assume in Section 3.2 by using the discrete system Jacobian.

All of the RVs presented in this thesis are projections of the system state at a specific point in the cycle as shown in Figure 3.11. The sample times correspond to the approximate time of foot placement for each step. Variations such as sampling an average or peak value of some function of state over the whole step are also promising but unexplored possibilities.

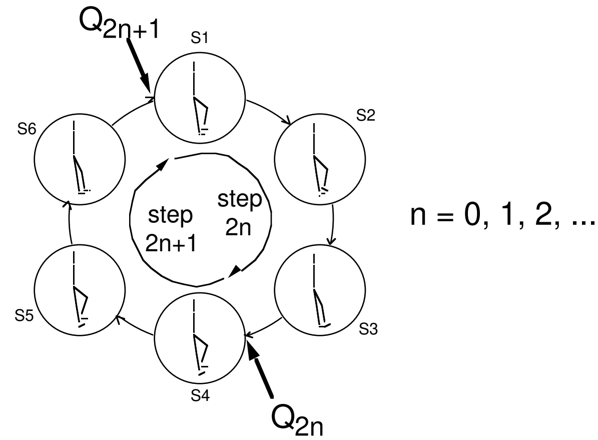


Figure 3.11 - End-of-step sampling times for Q_i.

RVs must be chosen to be meaningful functions of the system state in order to improve the chances of attaining a limit cycle. A number of general criteria should be met:

1. RVs should incorporate some key element(s) of the desired cycle and should reflect movement or position relative to the world frame. An example that meets both criteria is "uprightness" with respect to the world vertical axis for the case of walking.

2. The RVs should be easily controllable. It should be possible to reach a wide range of RV values from various initial states by choosing an appropriate parametric control perturbation. For example, the system must have DOFs which can affect the chosen RVs.

3. RVs should vary smoothly around a reasonable range of interest. For example, in the case of a walking biped, the range of interest might correspond to configurations with a relatively upright torso.

4. We should be able to estimate a target value for each RV which will keep the motion close to a desirable limit cycle. For example, to attain a walking limit cycle, we must be able to determine target RV values that will keep the biped balanced.

5. When multiple RVs are used, independence or near-independence of the RVs can be useful. RV interdependencies can be complex (e.g. non-smooth), making it more difficult to find parametric perturbations which cause only smooth variations over a wide range of RV values. For our bipedal control, we assume near-independent RVs.

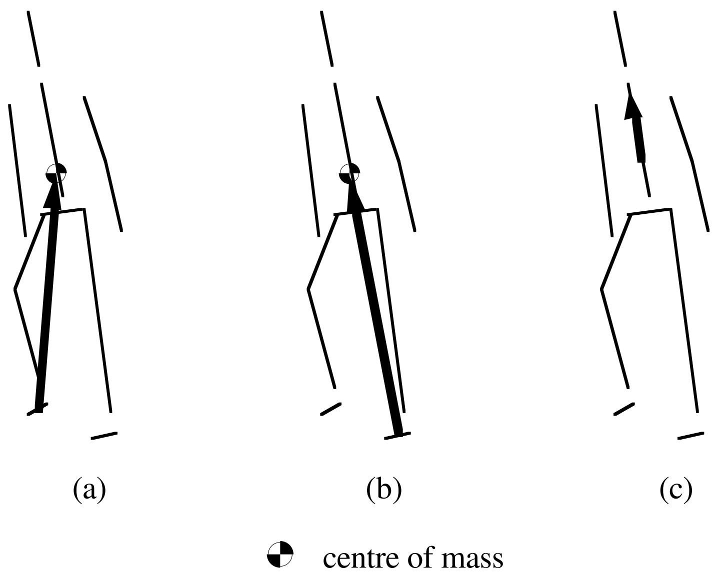


Figure 3.12 - Balance RV vectors
(a) swing-COM vector
(b) stance-COM vector
(c) up vector

In this thesis, we experiment with three choices of RVs, based on the vectors shown in Figure 3.12. The first, the *up-vector*, is based on the notion of torso "uprightness". The up-vector is fixed to and runs along the length of the torso in the human model and the head in the robo-bird model. The *swing-centre of mass (swing-COM) vector* describes the position of the COM of the biped with respect to the current swing foot. The *stance-centre of mass (stance-COM) vector* indicates the position of the COM with respect to the stance foot. The sampling time for all three types of RV are at the end of states S3 and S6. For the purpose of computing RVs, the swing and stance legs do not exchange until after the last base PCG state of the step. The leg which is the swing leg for most of the current step is used to compute the swing-COM RV. The stance-COM RV is treated in a similar fashion.

The RVs are two scalar components extracted from the given vector. The vector is sampled at the end of the current step, normalized, and projected onto the horizontal plane. The projection is then decomposed into the two components as illustrated in Figure 3.13 for the up-vector case. The components provide an indication of the forward and lateral lean of the chosen vector. The forward direction is defined as being orthogonal to the ground-plane projection of a line joining the biped's hips.

