CHAPTER

5 Balance Control

5.1 OVERVIEW

Balance control refers to the ability of a human to attain appropriate postures for a given task (e.g. standing, walking, running) within a given environment and to maintain the (dynamic) equilibrium of the body segments during task execution [61]. Numerous studies in the field of biomechanics and physical therapy have contributed to the understanding of the mechanisms used by humans during balance control [105,108,162,86,94,56,116]. Balance control of a humanoid robot bears many similarities to that of a human. Proper balance control is vital for the daily activities of humans. Likewise, balance control plays a central role within the overall control architecture of a humanoid robot. Improper balance control usually results in a fall that hinders the execution of the preplanned tasks. In addition, damage to the robot hardware may be incurred. Similar to balance control in humans [56], the control objectives for humanoid robot balance control can be divided into two large groups:

- balance control during *proactive* (preplanned) tasks;
- *reactive* balance control in response to unexpected external disturbances.

A preplanned task is usually determined by a set of motion/force trajectories, $\{T_{tg}(t)\}$, $t \in \{\overline{t_0, t_f}\}$. The motion trajectory of the robot quite often is expressed in terms of generalized coordinates and their time derivatives. Note that this information encodes the desired motion of the center of mass (CoM), provided one of the feet or both of them are attached to the ground. Furthermore, the force trajectory specifies a set of contact reaction wrenches that constrain the floating base in an appropriate manner, e.g. to support the robot against the gravity force or to ensure the desired way of propulsion. This information, together with that from the motion trajectory, encodes the desired motion of the centers of pressure (CoPs), which also play an important role in balance. Trajectory $T_{tg}(t)$ can be obtained in various ways, either off-line and/or on-line. The balance controller is based on an appropriate dynamic model of the robot, the task, and the environment (see Fig. 5.1). The task-space controller design depends on the low-level controller that is based either on position/velocity or torque control. The task-space controller provides feedback to the on-line trajectory modifier whose role is to ensure the *feasibility* of the input task, $T^{des}(t) = T_{tg}(t) + T_{tm}(t)$, via the $T_{tm}(t)$ component.

Early research on balance control of humanoid robots focused exclusively on preplanned tasks, mainly on cyclic, dynamic gaits on a flat ground [68,145]. As pointed out in [67], besides

Humanoid Robots
https://doi.org/10.1016/B978-0-12-804560-2.00012-2

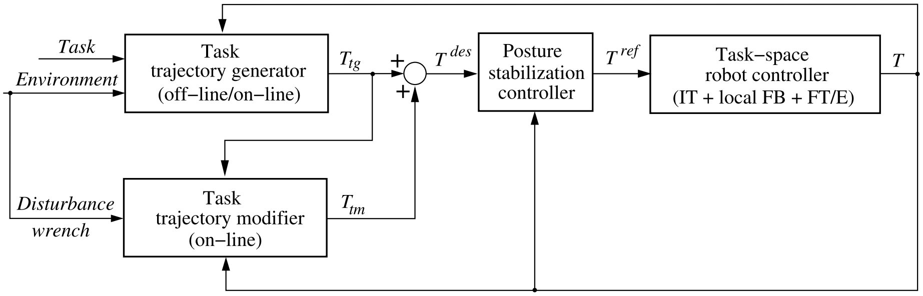


FIGURE 5.1 Generic balance controller structure. The task motion/force trajectory T_{tg} can be generated either off-line and then modified on-line [163,84], or directly on-line. The task trajectory modifier alters the generated trajectory via T_{tm}, s.t. the input to the posture stabilization controller (stabilizer), T^{des}, is *feasible*. The task-space controller: (1) uses inverse kinematics/dynamics transforms (ITs) to generate reference joint angle/torque for position/torque-controlled robots; (2) performs joint-level local feedback (FB) control; (3) uses forward transforms and estimation (FT/E) to generate the current states in terms of motion/force task coordinates (e.g. foot and CoM positions, CoP, reaction wrenches, trunk inclination).

the input trajectory determined by the output of a walking pattern generator, a “stabilizer” needs to be involved. In the ideal situation, the robot would follow a properly generated, dynamically consistent walking pattern. In reality, however, without a “stabilizer,” i.e. appropriate feedback control, it would be impossible to ensure steady gait; even small imperfections in the model and/or the environment would lead sooner than later to a fall. Note also that the performance of the controller depends very much on the input trajectory, i.e. the feedforward component. For example, to improve the dynamic stability of a gait, it is recommended to generate input joint angle trajectories not only for the legs but also for the torso [148,164,123] and arm motion [4]. Another example is walking on a slope. In this case, the trunk should be slightly bent forward s.t. the CoM ground projection remains always within the vicinity of the BoS center. In this way, a sufficiently large stability margin can be ensured [169]. Besides gait, there are other proactive tasks that require appropriate balance control. Examples are pushing [35], lifting [36], kicking [151], hitting [76], and retrieving [84] objects. The respective trajectory generation and balance control methods will be discussed in this chapter and in Chapter 7.

Reactive balance control, on the other hand, is used to accommodate *unexpected* external disturbances in order to avoid the loss of balance. This is done via the on-line trajectory modifier. A typical response to a disturbance is composed of two phases: reaction (reflexive type) and recovery. The reaction/recovery trajectory to be generated depends very much on the direction, magnitude, and point of application of the disturbance wrench. Note that the abovementioned two-phase reaction/recovery response pattern is known from studies in the field of human biomechanics and physical therapy, where reactive balance control methods have been used to assess balance disorders [56]. In the field of humanoid robotics, reactive balance control is likewise of utmost importance, endowing the robot with the ability to withstand the unexpected disturbances inherent to human-centered environments. There are various situations whereby such disturbances may be impressed on the robot. The disturbances may stem from:

- the ground (e.g. unstable [1] or uneven [72,32,70,62]),
- an external push (e.g. due to a physical human-humanoid or humanoid-humanoid interaction),
- the sudden appearance of an obstacle in front of the robot [101,75,152].

From a historical perspective, initially research on balance control on flat ground was solely based on the CoM/zero-moment point (ZMP) dynamical relations [157,156]. Later on, attention was drawn to the role of the angular momentum [129,130]. To ensure a real-time response capability, many studies on balance control resort to simple models, such as those described in Section 4.2. The focus with such models is on the spatial dynamics component of the equation of motion (cf. (4.155)) that includes the CoM states plus the angular momentum and its rate of change. Impedance-based control methods can be designed with such models to ensure a desirable whole-body compliance/admittance type of behavior in response to relatively small external disturbances. With regard to larger disturbances, research in the field of biomechanics on human balance control [162] has revealed that they are better accommodated by involving more complex responses, characterized by specific multijoint movements and muscle activation patterns (synergies). This implies the involvement of the joint-space partial dynamics, in addition to the spatial dynamics ones. Synergy-based balance control methods will be discussed in Chapter 7.

This chapter is organized in 14 sections. The following section defines the dynamic postural stability. In Section 5.3, stability analysis based on a simple inverted-pendulum (IP)-on-foot model is introduced. In Section 5.4, the ZMP manipulation-type stabilization on flat ground is discussed. Capture point (CP)-based analysis and stabilization are addressed in Section 5.5. Stabilization based on the maximum output admissible set method is explained in Section 5.7. The important role of the angular momentum in stability analysis and control is highlighted in Section 5.6. Methods of balance control based on the spatial momentum and its rate of change are introduced in Section 5.8. In Section 5.9, the role of the task-space controller in balance control is clarified. The following four sections are devoted to noniterative optimization methods for balance control. The optimization approaches to the body wrench distribution (WD) problem are discussed in Section 5.10. Spatial dynamics–based motion optimization is addressed in Section 5.11. Noniterative whole-body motion/force optimization methods are introduced in Section 5.12. In Section 5.13, balance control approaches that ensure the compliant whole-body behaviors in response to weak external disturbances are highlighted. In the final section, a number of methods for iterative optimization in balance control are introduced.

5.2 DYNAMIC POSTURAL STABILITY

Methods of balance stability analysis make predominant use of the simple IP models discussed in Section 4.3.2. Such methods were developed in the field of biomechanics to assess postural stability in humans, using the (linearized) constant-length IP-cart model. Thereby, the important role of the CoM/CoP dynamic relation has been revealed. In the field of robotics, balance stability analysis was based on the ZMP concept, originally introduced to assess dynamic postural stability during biped gait [154,155]. Recall that on flat ground, the

ZMP coincides with the CoP. The preferable model in the field of robotics is the LIP-cart model. As will be shown below, the linearized IP and LIP models yield identical results from the viewpoint of stability. One should bear in mind, though, that the LIP model restricts the set of postures by excluding a subset comprising the straightened leg (a kinematically singular posture) and postures in its vicinity. In this sense, the LIP-cart model–based analysis lacks completeness.

The LIP-cart balance stability model can be straightforwardly implemented with a real robot. In 3D, the spherical IP models described in Section 4.4 are relevant. In the general case, the equation of motion of a spherical IP is coupled in the sagittal and frontal planes. When linearized around the vertical, though, decoupling can be ensured. Thus, balance stability analysis and balance control design can be simplified by making use of two identical IP models on each of these planes. In the following discussions, both scalar and vectorial notations pertinent to IP models on the plane and spherical IP models in 3D, respectively, will be used.

The ZMP concept was helpful in establishing a number of balance control methods, leading to reliable results obtained with real robots. Nevertheless, gradually it became apparent that ZMP-based stability assessment is insufficient for the complete characterization of postural stability [127,50,7,143]. The inherent limitations of ZMP methods can be summarized as follows. They are:

1. valid only with planar contact models on flat ground;
2. cannot deal with:
   a. multicontact (feet and hand contact) postures,
   b. the presence of friction,
   c. foot rotation around an edge of the BoS polygon;
3. do not incorporate important information about:
   a. the full state (i.e. the CoM velocity),
   b. the stability margin that results from a limited BoS area.

Efforts have been made to alleviate the abovementioned ZMP-related problems. A number of researchers tackled for example the flat-ground and multicontact limitations of the ZMP concept by involving projections of the ZMP onto a virtual plane [147,145,66,131,136,132,9]. The problem of foot rotation around an edge of the BoS polygon stemming from an unbalanced moment was assessed via the so-called *foot rotation indicator* (FRI) [29] (called "fictitious ZMP" in [155]). These efforts have contributed to new algorithms and improvements in ZMP-based balance control algorithms. On the other hand, regarding implementation with a real robot, the most successful example are Honda's P2 [48] and ASIMO [150] robots. The so-called "model-ZMP" method was introduced to deal with an undesirable upper-body motion that appears during a cyclic gait because of model inaccuracies in both the robot and the environment. The balance controller makes use of a number of balance stability mechanisms, such as (cf. [67], p. 150):

- ankle torque control,
- foot placement modification (e.g. in a cyclic gait),
- desired (LIP-cart model) and actual CoM/ZMP error dynamics,
- the so-called "divergent component of motion" [149] (see Section 5.6.2),
- angular momentum control via the RWP model,

- impact absorption in the foot landing phase (passive and active type),
- full-body control.

Recently, a growing interest can be observed toward environments with relatively large terrain irregularities, e.g. as in a disaster zone. Generally, the same balance stability mechanisms as above can be applied. There are some differences though. Note that the vertical motion of the CoM cannot be ignored anymore, as with the LIP-cart model. Also, the role of angular momentum control increases. Since the rate of change of angular momentum can only be controlled via the contact wrenches, their appropriate distribution at the contacts has to be tackled as well.

In the remainder of this chapter, stability analysis methods for balance control used in the fields of biomechanics and humanoid robotics will be presented. It should be noted that rigorous balance stability analysis is available only for some of the balance mechanism components mentioned above, but not for the system as a whole. The reason is the complexity of the model (human or robot), characterized as underactuated, on a floating base, comprising a multi-DoF varying structure, and subjected to various types of external disturbances, including impacts. It should also be noted that the term "stability," as used in the two fields, lacks formal rigor due to the same reason. Attempts have been made to devise a more rigorous treatment e.g. based on the *viability theory* [5]. This theory was successfully applied to mobile robots on wheels. Note, however, that the success can be attributed to the simple point mass model. An adaptation to humanoid robots has been proposed via the so-called *viability kernel* [159]. The method is general but lacks constructiveness from the viewpoint of controller design. A combination with model-predictive control has been considered to alleviate the problem [160]. Nevertheless, the rigorous treatment of stability for complex models still remains an open issue. With simplified models, such as the IP-on-foot–type ones, rigorous results can be obtained, as will be shown in the following discussion.

5.3 INVERTED PENDULUM-ON-FOOT STABILITY ANALYSIS

The simple, nonlinear or linearized IP-on-foot models introduced in Section 4.3 (cf. Fig. 4.2A) have been employed in studies on balance control in the fields of biomechanics and physical therapy [116,64,52,51,54].

5.3.1 The Extrapolated CoM and the Dynamic Stability Margin

A dynamic stability region can be identified within the gCoM position/velocity phase plane with the help of an optimization procedure that accounts for gravity, friction, CoP,[1] and ankle torque constraints [116]. The static (BoS) and dynamic stability regions are shown in Fig. 5.2A. Apparently, the dynamic stability region is much larger than the static one. Consider, for example, a representative state outside the static stability region, State A. At this state, the body is strongly leaning forward. Nevertheless, forward fall can be prevented with

[1] In the biomechanics literature, the term CoP is exclusively used. As discussed in Section 4.3.2 this term has a broader meaning than the ZMP.

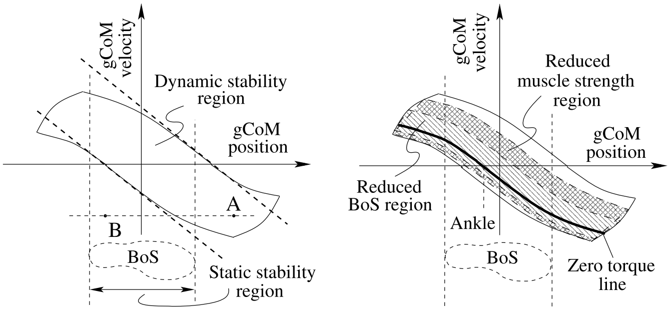


FIGURE 5.2 Left (A): Phase plane of the gCoM with static and dynamic stability regions [116]. The dynamic stability region can be approximated with a linear relation (the inclined dotted lines) to facilitate real-time balance controller design [52]. A and B are two representative states. Right (B): The dynamic stability region area shrinks for weaker (elderly) persons. This also happens with a smaller BoS area.

a backward (negative) CoM velocity. The same velocity is applied at State B, located within the static stability region. Note, however, that the system is dynamically unstable in this state: a backward fall will occur because the body is not sufficiently leaning forward. Furthermore, as apparent from Fig. 5.2B, the dynamic stability region shrinks significantly with a smaller or a slippery BoS. The region also shrinks considerably for (elderly) persons with weaker muscles. This finding can be directly transferred to the field of humanoid robots, thus emphasizing the important role of the ankle torque constraint in balance control.

The optimization procedure used to identify the dynamic stability region may not be suitable for on-line implementation due to computational cost limitations. The problem can be alleviated by approximating the region with a linear relation derived from a closed-form solution (4.12) [52]. This solution expresses the relative motion of the gCoM, $x_g(t)$, w.r.t. to a given CoP, say $\bar{x}_p$. As apparent from (4.12), the balance control mechanism depends on the initial state, $(x_{g0},\ v_{g0})$. On the other hand, note that the CoP is constrained by the size of the BoS, i.e. $-l_f \le \bar{x}_p \le l_f$, l_f denoting the half foot length. As an example, consider a state with positive initial gCoM velocity ($v_{g0} > 0$). It may then happen that the backward (negative) acceleration, $-m\ddot{x}_g$, is insufficient to prevent the gCoM x_g to pass CoP $\bar{x}_p$. When the gCoM passes this value, the CoM will accelerate further forward. As a consequence, CoP $x_p(t)$ will arrive at the BoS boundary l_f and the foot will begin to rotate in the clockwise direction. This is a potentially dangerous state that may result in posture instability and a fall. In order to prevent this from happening, the CoM should never pass the given CoP, i.e. $x_g(t) < \bar{x}_p,\ \forall t$. This condition can be rewritten with the help of (4.12) as

$$x_{g0} + \frac{v_{g0}}{\omega} \le \bar{x}_p, \tag{5.1}$$

where $\omega = \omega_{IP} = \sqrt{g/l}$. In the above derivation $|\tanh(\omega t)|_{t\to\infty} \to 1$ was used. One can conclude then that to avoid destabilization, the CoP $x_p(t)$ should move sufficiently fast toward the positive BoS boundary, l_f. In other words, the gCoM should always follow the CoP motion and come to a stop exactly when it reaches the CoP. Thus, when the CoP eventually arrives at the boundary, followed by the gCoM, the movement will end there with a stationary state. The same line of thought also holds for motion in the opposite direction, i.e. with a negative initial gCoM speed. By replacing $\bar{x}_p$ with the BoS boundaries, (5.1) can be rewritten in the general form, i.e.

$$-l_f \leq x_{ex}(t) \leq l_f, \tag{5.2}$$

where

$$x_{ex}(t) \equiv x_g(t) + \frac{\dot{x}_g(t)}{\omega}. \tag{5.3}$$

Here x_{ex} is named the "extrapolated CoM" (henceforth abbreviated as xCoM) [52]. The xCoM is useful in approximating the dynamic stability region in Fig. 5.2A. The approximation is indicated by the inclined dashed lines. Furthermore, the xCoM can be used to define the *dynamic stability margin*, i.e.

$$s(t) = \left| \pm l_f - x_{ex}(t) \right|. \tag{5.4}$$

The dynamic stability margin plays an important role in choosing an appropriate controller action, as will be shown in what follows.

5.3.2 Extrapolated CoM Dynamics

By investigating the dynamics of the xCoM, important conclusions about balance control stability can be obtained [51]. To this end, first recall the CoM/CoP dynamic relation for the IP-on-foot model (cf. (4.10)). We have

$$\ddot{x}_g = \omega^2(x_g - x_p). \tag{5.5}$$

Next, solve (5.3) for the gCoM velocity, i.e.

$$\dot{x}_g = -\omega(x_g - x_{ex}). \tag{5.6}$$

On the other hand, the time differential of (5.3) is

$$\dot{x}_{ex} = \dot{x}_g + \frac{\ddot{x}_g}{\omega}. \tag{5.7}$$

Insert (5.5) and (5.6) into the above equation to obtain the xCoM dynamics as

$$\dot{x}_{ex} = \omega(x_{ex} - x_p). \tag{5.8}$$

The gCoM and xCoM dynamics ((5.6) and (5.8), respectively) comprise two autonomous dynamic system components: a stable one, $\dot{x}_g = -\omega x_g$, and an unstable one, $\dot{x}_{ex} = \omega x_{ex}$. It can then be concluded that to obtain a stable CoM motion, it would suffice to stabilize the motion of the xCoM [51]. In other words, with a stable xCoM trajectory there is no need to care about the CoM trajectory at all; the CoM will follow the xCoM motion faithfully. This conclusion plays an important role in balance controller design, as will be shown in what follows.

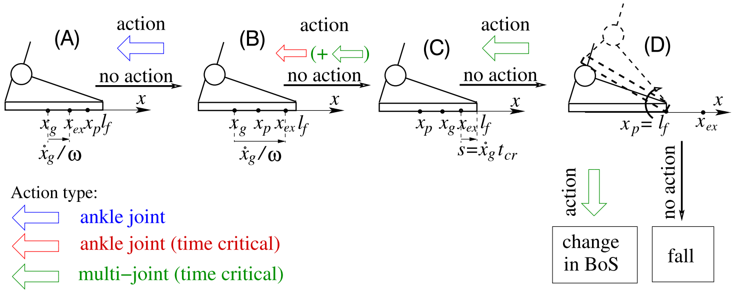


FIGURE 5.3 The balance stability mechanism depends on transitions between states of gCoM x_g, CoP x_p, and the extrapolated CoM (xCoM), x_{ex}, as well as on the BoS (foot length l_f) [52]. The figure shows some discrete states with $\dot{x}_g > 0$. In order to maintain balance, a controller action is needed. Without such action, the posture may become unstable as apparent from the rolling foot State (D). The ankle-joint action at State (B) is not time-critical. At State (C), a time-critical action is needed (depending on the dynamic stability margin $s = v_g t_{cr}$). The action may involve the ankle joint only, or alternatively, a multijoint motion could be invoked that yields a change in the angular momentum. At State (D), there are two possible time-critical actions invoked through a multijoint motion: either reverse foot rotation via a change in the partial angular momentum, or change the BoS (e.g. via a reactive step).

The xCoM dynamics provide an additional insight into the stability of the system. Using (5.6) and (5.8), the IP-on-foot dynamics can be represented in the following state-space form [23]:

$$\frac{d}{dt}\begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix}\begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} x_p. \tag{5.9}$$

This representation exhibits the stable (upper row) and unstable (lower row) autonomous dynamical system components mentioned above. As already noted, it would be sufficient to design a balance controller for the unstable part only. Such a design is presented in Section 5.6.3.

5.3.3 Discrete States With Transitions

The above analysis has clarified that three characteristic points, gCoM, xCoM, and CoP, denoted as x_g, x_{ex}, and x_p, respectively, play an important role in dynamic stability along with the BoS. In what follows, it will be shown that the relative positions of these points within the BoS can provide deeper insight into the balance control mechanism. Consider the following four states [52] (cf. Fig. 5.3):

(A) $x_g < x_{ex} < x_p < l_f$,
(B) $x_g < x_p < x_{ex} < l_f$,
(C) $x_p < x_g < x_{ex} < l_f$,
(D) $x_p = l_f < x_{ex}$.

Note that x_g, x_{ex}, and x_p provide information for the CoM position, velocity, and acceleration, respectively. Assume a positive initial gCoM speed and consider the possible transitions between the states for a given xCoM. First, note that with State (a), the gCoM will never reach the CoP. At some time instant, though, the gCoM motion will reverse direction. Then, State (b) will be invoked. With this state, the gCoM will pass the CoP and arrive at State (c), accelerating thereby further in the positive direction, as explained above. A time-critical action is then needed. The xCoM will reach the BoS boundary at the critical time instant t_{cr} that can be calculated from the closed-form solution (4.12). By assuming a constant $\bar{x}_p$, t_{cr} can be approximated as

$$t_{cr} \approx \frac{l_f - x_{ex}(t)}{\dot{x}_g(t)} = \frac{s(t)}{\dot{x}_g(t)}. \tag{5.10}$$

The time-critical action depends on the posture and the initial conditions. The ankle joint torque could be sufficient to ensure the transition to State (b). When the required fast response cannot be achieved with an ankle torque alone, a multijoint motion should be invoked to generate a change in the angular momentum.

State (d) is characterized by a change in the contact condition, from a line contact to a point contact in the 2D example, or from a plane contact to a line contact in 3D. This change stems form an unbalanced moment at the foot that may lead to a foot rolling around the BoS boundary. The magnitude of this moment can be assessed via the so-called *foot rotation indicator* (FRI) [29] (called "fictitious ZMP" in [155]). Note that the unbalanced moment implies a change in the *partial angular momentum*. The CoM equations (4.38) reveal that the state of the robot is determined by the CRB partial dynamics, including the rate of change of the *CRB angular momentum*. When the joints are locked, the unbalanced moment at the foot applies to the whole CRB, yielding eventually postural instability and a fall. When the joints are unlocked, the change in the partial angular momentum at the foot could be compensated via a change in the partial angular momentum of the rest of the links, e.g. trunk, arms and head. Thus, a possibility to avoid postural destabilization despite foot rotation exists. In [30], the so-called *zero rate of change of angular momentum* (ZRAM) point[2] was introduced to assess postural instability in relation to the angular momentum. Further details with regard to the role of the angular momentum in balance control will be provided in Section 5.6. When the robot is incapable of generating the required change of the angular momentum, another multijoint action can be invoked. This type of action is adopted from the so-called "change-in-support" strategy used by humans to maintain their upright stance [94,93]. The strategy refers to establishing a hand contact with the environment and/or initializing reactive stepping (step strategy). Further details will be given in Section 7.7.5.

5.3.4 Dynamic Stability Region in 2D

The BoS of a humanoid robot on flat ground is quite often represented by a convex 2D polygon. The respective CoP, gCoM, and xCoM then become 2D vector quantities. The CoM/CoP dynamics (5.5) are written in vector form as

$$\ddot{\boldsymbol{r}}_g = \omega^2(\boldsymbol{r}_g - \boldsymbol{r}_p). \tag{5.11}$$

[2] Referred to as "centroidal moment pivot (CMP)" in [121] (defined in (4.24)).

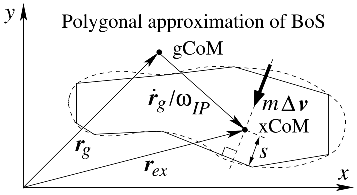


FIGURE 5.4 Dynamic stability margin s is defined in 2D as the shortest distance from the xCoM vector, $\boldsymbol{r}_{ex}$, to the BoS polygon [52]. The dynamic stability margin can also be interpreted as the minimum force impulse $M\Delta\boldsymbol{v}$ that, when acting along the line determined by the shortest distance, would deteriorate the balance.

Here $\boldsymbol{r}_g(t)$ and $\boldsymbol{r}_p(t)$ stand for the gCoM and CoP position vectors, respectively. Also, referring to (5.3), the xCoM can be written as

$$\boldsymbol{r}_{ex}(t) = \boldsymbol{r}_g(t) + \frac{\dot{\boldsymbol{r}}_g(t)}{\omega}. \tag{5.12}$$

In 2D, the dynamic stability margin (5.4) is interpreted as the shortest distance between the xCoM and the BoS polygon (cf. Fig. 5.4). It also can be interpreted as the minimum change of momentum, $M\Delta\boldsymbol{v}$, that would compromise the balance when applied in the direction determined by the shortest distance between the xCoM and the BoS polygon [52]. It then becomes apparent that the dynamic stability margin can vary with the direction of the applied disturbance.

To formally constrain the xCoM within the BoS polygon, use notation (4.32) introduced for the CoP constraint. Then we have

$$\boldsymbol{B}_s\boldsymbol{r}_{ex} \preceq \boldsymbol{c}. \tag{5.13}$$

It should be noted that a 2D stability margin definition appeared also in the robotics literature as the shortest distance between the CoP and the BoS polygon [59]. This definition, however, does not include full information about the current state since the gCoM velocity does not appear in the CoP definition.

5.4 ZMP MANIPULATION–TYPE STABILIZATION ON FLAT GROUND

Early research on biped gait control was mainly based on the LIP-cart model on flat ground. In Section 4.3.2 it was clarified that on flat ground, the (net) CoP coincides with the ZMP (see also [121]). The dynamic relations pertinent to the LIP-cart model were also introduced there. Note that the equation of motion, (4.5), is of the same form as that of the linearized IP-on-foot model used in Section 5.3. There is an explicit solution, (4.6), which facilitates balance stability analysis and control. Referring to the LIP-cart dynamics (4.18), the following simple ankle torque controller ($u = m_y$) can be employed [99,146,27,73]:

$$u = -k_d\dot{x}_g - k_p x_g. \tag{5.14}$$

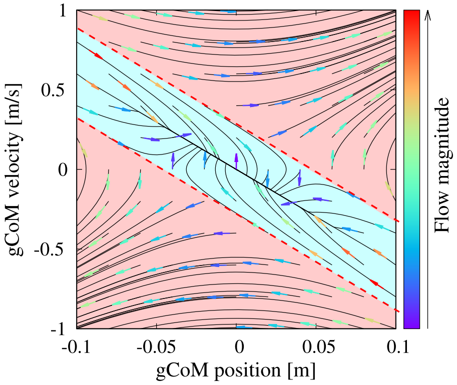


FIGURE 5.5 gCoM phase portrait for the IP/LIP-on-cart models. The approximation of the dynamic stability region in accordance with (5.2) is shown in light blue.

Although feedback gains k_p and k_d can be readily selected to obtain a desirable closed-loop response, the implementation of this controller is prone to a number of difficulties [67]. To tackle this problem, consider the following state-space representation of the gCoM dynamics (5.5):

$$\frac{d}{dt}\begin{bmatrix} x_g \\ \dot{x}_g \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix}\begin{bmatrix} x_g \\ \dot{x}_g \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega^2 \end{bmatrix} x_p, \tag{5.15}$$

where $\omega = \overline{\omega} = \sqrt{g/\bar{z}_g}$ and $\bar{z}_g$ denotes the constant height of the LIP-cart model. Apparently, the CoM state is used as the state variable and the CoP as the input ($u = x_p$). The solution to the above equation can be obtained in explicit form as [23]

$$\begin{bmatrix} x_g \\ \dot{x}_g \end{bmatrix} = \begin{bmatrix} \cosh(\omega t) & \frac{1}{\omega}\sinh(\omega t) \\ \omega \sinh(\omega t) & \cosh(\omega t) \end{bmatrix}\begin{bmatrix} x_{g0} \\ v_{g0} \end{bmatrix} + \begin{bmatrix} 1 - \cosh(\omega t) \\ -\omega \sinh(\omega t) \end{bmatrix} x_p. \tag{5.16}$$

The state-space representation (5.15) can be used for quantitative analysis of the IP/LIP-on-foot models, via their phase portraits. The unforced system generates the vector field and the respective flow. To obtain the dynamic stability region approximation, (5.2), make use of the forced system with input x_p in (5.15) set at the two BoS boundaries. As an example, consider the phase portrait in Fig. 5.5, generated with the LIP-cart model.[3] We have

$$\begin{aligned} \bar{z}_g &= 0.279\,\text{m} \rightarrow \omega = 5.94\ 1/\text{s}, \\ l_f &= 0.096\,\text{m}. \end{aligned} \tag{5.17}$$

The dynamic stability region in the figure is shown in light blue. As already clarified, the boundaries of this region play an important role in making decisions about proper controller actions for balance control.

[3] The data are derived from a small-size HOAP-2 robot [26] (see Section A.1).

The main result from the above brief stability analysis is that, to alleviate the problem of controller (5.14), the balance stabilizer should use the CoP/ZMP as a control input in lieu of the ankle torque. In what follows, a few such stabilizers will be introduced. The stabilizers can be formulated either in a scalar or vector form, $u = x_p$ or $\boldsymbol{u} = \boldsymbol{r}_p$, respectively. Note that the latter is needed when implementing a stabilizer with a real robot. In this case, the two vector components are obtained from the 3D LIP model. Since this model comprises two equations of motion that can be decoupled in the sagittal and frontal planes, it is possible to design two independent and identical stabilizers for each coordinate. In the following discussion, both the scalar and the vector notations will be used.

5.4.1 The ZMP Manipulation–Type Stabilizer

A stabilizer that uses the ZMP as a control input was first introduced in [97] under the name "ZMP manipulation" controller. Referring to the IP equation of motion in Cartesian coordinates, (4.4), the following feedback control laws are designed:

$$f_z = |f| \cos\theta = z_g^{des} - z_g - \dot{z}_g + Mg, \tag{5.18}$$

$$\begin{aligned} m\ddot{x}_g &= \frac{\sin\theta}{\cos\theta}(z_g^{des} - z_g - \dot{z}_g + Mg) \\ &= \frac{x_g - x_p}{z_g}(z_g^{des} - z_g - \dot{z}_g + Mg). \end{aligned} \tag{5.19}$$

The quantities appearing in these equations have been defined in Section 4.3.2; see also Fig. 4.2. First, it is easy to confirm that the closed-loop dynamics in the vertical direction yield exponential convergence of state z_g to z_g^{des}. Next, consider the following equation for the horizontal coordinate:

$$\beta\ddot{x}_g = x_g - x_p. \tag{5.20}$$

Observe that (5.20) can be rendered in the form (4.10) with $\beta^{des}(t) = 1/\omega^2(t)$, $\omega(t)$ being defined in (4.14). Having in mind (5.19), set

$$\beta(t) = \frac{Mz_g}{z_g^{des} - z_g - \dot{z}_g + Mg}. \tag{5.21}$$

As a result of the exponential convergence in the vertical direction, $\beta(t)$ in (5.21) will converge to $\beta^{des}(t)$, also exponentially.

Furthermore, by setting the ZMP as the control input in (5.20), $u = x_p$, two feedback control laws can be designed, as follows:

$$u = x_g + \dot{x}_g - v_g^{des} \tag{5.22}$$

and

$$u = 2x_g + \dot{x}_g - x_g^{des}. \tag{5.23}$$

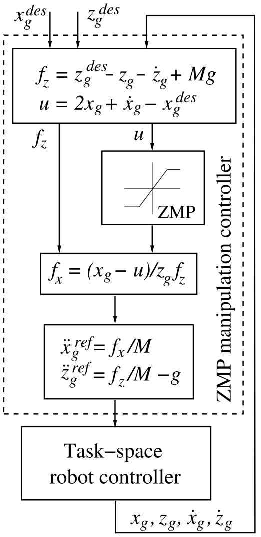

x_g^{des}
z_g^{des}
$f_z = z_g^{des} - z_g - \dot{z}_g + Mg$
$u = 2x_g + \dot{x}_g - x_g^{des}$
f_z
u
ZMP
$f_x = (x_g - u)/z_g f_z$
$\ddot{x}_g^{ref} = f_x/M$
$\ddot{z}_g^{ref} = f_z/M - g$
ZMP manipulation controller
Task–space robot controller
$x_g, z_g, \dot{x}_g, \dot{z}_g$

FIGURE 5.6 Block diagram for ZMP manipulation–type balance regulator based on the constant-length IP-on-cart model in Fig. 4.2B [97]. With reference to Fig. 5.1, it is apparent that: (1) the task trajectory T is defined in terms of the CoM coordinates and (2) the ZMP manipulation controller plays the role of the stabilizer.

Control law (5.22) is useful to track the reference gCoM speed v_g^{des}, e.g. as required in the case of gait control. Control law (5.23), on the other hand, is useful as a regulator w.r.t. the desired CoM position x_g^{des}, e.g. as required in upright balance control. The controller block diagram for the latter case is shown in Fig. 5.6. Note that the control input u passes through a limiter that saturates at the BoS boundaries. The task-space controller was implemented as a dynamic torque controller in [97].

5.4.2 Velocity-Based ZMP Manipulation–Type Stabilization in 3D

The ZMP manipulation–type stabilizer can be formulated in terms of velocity relations [145]. The design is based on the nonlinear spherical IP model introduced in Section 4.4.2. Thus, the ZMP and CoM coordinates are 2D and 3D quantities, respectively. Recall that the 3D CoM coordinate vector is denoted as $\boldsymbol{r}_C = \begin{bmatrix} \boldsymbol{r}_g^T & z_g \end{bmatrix}^T$. The control input is the ZMP, as in the original ZMP manipulation stabilizer discussed above: $\boldsymbol{u} = \boldsymbol{r}_p^{ref} \in \Re^2$. The block scheme of the controller is shown in Fig. 5.7. First, the *ZMP planner* calculates the control input from the desired CoM velocity vector, $\dot{\boldsymbol{r}}_C^{des} \in \Re^3$. Recall that the spherical IP dynamics can be considered decoupled in x and y, under the assumption of small deviation from the vertical. Hence, it is possible to design the spherical IP controller as two independent linearized IP-on-cart (cf. Fig. 4.2B) controllers, using either the conventional (e.g. as in [20]) or any other approach (e.g. as in [166]). Note that control input $\boldsymbol{u}$ has to be limited to comply with the ZMP constraint. Furthermore, the *ZMP manipulator* determines the reference CoM velocity vector $\dot{\boldsymbol{r}}_C^{ref}$

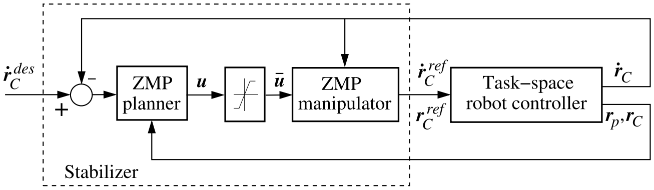


FIGURE 5.7 Block diagram of a velocity-based ZMP manipulation–type balance controller [145]. With reference to Fig. 5.1, it is apparent that: (1) the task trajectory T is defined in terms of the CoM velocity and (2) the stabilizer comprises two parts. The ZMP planner is in fact an IP-on-cart controller. The ZMP manipulator determines the (stabilized) reference CoM trajectory. The task-space controller makes use of the CoM inverse kinematics in combination with other velocity-level constraints and includes a local joint-angle feedback controller.

by integrating the respective accelerations. Then we obtain

$$\begin{aligned} \ddot{z}_g^{ref} &= k_z(\dot{z}_g^{ref} - \dot{z}_g), \\ \beta^{ref}\ddot{\boldsymbol r}_g^{ref} &= (\boldsymbol r_g^{ref} - \boldsymbol u), \end{aligned} \tag{5.24}$$

where k_z is a feedback gain and $\beta^{ref} = (z_g - z_p^{ref})/(\ddot{z}_g^{ref} + g)$ (cf. (4.14)). Note that the vertical position of the ZMP is not constant and that $\dot{\boldsymbol r}_C^{ref}$ is fed into the task-space robot controller. This controller is designed for a position-controlled robot and comprises an inverse kinematics solver that transforms the input into reference joint velocities. The solver also takes care of the joint-space and other constraints expressed in terms of velocities. The reference joint velocities drive a joint-space local feedback controller. Finally, current joint and ZMP data are obtained from the robot sensors and transformed into the current CoM velocity and ZMP vectors, $\dot{\boldsymbol r}_C$ and $\boldsymbol r_p$, respectively.

The velocity-based ZMP manipulation–type balance controller is efficient from a computational viewpoint: real-time implementation with position-controlled humanoid robots is possible. Also, the CoM and CoP are allowed to deviate in the vertical direction. Thus, the controller can be used for proactive tasks such as walking/stepping on an irregular terrain.

One notable drawback of the above controller is the lack of stability analysis. This problem can be alleviated by introducing the constant-height CoM (LIP) constraint, as shown in [10] (see also [12,11]). The LIP constraint is enforced in 3D via the sphere-on-plane model introduced in Section 4.4.2. With this constraint, there is no need to account for the vertical CoM coordinate. Thus, the controller can be designed based on the gCoM only. The block scheme is shown in Fig. 5.8. The desired gCoM position, $\boldsymbol r_g^{des}$, and its time derivatives are determined by a task motion planner. The ZMP planner calculates just the desired ZMP, $\boldsymbol r_p^{des}$, from the CoM/ZMP dynamics (5.11). Thereby, note that $\omega = \overline{\omega}$, according to the LIP constraint. The same equation is also used to obtain the actual ZMP, $\boldsymbol r_p$, whereby the actual gCoM velocity is given as

$$\dot{\boldsymbol r}_g = \boldsymbol u + \boldsymbol\epsilon. \tag{5.25}$$

Here $\boldsymbol\epsilon$ stands for the control error. The control input, on the other hand, is defined as

$$\boldsymbol u = \dot{\boldsymbol r}_g^{des} + \boldsymbol K_g \boldsymbol e_g - \boldsymbol K_p \boldsymbol e_p, \tag{5.26}$$

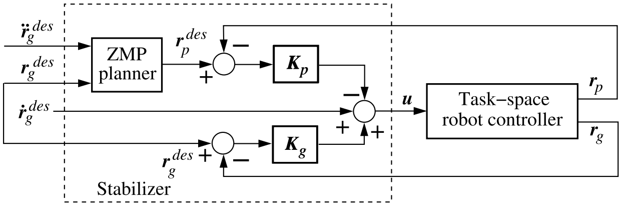


FIGURE 5.8 Block diagram of another velocity-based ZMP manipulation–type balance controller [10]. With reference to Fig. 5.1, it is apparent that: (1) the task trajectory T is defined in terms of the CoM motion and (2) the stabilizer comprises a ZMP planner and two loops for the CoM and ZMP motion proportional feedback control. The task-space controller makes use of a CoM inverse kinematics solution with a whole-body motion capability and includes a local joint-angle feedback controller.

where $\boldsymbol{e}_g \equiv \boldsymbol{r}_g^{des} - \boldsymbol{r}_g$ and $\boldsymbol{e}_p \equiv \boldsymbol{r}_p^{des} - \boldsymbol{r}_p$ are the gCoM and CoP errors, respectively; $\boldsymbol{K}_g = \text{diag}(k_g^x,\ k_g^y)$, $\boldsymbol{K}_p = \text{diag}(k_p^x,\ k_p^y)$ are control gain matrices. When the control gains are selected to satisfy the following conditions:

$$k_g^i > \varpi > 1, \quad 0 < k_p^i < \varpi - (\alpha^2/\varpi) - \gamma^2, \quad i \in \{x, y\},$$

for any positive constants $\alpha < \varpi$ and $\gamma < \sqrt{\varpi - (\alpha^2/\varpi)}$, then the controller is said to be input-to-state–stable, $(\boldsymbol{\epsilon},\ \dot{\boldsymbol{\epsilon}})$ and $(\boldsymbol{e}_p,\ \dot{\boldsymbol{e}}_p)$ standing for the input and state, respectively. Readers interested in the proof are referred to [11].

The task-space robot controller in Fig. 5.8 comprises similar components as the one in Fig. 5.7. A notable difference can be found in the resolution method of the CoM inverse kinematics. The 3D CoM velocity vector is expressed as the sum of two components. One of the components results from the motion of the support leg(s). The other component is determined via the inverse kinematics solution w.r.t. the desired spatial velocities of the other end links. In this way, the motion in the limbs will not disturb the balance determined from the simple CoM/ZMP dynamics model. This method has been named "Kinematic resolution of CoM Jacobian with embedded motion" [10]. Similar to the previous velocity-based ZMP manipulation controller, there is an advantage of computational efficiency and possibility for real-time implementation with position-controlled humanoid robots. Note, though, that in this controller no provisions have been made to constrain the desired ZMP (the output of the ZMP planner) to stay within the BoS. Thus, there is no guarantee that the foot will always be in full contact with the flat ground.

5.4.3 Regulator-Type ZMP Stabilizer

Making use of the state-space representation (5.15), a regulator-type ZMP stabilizer can be designed [143]. Again, the ZMP is used as the control input. First, the gCoM, ZMP, and BoS limit coordinates are redefined w.r.t. the desired gCoM position, x_g^{des}, as $x_g^* = x_g - x_g^{des}$, $x_p^* = x_p - x_g^{des}$, and $l_f^* = l_f - x_g^{des}$, respectively. Then, the reference input ZMP, $u = x_p^{ref}$, is determined as

$$u = -k_1 \dot{x}_g^* - k_2 x_g^*, \tag{5.27}$$

$k_1,\ k_2$ denoting feedback gains. The state-space representation of the stabilizer can then be written as

$$\frac{d}{dt}\begin{bmatrix} x_g^* \\ \dot{x}_g^* \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2(k_1+1) & \omega^2 k_2 \end{bmatrix}\begin{bmatrix} x_g^* \\ \dot{x}_g^* \end{bmatrix}. \tag{5.28}$$

Since there is no guarantee that u will honor the BoS constraints, $-l_f^* \leq u \leq l_f^*$, a limiter has to be involved, as was the case with the ZMP manipulation–type stabilizers discussed above. Under saturation at the boundaries, the state evolves according to

$$\frac{d}{dt}\begin{bmatrix} x_g^* \\ \dot{x}_g^* \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \omega^2 & 0 \end{bmatrix}\begin{bmatrix} x_g^* \\ \dot{x}_g^* \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega^2 \end{bmatrix}(\pm l_f^*). \tag{5.29}$$

This equation is in the same form as (5.15). It is important to note at this point that the BoS boundaries constitute a set of equilibrium points. We have

$$\{(x_g^*,\ \dot{x}_g^*) : (\pm l_f^*,\ \dot{x}_g^*),\ \forall \dot{x}_g^*\}.$$

When the state arrives at the boundary, it will be trapped there and, hence, will not converge to the desired (isolated) equilibrium state $(x_g^{des}, 0)$. This means that the boundaries should be excluded as admissible states. The dynamic stability region can then be determined in the same form as (5.2), so we have

$$-l_f^* < x_{ex}^* < l_f^*. \tag{5.30}$$

Here $x_{ex}^* \equiv x_g^* + \dot{x}_g^*/\omega$ is recognized as the xCoM represented in modified coordinates. The strict inequalities imply that the undesirable equilibria at the boundaries are to be avoided.

