Multi-contact Motion Planning and Control

Karim Bouyarmane, Stéphane Caron, Adrien Escande,
and Abderrahmane Kheddar

Contents

K. Bouyarmane (✉)
CNRS, Inria Nancy - Grand Est, Loria UMR 7503, Larsen team, Université de Lorraine, Vandœuvre-Lès-Nancy, France
e-mail: karim.bouyarmane@gmail.com

S. Caron
Interactive Digital Human (IDH), CNRS-University of Montpellier, Montpellier, France
e-mail: stephane.caron@lirmm.fr

A. Escande
CNRS-AIST Joint Robotics Laboratory (JRL), Tsukuba, Japan
e-mail: adrien.escande@gmail.com

A. Kheddar
Interactive Digital Human (IDH), CNRS-University of Montpellier, Montpellier, France
CNRS-AIST Joint Robotics Laboratory (JRL), Tsukuba, Japan
e-mail: kheddar@gmail.com

A. Goswami, P. Vadakkepat (eds.), *Humanoid Robotics: A Reference*,
https://doi.org/10.1007/978-94-007-6046-2_32

Abstract

The essence of humanoid robots is their ability to reproduce human skills in locomotion and manipulation. Early efforts in humanoid research were dedicated to bipedal walking, first on flat terrains and recently on uneven ones, while the manipulation capabilities inherit from the literature in bimanual and dexterous-hand manipulation. In practice, the two problems interact largely. Locomotion in cluttered spaces benefits from extra contacts between any part of the robot and the environment, such as when grippers grasp a handrail during stair climbing, while legs can conversely enhance manipulation capabilities, such as when arching the whole body to augment contact pressure at an end effector. The two problems share the same background: they are governed by non-smooth dynamics (friction and impacts at contacts) under viability constraints including dynamic stability. Consequently, they are now solved jointly. This chapter highlights the state-of-the-art techniques used for this purpose in multi-contact planning and control.

1 Introduction

Planning a path with contacts is much more complex than the classical piano mover's problem. Figure 1 illustrates the source of this complexity: rather than the obstacle-free part of a continuous space, the planner now needs to explore states located on a *constrained subspace* (and more precisely a collection of submanifolds; see [35]). Classical probabilistic methods don't apply anymore and need to be supplemented with, e.g., dedicated projections to ensure that all states (as well as the trajectories that connect them) satisfy contact constraints.

The left picture of Fig. 1 illustrates planning in the configuration space, where the robot boils down to a multidimensional point and the environment is split into the (collision-)free space and the obstacle space. State-of-the-art probabilistic roadmap planners randomly sample configurations and connect them using steering methods to build a roadmap that approximates the connectivity of the set of feasible configurations. The right part of Fig. 1 highlights the complexity when considering that the "obstacle" space is in fact a support since the robot has to "collide" (make contact) with at least one of its link. Once a contact is established, the following motion shall be free from collision during the transition to the next contact configuration – sliding is an exception to this rule. Roughly saying, we plan in the

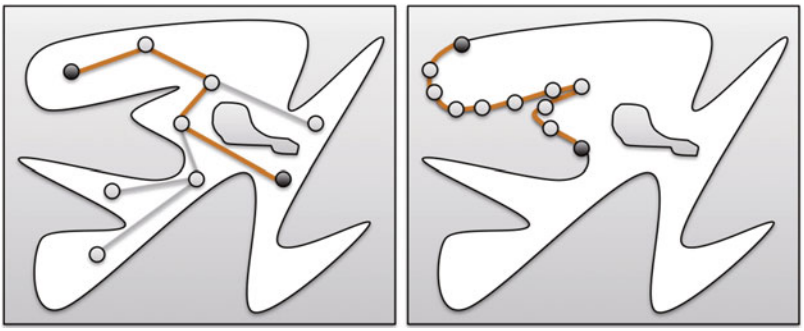

Fig. 1 Classical planning in the free configuration space using probabilistic approaches (*left*). Assuming "obstacles" are potential contact supports and any link of the robot can be used to contact the environment, planning occurs at the frontier of the free space and the obstacle space (which we call the contact space), which is of zero measure

frontier of the free space and the obstacle space, which is of zero measure (the probability to sample a configuration point at that exact frontier is zero).