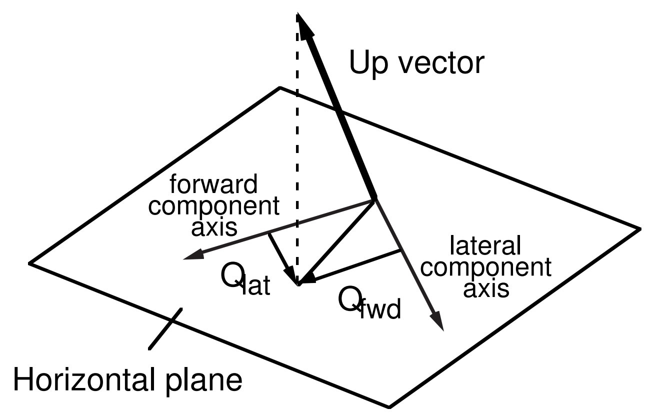


Figure 3.13 - Decomposition of up vector projection into RVs

3.6 Choice of Perturbation Parameters, ΔU

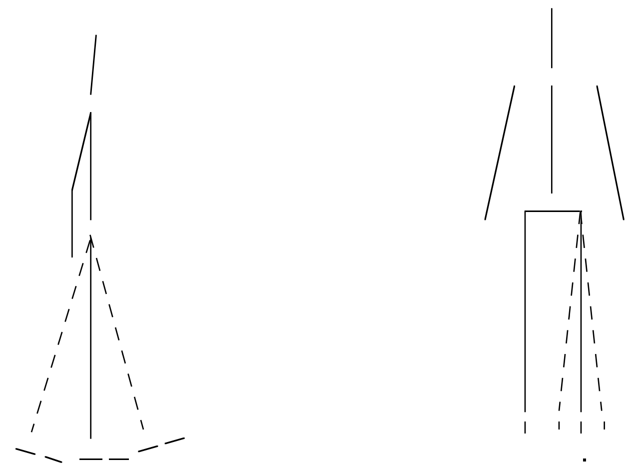
The balance control formulation of Figure 3.6 uses two linear parametric perturbations (LPPs) to control each step in the base PCG cycle. Recall from Section 2.3 that each LPP consists of two basic components, a *fixed PCG perturbation*, and a scalar multiplier. New values for each scalar multiplier are automatically computed for each step to balance that step. The fixed part of the PCG perturbations remain the same throughout a given walk. This section discusses the choice of fixed PCG perturbations which must be supplied by the animator. Throughout this section, the term "perturbation" will be used to refer strictly to the fixed part of an LPP rather than an arbitrarily scaled perturbation.

As in the case of RVs, the choice of perturbations should meet a minimal set of requirements to be effective. The criteria for good PCG perturbations are related to the criteria for suitable RVs, discussed in Section 3.5. In general, the following should hold:

1. The scaled perturbation should be able to reach a wide range of desired RV values from a wide variety of initial conditions.
2. Perturbations should provide smooth control over the RVs so that a suitable linear model of the discrete system can be constructed. The inverse of this model is used to determine the appropriate scaling factors for each perturbation for each step. For bipedal walking, the chosen perturbations provide near-linear control over each of the three balance control RVs presented in Section 3.5.
3. Perturbations should be designed to be as independent as possible in order to reduce or eliminate non-linear interactions between RVs.

The control perturbations chosen in this thesis affect the stance hip pitch and stance hip roll DOFs. They are shown in Figure 3.14. Items (c) and (d) show the pose table form of the perturbations which are scaled and added to a base PCG pose table such as that of Figure 3.7 (b). In our case, each of the chosen PCG perturbations affects a single DOF. Unity-valued perturbations are used so that the scalar multiplier units are in degrees. Each perturbation is applied to all poses (i.e. all states) of the related step, left or right. This provides a smoother and more effective control perturbation.

Figure 3.14 illustrates the hip pitch and roll perturbations for a body in free space. When applied to a body in contact with the ground, the perturbations primarily affect the biped as shown in Figure 3.15. Applying a stance hip pitch perturbation varies the torso angle in the sagittal plane. Applying a stance hip roll perturbation varies the torso angle in the coronal plane.



(a) (b)

	DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info
	S1	0	0	0	0	0	0	0	0	1	0	0	0	0
	S2	0	0	0	0	0	0	0	0	1	0	0	0	0
state	S3	0	0	0	0	0	0	0	0	1	0	0	0	0
	S4	0	0	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	0	0	0	0	0	0	0	0

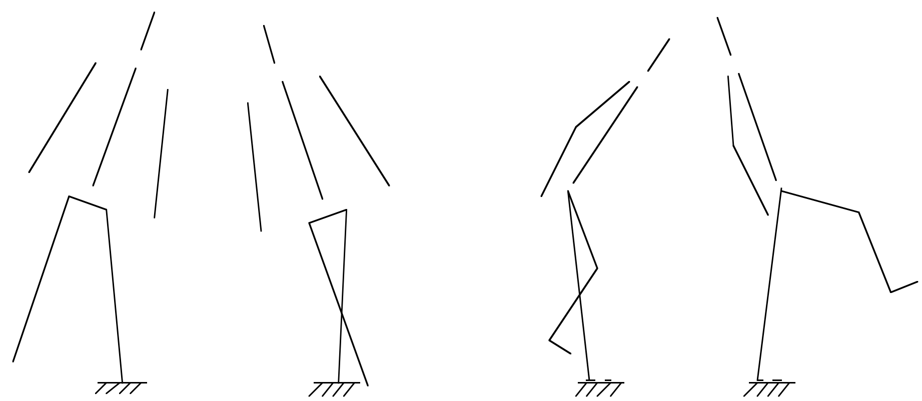
(c)