Furthermore, assume that the desired poles are given as ωq_1 and ωq_2. The feedback gains can then be determined as $k_1 = -(q_1 q_2 + 1)$ and $k_2 = (q_1 + q_2)/\omega$. Poles with negative real values will steer the state toward the isolated equilibrium $(x_g^*,\ \dot{x}_g^*) = (0,\ 0)$. Consequently, the CoM will approach the desired state, $(x_g^{des},\ 0)$.

Furthermore, a "constrained-control" region can be defined by restricting the action of control law (5.27) within the BoS boundaries. Then we have

$$-k_1 \dot{x}_g^* - k_2 x_g^* = \pm l_f^*. \tag{5.31}$$

Apparently, the constrained-control region is parametrized by the feedback gains. Whenever control u satisfies the constraint (5.31) for some given feedback gains and the respective xCoM (i.e. the CoM state) lies within the dynamic stability region (5.30), the posture of the robot will be stable.

It is interesting to observe that the pole assignment also determines whether the dynamic stability region and the constrained-control one intersect. Intersection implies a smaller operational domain for the regulator. To obtain a regulator with the largest possible operational domain requires the two regions to overlap. From the expressions for the four boundary conditions, (5.30) and (5.31), it is straightforward to determine that $k_2 = k_1/\omega$ should then hold. This condition implies that the poles are correlated via $(q_1+1)(q_2+1) = 0$. It will then suffice to assign the poles as $(-1,\ q),\ q < 0$. The value of q can be chosen to satisfy any other per-

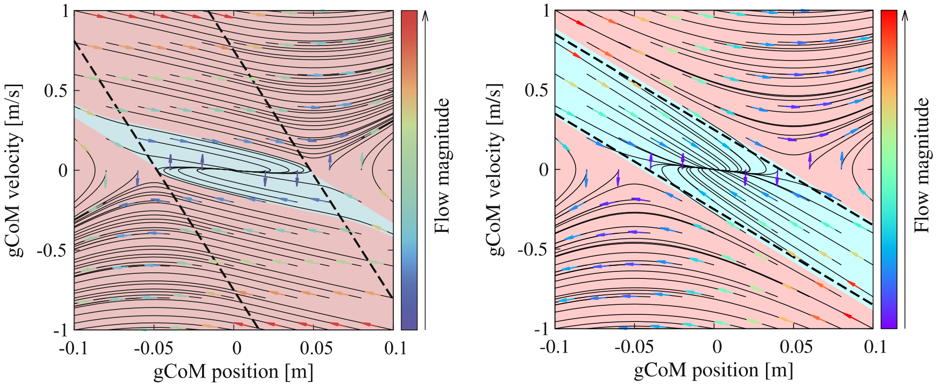


FIGURE 5.9 gCoM phase portraits with constrained-control and dynamic stability regions. The former are enclosed between the dashed lines. The dynamic stability regions are shown in light blue. Left (A): Intersecting (nonoptimal) case. Right (B): Overlapping case (named the "best CoM-ZMP regulator" in [143]).

formance constraints, e.g. the maximum torque limits. A regulator based on this type of pole assignment has been named the "best CoM-ZMP regulator" [143].

Two sample phase portraits of the regulator are shown in Fig. 5.9. They were generated with $x_g^{des} = 0$ and data set (5.17). With pole assignment $(q_1,\ q_2) = (-0.2,\ -0.2)$ and $(q_1,\ q_2) = (-1,\ -0.2)$, it can be confirmed that the constrained-control and dynamic stability regions will be intersecting and overlapping, as in the plot on the left and right side, respectively. When the xCoM is confined to the interior of the dynamic stability region, stable motion of the gCoM will be guaranteed. When the xCoM arrives at the boundary or leaves the region, reactive stepping should be initialized. This problem will be discussed in Section 7.7.5.

5.4.4 ZMP Stabilization in the Presence of GRF Estimation Time Lag

Most existing humanoid robots are position-controlled. The balance controllers discussed so far were designed for such robots. As already noted, their tasks are specified in terms of position/velocity. In order to implement a ZMP manipulation–type stabilizer, one has to account for the inherent GRF estimation lag in the actual GRF/ZMP readings. This lag results from the shock absorbing passive elements (rubber bushing) embedded into the robot soles, as well as from the GRF/ZMP sensor controller. For practical purposes, the following simple dynamic model can be used [71,98]:

$$x_p(s) = \frac{1}{T_p s + 1} x_p^{des}(s) \tag{5.32}$$

or

$$\dot{x}_p = -F_p(x_p - x_p^{des}), \tag{5.33}$$

where $F_p \equiv 1/T_p$. The time constant T_p can be determined experimentally. Combining this relation with CoM/ZMP dynamics (5.5), one arrives at the following linear state-space equation:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}u, \tag{5.34}$$

where $\boldsymbol{x} \equiv \begin{bmatrix} x_g & \dot{x}_g & x_p \end{bmatrix}^T$ is the state vector and

$$\boldsymbol{A} \equiv \begin{bmatrix} 0 & 1 & 0 \\ \omega^2 & 0 & -\omega^2 \\ 0 & 0 & -f_p \end{bmatrix}, \quad \boldsymbol{B} \equiv \begin{bmatrix} 0 \\ 0 \\ f_p \end{bmatrix}.$$

A tracking stabilizer can then be designed in a straightforward manner with the following control input:

$$u = x_p^{des} - \boldsymbol{K}\boldsymbol{e}_x,$$

where $\boldsymbol{e}_x \equiv (\boldsymbol{x}^{des} - \boldsymbol{x})$ is the tracking error. Feedback gains $\boldsymbol{K} \equiv \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}$ can be determined e.g. via pole placement, from the closed-loop dynamics

$$\frac{d}{dt}\boldsymbol{e}_x = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{e}_x.$$

This stabilizer was implemented under the LIP mode, whereby the stable pole of the pendulum ($\omega = \bar{\omega}$) was used to obtain the performance of the "best CoM/ZMP regulator" introduced in Section 5.4.3. The respective balance controller has been successfully tested by realizing a walking task with the HRP-4C humanoid robot [71].

5.4.5 Torso Position Compliance Control (TPCC)

The torso position compliance control (TPCC) [106] is a stabilization approach that uses the ZMP as a control input. The TPCC approach is similar to the ZMP manipulation–type approaches discussed in the previous subsections. The TPCC method will be derived below by representing the ZMP equation in terms of finite differences. TPCC comprises two components: (a) the IP control and (b) the ZMP compensation control.

(a) IP control: The 3D LIP model (cf. Section 4.4.2) illustrated in Fig. 5.10 will be employed in the design of the stabilizer; θ_y and θ_x (not shown in the figure) denote the rotation angles about the y- and x-axes, respectively. Assuming relatively small deviations of the 3D LIP from the vertical, the relationship between (θ_y, θ_x) and $\boldsymbol{r}_g - \boldsymbol{r}_p$ can be approximated as

$$\boldsymbol{\theta} \triangleq \begin{bmatrix} \theta_y \\ -\theta_x \end{bmatrix} \simeq \frac{1}{z_g}\left(\boldsymbol{r}_g - \boldsymbol{r}_p\right). \tag{5.35}$$

The linearized dynamics of the 3D LIP model around the vertical can be written as

$$z_g\ddot{\boldsymbol{\theta}} + \Delta\ddot{\boldsymbol{r}}_p = g\boldsymbol{\theta}; \tag{5.36}$$

$\Delta\ddot{\boldsymbol{r}}_p$, denoting a desired deviation of the ZMP, is considered as a control input.

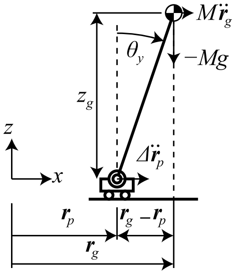

$M\ddot{\boldsymbol{r}}_g$
θ_y
$-Mg$
z_g
z
x
$\Delta\ddot{\boldsymbol{r}}_p$
$\boldsymbol{r}_p$
$\boldsymbol{r}_g - \boldsymbol{r}_p$
$\boldsymbol{r}_g$

FIGURE 5.10 3D LIP-on-cart model.

A simple PD controller can be designed as follows:

$$\Delta\ddot{\boldsymbol{r}}_p = K\left(\boldsymbol{\theta} + T_D\dot{\boldsymbol{\theta}}\right). \tag{5.37}$$

Substituting (5.37) into (5.36), the following second-order system is obtained:

$$\ddot{\boldsymbol{\theta}} + \frac{KT_D}{z_g}\dot{\boldsymbol{\theta}} + \frac{K-g}{z_g}\boldsymbol{\theta}. = \boldsymbol{0}. \tag{5.38}$$

This system can be stabilized with appropriate gains K $(> g)$ and T_D (> 0). Using the centered difference formula, the PD controller (5.37) is discretized as follows:

$$\Delta\boldsymbol{r}_p(t_{k+1}) = \Delta t^2 K\boldsymbol{\theta}(t_k) + \Delta t KT_D\left(\boldsymbol{\theta}(t_k) - \boldsymbol{\theta}(t_{k-1})\right) + \Delta\boldsymbol{r}_p(t_k) + \left(\Delta\boldsymbol{r}_p(t_k) - \Delta\boldsymbol{r}_p(t_{k-1})\right). \tag{5.39}$$

(b) ZMP compensation control: The IP controller determines the desired deviation of the ZMP. In order to track the desired deviation, a ZMP compensation controller is needed.

Recall the gCoM dynamics in 2D given in (5.11). Suppose that the ZMP $\boldsymbol{r}_p$ tracks the desired ZMP $\boldsymbol{r}_p^{des}$ by an increment in the CoM acceleration denoted as $\Delta\ddot{\boldsymbol{r}}_g$. The gCoM dynamics then assume the following form:

$$\ddot{\boldsymbol{r}}_g + \Delta\ddot{\boldsymbol{r}}_g = \omega^2\left(\boldsymbol{r}_g - \boldsymbol{r}_p^{des}\right). \tag{5.40}$$

Subtracting (5.11) from (5.40), the following equation is obtained:

$$\Delta\ddot{\boldsymbol{r}}_g = -\omega^2\Delta\boldsymbol{r}_p, \tag{5.41}$$

where $\Delta\boldsymbol{r}_p \triangleq \boldsymbol{r}_p^{des} - \boldsymbol{r}_p$. The above equation suggests that if $\Delta\boldsymbol{r}_p$ is positive, $\Delta\ddot{\boldsymbol{r}}_g$ must be negative (see Fig. 5.11). Using the centered difference formula, (5.41) is discretized as follows:

$$\Delta\boldsymbol{r}_g(t_{k+1}) = -\Delta t^2\omega^2\Delta\boldsymbol{r}_p(t_k) + \Delta\boldsymbol{r}_g(t_k) + \left(\Delta\boldsymbol{r}_g(t_k) - \Delta\boldsymbol{r}_g(t_{k-1})\right). \tag{5.42}$$

(c) Simplified TPCC: According to the two steps (a) and (b) above, first the desired deviation of the ZMP is calculated using (5.39) to stabilize a humanoid robot body; then the deviation

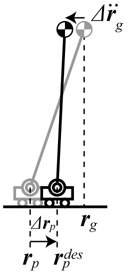


FIGURE 5.11 ZMP compensation control.

of the CoM is calculated using (5.42) to realize the desired deviation of the ZMP. It was experimentally confirmed in [106] that the simplified versions of the IP and the ZMP compensation controllers, written as

$$\Delta \boldsymbol{r}_p(t_{k+1}) = \Delta t^2 K \boldsymbol{\theta}(t_k) + \Delta t K T_D \left(\boldsymbol{\theta}(t_k) - \boldsymbol{\theta}(t_{k-1})\right) \tag{5.43}$$

and

$$\Delta \boldsymbol{r}_g(t_{k+1}) = -\Delta t^2 \omega^2 \Delta \boldsymbol{r}_p(t_k) + \Delta \boldsymbol{r}_g(t_k), \tag{5.44}$$

respectively, are also effective.

5.5 CAPTURE POINT–BASED ANALYSIS AND STABILIZATION

Recall the important result in the field of human balance control that the dynamic stability margin for such control is defined w.r.t. the xCoM (cf. Section 5.3). A similar result has been reported in the field of humanoid robotics, though from a different perspective. The goal was to determine the most appropriate foot placement when the change-of-BoS action (cf. Section 5.3.3) is to be undertaken to stabilize the posture after the application of an unknown disturbance. Such type of action is also known from studies on human postural stability, under the name “stepping strategy” or “stumbling strategy” [139,91,93]. This strategy will be discussed in Section 7.7.5.

5.5.1 Capture Point (CP) and Instantaneous Capture Point (ICP)

The special foot placement mentioned above was referred to as the “capture point” (CP) in [122] (see also [125]). The CP was obtained from the orbital energy pertinent to the LIP-on-cart model introduced in Section 4.3.1.

The CP is the only point *outside the BoS* where the LIP-on-cart model will become stationary after the step. In other words, at the CP, the gCoM coincides with a given CoP, $\bar{x}_p$, and also, the gCoM speed is zero; we have

$$x_g(t)|_{t\to\infty} = \bar{x}_p, \tag{5.45}$$

$$\dot{x}_g(t)|_{t\to\infty} = 0. \tag{5.46}$$

This result can also be derived [124,119] from the explicit solution of the gCoM dynamics (cf. (4.13)):

$$x_g(t) = \frac{1}{2}\left(x_{g0} - \bar{x}_p + \frac{v_{g0}}{\omega}\right)e^{\omega t} + \frac{1}{2}\left(x_{g0} - \bar{x}_p + \frac{v_{g0}}{\omega}\right)e^{-\omega t} + \bar{x}_p. \tag{5.47}$$

Hereby, the zero in the subscript notations signifies an initial value. The term that comprises the positive exponent is unstable; it has a destabilizing effect on the gCoM motion, as $t \to \infty$. This problem is tackled by letting the coefficient of the positive exponent become identically zero, so we have

$$x_{g0} - \bar{x}_p + \frac{v_{g0}}{\omega} = 0 \;\Rightarrow\; \bar{x}_p = x_{g0} + \frac{v_{g0}}{\omega}. \tag{5.48}$$

It is also worth noting that the above result can be confirmed via the explicit solution for the state-space representation of the gCoM dynamics, (5.16) [23]. Inserting (5.45) into the upper row of (5.16) and using the condition $|\tanh(\omega t)|_{t\to\infty} \to 1$, as in (5.1), it becomes apparent that $\bar{x}_p = x_{g0} + v_{g0}/\omega$. Then, inserting $\bar{x}_p$ into the lower row of (5.16), it is easy to confirm that condition (5.46) will also be satisfied.

The above result also implies that, with an arbitrary gCoM state $(x_g(t), \dot{x}_g(t))$, an expression that is *identical to the xCoM* (5.3) will be obtained. The only difference is in the natural angular frequency $\omega = \omega_{IP}$ for the linearized IP-on-foot model and $\omega = \overline{\omega}$ for the LIP-on-cart one.

Furthermore, note that in the above derivation, it was assumed that the step is taken *instantaneously*. This implies that an infinitely large ankle torque has to be employed. As will be shown in what follows, the above CP theoretical result can be implemented in motion generation and balance control schemes whereby the assumption will be relaxed. In other words, the step will be taken as fast as possible, but in finite time. The stepping time will depend on the maximum velocity and the torque limits of the robot actuators. When taking a step in finite time, the xCoM trajectory evolves in time as in (5.3). In [78] it was suggested to use the term "instantaneous capture point" (ICP); at the end of the step the ICP will coincide with the CP.

5.5.2 ICP-Based Stabilization

Recall the state-space representation of the gCoM and xCoM (or ICP) dynamics (cf. (5.9)):

$$\frac{d}{dt}\begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \omega \end{bmatrix}\begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} + \begin{bmatrix} 0 \\ -\omega \end{bmatrix} x_p. \tag{5.49}$$

The goal is to design a controller that can stabilize the unstable component, i.e. the xCoM/ICP dynamics in the lower row, via an appropriate control input $u = x_p$. This can be done via the explicit solution for the unstable component. Given a CP $\bar{x}_p$, the solution is

$$x_{ex}(t) = (x_{ex0} - \bar{x}_p)e^{\omega t} + \bar{x}_p. \tag{5.50}$$

The control input can be derived from this equation by replacing $x_{ex}(t) \leftarrow x_{ex}^{des}$, x_{ex}^{des} denoting the desired CP, $x_{ex0} \leftarrow x_{ex}(t)$, and $\bar{x}_p \leftarrow x_p(t)$. One then obtains [23]

$$u = \frac{x_{ex}^{des} - e^{\omega dT} x_{ex}}{1 - e^{\omega dT}}, \tag{5.51}$$

where dT is the (desired) time span until the arrival at the desired CP. In the following analysis, the convenient shorthand notation $b \equiv e^{\omega dT}$ will be introduced. The above control law can then be rewritten as

$$u = \frac{1}{1-b} x_{ex}^{des} - \frac{b}{1-b} x_{ex}. \tag{5.52}$$

It is straightforward to confirm that the unstable component of the xCoM/ICP dynamics, $\dot{x}_{ex} = \omega x_{ex}$, will be stabilized by the above control input via a negative feedback, with gain $b/(1-b) > 0$. A positive gain is obtained when dT is chosen to be positive. By inserting u into the state-space representation (5.9), one obtains the closed-loop dynamics as

$$\frac{d}{dt}\begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} = \begin{bmatrix} -\omega & \omega \\ 0 & \frac{\omega}{1-b} \end{bmatrix} \begin{bmatrix} x_g \\ x_{ex} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{\omega}{1-b} \end{bmatrix} x_{ex}^{des}, \tag{5.53}$$

with the eigenvalues $-\omega$ and $\omega/(1-b)$. The first eigenvalue is always stable, while the second one is stable for any $dT > 0$. This condition also determines the global stability of the system.

5.5.3 ICP Stabilization in the Presence of GRF Estimation Time Lag

Stabilizer (5.52) can be modified to deal with the GRF/ZMP lag discussed in Section 5.4.4. It will suffice to adjoin the first-order lag dynamics (5.33) to the system dynamics (5.49), by redefining the state-space vector $\boldsymbol{x}$ in (5.34) as $\boldsymbol{x} \equiv \begin{bmatrix} x_g & x_{ex} & x_p \end{bmatrix}^T$. The system matrices assume the form

$$\boldsymbol{A} \equiv \begin{bmatrix} -\omega & \omega & 0 \\ 0 & \omega & -\omega \\ 0 & 0 & -f_p \end{bmatrix}, \quad \boldsymbol{B} \equiv \begin{bmatrix} 0 \\ 0 \\ f_p \end{bmatrix}.$$

Inserting u from (5.52) into the system dynamics, one arrives at the closed-loop dynamics

$$\frac{d}{dt}\begin{bmatrix} x_g \\ x_{ex} \\ x_p \end{bmatrix} = \begin{bmatrix} -\omega & \omega & 0 \\ 0 & \omega & -\omega \\ 0 & -\frac{b}{1-b} f_p & -f_p \end{bmatrix} \begin{bmatrix} x_g \\ x_{ex} \\ x_p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{1-b} f_p \end{bmatrix} x_{ex}^{des}. \tag{5.54}$$

The eigenvalues of the system matrix $\boldsymbol{A}$ are

$$\lambda_{1,2} = \frac{\omega T_p - 1 \pm \sqrt{r}}{2T_p}, \quad \lambda_3 = -\omega,$$

where $r = (1 + \omega T_p)^2 + 4\omega T_p \, b/(1-b)$. Since λ_3 is always stable, the analysis focuses on the other two eigenvalues. Critical damping is achieved with $r = 0$. This condition determines

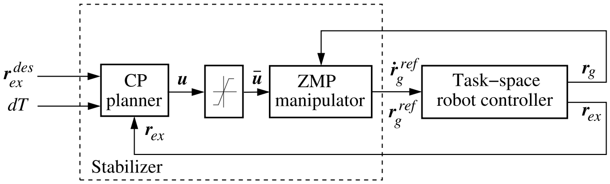


FIGURE 5.12 Block diagram of a CP type balance controller. With reference to Fig. 5.1, it is apparent that: (1) the task trajectory T is defined in terms of the CP motion and (2) the stabilizer comprises a CP planner and the "conventional" ZMP manipulator. The task-space controller makes use of the CoM inverse kinematics solution and includes a local joint-angle feedback controller. The actual CP is calculated from the actual gCoM and its velocity. The limiter restricts the control input to the BoS via a projection [23].

the following critical value for the time duration of the step:

$$dT_{cr} = \frac{2}{\omega} \ln \frac{1 + \omega T_p}{1 - \omega T_p}.$$

Furthermore, an oscillatory response stemming from the nonzero imaginary eigenvalues is undesirable. This case can be avoided as long as $dT > dT_{cr}$. On the other hand, note that dT_{cr} is a function of the time lag constant T_p. The influence of this constant becomes apparent from the eigenvalue behavior as dT grows to infinity, as we have $\lambda_1|_{dT \to \infty} = \omega - 1/T_p,\ \lambda_2|_{dT \to \infty} = 0$. From this result it can be concluded that the system will be stable with time lag constants $T_p < 1/\omega$.

It is worth noting that the above ICP stabilizer is robust w.r.t. a constant gCoM error Δx_g: the error results only in the constant control input offset $b\Delta x_g$. Since the system matrix remains unchanged, the global stability can be preserved [23].

5.5.4 ICP Dynamics and Stabilization in 2D

The implementation of the CP stabilizer in a real robot requires a 2D formulation. Such formulation is obtained in a straightforward manner from the decoupled dynamics of the 3D LIP model (cf. Section 4.4.2). The vector quantities are expressed in accordance with the notation introduced in Section 5.3.4. The xCoM/CP (5.3), the CP dynamics (5.8), and the CP based controller (5.52) are represented in 2D space as

$$\boldsymbol{r}_{ex}(t) = \boldsymbol{r}_g(t) + \frac{\dot{\boldsymbol{r}}_g(t)}{\omega}, \tag{5.55}$$

$$\dot{\boldsymbol{r}}_{ex} = \omega\left(\boldsymbol{r}_{ex} - \boldsymbol{r}_p\right), \tag{5.56}$$

$$\boldsymbol{r}_p^{ref} = \frac{1}{1 - e^{\omega dT}} \boldsymbol{r}_{ex}^{des} - \frac{e^{\omega dT}}{1 - e^{\omega dT}} \boldsymbol{r}_{ex}. \tag{5.57}$$

From the CP dynamics (5.56) it is apparent that the CoP $\boldsymbol{r}_p$ "pushes" the ICP $\boldsymbol{r}_{ex}(t)$ away, in the direction of the CoM velocity. Thus, geometrically the CoP and the ICP lie on the line determined by the CoM velocity.

The block scheme of the controller is shown in Fig. 5.12. An implementation of the method for gait generation and walking control is given in Section 7.2.1.

5.6 STABILITY ANALYSIS AND STABILIZATION WITH ANGULAR MOMENTUM COMPONENT

From the LIP-cart model–based stability analysis in Section 5.4 it became apparent that the underlying constant-height CoM constraint plays an important role in the proof of the stability. This constraint, however, limits the application of the model to flat-ground environments, theoretically. On the other hand, there always has been considerable interest in prospective applications within environments with irregular terrain. Basically, there are three possible approaches to deal with this problem. The first approach is based on the ZMP concept: find a properly defined virtual plane and project the ZMP on that plane [147,145,66,131,136,132,9]. The second approach also uses the ZMP, but in combination with the linear reaction wheel pendulum (LRWP)-on-foot model described in Section 4.3.3. This approach was suggested as an improvement to the original Model-ZMP method [48] used in Honda's robots [150]. The third approach is based on the extension of the ICP concept to three dimensions.

From a historical perspective, past studies on biped gait on flat ground focused almost exclusively on ZMP-based balance control, completely ignoring thereby the angular momentum component. A few exceptions deserve to be mentioned. In [148], variations in angular momentum generated indirectly via the optimized trunk motion were used to stabilize the gait. In [128,129], a simple angular momentum feedback controller was implemented to determine the ankle joint torque of the support leg during the gait. The importance of angular momentum control as a component of a more complex, optimization-based balance controller was revealed in [151,65]. The ankle torque–based angular momentum control was also experimentally tested with other proactive tasks, such as single-leg balance, squats, and kicking motion [74]. This work evolved into the "resolved momentum control" method for a whole-body motion control based on the CRB momentum [69]. It was shown that a stable gait pattern can be realized when the reference CRB angular momentum is set at zero. Later it was revealed that humans, indeed, actively regulate their CRB angular momentum (around zero) during walking on flat ground [120,39].

5.6.1 Stability Analysis Based on the LRWP Model

As shown in Chapter 4 with the LRWP models on the plane and in 3D, the rate of change of the CRB (centroidal) angular momentum alters the CoP by an additional component. The sum determines the "centroidal moment pivot" (CMP) defined in (4.25), (4.40) (see also (5.59) below). In this way, the centroidal moment (and hence the rate of change of the centroidal angular momentum) can be directly related to balance control. The LRWP model thus provides an additional control input that can be used to control the foot rotation occurring when the xCoM leaves the BoS, as depicted in Fig. 5.3D. Note that in this case, the CoP control input used in the ZMP methods is unavailable. The real robot can generate the desired centroidal moment via acceleration in its upper body joints, e.g. with trunk and arm "windmilling" rotation–type maneuvers, as observed in [82,92].

The gCoM dynamics can be obtained from (4.22) as

$$\ddot{x}_g - \omega^2 x_g = -\omega^2 x_{cmp}, \tag{5.58}$$

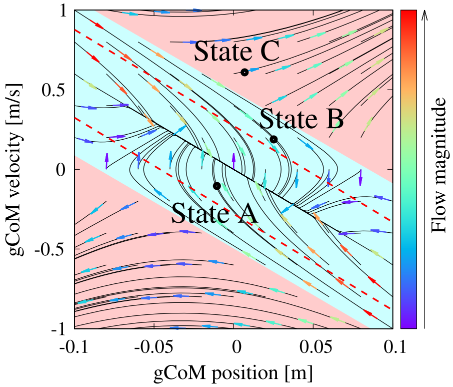


FIGURE 5.13 gCoM phase portrait for the LRWP-on-foot model. The RW torque effectively enlarges the dynamic stability region. The dynamic stability region with zero RW torque (dashed lines) is identical with that of the LIP-on-cart model in Fig. 5.5. State A is inside the LIP dynamic stability region (stable). State B, on the other hand, is outside this region (unstable). However, since State B is within the LRWP dynamic stability region, it would be possible to use a centroidal RW torque (e.g. arm "windmilling") to steer the state toward the LIP dynamic stability region. Finally, State C is outside the LRWP dynamic stability region (unstable). A change-of-BoS (e.g. a reactive step) is needed at this state to avoid a fall.

$$x_{cmp} \equiv x_p + x_{RW}, \tag{5.59}$$

$$x_{RW} \equiv \frac{m_c}{Mg} = \frac{I}{Mg}\ddot{\phi}_c. \tag{5.60}$$

Here x_{cmp} stands for the CMP; x_{RW} defines the additional component. Note that the definition assumes that the vertical CoM acceleration is negligible, s.t. $\ddot{z}_g \ll g$ and hence, the LIP constraint (4.16) holds. It turns out that this assumption can be validated with practical controllers for robots, not only on flat ground but also while negotiating irregular terrain [22,79, 161]. When compared to the gCoM dynamics of the LIP-on-cart model, (5.5), it is apparent that the reaction wheel contributes an additional forcing term. The results from the stability analysis in the previous sections apply directly; just replace x_p with x_{cmp} everywhere. For example, the xCoM dynamics, (5.8), become

$$\dot{x}_{ex} = \omega(x_{ex} - x_{cmp}). \tag{5.61}$$

Furthermore, note that with the LRWP model, the dynamic stability region (5.2) will be effectively enlarged as follows:

$$-x_{RW}^{max} - l_f \leq x_{ex}(t) \leq l_f + x_{RW}^{max}. \tag{5.62}$$

Here $x_{RW}^{max} \equiv m_c^{max}/(Mg)$, where m_c^{max} denotes the maximum RW torque. This is also true for the dynamic stability margin (5.4) and the critical time (5.10): just replace $l_f \to l_f + x_{RW}^{max}$ in the respective equations. Note also that by varying the RW torque from zero to m_c^{max}, the CP grows to a capture region with maximum area determined by (5.62). This is shown with the phase portrait in Fig. 5.13. It was generated with $m_c^{max} = 2$ Nm, in addition to the data set (5.17). The increased dynamic stability region is apparent, when compared to that of the LIP-on-cart model in Fig. 5.5. Three representative states are shown. State A, being within the LIP dynamic stability region (between the dashed lines), is dynamically stable and thus, balance control can be ensured only with the ankle torque. States B and C, on the other hand, are outside the LIP dynamic stability region. Thus, they are unstable. State B is within the LRWP dynamic stability region which means that an RW-type torque can be used to restore the dynamic balance. As already mentioned, such torque can be generated via the upper-body motion. There is no need to change the BoS in this case. Lastly, State C is outside the LRWP dynamic stability region. The only possible balance strategy in this case is the change-of-BoS one (e.g. reactive stepping or grasping a handrail).

To summarize, the LRWP model contribution to stability is threefold. First, it allows to deal with a rolling foot when the CoP control input is not available. Second, the dynamic stability region and margin, as well as the critical time limit, are enlarged when compared to those obtained with the LIP model. Third, when the foot/feet is/are in flat planar contact, the additional control input provided by the LRWP model increases the robustness of the balance controller w.r.t. disturbances stemming from external wrenches and/or unmodeled dynamics.

5.6.2 Stability Analysis in 3D: the Divergent Component of Motion

The stability analysis based on the linearized IP-on-foot model in Section 5.3.2 and the CP in Section 5.5 has shown that the gCoM comprises a stable and an unstable motion component. An important result was obtained: to achieve an overall stability it would be sufficient to stabilize the unstable component only. The question then is: to facilitate the implementation with a real robot, could this result be extended from 2D to 3D? In order to find an answer, the complete 3-DoF motion of the CoM has to be considered, instead of only the 2-DoF ground projection.

To this end, consider the following direct extension of the xCoM expression (5.3) to 3D:

$$\boldsymbol{r}_X \equiv \boldsymbol{r}_C + \frac{1}{\overline{\omega}_X}\dot{\boldsymbol{r}}_C, \tag{5.63}$$

where $\boldsymbol{r}_C = \begin{bmatrix} \boldsymbol{r}_g^T & z_g \end{bmatrix}^T$ is the CoM position (recall that $\boldsymbol{r}_g \in \Re^2$ stands for the gCoM position). The above expression was introduced in [21] under the name *divergent component of motion* (DCM), a term borrowed from [149]. The DCM is not bounded to the BoS; it floats in 3D space as the CoM does. Note also that in (5.63), the CoM motion is not subjected to any task-induced constraint, neither by a constant-length pendulum, as with the IP-on-foot model, nor by a constant-height one, as with the LIP model. Nevertheless, in the following derivations $\overline{\omega}_X$, denoting the natural angular frequency of the DCM dynamic system, is assumed constant. This condition can be relaxed, though, as shown in [55].

The CoM dynamics are obtained from (5.63) as

$$\dot{\boldsymbol{r}}_C = -\overline{\omega}_X(\boldsymbol{r}_C - \boldsymbol{r}_X). \tag{5.64}$$

The two components on the r.h.s. determine two vector field components: $-\overline{\omega}_X\boldsymbol{r}_C$ is stable (convergent); $\overline{\omega}_X\boldsymbol{r}_X$, on the other hand, is unstable (divergent); this is the DCM. Next, differentiate (5.63) w.r.t. time to obtain the DCM dynamics as

$$\dot{\boldsymbol{r}}_X = \dot{\boldsymbol{r}}_C + \frac{1}{\overline{\omega}_X}\ddot{\boldsymbol{r}}_C. \tag{5.65}$$

The CoM acceleration, appearing in the above formula, is determined by Newton's Third Law: $\ddot{\boldsymbol{r}}_C = \boldsymbol{f}_C/M$, $\boldsymbol{f}_C$ denoting the total *reaction* acting at the CoM; $\boldsymbol{f}_C$ results from the parallel shift of GRF $\boldsymbol{f}_r$ that acts at the CoP. The shifted GRF can be decomposed as follows:

$$\boldsymbol{f}_C = \boldsymbol{f}_I + \boldsymbol{f}_G, \tag{5.66}$$

$\boldsymbol{f}_G = Mg\boldsymbol{e}_z$ and $\boldsymbol{f}_I$ denoting the gravity and inertia reactions, respectively; $\boldsymbol{e}_z$, g, and M stand for the unit vector along the vertical, gravity acceleration, and total mass of the robot, respectively. The inertia reaction comprises three components stemming from the horizontal and the vertical CoM acceleration, as well as from a centroidal moment. The line of action of $\boldsymbol{f}_C$ intersects the (irregular) ground surface at the CMP, $\boldsymbol{r}_{cmp}$. As explained in Section 5.6.1 with the LRWP model (cf. (5.59)), a CMP that does not coincide with the CoP, i.e. $\boldsymbol{r}_{cmp} - \boldsymbol{r}_p \equiv \boldsymbol{r}_{RW} \neq \boldsymbol{0}$, is an indicator of the presence of a nonzero centroidal moment. In the 3D case, this moment is "generated" by a fictitious 3D RW assembly (cf. Section 4.4.3) around the normal to the plane determined by the CoP, the CMP, and the CoM. Note that $\boldsymbol{r}_{RW}$ is the modifier of the CMP obtained from the roll-pitch-yaw components of the centroidal moment.

Combining (5.64), (5.65), and (5.66), one obtains

$$\dot{\boldsymbol{r}}_X = \overline{\omega}_X \boldsymbol{r}_X - \overline{\omega}_X \boldsymbol{r}_C + \frac{g}{\overline{\omega}_X}\boldsymbol{e}_z + \frac{1}{M\overline{\omega}_X}\boldsymbol{f}_I. \tag{5.67}$$

The DCM, $\overline{\omega}_X \boldsymbol{r}_X$, is explicitly visible in the above equation. To deal with the divergence, ensure first the decoupling of the DCM from the CoM motion with an appropriate design of inertial reaction $\boldsymbol{f}_I$. To this end, in [22] the so-called "enhanced CMP" (eCMP) was defined. The eCMP, $\boldsymbol{r}_{ecmp}$, is located on the line of action of $\boldsymbol{f}_C$, s.t. $\boldsymbol{f}_C = k_a(\boldsymbol{r}_C - \boldsymbol{r}_{ecmp})$ holds; k_a is a positive constant to be determined as follows. First, the inertial reaction is expressed as

$$\boldsymbol{f}_I = \boldsymbol{f}_C - \boldsymbol{f}_G = k_a(\boldsymbol{r}_C - \boldsymbol{r}_{ecmp}) - Mg\boldsymbol{e}_z. \tag{5.68}$$

Then, insert the above $\boldsymbol{f}_I$ into (5.67) to obtain

$$\dot{\boldsymbol{r}}_X = \overline{\omega}_X \boldsymbol{r}_X + \frac{k_a - M\overline{\omega}_X^2}{M\overline{\omega}_X}\boldsymbol{r}_C - \frac{k_a}{M\overline{\omega}_X}\boldsymbol{r}_{ecmp} - \frac{g}{\overline{\omega}_X}\boldsymbol{e}_z. \tag{5.69}$$

The dependence of the DCM dynamics on the CoM motion can be nullified by choosing $k_a = M\overline{\omega}_X^2$. Then, (5.69) becomes

$$\dot{\boldsymbol{r}}_X = \overline{\omega}_X \left(\boldsymbol{r}_X - \boldsymbol{r}_{ecmp} - \bar{z}_{vrp}\boldsymbol{e}_z\right), \tag{5.70}$$

where $\bar{z}_{vrp} \equiv g/\overline{\omega}_X^2$. Constant $\bar{z}_{vrp}$ can be interpreted as the average CoM height obtained while the robot walks across irregular terrain [22]. Note that the definition of $\bar{z}_{vrp}$ implies that $\overline{\omega}_X = \sqrt{g/\bar{z}_{vrp}}$. This reveals that $\overline{\omega}_X$ has the meaning of a natural angular frequency for the DCM dynamics.

The above equation represents an important result clarifying that the DCM can be stabilized via the eCMP. The eCMP component, $-\bar{\omega}_X \boldsymbol{r}_{ecmp}$, can be characterized as a "DCM damper." There is also a constant damping component in the vertical direction stemming from the acceleration of gravity. For convenience, the sum of these two damping terms will be denoted as the so-called "virtual repellent point" (VRP) [22]. We have

$$\boldsymbol{r}_{vrp} \equiv \boldsymbol{r}_{ecmp} + \bar{z}_{vrp} \boldsymbol{e}_z. \tag{5.71}$$

The DCM dynamics and the shifted GRF can then be rewritten as

$$\dot{\boldsymbol{r}}_X = \bar{\omega}_X (\boldsymbol{r}_X - \boldsymbol{r}_{vrp}) \tag{5.72}$$

and

$$\begin{aligned} \boldsymbol{f}_C &\underset{(5.68)}{=} M\bar{\omega}_X^2 (\boldsymbol{r}_C - \boldsymbol{r}_{ecmp}) \\ &\underset{(5.70)}{=} M\bar{\omega}_X^2 \left(\boldsymbol{r}_C - \boldsymbol{r}_{vrp} + \frac{g}{\bar{\omega}_X^2} \boldsymbol{e}_z \right) \\ &= M\bar{\omega}_X^2 (\boldsymbol{r}_C - \boldsymbol{r}_{vrp}) + Mg\boldsymbol{e}_z, \end{aligned} \tag{5.73}$$

respectively. Note that since $\boldsymbol{f}_G = Mg\boldsymbol{e}_z$, from (5.66) it follows that $\boldsymbol{f}_I = M\bar{\omega}_X^2 (\boldsymbol{r}_C - \boldsymbol{r}_{vrp})$ must hold.

The above model implies the existence of a centroidal moment. We have

$$\boldsymbol{m}_C = [\boldsymbol{r}^{\times}_{\overleftarrow{PC}}] \boldsymbol{f}_C.$$

Here $\boldsymbol{m}_C$ comprises two components stemming from the inertia and gravity force components (cf. (5.66)):

$$\begin{aligned} \boldsymbol{m}_C &= \boldsymbol{m}_I + \boldsymbol{m}_G, \\ \boldsymbol{m}_I &= [\boldsymbol{r}^{\times}_{\overleftarrow{PC}}] \boldsymbol{f}_I, \\ \boldsymbol{m}_G &= Mg[\boldsymbol{r}^{\times}_{\overleftarrow{PC}}] \boldsymbol{e}_z. \end{aligned} \tag{5.74}$$

A geometrical interpretation of these results in 2D and 3D is shown in Fig. 5.14.

Furthermore, by introducing state-space coordinates $(\boldsymbol{r}_C, \ \boldsymbol{r}_X)$, the DCM dynamics (5.72) can be represented in state-space form as

$$\frac{d}{dt} \begin{bmatrix} \boldsymbol{r}_C \\ \boldsymbol{r}_X \end{bmatrix} = \begin{bmatrix} -\bar{\omega}_X & \bar{\omega}_X \\ 0 & \bar{\omega}_X \end{bmatrix} \begin{bmatrix} \boldsymbol{r}_C \\ \boldsymbol{r}_X \end{bmatrix} + \begin{bmatrix} 0 \\ -\bar{\omega}_X \end{bmatrix} \boldsymbol{r}_{vrp}. \tag{5.75}$$

The form of this equation is the same as that in (5.9) for the planar case. Note that in (5.75), the VRP determines the forcing term, in place of the CoP in (5.9).

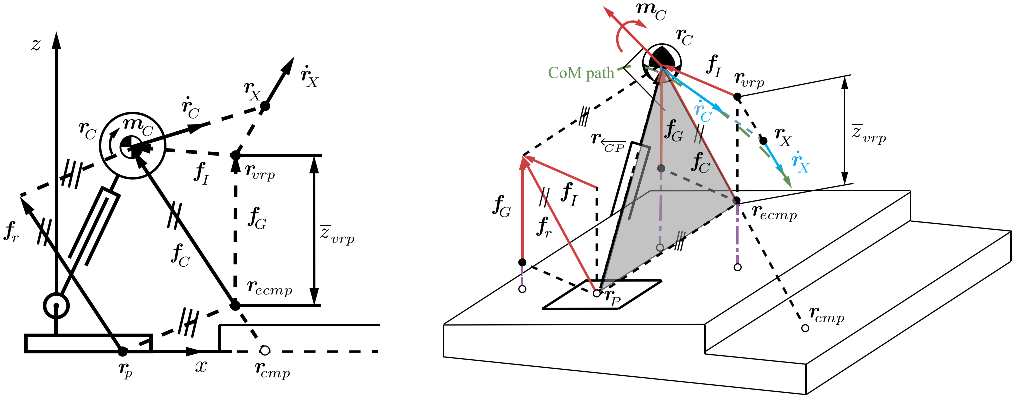


FIGURE 5.14 Geometrical interpretation of DCM relations with specific points: the CoP ($\boldsymbol{r}_p$), CoM ($\boldsymbol{r}_C$), CMP ($\boldsymbol{r}_{cmp}$), enhanced CMP (eCMP) ($\boldsymbol{r}_{ecmp}$), and VRP ($\boldsymbol{r}_{vrp}$) (based on [22]). Left (A): 2D representation. Right (B): 3D representation. All forces denote reactions (by convention reactions are positive). The system is stabilized via inertia reaction $\boldsymbol{f}_I$ that determines an appropriate VRP. The parallel shift of total GRF $\boldsymbol{f}_r = \boldsymbol{f}_C = \boldsymbol{f}_G + \boldsymbol{f}_I$ via the eCMP implies that the system is stabilized in the presence of a centroidal moment, $\boldsymbol{m}_C$, in the direction normal to the sagittal plane (2D case) or to the CoP-CoM-eCMP plane (the gray plane in the 3D figure).

5.6.3 DCM Stabilizer

The above stability analysis has led to the important result that the unstable DCM dynamics in 3D can be stabilized with an appropriate control law defined via the VRP. The envisioned balance controller should track the desired DCM $\boldsymbol{u} \equiv \begin{bmatrix} (\boldsymbol{r}_X^{des})^T & (\dot{\boldsymbol{r}}_X^{des})^T \end{bmatrix}^T$ faithfully. To this end, consider the following error dynamics [22]:

$$\dot{\boldsymbol{r}}_X^{ref} = \dot{\boldsymbol{r}}_X^{des} - k_x(\boldsymbol{r}_X - \boldsymbol{r}_X^{des}). \tag{5.76}$$

This system will be asymptotically stable with $k_x > 0$. The reference VRP can be expressed as (cf. (5.72))

$$\boldsymbol{r}_{vrp}^{ref} = \boldsymbol{r}_X - \frac{1}{\overline{\omega}_X}\dot{\boldsymbol{r}}_X^{ref}. \tag{5.77}$$

The closed-loop dynamics are derived in the conventional form

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{5.78}$$

$\boldsymbol{x} \equiv \begin{bmatrix} \boldsymbol{r}_C^T & \boldsymbol{r}_X^T \end{bmatrix}^T$ denoting the state vector. By inserting the control law (5.77) into the state-space equation (5.75) ($\boldsymbol{u} \equiv \boldsymbol{r}_{vrp}^{ref}$), one obtains

$$\boldsymbol{A} \equiv \begin{bmatrix} -\overline{\omega}_X\boldsymbol{E} & \overline{\omega}_X\boldsymbol{E} \\ \boldsymbol{0} & -k_x\boldsymbol{E} \end{bmatrix}, \quad \boldsymbol{B} \equiv \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ -k_x\boldsymbol{E} & \boldsymbol{E} \end{bmatrix}.$$

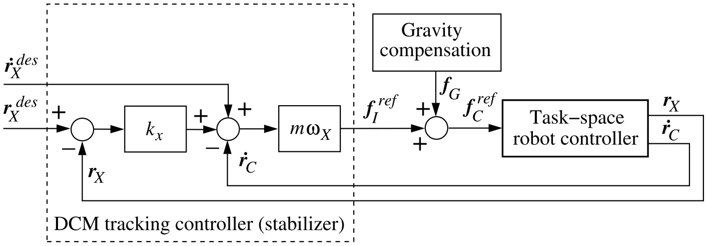


FIGURE 5.15 Block diagram of a DCM-type balance controller. With reference to Fig. 5.1, it is apparent that the task trajectory T is defined in terms of the DCM motion.

