

# Quantitative Measures of a Robot’s Physical Ability to Balance

Roy Featherstone

Dept. Advanced Robotics, Istituto Italiano di Tecnologia  
via Morego 30, 16163 Genova, Italy

Final submission to IJRR, RSS special issue, August 23, 2016

## Abstract

This paper presents quantitative measures of a robot’s physical ability to balance itself actively on a single point, line or area of support. These measures express the ratio of a change in the state of motion of the robot’s centre of mass to the amount of action required at the actuated joints in order to produce that change. They therefore represent measures of the gain of the robot mechanism as seen from the point of view of the balance control system. This paper is concerned mainly with ratios of velocities, called *velocity gains*, and it builds on earlier work by showing how these ratios can be defined and calculated for the case of a general planar or spatial robot balancing on a point, line or general rolling contact, or an area contact with a compliant surface. The paper concludes with three examples of use—design of a triple pendulum, analysis of a hydraulic quadruped, and expressing the physics of planar balancing—followed by a short discussion of gyroscopic balancing.

**Keywords:** legged robots, mobile robots, robot design, active balancing, balance control, centre of mass, velocity gain, robot dynamics.

## 1 Introduction

The ability to balance actively on a single point, line or small area of support is something that we take for granted in the natural world (Grizzle et al., 2005; Xinjilefu et al., 2009). Robots used to lack this skill, but this is no longer the case. In the commercial realm, there is the Segway Personal Transporter (Segway, 2015) and an increasing number of low-cost telepresence robots such as the Double (Double Robotics, 2015). In the research laboratories there is the Ballbot (Lauwers et al., 2006), the AcroBOX (Miyashita et al., 2006), the Cubli (Gajamohan et al., 2012), the spatial double inverted pendulum (Xinjilefu et al., 2009), the Acrobot (Hauser and Murray, 1990), and many more.

Active balancing is required whenever the polygon of support shrinks to a point or a line, or becomes very narrow in at least one direction. The robots listed above must all balance actively in order to remain upright. Humanoids, on the other hand, with their large flat feet, are designed not to need this skill. However, even a humanoid must balance

actively if it wants to stand on a convex surface, such as a hump. Active balancing may also be necessary if the supporting surface is compliant.

Robotic balancing presents us with an interesting control problem, but this has led to an imbalance in the way the topic has been treated: too much attention on the control system and not enough on the plant. For example, how good are the above robot mechanisms at balancing? Could they be better? And how does one measure a robot's physical ability to balance? This is a separate issue from the performance of the control system, and has received relatively little attention.

This paper builds on an idea originally proposed in Featherstone (2013), which can be stated in general terms as follows. The objective of the balance controller is to control the motion of the robot's centre of mass (CoM), but the only thing it can control directly is the motion of the actuated joints. Therefore, from the controller's point of view, the robot is a plant in which motion of the CoM is the output and motion of the actuated joints is the input. The performance of the robot can then be characterized by its *gain*—the ratio of output to input. If the gain is large then a relatively small movement of the actuated joints is sufficient to correct a balance error; but if the gain is small then a larger movement is required; and if the gain is zero then an infinite movement is required, meaning that balancing is physically impossible in the current configuration.

The gain can be defined in different ways, depending on how the motions are measured, resulting in several similar but not identical gains. The two that receive most attention in this paper are both ratios of velocities, and are therefore called velocity gains.

Whichever definition is chosen, the robot's gain provides a quantitative measure of the robot's physical ability to balance: it quantifies the amount of motion the robot must make in order to correct a balance error. The smaller the gain, the more work the robot has to do in order to maintain its balance. Likewise, the smaller the gain, the more the robot will wobble in response to sensor noise, and the smaller the minimum disturbance required to cause the robot to fall.

The value of the gain depends on the robot's dynamics, and it is a physical property of the robot mechanism. It is independent of the control law employed by the balance controller, but it does depend on the choice of which joint motions to use for balancing.

This paper is a revised and extended version of Featherstone (2015a). In addition to the main contributions of the earlier paper in presenting general definitions of velocity gain, detailed calculation procedures, and examples of use, this paper presents two more calculation methods, a substantially more detailed discussion of the properties and uses of velocity gains, definitions of the related concepts of momentum and position gain, and a section on gyroscopic balancing.

The rest of this paper is organised as follows: Section 2 defines the linear and angular velocity gain for the special case of a planar double pendulum with one actuated joint, and explains how to calculate them. Section 3 examines some of their properties. Sections 4 and 5 then extend the definitions and calculation details to general planar and spatial mechanisms, respectively, and Section 6 presents further details pertaining to general rolling contact. Then Section 7 presents three examples of practical use: the design of a triple-pendulum robotic balancer, analysis of the balancing ability of an existing hydraulic quadruped, and expressing the physical process of general planar balancing in a form suitable for control. Finally, Section 8 discusses the phenomenon of gyroscopic balancing and how to measure it using gyroscopic acceleration gains.

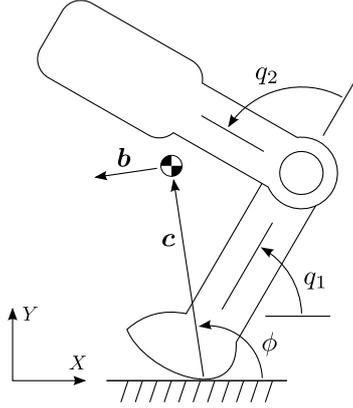


Figure 1: Definition of velocity gain (based on Fig. 1 in Featherstone (2013))

## 2 Velocity Gain

This section introduces the idea of velocity gain via the simple planar self-balancing robot shown in Figure 1. This robot consists of a lower link (link 1), which makes contact with a supporting surface (the ground) at a single point, and an upper link (link 2) which is connected to the lower link via an actuated revolute joint (joint 2, with joint variable  $q_2$ ). For the purpose of calculating the velocity gain, it is assumed that the ground is flat and horizontal, and that the lower link rolls without slipping or losing contact with the ground.

Although the lower link has been drawn as a leg with a foot, it could just as easily be a wheel. Any shape is acceptable, including sharp points, so long as the portion of the link that makes contact with the ground is strictly convex, so that it always makes contact at a single point.

This rolling contact between the lower link and the ground is effectively a one-degree-of-freedom (DoF) joint, which can be characterized by a single joint variable denoting an angle (joint 1, with joint variable  $q_1$  in the diagram). If the robot is balancing on a sharp point then the rolling contact simplifies to a revolute joint at the contact point.

The vector  $\mathbf{c}$ , having components  $c_x$  and  $c_y$ , gives the position of the robot's centre of mass (CoM) relative to the contact point; the angle  $\phi$  gives the direction of  $\mathbf{c}$  relative to the  $x$  axis (or any other datum, if preferred); and  $\mathbf{b} = [-c_y \ c_x]^T / |\mathbf{c}|$  is a unit vector perpendicular to  $\mathbf{c}$  in the direction of increasing  $\phi$ . ( $|\mathbf{c}|$  denotes the magnitude of  $\mathbf{c}$ .)

The objective of a balance controller is either to drive  $c_x$  to zero, or  $\phi$  to  $90^\circ$ , by motions of the actuated joint. Therefore, from the point of view of the balance controller, the robot can be regarded as a plant in which the output is either  $c_x$  or  $\phi$  and the input is  $q_2$ . To define a measure of the instantaneous relationship between input and output, we imagine that an impulse has been applied at joint 2 causing both a step change in  $\dot{q}_2$  and a step change in  $\dot{\mathbf{c}}$ . Two ratios can now be defined as follows:

$$G_v = \frac{\Delta \dot{c}_x}{\Delta \dot{q}_2} \quad \text{and} \quad G_\omega = \frac{\Delta \dot{\phi}}{\Delta \dot{q}_2} = \frac{\mathbf{b} \cdot \Delta \dot{\mathbf{c}}}{|\mathbf{c}| \Delta \dot{q}_2}, \quad (1)$$

in which the quantities  $\Delta \dot{c}_x$ ,  $\Delta \dot{\phi}$  and  $\Delta \dot{q}_2$  all denote step changes in the relevant velocities caused by the impulse at joint 2. Both ratios depend on the robot's configuration, but

not on the applied impulse. As they are both ratios of an output signal to an input signal, they have the nature of gains; and as they are both ratios of velocities, they will be called velocity gains. In particular,  $G_v$  is the robot's *linear velocity gain*, and  $G_\omega$  is its *angular velocity gain*. (Note that Featherstone (2013) defines only the angular velocity gain, but calls it  $G_v$ .)

The reason for applying an impulse at joint 2, rather than a force, is that it simplifies the calculations. Both gains could have been defined as the ratios of step changes in accelerations caused by a *step change* in the torque acting at joint 2, thus:

$$G_v = \frac{\Delta \ddot{c}_x}{\Delta \ddot{q}_2} \quad \text{and} \quad G_\omega = \frac{\Delta \ddot{\phi}}{\Delta \ddot{q}_2}. \quad (2)$$

There is no difference between the definitions in Eqs. 1 and 2. Note, in particular, that these gains are independent of the gravity and velocity terms in the robot's equation of motion; so they are valid in any gravitational field, and in any state of robot motion. Also, they are defined in any configuration, not only those in which the robot is balanced. In the special case of a balanced configuration, the two gains are related by

$$G_v = -|\mathbf{c}| G_\omega. \quad (3)$$

## 2.1 Calculation Methods

Velocity gains are easily calculated using standard functions. This section presents three general methods, each one adapted to the special case in Figure 1 in which there are only two joint variables. To facilitate the calculations, it is assumed that the following items are available:

1. a data structure *rob* containing a dynamic model of the robot (kinematic and inertia parameters, connectivity and joint type data);
2. a function  $\text{jsim}(\text{rob}, \mathbf{q})$  to calculate the joint-space inertia matrix  $\mathbf{H}$  of the mechanism described by *rob* in the configuration  $\mathbf{q}$ ; and
3. a function  $[\mathbf{c}, \dot{\mathbf{c}}] = \text{cmpv}(\text{rob}, \mathbf{q}, \dot{\mathbf{q}})$  to calculate  $\mathbf{c}$  and  $\dot{\mathbf{c}}$ , expressed in 2D world coordinates ( $y$  points up), given *rob*,  $\mathbf{q}$  and a vector of joint velocity variables  $\dot{\mathbf{q}}$ .