	DOF:	1	2	3:0	3:1	4	5:0	5:1	6:0	6:1	7	8:0	8:1	transition info
	S1	0	0	0	0	0	0	0	1	0	0	0	0	0
	S2	0	0	0	0	0	0	0	1	0	0	0	0	0
state	S3	0	0	0	0	0	0	0	1	0	0	0	0	0
	S4	0	0	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	0	0	0	0	0	0	0	0

(d)

Figure 3.14 - Right hip pitch and roll perturbations for a simple human model. During states S1 to S3, the right leg is the stance leg. DOF 6:1 is right hip pitch. DOF 6:0 is right hip roll.
(a) hip pitch (viewed from right)
(b) hip roll (viewed from rear)
(c) hip pitch (pose table)
(d) hip roll (pose table)

Figure 3.16 shows the relationship between the stance hip pitch perturbation and the forward and lateral component of each of the three types of RVs introduced in Section 3.5. Figure 3.17 shows similar results for the stance hip roll perturbation. Each curve represents the value of the particular RV at the end of a single step for a range of applied perturbation scalings. For each curve, a different initial state is used. The initial state for step *n* was generated by first simulating *n*-1 steps with the open-loop base PCG. By the fourth step, the figure is falling noticeably.



(a) (b)

Figure 3.15 - Typical effect of stance hip perturbations when dynamics are considered
(a) right stance hip roll
(b) right stance hip pitch

The graphs illustrate that hip pitch varies nearly linearly with Q_{fwd} for all three choices of RV. Similarly, hip roll is nearly linear with respect to Q_{lat}. These relationships provide evidence which supports our assumption that the discrete system can be modelled using a linear model. Despite the fact that the perturbations themselves are mutually independent, their effect on the RVs is not cleanly decoupled. Thus, they do not provide truly independent control over each RV dimension as desired. In general, however, the magnitudes of the undesired variations are not excessively large relative to the accessible range of RV values in the desired control dimensions. Completely independent control of the RVs would imply a diagonal discrete system Jacobian. The relatively small effects of each perturbation on other's control dimension mean that the off-diagonal elements of the discrete system Jacobian will be small compared to the diagonal elements.

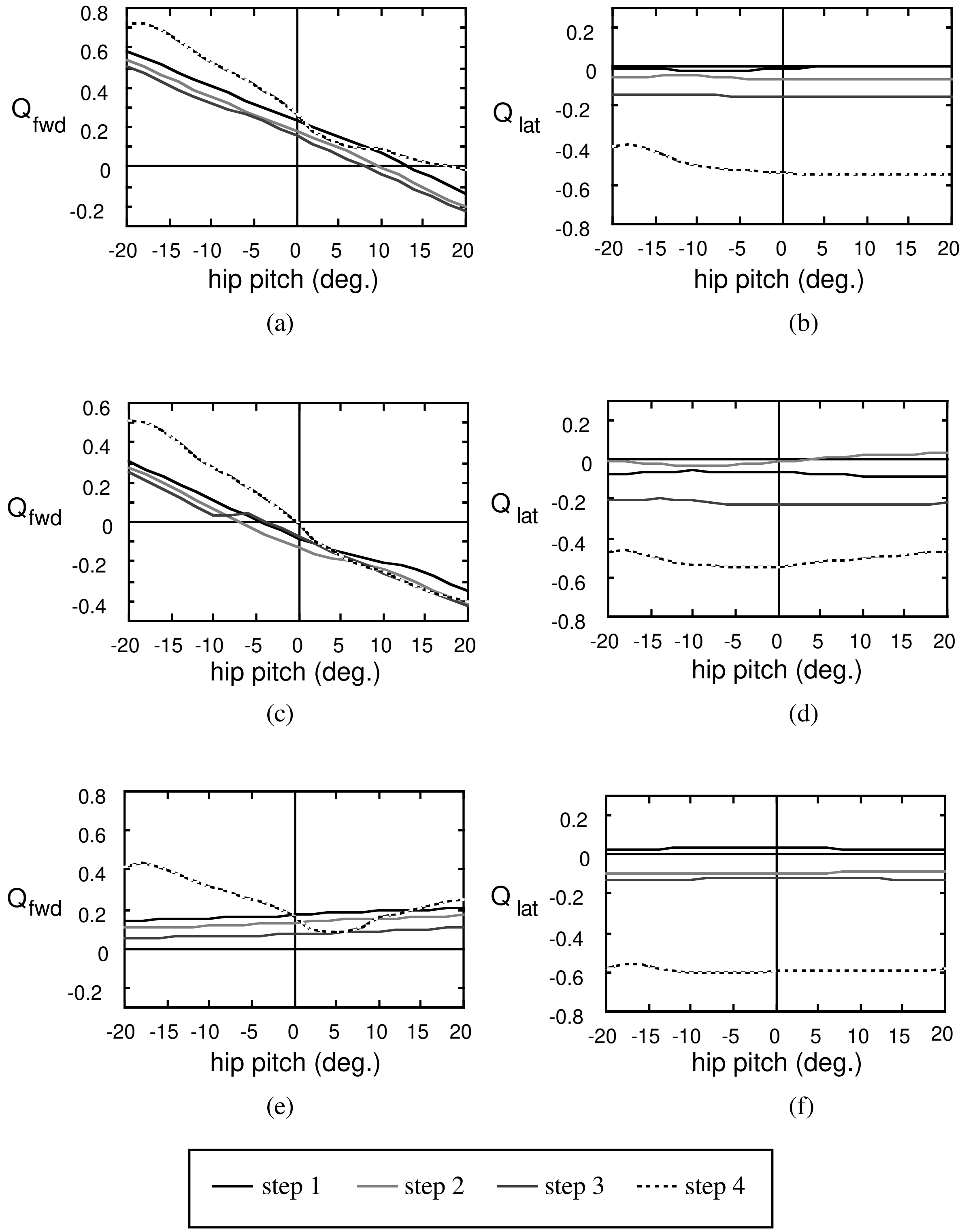


Figure 3.16 - Balance RV components vs linearly scaled hip pitch perturbation
(a) - (b) up vector
(c) - (d) swing-COM vector
(e) - (f) stance-COM vector