Since the system matrix $\boldsymbol{A}$ is stable for positive k_x and the control input $\boldsymbol{u}$ is assumed bounded, the above linear system can be characterized as bounded-input, bounded-output (BIBO)-stable.

The above result can be used to design a balance controller that incorporates the following reference (shifted) GRF:

$$\begin{aligned} \boldsymbol{f}_C^{ref} &\underset{(5.73)}{=} M\bar{\omega}_X^2\left(\boldsymbol{r}_C - \boldsymbol{r}_{vrp}^{ref} + \frac{g}{\bar{\omega}_X^2}\boldsymbol{e}_z\right) \\ &\underset{(5.77)(5.64)}{=} M\bar{\omega}_X\left(\dot{\boldsymbol{r}}_X^{ref} - \dot{\boldsymbol{r}}_C\right) + Mg\boldsymbol{e}_z. \end{aligned} \tag{5.79}$$

From (5.66) it follows that the first control term on the r.h.s. can be interpreted as a reference inertial force, $(\boldsymbol{f}_I)^{ref}$. This force ensures faithful tracking of the desired DCM, provided the BoS constraints are met. The block diagram of the DCM stabilizer is shown in Fig. 5.15. The design parameters are k_x and $\bar{\omega}_X = \sqrt{g/\bar{z}_{vrp}}$. The stabilizer is characterized with robustness against[4]: (1) a time-varying CoM motion tracking error, (2) a GRF/CoP estimation time-lag, (3) a mass estimation error, and (4) a constant external disturbance.

5.6.4 Summary and Conclusions

The DCM-based stabilization approach introduced above has been implemented with a number of real robots and has shown superior performance when compared to other balance controllers. The method has become indispensable in all areas of balance control such as walking gait generation and control, not only on flat ground but also on irregular terrain, and reactive stepping in response to an external perturbation. Implementation examples will be given in Chapter 7.

The method can be further improved by resorting to a time-varying reformulation, $\omega_X = \omega_X(t)$, as suggested in [55]. With such improvement, the somewhat fuzzy "average" CoM height notation $\bar{z}_{vrp}$ can be avoided. Any feasible desired vertical CoM trajectories can then be tracked under the same stability conditions. In a recent work, it has been shown how to use this improvement for a stretched-leg negotiation of irregular terrain [31].

[4] The robustness proof can be found in [22].

5.7 MAXIMUM OUTPUT ADMISSIBLE SET BASED STABILIZATION

The *maximum output admissible set* concept has been introduced as a linear control system approach whereby the state and/or the input are constrained [28]. The initial state of an unforced linear system is said to be output admissible w.r.t. a constraint set Y if the resulting output function satisfies the pointwise-in-time condition $y(t) \in Y,\ t \geq 0$. The set of all such possible initial conditions is called the maximum output admissible (MOA) set. It can then be guaranteed that all future states will comply with the constraints. Note that, as in the ICP/DCM methods, the initial state plays an important role.

The MOA set method is especially suitable for linear time-discrete systems. In this case, the MOA set[5] can be constructed iteratively, without a significant computational cost. Furthermore, the method can be used for systems with so-called exogenous inputs such as a large disturbance input [49]. From this brief introduction it becomes apparent that the MOA set method could be quite suitable for balance control, e.g. to enforce the BoS constraints during proactive tasks and, also, to ensure an appropriate reactive behavior in response to an external disturbance.

An implementation of the MOA set method for humanoid robot balance control has been pioneered in [165]. The implementation is based on the spherical IP model. Differently from the ICP-based stabilizer, the CoM is allowed to move in the vertical direction. A tracking stabilizer for the desired motion in this direction can be formulated as follows:

$$f_{rz}^{ref} = m(\ddot{z}_g + g) + k_{pv}(z_g^{des} - z_g) + k_{dv}(\dot{z}_g^{des} - \dot{z}_g), \tag{5.80}$$

k_{pv} and k_{dv} denoting the feedback gains. The control laws for the CoM motion in the horizontal directions, on the other hand, can be obtained via the gCoM/CoP dynamic relation (5.5). Since the spherical IP is linearized, the equation of motion in the x-direction is decoupled from that in y. It is important to note, though, that since relatively large vertical CoM accelerations are admissible (e.g. when running or jumping), the common coefficient in the gCoM/CoP dynamic relations is not constant (cf. (4.14)); we have

$$\omega(t) = \sqrt{\frac{\ddot{z}_g + g}{z_g}}.$$

Note also that under the assumption of zero disturbance in the vertical direction, it will be possible to obtain data sets $z_g(t)$ and $\omega(t)$ for the entire motion *in advance*.

Consider the state-space form of the gCoM/ZMP dynamic relation as given in (5.15) for the x-direction. It is straightforward to append a second equation of the same form for the y-direction. Furthermore, the recursive calculation of the MOA requires system discretization. The state-space equation for the two horizontal directions can then be written as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \boldsymbol{u}_k,$$

where $\boldsymbol{x}_k \equiv \begin{bmatrix} x_g(t_k) & \dot{x}_g(t_k) & y_g(t_k) & \dot{y}_g(t_k) \end{bmatrix}^T$ stands for the state-space vector at time t_k. We have matrices $\boldsymbol{A}_k = \Delta t \boldsymbol{A}(t_k) + \boldsymbol{E}$ and $\boldsymbol{B}_k = \Delta t \boldsymbol{B}(t_k)$, Δt is the sampling time, and $\boldsymbol{A}(t_k)$ and

[5] Also known as the constrained positively invariant (CPI) set [49].

$\boldsymbol{B}(t_k)$ are block-diagonal matrices with two identical block components apparent from (5.15). These matrices are functions of time since ω is a time variable. The control input is $\boldsymbol{u}_k$.

As a proactive task, consider a CoM motion regulator

$$\boldsymbol{u}_k = -\boldsymbol{K}\boldsymbol{x}_k,$$

where $\boldsymbol{K} \equiv \mathrm{diag}(k_{px},\ k_{dx},\ k_{py},\ k_{dy})$ is a feedback gain matrix. The closed-loop dynamics are obtained as

$$\boldsymbol{x}_{k+1} = \Delta\boldsymbol{A}_k\boldsymbol{x}_k, \tag{5.81}$$

where $\Delta\boldsymbol{A}_k \equiv \boldsymbol{A}_k - \boldsymbol{B}_k\boldsymbol{K}$. It is straightforward to determine $\boldsymbol{K}$ by well-known methods such as the pole placement or the LQ. Thus, the closed-loop dynamics will be asymptotically stable.

The MOA set can be constructed for an asymptotically stable system like the one in (5.81) iteratively. First, the control input is constrained within the BoS. Referring to (5.13),

$$\boldsymbol{B}_{sk}\boldsymbol{u}_k \preceq \boldsymbol{c}_k,$$

where $\boldsymbol{B}_{sk},\ \boldsymbol{c}_k$ determine the BoS at time instant t_k. Next, the state space is constrained within the MOA set defined as

$$O_\infty = \{\boldsymbol{x} : \boldsymbol{S}\boldsymbol{x} \preceq \boldsymbol{a}\}.$$

The MOA set can be represented as a convex polyhedron in state space. The iterative construction is summarized as

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{B}_s & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_s & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{B}_s \end{bmatrix} \begin{bmatrix} -\boldsymbol{K} \\ -\boldsymbol{K}\Delta\boldsymbol{A} \\ \vdots \\ -\boldsymbol{K}\Delta\boldsymbol{A}^l \end{bmatrix}, \quad \boldsymbol{a} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{c} \\ \vdots \\ \boldsymbol{c} \end{bmatrix},$$

where l denotes the iteration number. Given initial state $\boldsymbol{x}_0$, at the kth iteration $\boldsymbol{x}_k = \Delta\boldsymbol{A}^k\boldsymbol{x}_0$ and, hence, $\boldsymbol{u}_k = -\boldsymbol{K}\Delta\boldsymbol{A}^k\boldsymbol{x}_0$. Thus, $\boldsymbol{S}_k\boldsymbol{x}_0 \preceq \boldsymbol{a}$, where $\boldsymbol{S}_k$ is calculated from the above equation with $l = k$. The set $O_k = \{\boldsymbol{x} : \boldsymbol{S}_k\boldsymbol{x} \preceq \boldsymbol{a}\}$ is the kth output admissible set. If the initial state at t_0 belongs to this set, it will be guaranteed that all future states at $t_1,\ t_2, ..., t_k$ will satisfy the BoS constraint. Moreover, when the closed-loop system is asymptotically stable, as already confirmed, a finite number of iterations will be required, whereby l is determined by the condition $O_l = O_{l+1}$.

In addition to the regulator-type task, the performance of the MOA set stabilizer was also confirmed with a time-varying input task and BoS constraint, including a jumping/hopping task [165].

5.8 BALANCE CONTROL BASED ON SPATIAL MOMENTUM AND ITS RATE OF CHANGE

The spatial momentum–based balance control can ensure balance while performing a broad scope of tasks (both proactive and reactive) within various types of environment with coplanar, noncoplanar, and nonplanar, as well as with time-varying contact models. The method also plays an important role in whole-body control. Because of these advantages, the spatial momentum–based balance control has become a mainstream control approach in humanoid robotics and physically based computer animation [74,68,30,1,109,80,92,53,18,87,88,112,16,85,77,161,41].

Spatial momentum–based controllers comprise two main components: a stabilizer and a task-space controller (cf. Fig. 5.1). The stabilizer generates the input for the task-space controller in terms of spatial momentum or its rate of change. Stabilizers developed at an early stage used mainly the ZMP manipulation technique and were implemented in position-controlled humanoid robots (e.g. Honda's P2 [48]). Note that the ZMP manipulation–type stabilizer in Fig. 5.7 can be regarded as a linear momentum stabilizer since the produced control input is in terms of CoM velocity. Velocity-based spatial momentum balance controllers are simpler than dynamic balance controllers. Dynamic balance controllers make use of the rate of change of spatial momentum; they provide means for direct control of the contact/reaction wrenches, the overactuation in the closed chains and the CoPs. Velocity-based controllers do not have such capabilities. Nevertheless, they can control the net CoP, though only indirectly, as in the ZMP manipulation approach. It should be noted that the majority of the existing humanoid robots so far are position-controlled. Thus, velocity-based balance control based on spatial momentum plays an important role, as does dynamic balance control based on the rate of change of spatial momentum.

5.8.1 Fundamental Functional Dependencies in Balance Control

When designing a momentum-based balance controller it is important to have an in-depth understanding of the functional dependencies between the contact wrenches, the rates of change of linear/angular momenta, and the CoPs at the contact surfaces. Consider first the case of coplanar contacts at the feet. In this case, the tangential components of the rates of change of linear and angular momenta are relevant. These quantities, denoted as $\dot{\boldsymbol{p}}_t$ and $\dot{\boldsymbol{l}}_{C_t}$, respectively, appear in the ZMP/CoP equation, (4.38) (note that $\dot{\boldsymbol{p}}_t = \boldsymbol{f}_t = M\ddot{\boldsymbol{r}}_g$). The following functional dependencies exist:

$$\begin{aligned}
\dot{\boldsymbol{p}} &= \dot{\boldsymbol{p}}(\boldsymbol{f}_C),\\
\dot{\boldsymbol{l}}_{C_t} &= \dot{\boldsymbol{l}}_{C_t}(\dot{\boldsymbol{p}}_t, \boldsymbol{r}_p),\\
\boldsymbol{r}_p &= \boldsymbol{r}_p(\dot{\boldsymbol{p}}_t, \dot{\boldsymbol{l}}_{C_t}).
\end{aligned} \tag{5.82}$$

The first two dependencies are apparent from (4.38), the last one from the CRB dynamics (4.150). An important consequence of the above dependencies is that the rates of change of the two spatial momentum components are coupled. As apparent from (4.39) and (4.40), the coupling is via the CMP. We have

$$\begin{bmatrix} \dot{\boldsymbol{p}}_t \\ \dot{\boldsymbol{l}}_{C_t} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_t \\ \boldsymbol{m}_t \end{bmatrix} = M\omega_{IP}^2 \begin{bmatrix} (\boldsymbol{r}_g - \boldsymbol{r}_{cmp}) \\ -z_g \mathbb{S}^{\times}(\boldsymbol{r}_p - \boldsymbol{r}_{cmp}) \end{bmatrix} \in \Re^4. \tag{5.83}$$

This type of coupling exists not only for coplanar contacts, but also for noncoplanar ones. Indeed, using (5.73) and (5.74), it is straightforward to express the CRB momentum rates as

$$\dot{\hat{\mathcal{L}}}_C = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{l}}_C \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_I \\ \boldsymbol{m}_I \end{bmatrix} = M\omega_X^2 \begin{bmatrix} \boldsymbol{E} \\ [\boldsymbol{r}^{\times}_{\overleftarrow{PC}}] \end{bmatrix} (\boldsymbol{r}_C - \boldsymbol{r}_{vrp}) \in \Re^6. \tag{5.84}$$

From this equation, functional dependencies of the type (5.82) can be derived. The coupling between the two momentum rate components is apparently via $\boldsymbol{r}_{vrp}$.

The coupling problem plays an important role in balance controller design as well as in motion/force input generation for balance control. Consider as an example a humanoid robot in a double-stance posture. *Independent* control of the six spatial momentum components can only be achieved if all 12 leg actuators (6-DoF legs are assumed) are engaged. Thus, there is no actuator left to also control the net CoP, in an *independent* way. Attempting to design a balance controller with all control inputs being independent will result in an overconstrained system. The situation is similar when the robot is in a single-stance posture, when the 6D position of the upper body is to be controlled, in addition to that of the swing foot. It should be noted, though, that a rigorous assessment of the coupling problem from the viewpoint of input generation and control of balance is still missing. Various approaches have been suggested to deal with this problem, such as ignoring some of the angular momentum components (pitch and roll) [69], introducing weights [92] or prioritization [88,89]. These approaches have been shown to work in certain cases, but nevertheless, the ambiguity w.r.t. parameter settings (weights, priorities) remains.

A practical approach to the coupling problem is to design the momentum stabilizer in terms of the linear component only, based on a suitable reference CMP/VRP. The angular component will then be stabilized as a "byproduct" of the coupling, without the need to specify an explicit reference. This approach was used, in fact, in the DCM stabilizer described in Section 5.6.3. It was also used in [77,161]. The expression for $\dot{\boldsymbol{l}}_{C_t}$ in (5.83) shows that, as long as the difference between the CoP and the reference CMP is small, the angular momentum component will not contribute significantly to balance control (e.g. as with States (A)–(C) in Fig. 5.3). This is the usual situation while keeping balance at a standing posture or walking on regular terrain. In this case, the rate of change of angular momentum is quite small [120, 39]. This also implies that any proactive postural variations that *minimize (or conserve) the angular momentum* will not deteriorate the balancing capability. One example of such postural variation is the so-called "hip strategy," characterized by antiphase rotations in the leg and trunk body segments (cf. Section 7.6.3). On the other hand, note that when the xCoM leaves the BoS, as with State (D) in Fig. 5.3, a change of angular momentum via an appropriate upper-body maneuver will be needed to bring back the xCoM within the BoS limits [161]. Further details will be given in Section 5.8.6.

As a final remark it should be noted that the above dependencies are valid in the case of irregular terrain as well.

5.8.2 Resolved Momentum Control

The idea to design a balance controller based on the spatial momentum relation with the base quasivelocity, (4.91), was introduced in [69] under the name "resolved momentum control." The spatial momentum of the system is decomposed into three components stemming from the linear and angular motion of the base link ($\mathcal{V}_B$), the feet ($\mathcal{V}_{F_j}$), and the remaining joints ($\dot{\boldsymbol{\theta}}_{free}$). We have

$$\mathcal{L}_C(\boldsymbol{q}, \dot{\boldsymbol{q}}_B) = \mathcal{T}_B \begin{bmatrix} \mathcal{V}_B \\ \dot{\boldsymbol{\theta}}_{free} \end{bmatrix} + \sum_{j \in \{r,l\}} \mathcal{T}_j \mathcal{V}_{F_j}, \tag{5.85}$$

$\mathcal{T}_{(\circ)}$ denoting appropriate transforms that also account for the contact constraints. The following velocity control law was derived from the solution of the above equation:

$$\begin{bmatrix} \mathcal{V}_B \\ \dot{\boldsymbol{\theta}}_{free} \end{bmatrix} = \overline{\mathcal{T}}_B^{+} \left(\overline{\tilde{\mathcal{L}}_C^{ref} - \sum_{j \in \{r,l\}} \mathcal{T}_j \mathcal{V}_{F_j}^{ref}} \right) + (\boldsymbol{E} - \overline{\mathcal{T}}_B^{+} \overline{\mathcal{T}}_B) \begin{bmatrix} \mathcal{V}_B^{ref} \\ \dot{\boldsymbol{\theta}}_{free}^{ref} \end{bmatrix}. \tag{5.86}$$

Here $\tilde{\mathcal{L}}_C^{ref}$ and $\mathcal{V}_{F_j}^{ref}$ are reference values for the CRB spatial momentum and feet motion. These quantities constitute the high-priority task; $\mathcal{V}_B^{ref}$ and $\dot{\boldsymbol{\theta}}_{free}^{ref}$ are reference values for the base twist and the motion in the remaining joints (i.e. arm motion); they constitute the lower-priority task. The base twist $\mathcal{V}_B$, obtained from the above control law, is used to calculate the joint velocity in each leg. We have

$$\dot{\boldsymbol{\theta}}_{F_j} = \boldsymbol{J}^{-1}(\boldsymbol{q}_{F_j}) \mathcal{V}_{F_j} - \boldsymbol{J}^{-1}(\boldsymbol{q}_{F_j}) \mathbb{T}_{\overleftarrow{F_j B}} \mathcal{V}_B. \tag{5.87}$$

The overbar notation in (5.86) denotes a selective approach s.t. some of the components of spatial momentum are relaxed (ignored). This approach was introduced in [69] to avoid undesirable upper-body rotation observed when the above control law is applied to walking and kicking motion tasks. The problem with the original control law (without the relaxation) is that $\mathcal{V}_B^{ref}$ is constrained within the null space $\mathcal{N}(\mathcal{T}_B)$. This means that there is no guarantee that the desired CRB trajectories, specified by the reference base-link twist, will be tracked faithfully.

5.8.3 Whole-Body Balance Control With Relative Angular Momentum/Velocity

The resolved momentum framework was the result of a pioneering effort toward a velocity-based *whole-body balance control*. The control law formulation was flawed, however, since the coupling between the linear and angular momentum components was not addressed appropriately. As a result, not all of the components of the desired base-link twist could be controlled. Also, a rigorous stability assessment regarding the numerical instabilities reported in [69] was lacking. So far, stability assessment of velocity-based balance control has been done for simple IP models only, e.g. those used in the ZMP manipulation approach (cf. Section 5.4).

In this section, a *velocity-based whole-body balance controller* will be derived that can track the desired CRB trajectories with asymptotic stability. The design of the controller is based on the *momentum equilibrium principle* described in Section 4.7. In the case when the centroidal spatial momentum is expressed in terms of the mixed quasivelocity, the equilibrium relation can be written as (cf. (4.90))

$$\boldsymbol{H}_{CM}\dot{\boldsymbol{\theta}} = \mathbb{M}_C\mathcal{V}_C - \mathbb{M}_C\mathcal{V}_M. \tag{5.88}$$

The term on the l.h.s. is the coupling momentum; $\mathbb{M}_C\mathcal{V}_C$ and $\mathbb{M}_C\mathcal{V}_M$ denote the system and the CRB spatial momentum, respectively.[6] Since the rate of change of the SSM is determined by the total external force acting on the system (cf. (4.148)), it can be concluded that $\mathbb{M}_C\mathcal{V}_C$ stems from the reaction (contact) wrenches. Moreover, noting that the CRB (locked) inertia tensor is positive definite, the above momentum equilibrium can be recast as a *spatial velocity equilibrium* as follows:

$$\mathbb{M}_C^{-1}\boldsymbol{H}_{CM}\dot{\boldsymbol{\theta}} = \mathcal{V}_C - \mathcal{V}_M. \tag{5.89}$$

Here $\mathcal{V}_C \equiv \mathbb{M}_C^{-1}\mathcal{L}_C$ is the *centroidal twist*. The term on the l.h.s. , referred to as the *coupling spatial velocity*, is in balance with the *relative spatial velocity* $\Delta\mathcal{V} = \mathcal{V}_C - \mathcal{V}_M$ on the r.h.s.

Furthermore, recall that $\mathbb{M}_C^{-1}\boldsymbol{H}_{CM} = \begin{bmatrix}\boldsymbol{0}^T & \boldsymbol{J}_\omega^T\end{bmatrix}^T$ (cf. (4.97)). Thus, the above spatial velocity equilibrium relation can be represented componentwise as

$$\boldsymbol{v}_{C_I} = \boldsymbol{v}_{C_R}, \tag{5.90}$$

$$\boldsymbol{J}_\omega\dot{\boldsymbol{\theta}} = \boldsymbol{\omega}_C - \boldsymbol{\omega}_B \equiv \Delta\boldsymbol{\omega}, \tag{5.91}$$

where $\boldsymbol{J}_\omega(\boldsymbol{\theta}) = \boldsymbol{I}_C^{-1}(\boldsymbol{q})\boldsymbol{H}_C(\boldsymbol{q})$ (cf. (4.86)) and $\boldsymbol{\omega}_C = \boldsymbol{I}_C^{-1}\boldsymbol{l}_C$ is the *system angular velocity* (cf. (4.87)). The notation in the upper equation clarifies that the CoM velocity can be interpreted twofold: $\boldsymbol{v}_{C_I}$ is of *inertial* origin, while $\boldsymbol{v}_{C_R}$ stems from the *net* system twist that is of *reactive* origin. The origins can be distinguished only when the CoM motion is expressed in terms of acceleration, though (cf. Section 5.11.2). The velocities are indistinguishable, thus $\boldsymbol{v}_{C_I} = \boldsymbol{v}_{C_R} = \boldsymbol{v}_C,\ \Delta\boldsymbol{v}_C = \boldsymbol{0}$. Next, note that the lower equation expresses an *angular velocity equilibrium*: the *coupling angular velocity* $\boldsymbol{J}_\omega\dot{\boldsymbol{\theta}}$ is in balance with the *relative angular velocity* $\Delta\boldsymbol{\omega} \neq \boldsymbol{0}$. An important conclusion that can be made is that *the angular velocities of the system and the base link can be assigned in an independent way*. This conclusion clarifies why the flaw in the resolved momentum–type controller is avoidable.

It is assumed that the reference values for the CoM velocity and the angular velocity of the base link are known from the task assignment. The desired trajectories of the CRB twist, $\mathcal{V}_M^{des} = \begin{bmatrix}(\boldsymbol{v}_C^{des})^T & (\boldsymbol{\omega}_B^{des})^T\end{bmatrix}^T$, can be tracked with a conventional velocity controller derived from the constraint-consistent joint velocity solution (2.132). Then we have

$$\dot{\boldsymbol{\theta}}_1 = -\mathcal{J}_{cM}^{+}\mathbb{C}_{cC}^{T}\mathcal{V}_M^{ref} + \boldsymbol{N}(\mathcal{J}_{cM})\,\dot{\boldsymbol{\theta}}_u. \tag{5.92}$$

Note that $\overline{\mathcal{V}}^c = \boldsymbol{0}$ was used in the derivation, under the assumption of hard feet constraints on stationary ground; $\mathcal{V}_M^{ref}$ comprises the *independent* feedforward/feedback control compo-

[6] The respective abbreviations, SSM and CRB-SM, were introduced in Section 4.6.4.

nents

$$\boldsymbol{v}_C^{ref} = \boldsymbol{v}_C^{des} + K_{pC}\boldsymbol{e}_{pC}, \tag{5.93}$$

$$\boldsymbol{\omega}_B^{ref} = \boldsymbol{\omega}_B^{des} + K_{oB}\boldsymbol{e}_{oB}, \tag{5.94}$$

$\boldsymbol{e}_{pC} = \boldsymbol{r}_C^{des} - \boldsymbol{r}_C$ and $\boldsymbol{e}_{oB}$ denoting the CoM position error and the orientation error of the base link, respectively. The latter is expressed in any convenient form, e.g. with quaternions or using Euler's axis/angle formula (see e.g. [140], p. 139); $K_{(\circ)}$ are feedback gains. The control law (5.92) guarantees that $\mathcal{V}_M(t) = \mathcal{V}_M^{ref}(t)$ asymptotically, provided the contact states are maintained and the joint-space constraint Jacobian $\mathcal{J}_{cM}$ is full (row) rank.

The control input $\dot{\boldsymbol{\theta}}_1$ is useful in case of a double stance; the feet are fully constrained, s.t. $\mathbb{B}_c(\boldsymbol{q}_F) = \boldsymbol{E}$, $\mathbb{B}_m(\boldsymbol{q}_F) = \boldsymbol{0}$, while the hands are completely unconstrained, s.t. $\mathbb{B}_c(\boldsymbol{q}_H) = \mathbb{B}_m(\boldsymbol{q}_H) = \boldsymbol{0}$. In a single stance, the swing-foot motion control task could be embedded as a lower-priority task. To this end, determine the arbitrary joint velocity vector $\dot{\boldsymbol{\theta}}_u$ in (5.92) using the instantaneous-motion equation (2.129). The control joint velocity then assumes the form of (2.86), so we have

$$\dot{\boldsymbol{\theta}}_2 = -\mathcal{J}_{cM}^{+}\mathbb{C}_{cC}^{T}\mathcal{V}_M^{ref} + \overline{\mathcal{J}}_{mM}^{+}(\widetilde{\mathcal{V}}^{m})^{ref} + \boldsymbol{N}(\mathcal{J}_{cM})\boldsymbol{N}(\overline{\mathcal{J}}_{mM})\dot{\boldsymbol{\theta}}_u, \tag{5.95}$$

where $\overline{\mathcal{J}}_{mM} = \mathcal{J}_{mM}\boldsymbol{N}(\mathcal{J}_{cM})$ is the restricted end-link mobility Jacobian and

$$(\widetilde{\mathcal{V}}^{m})^{ref} = (\bar{\mathcal{V}}^{m})^{ref} + \left(\mathcal{J}_{mM}\mathcal{J}_{cM}^{+}\mathbb{C}_{cC}^{T} - \mathbb{C}_{mC}^{T}\right)\mathcal{V}_M^{ref}.$$

Here $(\bar{\mathcal{V}}^{m})^{ref}$ comprises a nonzero component for the swing leg, i.e.

$$\mathcal{V}_{F_{SW}}^{ref} = \mathcal{V}_{F_{SW}}^{des} + \boldsymbol{K}_{F_{SW}}\boldsymbol{e}_{F_{SW}}, \tag{5.96}$$

and the rest of the components of $(\bar{\mathcal{V}}^{m})^{ref}$ are zeros. The subscript SW stands for the swing leg, $\boldsymbol{e}_{F_{SW}}$ is the error twist, and $\boldsymbol{K}_{F_{SW}}$ is a feedback gain. Note that control input $\dot{\boldsymbol{\theta}}_2$ can also be used in the case of double stance, by adjusting the constraint conditions via the $\mathbb{B}_{(\circ)}$ matrices.

Relative Angular Momentum/Velocity (RAM/V) Balance Control

To obtain a controller with an enhanced balance control capability, it is recommendable to add a control term for the angular momentum. To this end, make use of the system angular velocity $\boldsymbol{\omega}_C$. Insert (5.95) into (5.91) and solve for the arbitrary $\dot{\boldsymbol{\theta}}_u$. Then, insert back into (5.92) to finally obtain the enhanced control law as

$$\begin{aligned}\dot{\boldsymbol{\theta}} &= -\mathcal{J}_{cM}^{+}\mathbb{C}_{cC}^{T}\mathcal{V}_M^{ref} + \overline{\mathcal{J}}_{mM}^{+}(\widetilde{\mathcal{V}}^{m})^{ref} + \overline{\boldsymbol{J}}_{\omega}^{+}\left(\Delta\boldsymbol{\omega}^{ref} - \tilde{\boldsymbol{\omega}}\right) + \boldsymbol{N}(\mathcal{J}_{cM})\boldsymbol{N}(\overline{\mathcal{J}}_{mM})\boldsymbol{N}(\overline{\boldsymbol{J}}_{\omega})\dot{\boldsymbol{\theta}}_u^{ref} \\ &= \dot{\boldsymbol{\theta}}^{c} + \dot{\boldsymbol{\theta}}^{m} + \dot{\boldsymbol{\theta}}^{am} + \dot{\boldsymbol{\theta}}^{n}, \quad \text{s.t. } \dot{\boldsymbol{\theta}}^{c} \succ \dot{\boldsymbol{\theta}}^{m} \succ \dot{\boldsymbol{\theta}}^{am} \succ \dot{\boldsymbol{\theta}}^{n}.\end{aligned} \tag{5.97}$$

Hereby, $\Delta\boldsymbol{\omega}^{ref} = \boldsymbol{\omega}_C^{ref} - \boldsymbol{\omega}_B^{ref}$,

$$\overline{\boldsymbol{J}}_{\omega} = \boldsymbol{J}_{\omega}\boldsymbol{N}(\mathcal{J}_{cM})\boldsymbol{N}(\overline{\mathcal{J}}_{mM}) \quad \text{and} \quad \tilde{\boldsymbol{\omega}} = \boldsymbol{J}_{\omega}\left(-\mathcal{J}_{cM}^{+}\mathbb{C}_{cC}^{T}\mathcal{V}_M^{ref} + \overline{\mathcal{J}}_{mM}^{+}(\widetilde{\mathcal{V}}^{m})^{ref}\right).$$

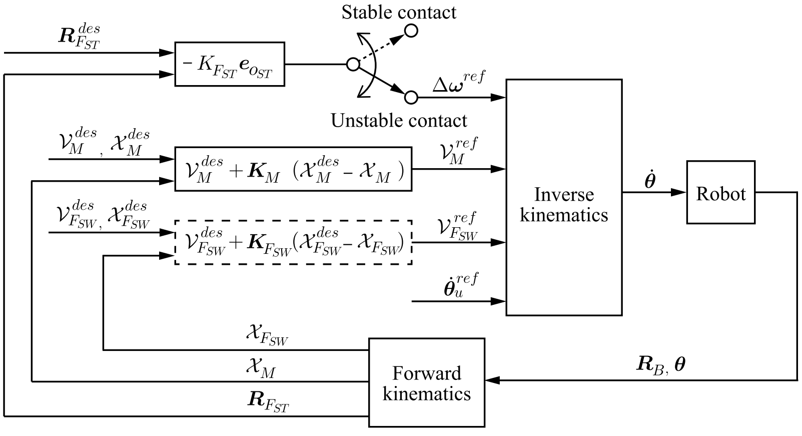

Stable contact
$\boldsymbol{R}_{F_{ST}}^{des}$
$-K_{F_{ST}}\boldsymbol{e}_{o_{ST}}$
$\Delta\boldsymbol{\omega}^{ref}$
Unstable contact
$\mathcal{V}_M^{des}, \mathcal{X}_M^{des}$
$\mathcal{V}_M^{des} + \boldsymbol{K}_M \ (\mathcal{X}_M^{des} - \mathcal{X}_M)$
$\mathcal{V}_M^{ref}$
Inverse kinematics
$\dot{\boldsymbol{\theta}}$
Robot
$\mathcal{V}_{F_{SW}}^{des}, \mathcal{X}_{F_{SW}}^{des}$
$\mathcal{V}_{F_{SW}}^{des} + \boldsymbol{K}_{F_{SW}}(\mathcal{X}_{F_{SW}}^{des} - \mathcal{X}_{F_{SW}})$
$\mathcal{V}_{F_{SW}}^{ref}$
$\dot{\boldsymbol{\theta}}_u^{ref}$
$\mathcal{X}_{F_{SW}}$
$\mathcal{X}_M$
$\boldsymbol{R}_{F_{ST}}$
Forward kinematics
$\boldsymbol{R}_B, \boldsymbol{\theta}$

FIGURE 5.16 Block diagram of the RAM/V controller. The Inverse kinematics block calculates the control joint velocity in accordance with (5.97). The swing leg (the quantities denoted by the $_{F_{SW}}$ subscripts) control is optional. The stance leg(s) (the quantities denoted by the $_{F_{ST}}$ subscripts) control is activated only when the contact(s) are unstable.

The control input $\dot{\boldsymbol{\theta}}$ in the last equation is composed of four components arranged in hierarchical order. The highest-priority component, $\dot{\boldsymbol{\theta}}^c$, is used to control the instantaneous motion of the CRB, via the contact constraints. The desired CRB translational (i.e. of the CoM) and rotational (of the base link) displacements are achieved via the movements in the leg(s). The rest of the control components are derived from within the null space $\mathcal{N}(\boldsymbol{\mathcal{J}}_{cM})$. These components will not disturb the main (the CRB) control task. The role of the second term, $\dot{\boldsymbol{\theta}}^m$, is to control the desired motion of the swing leg, when the robot is in a single stance. The role of the third term, $\dot{\boldsymbol{\theta}}^{am}$, is to control the system angular velocity in a way to ensure an appropriate inertial coupling w.r.t. the desired CRB rotational motion. Such coupling can only be achieved via motion in the arms since the legs and the upper body are controlled by the first two components. The last, fourth component, $\dot{\boldsymbol{\theta}}^n$, is used to enforce the joint velocity/angular constraints. To this end, the additional control input $\dot{\boldsymbol{\theta}}_u^{ref}$ can be determined via the gradient projection approach with the joint-limit avoidance potential function introduced in Section 2.7.4.

Note that when the robot is in a single stance and there is no desired motion task for the swing leg, the second component, $\dot{\boldsymbol{\theta}}^m$, becomes irrelevant. The motion of the swing leg will then be determined by the angular momentum component $\dot{\boldsymbol{\theta}}^{am}$. This means that the motion of the swing leg will contribute to postural stabilization, as does the motion in the arms. This contribution plays an important role when a large external disturbance is applied to the robot, as will be shown in Section 7.7.6.

The above controller will be referred to as the *relative angular momentum/velocity (RAM/V) controller*. The block diagram of the controller is shown in Fig. 5.16. The desired values for the CoM motion, the base-link rotation, the swing-leg motion, and the system angular momentum/velocity can be specified in an *independent* way. The controller does not account for the constraints (e.g. the CoP-in-BoS and friction cone constraints). But it provides means to avoid destabilization, via an appropriate arm (and possibly swing leg) motion generated by the relative angular velocity (RAV) control input $\Delta\boldsymbol{\omega}^{ref}$. Moreover, the controller can be endowed with a *self-stabilization* property. With such a property, the stability can be recovered even when the state is destabilized (e.g. a state with rolling foot/feet) by an inappropriate (desired) input or by an external disturbance of large magnitude, as will be explained below.

Special Cases: Balance Control That Conserves the System or the Coupling Angular Momentum

A straightforward approach to deal with the coupling problem in balance control is to constrain the motion of the robot within the angular momentum–conserving subset of motion (cf. (4.116)). This is achieved by setting the reference system angular velocity in (5.97) at zero throughout the motion. Then we have

$$\boldsymbol{\omega}_C^{ref}(t) = \mathbf{0} \Rightarrow \Delta\boldsymbol{\omega}^{ref} = -\boldsymbol{\omega}_B^{ref} = -(\boldsymbol{\omega}_B^{des} + K_{o_B}\boldsymbol{e}_{o_B}). \tag{5.98}$$

This means that the CoP (or ZMP) will depend only on the DCM stabilization task; the CoP will not be disturbed by the angular momentum task (i.e. the desired base-link rotation).

As an example, consider a humanoid robot in a double stance and performing a body twist motion task. During this task, the CoM is regulated to the initial position via control law (5.93). The torso twists around the vertical, first counterclockwise and then clockwise (twist angle ±10 degrees). The control of this motion is achieved via control law (5.94). The result is shown in animated form in Video 5.8-1 [43]. Note that the torso and arm rotations are antiphase; the arms are used to cancel out the moment component at the feet that is induced by twisting the torso. Thus, the CoP variation is almost zero (a slight deviation occurs because of the error terms).

Another possible approach to deal with the coupling problem is to make use of the instantaneous motion component $\dot{\boldsymbol{\theta}}_{cam}$ from within the Reaction Null Space (RNS) $\mathcal{N}(\boldsymbol{J}_\omega)$. In this case, the *coupling angular momentum* will be conserved at zero (cf. Section 4.7.3). This is achieved by setting the reference system angular velocity to equal the base angular velocity throughout the motion. Then we have

$$\boldsymbol{\omega}_C^{ref} = \boldsymbol{\omega}_B^{ref} \Rightarrow \Delta\boldsymbol{\omega}^{ref} = \mathbf{0}. \tag{5.99}$$

The resulting motion endows the RAM/V controller with the important property of *self-stabilization through angular momentum damping*. This property becomes apparent when the RAM/V controller is rewritten in terms of acceleration, as shown in Section 5.11.2. The property will be demonstrated below with an example.

Consider the same motion task as in the above example, performed with the RNS control constraint (5.99). The result is shown in an animated form in Video 5.8-2 [44]. Observe that, in this case, the torso and arm rotation is in-phase. That is, the arms are “actively” used to support the desired swift twisting motion of the torso.

For comparison, the same task was performed under the resolved momentum control law. The result is shown in an animated form in Video 5.8-3 [45]. Apparently, the desired swift torso twist is impossible to be achieved due to the reasons already explained.

A demonstration of a ball kicking task is shown in Video 5.8-4 [42]. There are three phases. First, the support foot is loaded by shifting the gCoM within the foot BoS. Thereby, the hands

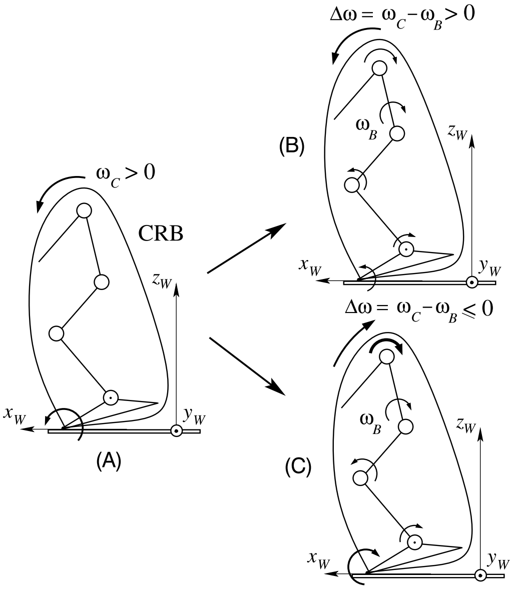


FIGURE 5.17 Stabilization with the relative angular velocity (RAV) $\Delta\omega$. (A) An unstable state of the CRB (joints locked) with positive system angular velocity ($\omega_C > 0$). (B) The system cannot be stabilized when the RAV is in the same direction as the system angular velocity. (C) A necessary condition for system stabilization is to generate an RAV in the direction opposite to the system angular velocity.

do not deviate significantly from their initial positions. Second, a prekick posture is attained whereby the kicking leg is extended to the side and the back. During this phase, the RNS control constraint is employed. This results in the arm motion being involved in the balancing. Then, during the final, kicking phase, the RAV constraint is switched from coupling to system angular momentum conservation. In this way robust balance control with minimum CoP deviation is ensured. As seen from the video, the motion looks quite "realistic." Note that the motion of the arms is determined solely by the two reference angular velocities, $\boldsymbol{\omega}_C^{ref}$ and $\boldsymbol{\omega}_B^{ref}$. This example demonstrates that a human-like performance of various dynamic motion tasks can be achieved by appropriately defining the relative angular velocity. This issue is an ongoing research effort.

5.8.4 RNS-Based Stabilization of Unstable Postures

Assume the robot has been destabilized, either proactively or by an external force. This means that the foot (when in a single stance) or the feet (when in a double stance) have begun to roll. If no swift action is taken, a fall will be unavoidable, as already clarified in Section 5.3.3 (cf. State D in Fig. 5.3).

The required swift action for contact stabilization can be generated in a straightforward way with the RAM/V controller introduced in Section 5.8.3. This will be explained with the help of the simple sagittal-plane model shown in Fig. 5.17. Assume the foot rolls around the toe tip counterclockwise, s.t. the system angular speed is positive in the chosen coordinate frame. Recall that when the joints are locked, $\omega_C = \omega_B = \omega_i$, $i \in \{\overline{1,n}\}$ is the angular speed of the CRB and also, the relative angular speed is zero, i.e. $\Delta\omega = \omega_C - \omega_B = 0$. When the robot links are allowed to rotate, in general the system angular speed will be different from that of the base link, and thus, the relative angular speed will be nonzero. For this particular example, when $\Delta\omega > 0$, the foot roll will persist and result in a fall. On the other hand, when $\Delta\omega \leq 0$, the foot will start rotating in the opposite (clockwise) direction, resulting in the recovery of the line contact and, eventually, of a stable posture.

Stabilization of Postures With Rolling Feet

Note that with the RNS input $\Delta\omega^{ref} = 0$ an appropriate arm motion will be generated to counter the foot rotation. This motion, however, does not yield contact stabilization but only a rocking motion, as shown in Video 5.8-5 [47]. To generate an arm motion that leads to contact stabilization, make use of the following control law:

$$\Delta\boldsymbol{\omega}^{ref} = -K_{F_{ST}} \boldsymbol{e}_{o_{ST}}, \tag{5.100}$$

where $\boldsymbol{e}_{o_{ST}}$ is the stance foot (the rolling foot) orientation error and $K_{F_{ST}}$ is a feedback gain. Motion control for contact stabilization with this control input is shown in Video 5.8-6 [46]. As seen in the video, the orientation of the feet converged to the desired state and the contacts could be stabilized. After the stabilization, the control input $\Delta\boldsymbol{\omega}^{ref}$ should be switched off to ensure that the arms do not move anymore.

Posture Stabilization on a Balance Board

As another example, consider a robot placed on a slightly damped balance board. High-friction contacts are employed to avoid a slip at the feet. The RAM/V controller with the RNS constraint is applied to stabilize the unstable states attained whenever the CoM of the robot is displaced from the (unstable) set of equilibrium points, i.e. from the vertical passing through the center of rotation of the balance board. The CoM motion control task is a regulation in the horizontal direction toward the equilibrium line. The base-link rotation task is also a regulation one: keeping the upper body upright. It is assumed that the actual base-link angular velocity is obtained from the IMU of the robot. Thus, there is no need to account for the rotational displacement of the balance board.

Snapshots from the simulation are displayed in Fig. 5.18. The result is shown in an animated form in Video 5.8-7 [107] and in graphic form in Fig. 5.19. Note that the initial posture is unstable. It takes about five seconds to stabilize the posture. Immediately after the stabilization, the robot destabilizes itself with a swift forward bend in the hips. Apparently, the RAM/V controller can handle such type of destabilization as well.