### Method I: Direct

Using the robot's impulsive equation of motion (Featherstone, 2008; Featherstone and Orin, 2008), we first calculate the velocity step at the passive joint (joint 1) corresponding to a unit step at the actuated joint (joint 2), both steps being caused by an impulse  $\iota_2$  at the actuated joint. For the mechanism in Figure 1 the impulsive equation is

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \Delta \dot{q}_1 \\ \Delta \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \iota_2 \end{bmatrix}, \quad (4)$$

and setting  $\Delta \dot{q}_2 = 1$  gives

$$\Delta \dot{q}_1 = \frac{-H_{12}}{H_{11}}. \quad (5)$$

Then  $\mathbf{c}$  and  $\Delta\dot{\mathbf{c}}$  are calculated using

$$[\mathbf{c}, \Delta\dot{\mathbf{c}}] = \text{cmpv}(\text{rob}, \mathbf{q}, \begin{bmatrix} \Delta\dot{q}_1 \\ 1 \end{bmatrix}), \quad (6)$$

and the gains are

$$G_v = \Delta\dot{c}_x \quad \text{and} \quad G_\omega = \Delta\dot{\phi} = \frac{\mathbf{b} \cdot \Delta\dot{\mathbf{c}}}{|\mathbf{c}|}. \quad (7)$$

### Method II: CoM Jacobian

The CoM Jacobian is a matrix that maps  $\dot{\mathbf{q}}$  to the velocity of the CoM (Sugihara et al., 2002). If the robot is balancing on a sharp point, rather than a rolling contact, then  $\dot{\mathbf{c}}$  is the velocity of the CoM; so we have  $\dot{\mathbf{c}} = \mathbf{J}\dot{\mathbf{q}}$ , and therefore also  $\Delta\dot{\mathbf{c}} = \mathbf{J}\Delta\dot{\mathbf{q}}$ , where  $\mathbf{J}$  denotes the CoM Jacobian. Equation 6 can therefore be replaced with

$$\Delta\dot{\mathbf{c}} = \mathbf{J} \begin{bmatrix} \Delta\dot{q}_1 \\ 1 \end{bmatrix} \quad (8)$$

plus some means of calculating  $\mathbf{c}$ . This method illustrates the fundamental difference between velocity gain and the CoM Jacobian: the latter is a general mapping from joint to CoM velocity vectors, whereas the former is a function of one particular joint velocity vector: the one that describes how the robot behaves when an impulse is applied at the actuated joint only, causing a unit-magnitude step in the velocity of that joint.

### Method III: Augmented Inertia Matrix

Let  $\text{rob}'$  denote a modified version of  $\text{rob}$  in which a fictitious prismatic joint parallel to the  $x$  axis has been inserted between the foot and the ground. To avoid renumbering the other joints, let us call it joint 0. This joint never moves, and therefore has no effect on the dynamic behaviour of the robot, but it does have an effect on the impulsive equation of motion, which now reads

$$\begin{bmatrix} H_{00} & H_{01} & H_{02} \\ H_{10} & H_{11} & H_{12} \\ H_{20} & H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \Delta\dot{q}_1 \\ \Delta\dot{q}_2 \end{bmatrix} = \begin{bmatrix} \iota_0 \\ 0 \\ \iota_2 \end{bmatrix}. \quad (9)$$

Because joint 0 lies between the foot and the ground, it follows that the ground-reaction impulse must pass through joint 0. This, in turn, implies that  $\iota_0$  must be the  $x$  component of the linear ground-reaction impulse, which means that  $\iota_0 = m\Delta\dot{c}_x$  where  $m$  is the total mass of the robot. Setting  $\Delta\dot{q}_2 = 1$  as before, we find that  $\Delta\dot{q}_1$  is still given by Eq. 5, but we now have a new way to calculate  $G_v$ :

$$G_v = \frac{\iota_0}{m} = \frac{1}{m}(H_{01}\Delta\dot{q}_1 + H_{02}\Delta\dot{q}_2) = \frac{1}{m}\left(H_{02} - \frac{H_{01}H_{12}}{H_{11}}\right). \quad (10)$$

This formula is particularly useful for calculating the linear velocity gain. To calculate the angular velocity gain by this method, the fictitious joint is defined to be a 2-DoF joint allowing translations in the  $x$  and  $y$  directions, so that  $\iota_0$  is a 2D vector with  $x$  and  $y$  components from which both  $\Delta\dot{c}_x$  and  $\Delta\dot{c}_y$  can be calculated.

## 2.2 Momentum Gain

An obvious alternative to a velocity gain is a ratio of two momenta, or, to be more precise, a ratio of a change in momentum to the impulse that causes it. In analogy with linear velocity gain, we can define linear momentum gain as the ratio of a change in the horizontal component of a robot's total linear momentum to the impulse at the actuated joint that causes it. Likewise, in analogy with angular velocity gain, we can define angular momentum gain as the ratio of a change in the *moment of momentum* of the robot's CoM about the support point to the impulse at the actuated joint that causes it.

Observe the careful definition of angular momentum gain: it is not defined in terms of the robot's total angular momentum about the support point because the actuated joint has no instantaneous influence over this quantity, so any such gain would always have the value zero. Instead, it is defined in terms of the component of the total angular momentum which is due to the motion of the CoM.

These definitions imply the following formulae for the linear and angular momentum gains of the robot in Figure 1:

$$G_m = \frac{m\Delta\dot{c}_x}{\iota_2} = mG_v \frac{\Delta\dot{q}_2}{\iota_2} \quad (11)$$

and

$$G_o = \frac{mc^2\Delta\dot{\phi}}{\iota_2} = mc^2G_\omega \frac{\Delta\dot{q}_2}{\iota_2}, \quad (12)$$

where  $G_m$  and  $G_o$  denote the linear and angular momentum gains, respectively,  $mc^2$  is the moment of inertia of the CoM about the support point (i.e., the moment of inertia that the robot would have if all of its mass were concentrated at the CoM), and

$$\frac{\Delta\dot{q}_2}{\iota_2} = \frac{H_{11}}{\det(\mathbf{H})} \quad (13)$$

(obtained by solving Eq. 4 for  $\Delta\dot{q}_2$  with  $\iota_2 = 1$ ). Note that  $G_m$  is also given by the ratio  $\iota_0/\iota_2$  from Eq. 9.

These equations show that  $G_m$  is a strictly positive multiple of  $G_v$ , and that  $G_o$  is a strictly positive multiple of  $G_\omega$  everywhere except  $\mathbf{c} = \mathbf{0}$ . The momentum gains therefore have the same signs and the same zero-crossing points as the corresponding velocity gains. In fact, momentum gains are so similar to velocity gains that they do not appear to offer any useful new property or insight beyond what is already available from velocity gains. So there is no objective reason to prefer them over velocity gains, and they are mentioned here mainly for the sake of completeness.

## 3 Properties and Examples

Velocity gains measure the degree of influence that motion of the actuated joint has over motion of the CoM: the larger the magnitudes of these gains, the greater the influence, and therefore the better the robot is at balancing. If the gains are zero at some particular configuration then the robot is physically incapable of balancing at that configuration.

The two gains have slightly different properties, and the choice of which one to use depends on circumstances. For example,  $G_\omega$  is not defined when  $\mathbf{c} = \mathbf{0}$ , so it is not

appropriate in situations where the CoM could coincide with the support point. Likewise,  $G_v$  does not distinguish between  $c_y > 0$  and  $c_y < 0$ , so it might not be appropriate in applications like swing-up control, where the CoM starts below the support point but must finish above it. However, in the context of a robot balancing on a horizontal supporting surface, we can safely assume that  $c_y > 0$  so that neither of these problems arise. Practical experience with these gains suggests that  $G_v$  is more useful for studying balance control systems and the physical process of balancing, while  $G_\omega$  is more useful for designing and analysing robot mechanisms.

### Geometric Effects

Neither  $G_\omega$  nor  $G_v$  is defined where there is a step change in the local curvature at the contact point because this causes a step change in the values of the velocity gains. (See Section 6.) This problem also arises if there is a step change in the curvature of the supporting surface, which is why we have assumed that it is flat. However, the shape of the supporting surface becomes progressively less relevant as the foot gets smaller, and is completely irrelevant if the robot is standing on a point foot. The instantaneous kinematics of a rolling contact depends on the sum of the curvatures of the contacting surfaces; so the effect of a curved supporting surface can easily be incorporated into the calculation of velocity gain if required.

If the radius of curvature is greater than  $|\mathbf{c}|$  in some neighbourhood of an equilibrium configuration then it is a stable equilibrium; otherwise it is unstable and active balancing is required to prevent the robot from falling. A robot may possess both stable and unstable equilibrium configurations. As the curvature approaches zero, the rolling contact approaches an area contact and the velocity gains approach infinity.

### Zero Crossings

At a balanced configuration, the zeros of  $G_v$  and  $G_\omega$  coincide (Eq. 3). In the neighbourhood of a zero crossing, balancing is impossible because both a positive and a negative motion of the actuated joint will correct a balance error in one direction, but no motion can correct an error in the other direction. An example is shown in Figure 2. This problem can be overcome if the robot has more than one actuated joint and at least one joint (or combination of joints) has a usable velocity gain.

Velocity gains typically undergo a zero crossing as the foot's radius of curvature is increased from zero towards infinity; and this typically happens before the transition from unstable to stable equilibrium. Thus, there will often be a range of foot curvatures that are worse for balancing than a point foot.

### Area Contact

In an area contact, the ground reaction forces define a centre of pressure (CoP) that can be used in place of the contact point in the definition of velocity gain. So  $\mathbf{c}$  in Figure 1 would be redefined to point from the CoP to the CoM.

If the foot and the supporting surface are both rigid then the velocity gains are infinite because zero movement is required at the actuated joint to cause a finite displacement of the CoP, and therefore a finite displacement of the CoM relative to the CoP. In this case,

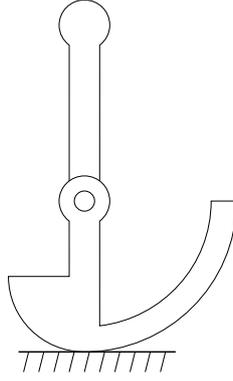


Figure 2: This robot's foot has a step change in curvature at the contact point such that  $G_\omega < 0$  to the left and  $G_\omega > 0$  to the right. As a result, any movement of the actuated joint can stop the robot from falling to the right, but no movement can stop it from falling to the left

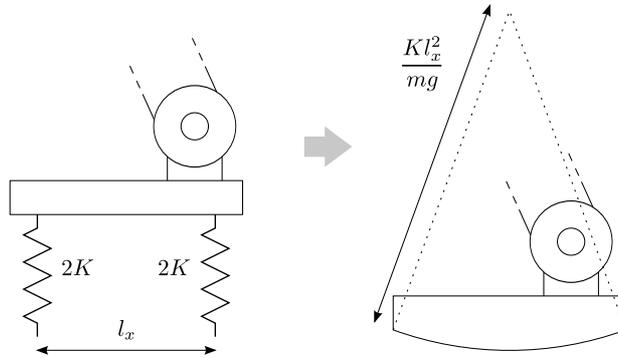


Figure 3: The equivalence between a flat foot resting on a compliant surface and a curved foot on a rigid surface

one could imagine using a ‘force-position gain’ (a ratio of a position to a force) instead of a velocity gain because there is now a well-defined relationship between actuated joint torque and CoP position.