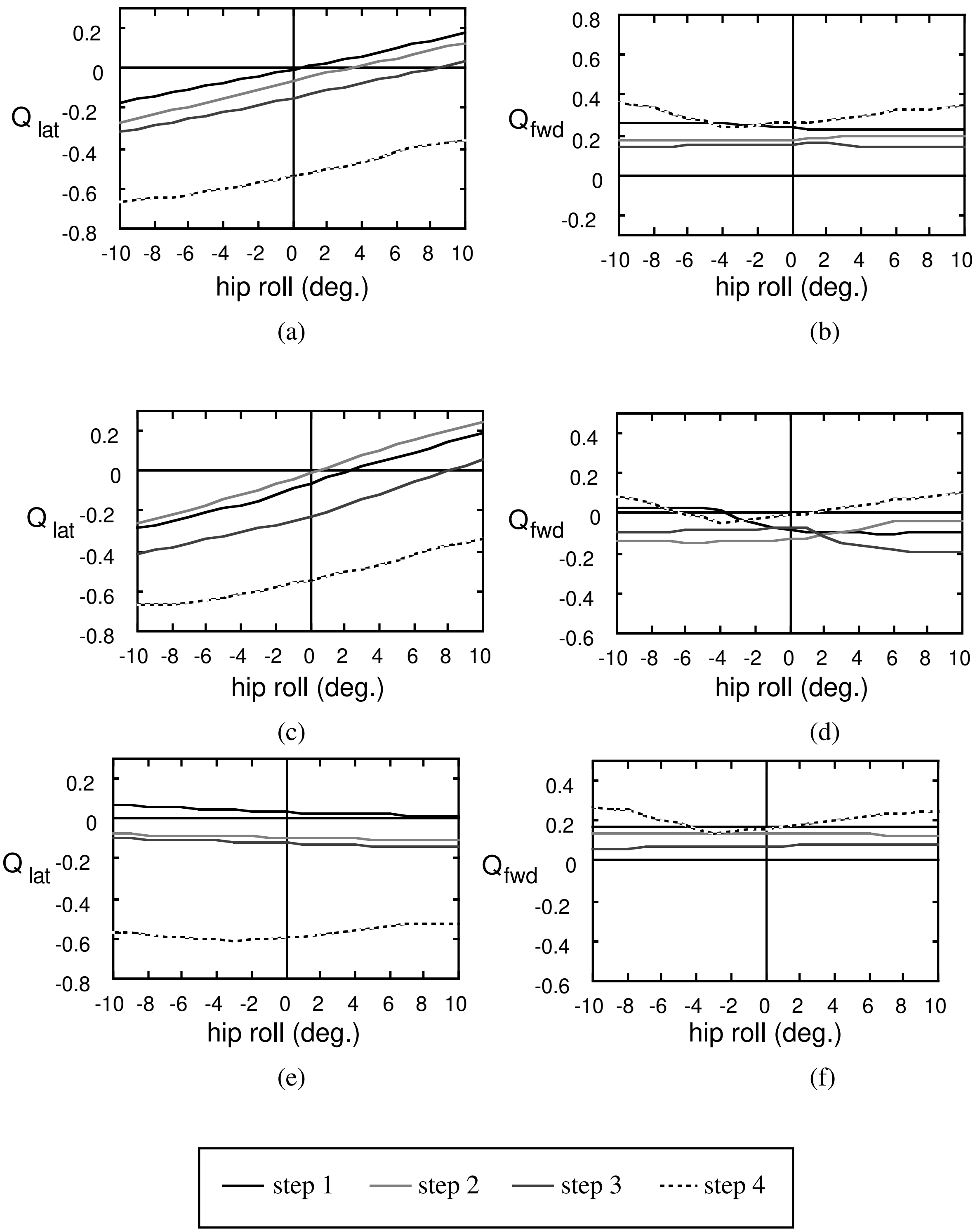


Figure 3.17 - Balance RV components vs linearly scaled hip roll perturbation
(a) - (b) up vector
(c) - (d) swing-COM vector
(e) - (f) stance-COM vector

These relationships allow us to attain final values for Q close to the desired values for each step using a simplified form of the Jacobian, discussed in the next section.

One interesting point to note is that the slope of the stance-COM curves in Figure 3.16 and Figure 3.17 are opposite in sign compared to the corresponding slopes for the up vector and swing-COM RVs. This is due to the mechanism through which the stance-COM angle changes. Figure 3.18 shows this effect exaggerated for clarity.