Summary and Conclusions

The above two examples, as well as those introduced in Section 5.8.3, demonstrate that the RAM/V controller can play a significant role in balance stability and control, especially when the angular momentum damping component from the RNS is used. The formulation is simple and yet quite efficient—there is no need to modify the contact model to account for the transitions between the stable and unstable states. When the contacts are stable, the RNS-based control input (5.100) should be switched off to avoid unnecessary arm movements. When the

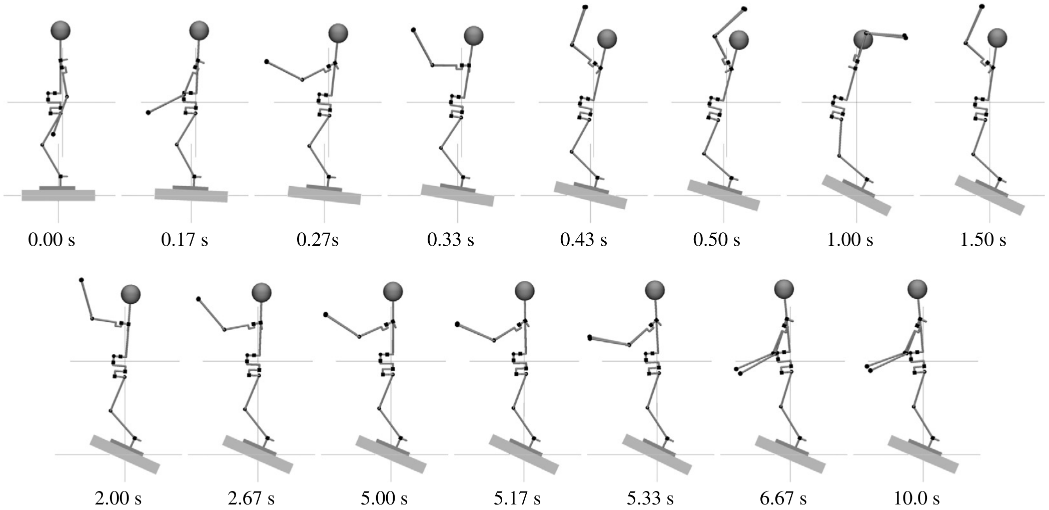


FIGURE 5.18 Human-like posture stabilization on a balance board under RAM/V plus RNS control. The initial posture is unstable since the CoM is displaced from the vertical passing through the center of rotation of the balance board. The posture is stabilized swiftly via the upper-limb motion (the trunk is commanded to keep its upright orientation). At $t = 5$ s the robot self-destabilizes by a fast forward bent. The final posture at $t = 10$ s represents a stable, static state.

contacts are unstable (the feet rotate), the control input can be used to ensure the recovery of the stability via the induced arm motion. Thereby, the system is forced to behave as a CRB, for any desired base-link state. The recovery is swift, with a subtle arm motion, for relatively small deviation of the feet from their equilibrium states. For larger deviations, the recovery can be prolonged; this results in a "windmilling" arm motion pattern for stabilization that is also sometimes used by humans to stabilize their posture in a critical situation.

The performance of the RAM/V controller can be further improved by a reformulation in terms of acceleration (to be presented in Section 5.11.2). With such a reformulation it becomes possible to control the angular momentum damping injected into the system. With such damping, the convergence to a stable contact state can be ensured, e.g. without the direct involvement of the foot rotation error, as in the rolling feet example.

5.8.5 An Approach to Contact Stabilization Within the Resolved Momentum Framework

The stabilization of the contacts can be ensured by introducing the friction cone constraints into the resolved momentum framework [50]. An appropriate CRB (reaction) wrench can then be generated that relates to the spatial momentum. The CRB wrench also relates to the reference CoP via the tangential GRM components. In the special case of a flat ground, the method yields the ZMP equation (4.38). Furthermore, the method provides a solution to the coupling problem as well. The solution comprises two steps, as apparent from Fig. 5.20. The first step involves the time derivatives of (5.85) and (5.86). Given the desired motion of the base link,

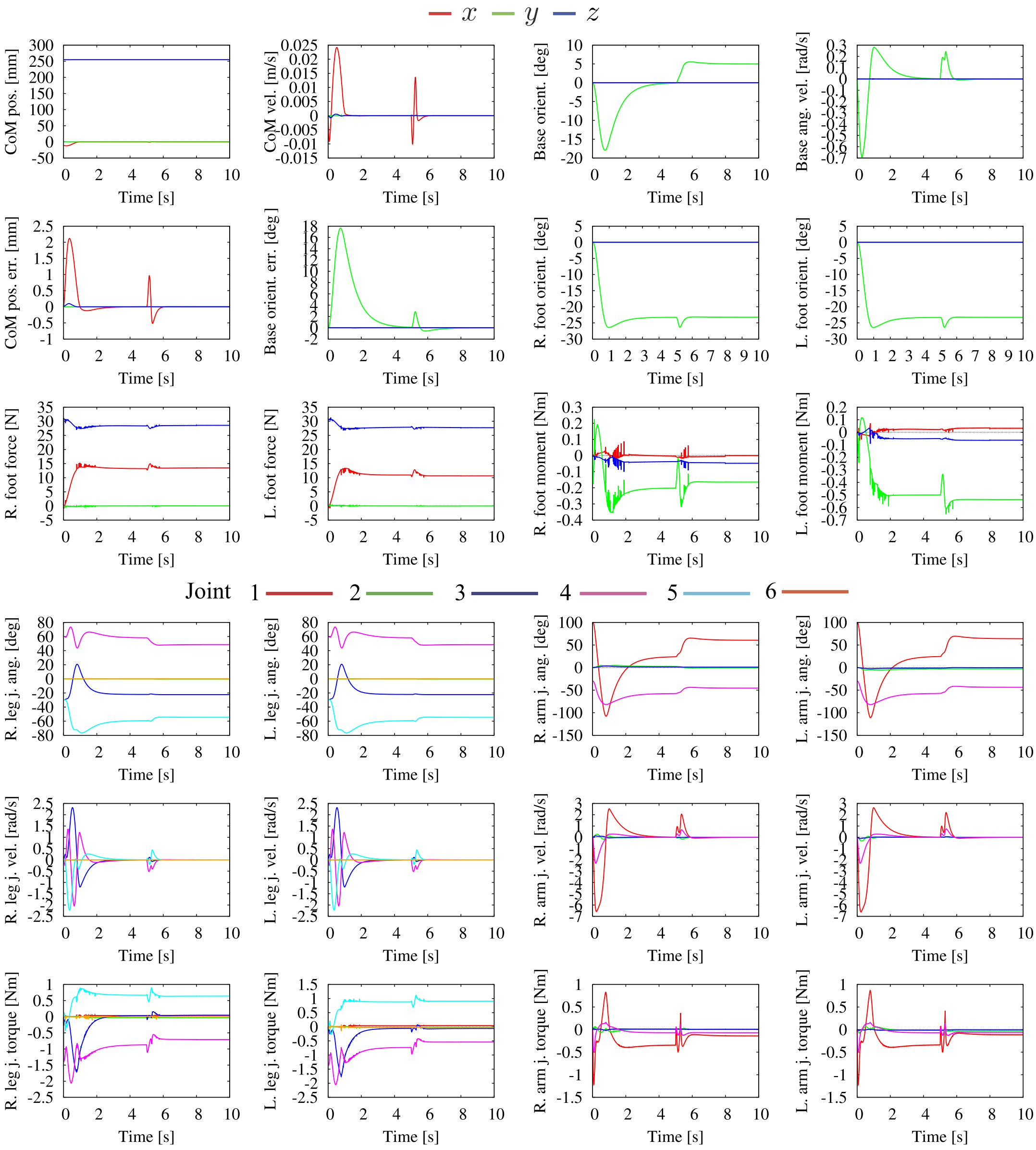


FIGURE 5.19 RNS-based stabilization on a balance board. Initially, the CoM is slightly displaced from the vertical equilibrium line passing through the balance board center, yielding an unstable posture. Stabilization of the base link requires about 5 s. After that, the robot self-destabilizes by a fast forward bent. The stabilization property of the controller can be reconfirmed for such type of destabilization. The joint numbering is as in the model described in Section A.1.

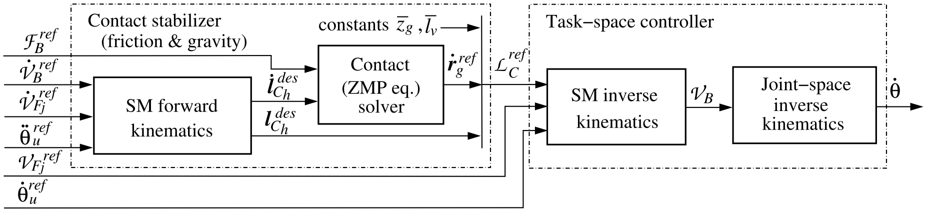


FIGURE 5.20 Block diagram of a Resolved-momentum (RM) trajectory generator [69], modified to account for friction at the contacts, via CRB contact wrench $\mathcal{F}_B^{ref}$ [50]. The spatial momentum (SM) forward kinematics block uses (5.85). The SM and joint-space inverse kinematics blocks refer to (5.86) and (5.87), respectively.

the end links (along the unconstrained motion directions), and additional posture variations in terms of acceleration, the desired *angular momentum rate* components in the horizontal directions $\dot{\boldsymbol{l}}_{C_h}^{des}$ are determined. Since the time derivative of (5.85) does not include information about the contact constraints, they will be ignored at the first step. At the second step, $\dot{\boldsymbol{l}}_{C_h}^{des}$ is inserted into the ZMP equation. This equation is then solved (under the LIP mode constraint) for the reference CoM trajectory that will account for the contact constraints and the desired momenta. The solution is obtained with the help of the preview control approach [68].

The task-space controller for resolving the CRB motion thus obtained is designed at the velocity level, based on the momentum equation (5.85) (cf. Fig. 5.20). Given the reference angular and linear momentum trajectories obtained at the first and second steps of the motion generation algorithm with contact constraints, the equation is resolved for the base quasivelocity $\mathcal{V}_B$. Finally, the control input for the joints rates, $\dot{\boldsymbol{\theta}}$, is obtained from (5.87) with $\mathcal{V}_B$. From the block diagram in Fig. 5.20 it should be apparent that the reference spatial acceleration of the base, $\dot{\mathcal{V}}_B^{ref}$, is modified by the algorithm in order to account for the contact conditions. In this way, the inherent inconsistency of the independently specified spatial momentum components can be resolved.

The performance of the resolved momentum controller was experimentally verified while tele-operating the end-link movements of a humanoid robot. Thereby, the base-link motion was generated autonomously to enlarge the workspace of the end link. The contacts were stabilized, also autonomously, via the ZMP equations [109]. The method has also been tested in a multicontact scenario with gait stabilization while holding a handrail [80].

5.8.6 Spatial Momentum Rate Stabilization Parametrized by the CMP/VRP

Because of the existing coupling between the rates of change of the linear and angular momenta, as mentioned in Section 5.8.1, it would be sufficient to formulate the stabilizer in terms of the linear component only. First, assume that the environment model includes a flat ground. In this case, the LIP mode constraint $\bar{\omega} = \sqrt{g/\bar{z}_g} = \text{const}$ can be invoked and consequently, the stability results from the preceding sections can be applied. The reference momentum rate is obtained from the upper row of (5.83) as

$$\dot{\boldsymbol{p}}_t^{ref} = M\bar{\omega}^2(\boldsymbol{r}_g - \boldsymbol{r}_{cmp}^{ref}). \tag{5.101}$$

The reference CMP can be derived from the xCoM/ICP dynamics of the LRWP, (5.61), as

$$\boldsymbol{r}_{cmp}^{ref} = \boldsymbol{r}_{ex} - \frac{1}{\bar{\omega}}\dot{\boldsymbol{r}}_{ex}^{des} + k_x(\boldsymbol{r}_{ex} - \boldsymbol{r}_{ex}^{des}), \tag{5.102}$$

k_x denoting a feedback gain. The angular momentum rate component will also be parametrized by $\boldsymbol{r}_{cmp}^{ref}$ because of the coupling, as already noted. The output of this CMP stabilizer is determined by inserting the CMP control law (5.102) into (5.101). The rate of change of the angular momentum does not have to be computed explicitly; it will be stabilized as a "byproduct" of the coupling, as noted in Section 5.8.1.

Furthermore, it is interesting to note that the above control law has been shown to perform well not only on flat ground but also on irregular terrain, using e.g. the edges of concrete blocks as footholds [161]. In critical situations, i.e. whenever the xCoM leaves the BoS, the angular momentum component, stabilized via the coupling with (5.102), will ensure a suitable upper-body lunge maneuver to drive the xCoM back toward the BoS. This result may not be too surprising, though, when noticing that the form of control law (5.102) is the same as that of the VRP control law (5.77). Recall that the latter was developed for irregular terrain without imposing the LIP mode constraint.

5.8.7 CRB Motion Trajectory Tracking With Asymptotic Stability

The CMP/VRP stabilizer has shown superior performance in locomotion on irregular terrain [22], including stepping with minimalistic footprints derived from approximated line- and point-type contacts [161]. This stabilizer, however, does not control the rate of change of the angular momentum independently from that of the linear momentum. Thus, there is no possibility to invoke proactive upper-body maneuvers with a desired angle, such as flexion/extension (i.e. forward/backward bending), inversion/eversion (i.e. side bending), and rotations around the vertical. These types of maneuvers or any combinations thereof are needed when performing such proactive tasks as picking up an object from the floor or reaching out for a target. Such maneuvers are also used in synergy-based accommodation of external perturbations during reactive balance tasks, as will be discussed in Section 7.6. The CRB momentum controllers discussed so far in the literature lack the capability of asymptotic tracking of the arbitrary angular momentum trajectories. In [88,112], for example, there is no angular orientation feedback at all. Also a proof of stability of the proposed controller is missing. This is also the case in [41]. In what follows, a CRB spatial momentum controller will be designed that ensures asymptotic tracking of any *constraint-consistent CRB motion trajectories*.

The CRB motion trajectories are specified via the rate of change of the spatial momentum. This quantity can be derived from the spatial dynamics component of the equation of motion. For simplicity and without loss of generality, the derivation will be based on the notation with the mixed quasivelocity (cf. Section 4.8.2). Implementations in terms of the base quasivelocity are discussed in Sections 5.10.4 and 5.12.1. The spatial dynamics, given in the upper part of (4.157), can be written as

$$\dot{\mathcal{L}}_C + \mathcal{G}_C = \mathcal{F}_C, \tag{5.103}$$

where $\dot{\mathcal{L}}_C \equiv \frac{d}{dt}\mathcal{L}_C(\boldsymbol{q},\dot{\boldsymbol{q}}_M)$. The wrench $\mathcal{F}_C \equiv \mathbb{C}_{cM}\overline{\mathcal{F}}^c$ is the net body wrench. This wrench, henceforth referred to as the *system wrench*,[7] drives the spatial dynamics. The reference rate of change of the spatial momentum can then be determined as (cf. (4.146)) as follows:

$$\dot{\tilde{\mathcal{L}}}_C^{ref} = \tilde{\mathbb{M}}_C \dot{\mathcal{V}}_M^{ref} + \dot{\tilde{\mathbb{M}}}_C \mathcal{V}_M. \tag{5.104}$$

Recall that the overtilde notation stands for the *CRB spatial momentum* (CRB-SM), that is, the system spatial momentum (SSM) component in the locked-joint case. Also recall that throughout this chapter it is assumed that the desired trajectories for the stabilizer are feasible linear and angular CRB motion trajectories. Such trajectories can be tracked under the conventional feedforward plus PD feedback control. We have

$$\dot{\boldsymbol{v}}_C^{ref} = \dot{\boldsymbol{v}}_C^{des} + K_{vC}\dot{\boldsymbol{e}}_{pC} + K_{pC}\boldsymbol{e}_{pC}, \tag{5.105}$$

$$\dot{\boldsymbol{\omega}}_B^{ref} = \dot{\boldsymbol{\omega}}_B^{des} + K_{\omega B}\boldsymbol{e}_{\omega B} + K_{oB}\boldsymbol{e}_{oB}. \tag{5.106}$$

Here $\boldsymbol{e}_{\omega B} = \boldsymbol{\omega}_B^{des} - \boldsymbol{\omega}_B$ denotes the error in the angular velocity of the base link, its orientation error $\boldsymbol{e}_{oB}$ and the CoM position error $\boldsymbol{e}_{pC}$ being defined in Section 5.8.3. The $K_{(\circ)}$ quantities stand for the PD feedback gains. The above two reference components constitute the reference twist, $\dot{\mathcal{V}}_M^{ref} = \begin{bmatrix}(\dot{\boldsymbol{v}}_C^{ref})^T & (\dot{\boldsymbol{\omega}}_B^{ref})^T\end{bmatrix}^T$; they are determined in a form widely used in end-link trajectory tracking with fixed-base robots. In the case of balance control of a floating-base humanoid robot, however, it is much more preferable to use the DCM stabilization result derived in Section 5.6.3. This result can be expressed in terms of the CoM acceleration as

$$\dot{\boldsymbol{v}}_C^{ref} = \omega_X\left(\dot{\boldsymbol{r}}_X^{des} + k_x(\boldsymbol{r}_X^{des} - \boldsymbol{r}_X) - \dot{\boldsymbol{r}}_C\right). \tag{5.107}$$

In the derivation, relations (5.76) and (5.79) were used.

The reference body wrench can be written as

$$\mathcal{F}_C^{ref} = \dot{\tilde{\mathcal{L}}}_C^{ref} + \mathcal{G}_C. \tag{5.108}$$

The role of this wrench is to compensate the effect of the gravity force and to linearize the CRB trajectory error dynamics. This is a necessary condition to achieve an asymptotic CRB motion trajectory tracking. Sufficiency depends on the feasibility of the desired CRB motion trajectories provided by the motion generator and the on-line motion modifier that uses feedback from the task-space controller (cf. Fig. 5.1). The latter imposes various motion/force constraints to ensure appropriate balance control. Details will be presented in Section 5.9.

5.9 TASK-SPACE CONTROLLER DESIGN FOR BALANCE CONTROL

The momentum rate–based balance controllers comprise task-space controllers that admit the rate of change of the spatial momentum (or equivalently, a body wrench) as the reference

[7] In [8], the term "gravito-inertial wrench" was used.

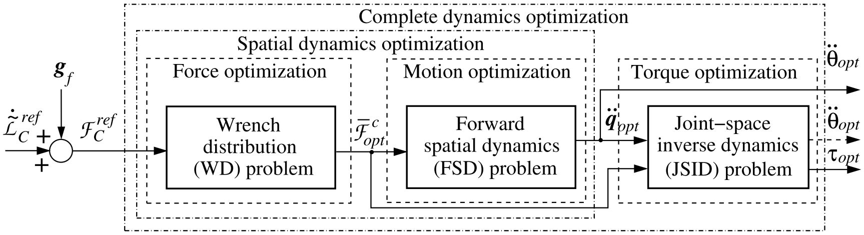

Complete dynamics optimization
Spatial dynamics optimization
Force optimization
Motion optimization
Torque optimization
$\boldsymbol{g}_f$
$\dot{\tilde{\mathcal{L}}}_C^{ref}$
+
+
$\mathcal{F}_C^{ref}$
Wrench distribution (WD) problem
$\overline{\mathcal{F}}_{opt}^c$
Forward spatial dynamics (FSD) problem
$\ddot{\boldsymbol{q}}_{opt}$
Joint–space inverse dynamics (JSID) problem
$\ddot{\theta}_{opt}$
$\ddot{\theta}_{opt}$
τ_{opt}

FIGURE 5.21 Generic task-space controller structure for optimization-based balance control. The momentum rate input $\dot{\mathcal{L}}_C^{ref}$ (or alternatively, $\mathcal{F}_C^{ref}$) is supplied by a stabilizer. The output of the controller can be specified either in terms of the joint torque or the joint acceleration, suitable for torque- or position-controlled robots, respectively.

input. A task-space controller design should account for the fact that the spatial dynamics do not *directly* depend on the joint torque. The reference input should be resolved to obtain joint acceleration as control input. In the case of a position-controlled humanoid robot, this control input is integrated twice to obtain joint angles to be fed to the control system. In the case of a torque-controlled robot, on the other hand, the joint torques for control can be derived from the joint-space partial dynamics component (the lower part of the equation of motion, e.g. as in (4.158)). This means that, in addition to the joint acceleration, the contact wrenches should be made available, i.e. either measured or calculated.

5.9.1 Generic Task-Space Controller Structure

The above considerations determine the structure of the task-space controller shown in Fig. 5.21. The input, supplied by the stabilizer, is the reference spatial momentum rate $\dot{\tilde{\mathcal{L}}}_C^{ref}$ or, alternatively, the reference CRB wrench $\mathcal{F}_C^{ref}$. The controller first generates the contact wrenches by solving the quasistatic *wrench distribution (WD) problem*. As discussed in Section 3.5.2, this is an underdetermined problem that is formulated as a constrained optimization task in the force domain. The solution is found either noniteratively (i.e. via generalized inverses) or iteratively (via a general solver). The constraints stem from the friction cone relations and the unilateral contacts at the feet. The friction cones can be approximated with polyhedral convex cones (cf. Section 3.3.2) to reduce the computational burden. Since these constraints are inequality-type, a general solver (i.e. a quadratic program or QP solver) might be preferred in lieu of noniterative optimization. Furthermore, the wrench optimization task is formulated to include other subtasks as well, such as the CoP localization (via the GRM) and/or the contact wrench minimization. As a result of the wrench optimization task, the difference $\|\mathcal{F}_C^{ref} - \mathbb{C}_c\overline{\mathcal{F}}_{opt}^c\|_W$ will be minimized. It is important to note that the outcome of the optimization task depends very much on the reference CRB wrench, $\mathcal{F}_C^{ref}$. The ideal case is when $\mathbb{C}_c\overline{\mathcal{F}}_{opt}^c = \mathcal{F}_C^{ref}$. This implies that the reference CRB wrench lies within the CRB-wrench cone, i.e. $\mathcal{F}_C^{ref}$ is a *contact-consistent* wrench (cf. Section 3.76); then we have

$$\dot{\tilde{\mathcal{L}}}_C^{ref} + \mathcal{G}_C = \mathcal{F}_C^{ref} \in \mathcal{F}_{BWC} = \{\mathbb{C}_{cC}, BWC\}.$$

Once the optimal contact wrenches $\overline{\mathcal{F}}^c_{opt}$ are found, the *forward spatial dynamics (FSD)* problem can be solved. The generalized acceleration $\ddot{\boldsymbol{q}}_{opt} = (\dot{\mathcal{V}}_M^{opt}, \ddot{\boldsymbol{\theta}}^{opt})$ is obtained from the spatial dynamics equation, i.e.

$$\mathbb{M}_C \dot{\mathcal{V}}_M^{opt} + \boldsymbol{H}_{CM}\ddot{\boldsymbol{\theta}}^{opt} + \mathcal{C}_M + \mathcal{G}_C = \mathbb{C}_{cC}\overline{\mathcal{F}}^c_{opt}. \tag{5.109}$$

Note that the FSD problem is also underdetermined. Thus, (differential) *motion optimization* has to be involved. The conventional approach is to introduce a subtask prioritization for the motion control tasks along the constrained and unconstrained motion directions, as well as for the limb and whole-body self-motion control (cf. Section 2.8.2). As a result of the optimization, a *constraint-consistent* $\ddot{\boldsymbol{q}}_{opt}$ is obtained.

The two-step optimization process just described represents a *sequential*-type optimization. This approach has been used in [87,88,167]. It is also possible to combine the generalized acceleration and force optimization steps into a single optimization task. This combination introduces, however, some computational overhead, as will be explained in Section 5.11.1. The formulation of two smaller optimization problems is preferable in lieu of a single but large one [40].

As already noted, in the case of a position-controlled humanoid robot, there is no need to calculate the joint torque; $\ddot{\boldsymbol{\theta}}^{opt}$ and its time integrals can be applied as a control input. In the case of a torque-controlled robot, on the other hand, the *joint-space inverse dynamics (JSID) problem* needs to be solved. Note that once the optimal solutions for the WD and FSD problems have been obtained, the joint torque can be calculated in a *unique* way from the lower part of (4.157). We have

$$\boldsymbol{\tau} = \boldsymbol{M}_{\theta M}\ddot{\boldsymbol{\theta}}^{opt} + \boldsymbol{H}_{CM}^T\dot{\mathcal{V}}_M^{opt} + \boldsymbol{c}_{\theta M} - \mathcal{J}_{cM}^T\overline{\mathcal{F}}^c_{opt}. \tag{5.110}$$

Note that the last term on the r.h.s. implicitely compensates the action of the gravity force. As noted in Section 3.6.4, this gravity compensation approach introduces errors; to avoid them the equivalent expression (3.74) should be preferably used. The last equation assumes then the form:

$$\boldsymbol{\tau} = \boldsymbol{M}_{\theta M}\ddot{\boldsymbol{\theta}}^{opt} + \boldsymbol{H}_{CM}^T\dot{\mathcal{V}}_M^{opt} + \boldsymbol{c}_{\theta M} + \boldsymbol{g}_{\theta} - \mathcal{J}_{cB}^T\overline{\mathcal{F}}^c_{opt}. \tag{5.111}$$

The optimization tasks can be structured as outlined in Fig. 5.22. There are two basic structures based on the FSD (upper two schemes) and JSID (lower two schemes) optimization. Each of these structures can be realized in a sequential/nonsequential form. Nonsequential realizations usually involve computational overhead. Therefore, when aiming at real-time control, the sequential schemes should be preferably used, provided contact-consistent inputs are available.

5.9.2 Optimization Task Formulation and Constraints

From the generic task-space controller structure introduced in Section 5.9.1 it became apparent that constrained optimization tasks have to be solved over the motion $\mathcal{M}$ and force $\mathcal{F}$ domains. When organizing the optimization process, three possibilities should be considered:

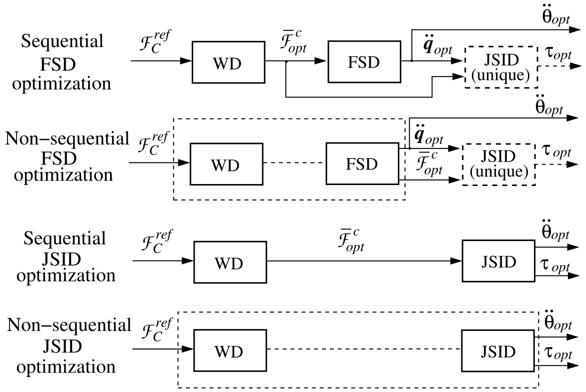


FIGURE 5.22 Optimization task structures for task-space control.

- a noniterative approach based on generalized inverses,
- an iterative approach based on a general solver, and
- a mixed, iterative/noniterative approach.

These possibilities will be explored in the following sections.

Furthermore, the optimization process is based on subsets of equality/inequality-type of constraints defined within the motion and force domains. The most frequently used sets of equality- and inequality-type constraints are presented in Table 5.1 and Table 5.2, respectively. Inequality-type constraints can be embedded into iterative optimization schemes in a straightforward manner. It is also possible to include such type of constraints in noniterative optimization task formulations [118].

One of the main concerns when formulating an optimization task is the lack of solution due to multiple contradicting constraints. This problem can be alleviated by relaxing one or more of the equality-type constraints, $\boldsymbol{Ax} - \boldsymbol{a} = \boldsymbol{0}$. This is done by reformulating the constraint as a quadratic objective function of the type

$$c_l = \|\boldsymbol{Ax} - \boldsymbol{a}\|^2_{W_l}.$$

Here $\boldsymbol{W}_l$ denotes a weight matrix. This approach will be henceforth referred to as the *constraint relaxation*. The subscripts appearing in Table 5.1 will be used to denote the penalty terms derived via constraint relaxation. For example, relaxing the joint acceleration tracking constraint, $\mathcal{M}_{JA}$, yields the penalty term

$$c_{JA} = \|\ddot{\boldsymbol{\theta}} - \ddot{\boldsymbol{\theta}}^{ref}\|^2_{W_{JA}}$$

in the objective function. It should be noted that inequality-type constraints can also be relaxed, via slack variables (cf. Section 2.60). Further details will be given in Section 5.14. Note that strict, equality-type constraints are sometimes referred to as the *hard* constraints. Penalty-type constraints $c_{(\circ)}$, on the other hand, are known as *soft* constraints [25]. The optimization

TABLE 5.1 Equality-type constraints

Name	Notation	Equation
Equation of motion	$\mathcal{F}_{EoM}$	(4.162)
Closed-chain constraint	$\mathcal{M}_{CC}$	(2.105)
Spatial dynamics tracking	$\mathcal{F}_{SD}$	$\mathbb{C}_c\overline{\mathcal{F}}^c = \mathcal{F}_C^{ref}$
Spatial momentum rate tracking	$\mathcal{F}_{SM}$	$\dot{\mathcal{L}}_C = \dot{\mathcal{L}}_C^{ref}$
Angular momentum minimization	$\mathcal{F}_{AM}$	$\boldsymbol{l}_C = \boldsymbol{0}$
CoP tracking	$\mathcal{F}_{CoP}$	(5.179)
End-link motion tracking	$\mathcal{M}_{EL}$	$\boldsymbol{\mathcal{J}}_m\ddot{\boldsymbol{\theta}} = (\dot{\overline{\mathcal{V}}}^m)^{ref} - \dot{\boldsymbol{\mathcal{J}}}_m\dot{\boldsymbol{\theta}}$
Contact wrench tracking	$\mathcal{F}_{CW}$	$\overline{\mathcal{F}}^c = (\overline{\mathcal{F}}^c)^{ref}$
Joint torque tracking	$\mathcal{F}_{JT}$	$\boldsymbol{\tau} = \boldsymbol{\tau}^{ref}$
Joint acceleration tracking	$\mathcal{M}_{JA}$	$\ddot{\boldsymbol{\theta}} = \ddot{\boldsymbol{\theta}}^{ref}$
Posture tracking	$\mathcal{M}_{PT}$	$\boldsymbol{\theta} = \boldsymbol{\theta}^{ref}$

TABLE 5.2 Inequality-type constraints

Name	Notation	Relation
Point-contact friction cone	$\mathcal{F}_{FC}$	(3.9)
Plane-contact torsional (yaw) friction	$\mathcal{F}_{FT}$	as in (3.11)
CRB wrench cone	$\mathcal{F}_{BWC}$	(3.77)
CoP-in-BoS	$\mathcal{F}_{BoS}$	(4.32), rewritten for each contact separately
Torque limits	$\mathcal{F}_{TL}$	$\tau_i^{min} \leq \tau_i \leq \tau_i^{max}$
Joint acceleration limits	$\mathcal{M}_{AL}$	$\ddot{\theta}_i^{min} \leq \ddot{\theta}_i \leq \ddot{\theta}_i^{max}$
Joint velocity limits	$\mathcal{M}_{VL}$	$\dot{\theta}_i^{min} \leq \dot{\theta}_i \leq \dot{\theta}_i^{max}$
Joint angle limits	$\mathcal{M}_{\theta L}$	$\theta_i^{min} \leq \theta_i \leq \theta_i^{max}$
Collision avoidance	$\mathcal{M}_{Col}$	

task can be written as

$$\min_{\boldsymbol{x}\in S} \sum_l c_l, \tag{5.112}$$

where S is a set of (usually convex) hard constraints. Note that the relaxation process is of relative character. When multiple constraints are relaxed, it might be difficult to adjust the numerous weights while aiming at a broad spectrum of tasks [18]. To obtain a realistic, "naturally looking" whole-body motion that avoids excessive postural variations, the weights should be determined in a way consistent with the given tasks [34]. This problem is still unsolved, though.

The form of the solution vector $\boldsymbol{x}$ depends on the optimization subtask; we have

$$\boldsymbol{x} = \begin{cases} \begin{bmatrix} (\overline{\mathcal{F}}^c)^T \end{bmatrix}^T & \text{contact wrench optimization,} \\ \begin{bmatrix} (\overline{\mathcal{F}}^c)^T & \ddot{\boldsymbol{q}}^T \end{bmatrix}^T & \text{spatial dynamics–based optimization,} \\ \begin{bmatrix} (\overline{\mathcal{F}}^c)^T & \ddot{\boldsymbol{q}}^T & \boldsymbol{\tau}^T \end{bmatrix}^T & \text{complete dynamics optimization.} \end{cases} \tag{5.113}$$

The optimization task formulations outlined here can be used in noniterative as well as in multistep (iterative) optimization schemes. From the viewpoint of computational cost, a noniterative, generalized inverse–based optimization is preferable. The problem is that inequality-type constraints cannot be handled directly in this case. Nevertheless, as already noted, it is possible to employ an indirect approach based on appropriately defined quadratic objectives (i.e. a penalty-type approach) [118].

5.10 NONITERATIVE BODY WRENCH DISTRIBUTION METHODS

To obtain the noniterative solution to the wrench distribution problem, solve (5.109) for the contact wrenches as follows:

$$\overline{\mathcal{F}}^{c}_{opt} = \mathbb{C}^{-W}_{cC}(\dot{\tilde{\mathcal{L}}}^{ref}_{C} + \mathcal{G}_{C}) + \boldsymbol{N}(\mathbb{C}_{cC})\overline{\mathcal{F}}_{a}^{c}. \tag{5.114}$$

This solution is of the same form as (3.62). From the discussion in Section 3.5.2, recall that the null-space term (the second term on the r.h.s.) can be parametrized by the internal wrench. The role of this term is to generate *quasistatic* tangential contact wrench components. These components can be used to enforce the friction constraints. Furthermore, the body wrench ($\mathcal{F}^{ref}_{C} = \dot{\tilde{\mathcal{L}}}^{ref}_{C} + \mathcal{G}_{C}$) is distributed via the particular solution term (the first term on the r.h.s.). Note that a weighted generalized inverse of the contact map is employed. With the trivial choice, $\boldsymbol{W} = \boldsymbol{E}$, the body wrench will be distributed via the pseudoinverse. Such distribution, however, is problematic (cf. Section 3.5.4). The problem becomes apparent if one considers the simple proactive task of maintaining a static posture on even floor. Since there is no motion in the joints, $\dot{\tilde{\mathcal{L}}}^{ref}_{C}$ can be assumed zero.[8] Thus, only the gravity term $\mathcal{G}_{C}$ will be distributed. With a symmetric posture, the net CoP will be at the center of the *net BoS*; $\mathcal{G}_{C}$ is distributed equally over the feet with the pseudoinverse. This implies consistency with the statics. Next, consider an asymmetric posture s.t. the net CoP is located within the left or right *foot BoS*. The respective foot will be referred to as "loaded." The normal reaction force at the loaded foot compensates the impressed gravity force. The normal reaction at the unloaded foot, on the other hand, is almost zero. This implies a highly asymmetric distribution. Note that in this case, the pseudoinverse solution will be *inconsistent* with the static state. From this observation it becomes apparent that the pseudoinverse-based distribution is inappropriate for tasks that involve foot loading/unloading, e.g. while walking or stepping in response to a disturbance.

5.10.1 Pseudoinverse-Based Body-Wrench Distribution

On the other hand, a pseudoinverse-based distribution can be quite useful in adaptation to unknown terrain. Consider a general contact model with multiple (user-specified) point

[8] In reality, there will always be small errors stemming from flexibilities in the joints, the environment, etc., which means that $\dot{\tilde{\mathcal{L}}}^{ref}_{C}$ cannot be completely ignored.

contacts at each end link [63]. Note that with such a model, the contacts do not have to be coplanar. The *net CoP* can be written as

$$x_P = \frac{\sum_{i=1}^{\kappa} x_j f_{zi}}{\sum_{i=1}^{\kappa} f_{zi}}, \quad y_P = \frac{\sum_{i=1}^{\kappa} y_j f_{zi}}{\sum_{i=1}^{\kappa} f_{zi}}, \quad \boldsymbol{f}_C = \sum_{i=1}^{\kappa} \boldsymbol{f}_i,$$

where $\boldsymbol{r}_i, \ \boldsymbol{f}_i, \ i \in \{\overline{1, \kappa}\}$ denote the position and reaction force at the contact points. These relations can be rewritten in the following form:

$$\begin{bmatrix} x_P \\ y_P \\ 1 \end{bmatrix} f_{Cz} = \begin{bmatrix} x_1 & x_2 & \dots & x_\kappa \\ y_1 & y_2 & \dots & y_\kappa \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} f_{z1} \\ f_{z2} \\ \dots \\ f_{z\kappa} \end{bmatrix},$$

$$\begin{bmatrix} x_P \\ 1 \end{bmatrix} f_{Cx} = \begin{bmatrix} x_1 & x_2 & \dots & x_\kappa \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} f_{x1} \\ f_{x2} \\ \dots \\ f_{x\kappa} \end{bmatrix},$$

$$\begin{bmatrix} y_P \\ 1 \end{bmatrix} f_{Cy} = \begin{bmatrix} y_1 & y_2 & \dots & y_\kappa \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} f_{y1} \\ f_{y2} \\ \dots \\ f_{y\kappa} \end{bmatrix}.$$

Given x_P^{des}, y_P^{des}, and $\boldsymbol{f}_C^{des}$, it is straightforward to solve the above matrix-vector equations for $\boldsymbol{f}_i^{des}$, by employing the pseudoinverse. Because of the minimum-norm solution, the CRB wrench will be distributed evenly over the contact points. If contacts are lost, e.g. by stepping over an edge, the even distribution yields foot rotation until another contact is made. In this way, the system can adapt to an unknown terrain, without force measurements.

Furthermore, the minimum-norm solutions in the tangential $(x, \ y)$ directions can be modified by adding a null-space term for internal force control. It then becomes possible to address the friction problem and to avoid a slip. The internal force can be derived, for example, from an impedance control law, as suggested in [62].

This method has been used in reactive balance control with a torque-controlled robot and shown a satisfactory response to a relatively weak unknown disturbance [63,62]. On the other hand, this method is not suitable for reactive step-type balance control under a strong disturbance, since the feet cannot be loaded/unloaded with the pseudoinverse-based distribution, as already noted. The method cannot be used in gait control either, due to the same reason.

5.10.2 The ZMP Distributor

Aiming at walking, the ZMP distributor approach attempts to distribute the body wrench in a way consistent with the static states [71]. This implies asymmetric distribution. When the net CoP is within the BoS of Foot $j, \ j \in \{r, l\}$, that foot is fully loaded (which implies that the other foot will be fully unloaded). This means that the gravity force term $\boldsymbol{f}_G$ will be distributed in a *discrete manner*, i.e. without accounting for the distance between the CoP and the foot BoS boundary. The contact force components thus obtained are then used to calculate the desired ankle torques (or contact moment components), from

$$\bar{\boldsymbol{m}}_j^c = (\boldsymbol{r}_{A_j} - \boldsymbol{r}_{P_j}^{des}) \times \bar{\boldsymbol{f}}_j^c. \tag{5.115}$$

Here $\bar{\boldsymbol{f}}_j^c$ and $\bar{\boldsymbol{m}}_j^c$ are the components of the contact wrench and $\boldsymbol{r}_{P_j}^{des}$ and $\boldsymbol{r}_{A_j}$ are the positions of the CoP and the ankle, respectively. On the other hand, when the net CoP is in between the feet, a *force distribution ratio* is determined *heuristically* and the net force is distributed accordingly. The desired contact force components are then used to obtain the desired net ankle torque. This torque is also distributed via the heuristics.

It should be noted, though, that because of the discretization and the heuristics, the solution cannot be fully consistent with the static states. Note also that the desired CoP is needed for the distribution. Although the method has shown satisfactory performance during experiments with a real robot (HRP-4C) [71], better solutions that avoid the abovementioned shortcomings do exist, as will be shown in Sections 5.10.3 and 5.10.4.

5.10.3 Proportional Distribution Approach

A proportional body wrench distribution approach can be devised based on the observation that the vertical GRF appears in the friction constraints (friction cone and torsional friction) as well as in the CoP-in-BoS constraint [89]. A force distribution ratio can be determined from the vertical GRF component as

$$\eta = f_{rz}/f_{Gz},$$

where $f_{Gz} = f_{rz} + f_{lz}$ is the reaction stemming from the gravity force. Note that $0 \leq \eta \leq 1$. This ratio is then used to calculate all GRF components from

$$f_{ri} = \eta f_{Gi}, \quad f_{li} = (1 - \eta) f_{Gi}, \tag{5.116}$$

where $i \in \{x, y, z\}$. The total reaction acting at the CoM (the CRB force) can then be distributed in a *continuous way*, by varying η appropriately. For example, with $\eta = 0.5$, a symmetric distribution is obtained. With $\eta = 0$ or $\eta = 1$, on the other hand, a highly asymmetric distribution is obtained. In this way, consistency with the static states can be ensured. Note also that with the above ratio, the friction cone constraints will always be satisfied.

Furthermore, η could also be used to distribute the CRB moment in a similar fashion, i.e.

$$m_{ri} = \eta(m_{ri} + m_{li}), \quad m_{li} = (1 - \eta)(m_{ri} + m_{li}).$$

There is no guarantee, however, that the CoP-in-BoS and torsional friction constraints will be satisfied, since the three GRM components depend on the GRF. In [89], analytically derived bounds for η were suggested to alleviate the problem.

The proportional distribution approach avoids the disadvantage of the discrete wrench distribution of the ZMP distributor approach. An implementation example, to be introduced in Section 5.12.2, will provide further insight.

5.10.4 The DCM Generalized Inverse

A noniterative wrench distribution approach that can distribute the body wrench dynamically at multicontact postures is based on the so-called DCM generalized inverse (DCM-GI) [58]. The main idea is to design the weight matrix in (5.114) as a function of the DCM. The design should also avoid the discrete distribution disadvantage of the ZMP distributor, allowing for complete loading/unloading of the links in contact. In this way, undesired foot/hand roll during contact transitions can be avoided. The distribution approach should also ensure *consistency with statics*, taking into account the friction cone, torsional friction and CoP-in-BoS constraints, in a similar fashion as the proportional distribution approach from Section 5.10.3, but with the added capability of *multicontact transitions*.

Assuming κ robot links are in contact with the environment, the body wrench can be expressed as (cf. (3.57))

$$\mathcal{F}_C = \begin{bmatrix} \boldsymbol{f}_C \\ \boldsymbol{m}_C \end{bmatrix} = \sum_{k=1}^{\kappa} \left(\begin{bmatrix} \boldsymbol{E}_3 & \boldsymbol{0}_3 \\ -[\boldsymbol{r}^{\times}_{\overleftarrow{Ck}}] & \boldsymbol{E}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_k \\ \boldsymbol{m}_k \end{bmatrix} \right). \tag{5.117}$$

Here $\boldsymbol{r}_{\overleftarrow{Ck}}$ is the position vector pointing from the kth local coordinate frame to the centroidal coordinate frame. The local coordinate frame is attached to the projection of the ankle/wrist joint on the contact surface. It is apparent that the body-GRM $\boldsymbol{m}_C$ depends on the foot/hand reaction wrenches. This dependence should be eliminated before deciding the weight coefficients for the distribution. This can be done by subtracting the GRF-induced GRM from the body wrench (cf. (3.58)), as follows:

$$\mathcal{F}_{net} \equiv \mathcal{F}_C - \sum_{k=1}^{\kappa} \begin{bmatrix} \boldsymbol{0}_3 \\ [\boldsymbol{r}^{\times}_{\overleftarrow{Ck}}]\boldsymbol{f}_k \end{bmatrix}. \tag{5.118}$$

From Section 3.6.1 recall that this operation ensures that $\mathcal{F}_{net}$ can be represented as a sum of contact wrenches, i.e. $\mathcal{F}_{net} = \sum_{k=1}^{\kappa} \mathcal{F}_k$. Here $\mathcal{F}_{net}$ is to be distributed with the help of weights appearing on the diagonal of the following weighting matrix:

$$\boldsymbol{W} = \text{diag}(\boldsymbol{W}_1 \;\; \boldsymbol{W}_2 \;\; \dots \;\; \boldsymbol{W}_\kappa) \in \Re^{c \times c},$$

where $\boldsymbol{W}_k \in \Re^{c_k \times c_k}$, c_k denoting the number of contact constraints at Contact k. When $c_k = 6$, we have

$$\boldsymbol{W}_k \equiv \text{diag}\left(\boldsymbol{w}^f_k \;\; \boldsymbol{w}^m_k \right). \tag{5.119}$$

Here $\boldsymbol{w}^f_k = \begin{bmatrix} w^f_{k_x} & w^f_{k_y} & w^f_{k_z} \end{bmatrix}^T$ and $\boldsymbol{w}^m_k = \begin{bmatrix} w^m_{k_x} & w^m_{k_y} & w^m_{k_z} \end{bmatrix}^T$ are vectors composed of positive weights for the distribution of the force and moment components of the net wrench. Since the weighing matrix is diagonal, the weights will determine the distribution ratios for the force and moment components as

$$w^f_{k_i}/w^f_{\bar{k}_i} = f_{\bar{k}_i}/f_{k_i}, \tag{5.120}$$

$$w^m_{k_i}/w^m_{\bar{k}_i} = m_{\bar{k}_i}/m_{k_i},$$

respectively, $i \in \{x, y, z\}$. Index $\bar{k}$ stands for "not k," that is, if $k \in \{\overline{1, \kappa}\}$ then $\bar{k} \in \{\overline{1, \kappa}\}/k$. From the above relation, it is apparent that the net wrench is distributed componentwise, i.e.

$$f_{k_i} = \frac{\tilde{w}^f_i}{w^f_{k_i}} f_{net_i}, \quad m_{k_i} = \frac{\tilde{w}^m_i}{w^m_{k_i}} m_{net_i}, \tag{5.121}$$

f_{net_i} and m_{net_i} denoting the scalar force and moment components, respectively, of the net wrench. The tilde notation over the weigh coefficients are introduced to account for the case of multiple contacts, i.e.

$$\tilde{w}^f_i = \frac{P^f_i}{\sum_{k=1}^{\kappa} \frac{P^f_i}{w^f_k}}, \quad P^f_i = \prod_{k=1}^{\kappa} w^f_{k_i}, \quad \tilde{w}^m_i = \frac{P^m_i}{\sum_{k=1}^{\kappa} \frac{P^m_i}{w^m_k}}, \quad P^m_i = \prod_{k=1}^{\kappa} w^m_{k_i}.$$

In what follows, simple examples will be presented for better understanding of formula (5.121). For clarity but without loss of generality, the DCM-GI-based approach will be applied first to a double-stance (DS) posture, s.t. $c = 12$. The envisioned DS to single-stance (SS) as well as SS-to-DS transitions can be achieved with the design based on the following three policies.