However, if there is compliance at the contact then it defines a relationship between the location of the CoP and the orientation of the foot that resembles the behaviour of the contact point in a rolling contact. The effect is illustrated in Figure 3, which shows the foot (or two feet acting as one) of a humanoid standing on a compliant surface modelled as a set of four springs of stiffness  $K$  (shown as two springs of stiffness  $2K$ ) located at the corners of a rectangle. If the foot is rotated by a small angle  $\theta$  in the plane of the drawing then the CoP moves by an amount  $Kl_x^2\theta/mg$ , where  $m$  is the mass of the robot and  $g$  is the gravitational acceleration. This is the same movement that one would get from rotating a curved foot with a radius of curvature of  $Kl_x^2/mg$ , so the two situations are equivalent from a balancing point of view. If the rectangle has dimensions  $l_x \times l_y$  then the equivalent radius of curvature in the other direction is  $Kl_y^2/mg$ .

In this context, it does not matter whether the compliance is physically located in the ground, the foot, the ankle, or any combination of the three. Tangential compliance has not been considered here on the grounds that it is likely to have less effect than the

interplay between normal compliance and foot angle.

The remark above about zero crossings and foot curvature also applies in this case; so the possibility exists that there could be a range of compliances for which it is harder to balance on a flat foot than it would be to balance on a point foot on firm ground.

### Position Gain

If velocity gain is integrated along a path in the robot's configuration space, then the result provides an estimate of the cumulative effect on the CoM of moving the robot along that path. It is only an estimate because it does not take into account the effects of gravity or the robot's initial velocity. Nevertheless, it can provide a designer with an indication of the effect of an end-to-end movement of a particular joint, or other similar information, which can be used to help design a robot. An integral of a velocity gain can be called a position gain. Whereas velocity gains are properties of particular configurations, position gains are properties of particular paths.

### Parameter Variations

It can be seen from the definitions in Eq. 1 that  $G_\omega$  is a dimensionless quantity, whereas  $G_v$  has the physical dimension of length. It therefore follows that both gains are invariant with respect to a uniform scaling of all masses in the robot, and that  $G_\omega$  is invariant with respect to a uniform scaling of all lengths. These properties carry over into the general case provided that the joint variables of the joints used for balancing are all angles. The gains are also independent of the robot's velocity variables and the strength of the gravitational field. A useful special property of  $G_\omega$  is that if the robot is balancing on a sharp point then  $G_\omega$  is independent of  $q_1$ . This property carries over to the case of a general robot balancing on a knife edge.

Although the velocity gains do depend on the mass distribution in general, they are invariant with respect to certain specific redistributions of mass within the robot mechanism. For example, at any revolute joint within the mechanism, a particle of mass  $m$  can be added to one of the two bodies connected by the joint, and a particle of mass  $-m$  added to the other body, provided that both particles are coincident and located on the joint's rotation axis (see Featherstone, 2008, §9.7). If the robot is balancing on a sharp point or edge then a point mass can be added (or subtracted) at the contact point, or anywhere along the line of the contact edge, without affecting the value of  $G_\omega$ . On its own, this modification does affect the value of  $G_v$ , but a combination of a point mass at the foot and a uniform scaling of lengths can be found that leaves  $G_v$  unaffected.

The freedom to vary parameters in this way means that if a designer can identify a single mechanism with a desirable balancing property then a multi-parameter family of other mechanisms with the same property can immediately be generated simply by varying the kinematic and inertia parameters in ways that have no effect on the velocity gains.

## 3.1 The Special Case of a 2R Mechanism

The case of a planar double pendulum balancing on a sharp point is especially simple, and is amenable to more detailed analysis. This balancer is equivalent to a planar 2R

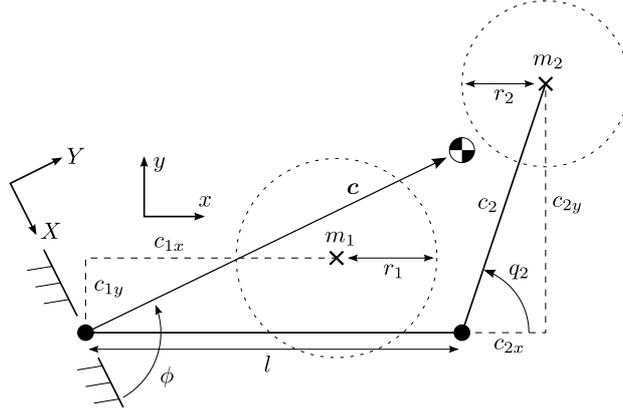


Figure 4: Parameters of the planar 2R balancer

mechanism with a passive first joint, and it can be characterized by the eight parameters  $m_1, r_1, c_{1x}, c_{1y}, m_2, r_2, c_2$  and  $l$ , as shown in Figure 4. ( $r_1$  and  $r_2$  are radii of gyration—the rotational inertia of each link about its CoM is  $m_i r_i^2$ .) The formulae for its velocity gains are

$$G_v = \frac{c_Y H_{12}}{H_{11}} - \frac{m_2 c_{2Y}}{m_1 + m_2} \quad (14)$$

and

$$G_\omega = \frac{m_2 \mathbf{c} \cdot \mathbf{c}_2}{(m_1 + m_2) c^2} - \frac{H_{12}}{H_{11}}, \quad (15)$$

where  $c_Y$  and  $c_{2Y}$  are the  $y$  components of  $\mathbf{c}$  and  $\mathbf{c}_2$  expressed in 2D world coordinates. These formulae are derived in Appendix A.

The formulae simplify in various ways if the parameters take special values. The most important of these is when  $c_2 = 0$ , in which case link 2 is acting as a flywheel. The formulae for this case are

$$G_v = \frac{c_Y H_{12}}{H_{11}} \quad \text{and} \quad G_\omega = \frac{-H_{12}}{H_{11}}, \quad (16)$$

where  $H_{12}/H_{11}$  is now a constant equal to the ratio of the rotational inertia of the flywheel to the rotational inertia of the whole robot about the foot (see Appendix A). So  $G_\omega$  is a constant in this case.

As this balancer has only a single internal joint, there are a total of four freedoms to vary the parameters without affecting  $G_\omega$ : uniform scalings of mass and length, an arbitrary point mass at the foot, and a pair of equal and opposite point masses at joint 2. The first and last of these also leave  $G_v$  unaffected, as well as a combination of the second and third. So  $G_\omega$  is effectively a function of four parameters and one joint variable, whereas  $G_v$  is a function of five parameters and two joint variables. This greater simplicity of  $G_\omega$  explains why it is a better choice for designing and analysing mechanisms.

Various simplifications can be applied to this mechanism without affecting  $G_\omega$ . For example, if  $c_2 \neq 0$  then, by adding a negative mass of  $-m_2 r_2^2 / (c_2^2 + r_2^2)$  to link 2 at joint 2 and the corresponding positive mass to link 1, we obtain  $r_2 = 0$  in the modified mechanism. Thus, without loss of generality, if  $c_2 \neq 0$  then we can model link 2 as a point mass on a stick. In like manner, if  $c_1 \neq 0$  then adding a negative mass of  $-m_1 r_1^2 / (c_1^2 + r_1^2)$

to link 1 at the foot will reduce  $r_1$  to zero. So it will usually be possible to assume  $r_1 = r_2 = 0$  without loss of generality. The only common exception is if  $c_2 = 0$  in the original mechanism, in which case other simplifications are possible, as shown in Eq. 16.

### 3.2 An Impossible Balancer

To illustrate the role that velocity gains can play in the process of discovery, consider the following problem: is it possible for a 2R balancer to be completely impossible to balance; that is, to have zero velocity gain everywhere? In the absence of the concept of velocity gain, a question like this might not even be posed. To attack this problem, we choose  $G_\omega$  over  $G_v$  because it is already independent of  $q_1$ , thereby simplifying the problem to one of finding a mechanism with  $G_\omega = 0$  for every value of  $q_2$ .

A trivial solution to this problem consists of a massless link 1 and a point mass on a stick for link 2. As all the mass is concentrated at a single point, the ground reaction force must pass through this point. Therefore, if the robot is initially at rest, and ignoring the effect of gravity, movement of joint 2 can produce only a radial movement of the CoM relative to the foot, which alters the length of  $\mathbf{c}$  but not its direction.

To this mechanism we can add an arbitrary point mass at the foot without affecting  $G_\omega$ . Furthermore, bearing in mind the mass and scale invariance of  $G_\omega$ , we can choose  $m_1 = m_2 = 1$  and  $l = 1$  without loss of generality. We now have a mechanism in which both links have mass, but this mechanism is not yet realizable because each link's mass is concentrated at a single point.

To solve this problem, we have to assign positive values to  $r_1$  and  $r_2$ . This can be done by first setting  $c_2 = l$ , so that the two point masses and joint 2 form an isosceles triangle with the joint at the apex and the base on the same line as  $\mathbf{c}$ . The mechanism is now symmetrical about the perpendicular bisector of the base. Given that the base does not change direction as  $q_2$  varies, it follows that the angular velocities of the two links are always equal and opposite. This, in turn, implies that if we add equal amounts of rotational inertia to both links then the angular momenta associated with these inertias will always cancel exactly, implying that they have no net effect on the orientation of  $\mathbf{c}$ .

We now have a one-parameter family of physically realizable mechanisms having the property that  $G_\omega = 0$  everywhere. Their parameters are:  $m_1 = m_2 = 1$ ,  $l = c_2 = 1$ ,  $c_{1x} = c_{1y} = 0$ ,  $r_1 = r_2$ , and  $r_1$  can be chosen arbitrarily. Add to this the four freedoms to vary the parameters of a 2R balancer without affecting  $G_\omega$  and we have a five-parameter set of physically realizable ‘impossible’ balancers. In particular, by adding positive mass to link 1 at joint 2, and corresponding negative mass to link 2, it is possible to draw the CoM of link 1 away from the foot, so that  $c_{1x} > 0$ .

An example of such a machine is shown in Figure 5. There are mass concentrations in the two side lobes and the tip of link 2, but there is also distributed mass throughout the structure. The side lobes are needed in order to obtain a relatively large value of  $r_1$  in combination with a relatively small value of  $c_{1x}$ . Other physical realizations are possible.

## 4 Extension to Multiple Actuated Joints

Consider a general planar robot having  $n$  DoF, which includes the passive contact with the ground. The vector of position variables is partitioned into passive and actuated

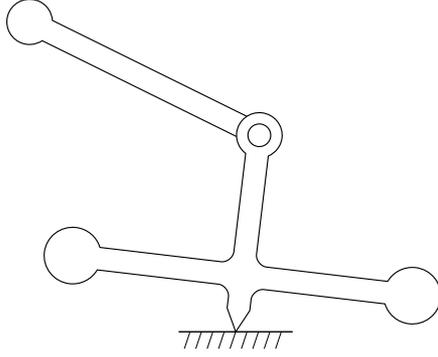


Figure 5: An ‘impossible’ balancer

variables as follows:  $\mathbf{q} = [\mathbf{q}_p^T \mathbf{q}_a^T]^T$ , where  $\mathbf{q}_p = [q_1]$  contains just the angle of joint 1, as defined previously, and  $\mathbf{q}_a = [q_2 \ q_3 \ \dots \ q_n]^T$  contains the variables for the internal motion freedoms of the mechanism, which are assumed to be fully actuated. If the mechanism contains passive motion freedoms, such as a spring-loaded joint, then it is assumed that the amount of movement in these joints is small enough to be ignored.