The contact planning problem is a hybrid process that alternates between a discrete choice of contact sequencing sets and continuous motion sets. Each of these sets has its own constraints. The topology of the configuration space is much more complex: contacts define strata of different dimensions, and each stratum is foliated into a continuum of leaves. A leaf is a submanifold of the stratum in which the robot is fully actuated (See, e.g., Figure 2 in [35] for an illustration of this terminology.). The discrete part of contact planning mixes three kinds of choices. First, when the robot is in a given contact combination, it chooses among next combinations. For example, the robot can attempt to add a contact and go to some configuration (including the new contact's constraint) or add simultaneously two contacts and go to another configuration different from those reachable when adding only one contact. The latter reasoning also applies to contact removal. The binary choice of either removing or adding a contact from a given contact configuration. Second, when adding a contact, there is a continuous choice to be made as to where the contact should occur, i.e., what will be the contact configuration of the next submanifolds. Finally, one must chose the continuous trajectory from one configuration corresponding to the first contact combination to one corresponding to the next combination.

Other difficulties arise in the presence of obstacles or when other robot limitations (range of motion, actuation limits, etc.) reduce the feasible space, so that it might not always be possible to find a path between two contact configurations. There can also be several contact surfaces, increasing the number of combinations for single, double, and triple contact submanifolds; for example, not only the choice of the feet in contact must be made but also the pairing of each contact foot with a surface.

In [14], the complexity of multi-contact planning together with various planning and control approaches is highlighted in three toy example robots exemplifying locomotion, manipulation, and hybrid locomotion-and-manipulation systems having the lowest possible configuration spaces' dimension in order to capture its topology and complements very well the material presented in this chapter.

To summarize, contact planning features three kinds of choices:

- The discrete choice of the sequence of contact combinations, which corresponds to a sequence of foliations,
- The continuous choice of contact locations, that is to say the choice of a contact configuration within a foliation, and
- The continuous choice of a path between two contact combinations.

This triple level of choices needs to be assessed with regard to the huge size of the set of discrete choices, furthermore noting that each of these choices highly depends on all the other choices made before it. In particular, the continuous choice of a contact configuration at a given time will impact further possibilities of discrete next contact choices. The addition of physical considerations such as static or dynamic equilibrium adds complexity to the problem and precludes the use of the reduction property [87].

Notable works in contact planning include the planner of [17] for climbing robots, later adapted to humanoid robots by [46]. Contacts happen at fixed predefined locations in the environment, sometimes sampled at random on given surfaces. Due to the use of a randomized planner, the output plan is jerky: smoothing is applied in a post-processing phase, yet it cannot eliminate all unnecessary movements, and overall motions are still unnatural. This fact owes both to randomized planning and fixed contact locations that might not be suitable. In [46], motion primitives were used to get smoother paths and choose more "natural" contacts, yet at the cost of generality regarding the class of discoverable output motions. To avoid this drawback, later works [12,35,68] allowed contacts to be taken anywhere on predefined surfaces by selecting robot postures and contact locations through minimization of a cost function. The user also gained some control over the overall look of the solution by tuning cost-function parameters.

Multi-contact planning extends beyond locomotion problems. It addresses various instances of manipulation problems, as well as hybrid locomotion-and-manipulation problems. It is commonly thought that locomotion resides in the legs, while arms are devoted to manipulation: in this case, humanoid robots are seen as dual-arm manipulators, either standing on fixed foot locations while performing a collision-free motion with the rest of their limbs [55,98] or as walking bipeds that plan their path in a 2D horizontal plane among obstacles [27,56]. When seen as dual-arm walking robots, the two approaches can be combined by decoupling the two problems [37,75,100]. However, there are situations where this decoupling does not hold: for instance, humanoid robots have used their arms to push boxes [43], climb ladders [91], or crawl under tables [35]. From a motion planning and control point of view, locomotion and manipulation (including in-body object manipulation) are

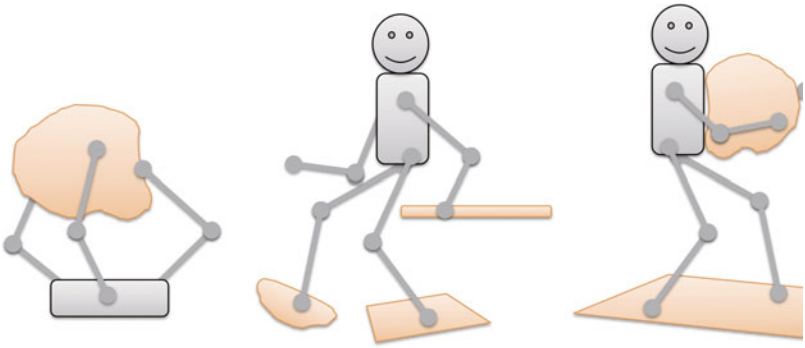

Fig. 2 Multi-contact commonalities in dexterous manipulation of an object (*left*), non-gaited locomotion using various environment support (*middle*), and whole-body loco-manipulation combining the left two cases (*right*)

conceptually the same problem. Their common ground lies in the way they solve under-actuation through contact forces. A locomoting robot is underactuated in the sense that the position of its mobile base is not controlled directly through actuators torques but rather results from both actuated torques and contact forces applied to the environment. Similarly, a manipulated object is also underactuated as its position is an indirect result of the actuation of the manipulator through contact forces. Both problems obey analogous form Lagrangian dynamics equations, they both involve friction, and they both have contact strata of various dimensions, as depicted in Fig. 2.