Vertical GRF Force Distribution Policy

Denote the weighting matrix for double stance as

$$\boldsymbol{W}_X = \mathrm{diag}(\boldsymbol{W}_R \quad \boldsymbol{W}_L) \in \Re^{12 \times 12}.$$

The net contact wrench is $\mathcal{F}_F = \mathcal{F}_R + \mathcal{F}_L$,[9] in accordance with (5.118). Referring to (5.120), the following ratio will be used to realize the approach proposed herein:

$$f_{R_z}/f_{L_z} = \frac{||\boldsymbol{r}_P - (\boldsymbol{r}_L + \boldsymbol{r}_{\overleftarrow{P_L L}})||}{||\boldsymbol{r}_P - (\boldsymbol{r}_R + \boldsymbol{r}_{\overleftarrow{P_R R}})||}. \tag{5.122}$$

Here $\boldsymbol{r}_P \in \Re^3$ denotes the net CoP, $\boldsymbol{r}_k$ is the position of the ankle center k, and $\boldsymbol{r}_{\overleftarrow{P_k k}}$ is the position vector pointing from $\boldsymbol{r}_k$ to the local CoP $\boldsymbol{r}_{P_k}$. The vertical reaction component at each foot contact is determined by the following inner product:

$$w^f_{k_z} = s_k \left[\Delta r_x \quad \Delta r_y \quad 0\right] \left(\boldsymbol{r}_X - \boldsymbol{r}^{ds}_k\right), \tag{5.123}$$

where $\boldsymbol{r}^{ds}_k = \boldsymbol{r}_k + \Delta \boldsymbol{r}_k$. As shown in Fig. 5.23, the horizontal components of the $\Delta \boldsymbol{r}_k$ vectors specify the DS area relative to the local contact frames, fixed at the ground projections $\boldsymbol{r}_k$ of

[9] This is also true for the force/moment component vectors, i.e. $\boldsymbol{f}_F = \boldsymbol{f}_R + \boldsymbol{f}_L$, $\boldsymbol{m}_F = \boldsymbol{m}_R + \boldsymbol{m}_L$.

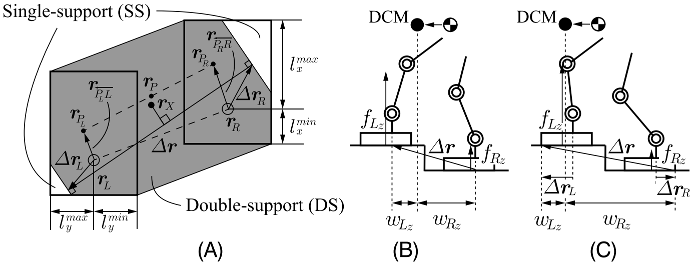


FIGURE 5.23 Geometrical relations pertinent to the wrench distribution policy. (A) The double-support area is specified either by the user or the system, via $\Delta\boldsymbol{r}_k$. (B) With $\Delta\boldsymbol{r}_k$ set at zero, the vertical reactions pass through the ankles. (C) With $\Delta\boldsymbol{r}_k$ set at the outer vertices, the entire BoS can be used as a double-support area.

the ankle centers. The $\Delta\boldsymbol{r}_k$ vectors are geometrically constrained within the foot BoS; we have

$$l_h^{min} \leq \Delta r_{k_h} \leq l_h^{max}, \ h \in \{x, y\}. \tag{5.124}$$

As apparent from (5.123), the vertical component of vector $\Delta\boldsymbol{r} = \boldsymbol{r}_L^{ds} - \boldsymbol{r}_R^{ds}$ is set at zero. This renders the wrench distribution problem a planar one. The terms $s_R = 1$, $s_L = -1$ ensure the positive definiteness of the vertical reactions, in agreement with the unilateral contact condition at the feet. Note that the distribution weight (5.123) depends on the distance between the DCM and the dynamic stability area. Note also that when the difference $\boldsymbol{r}_X - \boldsymbol{r}_k^{ds} = \boldsymbol{0}$, the weighting matrix becomes singular; this case should be avoided.

Consider first the static case, $\dot{\boldsymbol{r}}_C = \boldsymbol{0}$ and $\boldsymbol{r}_X = \boldsymbol{r}_C$. When the CoM is within the DS area, $w_{k_z}^f < 1$. At DS-SS boundary k ($\boldsymbol{r}_C = \boldsymbol{r}_k^{ds}$), $w_{k_z}^f = 1$ and $w_{\bar{k}_z}^f = 0$. The vertical force distribution policy can then be characterized as continuous and consistent with the statics. For example, when $\Delta\boldsymbol{r}_k = \boldsymbol{0}$, as in Fig. 5.23B, minimal load in the ankle joints will be obtained with a static stance s.t. the CoM is over ankle k. The reaction force at that ankle becomes then a *structural force* (i.e. the reaction does not generate a moment). At ankle $\bar{k}$ there will be no reaction at all. Humans sometimes use such a load/unload static posture to relax. In another example, shown in Fig. 5.23C, $\Delta\boldsymbol{r}_k$ is set at the outer vertices of the BoS. In this way, the entire BoS constitutes the area of dynamic stability. This also implies that no transition is meant to occur, i.e. only a swaying task is envisioned.

In addition to statics, the above vertical GRF distribution policy is also consistent with the dynamic states, thanks to the use of the DCM. This will be demonstrated with dynamic contact transitions in an example below.

Friction Policy

The net wrench distribution relation (5.121) can be used to express the friction cone constraint (3.9) as follows:

$$\frac{\sqrt{\left((\tilde{w}_x^f/w_{k_x}^f)f_{net_x}\right)^2 + \left((\tilde{w}_y^f/w_{k_y}^f)f_{net_y}\right)^2}}{(\tilde{w}_z^f/w_{k_z}^f)f_{net_z}} \leq \mu_k.$$

This is an inequality constraint that cannot be adopted directly. To minimize the effect of friction, make use of the following equality-based friction cone policy. From the above relation,

$$\frac{\sqrt{\left((\tilde{w}_x^f/w_{k_x}^f)f_{net_x}\right)^2+\left((\tilde{w}_y^f/w_{k_y}^f)f_{net_y}\right)^2}}{(\tilde{w}_z^f/w_{k_z}^f)f_{net_z}}=\frac{\sqrt{f_{net_x}^2+f_{net_y}^2}}{f_{net_z}}$$

or

$$\left((\tilde{w}_x^f/w_{k_x}^f)f_{net_x}\right)^2+\left((\tilde{w}_y^f/w_{k_y}^f)f_{net_y}\right)^2=(\tilde{w}_z^f/w_{k_z}^f)^2(f_{net_x}^2+f_{net_y}^2).$$

Now, assume that $w_{k_x}^f=w_{k_y}^f\equiv w_{k_h}^f,\ h\in\{x,y\}$. Then

$$(\tilde{w}_h^f/w_{k_h}^f)^2(f_{net_x}^2+f_{net_y}^2)=(\tilde{w}_z^f/w_{k_z}^f)^2(f_{net_x}^2+f_{net_y})^2, \tag{5.125}$$

where $\tilde{w}_h^f/w_{k_h}^f=\tilde{w}_z^f/w_{k_z}^f$ and $w_{k_h}^f=w_{k_z}^f$.

Next, consider the torsional friction constraint $\mathcal{F}_{FT}$ (cf. Table 5.2 and (3.11)), rewritten with the help of (5.121) as

$$\frac{(\tilde{w}_z^m/w_{k_z}^m)|m_{net_z}|}{(\tilde{w}_z^f/w_{k_z}^f)f_{net_z}}\leq\gamma_k.$$

Based on this inequality, the following torsional friction distribution policy will be adopted:

$$\frac{(\tilde{w}_z^m/w_{k_z}^m)|m_{net_z}|}{(\tilde{w}_z^f/w_{k_z}^f)f_{net_z}}=\frac{|m_{net_z}|}{f_{net_z}},$$

and hence,

$$\tilde{w}_z^n/w_{k_z}^n=\tilde{w}_z^f/w_{k_z}^f.$$

CoP Allocation Policy

The net wrench distribution relation (5.121) can be used to express the foot CoP in the local contact frame as follows:

$$\boldsymbol{r}_{P_k}=\begin{bmatrix}-\left((\tilde{w}_y^m/w_{k_y}^m)m_{net_y}\right)/\left((\tilde{w}_z^f/w_{k_z}^f)f_{net_z}\right)\\ \left((\tilde{w}_x^m/w_{k_x}^m)m_{net_x}\right)/\left((\tilde{w}_z^f/w_{k_z}^f)f_{net_z}\right)\end{bmatrix}.$$

Apparently, the foot CoP location depends on the vertical force distribution. This implies that when the foot is going to be unloaded, the foot CoP will be shifted toward the BoS boundary, eventually resulting in a foot roll. To avoid this problem, the effect of the asymmetric vertical reaction distribution should be compensated for in the CoP equation. This can be done when

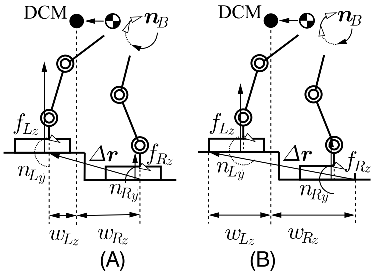


FIGURE 5.24 Relations between vertical foot reactions, tangential reaction moments and the net base moment. (A) A smaller DS area setting yields smaller reaction moments at the feet. (B) A larger DS area setting yields larger reaction moments and increases the possibility of a foot roll.

the following relation holds:

$$\begin{bmatrix} -\left((\tilde{w}_y^m/w_{k_y}^m)m_{net_y}\right)/\left((\tilde{w}_z^f/w_{k_z}^f)f_{net_z}\right) \\ \left((\tilde{w}_x^m/w_{k_x}^m)m_{net_x}\right)/\left((\tilde{w}_z^f/w_{k_z}^f)f_{net_z}\right) \end{bmatrix} = \begin{bmatrix} -m_{net_y}/f_{net_z} \\ m_{net_x}/f_{net_z} \end{bmatrix}.$$

Hence,

$$\tilde{w}_x^m/w_{k_x}^m = \tilde{w}_y^m/w_{k_y}^m = \tilde{w}_z^f/w_{k_z}^f.$$

The CoP allocation problem can be related to the setting of the dynamic stability area, as follows. First, note that from (5.117) (see also Fig. 5.24), it is apparent that when the contribution of the net vertical GRF $\boldsymbol{f}_{net}$ to the body moment $\boldsymbol{m}_B$ is relatively large (filled arrow heads in Fig. 5.24), that of the reaction moments (transparent arrow heads in Fig. 5.24) will be relatively small. Thus, larger force weights will yield a smaller net moment to be distributed, i.e. a smaller net foot GRM $\boldsymbol{m}_{net}$. On the other hand, according to (5.123), the force weights are determined from the dynamic stability area. With a smaller dynamic stability area, larger force weights will be obtained. In this way, the reaction moments will be minimized and the CoPs will be allocated closer to the local frames at the ankle/wrist projections. With a larger dynamic stability area it is more likely that a foot roll will occur since the CoPs will allocate apart from the local frames due to the larger ground reaction moments (see Fig. 5.24B).

Final Result

As a result of the friction cone, torsional friction, and the CoP allocation policies, the weight matrix in the DCM-GI assumes the simple form

$$\boldsymbol{W}_k = (w_{k_z}^f \boldsymbol{E}_6). \tag{5.126}$$

This result emphasizes the important role of an appropriate vertical reaction distribution. It should be noted that the DCM-GI distribution algorithm introduces an error since the net wrench (5.118) that is distributed is different from the real body wrench. But the error is small and only in the force component.

The DCM-GI wrench distribution approach can be extended in a straightforward manner to cope with noncoplanar contact postures [58]. In such a case, internal wrench control is needed. This can be achieved via the null space component in (5.114). The method can also be applied at multicontact (e.g. triple-support) postures. An implementation example for this case will be presented below. Furthermore, the method is quite suitable for reactive balance control in response to unknown external disturbances. An implementation example will be presented in Section 7.7.

Simple Balance Controller With DCM-GI-Based Body Wrench Distribution

The reference contact wrenches obtained under the DCM-GI based body wrench distribution approach can be written as (cf. (5.114))

$$(\overline{\mathcal{F}}^c)^{ref} = \mathbb{C}_{cC}^{-W_X}(\dot{\mathcal{L}}_C^{ref} + \mathcal{G}_C) + \boldsymbol{N}(\mathbb{C}_{cC})\overline{\mathcal{F}}_a^c. \tag{5.127}$$

As already noted, the internal wrench control component (i.e. the null-space term) is needed only in the case of noncoplanar contacts. The linear and angular components of $\dot{\mathcal{L}}_C^{ref}$ can be determined from the asymptotically stable CRB motion tracking controller described in Section 5.8.7.

A relatively simple balance controller for torque-controlled robots can be derived from the JSID solution, (4.244). We have

$$\boldsymbol{\tau} = \boldsymbol{g}_\theta - \mathcal{J}_{cB}^T(\overline{\mathcal{F}}^c)^{ref} - \boldsymbol{N}^*(\mathcal{J}_{cM})\boldsymbol{M}_{\theta_M}\boldsymbol{K}_{\theta_D}\dot{\boldsymbol{\theta}}. \tag{5.128}$$

The last term stands for joint damping that is used to suppress the motion in the arms. This term is derived from the null space of the mixed quasivelocity constraint Jacobian to ensure that there will be no interference with the other two terms. As clarified at the end of Section 3.6.4, the distributed gravity compensation with $\boldsymbol{g}_\theta$, in combination with the base quasivelocity constraint Jacobian map $\mathcal{J}_{cB}^T$, yields better results than the lumped gravity compensation with $\mathcal{J}_{cM}^T\overline{\mathcal{F}}^c(\mathcal{G}_C)$. The interference between the gravity compensation and the $(\overline{\mathcal{F}}^c)^{ref}$ term can then be avoided. The control block diagram is shown in Fig. 5.25. This controller can be used for balance control in single or double stance and during the respective transitions. An implementation with a proactive balance control task will be given in Section 5.12.2. Another implementation with a reactive task can be found in Section 7.7.

Implementation Example

The following example demonstrates the DCM-GI body wrench distribution control capability in a proactive task with multicontact transitions. The reference rates of change of the linear and angular momentum can be determined from (5.107) and (5.106), respectively. In this particular example, a virtual viscous damper for the angular motion is used, i.e. $\dot{\boldsymbol{l}}_B^{ref} = -D_\omega \boldsymbol{\omega}_B$, D_ω denoting the damping gain.

The performance of the DCM-GI balance controller is examined via simulations in OpenHRP3 (cf. Chapter 8). A small-size humanoid robot model with parameters similar to those of the HOAP-2 robot [26] is used. For the numbering of the joints and other relevant data, see Section A.1. The footprint (BoS) is $(l_x^{max}, l_x^{min}, l_y^{max}, l_y^{min}) = (58, -40, 31.5, -31.5)$ mm w.r.t. the ankle joint center. The gains in the spatial momentum rate stabilizer are set at

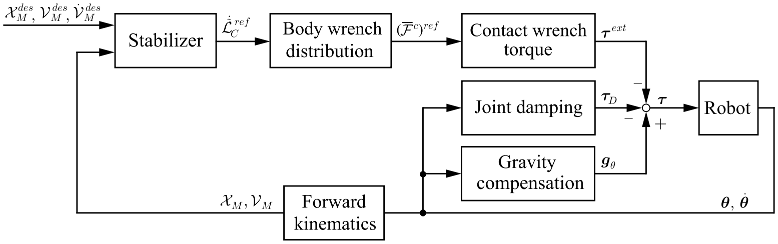


FIGURE 5.25 Control block-diagram for balance control of a torque-controlled robot in single/double stance. The controller ensures asymptotic trajectory tracking of feasible desired CRB trajectories and a DCM-GI based body-wrench distribution approach.

$k_X = 300$ and $D_\omega = 50$. The torque-based whole-body controller described in Section 5.12.1 is used. The motion of the feet and hand end links in the unconstrained motion directions is controlled as a lower-priority subtask, via a joint torque component obtained from a conventional kinematic PD feedback plus feedforward control law (cf. Fig. 5.30).

Initially, the robot is in a triple-support (TS) posture: double-support (DS) stance with plane contacts at the feet plus right hand in point contact. Since the foot contacts are coplanar, no internal wrench control is needed. Thus, the null-space term in (5.114) is set at zero. The boundaries of the dynamic stability area are determined by the locations of the two ankle joints and the hand contact, i.e. set $\Delta \boldsymbol{r}_k = \boldsymbol{0}$, $k = 1, 2, 3$. The proactive balance control task is defined as a DCM tracking task to obtain two consecutive contact transitions: first lifting the left foot (TS-to-DS) and then lifting the hand (DS-to-SS[10]). The desired DCM values are set at $\boldsymbol{r}_X^{des}(t_0) = (0, 0, 0)$, $\boldsymbol{r}_X^{des}(t_1) = (30, -75, 0)$ mm, $\boldsymbol{r}_X^{des}(t_2) = (25, -40, 0)$ mm, respectively, whereby $t_0 = 3$ s, $t_1 = 7$ s, and $t_2 = 9$ s. These positions are connected via a fifth-order spline.

Snapshots and graphs from the result of the simulation are displayed in Fig. 5.26 and Fig. 5.27, respectively. The result is shown in animated form in Video 5.10-1 [57]. The trajectory of the CoM is apparent from Fig. 5.27A. First, the CoM is shifted toward the right foot and slightly forward, toward the hand contact. The contact force at the left foot decreases thereby gradually (cf. Fig. 5.27B). At $t_1 = 7$ s, the left foot contact force becomes zero. A continuous contact transition is then ensured from TS to DS, resulting in right foot and hand contact only. Such transition is possible because the body wrench distribution method ensures the same distribution ratio. After the transition, the left foot lifts off vertically. At the same time, the CoM is moved above the right-foot ankle joint, whereby the hand contact force decreases gradually (cf. Fig. 5.27C). At $t_2 = 9$ s, the hand contact force becomes zero. A contact transition is then ensured from DS to SS without generating excessive reactions. The contact force at the right foot is shown in Fig. 5.27D.

5.10.5 The VRP Generalized Inverse

As shown in Section 5.10.4, the DCM-GI wrench distribution approach can be used in multicontact transitions. It should be noted, though, that for tasks that require swift changes

[10] SS stands for "single support."

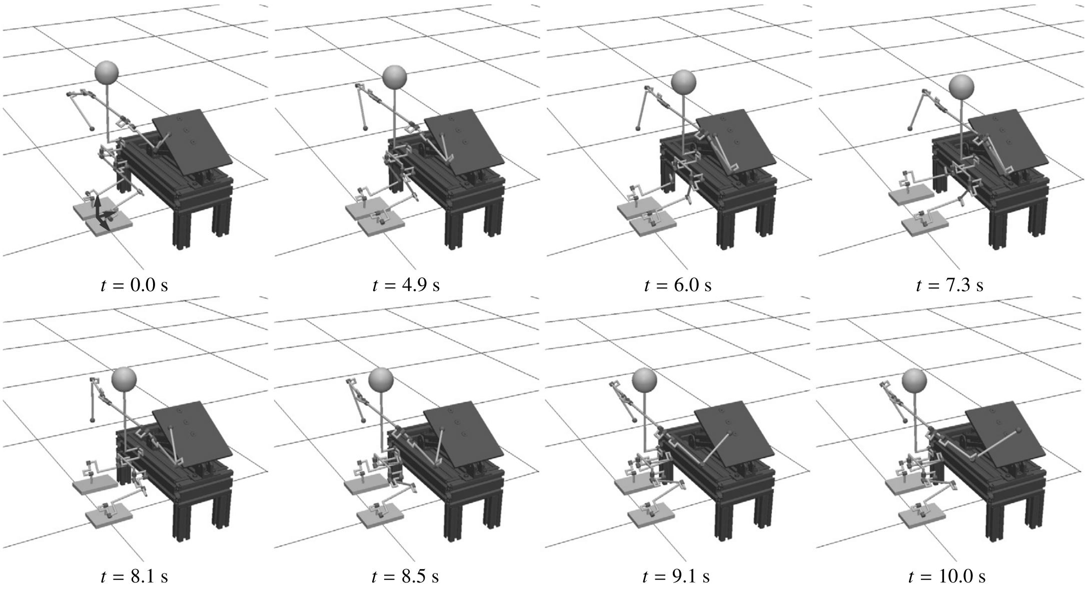


FIGURE 5.26 Multicontact transition task. Snapshots: initially, the robot is in double stance on flat ground, the right hand being in (point) contact with an inclined surface (a triple-support (TS) posture). The body wrench is redistributed via the DCM-GI, as follows. First, the left foot contact is gradually unloaded s.t. the foot can be lifted off ground (at $t = 7.3$ s, a double-support (DS) posture). Then, the hand contact is gradually unloaded s.t. the hand can be lifted off the table (at $t = 10.0$ s, a single-support (SS) posture).

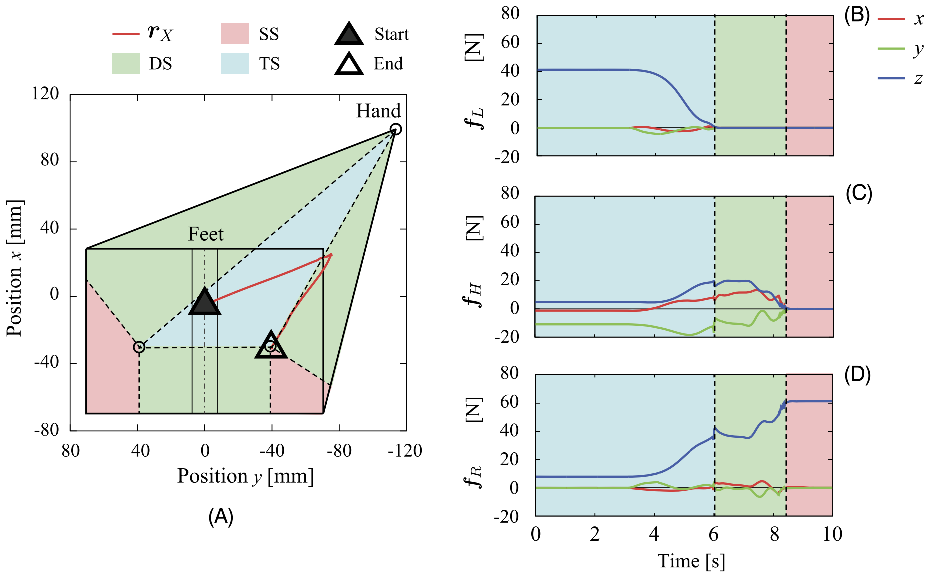


FIGURE 5.27 Multicontact transition task. Graph (A): TS/DS/SS dynamic stability area setting and trajectory of the DCM. The circles within the feet denote the ankle joints. Graphs (B)–(D): GRFs at the contacts.

in the contact states, such as dynamic walking or reactive stepping, the distribution should be based on a second-order formulation rather that on a first-order one, as is the case with the DCM $\boldsymbol{r}_X$. The VRP $\boldsymbol{r}_{vrp}$ represents a second-order formulation that turns out to be quite suitable for such type of distribution.

A wrench distribution based on the VRP can be achieved in a straightforward way within the DCM-GI framework: in (5.123) just substitute $\boldsymbol{r}_X$ with $\boldsymbol{r}_{vrp}$. All other relations derived in Section 5.10.4 remain valid. The weighting matrix obtained in this way will be denoted as $\boldsymbol{W}_{vrp}$. The respective weighted generalized inverse will be referred to as the *VRP generalized inverse* or VRP-GI for short. Note that, since rapid contact transitions are assumed under VRP-GI-based wrench distribution, there is no need to specify a stability area that is different from the minimal one, i.e. $\Delta\boldsymbol{r}_k = \boldsymbol{0}$.

To understand the implications of using the VRP instead of the DCM, consider a *dynamic state* s.t. the robot is in a double stance, the rate of change of the angular momentum is zero, and thus, the net CoP is the vertical projection of the VRP. Furthermore, assume that the local CoP $\boldsymbol{r}_{\overleftarrow{P_k k}}$ is at the ground projection of the respective ankle joint (cf. Fig. 5.23A). This means that the net CoP lies on the line connecting the projections of the ankle joints (i.e. on the line determined by $\Delta\boldsymbol{r}$). The dynamic stability area is set at the minimum, as already noted. At such a state, it is desirable to obtain a wrench distribution that does not generate GRMs.

It can be confirmed that the abovementioned wrench distribution can be achieved with the VRP-GI. On the contrary, this cannot be done with the DCM-GI since the DCM does not lie on the above line; at a dynamic state the ground projection of the DCM (i.e. the xCoM) is always offset from the line (cf. Fig. 5.23A). In general, at states that differ from the above special case, the moments at the ankle joints induced by the VRP-GI distribution will be smaller than those induced by the DCM-GI distribution. This also supports faster contact transitions—a highly desirable property in a dynamic gait task, as already mentioned. An implementation of the VRP-GI in dynamic walking will be discussed in Section 7.4.

5.10.6 Joint Torque–Based Contact Wrench Optimization

From the preceding discussion it became apparent that the body WD problem plays an important role in balance control. The solutions provided demonstrate that the problem is not a trivial one. As an alternative to the distribution of the body wrench one could just minimize the contact wrenches. The respective optimization framework can be based on the inverse solution to the *complete dynamics* (cf. (4.238)); then we have

$$\boldsymbol{\tau} = \boldsymbol{V}^{\dagger}\boldsymbol{V}_r^T(\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{h}) + \left(\boldsymbol{E} - \boldsymbol{V}^{\dagger}\boldsymbol{V}_r^T\boldsymbol{S}^T\right)\boldsymbol{W}^{-1}\boldsymbol{\tau}_a. \tag{5.129}$$

Recall that $\boldsymbol{V}^{\dagger} = \boldsymbol{W}^{-\frac{1}{2}}\left(\boldsymbol{V}_r^T\boldsymbol{S}^T\boldsymbol{W}^{-\frac{1}{2}}\right)^{+} \in \Re^{n\times r}$, $\boldsymbol{V}_r$ providing the minimal null-space basis associated with the underdetermined task; $\boldsymbol{S} \in \Re^{n\times(n+6)}$ is the underactuation filtering matrix s.t. $\mathcal{Q} = \boldsymbol{S}^T\boldsymbol{\tau}$ represents the generalized force vector (cf. (2.102)). Note that since the above solution is derived with the constraint force elimination approach (cf. Section 4.12), the contact constraints are absent. Note also that the spatial dynamics do not appear explicitly. This means that the WD problem cannot be directly addressed. It is possible, though, to optimize the contact wrenches indirectly. Indeed, notice that the solution is parametrized by a symmetric positive definite weight matrix, $\boldsymbol{W}$, and an arbitrary joint torque vector, $\boldsymbol{\tau}_a$. These parameters can be determined via optimization. An appropriate objective function that combines quadratic and linear components can be formulated as follows [126]:

$$\min_{\boldsymbol{x}}\left(c_{JT}+c_{CW}+\boldsymbol{b}_{JT}^T\boldsymbol{\tau}+\boldsymbol{b}_{CW}^T\boldsymbol{\tau}^c\right), \tag{5.130}$$

where $\boldsymbol{x}=\begin{bmatrix}\boldsymbol{\tau}^T & (\overline{\mathcal{F}}^c)^T\end{bmatrix}^T$. The terms c_{JT} and c_{CW} are penalty-type, quadratic form relaxations of the equality constraints $\mathcal{F}_{JT}|_{\boldsymbol{\tau}^{ref}=0}$ and $\mathcal{F}_{CW}|_{(\bar{\mathcal{F}}^c)^{ref}=0}$, respectively; $\boldsymbol{b}_{(\circ)}$ denote arbitrary vectors that parametrize the linear components. Note that the quadratic components $c_{(\circ)}$ are also parametrized, via appropriate weight matrices $\boldsymbol{W}_{(\circ)}$. The terms c_{JT} and c_{CW} minimize the total joint torque and the contact wrenches, respectively. The solution to the above minimization task is obtained as [126]:

$$\boldsymbol{W}=\boldsymbol{W}_{JT}+\boldsymbol{S}\boldsymbol{W}_{CW}\boldsymbol{S}^T, \tag{5.131}$$

$$\boldsymbol{\tau}_a=-\boldsymbol{b}_{JT}+\boldsymbol{S}\boldsymbol{W}_{CW}(\boldsymbol{M}\ddot{\boldsymbol{q}}+\boldsymbol{h})+\boldsymbol{S}\boldsymbol{b}_{CW}. \tag{5.132}$$

Although the contact wrenches do not appear explicitly in (5.129), it is possible to minimize them via the c_{CW} term. Note that with

$$\boldsymbol{W}_{CW}=\boldsymbol{J}_c\boldsymbol{W}_{\kappa}\boldsymbol{J}_c^T, \tag{5.133}$$

the joint torque stemming from the constraints will be minimized, i.e. $c_{CW}=\|\boldsymbol{\tau}^c\|^2_{W_\kappa}$. This means that *the contact wrench minimization problem can be reformulated as a joint torque minimization problem.* In the above formula, $\boldsymbol{J}_c$ stands for the system constraint Jacobian.

There are a few possibilities to tackle the (implicit) contact wrench optimization problem. The simplest approach is just to minimize the quadratic cost for the contact wrenches, with a suitably defined $\boldsymbol{W}_{CW}$. To this end, set $\boldsymbol{W}_{JT}=\boldsymbol{0}$ and $\boldsymbol{b}_{JT}=\boldsymbol{0}=\boldsymbol{b}_{CW}$ in (5.131) and (5.132). Note that with $\boldsymbol{W}_{CW}$ as in (5.133) and $\boldsymbol{W}_\kappa=\boldsymbol{M}^{-1}$, the c_{CW} minimization subtask will yield a dynamically consistent (in the sense of Gauss' principle of least constraint) solution. Next, to tackle the friction constraints, one could attempt to minimize the tangent reaction force and the reaction moment components. This can be done with the help of the following weight matrix [126]:

$$\boldsymbol{W}_{CW}=\begin{bmatrix}\mathbb{R}_{F_r}^T\boldsymbol{W}_{F_r}\mathbb{R}_{F_r} & \boldsymbol{0}\\ \boldsymbol{0} & \mathbb{R}_{F_l}^T\boldsymbol{W}_{F_l}\mathbb{R}_{F_l}\end{bmatrix}. \tag{5.134}$$

The submatrices $\boldsymbol{W}_{F_j}=\text{diag}(\alpha_{f_jx},\ \alpha_{f_jy},\ 1,\ \alpha_{m_jx},\ \alpha_{m_jy},\ \alpha_{m_jz})$ contain the weights for the components. This type of minimization yields a relatively conservative solution.

The above two contact wrench minimization approaches are of limited use, though, since they yield equal vertical force distribution that was shown to be inconsistent with the static states. A better approach would be a force control–based tracking of dynamically consistent reference contact wrench components. The objective function can then be formulated as

$$\min_{\boldsymbol{x}} c_{CW},$$

where $\boldsymbol{x} = \overline{\mathcal{F}}^c$, $c_{CW} = \|\overline{\mathcal{F}}^c - (\overline{\mathcal{F}}^c)^{ref}\|_{W_t}$, $(\overline{\mathcal{F}}^c)^{ref} \neq \mathbf{0}$. the minimization is achieved with the choice of $\boldsymbol{W}_{CW} = \boldsymbol{W}_t$ and $\boldsymbol{b}_{CW}$ defined from $\boldsymbol{J}_c\, \boldsymbol{b}_{CW} = -(\overline{\mathcal{F}}^c)^{ref}\boldsymbol{W}_t$ [126]. However, since this is a penalty-type approach; full consistency with the static states and/or exact contact wrench tracking is not guaranteed. Also, the determination of dynamically consistent reference contact wrench components is a nontrivial problem.

5.11 NONITERATIVE SPATIAL DYNAMICS–BASED MOTION OPTIMIZATION

Motion can be optimized as a subtask of a spatial dynamics–based sequential optimization (cf. Fig. 5.21). As a result of the optimization, a solution to the FSD problem will be obtained. This type of optimization is suitable for position-controlled robots. In torque-controlled robots, motion optimization is used as a control component. Implementation examples with torque-based control will be presented in Section 5.12.

5.11.1 Independent Motion Optimization With CRB Wrench-Consistent Input

To obtain a solution to the FSD problem, insert the solution to the WD problem, (5.114), into the spatial dynamics equation, (5.109). Then we have

$$\mathbb{M}_C \dot{\mathcal{V}}_M^{opt} + \boldsymbol{H}_{CM}\ddot{\boldsymbol{\theta}}^{opt} + \mathcal{C}_M + \mathcal{G}_C = \mathbb{C}_{cC}\mathbb{C}_{cC}^{-W}(\dot{\mathcal{L}}_C^{ref} + \mathcal{G}_C) + \mathbb{C}_{cC}\boldsymbol{N}(\mathbb{C}_{cC})\overline{\mathcal{F}}_a^c. \tag{5.135}$$

This equation simplifies to

$$\mathbb{M}_C \dot{\mathcal{V}}_M^{opt} + \boldsymbol{H}_{CM}\ddot{\boldsymbol{\theta}}^{opt} = \dot{\mathcal{L}}_C^{ref} - \mathcal{C}_M. \tag{5.136}$$

The gravity and null-space terms were annihilated by using $\mathbb{C}_{cC}\mathbb{C}_{cC}^{-W} = \boldsymbol{E}$ and $\mathbb{C}_{cC}\boldsymbol{N}(\mathbb{C}_{cC}) = \mathbf{0}$, respectively. Also, the nonlinear velocity-dependent term $\mathcal{C}_M$ will be partially annihilated under the assumption of perfect model matching.

The above result shows that motion optimization could be performed *independently* from the WD problem. The reference rate of change of the CRB momentum is assumed *contact-consistent*, s.t. $\dot{\mathcal{L}}_C^{ref} + \mathcal{G}_C = \mathcal{F}_C^{ref} \in \mathcal{F}_{BWC}$ holds. This is a necessary condition for using (5.136) in an independent motion optimization. If this condition is not satisfied, an on-line modification of the input trajectories will be required (cf. Fig. 5.1). In this case, the output from the WD problem, $\mathbb{C}_{cC}\overline{\mathcal{F}}_{opt}^c$, is used in place of $\dot{\mathcal{L}}_C^{ref}$ in (5.136) to obtain an "admissible" rate of change of the CRB momentum [88]. This implies that the WD and motion optimization problems cannot be solved independently. On the other hand, when the initially generated reference input $\dot{\mathcal{L}}_C^{ref}$ is feasible, there is no necessity to involve a *combined* motion/force optimization task. Thus, the computational overhead stemming from the annihilation of the null-space (internal force), gravity, and nonlinear velocity force components can be avoided. Moreover, the WD problem and motion optimization tasks can be invoked in parallel threads, leading to a further decrease in computational time. In conclusion, it is highly desirable to rely upon a contact and motion planning system capable of generating feasible inputs. Asymptotically stable whole-body balance control can then be achieved, as clarified in Section 5.8.7, in addition to the decreased computational overhead.

5.11.2 Stabilization With Angular Momentum Damping

A motion optimization with unconstrained hand motion implies that the motion in the arms, obtained as a result of the optimization, supports the main balance stabilization task. Some results in this direction have already been reported in the literature. In an early work [81], an arm "windmilling" motion was generated as the outcome of an iterative, QP optimization. In [112], a centroidal momentum balance controller generated an "unintentional" arm motion. Note, however, that the mechanism of such type of arm motion generation and the interference with the other balance control subtasks has not been clarified yet.

From Section 5.8.3 recall that the RAM/V plus RNS balance controller generated motion in the unconstrained limbs that injected angular momentum damping into the system. In this way, the balance controller was endowed with a highly desirable self-stabilization property. The controller was able to stabilize the unstable (foot roll) contact states without any special provisions, such as contact transition modeling. In this subsection, the RAM/V balance control approach will be reformulated using the *rate of change of the RAM/V*. Analysis at the acceleration level can then be performed, e.g. to exactly reveal the angular momentum damping mechanism.

The spatial dynamics of the robot can be represented in terms of spatial acceleration by premultiplying (5.109) by the inverse of the CRB inertia tensor; then we have

$$\mathbb{M}_C^{-1}\boldsymbol{H}_{CM}\ddot{\boldsymbol{\theta}} + \mathbb{M}_C^{-1}(\mathcal{C}_M + \mathcal{G}_C) = \dot{\mathcal{V}}_C - \dot{\mathcal{V}}_M. \tag{5.137}$$

Here $\mathbb{M}_C^{-1}\boldsymbol{H}_{CM} = \begin{bmatrix}\boldsymbol{0}^T & \boldsymbol{J}_\omega^T\end{bmatrix}$ (cf. (4.97)). Spatial velocity $\dot{\mathcal{V}}_C \equiv \mathbb{M}_C^{-1}\mathbb{C}_{cC}\overline{\mathcal{F}}^c$ is referred to as the *system spatial acceleration*. This quantity should be generated in compliance with the *force-task constraints* (i.e. friction cone and CoP-in-BoS constraints). The *CRB spatial acceleration* $\dot{\mathcal{V}}_M$, on the other hand, should be generated in agreement with the *motion task constraints*. The control joint acceleration can then be determined by solving a *noniterative* least-squares optimization task. Following the procedure in Section 5.8.3 for the velocities, first rewrite (5.137) componentwise, i.e.

$$\boldsymbol{a}_g = \dot{\boldsymbol{v}}_{C_R} - \dot{\boldsymbol{v}}_{C_I}, \tag{5.138}$$

$$\boldsymbol{J}_\omega\ddot{\boldsymbol{\theta}} = \dot{\boldsymbol{\omega}}_C - \dot{\boldsymbol{\omega}}_B - \boldsymbol{I}_C^{-1}\boldsymbol{c}_{mM}. \tag{5.139}$$

The nonlinear velocity-dependent moment $\boldsymbol{c}_{mM}$ is defined in (4.140). From the upper equation it is apparent that the acceleration of the gravity force, $\boldsymbol{a}_g$, is expressed as a *relative CoM acceleration*, $\Delta\dot{\boldsymbol{v}}_C = \dot{\boldsymbol{v}}_{C_R} - \dot{\boldsymbol{v}}_{C_I}$. The CoM acceleration component, $\dot{\boldsymbol{v}}_{C_R} \equiv \dot{\boldsymbol{v}}_C(\overline{\mathcal{F}}^c)$, that stems from the reaction (contact) wrenches is clearly distinguished from the *inertial* CoM acceleration component, $\dot{\boldsymbol{v}}_{C_I} \equiv \dot{\boldsymbol{v}}_C$. The CoM acceleration $\dot{\boldsymbol{v}}_{C_R}$ compensates the gravity CoM acceleration $\boldsymbol{a}_g$ at all postures, for any inertial acceleration of the CoM. In other words, *the effect of the gravity field on the spatial dynamics can be completely ignored*. It also becomes apparent that it will suffice to *formulate the optimization task for the joint acceleration in terms of the relative angular acceleration*, i.e.

$$\Delta\dot{\boldsymbol{\omega}} = \dot{\boldsymbol{\omega}}_C - \dot{\boldsymbol{\omega}}_B. \tag{5.140}$$

The desired CRB trajectories, $\dot{\mathcal{V}}_M^{des} = \begin{bmatrix}(\dot{\boldsymbol{v}}_{C_I}^{des})^T & (\dot{\boldsymbol{\omega}}_B^{des})^T\end{bmatrix}^T$, can be tracked with asymptotic stability in accordance with the results in Section 5.8.7, under the following kinematic control law:

$$\ddot{\boldsymbol{\theta}} = \mathcal{J}_{cM}^{+}\left(\dot{\bar{\mathcal{V}}}^{c} - \mathbb{C}_{cC}^{T}\dot{\mathcal{V}}_M^{ref} - \dot{\mathcal{J}}_{cM}\dot{\boldsymbol{\theta}} - \dot{\mathbb{C}}_{cC}^{T}\mathcal{V}_M\right) + \boldsymbol{N}(\mathcal{J}_{cM})\,\ddot{\boldsymbol{\theta}}_u. \tag{5.141}$$

This control law was derived from the time differential of the instantaneous motion constraint (2.125).

Usually, it is assumed that the desired CRB trajectories are feasible (contact-consistent) and that $\mathcal{J}_{cM}(\boldsymbol{q})$ is full (row) rank throughout the motion. When there is no guarantee that the desired CRB trajectories are contact-consistent or when modeling and other errors yield an unbounded reference input, the contact conditions will be violated, resulting e.g. in a foot roll that may lead to loss of balance. This problem can be addressed by adding a *relative angular acceleration control component*. To this end, determine the unconstrained joint acceleration $\ddot{\boldsymbol{\theta}}_u$ in (5.141) with the help of the angular acceleration relation (5.139). As a result, one arrives at

$$\ddot{\boldsymbol{\theta}}^{ref} = (\boldsymbol{E} - \overline{\boldsymbol{J}}_{\omega}^{+}\boldsymbol{J}_{\omega})\,\mathcal{J}_{cM}^{+}\left(\dot{\bar{\mathcal{V}}}^{c} - \mathbb{C}_{cC}^{T}\dot{\mathcal{V}}_M^{ref}\right) + \overline{\boldsymbol{J}}_{\omega}^{+}\Delta\dot{\boldsymbol{\omega}}^{ref} + \boldsymbol{N}(\mathcal{J}_{cM})\boldsymbol{N}(\overline{\boldsymbol{J}}_{\omega})\,\ddot{\boldsymbol{\theta}}_u^{ref} + \dot{\boldsymbol{\theta}}_{nl}, \tag{5.142}$$

where $\overline{\boldsymbol{J}}_{\omega} = \boldsymbol{J}_{\omega}\boldsymbol{N}(\mathcal{J}_{cM})$ and

$$\dot{\boldsymbol{\theta}}_{nl} = -(\boldsymbol{E} - \overline{\boldsymbol{J}}_{\omega}^{+}\boldsymbol{J}_{\omega})\,\mathcal{J}_{cM}^{+}\left(\dot{\mathcal{J}}_{cM}\dot{\boldsymbol{\theta}} + \dot{\mathbb{C}}_{cC}^{T}\mathcal{V}_M\right) - \overline{\boldsymbol{J}}_{\omega}^{+}\boldsymbol{I}_C^{-1}\boldsymbol{c}_{mM}.$$

Here $\ddot{\boldsymbol{\theta}}_u^{ref}$ is an arbitrary joint acceleration that comprises several additional joint-level control inputs. In the case of a single stance, for example, the motion of the swing foot can be constrained by the time derivative of (2.130), i.e.

$$\boldsymbol{J}_{mM}\ddot{\boldsymbol{q}}_M + \dot{\boldsymbol{J}}_{mM}\dot{\boldsymbol{q}}_M = \dot{\bar{\mathcal{V}}}^{m}. \tag{5.143}$$

This constraint is treated as a lower-priority task w.r.t. the balance control task. In other words, the arbitrary $\ddot{\boldsymbol{\theta}}_u^{ref}$ in (5.142) is to be resolved via (5.143). The form of the final solution is quite complicated and will be omitted here. The other components of $\ddot{\boldsymbol{\theta}}_u^{ref}$ should be designed to ensure joint velocity damping and joint-limit avoidance.