If the mechanism is a kinematic tree then  $\mathbf{q}_a$  is the vector of joint position variables; otherwise it is a vector of generalized coordinates from which the joint variables can be calculated, in which case  $\mathbf{H}$  is a generalized inertia matrix. Instructions for calculating such matrices can be found in several textbooks (e.g., Featherstone, 2008),<sup>1</sup> and it is assumed that a function `gim(rob, q)`, analogous to `jsim`, is available to calculate this matrix. We shall use `gim` in place of `jsim` from here on.

From a balancing point of view, the fundamental difference between a single actuated freedom and multiple actuated freedoms is that in the latter case the control system can choose which motion freedoms to allocate to the task of achieving or maintaining balance. This choice affects the velocity gain, since some movements have more influence on the CoM than others. Indeed, it will usually be possible to find  $n - 2$  directions of motion freedom in which the velocity gain is zero—in other words, an  $(n - 2)$ -dimensional null space of motions that the robot can make without affecting its balance.

To cater for this new possibility, we modify Eq. 7 as follows:

$$G_v(\Delta\dot{\mathbf{q}}_a) = \Delta\dot{\mathbf{c}}_x, \quad G_\omega(\Delta\dot{\mathbf{q}}_a) = \Delta\dot{\phi} = \frac{\mathbf{b} \cdot \Delta\dot{\mathbf{c}}}{|\mathbf{c}|}, \quad (17)$$

where  $\Delta\dot{\mathbf{q}}_a$  is a velocity step chosen by the user, and  $\Delta\dot{\mathbf{c}}$  and  $\Delta\dot{\phi}$  are the steps in  $\dot{\mathbf{c}}$  and  $\dot{\phi}$  resulting from the actuation impulse  $\boldsymbol{\nu}_a$  that causes  $\Delta\dot{\mathbf{q}}_a$ . The gains are then calculated in almost the same way as before. First we solve the impulsive equation of motion for  $\Delta\dot{\mathbf{q}}_p$ :

$$\begin{bmatrix} \mathbf{H}_{pp} & \mathbf{H}_{pa} \\ \mathbf{H}_{ap} & \mathbf{H}_{aa} \end{bmatrix} \begin{bmatrix} \Delta\dot{\mathbf{q}}_p \\ \Delta\dot{\mathbf{q}}_a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\nu}_a \end{bmatrix} \quad (18)$$

giving

$$\Delta\dot{\mathbf{q}}_p = -\mathbf{H}_{pp}^{-1} \mathbf{H}_{pa} \Delta\dot{\mathbf{q}}_a. \quad (19)$$

<sup>1</sup>If  $\mathbf{H}'$  is a joint-space inertia matrix and  $\mathbf{G}$  is a mapping from generalized to joint-space velocity variables then  $\mathbf{H} = \mathbf{G}^T \mathbf{H}' \mathbf{G}$ .

We then calculate  $\mathbf{c}$  and  $\Delta\dot{\mathbf{c}}$  from

$$[\mathbf{c}, \Delta\dot{\mathbf{c}}] = \text{cmpv}(\text{rob}, \mathbf{q}, \begin{bmatrix} \Delta\dot{\mathbf{q}}_p \\ \Delta\dot{\mathbf{q}}_a \end{bmatrix}), \quad (20)$$

and then the gains using Eq. 17.

To understand the definitions in Eq. 17, we imagine that the user has defined a *virtual joint*, with joint variable  $q_v$ , which specifies the movement to be used for balancing. The relationship between  $q_v$  and  $\mathbf{q}_a$  is given by  $q_i = f_i(q_v)$ ,  $i = 2 \dots n$ , where the functions  $f_i$  are chosen by the user. The corresponding relationship between velocity variables is then  $\dot{q}_i = (\partial f_i / \partial q_v) \dot{q}_v$ . A unit-magnitude step in the value of  $\dot{q}_v$  therefore corresponds to a step in  $\dot{\mathbf{q}}_a$  given by

$$\Delta\dot{\mathbf{q}}_a = \begin{bmatrix} \frac{\partial f_2}{\partial q_v} & \frac{\partial f_3}{\partial q_v} & \dots & \frac{\partial f_n}{\partial q_v} \end{bmatrix}^T. \quad (21)$$

So  $\Delta\dot{\mathbf{q}}_a$  in Eq. 17 represents a unit step in a velocity variable defined by the user.

Referring back to Section 2.1, the direct method now consists of calculating  $\Delta\dot{\mathbf{q}}_p$  via Eq. 19, then  $\mathbf{c}$  and  $\Delta\dot{\mathbf{c}}$  via Eq. 20, and then  $G_v$  and  $G_\omega$  via Eq. 17. In the other two methods, Eqs. 8, 9 and 10 become

$$\Delta\dot{\mathbf{c}} = \mathbf{J} \begin{bmatrix} \Delta\dot{\mathbf{q}}_p \\ \Delta\dot{\mathbf{q}}_a \end{bmatrix}, \quad (22)$$

$$\begin{bmatrix} H_{00} & \mathbf{H}_{0p} & \mathbf{H}_{0a} \\ \mathbf{H}_{p0} & \mathbf{H}_{pp} & \mathbf{H}_{pa} \\ \mathbf{H}_{a0} & \mathbf{H}_{ap} & \mathbf{H}_{aa} \end{bmatrix} \begin{bmatrix} 0 \\ \Delta\dot{\mathbf{q}}_p \\ \Delta\dot{\mathbf{q}}_a \end{bmatrix} = \begin{bmatrix} \iota_0 \\ \mathbf{0} \\ \boldsymbol{\iota}_a \end{bmatrix} \quad (23)$$

and

$$G_v(\Delta\dot{\mathbf{q}}_a) = \frac{1}{m}(\mathbf{H}_{0a} - \mathbf{H}_{0p} \mathbf{H}_{pp}^{-1} \mathbf{H}_{pa}) \Delta\dot{\mathbf{q}}_a. \quad (24)$$

One new property of the velocity gains in Eq. 17 is that they are linear in their arguments. Thus, for any two scalars  $\alpha_1$  and  $\alpha_2$ , and any two vectors  $\dot{\mathbf{q}}_1$  and  $\dot{\mathbf{q}}_2$ , we have

$$G_v(\alpha_1 \dot{\mathbf{q}}_1 + \alpha_2 \dot{\mathbf{q}}_2) = \alpha_1 G_v(\dot{\mathbf{q}}_1) + \alpha_2 G_v(\dot{\mathbf{q}}_2) \quad (25)$$

and

$$G_\omega(\alpha_1 \dot{\mathbf{q}}_1 + \alpha_2 \dot{\mathbf{q}}_2) = \alpha_1 G_\omega(\dot{\mathbf{q}}_1) + \alpha_2 G_\omega(\dot{\mathbf{q}}_2). \quad (26)$$

An immediate consequence is that we only need to know the gains for the individual variables in  $\dot{\mathbf{q}}_a$ , since all other gains are just linear combinations of these ones. If we define  $G_{vi}$  and  $G_{\omega i}$  to be the gains associated with variable  $i$  then we can define matrices

$$\mathbf{G}_v = [G_{v2} \quad G_{v3} \quad \dots \quad G_{vn}] \quad (27)$$

and

$$\mathbf{G}_\omega = [G_{\omega 2} \quad G_{\omega 3} \quad \dots \quad G_{\omega n}] \quad (28)$$

such that

$$G_v(\Delta\dot{\mathbf{q}}_a) = \mathbf{G}_v \Delta\dot{\mathbf{q}}_a \quad (29)$$

and

$$G_\omega(\Delta\dot{\mathbf{q}}_a) = \mathbf{G}_\omega \Delta\dot{\mathbf{q}}_a. \quad (30)$$

From Eq. 24 it follows immediately that

$$\mathbf{G}_v = \frac{1}{m}(\mathbf{H}_{0a} - \mathbf{H}_{0p} \mathbf{H}_{pp}^{-1} \mathbf{H}_{pa}). \quad (31)$$

## 4.1 The Choice of Metric

Consider a planar triple pendulum having two actuated joints. How is  $\Delta\dot{\mathbf{q}}_a$  to be chosen for this robot? If only one joint is to be used for balancing then the two obvious possibilities are  $[1\ 0]^T$  and  $[0\ 1]^T$ ; but what if both are to be used in some combination? Suppose, for example, that the two joints are to be used in a ratio  $\alpha : 1$  and the problem is to find the best value for  $\alpha$ . There are now several possibilities for  $\Delta\dot{\mathbf{q}}_a$ , depending on the choice of metric: for example,  $[\alpha\ 1]^T/\beta$  with  $\beta = 1 + |\alpha|$ ,  $\sqrt{1 + \alpha^2}$  or  $\max(1, |\alpha|)$ , which correspond to an unweighted 1-norm, 2-norm and  $\infty$ -norm, respectively.

Choices like these directly affect both the values of the velocity gains and the comparison of one gain with another; so it is important to adopt a disciplined approach to choosing a metric. The choice must be justifiable in some objective sense, otherwise the computed values could be viewed as arbitrary.

Although it is not possible to anticipate every circumstance in which a velocity gain might be used, it is possible to give some general guidelines, as follows. If the main concern is with joint position or velocity limits, and each joint's limits are independent of the others, then an  $\infty$ -norm is appropriate because each joint can reach its motion limits independently but their effects on the CoM sum. So moving two joints simultaneously at their maximum velocities will have a bigger effect than moving only one, and therefore should be given a larger velocity gain. Alternatively, if the main concern is with energy consumption, and the energy cost of moving a joint is linearly proportional to its velocity, then the 1-norm is appropriate because it scales the motion according to its energy cost. If the energy cost were proportional to the square of joint velocity then the 2-norm should be used instead. Another circumstance where the 2-norm is appropriate is if balance performance is being considered alongside other performance measures, such as manipulability indices, which are usually defined using 2-norms so as to give rise to ellipsoids. In this case, using a 2-norm in the calculation of velocity gain puts it on the same footing as the other performance measures.

Another issue to consider is the weighting of the norm. In general, it is preferable to use a weighted norm because it can account more accurately for the physical attributes of the robot. Obvious possibilities include weighting according to the position or velocity limits of each joint, or according to the energy cost of moving each joint.

Yet another issue to consider is the effect of a nonlinear transmission from an actuator to a joint. If the actuator has a fixed maximum speed, but the transmission ratio varies with joint angle, then the joint has a configuration-dependent maximum speed. This suggests that weights should, in general, be allowed to vary with the robot's configuration.