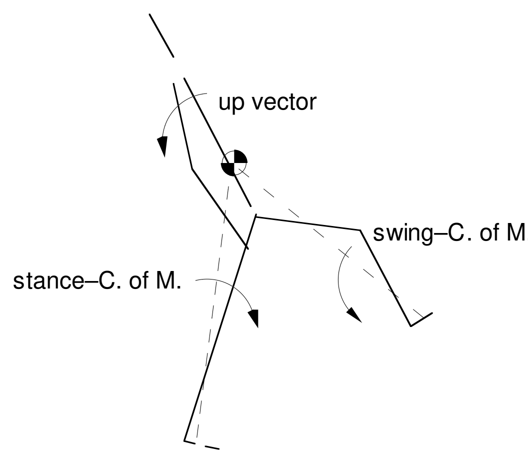


Figure 3.18 - Direction of change of forward RV components with hip pitch

3.7 Linear, Sampled "Balance" Control

This section describes how the base PCG, perturbations and various RVs discussed in earlier sections can be used to compute and apply the discrete system model parameters (J and Q^{nom}) to generate a balanced walk. The "balancing" is done by choosing appropriate RV target values for each step and finding the scaling factors to apply to the PCG perturbations to reach them. The scaling factors are determined automatically using the inverse of the linear discrete system model (Eq. 3.11). The balancing process is repeated, one step at a time, for as many steps as desired. In some cases, the resulting walk is erratic and wanders. In others, a walking limit cycle is reached.

Two separate problems must be solved in order to achieve the motions we seek:

1. We must choose an RV target value, Q^d, on the cycle, which is likely to drive the system into a reasonable limit cycle.
2. We must construct the discrete system model of Eq. 3.9 and then use it to compute the final control perturbation for each step. This involves calculating the discrete system Jacobian, J, and the nominal operating point, Q^{nom} for each step.

3.7.1 Desired RV Values

The desired (target) values for the RVs should be chosen each step to keep the system near a limit cycle. We choose to use a constant target value for all steps. The idea is that by forcing the RVs to the same desired value at the same point in each cycle, a limit cycle will be generated. This approach is sufficient both to drive the system into a stable limit cycle initially and to maintain it. Allowing the RV targets to vary from one step to the next can be useful, but requires a way to select them. This possibility is further explored in Chapter 5 where variation in Q^d is used to provide control over the biped's walking speed.

Once the choice is made to use constant RV targets, the particular values must be chosen. This is essentially a trial-and-error process. However, common sense and the behaviour of the open-loop system can provide valuable clues. For example, if a straight forward walking motion is desired, the lateral RV target can be quickly estimated using the final pose in one PCG step as shown in Figure 3.19. Similarly, the first few steps of the open loop motion (such as that in Figure 3.8) can provide a reasonable estimate for the forward component targets. Finally, performing a number of trials with different forward Q^d values provides a solution, if one exists, as well as information on the useful range of target values. In this case, a "trial" is an attempt to balance the biped's motion by applying our proposed control technique with some particular trial value of Q^d.

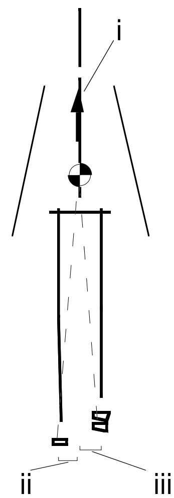


Figure 3.19 - Estimate of desired lateral RV components
(based on S6 from Figure 3.7 (a) - rear view)
i) up vector (vertical)
ii) stance-COM
iii) swing-COM

3.7.2 Constructing and Applying the Discrete System Model

Once an RV target has been chosen, the discrete system model is reconstructed and applied at each step as shown in Figure 3.20. In general, a single fixed model is not sufficient to represent all possible steps. While RVs provide a reduced, low dimensional representation of system state, they are also an incomplete and ambiguous representation of the system state. For example, the chosen RVs use position information and no velocity information. Such unobserved parts of the system state cause variations in both Q^{nom} and J. Figure 3.21 illustrates this idea. This result is evident in graphs of Figure 3.16 and Figure 3.17 where both the average slope and the offset of the curves can vary from one step to the next. These variations are especially apparent in the lateral control dimension.

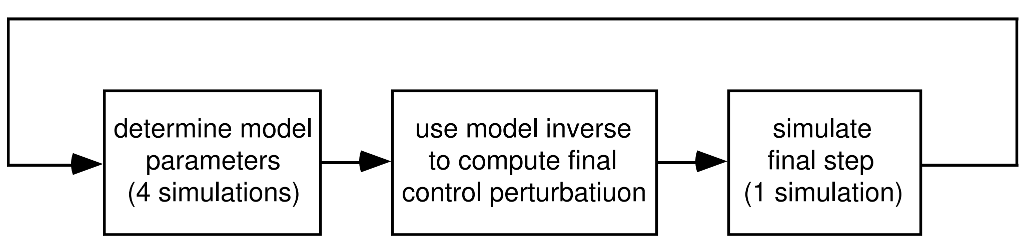


Figure 3.20 - Balancing process for each step.