The control law (5.142) will be referred to as the *relative angular acceleration (RAA) balance controller*. The RAA controller can function as a DCM stabilizer. This becomes apparent by setting the reference inertial CoM trajectories $\dot{\boldsymbol{v}}_{C_I}^{ref}(t)$ in accordance with (5.107). Note that the DCM stabilization is achieved as a high-priority task. In addition to this important function, the RAA controller provides an input for controlling the rate of change of the angular momentum in an *independent way*. Being set at a lower-priority level, this subtask will not disturb the main DCM stabilization task.

In order to understand the role of the *independent* control of the rate of change of the angular momentum, consider first the two constraints for the (centroidal) system and coupling angular momentum conservation, (5.98) and (5.99), respectively. At the acceleration level, these conditions assume the following form:

$$\boldsymbol{\omega}_C^{ref} = \mathbf{0} \Rightarrow \dot{\boldsymbol{\omega}}_C^{ref} = -D_\omega \boldsymbol{\omega}_C, \tag{5.144}$$

$$\Delta\boldsymbol{\omega}^{ref} = \mathbf{0} \Rightarrow \Delta\dot{\boldsymbol{\omega}}^{ref} = -D_\omega \Delta\boldsymbol{\omega}, \tag{5.145}$$

respectively, D_ω denoting a PD damping gain. With these formulations, the integrability of the control joint acceleration (5.142) will be ensured. In other words, the RAA balance controller will behave as a RAM/V balance controller at the velocity level. Recall that the latter was capable of handling critical situations with unstable contacts (rolling feet), without the need to use any contact models. The above control inputs provide for a system angular damping that cannot be achieved with the high-priority DCM stabilization task.

First, consider the role of the system angular velocity damper (5.144). The optimal joint acceleration can be obtained from the rotational dynamics in the equation of motion with mixed quasivelocity, (4.158). We have

$$\boldsymbol{I}_C \dot{\boldsymbol{\omega}}_B^{ref} + \boldsymbol{H}_C \ddot{\boldsymbol{\theta}} + \boldsymbol{c}_{mM} = -D_\omega \boldsymbol{\omega}_C. \tag{5.146}$$

Note that the driving term $\mathbb{C}_{cC_m} \overline{\mathcal{F}}^c$ in (4.158) has been replaced with the damping term to ensure the angular momentum conservation constraint. Note also that the constraint-consistent solution to (5.146) yields exactly the RAA control law (5.142), wherein the reference system angular acceleration is replaced with the damping term.

Next, consider the role of the RAV damper (5.145). With this constraint, the RAA control law can be used to generate reactive synergies in response to large impact–type disturbances. Note that even when the impact induces a rolling feet motion, the RAA controller is capable of restoring a stable posture, unless the impact is very large, of course.

The RAA controller can be embedded into a general torque controller with a wrench distribution component. The block diagram of such a controller is shown in Fig. 5.28. The output of the RAA controller is the quasiacceleration $\ddot{\boldsymbol{q}}_M^{ref}$. The reference rate of change of the system spatial momentum is obtained in a straightforward way from this output and the reference CRB trajectories $\dot{\mathcal{V}}_M^{ref}$ as

$$\dot{\mathcal{L}}_C^{ref} = \boldsymbol{\mathcal{A}}_C \ddot{\boldsymbol{q}}_M^{ref} + \dot{\boldsymbol{\mathcal{A}}}_C \dot{\boldsymbol{q}}_M, \tag{5.147}$$

where

$$\boldsymbol{\mathcal{A}}_C = \begin{bmatrix} \mathbb{M}_C & \boldsymbol{H}_{CM} \end{bmatrix} \in \Re^{6\times(6+n)}.$$

Note that this matrix is similar to the centroidal momentum matrix appearing in [111,112]. The sum of $\dot{\mathcal{L}}_C^{ref}$ and the gravity wrench are then distributed at the contacts, e.g. with the VRP-GI (cf. Section 5.10.5). Finally, the control joint torque is obtained via the JSID solution.

The above controller provides for enhanced balance control. It is very efficient from a computational point of view since it does not rely on recursive optimization. Implementations of this controller will be presented in Section 7.7.4.

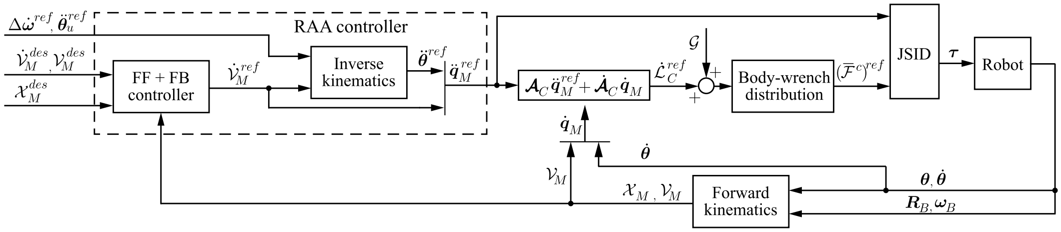


FIGURE 5.28 Block diagram of a torque controller with a RAA control component. The FF+FB controller includes feedforward and feedback control terms for CRB trajectory tracking. In particular, DCM stabilization can be embedded within this component along with the desirable base-link orientation.

5.11.3 Motion Optimization With Task-Based Hand Motion Constraints

The balance controller described in Section 5.11.2 is quite suitable for dynamic motion generation and control whereby the upper-body motion plays an important role and the arms are free to move. In tasks where the hand(s) (or other parts) come in contact with the environment or are required to track a desired trajectory, a slightly different approach to motion optimization is needed. The following derivations will be based on the formulation of spatial momentum w.r.t. the base link, $\mathcal{L}_B(\boldsymbol{q}, \dot{\boldsymbol{q}}_B) = \boldsymbol{\mathcal{A}}_B\dot{\boldsymbol{q}}_B$, as in (4.92). With this formulation, there is a slight expense to pay, though, since the transform (2.121) and its time derivative have to be involved in the balance control component.

The motion optimization task is defined by combining the rate of change of the spatial momentum with the second-order contact constraints. We have

$$\begin{bmatrix} \boldsymbol{\mathcal{A}}_B \\ \boldsymbol{J}_{cB} \end{bmatrix} \ddot{\boldsymbol{q}}_B + \begin{bmatrix} \dot{\boldsymbol{\mathcal{A}}}_B \\ \dot{\boldsymbol{J}}_{cB} \end{bmatrix} \dot{\boldsymbol{q}}_B = \begin{bmatrix} \dot{\tilde{\mathcal{L}}}_B^{ref} \\ \boldsymbol{0} \end{bmatrix}. \tag{5.148}$$

The following compact-form representation of the above equation will be used in the derivations:

$$\overline{\boldsymbol{\mathcal{A}}}_B\ddot{\boldsymbol{q}}_B + \dot{\overline{\boldsymbol{\mathcal{A}}}}_B\dot{\boldsymbol{q}}_B = \dot{\overline{\mathcal{L}}}_B^{ref}, \tag{5.149}$$

where $\overline{\boldsymbol{\mathcal{A}}}_B$ and $\dot{\overline{\mathcal{L}}}_B^{ref}$ denote the coefficient matrix for the acceleration and the term on the r.h.s. in (5.148), respectively. The above equation is solved for the joint acceleration. Then we have

$$\ddot{\boldsymbol{q}}_B = \overline{\boldsymbol{\mathcal{A}}}_B^{+}\left(\dot{\overline{\mathcal{L}}}_B^{ref} - \dot{\overline{\boldsymbol{\mathcal{A}}}}_B\dot{\boldsymbol{q}}_B\right) + \boldsymbol{N}(\overline{\boldsymbol{\mathcal{A}}}_B)\ddot{\boldsymbol{q}}_{Ba}. \tag{5.150}$$

Note that the solution is constraint-consistent and that the overbar notation is in agreement with the *constraint-consistent spatial momentum* introduced in Section 4.10 (cf. (4.154)). Note also that the two subtasks of controlling the rate of change of the spatial momentum and the motion constraints are arranged as a single control task, without any priority. This is reasonable since both subtasks play a crucial role in balance control.

Furthermore, the motion of the hand(s) along the unconstrained motion directions will be constrained by the time derivative of (2.95), i.e.

$$\boldsymbol{J}_{mB}\ddot{\boldsymbol{q}}_B + \dot{\boldsymbol{J}}_{mB}\dot{\boldsymbol{q}}_B = \dot{\overline{\mathcal{V}}}^m. \tag{5.151}$$

This constraint is treated as a lower-priority task w.r.t. the balance control task. In other words, the arbitrary $\ddot{\boldsymbol{q}}_{Ba}$ in (5.150) is to be resolved via (5.151). The final solution is then to be obtained as

$$\ddot{\boldsymbol{q}}_B = \ddot{\boldsymbol{q}}_1 + \overline{\boldsymbol{J}}^+_{mB}\left((\dot{\overline{\mathcal{V}}}^m)^{ref} - \boldsymbol{J}_{mB}\ddot{\boldsymbol{q}}_1\right) + \boldsymbol{N}(\overline{\boldsymbol{\mathcal{A}}}_B)\boldsymbol{N}\left(\overline{\boldsymbol{J}}_{mB}\right)\ddot{\boldsymbol{q}}_{Ba} + \boldsymbol{h}. \tag{5.152}$$

Hereby, $\ddot{\boldsymbol{q}}_1 = \overline{\boldsymbol{\mathcal{A}}}^+_B\dot{\overline{\mathcal{L}}}^{ref}_B$ is the solution to the highest-priority subtask (the balance control task), $\overline{\boldsymbol{J}}_{mB} = \boldsymbol{J}_{mB}\boldsymbol{N}(\overline{\boldsymbol{\mathcal{A}}}_B)$, and the nonlinear term

$$\boldsymbol{h} = -\left((\boldsymbol{E} - \overline{\boldsymbol{J}}^+_{mB}\boldsymbol{J}_{mB})\overline{\boldsymbol{\mathcal{A}}}^+_B\dot{\overline{\boldsymbol{\mathcal{A}}}}_B + \overline{\boldsymbol{J}}^+_{mB}\dot{\boldsymbol{J}}_{mB}\right)\dot{\boldsymbol{q}}_B.$$

The reference end-link twist is determined in the conventional way (feedforward plus PD feedback control) as follows:

$$(\dot{\overline{\mathcal{V}}}^m)^{ref} = (\dot{\overline{\mathcal{V}}}^m)^{des} + \boldsymbol{K}_{v_m}\left((\overline{\mathcal{V}}^m)^{des} - \overline{\mathcal{V}}^m\right) + \boldsymbol{K}_{p_m}\left((\overline{\mathcal{X}}^m)^{des} - \overline{\mathcal{X}}^m\right), \tag{5.153}$$

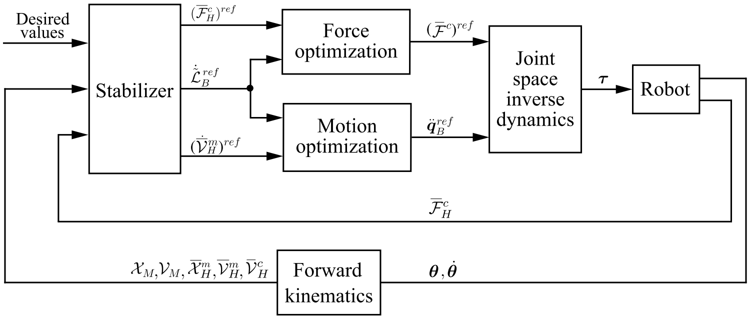
$\boldsymbol{K}_{v_m} = \text{diag}(K_{v_m}\boldsymbol{E},\ K_{\omega_m}\boldsymbol{E})$ and $\boldsymbol{K}_{p_m} = \text{diag}(K_{p_m}\boldsymbol{E},\ K_{o_m}\boldsymbol{E})$ denoting PD feedback gains. The arbitrary $\ddot{\boldsymbol{q}}_{Ba}$ in (5.152) can be reused to impose further constraints, e.g. joint limits and damping.

5.12 NONITERATIVE WHOLE-BODY MOTION/FORCE OPTIMIZATION

The WD (force) and FSD (motion) optimization approaches discussed in the last two sections can be combined in the design of a whole-body motion/force optimization–based balance control methods.

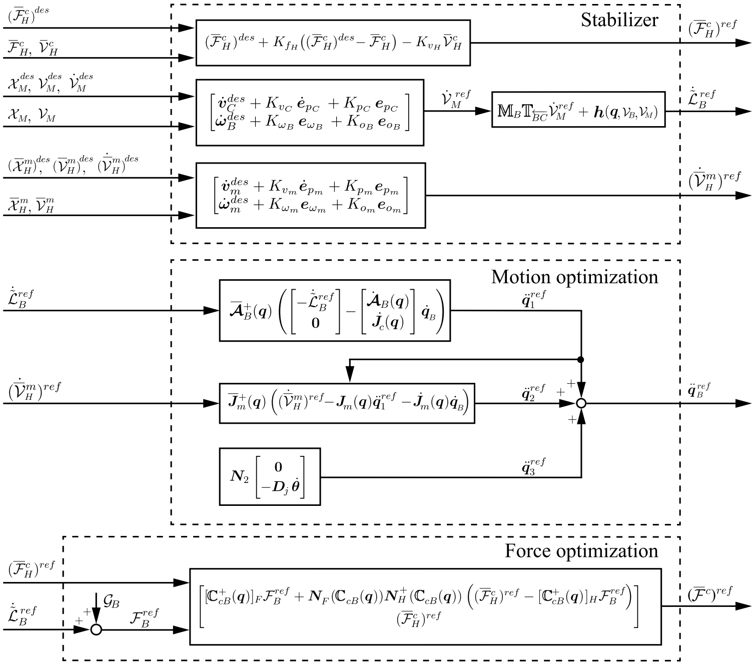
5.12.1 Multicontact Motion/Force Controller Based on the Closed-Chain Model

Balance control at multicontact postures, i.e. when the end links form interdependent closed loops via a common closure link (the environment), is envisioned. The controller should also ensure the end-link trajectory tracking along the unconstrained motion directions, as well as contact transitions. The structure of such controller is presented in Fig. 5.29. Feasible (contact-consistent) desired task trajectories are assumed, s.t. the stabilizer generates a feasible reference rate of change of the spatial momentum $\dot{\overline{\mathcal{L}}}^{ref}_B$. As already noted, asymptotic stability can then be ensured. In addition, the force and motion optimization tasks can be performed in parallel, as apparent from the figure. The unique solution to the JSID problem can then be obtained in a straightforward way, as explained in Section 5.9.1 (cf. (5.111)).



Desired values
Stabilizer
$(\overline{\mathcal{F}}_H^c)^{ref}$
$\dot{\mathcal{L}}_B^{ref}$
$(\dot{\overline{\mathcal{V}}}_H^m)^{ref}$
Force optimization
Motion optimization
$(\overline{\mathcal{F}}^c)^{ref}$
$\ddot{q}_B^{ref}$
Joint space inverse dynamics
τ
Robot
$\overline{\mathcal{F}}_H^c$
$\mathcal{X}_M, \mathcal{V}_M, \overline{\mathcal{X}}_H^m, \overline{\mathcal{V}}_H^m, \overline{\mathcal{V}}_H^c$
Forward kinematics
$\theta, \dot{\theta}$

FIGURE 5.29 Outline of a motion/force controller based on the closed-chain formulation. The controller ensures balance control (also in the presence of unexpected external disturbances) while performing motion/force control tasks with the hands.



Stabilizer
$(\overline{\mathcal{F}}_H^c)^{des}$
$\overline{\mathcal{F}}_H^c, \overline{\mathcal{V}}_H^c$
$(\overline{\mathcal{F}}_H^c)^{des} + K_{f_H}((\overline{\mathcal{F}}_H^c)^{des} - \overline{\mathcal{F}}_H^c) - K_{v_H}\overline{\mathcal{V}}_H^c$
$(\overline{\mathcal{F}}_H^c)^{ref}$
$\mathcal{X}_M^{des}, \mathcal{V}_M^{des}, \dot{\mathcal{V}}_M^{des}$
$\mathcal{X}_M, \mathcal{V}_M$
$\begin{bmatrix} \dot{v}_C^{des} + K_{v_C}\dot{e}_{p_C} + K_{p_C}e_{p_C} \\ \dot{\omega}_B^{des} + K_{\omega_B}e_{\omega_B} + K_{o_B}e_{o_B} \end{bmatrix}$
$\dot{\mathcal{V}}_M^{ref}$
$\mathbb{M}_B\mathbb{T}_{\overleftarrow{BC}}\dot{\mathcal{V}}_M^{ref} + h(q, \mathcal{V}_B, \mathcal{V}_M)$
$\dot{\mathcal{L}}_B^{ref}$
$(\overline{\mathcal{X}}_H^m)^{des}, (\overline{\mathcal{V}}_H^m)^{des}, (\dot{\overline{\mathcal{V}}}_H^m)^{des}$
$\overline{\mathcal{X}}_H^m, \overline{\mathcal{V}}_H^m$
$\begin{bmatrix} \dot{v}_m^{des} + K_{v_m}\dot{e}_{p_m} + K_{p_m}e_{p_m} \\ \dot{\omega}_m^{des} + K_{\omega_m}e_{\omega_m} + K_{o_m}e_{o_m} \end{bmatrix}$
$(\dot{\overline{\mathcal{V}}}_H^m)^{ref}$
Motion optimization
$\dot{\mathcal{L}}_B^{ref}$
$\overline{\mathcal{A}}_B^+(q)\left(\begin{bmatrix} -\dot{\mathcal{L}}_B^{ref} \\ 0 \end{bmatrix} - \begin{bmatrix} \dot{\mathcal{A}}_B(q) \\ \dot{J}_c(q) \end{bmatrix}\dot{q}_B\right)$
$\ddot{q}_1^{ref}$
$(\dot{\overline{\mathcal{V}}}_H^m)^{ref}$
$\overline{J}_m^+(q)\left((\dot{\overline{\mathcal{V}}}_H^m)^{ref} - J_m(q)\ddot{q}_1^{ref} - \dot{J}_m(q)\dot{q}_B\right)$
$\ddot{q}_2^{ref}$
+
+
+
$\ddot{q}_B^{ref}$
$N_2\begin{bmatrix} 0 \\ -D_j\dot{\theta} \end{bmatrix}$
$\ddot{q}_3^{ref}$
Force optimization
$(\overline{\mathcal{F}}_H^c)^{ref}$
$\dot{\mathcal{L}}_B^{ref}$
$\mathcal{G}_B$
+
+
$\mathcal{F}_B^{ref}$
$\begin{bmatrix} [\mathbb{C}_{cB}^+(q)]_F\mathcal{F}_B^{ref} + N_F(\mathbb{C}_{cB}(q))N_H^+(\mathbb{C}_{cB}(q))\left((\overline{\mathcal{F}}_H^c)^{ref} - [\mathbb{C}_{cB}^+(q)]_H\mathcal{F}_B^{ref}\right) \\ (\overline{\mathcal{F}}_H^c)^{ref} \end{bmatrix}$
$(\overline{\mathcal{F}}^c)^{ref}$

FIGURE 5.30 The equations used in the Stabilizer, Force and Motion optimization blocks in Fig. 5.29.

The equations used in the Stabilizer, Force, and Motion optimization blocks are shown in Fig. 5.30. Most of these equations have already been introduced in various parts of this text. Apparently, the Stabilizer comprises three control components. From the top to the bottom they are: (1) conventional force control in the constrained motion directions of the hands, (2) the asymptotic stabilization approach for balance control that was introduced in Section 5.8.7, and (3) conventional motion control in the unconstrained motion directions of the hands, as in (5.153). Note that in this particular implementation of the balance stabilizer, spatial momentum is represented in terms of the base coordinates, while the asymptotic stabilization controller is based on a mixed quasivelocity. A transformation is needed in this case. This transformation appears in the block on the right side of the Stabilizer.[11] The term $\boldsymbol{h}(\boldsymbol{q}, \mathcal{V}_B, \mathcal{V}_M) \approx \dot{\mathbb{T}}_{\overleftarrow{BC}} \mathcal{V}_M + \dot{\mathbb{M}}_B \mathcal{V}_B$ collects the nonlinear terms in the transformation.

Furthermore, the Motion optimization block in Fig. 5.30 also comprises three control components. From the top to the bottom they are: (1) motion resolution from the rate of change of the CRB momentum, (2) motion resolution along the unconstrained motion directions, and (3) a joint damping term that is used to suppress excessive motion stemming from kinematic redundancy. The term $\boldsymbol{N}_2$ denotes a projector onto the common null space of the two higher-priority subtasks (balance and end-link motion control). The equations used are those derived in Section 5.11.3. The Force optimization block solves the wrench distribution problem in parallel with motion optimization, via the quasistatic relation derived in Section 3.6.2 (cf. (3.66)).

The above controller is highly efficient from computational viewpoint and robust w.r.t. disturbances. The performance can be confirmed with the following example. The robot's task is to clean an inclined flat surface with the right hand. The hand moves in the sagittal plane along a straight-line trajectory in a cyclic manner, applying a desired force of magnitude 2 Nm in the normal (constrained motion) direction. The model of a small-size robot[12] is used in the simulation. The result is shown in Video 5.12-1 [138]. The data graphs are displayed in Fig. 5.31. As apparent from the right-arm position/orientation error and the hand-force graphs (the measured hand force is displayed in the local sensor frame), the controller is capable of performing the cleaning task accurately while keeping balance. The right-hand position/orientation errors can be further decreased by shifting the hand motion control task to the upper level in the hierarchy. In doing so, a controller comprising only two levels of hierarchy will be obtained: a higher-priority level for the motion control in the constrained and unconstrained motion directions and a lower-priority level for the joint damping control. It is also worth noting that this controller is capable of controlling the reaction to an external disturbance, in a compliant way via base-link motion. Thereby, the hand motion/force task performance degrades only slightly. This will be confirmed in Section 5.13.3.

5.12.2 Motion/Force Optimization Based on the Operational-Space Formulation

A hierarchical framework for a noniterative motion/force optimization based on the Operational-space formulation was developed in [134,133,135]. The motion/force tasks are embedded into the following quasistatic notation:

$$\boldsymbol{\tau} = \sum_i \overline{\boldsymbol{J}}_i^T \mathcal{F}_i^{ref} + \boldsymbol{N}^*(\boldsymbol{\tau}^n)^{ref} + (\boldsymbol{\tau}^{int})^{ref}. \tag{5.154}$$

[11] An approximation is used in the implementation under the assumption that the CoM does not accelerate significantly in the case of a multicontact posture.

[12] The model parameters are similar to those of the HOAP-2 robot [26]. For the numbering of the joints and other relevant data, see Section A.2.

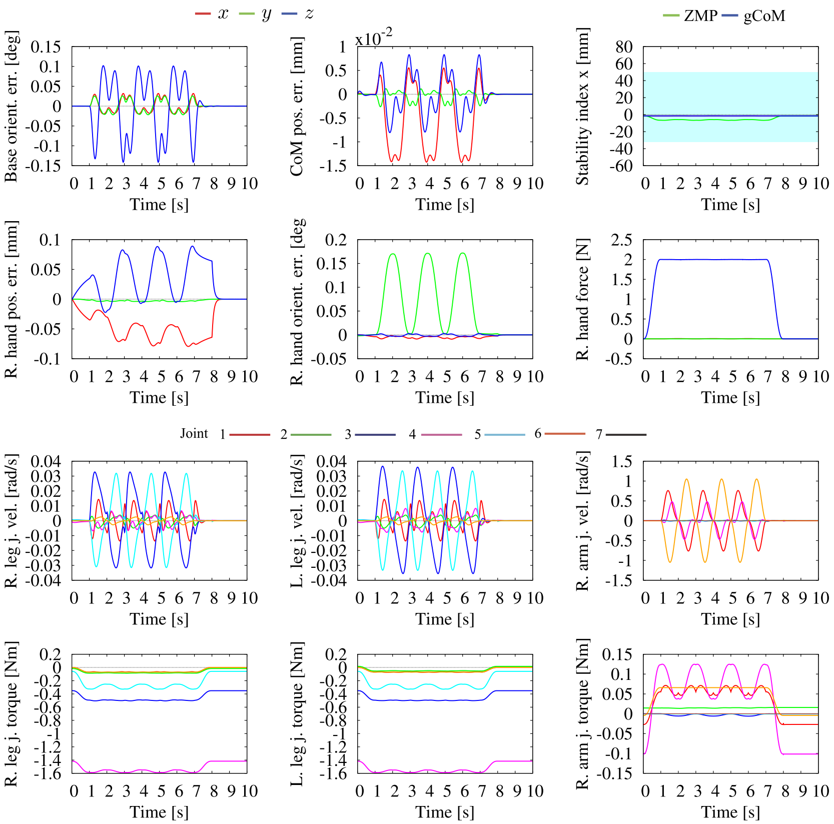


FIGURE 5.31 Data graphs from the simulation experiment with the multicontact motion/force controller, whereby the right hand cleans an inclined flat surface along a cyclic straight-line path by applying a desired force of 2 N. The desired CoM position and base orientation are constant (the initial values). R and L stand for "right" and "left," respectively. No external disturbance is present.

Here $\bar{\boldsymbol{J}}_i$ is the Jacobian for task i restricted by the null space of the contact constraints as well as by the null spaces of higher-priority tasks, $\boldsymbol{N}^*$ is a projector onto the intersection of all null spaces, and $\mathcal{F}_i^{ref}$, $(\boldsymbol{\tau}^n)^{ref}$, and $(\boldsymbol{\tau}^{int})^{ref}$ are reference inputs for motion/force control task i, postural variation control performed with the remaining DoFs, and internal force control, respectively.

The method provides a general platform for multicontact, whole-body balance control. The solution has been derived with the help of the dynamically consistent generalized inverse (cf. Section 4.52). Hence, the internal force controller can be designed in an independent way, without influencing the rest of the tasks. The controller adjusts the internal force to satisfy the friction cone and the CoP allocation constraints; $(\boldsymbol{\tau}^{int})^{ref}$ can also ensure adaptation to a dynamically varying environment, similarly to the pseudoinverse-based wrench distribution approach described in Section 5.10.1.

The method, however, has some shortcomings. First, since an off-line approach[13] is used to verify the feasibility of the solution, the method is unsuitable for real-time control. Second, since the solution to the WD problem is based on static relations only, the acceleration of the CoM is excessively restricted. The end-link CoPs are also allocated in a conservative manner, i.e. within the vicinity of the centers of the respective BoS.

Real-Time Implementation for Balance Control

A real-time implementation of the above method for balance control has been suggested in [89]. The proportional wrench distribution policy described in Section 5.10.3 is employed to ensure continuous double/single stance transitions, consistent with statics. A special task-priority assignment policy is adopted to handle overconstraints pertinent to the legs. Recall that such overconstraints stem from the spatial momentum/CoP interdependencies outlined in Section 5.8.1. The overconstraint problem can be tackled by introducing a two-level task hierarchy, T_{high} and T_{low}, denoting the high- and low-priority tasks, respectively. Eq. (5.154) is rewritten as

$$\boldsymbol{\tau} = \overline{\boldsymbol{J}}_{low}^T \mathcal{F}_{low}^{ref} + \boldsymbol{N}_{high}^* \boldsymbol{\tau}_a^{ref} + (\boldsymbol{\tau}^{int})^{ref}. \tag{5.155}$$

In [89], the high-priority task was set as the CoM motion in the vertical direction. Recall that CoM acceleration in this direction generates the vertical GRF that appears as denominator in the GRM equations; $\boldsymbol{\tau}_a^{ref}$ is determined in an appropriate way from the desired horizontal GRM components (i.e. the ankle torques). In this way, the CoP control that works even when the vertical GRF becomes zero[14] can be ensured. Note that the null-space CoP control term is not decoupled from the low-priority task term (the first term on the r.h.s.) since the null space of the high-priority task is different from $\mathcal{N}(\overline{\boldsymbol{J}}_{low})$. All low-priority control tasks, i.e. the CoM motion and base orientation control, will be disturbed by the null-space term. But this disturbance will be only a temporary one, under the assumption that the CoP control does not have to be active at all times. This becomes apparent from the flowchart of the balance controller shown in Fig. 5.32.

Example

The performance of the above balance controller will be demonstrated with a simple example. The robot is in a double stance, its feet being aligned. High friction contacts are assumed, s.t. $c_F = 12$. The goal is to shift the xCoM as much as possible forward and then to stabilize the posture without losing balance. This task can be accomplished with the simple controller in Fig. 5.25 that ensures asymptotic trajectory tracking of feasible desired CRB trajectories and the DCM-GI-based body wrench distribution approach. In this example, the null-space solution component in (5.127) does not need to be included since double stance on a flat floor is assumed and, thus, there is no need of internal wrench control.

[13] "Stability cloud" scanning.

[14] Recall that a zero vertical GRF is a necessary condition for continuous double/single-stance transitions, consistent with statics.

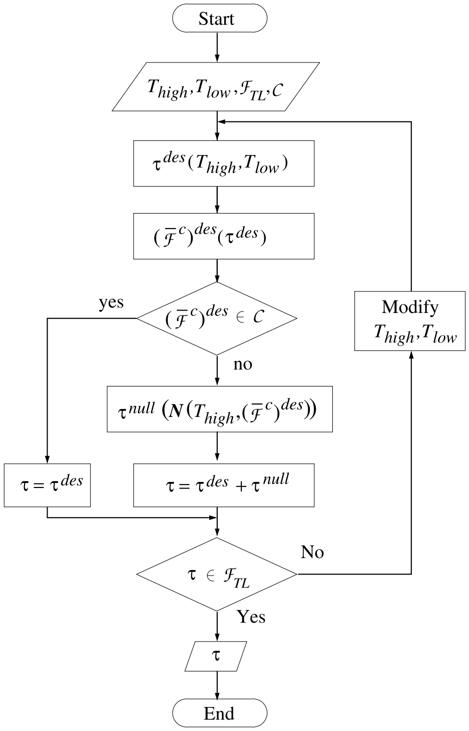

Start
$T_{high}, T_{low}, \mathcal{F}_{TL}, C$
$\tau^{des}(T_{high}, T_{low})$
$(\bar{\mathcal{F}}^c)^{des}(\tau^{des})$
$(\bar{\mathcal{F}}^c)^{des} \in C$
yes
no
Modify T_{high}, T_{low}
$\tau^{null}\ (N(T_{high}, (\bar{\mathcal{F}}^c)^{des}))$
$\tau = \tau^{des}$
$\tau = \tau^{des} + \tau^{null}$
$\tau \in \mathcal{F}_{TL}$
No
Yes
τ
End

FIGURE 5.32 Flowchart of a real-time balance controller based on the Operational-space formulation [89]. The null-space of the high-priority task, T_{high}, is used to enforce the contact wrench constraints, s.t. $(\bar{\mathcal{F}}^c)^{des} \in CWC$. The tasks may have to be modified to avoid torque limit violation (the branch on the r.h.s.).

The controller in Fig. 5.25 is augmented with the high-priority control component $\boldsymbol{N}^*_{high}\boldsymbol{\tau}_a^{ref}$, whereby $\boldsymbol{N}^*_{high} = \boldsymbol{N}(\boldsymbol{\mathcal{J}}_{cM_Z}^T)$, $\boldsymbol{\mathcal{J}}_{cM_Z} \in \Re^{1\times n}$ denoting the constraint Jacobian component in the vertical z-direction. This component is invoked in critical situations only, e.g. when the ZMP enters a predetermined safety margin within the BoS. The high-priority control component is capable of inducing slight torso rotations and CoM vertical displacements. As a result, the output of the "conventional stabilizer" will be modified to constrain any excessive torque in the ankles. The block diagram of the augmented controller is shown in Fig. 5.33.

The details of the implementation are as follows. The reference joint torque for the high-priority task is

$$\boldsymbol{\tau}_a^{ref} = -\bar{\boldsymbol{\mathcal{J}}}^T \boldsymbol{m}^{ref} \in \Re^n, \tag{5.156}$$

$$\boldsymbol{m}^{ref} = \mathbb{S}^{\times} K_p(\boldsymbol{r}_p^{SM} - \boldsymbol{r}_p) \in \Re^2. \tag{5.157}$$

Here $\boldsymbol{m}^{ref}$ denotes the reference pitch moments in the two ankles. These are obtained via tangential forces $K_p(\boldsymbol{r}_p^{SM} - \boldsymbol{r}_p)$, where $\boldsymbol{r}_p^{SM}$ is the predetermined BoS safety margin, K_p denoting

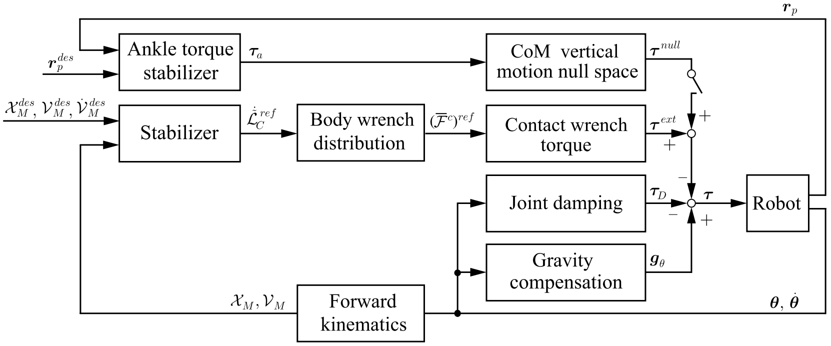

$\boldsymbol{r}_p$
Ankle torque stabilizer
$\boldsymbol{r}_p^{des}$
$\boldsymbol{\tau}_a$
CoM vertical motion null space
$\boldsymbol{\tau}^{null}$
$\mathcal{X}_M^{des}, \mathcal{V}_M^{des}, \dot{\mathcal{V}}_M^{des}$
Stabilizer
$\dot{\mathcal{L}}_C^{ref}$
Body wrench distribution
$(\overline{\mathcal{F}}^c)^{ref}$
Contact wrench torque
$\boldsymbol{\tau}^{ext}$
+
+
−
Joint damping
$\boldsymbol{\tau}_D$
−
+
$\boldsymbol{\tau}$
Robot
Gravity compensation
$\boldsymbol{g}_\theta$
$\mathcal{X}_M, \mathcal{V}_M$
Forward kinematics
$\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}$

FIGURE 5.33 The control block-diagram of the simple balance control in Fig. 5.25 is augmented with the $\boldsymbol{\tau}^{null}$ control component. This component is invoked in critical situations only, to ensure stabilization via slight upper-body postural variations, e.g. trunk rotation and CoM motion in the vertical direction. These variations result from the control algorithm; there is no need to preprogram them.

a p.d. feedback gain. The Jacobian

$$\overline{\mathcal{J}} = \boldsymbol{S}_p {}^{M}\mathbb{X}_{F_j \overleftarrow{FM}} \mathcal{J}_{cM} \in \Re^{2\times n},$$

where

$$\begin{bmatrix} {}^{M}\mathbb{X}_{F_R \overleftarrow{FM}} & {}^{M}\mathbb{X}_{F_L \overleftarrow{FM}} \end{bmatrix} \in \Re^{6\times 12}.$$

Here $\boldsymbol{S}_p \in \Re^{2\times 6}$ selects the pitch angular velocity components in the ankles. The graphs from the simulation are displayed in Fig. 5.34. The stability margin is set at 10 mm from the "toe" line of the BoS, as apparent from the CoP graphs (the pink area). The CoP/ZMP enters the margin at $t = 20$ s. As a result, the high-priority control component is invoked, resulting in a slight displacement of the CoM downward and a slight base rotation, as apparent from the respective graphs. The joint velocities change drastically but in a stable way. The pitch ankle torques are constrained and the posture can be stabilized without a fall. The plots in the bottom line display the pitch ankle torques and zoomed-in CoP motion for the right and left foot. The red/blue line graphs are for the cases when the high-priority control component is activated/deactivated. Note that the CoP trajectories are identical during most of the time upon entering the stability margin. At the end, the high-priority component stabilizes the posture at the "toe" boundary of the BoS, which is quite amazing. In contrast, without the activation, it is impossible to stabilize the posture. The simulation result is presented in animated form in Video 5.12-2 [33].

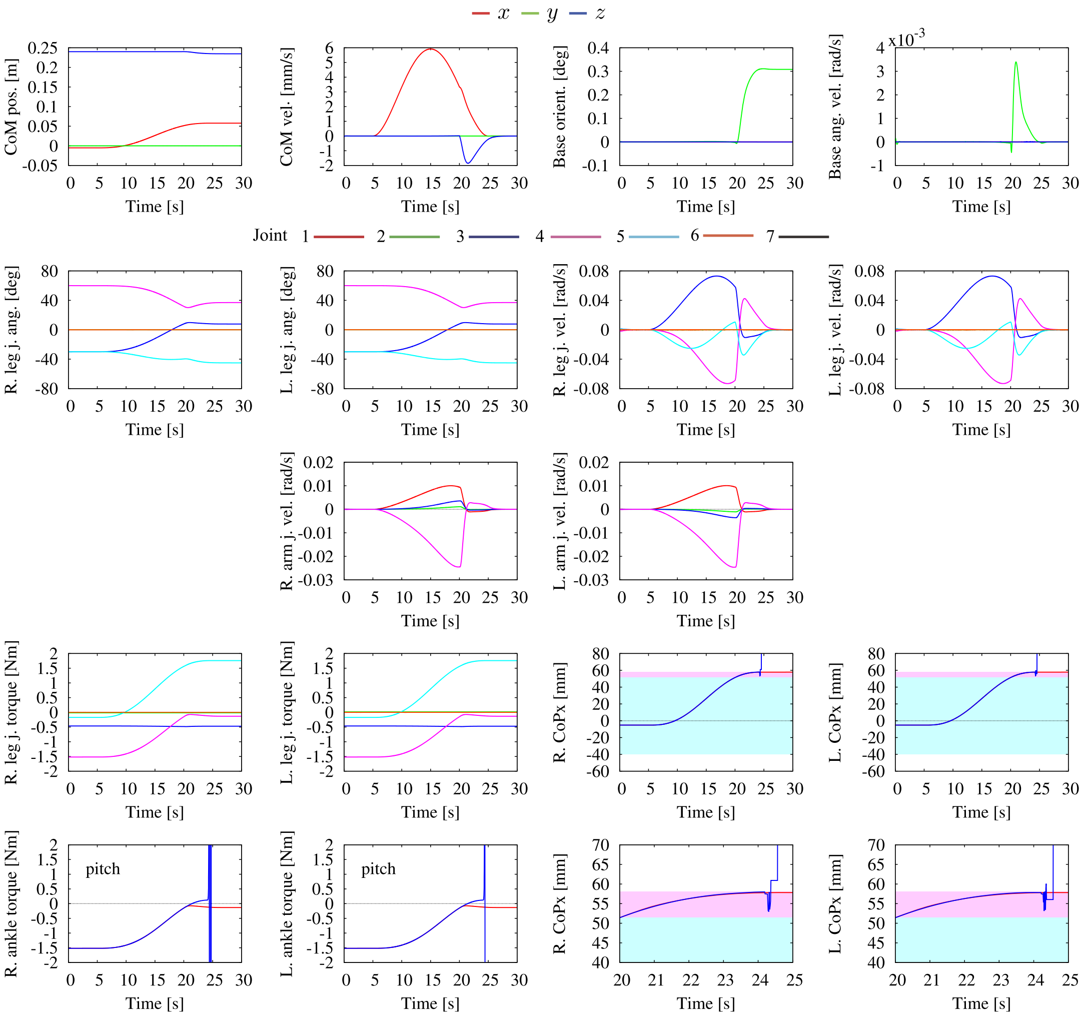


FIGURE 5.34 The high-priority $\boldsymbol{\tau}^{null}$ component is invoked in a critical situation, i.e. when the CoPs enter the BoS safety margin (the pink area). The robot is then able to stabilize the posture at the edge of the BoS (apparent from the red lines in the bottom-row graphs) via a slight upper-body rotation and the CoM downward motion, as seen in the upper-row graphs. Such a stabilization results in limiting the ankle pitch torques, which is impossible to achieve without activating $\boldsymbol{\tau}^{null}$ (see the blue lines in the bottom-row graphs).

5.13 REACTIVE BALANCE CONTROL IN RESPONSE TO WEAK EXTERNAL DISTURBANCES

An important feature of the balance controller is its robustness w.r.t. external disturbances. This issue was left out of the discussion so far. In Section 5.5 it was only mentioned that the LRWP model provides a possibility to increase the robustness via the angular momentum component. In Section 5.6.3 it was also mentioned that the DCM stabilizer is robust against a constant external disturbance. In this section, compliance-based control approaches that can ensure balance in the presence of relatively small external disturbances will be introduced. Balance control methods for accommodating larger disturbances will be introduced in Section 7.7.

5.13.1 Gravity Compensation–Based Whole-Body Compliance With Passivity

A whole-body compliant–type behavior is appropriate to accommodate relatively weak unknown external disturbances. Such type of behavior can be obtained with a humanoid robot controlled by a simple gravity compensating torque controller [63]. The control model is derived from the equation of motion in mixed quasicoordinates, (4.158). Using the notation $\boldsymbol{f}_C = \mathbb{C}_{cC_f}\overline{\mathcal{F}}^c$, $\mathbb{C}_{cC_f}$ defined in (2.127), the CoM partial dynamics (given in the upper row of (4.158)) can be written as

$$M\dot{\boldsymbol{v}}_C = \boldsymbol{f}_C - \boldsymbol{g}_f, \tag{5.158}$$

$\boldsymbol{g}_f = M\boldsymbol{a}_g$. Hereby, $\mathbb{C}_{cC_f}\mathbb{C}^{\#}_{cC_f} = \boldsymbol{E}$ was used. To derive the control joint torque, consider the lower part of (4.158), rewritten as

$$\begin{bmatrix} \boldsymbol{I}_C & \boldsymbol{H}_C \\ \boldsymbol{H}_C^T & \boldsymbol{M}_{\theta_M} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{\omega}}_B \\ \ddot{\boldsymbol{\theta}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{c}_{mM} \\ \boldsymbol{c}_{\theta_M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbb{C}_{cC_m}\mathbb{C}^{\#}_{cC_f} - [\boldsymbol{r}^{\times}_{\overleftarrow{CB}}] \\ \boldsymbol{\mathcal{J}}^T_{cM}\mathbb{C}^{\#}_{cC_f} - \boldsymbol{J}^T_{\overleftarrow{CB}} \end{bmatrix} \boldsymbol{f}_C. \tag{5.159}$$

The coefficient block matrices on the l.h.s. and r.h.s. will be denoted as $\boldsymbol{I}$ and $\boldsymbol{J}^T$, respectively. The control joint torque that compensates the gravity force can be derived from

$$\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{\tau} \end{bmatrix} = \boldsymbol{J}^T(\boldsymbol{f}_C^{des} - \boldsymbol{g}_f). \tag{5.160}$$

Note that, by assuming a quasistatic state, the appearance of nonlinear velocity terms has been avoided.[15] Furthermore, it is straightforward to show [63] that with

$$\boldsymbol{f}_C^{des} = \left(\boldsymbol{E} + (M\boldsymbol{J}\boldsymbol{I}^{-1}\boldsymbol{J}^T)^{-1}\right)\boldsymbol{f}_C^{ref} \tag{5.161}$$

the closed-loop system yields $\boldsymbol{f}_C \approx \boldsymbol{f}_C^{ref} + \boldsymbol{g}_f$ and, ultimately, $M\dot{\boldsymbol{v}}_C \approx \boldsymbol{f}_C^{ref}$. It is possible to avoid the inverse inertia term in (5.161) by using $\boldsymbol{f}_C^{ref}$ in place of $\boldsymbol{f}_C^{des}$ in (5.160). Then, with $\boldsymbol{f}_C^{ref} = \boldsymbol{0}$, the joint torque merely compensates the gravity force, conserving thereby the linear momentum.