## 5 Extension to 3D

In 3D the foot can make either a point or a line contact with the ground, the latter being defined as contact at two or more points all lying on a single straight line. Any greater degree of contact results in a polygon of support, and is equivalent to an area contact. The case of a line contact can be handled using equations identical to the ones in the previous section; so this section covers only the case of a point contact.

Consider a general spatial mechanism making a single point contact with a horizontal supporting surface located in the  $x$ - $y$  plane of a Cartesian coordinate system with the  $z$

axis pointing up (3D world coordinates). The link that makes this contact (the leg) is able to roll without slipping in both the  $x$  and  $y$  directions, and it is able to spin about the contact normal. It therefore has three degrees of instantaneous motion freedom relative to the supporting surface.

Rolling in 3D is a well-known example of a nonholonomic constraint, which we can characterize with five position variables,  $q_x$ ,  $q_y$ ,  $q_1$ ,  $q_2$  and  $q_3$ , and three independent velocity variables,  $\dot{q}_1$ ,  $\dot{q}_2$  and  $\dot{q}_3$ , where  $q_1$ ,  $q_2$  and  $q_3$  are any set of Euler angles describing the orientation of the leg, and  $q_x$  and  $q_y$  give the coordinates of the contact point. The dependent variables  $\dot{q}_x$  and  $\dot{q}_y$  must be calculated from the values of the other variables and a description of the shape of the foot. If the robot is balancing on a sharp contact point then the rolling contact simplifies to a spherical joint and  $\dot{q}_x = \dot{q}_y = 0$ .

Overall, the robot has  $n$  degrees of velocity freedom, which can be partitioned into 3 passive DoF at the contact and  $n - 3$  actuated DoF. We therefore redefine  $\dot{\mathbf{q}}_p = [\dot{q}_1 \ \dot{q}_2 \ \dot{q}_3]^T$  and  $\dot{\mathbf{q}}_a = [\dot{q}_4 \ \cdots \ \dot{q}_n]^T$ . We continue to assume the existence of the data structure *rob*, describing both the robot and its contact with the ground, and the two functions *gim* and *cmpv*, which must now perform their calculations in 3D.  $\mathbf{c}$  and  $\dot{\mathbf{c}}$  are now 3D vectors, and  $\mathbf{b}$  is no longer needed.

The job of a balance controller in 3D is to bring the CoM directly above the support point. This means either driving both  $c_x$  and  $c_y$  to zero, or making  $\mathbf{c}$  point upwards. We therefore define the 3D versions of the linear and angular velocity gains as follows:

$$G_v(\Delta\dot{\mathbf{q}}_a) = \begin{bmatrix} \Delta\dot{c}_x \\ \Delta\dot{c}_y \end{bmatrix} \quad \text{and} \quad G_\omega(\Delta\dot{\mathbf{q}}_a) = \frac{\mathbf{c} \times \Delta\dot{\mathbf{c}}}{|\mathbf{c}|^2}. \quad (32)$$

$G_v$  is now a 2D vector, and  $G_\omega$  is a 3D angular velocity vector perpendicular to both  $\mathbf{c}$  and  $\dot{\mathbf{c}}$  that describes the rate of change of the direction of  $\mathbf{c}$ . These definitions are compatible with their planar counterparts, and simplify to the planar versions if the robot happens to be planar and moving in a vertical plane.

The calculation of these velocity gains follows the same procedure as before: first  $\Delta\dot{\mathbf{q}}_p$  is calculated from the given value of  $\Delta\dot{\mathbf{q}}_a$  using Eq. 19, then  $\mathbf{c}$  and  $\Delta\dot{\mathbf{c}}$  are calculated using Eq. 20, and then the gains are calculated using Eq. 32. Alternatively,  $\Delta\dot{\mathbf{c}}$  can be calculated using Eq. 22; and  $G_v$  can be calculated directly from Eq. 24 provided that joint 0 is defined to be a 2-DoF joint allowing translations in the  $x$  and  $y$  directions.

## 5.1 New Issues in 3D

Although the calculation procedure is almost unchanged, two other aspects of the problem have changed significantly: the gains are now vector-valued, and the robot is now capable of falling in any direction in the horizontal plane. The latter implies that the balance controller must now control two coordinates of the CoM, and therefore must devote two degrees of actuated motion freedom to this task. So the balance controller requires two virtual joints with linearly independent velocity gains.

Instead of a single scalar-valued velocity gain, the balancing performance of the robot is now characterized by a pair of vector-valued velocity gains. How should we extract from this data a single number quantifying the robot's ability to balance? A reasonable answer is illustrated in Figure 6, which shows two linear velocity gains,  $G_1$  and  $G_2$ , plotted in the  $x$ - $y$  plane. It also shows the parallelogram of equal-magnitude combinations of these two

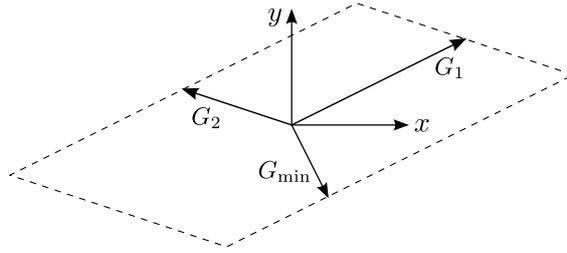


Figure 6: Velocity gain as a function of direction in the  $x$ - $y$  plane

gains according to an  $\infty$ -norm, which is appropriate if the motion limits of the two virtual joints are independent. If a 2-norm is used instead then the parallelogram becomes an ellipse. The figure also shows the shortest vector from the origin to the parallelogram, labelled  $G_{\min}$ . The direction of this vector is the direction in which the robot is least able to balance, and therefore most likely to fall, and so the magnitude of this vector provides a reasonable measure of the robot's ability to balance.

Another issue is that there are now three passive DoF at the contact, but the balance controller is only controlling two of them. The uncontrolled freedom is the freedom of the foot to spin about the contact normal. This motion must be taken into account, because spinning motions do occur during balancing, but there is no need to control these motions for the purpose of achieving and maintaining balance. Indeed, the spin freedom is sometimes described as intrinsically uncontrollable, although this is only correct if spinning is frictionless. The subject of spin control is separate from balance control, and is outside the scope of this paper.

## 6 Rolling Contact Details

This section explains how to implement gim and cmpv for a general rolling contact. The problem is that standard methods for calculating quantities like  $\mathbf{H}$  and  $\mathbf{c}$  expect the robot mechanism to contain only revolute and prismatic joints; so a general rolling contact must be modelled as a chain of revolute and prismatic joints which are subject to a kinematic constraint. In particular, a 2D rolling contact will be modelled as two prismatic joints followed by a revolute joint, and a 3D rolling contact by three prismatic joints followed by three revolute joints. In both cases, the revolute joints are exactly as defined in previous sections, and the prismatic joints are constrained so that their variables depend on the revolute joint variables and the shape of the foot.

We now introduce two new functions, stdH and stdcm, which perform the following calculations using standard methods:

$$\mathbf{H}' = \text{stdH}(\text{rob}, \mathbf{q}') \quad (33)$$

and

$$[\mathbf{c}_o, \Delta \dot{\mathbf{c}}_o] = \text{stdcm}(\text{rob}, \mathbf{q}', \Delta \dot{\mathbf{q}}'), \quad (34)$$

where  $\mathbf{q}'$  is the given value of  $\mathbf{q}$  (i.e., the value supplied to gim or cmpv) augmented with the necessary number of prismatic joint variables;  $\mathbf{H}'$  and  $\Delta \dot{\mathbf{q}}'$  are the similarly-augmented generalized inertia matrix and velocity step vector, respectively; and  $\mathbf{c}_o$  and

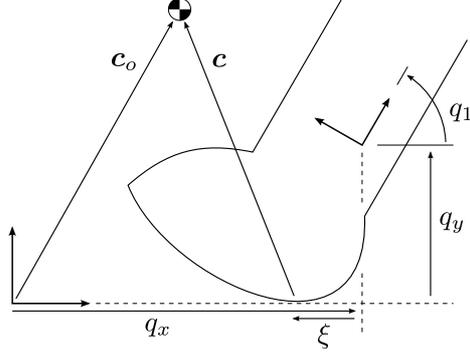


Figure 7: Kinematics of rolling in 2D

$\Delta \dot{\mathbf{c}}_o$  are the position and velocity steps of the CoM relative to the origin of the world coordinate frame (i.e., not the contact point). To calculate the kinematics of the rolling contact, we introduce two more functions, roll2 and roll3, which are described below.

## Rolling in 2D

Figure 7 shows the kinematic model of a planar rolling contact. A coordinate frame is fixed in the leg, and the three variables  $q_x$ ,  $q_y$  and  $q_1$  give the position and orientation of this frame relative to the 2D world coordinate frame, which is positioned so that its  $x$  axis lies on the support surface. A fourth variable,  $\xi$ , serves to locate the contact point. The kinematics are calculated by calling roll2 as follows:

$$[q_x, q_y, \xi, X, Y, \Xi] = \text{roll2}(\text{rob}, q_1), \quad (35)$$

where  $X = dq_x/dq_1$ ,  $Y = dq_y/dq_1$  and  $\Xi = d\xi/dq_1$ . With this data, gim calculates  $\mathbf{H}$  using the formula

$$\mathbf{H} = \mathbf{G}^T \mathbf{H}' \mathbf{G}, \quad (36)$$

where  $\mathbf{H}'$  is calculated using Eq. 33 with  $\mathbf{q}' = [q_x \ q_y \ \mathbf{q}^T]^T$ , and

$$\mathbf{G} = \begin{bmatrix} X & \mathbf{0} \\ Y & \mathbf{0} \\ 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{(n-1) \times (n-1)} \end{bmatrix}. \quad (37)$$

Likewise,  $\text{cmpv}(\text{rob}, \mathbf{q}, \Delta \dot{\mathbf{q}})$  calculates  $\mathbf{c}$  and  $\Delta \dot{\mathbf{c}}$  via Eq. 34 with

$$\Delta \dot{\mathbf{q}}' = \begin{bmatrix} X \Delta \dot{q}_1 \\ Y \Delta \dot{q}_1 \\ \Delta \dot{\mathbf{q}} \end{bmatrix}, \quad (38)$$

followed by

$$\mathbf{c} = \mathbf{c}_o - \begin{bmatrix} q_x + \xi \\ 0 \end{bmatrix} \quad (39)$$

( $\xi$  is negative in Figure 7) and

$$\Delta \dot{\mathbf{c}} = \Delta \dot{\mathbf{c}}_o - \begin{bmatrix} (X + \Xi) \Delta \dot{q}_1 \\ 0 \end{bmatrix}. \quad (40)$$

It was mentioned earlier that velocity gains are not defined if there is a step change in foot curvature at the contact point. This effect enters into the calculation via the derivatives  $X$ ,  $Y$  and  $\Xi$ , which depend on the curvature at the contact point, and which undergo a step change in value if there is a step change in curvature.