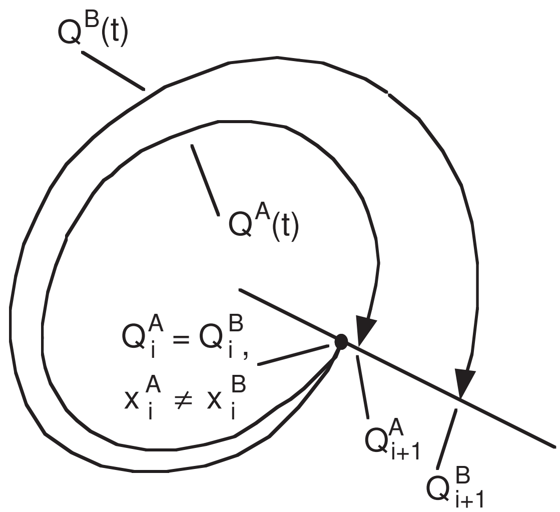


Figure 3.21 - Unobserved state problem. Two unperturbed RV trajectories beginning with identical values for Q_i but different initial states.

Because of the variation of Q^{nom} and J from one step to the next, it is (unfortunately) necessary to construct a new discrete system model for each step. This is done by sampling $h(K_i)$ to determine the model parameters for the current step, Q_i^{nom} and J_i. The inverse of this linear model, J^{-1}, is then used to choose the perturbation scalings necessary to reach Q_{i+1}^d. Figure 3.22 illustrates this process for a one-dimensional system.

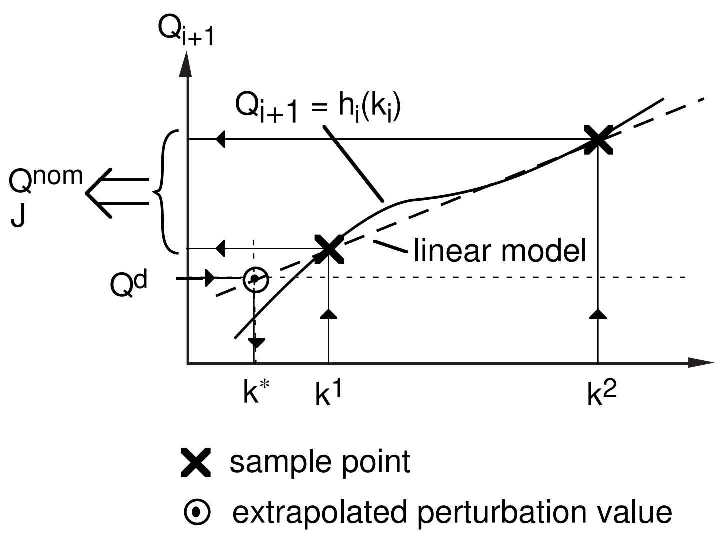


Figure 3.22 - Model construction and extrapolation. k^1 and k^2 are two sample points used to construct the forward model. The inverse model and Q^d yield the necessary applied perturbation scaling, k^*

In general, there is no guarantee that the discrete system Jacobian will be invertible for each step. However, with well-chosen choices of RVs and LPPs, singularities generally occur only for trajectories far enough from useful limit cycles to be of little interest. Typical singularities for the human model, occur when it is lying on its side and when the current stance leg does not contact the ground at all during the current step.

The general form of the Jacobian for a 2D system is:

$$J = \begin{bmatrix} \frac{\partial q_1}{\partial k_1} & \frac{\partial q_1}{\partial k_2} \\ \frac{\partial q_2}{\partial k_1} & \frac{\partial q_2}{\partial k_2} \end{bmatrix} \tag{3.10}$$

Using finite differences, a minimum of N+1 samples are required to construct the linear model of $h(K)$ for a control system of dimension N: one for the nominal operating point and one for each control dimension, each of which yields a column in the Jacobian. In this case, each "sample" consists of a simulation of one step with a different value of K. In practice, a greater number of samples may be required.

For the two dimensions of our bipedal control system, we use four sample simulations, two for each dimension. An additional simulation computes the final motion for the step after the final PCG scalings have been chosen using the model. Rather than using the complete Jacobian, we choose to work with two simplified forms:

1. Assume independent control dimensions and observe only the primary RV. This corresponds to the assumption of a diagonal Jacobian (i.e. $\partial q_i / \partial k_j = 0$, $i \neq j$). We will refer to this form as *superposition* (SP) sampling.
2. Assume near-independent control dimensions but allow the operating point to move as each final perturbation scaling is determined. This corresponds to a form of triangular

Jacobian which makes use the most up-to-date values of the model parameters in a manner similar to the Gauss-Seidel method for solving linear systems.

For the first, two independent 1D models are constructed then combined in the final simulation. For the second, two variations are considered. In the first, a 1D model is constructed in the forward control dimension and inverted to obtain the required K^*_{fwd}. A second 1D model is then constructed in the lateral control dimension making use of the known value of K^*_{fwd}. We refer to this as forward-then-lateral (F-L) sampling. Lateral-then-forward (L-F) sampling is similar but performs 1D control in the lateral dimension first, then uses this result to construct a 1D model in the forward dimension. We might expect that one or both of these approaches perform better than superposition sampling since they both incorporate additional knowledge of the perturbations to be applied. Somewhat surprisingly, this turns out not to be the case as we shall see in Chapter 4. All three approaches give comparable results, with superposition slightly outperforming the other two. The three sampling strategies investigated are illustrated in Figure 3.23.