On the other hand, noting that Jacobian $\boldsymbol{J}$ is nonsquare, a null-space term needs to be added in (5.160). This term will induce internal motion (or self-motion), though, that needs to be suppressed. The simplest way to do this is to add joint damping. The closed-loop system dynamics then assume the following form:

$$M\dot{\boldsymbol{v}}_C = \boldsymbol{f}_C^{ref} + \boldsymbol{f}_D + \boldsymbol{f}_{ext}, \tag{5.162}$$

[15] Such terms can cause problems stemming from noisy signal readings.

$$\boldsymbol{I}\ddot{\boldsymbol{q}}_{\omega}+\boldsymbol{C}=\boldsymbol{J}^{T}\boldsymbol{f}_{C}^{ref}-\boldsymbol{D}\dot{\boldsymbol{q}}_{\omega}, \tag{5.163}$$

where $\dot{\boldsymbol{q}}_{\omega}=\begin{bmatrix}\boldsymbol{\omega}_B^T & \dot{\boldsymbol{\theta}}^T\end{bmatrix}^T$, $\boldsymbol{C}$ collects the nonlinear velocity-dependent terms, $\boldsymbol{D}$ is the damping matrix, and $\boldsymbol{f}_D$ is the damping force mapped at the CoM. Note also that the external force $\boldsymbol{f}_{ext}$ has been added. This force is used as the driving input during the reflex phase. Thereby, the system passivity will be ensured, s.t.

$$\frac{1}{2}M\boldsymbol{v}_C^T\boldsymbol{v}_C+\frac{1}{2}\dot{\boldsymbol{q}}_{\omega}\boldsymbol{I}\dot{\boldsymbol{q}}_{\omega}^T=\frac{1}{2}\dot{\boldsymbol{q}}_{\omega}^T(M\boldsymbol{J}^T\boldsymbol{J}+\boldsymbol{I})\dot{\boldsymbol{q}}_{\omega}\leq\int(\boldsymbol{f}_C^{ref})^T\boldsymbol{J}\dot{\boldsymbol{q}}_{\omega}dt+\int\boldsymbol{f}_{ext}^T\boldsymbol{J}\dot{\boldsymbol{q}}_{\omega}dt \tag{5.164}$$

holds. After the disturbance, a recovery phase control is invoked based on a simple PD feedback control law. We have

$$\boldsymbol{f}_C^{ref}=-\boldsymbol{K}_D(\dot{\boldsymbol{r}}_C-\dot{\boldsymbol{r}}_C^{des})-\boldsymbol{K}_P(\boldsymbol{r}_C-\boldsymbol{r}_C^{des}). \tag{5.165}$$

Here $\boldsymbol{K}_D$ and $\boldsymbol{K}_P$ denote the feedback gains. The CoM initial position, $\boldsymbol{r}_C^{init}$, is recovered with the settings $\boldsymbol{r}_C^{des}=\boldsymbol{r}_C^{init}$ and $\dot{\boldsymbol{r}}_C^{des}=\boldsymbol{0}$. The desired ZMP can be determined from the above $\boldsymbol{f}_C^{ref}$ in a straightforward manner, under the assumption of a negligible angular momentum. Finally, the CoP-in-BoS constraint is imposed on the desired ZMP thus determined.

The above balance controller has the advantage of simplicity and robustness stemming from passivity. There is no need of a precise dynamic model of the robot. The performance of the controller was experimentally examined in combination with the noniterative wrench distribution technique described in Section 5.10.1. The quasistatic and wrench distribution approaches used in the controller have been confirmed to yield robust whole-body postural variations in response to relatively weak disturbances applied at arbitrary parts of the body [63]. The robot was also capable of adaptation to uneven terrain [62]. The controller does not provide any means for reactive stepping under stronger disturbances, however.

The performance of the controller can be improved in two main aspects [114]. First, it is straightforward to add the capability of dealing with external moment–type disturbances. This can be done by determining a reference CRB moment derived from a PD feedback body-orientation control law, using suitable local coordinates (e.g. the quaternion notation). Second, instead of generalized inverse–based optimization, the WD problem could be handled with iterative optimization. In this way, the GRF/GRM/CoP interdependencies (cf. Section 5.8.1) can be resolved by involving soft constraints, using sequential or nonsequential optimization, e.g. as described in Section 5.14.4. In this case, there is no need to determine a desired ZMP, which leads to an advantage as clarified with the DCM generalized inverse method in Section 5.10.4. In [114], one of the soft constraints in the iterative optimization was selected to minimize the Euclidean norm of the contact forces. This, however, leads to equal distribution which inhibits reactive stepping, as already clarified.

5.13.2 Whole-Body Compliance With Multiple Contacts and Passivity

Multicontact balance control refers to the case when the robot contacts the environment with one or two hands or any other intermittent links, while in a single or double stance. A number of tasks exist that require multicontact balance control, e.g. painting a wall [13], cleaning a surface with one of the hands while supporting the weight with the other [135], sitting on a chair [90], ladder climbing [110,16,24,153], pivoting heavy objects [102,104], pushing heavy objects with shoulder–elbow or hip [103], and stair climbing with handrail support [80,83]. Multicontact balance control implies a closed-chain formulation with a single loop-closure link (the environment), whereby interdependent closed loops are formed (cf. Section 3.6).

A possible approach to design a balance controller for the case of multiple contacts is to make use of a dynamic model in terms of spatial (end-link) coordinates (cf. Section 4.228). As already discussed, the representation of dynamics in mixed quasicoordinates is quite convenient for balance control. The underlying velocity relation is derived from (2.131) as

$$\begin{bmatrix} \mathcal{V}_M \\ \mathcal{V} \end{bmatrix} = \begin{bmatrix} \boldsymbol{E} & \boldsymbol{0} \\ \mathbb{C}_C^T & \boldsymbol{\mathcal{J}}_M \end{bmatrix} \begin{bmatrix} \mathcal{V}_M \\ \dot{\boldsymbol{\theta}} \end{bmatrix}. \tag{5.166}$$

The 4 × 4 block matrix on the r.h.s., denoted as $\boldsymbol{T}_M$, is used as a transformation matrix for the equation of motion in accordance with the transformation procedure described in Section 4.228 (see also [38]). The original transformation matrix, $\boldsymbol{T}_B$, is thereby replaced with $\boldsymbol{T}_M$. After the transformation, the equation of motion assumes the following form:

$$\boldsymbol{\mathcal{M}}_M(\boldsymbol{q}) \begin{bmatrix} \dot{\mathcal{V}}_M \\ \dot{\mathcal{V}} \end{bmatrix} + \boldsymbol{\mathcal{C}}_M(\boldsymbol{q}, \dot{\boldsymbol{q}}) \begin{bmatrix} \mathcal{V}_M \\ \mathcal{V} \end{bmatrix} + \boldsymbol{g}(\boldsymbol{q}) = \begin{bmatrix} -\mathbb{C}_C \\ \boldsymbol{E} \end{bmatrix} \boldsymbol{\mathcal{J}}_M^{-T} \boldsymbol{\tau} + \boldsymbol{T}_M^{-T} \mathcal{Q}_{ext}. \tag{5.167}$$

The following remarks are due. First, note that to support a controller design, the nonlinear velocity-dependent term has been recast into a vector matrix form. Second, note that the gravity term is unaltered ($\boldsymbol{g}_M(\boldsymbol{q}) = \boldsymbol{g}(\boldsymbol{q})$) since, with mixed quasicoordinates, the CoM dynamics are decoupled from the rest. Third, observe that the transformation involves the inverse of $\boldsymbol{T}_M$, which implies that the Jacobian $\boldsymbol{\mathcal{J}}_M$ is assumed square (nonredundant limbs) and nonsingular. Finally, note that an external force term has been added (the last term on the r.h.s.) to account for unexpected perturbations.

Furthermore, the last equation can be structured via the notation of constrained/unconstrained motion directions at the contact joints. Then, the first terms on the l.h.s. and the r.h.s. become

$$\boldsymbol{\mathcal{M}}_M(\boldsymbol{q}_M) \begin{bmatrix} \dot{\mathcal{V}}_M \\ \dot{\bar{\mathcal{V}}}^c \\ \dot{\bar{\mathcal{V}}}^m \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} -\mathbb{C}_C \boldsymbol{\mathcal{J}}_M^{-T} \\ \boldsymbol{\mathcal{J}}_{Mc}^{-T} \\ \boldsymbol{\mathcal{J}}_{Mm}^{-T} \end{bmatrix} \boldsymbol{\tau} \equiv \begin{bmatrix} \mathcal{F}^M \\ \bar{\mathcal{F}}^c \\ \bar{\mathcal{F}}^m \end{bmatrix}, \tag{5.168}$$

respectively. This structuring supports the three basic control tasks: the CRB (i.e. momentum rate) motion control, the contact wrench force control (i.e. the wrench distribution problem), and the end-link motion control along the unconstrained motion directions.

The desired closed-loop behavior is determined as [38]

$$\boldsymbol{\mathcal{M}}_M \begin{bmatrix} \Delta\dot{\mathcal{V}}_M \\ \boldsymbol{0} \\ \Delta\dot{\bar{\mathcal{V}}}^m \end{bmatrix} + \boldsymbol{\mathcal{C}}_M \begin{bmatrix} \Delta\mathcal{V}_M \\ \boldsymbol{0} \\ \Delta\bar{\mathcal{V}}^m \end{bmatrix} = \boldsymbol{T}_M^{-T} \mathcal{Q}_{ext} - \begin{bmatrix} \mathcal{F}_u^M \\ \bar{\mathcal{F}}_u^c \\ \bar{\mathcal{F}}_u^m \end{bmatrix}, \tag{5.169}$$

where the subscript u stands for the control input and $\Delta(\circ)$ denotes an error term. The CRB ($\mathcal{F}_u^M$) and end-link motion ($\overline{\mathcal{F}}_u^m$) control inputs comprise PD feedback components. This implies that the motion of the CRB is controlled via a virtual linear/torsional spring-damper system. The end-link motion control input comprises also a desired feedforward (inertia force) component. The resulting end-link behavior is of impedance type. Note that the motion along the constrained motion directions is inhibited and, therefore, $\Delta\overline{\mathcal{V}}^c$ has been assumed zero.

The contact wrenches, $\overline{\mathcal{F}}_u^c$, can be determined from the spatial dynamics. To this end, first note that from (5.167) and (5.169), one obtains

$$\begin{bmatrix} -\mathbb{C}_C \\ \boldsymbol{E} \end{bmatrix} \boldsymbol{\mathcal{J}}_M^{-T}\boldsymbol{\tau} = \boldsymbol{\mathcal{M}}_M \begin{bmatrix} \dot{\mathcal{V}}_M^{des} \\ \mathbf{0} \\ (\dot{\overline{\mathcal{V}}}^m)^{des} \end{bmatrix} + \boldsymbol{\mathcal{C}}_M \begin{bmatrix} \mathcal{V}_M^{des} \\ \mathbf{0} \\ (\overline{\mathcal{V}}^m)^{des} \end{bmatrix} + \boldsymbol{g} - \begin{bmatrix} \mathcal{F}_u^M \\ \overline{\mathcal{F}}_u^c \\ \overline{\mathcal{F}}_u^m \end{bmatrix}. \tag{5.170}$$

Next, the above equation is decomposed into two parts in accordance with the projector in the joint torque term. Then we have

$$-\mathbb{C}_C\left(\boldsymbol{\mathcal{J}}_M^{-T}\boldsymbol{\tau}\right) = \boldsymbol{\mathcal{M}}_{M1} \begin{bmatrix} \dot{\mathcal{V}}_M^{des} \\ \mathbf{0} \\ (\dot{\overline{\mathcal{V}}}^m)^{des} \end{bmatrix} + \boldsymbol{\mathcal{C}}_{M1} \begin{bmatrix} \mathcal{V}_M^{des} \\ \mathbf{0} \\ (\overline{\mathcal{V}}^m)^{des} \end{bmatrix} + \begin{bmatrix} M\boldsymbol{a}_g \\ \mathbf{0} \end{bmatrix} - \mathcal{F}_u^M, \tag{5.171}$$

$$\boldsymbol{\mathcal{J}}_M^{-T}\boldsymbol{\tau} = \boldsymbol{\mathcal{M}}_{M2} \begin{bmatrix} \dot{\mathcal{V}}_M^{des} \\ \mathbf{0} \\ (\dot{\overline{\mathcal{V}}}^m)^{des} \end{bmatrix} + \boldsymbol{\mathcal{C}}_{M2} \begin{bmatrix} \mathcal{V}_M^{des} \\ \mathbf{0} \\ (\overline{\mathcal{V}}^m)^{des} \end{bmatrix} - \mathcal{F}_u^*, \tag{5.172}$$

where $\mathcal{F}_u^* \equiv \begin{bmatrix} (\overline{\mathcal{F}}_u^c)^T & (\overline{\mathcal{F}}_u^m)^T \end{bmatrix}^T$. To obtain a representation in terms of spatial dynamics, eliminate the joint torque from the last two equations, i.e.

$$\mathbb{C}_C\mathcal{F}_u^* = (\boldsymbol{\mathcal{M}}_{M1} + \mathbb{C}_C\boldsymbol{\mathcal{M}}_{M2}) \begin{bmatrix} \dot{\mathcal{V}}_M^{des} \\ \mathbf{0} \\ (\dot{\overline{\mathcal{V}}}^m)^{des} \end{bmatrix} + (\boldsymbol{\mathcal{C}}_{M1} + \mathbb{C}_C\boldsymbol{\mathcal{C}}_{M2}) \begin{bmatrix} \mathcal{V}_M^{des} \\ \mathbf{0} \\ (\overline{\mathcal{V}}^m)^{des} \end{bmatrix} + \begin{bmatrix} M\boldsymbol{a}_g \\ \mathbf{0} \end{bmatrix} - \mathcal{F}_u^M. \tag{5.173}$$

The driving force of the spatial dynamics, $\mathcal{F}_u^M$, is composed by PD feedback control terms, as already noted. The only unknown quantity is the contact wrench, $\overline{\mathcal{F}}_u^c$. It is easy to confirm that (5.173) is underdetermined w.r.t. the contact wrench. Hence, to solve the wrench distribution problem, optimization should be invoked.

The optimization approach suggested in [38] is based on soft constraints. The objective function minimizes the sum of three quadratic (penalty) terms determined as residuals w.r.t. $(\overline{\mathcal{F}}^c)^{des}$, $\overline{\mathcal{F}}_u^m$ and (5.173), over $\boldsymbol{x} = \mathcal{F}^*$. The constraints are composed of the $\mathfrak{F}_{FC}$, $\mathfrak{F}_{BoS}$, and $\mathfrak{F}_{TL}$ subdomains. To maintain the contacts, minimal vertical GRFs are required as an additional constraint. Note that the contact wrench penalty term is defined w.r.t. the desired values. In this way, the need of contact wrench measurement for balance can be avoided. The unique solution for the control joint torque[16] is derived from (5.172) as

$$\boldsymbol{\tau} = \mathcal{J}_M^T \left(\boldsymbol{\mathcal{M}}_{M2} \begin{bmatrix} \dot{\mathcal{V}}_M^{des} \\ \mathbf{0} \\ (\dot{\bar{\mathcal{V}}}^m)^{des} \end{bmatrix} + \boldsymbol{\mathcal{C}}_{M2} \begin{bmatrix} \mathcal{V}_M^{des} \\ \mathbf{0} \\ (\bar{\mathcal{V}}^m)^{des} \end{bmatrix} - \mathcal{F}_{opt}^* \right). \tag{5.174}$$

This expression is also used in the joint torque-limit constraint, $\mathcal{F}_{TL}$. From the above derivations, it is apparent that the optimization approach can be characterized as a nonsequential, spatial dynamics–based one.

The controller thus designed has the property of asymptotic stability and passivity. To show this, note first that under the assumption of maintaining the balancing contact wrenches, the dimension of the system can be reduced. This is done via the constraint elimination approach. The closed-loop system of reduced dimension becomes

$$\boldsymbol{\mathcal{M}}_M^* \begin{bmatrix} \Delta\dot{\mathcal{V}}_M \\ \Delta\dot{\bar{\mathcal{V}}}^m \end{bmatrix} + \boldsymbol{\mathcal{C}}_M^* \begin{bmatrix} \Delta\mathcal{V}_M \\ \Delta\bar{\mathcal{V}}^m \end{bmatrix} = \boldsymbol{T}^* \boldsymbol{T}_M^{-T} \mathcal{Q}_{ext} - \begin{bmatrix} \mathcal{F}_u^M \\ \bar{\mathcal{F}}_{opt}^m \end{bmatrix} \tag{5.175}$$

or

$$\boldsymbol{\mathcal{M}}_M^* \begin{bmatrix} \Delta\dot{\mathcal{V}}_M \\ \Delta\dot{\bar{\mathcal{V}}}^m \end{bmatrix} + \boldsymbol{\mathcal{C}}_M^* \begin{bmatrix} \Delta\mathcal{V}_M \\ \Delta\bar{\mathcal{V}}^m \end{bmatrix} + \begin{bmatrix} \mathcal{F}_u^M \\ \bar{\mathcal{F}}_u^m \end{bmatrix} = \boldsymbol{T}^* \boldsymbol{T}_M^{-T} \mathcal{Q}_{ext} - \begin{bmatrix} \mathbf{0} \\ (\bar{\mathcal{F}}^m)^{des} \end{bmatrix}, \tag{5.176}$$

where $\boldsymbol{T}^*$ is a transform that removes the middle (contact wrench) row from the equation of motion. Apparently, the difference term on the r.h.s. drives the error dynamics. Asymptotic stability in the sense of PD+ control [115] can be confirmed for the case of unconstrained motion, i.e. when the driving force is zero. Passivity can be confirmed for the regulation case, in accordance with [113].

The above controller has been implemented with a torque-controlled robot and shown to perform well under multicontacts, or when standing on soft ground and subjected to small external perturbations. Note that motion redundancy was not considered in the above derivations. In the implementation, the redundancy was resolved via a joint-space impedance component in the null space of the transposed Jacobian. Another problem is the assumption of nonsingular configurations. In [38], it was suggested to use the damped least-squares regularization method. This method leads to deteriorated tracking in the vicinity of a singular configuration, though (cf. also the discussion in Chapter 2).

5.13.3 Multicontact Motion/Force Control With Whole-Body Compliance

The multicontact balance controller described in Section 5.12.1 has the capability to react to weak disturbances. This is demonstrated with the same task (surface cleaning) as in the example in Section 5.12.1. While performing the task, the robot is subjected to two unexpected disturbances of 500 ms duration each. First, a pitch moment of magnitude 3 Nm is applied at the base link. This disturbance is accommodated by the torso rotation, as apparent from the

[16] Recall the assumption for a square Jacobian and, hence, no kinematic redundancy. This assumption will be dropped later on.

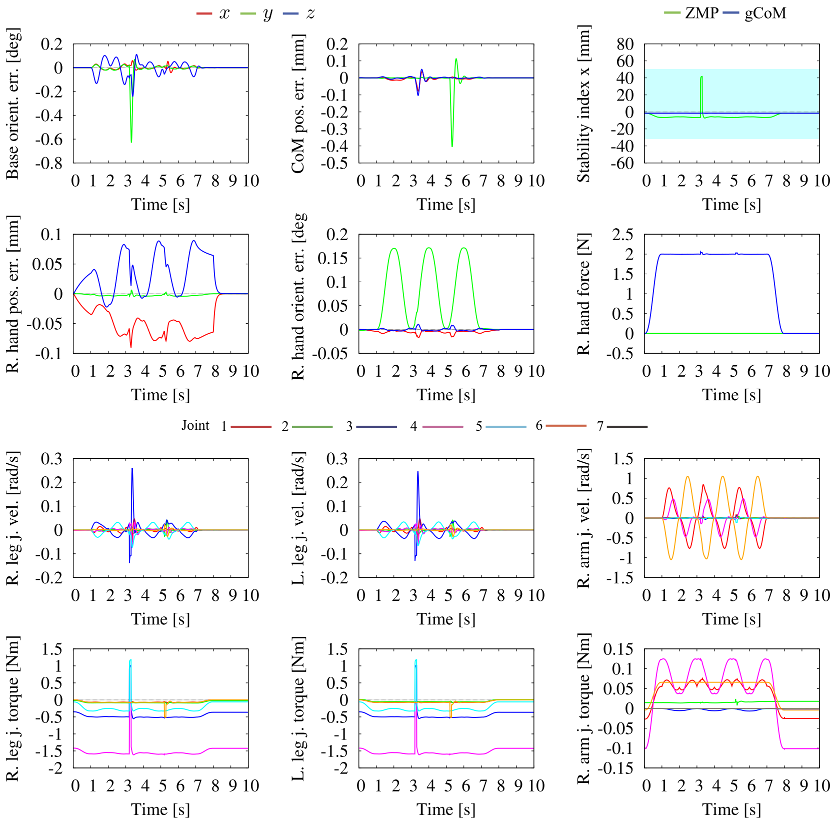


FIGURE 5.35 Multicontact motion/force control task as in Section 5.12.1. Two unexpected disturbances are applied; first, a pitch moment of magnitude 3 Nm at the base link and then, a lateral push of magnitude 5 N at the CoM. The disturbances are accommodated with a torso rotation and a CoM displacement, respectively. The deterioration of the hand motion/force task performance by the disturbances is negligible. R and L stand for “right” and “left,” respectively.

base orientation error graph in Fig. 5.35. Next, a lateral push of magnitude 5 N is applied at the CoM. This disturbance is accommodated via the CoM displacement, as apparent from the CoM position error graph and the joint velocity and torque graphs in the same figure. Note that the right-hand position/orientation errors and force differ insignificantly from those obtained without disturbance (cf. Fig. 5.31). This demonstrates the decoupling capability of the controller. The simulation result is shown in animated form in Video 5.13-1 [137].

5.14 ITERATIVE OPTIMIZATION IN BALANCE CONTROL

A number of balance control methods have been developed throughout the years that make use of iterative convex optimization techniques. The main advantage is that inequality constraints can be directly included into the optimization task. At present, off-the-shelf QP solvers are preferably used.

The general form of a QP task formulation that includes both equality- and inequality-type constraints is given in (2.57). In the case of balance control, the objective is to minimize a sum of appropriately defined linear-quadratic functions, in a form similar to that used with noniterative optimization (e.g. as in (5.130)). A variety of task formulations for balance control have been implemented so far. There are two main concerns when formulating the iterative optimization task. One of them is the computational load. In this aspect, it is preferable to formulate a few smaller tasks than a large one, as already mentioned in Section 5.9. This approach is referred to as *sequential* optimization [60]. Examples of such optimization can be found in [13,14,88]. Note that sequential optimization can include mixed iterative/noniterative minimization tasks [142,167,38,37]. In this way, the computational burden can be further decreased due to the inherent time efficiency of the noniterative tasks.

The other main concern is the lack of a solution because of the presence of multiple contradicting constraints. There are three possible approaches to avoid such overconstraints. First, an equality-type constraint relaxation could be employed, as discussed in Section 5.9.2. Second, the so-called *slack variables* could be employed (cf. Section 2.60). The role of these variables is also a constraint relaxation, especially in the case of inequality-type constraints. Recall (from (2.61)) that the variables themselves are subjected to the minimization. Third, an order of priority could be introduced among the constraints. When compared to weight tuning, the prioritization approach yields the advantage of decoupled minimization tasks. In this way, stability can be increased, as reported in [18]. The prioritization approach, however, has also some drawbacks. Fixing the priorities may not always guarantee good results. Introducing variable priorities, on the other hand, is not that straightforward either (cf. the discussion in Section 2.8.3).

As a consequence of the ambiguity in weight tuning/priority assignment, it is difficult to prove the performance stability of an iterative optimization method. The tuning process could be cumbersome and special care could be needed to eventually obtain satisfactory performance. It should be noted, though, that despite this problem a number of studies have been based on the iterative optimization. Representative examples will be given in what follows.

5.14.1 A Brief Historical Overview

One of the first balance controllers based on iterative optimization was introduced in [151, 65] for a proactive single-leg balancing task. The objective function is formulated in terms of the joint angle increments. The joint-angle input trajectory is modified based on the result of the optimization. The optimization makes use of constraints $\mathcal{F}_{JT}$, $\mathcal{F}_{CoP}$, and $\mathcal{M}_{SM}$. Two components from $\mathcal{M}_{SM}$ are used to restrain the gCoM motion. Note that this constraint is quite conservative—it restricts the solution space of the optimizer excessively by not allowing the gCoM to move within the BoS. The three angular momentum components in $\mathcal{M}_{SM}$ are subjected to inequality-type constraints to avoid excessive accumulation of angular momentum.

The balance controller in [81] makes use of two types of control laws to ensure an appropriate reactive behavior in response to external disturbances applied while standing in a symmetric double-stance posture. In the case of a relatively small disturbance, a PD feedback control law is preferably used. The advantage of this controller is its suitability for real-time implementation. This controller, however, is inadequate in the case of larger disturbances. Such disturbances are dealt with an QP optimization controller. The objective function minimizes the cost of the joint acceleration, $c_{JA}|_{\ddot{\boldsymbol{\theta}}^{ref}=0}$, and maximizes the CoM acceleration in the lateral direction, to avoid falls. The constraints are $\mathcal{F}_{BoS}$ and $\mathcal{M}_{AL}$, along with a PD-type restraint of gCoM motion, with components from $\mathcal{F}_{SM}$. There is also a specific constraint that makes use of symmetry relations within the closed-loop chain formed by the double stance. Human-like balance motion patterns can be achieved with this balance controller, including the use of an arm ("windmilling") motion.

The balance controller described in [144] is also based on the QP approach. The input trajectories are specified in terms of the joint angles, CoM/ZMP motion, and the vertical component of the GRF, in a consistent way. The task of the optimizer is to ensure stabilization by taking into account the ZMP/gCoM dynamics when reacting to relatively small, short-term disturbances. The resulting long-term deviation from the CoM input trajectory is compensated in a feedback manner with PD feedback gains. Effectively, the method can be regarded as an indirect ZMP manipulation approach. The joint variables are obtained from the output of the QP optimizer that uses the CoM inverse kinematics and other kinematic subtasks as constraints.

5.14.2 SOCP-Based Optimization

A family of balance control algorithms can be formulated as *second-order cone programming* (SOCP) problems, i.e. as a special class of convex optimization problems. Recall that the SOCP formulation (cf. (3.79)) is especially suitable when enforcing the friction cone constraints. In [117], examples of three convex optimization subtasks are given. First, the CoP-in-BoS constraint is defined as a convex minimization problem by requiring that $\|\boldsymbol{r}_p - \boldsymbol{r}_p^{des}\| \leq \epsilon_p$, ϵ_p being a user-defined constant threshold. Second, the gCoM-in-BoS constraint is defined in the form of a linear inequality obtained from the finite-difference approximation of the current gCoM. Third, the closed-loop kinematic relations are enforced as equality-type constraints. In this way, a dynamic environment (e.g. translating and/or rotating support) can be included into the model. The optimizer performed well in both proactive (a kicking motion) and reactive (support displacement) balance control tasks.

The SOCP approach seems quite promising. Different types of constraints can be mixed: nonlinear and linear, as well as equality- and inequality-type ones. There are problematic issues, though, similar to those pertinent to the QP solvers. The most important is the difficulty in proving the stability of the balance controller. There are a number of parameters that need to be set in a consistent way. The sequential approach above provides a nice solution. Other parameters, notably those related to the feedback-type controller performance (e.g. the ϵ_p error threshold in the first SOCP controller in this section and the PD gains for the CoM regulator in Section 5.14.1) can only be tuned in an intuitive way. Furthermore, the real-time implementation of an SOCP controller is still problematic. This problem is expected to be alleviated in the near future with the development of faster SOCP algorithms [6], in combination with high-performance computational hardware.

5.14.3 Iterative Contact Wrench Optimization

The contact wrench optimization problem can be cast as a QP task, with solution $\boldsymbol{x} = \overline{\mathcal{F}}^c$. The friction cone and the CoP inequality-type constraints can be directly included into the task formulation. The optimal contact wrench components, $\overline{\mathcal{F}}^c_{opt}$, will determine the best location of the CoPs together with appropriate reaction forces and a vertical moment at each foot. In this way, a foot slip, transitions from plane to line or point contacts via a foot roll and/or pitch rotations as well as torsional foot (yaw) rotation can be avoided.

A reasonable objective is the minimization of the ankle torques. The motivation behind this objective is threefold: (1) energy efficiency,[17] (2) minimization of the performance requirements on the ankle actuators [158], and (3) avoidance of foot roll [88]. A "generic" formulation of the problem is obtained from (5.115) as [158]

$$\min_{\boldsymbol{x}} \sum_{j\in\{r,l\}} \|\bar{\boldsymbol{\tau}}^c_j - (\boldsymbol{r}_{A_j} - \boldsymbol{r}_{P_j}) \times \bar{\boldsymbol{f}}^c_j\|^2. \tag{5.177}$$

The constraints are $\mathcal{F}_{SD}$, $\mathcal{F}_{FC}$, and $\mathcal{F}_{BoS}$, as defined in Section 5.9.2. On flat ground, an additional constraint should be included to restrict the vertical coordinate of the CoP to zero.

There are two problems associated with the above optimization. First, the formulation is *nonconvex* due to the vector cross-product. Second, there is no guarantee that the CoPs will be located away from the boundaries of the BoS, to avoid a foot roll. The first problem can be tackled by resorting to a point contact model, similar to that described in Section 5.10.1; see also [50,2,63,168]. In this case, the vector cross-product is determined from the position vectors of the point contacts. These are assumed constant during the optimization time interval. Thus a skew-symmetric matrix representation of the vector cross-product can be employed. The objective function can then be represented as a semidefinite quadratic form that renders the optimization problem convex, i.e.

$$\min_{\boldsymbol{x}} c_{CW}, \tag{5.178}$$

with c_{CW} determined as the relaxation of $\mathcal{F}_{CW}|_{(\bar{\mathcal{F}}^c)^{ref}=0}$. The same constraints as above are used. Recall that when the weight matrix $\boldsymbol{W}_{CW}$ is defined as in (5.133), the objective function assumes the form $c_{CW} = \|\boldsymbol{\tau}^c\|^2_{W_\kappa}$. This implies that the contact wrenches can be minimized implicitly, via the joint torque.

The second problem can be tackled by amending the objective function with the relaxation of the CoP tracking constraint:

$$\mathbb{C}_P \bar{\boldsymbol{\tau}}^c_t = \boldsymbol{r}^{ref}_P \tag{5.179}$$

to obtain

$$\min_{\boldsymbol{x}} c_{CW} + c_{CoP}. \tag{5.180}$$

[17] The equivalent of the metabolic cost/energy [3,15] in the field of biomechanics.

With (5.179), the tangential contact moments $\bar{\boldsymbol{\tau}}_t^c$ are mapped via the $\mathbb{C}_P$ to obtain the current CoPs. The current CoPs are then penalized via the c_{CoP} at the desired CoPs, $\boldsymbol{r}_P^{ref}$. The usual, though somewhat conservative setting for $\boldsymbol{r}_P^{ref}$ is at the ground projection of the ankle joint. In this case, there is no need to use the CoP-in-BoS constraint, $\mathcal{F}_{BoS}$, anymore. There are also other choices for c_{CoP} that can avoid the conservative setting problem, e.g. to more directly penalize possible transitions to edge contacts [158].

In some cases, there may be no solution to the optimization task at all. This problem can be alleviated via the relaxation of the strict spatial dynamics equality constraint $\mathcal{F}_{SD}$ [142]. We have

$$\min_{\boldsymbol{x}} c_{CW} + c_{SD}. \tag{5.181}$$

With this formulation, there will always be a solution but the acceleration component in $\mathcal{F}_C^{ref}$ (i.e. the rate of change of the spatial momentum) cannot be guaranteed anymore.

5.14.4 Iterative Spatial Dynamics Optimization

It is possible to employ an iterative approach to optimize the contact wrenches *together with* the joint accelerations appearing in the spatial dynamics equation, (5.109), with the solution $\boldsymbol{x} = \begin{bmatrix} \ddot{\boldsymbol{\theta}}^T & (\bar{\mathcal{F}}^c)^T \end{bmatrix}^T$. Examples of sequential as well as nonsequential optimization task formulations will be presented below.

Sequential Approach

The following objective function is an example of a spatial dynamics–based optimization [88]:

$$\min_{\boldsymbol{x}} \sum_{j \in \{r,l\}} \left(\|\bar{\boldsymbol{f}}_j^c\|_{W_f}^2 + \|\bar{\boldsymbol{\tau}}_j^c\|_{W_\tau}^2 \right) + \|\dot{\boldsymbol{p}}(\boldsymbol{f}^c) - \dot{\boldsymbol{p}}^{ref}\|_{W_p}^2 + \|\dot{\boldsymbol{l}}_C(\bar{\mathcal{F}}^c) - \dot{\boldsymbol{l}}_C^{ref}\|_{W_l}^2. \tag{5.182}$$

There are four subtasks. The first two minimize the contact wrench components, i.e. the GRFs and the GRMs (or equivalently, the CoP location/ankle torques). The last two minimize the errors in the linear and angular momentum rate components. It is assumed that the reference momentum rate components, $\dot{\boldsymbol{p}}^{ref}$ and $\dot{\boldsymbol{l}}_C^{ref}$, are specified in an independent way, via a proportional-derivative CoM motion and proportional angular momentum feedback control, respectively. This poses a problem, however, since the two momentum rate components are coupled, as noted in Section 5.8.1. This is the reason why the above objective attempts to minimize the two components in an independent way. With regard to the GRF and GRM minimization subtasks, it should be noted that the former restricts the application field of the method since the foot loading/unloading will be rendered impossible, while the latter will render the formulation nonconvex, because of the GRF cross-product terms (cf. (5.115)).

To alleviate the inherent problems in the above optimization, the optimization task is reformulated as follows. A possible solution to the nonconvexity problem was presented in Section 5.14.3. Thereby, a contact model that approximates the plane contact with multipoint contacts was used. Here, another possibility will be explored that avoids the use of a multipoint contact model. To this end, the objective function will be reformulated to yield a sequential problem: first minimize the GRFs and then the GRMs [88]. The GRFs are determined by the following minimization task:

$$\min_{\boldsymbol{x}} \quad \|\dot{\boldsymbol{p}}(\boldsymbol{f}^c) - \dot{\boldsymbol{p}}^{ref}\|^2 + \|\dot{\boldsymbol{l}}_C(\bar{\mathcal{F}}^c) - \dot{\boldsymbol{l}}_C^{ref}\|^2_{W_l} + \sum_{j \in \{r,l\}} \|\bar{\boldsymbol{f}}_j^c\|^2_{W_f} \tag{5.183}$$

under the $\mathcal{F}_{FC}$ constraints. Note that unit weights are used for the linear momentum rate error subtask. In this way, the subtask is rendered high-priority. On the other hand, note that the linear momentum rate depends on the GRF. In order to avoid interference with the GRF minimization subtask (the third term), the weights should be set sufficiently small. In other words, the priority of the GRF minimization subtask should be set at the lowest level. Next, note that very small (approximately zero) angular momentum rate errors will yield very small ankle torques. Nevertheless, the angular momentum rate errors cannot be ignored in reality. This means that appropriate GRM/ankle torques have to be determined that will minimize the errors. Since the optimal GRFs are already known, the GRMs can be expressed with the linear relation:

$$\bar{\boldsymbol{\tau}}_j^c = \left[(\bar{\boldsymbol{f}}^c_{j\ opt})^\times\right](\boldsymbol{r}_{P_j} - \boldsymbol{r}_{A_j}) + \boldsymbol{e}_z m_z. \tag{5.184}$$

Here $\boldsymbol{e}_z m_z$ accounts for the reaction torque around the normal. This equation renders the GRM minimization (or CoP allocation) problem convex. It is then straightforward to design an appropriate objective function [88].

The contact wrenches (the GRFs and GRMs) resulting from the above two sequential optimization subtasks are then used to obtain the *admissible* rate of change of the spatial momentum, $\dot{\mathcal{L}}_C^{ref}$. The final optimization subtask in the cascade can then be invoked, i.e.

$$\min_{\ddot{\boldsymbol{\theta}}} c_{SM} + c_{JA}|_{\ddot{\boldsymbol{\theta}}^{ref}=0}. \tag{5.185}$$

The c_{SM} objective stems from the relaxation of the $\mathcal{F}_{SM}$ equality constraint. Apparently, this final optimization subtask falls also within the motion constraint domain. For example, as shown in [112], this objective can be used to maintain the initial posture of the upper body while standing on a narrow support, in the presence of disturbances.

Nonsequential Approach

A nonsequential spatial dynamics–based iterative optimization approach has been explored in [77,161]:

$$\min_{\boldsymbol{x}} \left(\ c_{CW} + c_{CoP} + c_{SM} + c_{EL} + c_{JA}|_{\ddot{\boldsymbol{\theta}}^{ref}=\mathbf{0}} \right). \tag{5.186}$$

The force constraints are the equality-type $\mathcal{F}_{SD}$ and the friction cone constraint $\mathcal{F}_{FC}$. The inequality-type motion constraint, $\mathcal{M}_{AL}$, is used to enforce the joint limits. Since all constraints are formulated as components of the objective function, problems related to an overconstrained system can be avoided. This, however, is a compromise that leads to the problem of weight adjustment. As already mentioned, such an adjustment can be done, but only in an intuitive, nonrigorous way.

5.14.5 Complete Dynamics–Based Optimization

A number of multiobjective QP methods based on complete dynamics formulations have been developed in the past decade, mainly for the purpose of animating articulated figures, including humans [2,13,14,141,92,100,18,95,96]. In these works, the main aim is to generate a "realistically looking" *dynamic motion*. This goal cannot be achieved with kinematics- or quasistatics-based approaches. On the other hand, complete dynamics–based formulations in the field of humanoid robotics are still quite rare. There are a number of reasons for that. First of all, note that such formulations are useful for torque-controlled robots. Only a few such robots exist at present. The formulations are based on dynamic models that cannot be obtained in a straightforward manner. Indeed, the dynamic parameters from CAD data are imprecise since components like motors, wiring, connectors, and boards are not included. Thus, cumbersome identification procedures have to be used to obtain a precise dynamic model. Furthermore, dynamic models impose a relatively heavy computational burden. Therefore, it is quite challenging to ensure feedback control in real-time (typically, within 1 ms). Performing experiments with a torque-controlled robot is also quite demanding. Nevertheless, these hurdles can be overcome as will become apparent in what follows.

The optimal solution to the optimization task, based on the complete dynamics, is represented in the form of (5.113). The optimization task is formulated as a multiobjective function composed of penalty-type terms with or without a hierarchical structure for task prioritization.

Hierarchical Multiobjective Optimization With Hard Constraints

The problem of an overconstrained system can be tackled by introducing a hierarchical structure among the tasks. This approach, known as "hierarchical QP" (HQP), was discussed in Section 2.8.3. Recall also that a hierarchical structure can lessen the computational burden to some extent. The hierarchy defines the fixed task priorities. The constraints that need to be strictly enforced (hard constraints) are inequality-type, physical constraints. They include the joint angle and torque limits ($\mathcal{M}_{AL}$ and $\mathcal{F}_{TL}$), the friction constraints ($\mathcal{F}_{FC}$ and $\mathcal{F}_{FT}$), and the CoP-in-BoS constraint, $\mathcal{F}_{BoS}$. The notation pertinent to the general QP task formulation, (2.57), will be used below. The form of the solution vector is given in (5.113). The optimization task can be written as [40]

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{v},\boldsymbol{w}} \ & \|\boldsymbol{v}\|^2 + \|\boldsymbol{w}\|^2 \\ \text{s.t.} \quad & \boldsymbol{W}(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{a}) = \boldsymbol{w}, \\ & \boldsymbol{V}(\boldsymbol{B}\boldsymbol{x}+\boldsymbol{b}) \le \boldsymbol{v}. \end{aligned} \tag{5.187}$$

Vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ denote the slack variables that are used for constraint relaxation to avoid numerical instabilities. Recall that these variables are determined as an outcome from the optimization process; they do not need to be specified by the user. On the other hand, matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ determining the relative weights for the subtasks are used-specified. Note that the above formulation is penalty-type. Strict task prioritization can be enforced with the help of the method introduced in Section 2.8.3 (see also [17,18]). First, denote one particular optimal solution at priority level r as $(\boldsymbol{x}_r^*,\ \boldsymbol{v}_r^*)$. All optimal solutions can then be expressed as

$$\boldsymbol{x} = \boldsymbol{x}_r^* + \boldsymbol{N}_r \boldsymbol{p}_{r+1} \tag{5.188}$$

$$\begin{aligned}&\overline{\boldsymbol{A}}_r\boldsymbol{x} + \bar{\boldsymbol{a}}_r \leq \boldsymbol{v}_r^*,\\ &\dots\\ &\overline{\boldsymbol{A}}_1\boldsymbol{x} + \bar{\boldsymbol{a}}_1 \leq \boldsymbol{v}_1^*.\end{aligned} \tag{5.189}$$

Here $\boldsymbol{N}_r$ is a projector onto the compound null space of all higher-priority equality-type constraints, $\mathcal{N}(\overline{\boldsymbol{B}}_r) \cap \dots \cap \mathcal{N}(\overline{\boldsymbol{B}}_1)$. Vector $\boldsymbol{p}_{r+1}$ is arbitrary; it parametrizes the compound null space. The overbar notation stands for a quantity restricted by the respective weight matrix, s.t. $\overline{\boldsymbol{A}} = \boldsymbol{V}\boldsymbol{A}$, $\bar{\boldsymbol{a}} = \boldsymbol{V}\boldsymbol{a}$, $\overline{\boldsymbol{B}} = \boldsymbol{W}\boldsymbol{B}$, and $\bar{\boldsymbol{b}} = \boldsymbol{W}\boldsymbol{b}$. The null-space projectors can be computed via the singular-value decomposition SVD (cf. Section 2.7.1). The SVD procedure can be invoked in parallel to the QP task at level $r-1$. In the implementation reported in [40] it was observed that the SVD thus adds only an insignificant computational overhead.

With the above notation, the minimization task at level $r+1$ can be written as

$$\begin{aligned}\min_{\boldsymbol{v}_{r+1},\, \boldsymbol{p}_{r+1}} \quad & \|\boldsymbol{v}_{r+1}\|^2 + \|\overline{\boldsymbol{B}}_{r+1}(\boldsymbol{x}_r^* + \boldsymbol{N}_r\boldsymbol{p}_{r+1}) + \bar{\boldsymbol{b}}_{r+1}\|^2\\ \text{s.t.} \quad & \overline{\boldsymbol{A}}_{r+1}(\boldsymbol{x}_r^* + \boldsymbol{N}_r\boldsymbol{p}_{r+1}) + \bar{\boldsymbol{a}}_{r+1} \leq \boldsymbol{v}_{r+1},\\ & \overline{\boldsymbol{A}}_r(\boldsymbol{x}_r^* + \boldsymbol{N}_r\boldsymbol{p}_{r+1}) + \bar{\boldsymbol{a}}_r \leq \boldsymbol{v}_r^*,\\ & \dots\\ & \overline{\boldsymbol{A}}_1(\boldsymbol{x}_r^* + \boldsymbol{N}_r\boldsymbol{p}_{r+1}) + \bar{\boldsymbol{a}}_1 \leq \boldsymbol{v}_1^*.\end{aligned} \tag{5.190}$$

Apparently, the priority solutions are obtained in a recursive fashion. With this formulation it can be guaranteed that the optimal solution will be found.