Although the kinematics of a general rolling contact can be very complicated, it is worth mentioning that if the leg happens to be a circular wheel, and the leg coordinate frame is at the centre of the wheel, then  $\xi = \Xi = Y = 0$  and  $y = X = R$  (the radius of the wheel).

## Rolling in 3D

The calculation procedure for a general rolling contact in 3D follows the same pattern as for a 2D rolling contact, but with some differences in the details. First, the kinematic model now has three prismatic joints, with variables  $q_x$ ,  $q_y$  and  $q_z$ , and the scalar variable  $\xi$  has been replaced by a vector  $\boldsymbol{\xi} = [\xi_x \ \xi_y]^T$  giving the  $x$  and  $y$  coordinates of the contact point relative to the origin of the leg coordinate frame. The kinematics of the contact are computed by

$$[q_z, \boldsymbol{\xi}, \mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Xi}] = \text{roll3}(\text{rob}, \mathbf{q}_p), \quad (41)$$

where  $\mathbf{X} = dq_x/d\mathbf{q}_p$ ,  $\mathbf{Y} = dq_y/d\mathbf{q}_p$ ,  $\mathbf{Z} = dq_z/d\mathbf{q}_p$  and  $\boldsymbol{\Xi} = d\boldsymbol{\xi}/d\mathbf{q}_p$ .  $\mathbf{X}$ ,  $\mathbf{Y}$  and  $\mathbf{Z}$  are now  $1 \times 3$  matrices, and  $\boldsymbol{\Xi}$  is a  $2 \times 3$  matrix.  $\mathbf{H}$  is still given by Eq. 36, but now with

$$\mathbf{G} = \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{Y} & \mathbf{0} \\ \mathbf{Z} & \mathbf{0} \\ \mathbf{1}_{3 \times 3} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{(n-3) \times (n-3)} \end{bmatrix} \quad (42)$$

and  $\mathbf{q}' = [q_x \ q_y \ q_z \ \mathbf{q}^T]^T$  in Eq. 33. The formulae for  $\mathbf{c}$  and  $\Delta\dot{\mathbf{c}}$  are now

$$\mathbf{c} = \mathbf{c}_o - \begin{bmatrix} q_x + \xi_x \\ q_y + \xi_y \\ 0 \end{bmatrix} \quad (43)$$

and

$$\Delta\dot{\mathbf{c}} = \Delta\dot{\mathbf{c}}_o - \begin{bmatrix} \left( \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} + \boldsymbol{\Xi} \right) \Delta\dot{\mathbf{q}}_p \\ 0 \end{bmatrix}, \quad (44)$$

where  $\mathbf{c}_o$  and  $\Delta\dot{\mathbf{c}}_o$  are still given by Eq. 34 (now working in 3D), but with

$$\Delta\dot{\mathbf{q}}' = \begin{bmatrix} \mathbf{X} \Delta\dot{\mathbf{q}}_p \\ \mathbf{Y} \Delta\dot{\mathbf{q}}_p \\ \mathbf{Z} \Delta\dot{\mathbf{q}}_p \\ \Delta\dot{\mathbf{q}} \end{bmatrix}. \quad (45)$$

Observe that roll3 does not calculate  $q_x$  and  $q_y$ . This is because the rolling contact is nonholonomic, and these variables must instead be calculated by integrating  $\dot{q}_x$  and  $\dot{q}_y$ . However, it turns out that this calculation is unnecessary because  $\mathbf{H}$  and  $\mathbf{c}$  do not depend

on these two variables. So it is possible to set  $q_x = q_y = 0$  in the above equations without altering the final results. (The same is also true of  $q_x$  in the planar rolling case.)

Although the kinematics of a general rolling contact are very complicated, if the leg happens to be a sphere, and the leg coordinate frame is located at the centre of the sphere, then  $\boldsymbol{\xi}$ ,  $\boldsymbol{\Xi}$  and  $\boldsymbol{Z}$  are all zero, and  $q_z$  equals the radius of the sphere.

## 7 Practical Examples

This section presents two examples of how velocity gains can be used to design and analyse a robot's balancing performance, and one example of their use in describing the physics of balancing for the purpose of control. Two more design examples can be found in Driessen (2015) and Featherstone (2013). For comparison, an example of designing a balancing machine without the aid of velocity gains can be found in Leavitt et al. (2004).

### 7.1 Design of a Triple Pendulum

A planar triple pendulum balancing on a point is the simplest example of a balancing machine having two actuated joints, and therefore presenting the control system with a choice of motions to use for balancing. This section illustrates the use of velocity gain to design a planar triple pendulum that is good at balancing. Joints 2 and 3 in this mechanism are actuated, and joint 1 is passive.

Figure 8 shows contour plots of  $G_\omega$  for joints 2 and 3, plotted against  $q_2$  from 0 to  $\pi$  and  $q_3$  from  $-\pi$  to  $\pi$ . ( $G_\omega$  is independent of  $q_1$ .) The plots for negative values of  $q_2$  are identical to the ones shown, but rotated by  $180^\circ$  about  $(0, 0)$ . The colours red, orange, yellow and green onwards can be regarded as bad, poor, OK and good. White corresponds to positive values, and the boundary between red and white is the set of configurations where balancing is physically impossible using the chosen joint.

The top two plots refer to an initial design in which all three links have a length of 0.3m and a mass of 0.5kg, the mass being concentrated at a point at the far end of each link. (This is not unrealistic for reasons explained in Section 3.) The bottom two plots refer to an improved design in which the link lengths are 0.2m, 0.25m and 0.35m, and the masses are 0.7kg, 0.5kg and 0.3kg. As  $G_\omega$  is both mass- and scale-invariant, neither the absolute masses nor the absolute link lengths matter: only the ratios are important.

The improved design was obtained by manually exploring the parameter space and comparing plots like those in Figure 8. However, the point here is not to advocate ad-hoc design, but rather to illustrate the kind of feedback that velocity gains can provide to the designer. Automated optimization is an indispensable tool, but optimizing blindly is not a good idea.

In addition to their use in design, graphs like these can be used as maps for a high-level controller, telling it which configurations are good for balancing and which are bad, and telling it also which joints, or combinations of joints, are most effective for maintaining balance in each region of configuration space.

The graphs show that the improved design has better velocity gains in most of its configuration space, although there is a problem area in the vicinity of  $(\pi, 0)$  where the gain changes sign. This is unlikely to be a problem in practice, as most or all of these configurations will be unreachable. To give an idea of what the robot looks like, three

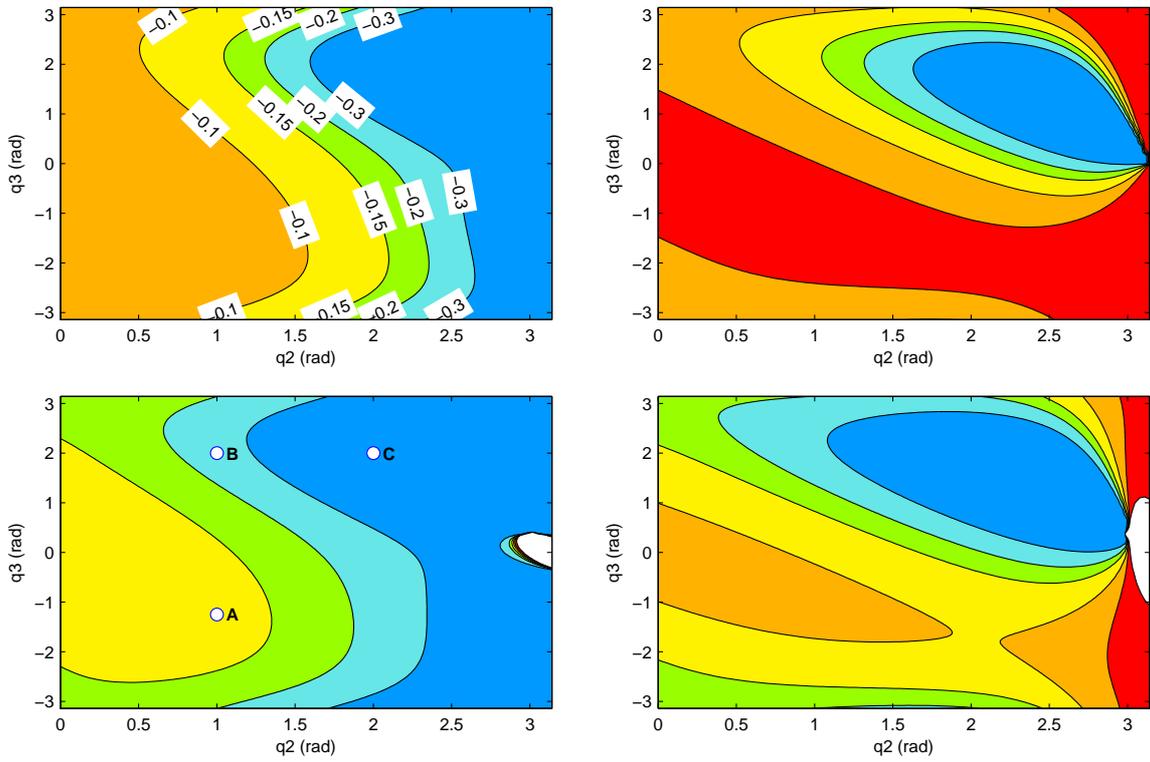


Figure 8: Contour plots of angular velocity gain ( $G_\omega$ ) for joint 2 (left column) and joint 3 (right column) of an initial design (top row) and improved design (bottom row) of planar triple pendulum. Contours: white  $>0$ , red 0 to  $-0.06$ , orange  $-0.06$  to  $-0.1$ , others as shown in the top left plot. A, B, C: configurations shown in Figure 9

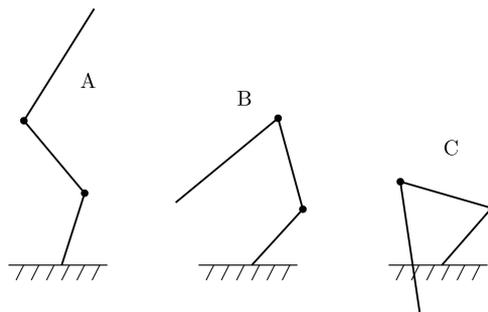


Figure 9: Three specimen configurations of the improved triple pendulum

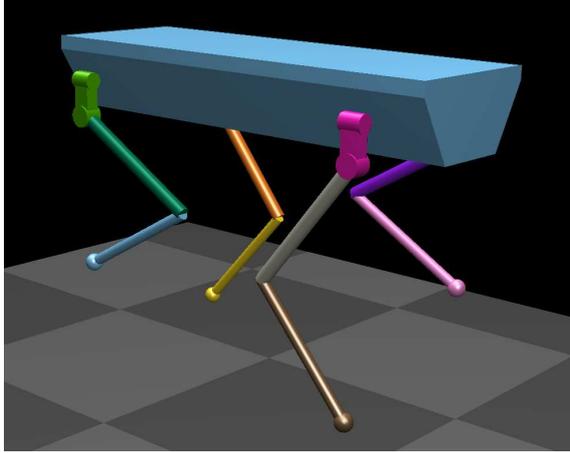


Figure 10: HyQ balancing on diametrically opposite legs (torso height 0.6m, leg configuration  $C$  in Table 1)

configurations have been picked out and shown in Figure 9. It can be seen that configuration  $C$  is unreachable because of interference with the ground. The robot is straight when  $q_2 = q_3 = 0$ .