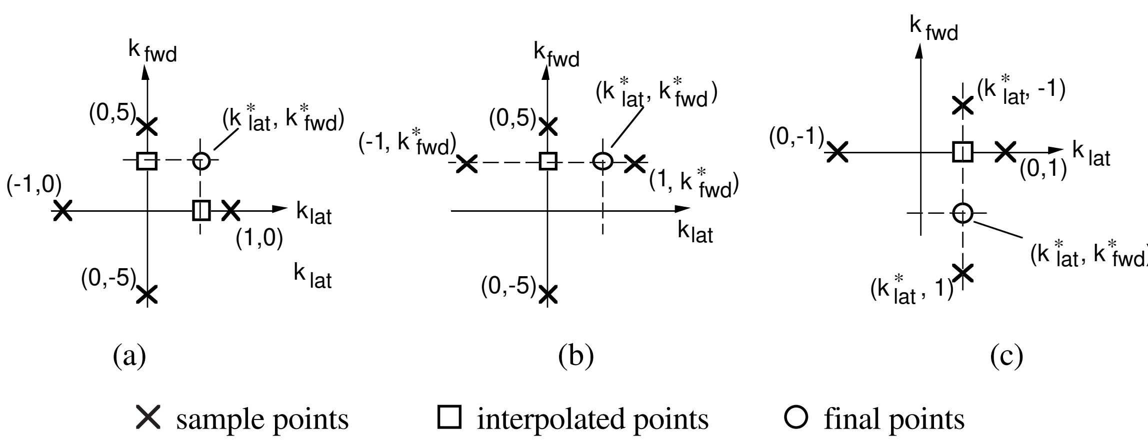


Figure 3.23 - 2D sampling strategies
(a) superposition (SP)
(b) forward then lateral (F-L)
(c) lateral then forward (L-F)

The final parameters to consider in constructing a good linear model of the discrete system are the perturbation scaling factors used to sample $h(K)$ (e.g. k^1 and k^2 in Figure 3.22). Since the discrete system is quite linear, the precise location of the sample points is not critical. There are, however, a few potential pitfalls to be avoided in choosing them:

1. Since we cannot necessarily predict the sign of the required perturbations from one step to the next, it is often best to sample $h(K)$ symmetrically about the nominal operating point, K=0, rather than at the nominal point itself. Figure 3.24 (a) shows such a case.
2. Samples should span a sufficiently large range in order to avoid local distortions in $h(K)$. A typical failure due to this problem can be seen in Figure 3.24 (b).
3. Sample points should not consist of excessively large control perturbations. When the current state is far from the limit cycle, large control perturbations can result in the biped falling over. Including a fall as a sample can lead to a poor model. This situation is illustrated in Figure 3.24 (c).

Sample scaling factors of $k_{fwd} = \pm 5$ degrees for hip pitch and $k_{lat} = \pm 1$ degree for hip roll give acceptable linear models for our biped models over a wide variety of gaits.