The above optimization method has been implemented with a torque-controlled humanoid robot of 25 DoFs [40]. Five priority levels were introduced that cover a large number of the constraints listed in Section 5.9.2. In the experiments, only 14 DoFs (no upper body) were used. With the complete dynamics formulation, about 5 ms were needed on an Intel Core i7-2600 CPU with 3.4 GHz. With the 14-DoF model, the $\mathcal{F}_{EoM}$ equality-type constraint was excluded. In this case, real-time performance (under 1 ms) has been achieved. Satisfactory performance has been reported for both proactive- and reactive-type tasks, despite the presence of the dynamic model inaccuracies.

Penalty-Based Multiobjective Optimization With Soft Constraints

Soft constraints always provide a solution, without numerical instabilities that sometimes may occur when using hard constraints. A penalty-type formulation can be employed to derive the optimal solution from the reference total force acting at the CoM, $\boldsymbol{f}_C^{ref}$ [22]. This force is obtained as the output of the DCM stabilizer in Section 5.6.3. The formulation is

$$\min_{\boldsymbol{x}} \|\mathbb{C}_{cC_f}\overline{\mathcal{F}}^c - \boldsymbol{f}_C^{ref}\|_{W_c} + \|\Delta\boldsymbol{r}(\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \overline{\mathcal{F}}^c(\boldsymbol{C}_\oplus))|_{W_r}. \tag{5.191}$$

The relevant constraints are: $\mathcal{F}_{EoM}$, $\mathcal{F}_{FC}$, $\mathcal{F}_{TL}$, and $\mathcal{M}_{CL}$ (cf. Tables 5.1 and 5.2). The friction cone constraints are approximated by pyramids (cf. Section 3.1; see also e.g. [88]), vector $\boldsymbol{C}_\oplus$ being composed of stacked $\boldsymbol{C}_k$ matrices (cf. (3.12)) that collect the pyramid basis vectors. Note that $\boldsymbol{C}_\oplus$ parametrizes the contact wrenches. The two objectives in (5.191) minimize: (i) the

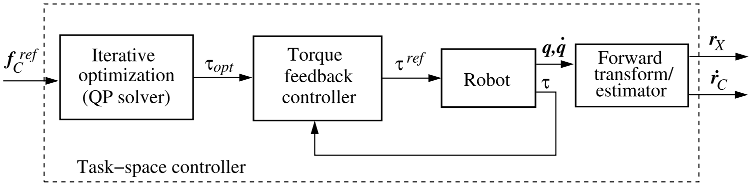


FIGURE 5.36 Block diagram of a task-space controller for the DCM stabilizer in Section 5.6.3. The task-space controller is designed for implementation with a torque-controlled robot. The QP solver uses the complete robot dynamics as a minimization constraint.

error of the total force acting at the CoM and (ii) the motion control errors along the unconstrained motion directions of the end links, as well as the self-motion control errors (the $\Delta \boldsymbol{r}$ term). The QP solver is part of the task-space controller appearing in the DCM-based balance controller in Fig. 5.15. The components of the task-space controller are shown in Fig. 5.36. Apparently, the QP solver determines the joint torque $\boldsymbol{\tau}_{opt}$ as a control input for the local joint-space controller.

Inverse Dynamics Plus Inverse Kinematics–Based Optimization Approach

A dynamic model of sufficient precision may not always be available. To cope with problems stemming from an imperfect dynamic model, one could use the information inherent to the (precise) inverse kinematics model. An inverse dynamics plus inverse kinematics–based optimization approach seems to be especially suitable for hydraulically driven humanoid robots with mixed force/velocity input commands [40,25]. In the implementation discussed in [25], two QP problems for the inverse dynamics and inverse kinematics are solved at each time step. The inverse dynamics solver ($\boldsymbol{x}$ as in (5.113)) is updated with the sensor states. The inverse kinematics solver ($\boldsymbol{x} = \dot{\boldsymbol{q}}_{int}$), on the other hand, is updated with states obtained via an integration, to avoid instabilities. This leads to accumulation of errors in the contact positions, though. The problem can be alleviated by employing a "leaky" integrator. The objective function (soft constraints), derived from the linear form $\boldsymbol{W}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{a})$, contains rows of mixed dynamic and kinematic objectives. The dynamic objectives stem from the relaxation of the $\mathcal{F}_{CoP}$ and $\mathcal{M}_{EL}$ constraints, as well as from additional dynamic objectives that include: (1) state tracking ($\boldsymbol{x} = \boldsymbol{x}^{ref}$), (2) weight distribution in accordance with (5.116) (only in the z-direction), and (3) suppression of sudden changes in the torque commands. The hard constraints used are: the equation of motion, the friction cone, and the CoP-in-BoS ($\mathcal{F}_{EoM}$, $\mathcal{F}_{FC}$, and $\mathcal{F}_{BoS}$, respectively). The kinematic objectives included are: (1) the end-link velocity tracking, (2) the direct joint-velocity tracking ($\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\theta}}^{ref}$), and (3) the suppression of sudden changes in the velocity commands. Further implementation details can be found in [25].

5.14.6 Mixed Iterative/Noniterative Optimization Approaches

One possibility to reduce the high computational burden inherent to iterative optimization based on the complete dynamics is to invoke a two-step sequential optimization scheme with mixed, iterative/noniterative optimization. The iterative part consists of spatial dynamics–based contact wrench optimization under specific constraints. The noniterative part solves the complete inverse dynamics problem via a suitable generalized inverse. In [142], for example, the iterative optimization task is defined as

$$\min_{\boldsymbol{x}} c_{SD} + c_{CW}|_{(\bar{\mathcal{F}}^c)^{ref}=0},$$

where $\boldsymbol{x} = \overline{\mathcal{F}}^c$. This task is solved under the inequality-type constraints $\mathcal{F}_{FC}$ and $\mathcal{F}_{BoS}$. Then, the equation of motion, $\mathcal{F}_{EoM}$, and the closed-chain motion constraint, $\mathcal{M}_{CC}$, are combined into one equation. We have

$$\begin{bmatrix} \boldsymbol{M}(\boldsymbol{q}) & -\boldsymbol{S} \\ \boldsymbol{J}_c & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{c}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_c^T \overline{\mathcal{F}}_{opt}^c \\ -\dot{\boldsymbol{J}}_c \dot{\boldsymbol{q}} \end{bmatrix}. \tag{5.192}$$

Here $\overline{\mathcal{F}}_{opt}^c$ are the contact wrenches obtained from the iterative optimization task. This equation is solved via a regularized (left) pseudoinverse for the joint acceleration and torque. Simple reactive tasks were examined in [142], with a torque-controlled robot (the Sarcos humanoid robot). The spatial dynamics objective was used mainly for the CoM PD feedback control. The desired rate of change of the angular momentum was set in a conservative way, to maintain the initial torso orientation while the robot was pushed on the back. Note however, that this approach does not allow for a sufficiently compliant response to the disturbance, e.g. to obtain a sufficiently large torso rotation as required under the hip strategy.

Using Generalized Acceleration Input Data Obtained From a Motion Capture System

Another example of a mixed, iterative contact wrench/noniterative complete dynamics optimization approach is reported in [167]. Desired generalized accelerations, $\ddot{\boldsymbol{q}}^{des}(t)$, obtained from a motion capture system, are used as the inputs. The iterative task is formulated as a penalty-type relaxation of the spatial dynamics equation expressed in terms of the base quasivelocity as follows:

$$\min_{\boldsymbol{f}} c_{SD} = \|\boldsymbol{\mathcal{A}}_{\overleftarrow{CB}} \ddot{\boldsymbol{q}}_B^{des} + \mathcal{C}_B - \mathbb{C}_{cB_f} \boldsymbol{f}\|^2,$$

subjected to the $\mathcal{F}_{FC}$ constraints. A point-contact model is used, the rectangular feet comprising four point contacts at the vertices. Vector $\boldsymbol{f}$ collects all reactions. The optimal $\boldsymbol{f}_{opt}$ is then used to derive optimal contact wrenches, $\overline{\mathcal{F}}_{opt}^c$, in a straightforward manner. Furthermore, the noniterative minimization task is defined as

$$\min_{\boldsymbol{\tau}} c_{JA} + c_{JT}|_{\boldsymbol{\tau}^{ref}=0},$$

subject to the complete dynamics equality constraint, $\mathcal{F}_{EoM}(\overline{\mathcal{F}}_{opt}^c)$, and the end-link motion equality constraints $\mathcal{M}_{EL}$. This task is solved via an appropriately defined generalized inverse.

Hierarchical Task Formulation With Decoupling

The mixed, iterative contact wrench/noniterative complete dynamics optimization approach can be further improved by introducing a hierarchical structure for the noniterative part, via the null-space projectors [38,37]. To this end, first rewrite the complete dynamics representation in mixed quasivelocity, as in (4.158), in the following quasistatic form:

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{\tau} \end{bmatrix} = \begin{bmatrix} \boldsymbol{g}_f \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} - \mathbb{J}^T\mathbb{F}. \tag{5.193}$$

The second term on the r.h.s. encodes the hierarchical structure [37]; we have

$$\mathbb{J}^T = \begin{bmatrix} \mathbb{J}_f^T \\ \mathbb{J}_m^T \\ \mathbb{J}_\tau^T \end{bmatrix} \equiv \begin{bmatrix} \boldsymbol{J}_c^T & \boldsymbol{N}_2^*\boldsymbol{J}_m^T & \boldsymbol{N}_3^*\boldsymbol{J}_q^T & \boldsymbol{N}_4^*\boldsymbol{S}^T \end{bmatrix}, \quad \mathbb{F} \equiv \begin{bmatrix} \overline{\mathcal{F}}^c \\ \overline{\mathcal{F}}^m \\ \mathcal{F}_C \\ \boldsymbol{\tau}_a \end{bmatrix}. \tag{5.194}$$

The $\boldsymbol{N}_{(\circ)}^* \in \Re^{(n+6)\times(n+6)}$ matrices denote the null-space projectors defined in Section 4.5.3. Their subscripts signify the priority level. It is important to emphasize that these null-space projectors ensure decoupling and thus guarantee the *asymptotic control stability* of the hierarchical system. The constraint ($\boldsymbol{J}_c$), mobility ($\boldsymbol{J}_m$), and spatial dynamics ($\boldsymbol{J}_q$) Jacobians are defined in (2.96), (2.95), and (4.99), respectively. The underactuation filtering matrix, $\boldsymbol{S}$, is defined in (2.102). The highest priority is assigned to the physical (contact) constraints. Motion in the mobility directions has higher priority than the CRB motion (used for balance control and to ensure a compliant whole-body response to an external disturbance). In this way, the object handling is not influenced by the CRB motion. The lowest priority is assigned to the posture control (e.g. in the form of a joint impedance) via the arbitrary torque vector $\boldsymbol{\tau}_a$. Furthermore, from the above two equations it is apparent that the following relations must hold:

$$\boldsymbol{g}_f - \mathbb{J}_f^T\mathbb{F} = \mathbf{0} = \mathbb{J}_m^T\mathbb{F}. \tag{5.195}$$

The iterative wrench distribution problem is then formulated simply as $\min_{\boldsymbol{x}} c_{CW}$, $\boldsymbol{x} = \overline{\mathcal{F}}^c$, under the constraints $\mathcal{F}_{FC}$, $\mathcal{F}_{FT}$, $\mathcal{F}_{BoS}$, $\mathcal{F}_{TL}$, and (5.195). The soft constraints c_{CW} require reference $(\overline{\mathcal{F}}^c)^{ref}$ to be specified as an input by a planner. The optimal contact wrenches are inserted, together with appropriate reference values for the rest of the $\mathbb{F}$ components, into the lower part of (5.193) to obtain the control torque as

$$\boldsymbol{\tau} = -\mathbb{J}_\tau^T\mathbb{F}.$$

It is important to note that no external force measurement is needed with this approach (cf. Section 4.5.3).

5.14.7 Computational Time Requirements

A comparison of the maximum computational times obtained with some of the iterative optimization approaches discussed in this chapter is given below. The following computational environment was used: a desktop PC with Intel Core i7 340 GHz CPU and 8 GB RAM,

TABLE 5.3 Comparison of the maximum computational times for sequential optimization

	Optimization			
	WD [μs]	**FSD [μs]**	**JSID [μs]**	**Total [μs]**
sequential WD → FSD	49	212	–	261
sequential WD → JSID	49	–	873	922

running Linux with the Real-Time PREEMPT patch. QuadProg++ [19] was used as the iterative solver.

The computational time requirements for the two sequential optimization approaches (cf. Fig. 5.22) are given in Table 5.3. In the case of nonsequential optimization, the formulation based on the FSD required 431 μs. The optimization based on the complete dynamics (CD) required 1728 μs, with variations of about ± 100 μs depending on the constraints.

For comparison, the two noniterative optimization approaches, DCM-GI-based WD and FSD optimization, required 1.62 μs and 2.12 μs, respectively.

References

[1] M. Abdallah, A. Goswami, A biomechanically motivated two-phase strategy for biped upright balance control, in: IEEE Int. Conf. on Robotics and Automation, Barcelona, Spain, 2005, pp. 1996–2001.

[2] Y. Abe, M. Da Silva, J. Popović, Multiobjective control with frictional contacts, in: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2007, pp. 249–258.

[3] P.G. Adamczyk, S.H. Collins, A.D. Kuo, The advantages of a rolling foot in human walking, Journal of Experimental Biology 209 (2006) 3953–3963.

[4] Y. Aoustin, A.M. Formal'skii, On optimal swinging of the biped arms, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 2922–2927.

[5] J.P. Aubin, Viability Theory, Birkhäuser, Basel, 2009.

[6] S.P. Boyd, B. Wegbreit, Fast computation of optimal contact forces, IEEE Transactions on Robotics 23 (2007) 1117–1132.

[7] T. Bretl, S. Lall, Testing static equilibrium for legged robots, IEEE Transactions on Robotics 24 (2008) 794–807.

[8] S. Caron, Q. Cuong Pham, Y. Nakamura, Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics, in: Robotics: Science and Systems XI. Robotics: Science and Systems Foundation, Rome, Italy, 2015, pp. 28–36.

[9] S. Caron, Q.-C. Pham, Y. Nakamura, ZMP support areas for multicontact mobility under frictional constraints, IEEE Transactions on Robotics 33 (2017) 67–80.

[10] Y. Choi, D. Kim, Y. Oh, B.-J. You, Posture/walking control for humanoid robot based on kinematic resolution of CoM Jacobian with embedded motion, IEEE Transactions on Robotics 23 (2007) 1285–1293.

[11] Y. Choi, D. Kim, B.-J. You, On the walking control for humanoid robot based on the kinematic resolution of CoM Jacobian with embedded motion, in: IEEE International Conference on Robotics and Automation, 2006, pp. 2655–2660.

[12] Y. Choi, B.-j. You, S.-r. Oh, On the stability of indirect ZMP controller for biped robot systems, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2004, pp. 1966–1971.

[13] C. Collette, A. Micaelli, C. Andriot, P. Lemerle, Dynamic balance control of humanoids for multiple grasps and non coplanar frictional contacts, in: IEEE-RAS International Conference on Humanoid Robots, IEEE, 2007, pp. 81–88.

[14] C. Collette, A. Micaelli, C. Andriot, P. Lemerle, Robust balance optimization control of humanoid robots with multiple non coplanar grasps and frictional contacts, in: IEEE International Conference on Robotics and Automation, 2008, pp. 3187–3193.

[15] S.H. Collins, P.G. Adamczyk, A.D. Kuo, Dynamic arm swinging in human walking, Proceedings. Biological Sciences/The Royal Society 276 (2009) 3679–3688, arXiv:1010.2247.

[16] H. Dai, A. Valenzuela, R. Tedrake, Whole-body motion planning with centroidal dynamics and full kinematics, in: IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 2014, pp. 295–302.

[17] M. De Lasa, A. Hertzmann, Prioritized optimization for task-space control, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 5755–5762.

[18] M. De Lasa, I. Mordatch, A. Hertzmann, Feature-based locomotion controllers, ACM Transactions on Graphics 29 (2010) 131.

[19] L. Di Gaspero, QuadProg++: a C++ library for quadratic programming which implements the Goldfarb-Idnani active-set dual method, https://github.com/liuq/QuadProgpp, 2016.

[20] R.C. Dorf, R.H. Bishop, Modern Control Systems, 13 edition, Pearson, 2017.

[21] J. Englsberger, C. Ott, A. Albu-Schaffer, Three-dimensional bipedal walking control using divergent component of motion, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 2600–2607.

[22] J. Englsberger, C. Ott, A. Albu-Schaffer, Three-dimensional bipedal walking control based on divergent component of motion, IEEE Transactions on Robotics 31 (2015) 355–368.

[23] J. Englsberger, C. Ott, M.A. Roa, A. Albu-Schaffer, G. Hirzinger, Bipedal walking control based on capture point dynamics, in: IEEE International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 4420–4427.

[24] S. Feng, E. Whitman, X. Xinjilefu, C.G. Atkeson, Optimization based full body control for the Atlas robot, in: IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 2014, pp. 120–127.

[25] S. Feng, E. Whitman, X. Xinjilefu, C.G. Atkeson, Optimization-based full body control for the DARPA robotics challenge, Journal of Field Robotics 32 (2015) 293–312.

[26] Fujitsu, Miniature Humanoid Robot HOAP-2 Manual, 1st edition, Fujitsu Automation Co., Ltd, 2004 (in Japanese).

[27] J. Furusho, A. Sano, Sensor-based control of a nine-link biped, The International Journal of Robotics Research 9 (1990) 83–98.

[28] E.G. Gilbert, K.T. Tan, Linear systems with state and control constraints: the theory and application of maximal output admissible sets, IEEE Transactions on Automatic Control 36 (1991) 1008–1020.

[29] A. Goswami, Postural stability of biped robots and the foot-rotation indicator (FRI) point, The International Journal of Robotics Research 18 (1999) 523–533.

[30] A. Goswami, V. Kallem, Rate of change of angular momentum and balance maintenance of biped robots, in: IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 2004, pp. 3785–3790.

[31] R.J. Griffin, G. Wiedebach, S. Bertrand, A. Leonessa, J. Pratt, Straight-leg walking through underconstrained whole-body control, in: IEEE International Conference on Robotics and Automation, ICRA, 2018, pp. 5747–5757.

[32] J.-S. Gutmann, M. Fukuchi, M. Fujita, Stair climbing for humanoid robots using stereo vision, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004, pp. 1407–1413.

[33] T. Hamano, D. Nenchev, Posture stabilization at the BoS boundary, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[34] K. Harada, K. Hauser, T. Bretl, J.-C. Latombe, Natural motion generation for humanoid robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 833–839.

[35] K. Harada, S. Kajita, K. Kaneko, H. Hirukawa, Pushing manipulation by humanoid considering two-kinds of ZMPs, in: IEEE International Conference on Robotics and Automation, 2003, pp. 1627–1632.

[36] K. Harada, S. Kajita, H. Saito, M. Morisawa, F. Kanehiro, K. Fujiwara, K. Kaneko, H. Hirukawa, A humanoid robot carrying a heavy object, in: IEEE International Conference on Robotics and Automation, 2005, pp. 1712–1717.

[37] B. Henze, A. Dietrich, C. Ott, An approach to combine balancing with hierarchical whole-body control for legged humanoid robots, IEEE Robotics and Automation Letters 1 (2016) 700–707.

[38] B. Henze, M.A. Roa, C. Ott, Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios, The International Journal of Robotics Research 35 (2016) 1522–1543.

[39] H. Herr, M. Popovic, Angular momentum in human walking, Journal of Experimental Biology 211 (2008) 467–481.

[40] A. Herzog, L. Righetti, F. Grimminger, P. Pastor, S. Schaal, Balancing experiments on a torque-controlled humanoid with hierarchical inverse dynamics, in: IEEE International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 981–988, arXiv:1305.2042.

[41] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, L. Righetti, Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid, Autonomous Robots 40 (2016) 473–491, arXiv:1410.7284v1.

[42] R. Hinata, D. Nenchev, Kicking motion generated with the relative angular momentum (RAM) balance controller, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[43] R. Hinata, D. Nenchev, Upper-body twists generated with centroidal angular momentum conservation, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[44] R. Hinata, D. Nenchev, Upper-body twists generated with coupling angular momentum conservation, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[45] R. Hinata, D. Nenchev, Upper-body twists generated with the resolved momentum approach, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[46] R. Hinata, D.N. Nenchev, Velocity-based contact stabilization of a rolling foot (convergent case), Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[47] R. Hinata, D.N. Nenchev, Velocity-based contact stabilization of a rolling foot (divergent case), Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[48] K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of Honda humanoid robot, in: IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, 1998, pp. 1321–1326.

[49] K. Hirata, M. Fujita, Analysis of conditions for non-violation of constraints on linear discrete-time systems with exogenous inputs, in: 36th IEEE Conference on Decision and Control, IEEE, 1997, pp. 1477–1478.

[50] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, M. Morisawa, A universal stability criterion of the foot contact of legged robots – adios ZMP, in: IEEE International Conference on Robotics and Automation, 2006, pp. 1976–1983.

[51] A.L. Hof, The 'extrapolated center of mass' concept suggests a simple control of balance in walking, Human Movement Science 27 (2008) 112–125.

[52] A.L. Hof, M.G.J. Gazendam, W.E. Sinke, The condition for dynamic stability, Journal of Biomechanics 38 (2005) 1–8.

[53] A. Hofmann, M. Popovic, H. Herr, Exploiting angular momentum to enhance bipedal center-of-mass control, in: IEEE Int. Conf. on Robotics and Automation, Kobe, Japan, 2009, pp. 4423–4429.

[54] M.H. Honarvar, M. Nakashima, A new measure for upright stability, Journal of Biomechanics 47 (2014) 560–567.

[55] M.A. Hopkins, D.W. Hong, A. Leonessa, Humanoid locomotion on uneven terrain using the time-varying divergent component of motion, in: IEEE-RAS International Conference on Humanoid Robots, 2014, pp. 266–272.

[56] F.B. Horak, S.M. Henry, A. Shumway-Cook, Postural perturbations: new insights for treatment of balance disorders, Physical Therapy 77 (1997) 517–533.

[57] M. Hosokawa, D.N. Nenchev, DCM-GI based contact transition control, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[58] M. Hosokawa, D.N. Nenchev, T. Hamano, The DCM generalized inverse: efficient body-wrench distribution in multi-contact balance control, Advanced Robotics 32 (2018) 778–792.

[59] Q.H.Q. Huang, S. Kajita, N. Koyachi, K. Kaneko, K. Yokoi, H. Arai, K. Komoriya, K. Tanie, A high stability, smooth walking pattern for a biped robot, in: IEEE International Conference on Robotics and Automation, 1999, pp. 65–71.

[60] M. Hutter, H. Sommer, C. Gehring, M. Hoepflinger, M. Bloesch, R. Siegwart, Quadrupedal locomotion using hierarchical operational space control, The International Journal of Robotics Research 33 (2014) 1047–1062.

[61] F.E. Huxham, P.A. Goldie, A.E. Patla, Theoretical considerations in balance assessment, Australian Journal of Physiotherapy 47 (2001) 89–100.

[62] S.-H. Hyon, Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces, IEEE Transactions on Robotics 25 (2009) 171–178.

[63] S.-H. Hyon, J. Hale, G. Cheng, Full-body compliant human-humanoid interaction: balancing in the presence of unknown external forces, IEEE Transactions on Robotics 23 (2007) 884–898.

[64] K. Iqbal, Y.-c. Pai, Predicted region of stability for balance recovery: motion at the knee joint can improve termination of forward movement, Journal of Biomechanics 33 (2000) 1619–1627.

[65] S. Kagami, F. Kanehiro, Y. Tamiya, M. Inaba, H. Inoue, AutoBalancer: an online dynamic balance compensation scheme for humanoid robots, in: Int. Workshop Alg. Found. Robot, WAFR, 2000, pp. 329–339.

[66] S. Kagami, T. Kitagawa, K. Nishiwaki, T. Sugihara, M. Inaba, H. Inoue, A fast dynamically equilibrated walking trajectory generation method of humanoid robot, Autonomous Robots 12 (2002) 71–82.

[67] S. Kajita, H. Hirukawa, K. Harada, K. Yokoi, Introduction to Humanoid Robotics, Springer Verlag, Berlin, Heidelberg, 2014.

[68] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, in: IEEE International Conference on Robotics and Automation, 2003, pp. 1620–1626.

[69] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Resolved momentum control: humanoid motion planning based on the linear and angular momentum, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, Nevada, 2003, pp. 1644–1650.

[70] S. Kajita, M. Morisawa, K. Harada, K. Kaneko, F. Kanehiro, K. Fujiwara, H. Hirukawa, Biped walking pattern generator allowing auxiliary ZMP control, in: IEEE International Conference on Intelligent Robots and Systems, 2006, pp. 2993–2999.

[71] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, K. Yokoi, Biped walking stabilization based on linear inverted pendulum tracking, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 2010, pp. 4489–4496.

[72] S. Kajita, K. Tani, Study of dynamic walk control of a biped robot on rugged terrain – derivation and application of the linear inverted pendulum mode, Journal of Robotics and Mechatronics 5 (1993) 516–523.

[73] S. Kajita, K. Tani, Experimental study of biped dynamic walking, IEEE Control Systems Magazine 16 (1996) 13–19.

[74] S. Kajita, K. Yokoi, M. Saigo, K. Tanie, Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback, in: IEEE International Conference on Robotics and Automation, 2001, pp. 3376–3382.

[75] K. Kaneko, F. Kanehiro, S. Kajita, M. Morisawa, K. Fujiwara, K. Harada, H. Hirukawa, Motion suspension system for humanoids in case of emergency; real-time motion generation and judgment to suspend humanoid, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2006, pp. 5496–5503.

[76] A. Konno, T. Myojin, T. Matsumoto, T. Tsujita, M. Uchiyama, An impact dynamics model and sequential optimization to generate impact motions for a humanoid robot, The International Journal of Robotics Research 30 (2011) 1596–1608.

[77] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, J. Pratt, Design of a momentum-based control framework and application to the humanoid robot Atlas, International Journal of Humanoid Robotics 13 (2016) 1650007.

[78] T. Koolen, T. de Boer, J. Rebula, A. Goswami, J. Pratt, Capturability-based analysis and control of legged locomotion, Part 1: theory and application to three simple gait models, The International Journal of Robotics Research 31 (2012) 1094–1113.

[79] T. Koolen, M. Posa, R. Tedrake, Balance control using center of mass height variation: limitations imposed by unilateral contact, in: IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), IEEE, 2016, pp. 8–15.

[80] K. Koyanagi, H. Hirukawa, S. Hattori, M. Morisawa, S. Nakaoka, K. Harada, S. Kajita, A pattern generator of humanoid robots walking on a rough terrain using a handrail, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 2617–2622.

[81] S. Kudoh, T. Komura, K. Ikeuchi, The dynamic postural adjustment with the quadratic programming method, in: IEEE/RSJ International Conference on Intelligent Robots and System, 2002, pp. 2563–2568.

[82] S. Kudoh, T. Komura, K. Ikeuchi, Stepping motion for a human-like character to maintain balance against large perturbations, in: IEEE International Conference on Robotics and Automation, 2006, pp. 2661–2666.

[83] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, K. Mombaur, Optimal control for whole-body motion generation using center-of-mass dynamics for predefined multi-contact configurations, in: IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids), 2015, pp. 684–689.

[84] J.J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, H. Inoue, Dynamically-stable motion planning for humanoid robots, Autonomous Robots 12 (2002) 105–118.

[85] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, R. Tedrake, Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot, Autonomous Robots 40 (2016) 429–455.

[86] A. Kuo, An optimal control model for analyzing human postural balance, IEEE Transactions on Biomedical Engineering 42 (1995) 87–101.

[87] S.-H. Lee, A. Goswami, Ground reaction force control at each foot: a momentum-based humanoid balance controller for non-level and non-stationary ground, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Taipei, Taiwan, 2010, pp. 3157–3162.

[88] S.-H.H. Lee, A. Goswami, A momentum-based balance controller for humanoid robots on non-level and non-stationary ground, Autonomous Robots 33 (2012) 399–414.

[89] Y. Lee, S. Hwang, J. Park, Balancing of humanoid robot using contact force/moment control by task-oriented whole body control framework, Autonomous Robots 40 (2016) 457–472.

[90] S. Lengagne, J. Vaillant, E. Yoshida, A. Kheddar, Generation of whole-body optimal dynamic multi-contact motions, The International Journal of Robotics Research 32 (2013) 1104–1119.

[91] C.W. Luchies, N.B. Alexander, A.B. Schultz, J. Ashton-Miller, Stepping responses of young and old adults to postural disturbances: kinematics, Journal of the American Geriatrics Society 42 (1994) 506–512.

[92] A. Macchietto, V. Zordan, C.R. Shelton, Momentum control for balance, ACM Transactions on Graphics 28 (2009) 80.

[93] B. Maki, W. Mcilroy, G. Fernie, Change-in-support reactions for balance recovery, IEEE Engineering in Medicine and Biology Magazine 22 (2003) 20–26.

[94] B.E. Maki, W.E. McIlroy, The role of limb movements in maintaining upright stance: the “change-in-support” strategy, Physical Therapy 77 (1997) 488–507.

[95] D. Mansour, A. Micaelli, P. Lemerle, A computational approach for push recovery in case of multiple noncoplanar contacts, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, San Francisco, CA, USA, 2011, pp. 3213–3220.

[96] D. Mansour, A. Micaelli, P. Lemerle, Humanoid push recovery control in case of multiple non-coplanar contacts, in: IEEE International Conference on Intelligent Robots and Systems, 2013, pp. 4137–4144.

[97] K. Mitobe, G. Capi, Y. Nasu, Control of walking robots based on manipulation of the zero moment point, Robotica 18 (2000) 651–657.

[98] K. Miura, M. Morisawa, F. Kanehiro, S. Kajita, K. Kaneko, K. Yokoi, Human-like walking with toe supporting for humanoids, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 2011, pp. 4428–4435.

[99] F. Miyazaki, S. Arimoto, A control theoretic study on dynamical biped locomotion, Journal of Dynamic Systems, Measurement, and Control 102 (1980) 233.

[100] I. Mordatch, M. de Lasa, A. Hertzmann, Robust physics-based locomotion using low-dimensional planning, ACM Transactions on Graphics 29 (2010) 1.

[101] M. Morisawa, K. Kaneko, F. Kanehiro, S. Kajita, K. Fujiwara, K. Harada, H. Hirukawa, Motion planning of emergency stop for humanoid robot by state space approach, in: IEEE International Conference on Intelligent Robots and Systems, 2006, pp. 2986–2992.

[102] M. Murooka, S. Noda, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Achievement of pivoting large and heavy objects by life-sized humanoid robot based on online estimation control method of object state and manipulation force, Journal of the Robotics Society of Japan 32 (2014) 595–602.

[103] M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot, in: IEEE International Conference on Robotics and Automation, 2015, pp. 5682–5689.

[104] M. Murooka, R. Ueda, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Global planning of whole-body manipulation by humanoid robot based on transition graph of object motion and contact switching, Advanced Robotics 31 (2017) 322–340.

[105] M.P. Murray, A. Seireg, R.C. Scholz, Center of gravity, center of pressure, and supportive forces during human activities, Journal of Applied Physiology 23 (1967) 831–838.

[106] K. Nagasaka, M. Inaba, H. Inoue, Stabilization of dynamic walk on a humanoid using torso position compliance control, in: Proceedings of the 17th Annual Conference of the Robotics Society of Japan, 1999, pp. 1193–1194.

[107] T. Nakamura, D.N. Nenchev, Velocity-based posture stabilization on a balance board, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[108] L.M. Nashner, G. McCollum, The organization of human postural movements: a formal basis and experimental synthesis, Behavioral and Brain Sciences 8 (1985) 135–150.

[109] E.S. Neo, K. Yokoi, S. Kajita, K. Tanie, Whole-body motion generation integrating operator's intention and robot's autonomy in controlling humanoid robots, IEEE Transactions on Robotics 23 (2007) 763–775.

[110] S. Noda, M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Generating whole-body motion keep away from joint torque, contact force, contact moment limitations enabling steep climbing with a real humanoid robot, in: IEEE International Conference on Robotics and Automation, 2014, pp. 1775–1781.

[111] D.E. Orin, A. Goswami, Centroidal momentum matrix of a humanoid robot: structure and properties, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Nice, France, 2008, pp. 653–659.

[112] D.E. Orin, A. Goswami, S.H. Lee, Centroidal dynamics of a humanoid robot, Autonomous Robots 35 (2013) 161–176.

[113] C. Ott, A. Kugi, Y. Nakamura, Resolving the problem of non-integrability of nullspace velocities for compliance control of redundant manipulators by using semi-definite Lyapunov functions, in: IEEE International Conference on Robotics and Automation, 2008, pp. 1999–2004.

[114] C. Ott, M.A. Roa, G. Hirzinger, Posture and balance control for biped robots based on contact force optimization, in: IEEE-RAS International Conference on Humanoid Robots, Bled, Slovenia, 2011, pp. 26–33.

[115] B. Paden, R. Panja, Globally asymptotically stable 'PD+' controller for robot manipulators, International Journal of Control 47 (1988) 1697–1712.

[116] Y.-c. Pai, J. Patton, Center of mass velocity-position predictions for balance control, Journal of Biomechanics 30 (1997) 347–354.

[117] J. Park, J. Haan, F.C. Park, Convex optimization algorithms for active balancing of humanoid robots, IEEE Transactions on Robotics 23 (2007) 817–822.

[118] K. Park, P. Chang, J. Salisbury, A unified approach for local resolution of kinematic redundancy with inequality constraints and its application to nuclear power plant, in: International Conference on Robotics and Automation, IEEE, 1997, pp. 766–773.

[119] O.E.R. Ponce, Generation of the Whole-Body Motion for Humanoid Robots with the Complete Dynamics, Ph.D. thesis, Toulouse University, France, 2014.

[120] M. Popovic, A. Hofmann, H. Herr, Angular momentum regulation during human walking: biomechanics and control, in: IEEE International Conference on Robotics and Automation, 2004, pp. 2405–2411.

[121] M.B. Popovic, A. Goswami, H. Herr, Ground reference points in legged locomotion: definitions, biological trajectories and control implications, The International Journal of Robotics Research 24 (2005) 1013–1032.

[122] J. Pratt, J. Carff, S. Drakunov, A. Goswami, Capture point: a step toward humanoid push recovery, in: IEEE-RAS International Conference on Humanoid Robots, Genoa, Italy, 2006, pp. 200–207.

[123] Qiang Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie, Planning walking patterns for a biped robot, IEEE Transactions on Robotics and Automation 17 (2001) 280–289.

[124] O.E. Ramos, N. Mansard, P. Soueres, Whole-body motion integrating the capture point in the operational space inverse dynamics control, in: IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 2014, pp. 707–712.

[125] J. Rebula, F. Cañnas, J. Pratt, A. Goswami, Learning capture points for humanoid push recovery, in: IEEE-RAS International Conference on Humanoid Robots, 2008, pp. 65–72.

[126] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, S. Schaal, Optimal distribution of contact forces with inverse-dynamics control, The International Journal of Robotics Research 32 (2013) 280–298.

[127] T. Saida, Y. Yokokohji, T. Yoshikawa, FSW (feasible solution of wrench) for multi-legged robots, in: IEEE International Conference on Robotics and Automation, 2003, pp. 3815–3820.

[128] A. Sano, J. Furusho, 3D dynamic walking of biped robot by controlling the angular momentum, Transactions of the Society of Instrument and Control Engineers 26 (1990) 459–466.

[129] A. Sano, J. Furusho, Realization of natural dynamic walking using the angular momentum information, in: IEEE International Conference on Robotics and Automation, Tsukuba, Japan, 1990, pp. 1476–1481.

[130] A. Sano, J. Furusho, Control of torque distribution for the BLR-G2 biped robot, in: Fifth International Conference on Advanced Robotics, vol. 1, IEEE, 1991, pp. 729–734.

[131] P. Sardain, G. Bessonnet, Forces acting on a biped robot. Center of pressure—zero moment point, IEEE Transactions on Systems, Man and Cybernetics. Part A. Systems and Humans 34 (2004) 630–637.

[132] T. Sato, S. Sakaino, E. Ohashi, K. Ohnishi, Walking trajectory planning on stairs using virtual slope for biped robots, IEEE Transactions on Industrial Electronics 58 (2011) 1385–1396.

[133] L. Sentis, Synthesis and Control of Whole-Body Behaviors in Humanoid Systems, Ph.D. thesis, Stanford University, 2007.

[134] L. Sentis, O. Khatib, Synthesis of whole-body behaviors through hierarchical control of behavioral primitives, International Journal of Humanoid Robotics 02 (2005) 505–518.

[135] L. Sentis, O. Khatib, Compliant control of multicontact and center-of-mass behaviors in humanoid robots, IEEE Transactions on Robotics 26 (2010) 483–501.

[136] M. Shibuya, T. Suzuki, K. Ohnishi, Trajectory planning of biped robot using linear pendulum mode for double support phase, in: IECON Proceedings (Industrial Electronics Conference), 2006, pp. 4094–4099.

[137] T. Shibuya, D. Nenchev, Inclined surface cleaning with motion/force control under external disturbances, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[138] T. Shibuya, S. Sakaguchi, D. Nenchev, Inclined surface cleaning with motion/force control, Robotic Life Support Laboratory, Tokyo City University, 2018 (Video clip), https://doi.org/10.1016/B978-0-12-804560-2.00012-2.

[139] A. Shumway-Cook, F.B. Horak, Vestibular rehabilitation: an exercise approach to managing symptoms of vestibular dysfunction, Seminars in Hearing 10 (1989) 196–209.

[140] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, Planning and Control, Springer-Verlag, London, 2009.

[141] M. da Silva, Y. Abe, J. Popović, Interactive simulation of stylized human locomotion, ACM Transactions on Graphics 27 (2008) 1.

[142] B.J. Stephens, C.G. Atkeson, Dynamic balance force control for compliant humanoid robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 1248–1255.

[143] T. Sugihara, Standing stabilizability and stepping maneuver in planar bipedalism based on the best COM-ZMP regulator, in: IEEE International Conference on Robotics and Automation, Kobe, Japan, 2009, pp. 1966–1971.

[144] T. Sugihara, Y. Nakamura, Whole-body cooperative balancing of humanoid robot using COG Jacobian, in: IEEE/RSJ Int. Conf. on Intelligent Robots and System, Lausanne, Switzerland, 2002, pp. 2575–2580.

[145] T. Sugihara, Y. Nakamura, H. Inoue, Real-time humanoid motion generation through ZMP manipulation based on inverted pendulum control, in: IEEE Int. Conf. on Robotics and Automation, Washington, DC, 2002, pp. 1404–1409.

[146] A. Takanishi, M. Ishida, Y. Yamazaki, I. Kato, The realization of dynamic walking by the biped walking robot WL-10RD, Journal of the Robotics Society of Japan 3 (1985) 325–336.

[147] A. Takanishi, H.-o. Lim, M. Tsuda, I. Kato, Realization of dynamic biped walking stabilized by trunk motion on a sagittally uneven surface, in: IEEE International Workshop on Intelligent Robots and Systems, 1990, pp. 323–330.

[148] A. Takanishi, M. Tochizawa, H. Karaki, I. Kato, Dynamic biped walking stabilized with optimal trunk and waist motion, in: IEEE/RSJ International Workshop on Intelligent Robots and Systems, 1989, pp. 187–192.

[149] T. Takenaka, T. Matsumoto, T. Yoshiike, Real time motion generation and control for biped robot – 1st report: walking gait pattern generation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009, pp. 1084–1091.

[150] T. Takenaka, T. Matsumoto, T. Yoshiike, T. Hasegawa, S. Shirokura, H. Kaneko, A. Orita, Real time motion generation and control for biped robot – 4th report: integrated balance control, in: IEEE/RSJ Int. Conf. on Intelligent Robots and System, 2009, pp. 1601–1608.

[151] Y. Tamiya, M. Inaba, H. Inoue, Realtime balance compensation for dynamic motion of full-body humanoid standing on one leg, Journal of the Robotics Society of Japan 17 (1999) 112–118.

[152] T. Tanaka, T. Takubo, K. Inoue, T. Arai, Emergent stop for humanoid robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2006, pp. 3970–3975.

[153] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, F. Kanehiro, Multi-contact vertical ladder climbing with an HRP-2 humanoid, Autonomous Robots 40 (2016) 561–580.

[154] M. Vukobratović, Contribution to the study of anthropomorphic systems, Kybernetika 8 (1972) 404–418.

[155] M. Vukobratovic, B. Borovac, Zero-moment point — thirty five years of its life, International Journal of Humanoid Robotics 01 (2004) 157–173.

[156] M. Vukobratović, A.A. Frank, D. Juricić, On the stability of biped locomotion, IEEE Transactions on Bio-Medical Engineering 17 (1970) 25–36.

[157] M. Vukobratovic, D. Juricic, Contribution to the synthesis of biped gait, IEEE Transactions on Biomedical Engineering BME-16 (1969) 1–6.

[158] P.M. Wensing, G. Bin Hammam, B. Dariush, D.E. Orin, Optimizing foot centers of pressure through force distribution in a humanoid robot, International Journal of Humanoid Robotics 10 (2013) 1350027.

[159] P.-B. Wieber, On the stability of walking systems, in: Third IARP International Workshop on Humanoid and Human Friendly Robotics, Tsukuba, Japan, 2002, pp. 1–7.

[160] P.-B. Wieber, Viability and predictive control for safe locomotion, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008, pp. 1103–1108.

[161] G. Wiedebach, S. Bertrand, T. Wu, L. Fiorio, S. McCrory, R. Griffin, F. Nori, J. Pratt, Walking on partial footholds including line contacts with the humanoid robot Atlas, in: IEEE-RAS International Conference on Humanoid Robots, 2016, pp. 1312–1319.

[162] D. Winter, A.E. Patla, J.S. Frank, Assessment of balance control in humans, Medical Progress Through Technology 16 (1990) 31–51.

[163] A. Witkin, M. Kass, Spacetime constraints, ACM SIGGRAPH Computer Graphics 22 (1988) 159–168.

[164] J. Yamaguchi, A. Takanishi, I. Kato, Development of a biped walking robot compensating for three-axis moment by trunk motion, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, pp. 561–566.

[165] K. Yamamoto, Control strategy switching for humanoid robots based on maximal output admissible set, Robotics and Autonomous Systems 81 (2016) 17–32.

[166] R. Yang, Y.-Y. Kuen, Z. Li, Stabilization of a 2-DOF spherical pendulum on X-Y table, in: IEEE International Conference on Control Applications, 2000, pp. 724–729.

[167] Yu Zheng, K. Yamane, Human motion tracking control with strict contact force constraints for floating-base humanoid robots, in: IEEE-RAS International Conference on Humanoid Robots, IEEE, Atlanta, GA, USA, 2013, pp. 34–41.

[168] Y. Zheng, C.-M. Chew, Fast equilibrium test and force distribution for multicontact robotic systems, Journal of Mechanisms and Robotics 2 (2010) 021001.

[169] Y.F. Zheng, J. Shen, Gait synthesis for the SD-2 biped robot to climb sloping surface, IEEE Transactions on Robotics and Automation 6 (1990) 86–96.