The graphs also show that joint 2 has a better velocity gain than joint 3 almost everywhere, and that the velocity gain of joint 2 improves with increasing angle. These results are no surprise: joint 2 moves more mass than joint 3, so it is hardly surprising that it has more effect on the CoM; and increasing  $q_2$  folds the robot so as to bring the CoM closer to the support point, so that  $|\mathbf{c}|$  in the denominator of Eq. 17 becomes smaller. The blue regions in the graphs for joint 3 occur at configurations where the mechanism is curled up, like configuration  $C$ , and the CoM is relatively close to the support point. Most of these configurations will be unreachable.

An investigation of this kind can be used to analyze existing robots as well as design new ones. With an existing robot, one cannot alter the joint velocity gains, unless one is prepared to add masses here and there. However, one can compose maps like those in Figure 8 to show which configurations are good for balancing and which are not, and to show which joints, or combinations of joints, are the best to use. Adding mass is not necessarily a bad idea: consider, for example, the balancing pole of a tightrope walker—a relatively small additional mass that improves substantially the artist’s ability to balance. A velocity-gain analysis can show how much mass is needed, where best to put it, and how big an effect it will have.

## 7.2 HyQ Balancing

This section presents an analysis of the balancing ability of the hydraulic quadruped HyQ Semini et al. (2011). It is proposed to make this robot balance on two diametrically opposite legs in postures like the one shown in Figure 10. Specifically, the two raised legs will be held in a fixed configuration, so that they behave like an extension of the torso, and the robot will balance by tipping the torso about a line joining the hip centres (the centres of the top cylinders in the figure) of the two supporting legs. The supporting feet are positioned directly below the hip centres. The available parameters are: the angles

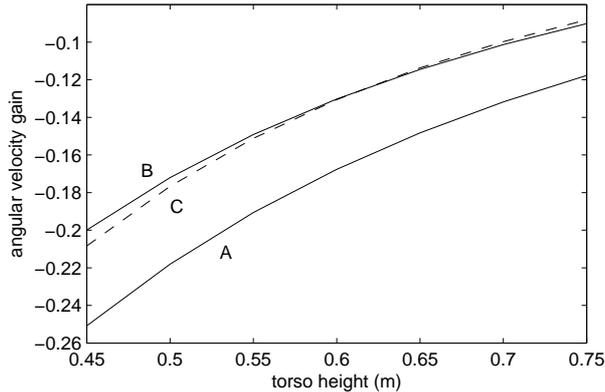


Figure 11: Angular velocity gain versus torso height for the raised-leg configurations listed in Table 1

	left front leg			right hind leg		
A	$-90^\circ$	$10^\circ$	$-20^\circ$	$-90^\circ$	$-10^\circ$	$20^\circ$
B	$-90^\circ$	$70^\circ$	$-140^\circ$	$-90^\circ$	$-70^\circ$	$140^\circ$
C	$0^\circ$	$60^\circ$	$-120^\circ$	$0^\circ$	$-60^\circ$	$120^\circ$

Table 1: Raised-leg configurations

defining the posture of the raised legs, and the height of the torso above the ground. The latter is measured from the origin of the torso coordinate frame, which lies on the line of tipping.

To calculate the velocity gains of this robot, for the purpose of balancing in this particular manner, the first step is to define a kinematic mapping from the operational space of balancing to the joint space of the robot. The latter consists of the 12 revolute joints of the robot mechanism plus the torso’s 6 DoF relative to the ground. The former consists of two virtual revolute joints: a rotation about the line joining the two supporting feet, with joint variable  $\theta_1$ , which represents the robot’s passive freedom to fall over, and a rotation about the line joining the two supporting hip centres, with joint variable  $\theta_2$ , which represents the movement that the robot uses to balance. In effect, the kinematics function maps an inverted double pendulum onto the robot. The calculation of velocity gain then proceeds as explained in Section 2, as if the robot really were a double pendulum, but with `jsim` and `cmpv` basing their calculations on the kinematically-mapped HyQ.

Figure 11 plots angular velocity gain against torso height for the raised-leg configurations listed in Table 1. These configurations are: (A) legs fully extended, pointing sideways; (B) legs fully retracted, pointing sideways; and (C) legs partly retracted, pointing down (as shown in Fig. 10). The angles in this table are all within the HyQ’s joint limits.

The first thing to notice about this graph is that there is a factor of 3 difference between the best and the worst configuration; so the choice of configuration can be expected to have a substantial effect on the quality of the robot’s balancing, and on the risk of falling over. This alone is enough to justify a velocity-gain analysis of this robot.

The next thing to notice is that the main determining factor is torso height, which accounts for more than a factor of two variation in velocity gain over the range considered,

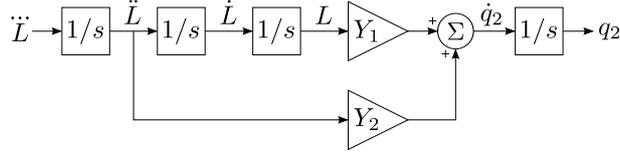


Figure 12: Block diagram expressing the physics of balancing in the plane

whereas extending the raised legs sideways produces only a 20–30% improvement. When the raised legs are not extended, they have little effect. One can deduce from this data that the robot’s safest strategy is to crouch down as much as possible, unload the two non-supporting legs, and then move them directly out to configuration A.

There are two sources of disturbance acting on this robot: IMU sensor noise and unknown forces exerted by its umbilical. The IMU data sheet states a noise magnitude of  $0.5^\circ$ , which will make the robot wobble. At the worst-case velocity gain of 0.08 (ignoring the sign), a  $0.5^\circ$  error would require a  $6^\circ$  correction if the robot could do it instantaneously. However, as the robot cannot move infinitely quickly, the actual correction magnitude will be significantly in excess of  $6^\circ$  because gravity will be acting to exacerbate the balance error during the course of the motion. Furthermore, the control system is aiming not merely to arrest the robot’s fall, but to bring the robot back to its commanded position. This requires tipping the robot in the opposite direction, which means that the control system must necessarily overshoot.

The exact magnitude of the resulting wobbles will depend on many details, but simulation studies on other robots suggest wobble magnitudes of 2 to 3 times the theoretical minimum predicted by the velocity gain, so roughly  $12\text{--}18^\circ$  for the HyQ. Add to this a disturbance from the umbilical, and one can see that there is a significant possibility that the robot will reach its  $30^\circ$  torso rotation limit and lose its balance. In the best-case configuration, the wobbles will be smaller by a factor of 3, and the risk of losing balance will be greatly reduced.

### 7.3 Planar Balance Control

According to Featherstone (2015b), the physical process of balancing on a sharp point in the plane can be expressed in the form of a block diagram like the one shown in Figure 12. The symbols  $q_2$  and  $L$  denote the position variable of the actuated joint and the angular momentum of the whole robot about the support point, respectively, and  $Y_1$  and  $Y_2$  are parameters of the balancing behaviour.

This diagram applies to any planar robot, not just a double pendulum; so it makes the assertion that the balancing behaviour of any planar robot can be characterized by just two numbers. Furthermore, it is shown in Featherstone (2015b) that a very effective balance controller can be obtained simply by treating the block diagram as a plant and closing a feedback loop directly around it. This implies that the feedback gains depend only on  $Y_1$ ,  $Y_2$  and the desired performance level of the controller.

So the two parameters  $Y_1$  and  $Y_2$  play a crucial role both in describing the robot’s balancing behaviour and in controlling it. Their values are given by the formulae

$$Y_1 = \frac{1}{mgT_c^2G_v}, \quad Y_2 = \frac{-1}{mgG_v}, \quad (46)$$

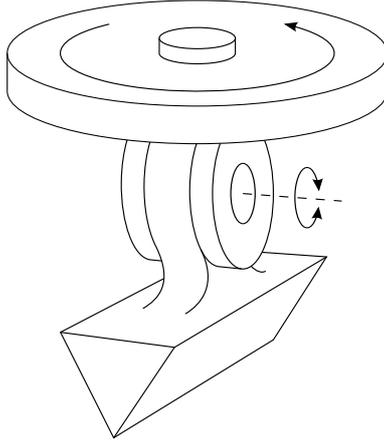


Figure 13: A gyroscopic-force balancer

where  $m$ ,  $g$ ,  $T_c$  and  $G_v$  denote the robot's mass, gravitational acceleration, the robot's time constant of toppling treating it as a single rigid body (i.e., a simple pendulum), and the robot's linear velocity gain in its current configuration. Thus, linear velocity gain plays a central role in both describing and controlling planar balancing behaviour. Clearly, velocity gains also play an important role in 3D balancing, but the details are still a work in progress.

## 8 Gyroscopic Balancing

Throughout this paper it has been assumed that balancing is accomplished by means of inertial forces. However, there is an alternative: balancing by means of gyroscopic forces. Unfortunately, the velocity gains discussed so far cannot measure this type of balancing, so this section explains briefly the reason why, and shows how to define an alternative gain that can measure this kind of balancing.

Although the great majority of robotic balancing is accomplished using inertial forces, there are examples in the literature of robots that employ gyroscopic forces instead or in addition to inertial forces, such as the Gyrover (Brown and Xu, 1997). The big advantage of gyroscopic balancing is that gyroscopic forces can be very large, so a balancer that employs these forces can withstand much larger disturbances than one that uses inertial forces. The Gyrover, for example, can fall over onto its side and still get up again. (The Cubli can also get up from its side, but it uses sudden braking of a momentum wheel, which is an inertial force (Gajamohan et al., 2012).)

An example of a gyroscopic balancer is shown in Figure 13. It consists of a foot that makes a knife-edge contact with the ground, a middle body that is connected to the foot via an actuated revolute joint, and a rotating flywheel connected to the middle body. It resembles a planar inverted double pendulum, but with an important difference: the axis of the actuated joint is at right angles to the line of contact instead of parallel to it. As a result, this machine cannot balance by means of inertial forces, and its angular velocity gain is zero everywhere. Nevertheless, if the flywheel is rotating then this machine can balance.

The explanation for this situation is as follows. Inertial forces arise from accelerations,

whereas gyroscopic forces arise from products of velocities. So, if an impulse is applied at the actuated joint of an inertial-force balancer then it produces a step change in the velocity of the CoM; but an impulse applied to a gyroscopic balancer produces instead a step change in the acceleration of the CoM. This is a fundamental difference, and it implies that a gyroscopic balancer needs a different control system to an inertial balancer. It also implies that velocity gains cannot measure gyroscopic balancing ability because it is a relationship between a velocity and an acceleration rather than between two velocities.