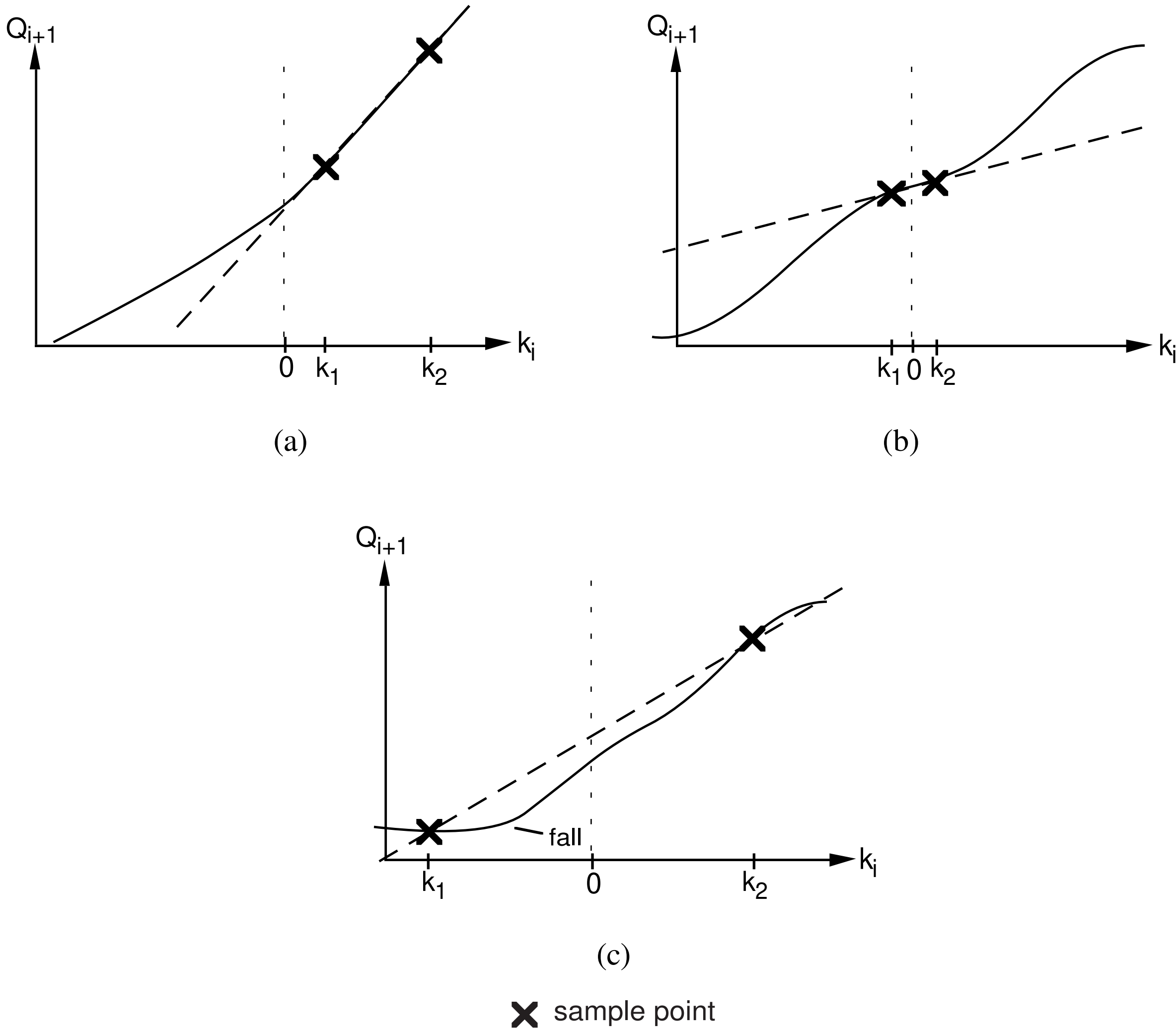


Figure 3.24 - Sampling point pitfalls in discrete model construction
(a) sample scalings non-symmetric
(b) sample scalings too small
(c) sample scalings too large

3.8 Torso Servoing

While it is preferable for animation to have general control solutions, more specific control solutions are sometimes useful or necessary. The control described in previous sections can be used to generate successful walking gaits. However, the resulting walks exhibit a characteristic bobbing[2] of the torso in the sagittal plane. This is due, in part, to the choice of simple stance hip

---

[2] rocking back and forth.

perturbations and, in part, to the fact that the motion is only constrained at a single point in the cycle for each step.

One possible solution to this problem is to use the waist pitch degree-of-freedom of our human model to continuously servo the torso to a desired angle from the vertical in the sagittal plane. The torso servoing control is applied to the biped's waist pitch DOF. The applied torque for a planar example is calculated as

$$\tau = k_p \cdot (\phi - \phi_{desired}) - k_d \cdot \dot{\phi}$$

where

τ is the applied joint torque

$\phi_{desired}$ is the desired torso angle from vertical in the world frame (constant)

ϕ is the torso angle with respect to world frame vertical

$\dot{\phi}$ is the time derivative of ϕ

and

k_p and k_d are the proportional and derivative gains

The parameters τ, $\phi_{desired}$, and ϕ are shown in Figure 3.25

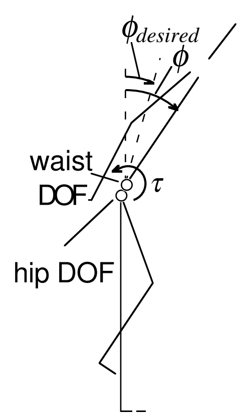


Figure 3.25 - Torso servo parameters

With continuously-applied torso servoing, the torso motion becomes considerably smoother. However, it also presents an additional problem when used with balance control. The application of a torso servo precludes the use of torso-based RVs since these will no longer vary with hip perturbations. Redefining the up vector using a vector fixed to the pelvis as illustrated in Figure 3.26, rather than the torso, is sufficient to overcome this problem.

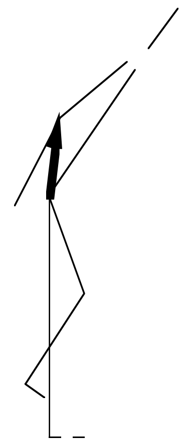


Figure 3.26 - Pelvis-based up vector balance indicator

Torso servoing ensures that the upper body of the biped remains upright, and provides increased stability by damping large upper body movements. However, it should be noted that it does nothing to prevent the biped from falling over since the legs must still support the upper body. An example of a fall with torso servoing applied can be seen in Figure 3.27.

There are a number of benefits in addition to a more aesthetic result. Reducing the highly dynamic oscillations of the torso (a significant portion of the body mass) results in a successful limit cycle for a larger range of Q^d. As well, longer stride times and very slow walks become possible.

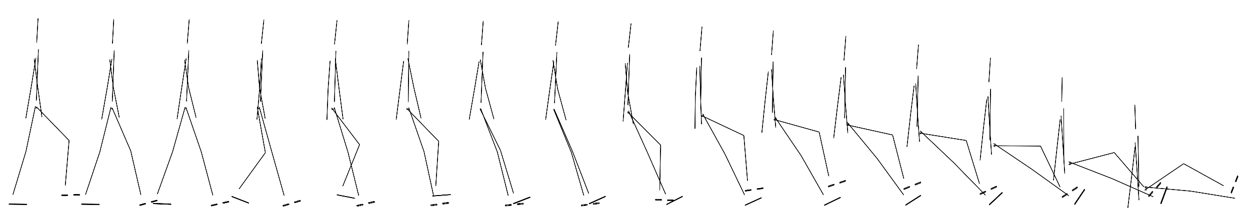


Figure 3.27 - Falling with torso servoing enabled.

3.9 Conclusions

In this chapter, we have described a control system capable of generating and controlling stable 3D bipedal walking limit cycles from unstable open loop motions. A number of key simplifications were made, leading to a well-behaved discrete system which is remodelled for each step using a discrete linear model. The details of each of the system components and their roles in the overall system were described and examples were given for a walking human model.

In the following chapter, we will present some of the results obtained by applying this control strategy to our dynamic models.