To quantify gyroscopic balancing ability in a device like the one in Figure 13, we could define a gyroscopic acceleration gain as follows:

$$G_{\text{gv}} = \frac{\Delta \ddot{c}_x}{\Delta \dot{q}_2} \quad \text{or} \quad G_{\text{g}\omega} = \frac{\Delta \ddot{\phi}}{\Delta \dot{q}_2} \quad (47)$$

in analogy with Eq. 1. Unfortunately, these gains now include the dimension of time, and they are linearly proportional to flywheel angular velocity. So if a designer wished to compare two designs then it would be necessary to choose the velocities of the two flywheels (or their ratio), and the choice would have to be justifiable in some objective sense. It is a similar problem to the one discussed in Section 4.1. One possibility is to set the velocities equal; another is to set the momenta equal; and a third might be to set the energy costs equal (i.e., the costs of maintaining the two flywheels' spin rates).

A general procedure for calculating gyroscopic acceleration gains is set out below. Note that this procedure accounts for all velocity-dependent forces. To extract the purely gyroscopic effect from the others, repeat the procedure with the flywheel velocity set to zero, and subtract the second result from the first.

1. Calculate  $\Delta \dot{\mathbf{q}}$  resulting from an impulse at the actuated joint such that the step in the actuated joint velocity is unity.
2. Using an inverse dynamics function with gravity and acceleration terms set to zero, calculate the vectors  $\mathbf{C}(\dot{\mathbf{q}})$  and  $\mathbf{C}'(\dot{\mathbf{q}} + \Delta \dot{\mathbf{q}})$  containing the velocity-dependent joint-space forces before and after the step.
3. Solve  $\mathbf{H} \Delta \ddot{\mathbf{q}} = \mathbf{C}' - \mathbf{C}$  to obtain the acceleration step due to velocity effects.
4. Work out the change in CoM acceleration due to  $\Delta \ddot{\mathbf{q}}$ . (Tip: pretend that it is a velocity step and use `cmpv`.)

## 9 Conclusion

This paper has developed and extended the idea of velocity gain, which was proposed originally in Featherstone (2013). Velocity gains provide a means of quantifying a robot's physical ability to balance by expressing it as a ratio of the desired outcome, which is movement of the centre of mass, to the amount of action at the actuated joints that is needed to produce it. The larger this ratio, the less work the robot has to do in order to maintain its balance, and therefore the better it is at balancing.

This paper defines linear and angular velocity gains for general planar and spatial robots balancing on a single point or line, including a general rolling contact, and describes several methods for calculating them via standard dynamics functions. It also discusses

their properties, the related concepts of position and momentum gain, and how to adapt velocity gain to the case of a robot balancing on an area of support.

The paper continues with three examples of use: facilitating the design of a planar triple pendulum so as to be good at balancing, analyzing the balancing ability of an existing hydraulic quadruped, HyQ, in order to choose a good configuration for a balancing experiment, and describing the physical process of balancing for the purpose of balance control. The HyQ example employs a kinematic mapping from the robot's joint space to a virtual double pendulum, thereby illustrating how velocity gains can be employed to analyse proposed behaviours of robots as well as the robots themselves.

The paper concludes with a brief discussion of balancing by means of gyroscopic forces, which is fundamentally different from the inertial-force balancing assumed in the rest of this paper, and requires a special gyroscopic acceleration gain in order to quantify it.

## Funding

This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

## References

- Brown, H. B. Jr., and Xu, Y. (1997). A Single-Wheel Gyroscopically Stabilized Robot. *IEEE Robotics & Automation Magazine*, 4(3):39–44.
- Double Robotics. (2015). ‘Double’ telepresence robot. <http://www.doublerobotics.com>, accessed Dec. 2015.
- Driessen, J. J. M. (2015). *Machine and Behaviour Co-design of a Powerful Minimally Actuated Hopping Robot*. M.Sc. Thesis, Delft University of Technology, Dept. Biomechanical Engineering. Available at: <http://repository.tudelft.nl>
- Featherstone, R. (2008). *Rigid Body Dynamics Algorithms*. New York: Springer.
- Featherstone, R. (2013). Analysis and Design of Planar Self-Balancing Double-Pendulum Robots. In: V. Padois, P. Bidaud & O. Khatib (eds.), *RoManSy 19 — Robot Design, Dynamics and Control*, Vienna: Springer, pp. 259–266.
- Featherstone, R. (2015a). Quantitative Measures of a Robot's Ability to Balance. In: *Proc. Robotics: Science and Systems*, Rome, Italy, July 13–17. Available at: <http://roboticsproceedings.org/rss11/p26.html>
- Featherstone, R. (2015b). A New Simple Model of Balancing in the Plane. *Int. Symp. Robotics Research*, Sestri Levante, Italy, Sept. 12–15.
- Featherstone, R., and Orin, D. E. (2008). Dynamics. In: B. Siciliano & O. Khatib (eds.), *Springer Handbook of Robotics*, Berlin: Springer, pp. 35–65.
- Gajamohan, M., Merz, M., Thommen, I., and D'Andrea, R. (2012). The Cubli: A Cube that can Jump up and Balance. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 7–12, pp. 3722–3727.

- Grizzle, J. W., Moog, C. H., and Chevallereau, C. (2005). Nonlinear Control of Mechanical Systems With an Unactuated Cyclic Variable. *IEEE Trans. Automatic Control*, 50(5):559–576.
- Hauser, J., and Murray, R. M. (1990). Nonlinear Controllers for Non-Integrable Systems: the Acrobot Example. In: *Proc. American Control Conf.*, San Diego, CA, May 23–25, pp. 669–671.
- Lauwers, T. B., Kantor, G. A., and Hollis, R. L. (2006). A Dynamically Stable Single-Wheeled Mobile Robot with Inverse Mouse-Ball Drive. In: *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, FL, May 15–19, pp. 2884–2889.
- Leavitt, J., Bobrow, J. E., and Sideris, A. (2004). Robust Balance Control of a One-Legged, Pneumatically-Actuated, Acrobot-Like Hopping Robot. In: *Proc. IEEE Int. Conf. Robotics and Automation*, New Orleans, LA, April 26 – May 1, pp. 4240–4245.
- Miyashita, N., Kishikawa, M., and Yamakita, M. (2006). 3D Motion Control of 2 Links (5 DOF) Underactuated Manipulator Named AcroBOX. In: *Proc. American Control Conf.*, Minneapolis, MN, June 14–16, pp. 5614–5619.
- Segway Inc. (2015). Personal Transporter. <http://www.segway.com>, accessed Dec. 2015.
- Semini, C., et al. (2011). Design of HyQ – a Hydraulically and Electrically Actuated Quadruped Robot. *Proc. IMechE Part I: J. Systems and Control Engineering*, 225(6):831–849.
- Sugihara, T., Nakamura, Y., and Inoue, H. (2002). Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control. In: *Proc. IEEE Int. Conf. Robots and Automation*, Washington, DC, May 11–15, pp. 1404–1409.
- Xinjilefu, Hayward, V., and Michalska, H. (2009). Stabilization of the Spatial Double Inverted Pendulum Using Stochastic Programming Seen as a Model of Standing Posture Control. In: *Proc. 9th IEEE Int. Conf. Humanoid Robots*, Paris, France, Dec. 7–10, pp. 367–372.

## A Formulae for $G_v$ and $G_\omega$

This appendix derives the formulae for  $G_v$  and  $G_\omega$  for the special case of a planar double pendulum balancing on a sharp point, which is equivalent to a planar 2R mechanism with a passive first joint. Referring back to Figure 4,  $G_v$  can be expressed as the sum of two terms,  $G_{v1}$  and  $G_{v2}$ , where  $G_{vi}$  is the contribution of joint  $i$  to the motion of the CoM. Likewise,  $G_\omega$  can be expressed as the sum of two terms,  $G_{\omega1}$  and  $G_{\omega2}$ . As joint 1 changes only the direction of  $\mathbf{c}$ , not its magnitude, we have

$$G_{v1} = \frac{\partial c_X}{\partial q_1} \Delta \dot{q}_1 = -c_Y \Delta \dot{q}_1 = \frac{c_Y H_{12}}{H_{11}} \quad (48)$$

and

$$G_{\omega1} = \frac{\partial \phi}{\partial q_1} \Delta \dot{q}_1 = \Delta \dot{q}_1 = \frac{-H_{12}}{H_{11}} \quad (49)$$

(using Eq. 5). To calculate  $G_{v2}$  and  $G_{\omega2}$  we use the fact that  $\mathbf{c} = (m_1\mathbf{c}_1 + m_2(\mathbf{c}_2 + \mathbf{l})) / (m_1 + m_2)$  and joint 2 affects only the direction of  $\mathbf{c}_2$ . Thus,

$$G_{v2} = \frac{\partial c_X}{\partial q_2} = \frac{m_2}{m_1 + m_2} \frac{\partial c_{2X}}{\partial q_2} = \frac{-m_2 c_{2Y}}{m_1 + m_2} \quad (50)$$

and

$$\begin{aligned} G_{\omega2} &= \frac{1}{c} \left( \mathbf{b} \cdot \frac{\partial \mathbf{c}}{\partial q_2} \right) = \frac{1}{c^2} \left( c_x \frac{\partial c_y}{\partial q_2} - c_y \frac{\partial c_x}{\partial q_2} \right) \\ &= \frac{m_2}{(m_1 + m_2)c^2} \left( c_x \frac{\partial c_{2y}}{\partial q_2} - c_y \frac{\partial c_{2x}}{\partial q_2} \right) \\ &= \frac{m_2(c_x c_{2x} + c_y c_{2y})}{(m_1 + m_2)c^2} = \frac{m_2 \mathbf{c} \cdot \mathbf{c}_2}{(m_1 + m_2)c^2}. \end{aligned} \quad (51)$$

So

$$G_v = \frac{c_Y H_{12}}{H_{11}} - \frac{m_2 c_{2Y}}{m_1 + m_2} \quad (52)$$

and

$$G_\omega = \frac{m_2 \mathbf{c} \cdot \mathbf{c}_2}{(m_1 + m_2)c^2} - \frac{H_{12}}{H_{11}}. \quad (53)$$

The formulae for  $H_{11}$  and  $H_{12}$  are

$$H_{11} = m_1(r_1^2 + c_1^2) + m_2(r_2^2 + (c_{2x} + l)^2 + c_{2y}^2) \quad (54)$$

and

$$H_{12} = m_2(r_2^2 + c_{2x}(c_{2x} + l) + c_{2y}^2). \quad (55)$$

The easiest calculation procedure is to perform all calculations in link-1 coordinates, and then use the coordinate transform to world coordinates to get  $c_Y$  and  $c_{2Y}$  for  $G_v$ .