Accounting for this common ground, modern developments have converged toward a framework general enough to be applied to all these settings: contacts are represented by both geometric constraints and contact force variables, all being solved as part of a numerical optimization problem [19, 68, 82].

In this chapter, fundamentals about the retained contact friction model (namely, the Coulomb friction model) are first presented in Sect. 2. Section 3 is about the core multi-contact planning (MCP) problem and details an MCP algorithm, as an example approach mainly established by the authors among others to solve the problem. Section 4 presents a method for multi-contact control, that is, a method to execute the plans output by a given MCP algorithm in a dynamically and physically consistent way, under the Coulomb friction model contact constraints. The multi-contact control is based on a quadratic programming (QP) formulation, also presented in this chapter as an approach developed by the authors among others and that is now widely established for multi-contact control in the humanoid research community. Section 5 lists different ways to incorporate model-predictive control in multi-contact planning and control. Section 6 reviews past, present, and future envisioned applications of multi-contact planning and control for humanoid robots. Finally, Sect. 7 concludes the chapter.

2 Contact Physics

At the heart of multi-contact planning and control lies the physics of *contact* which we will review before delving into planning and control algorithms.

Contact results from mesoscopic phenomena, which happen on different spatial and time scales than the (macroscopic) rigid-body dynamics that otherwise govern the mechanical behavior of robots. We use contact models to predict what reaction forces and motions a contact may yield in response to an external load.

Definition 1 (Contact mode). A contact mode is the description of all six degrees of freedom and degrees of constraint of a contact.

For instance, the *fixed* contact mode has all six degrees constrained (no translation, no rotation at contact), while the translational *sliding* contact mode has two degrees of freedom (sliding along the contact plane) and four degrees of constraint (no rotation, no detaching from the plane). Most methods developed to date rely on the assumption that contacts, once made, are fixed. This precludes other contact modes [4] that can happen in practice, notably sliding. Switching between contact modes happens when certain boundary conditions are crossed, which is what we generally want to avoid, unless it is planned for and/or controlled. We will therefore derive these boundary conditions, known as *contact-stability conditions*, and make sure that the robot fulfills them while moving, in order to sustain its contacts.

Definition 2 (Contact stability [88]). A motion is *contact stable* when all contacts between the robot in the environment remain in the same *contact mode* throughout the motion. Otherwise, the motion is said to be *contact switching*.

Note how, despite terminology resemblance, the word "stability" here is not a priori related to the notion of (e.g., asymptotic) stability found in control theory. To allow for some distinction, the expression *contact stability* is customarily used as a whole.

Consider the case of a 2D mass in contact with a single surface as depicted in Fig. 3. The contact remains fixed as long as the contact force $\boldsymbol{f}^c = m\boldsymbol{g} - \boldsymbol{f}^{\text{ext}}$ lies within the Coulomb friction cone $\mathscr{K}$. As soon as $\boldsymbol{f}^c$ exits $\mathscr{K}$, the contact switches to the sliding mode. The property $\boldsymbol{f}^c \in \mathscr{K}$ is therefore the contact-stability condition for this simple mass in contact. In what follows, we will derive more general contact-stability criteria for both the humanoid multibody system and its reduced models (zero-tilting moment point, centroidal dynamics).

2.1 Single-Body Contact Stability

Consider the set of points $\{C_i\}$ where the robot contacts its environment. Assuming that the environment surface is smooth enough, one can consider its unit normal $\boldsymbol{n}_i$ at C_i, pointing from the environment to the robot. Let $\boldsymbol{f}_i^c$ denote the *contact force* exerted at C_i by the environment onto the robot:

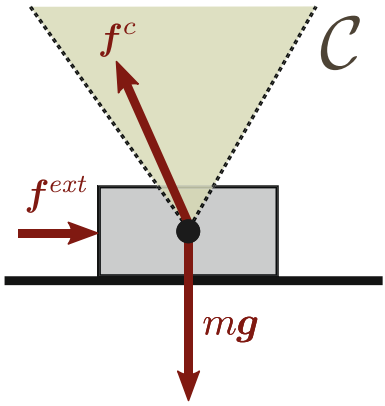

Fig. 3 Mass in fixed contact with a surface and pushed by an external load $\boldsymbol{f}^{\text{ext}}$. The contact-stability condition is $\boldsymbol{f}^c \in \mathscr{K}$ with $\mathscr{K}$ the Coulomb friction cone between the two contacting surfaces. If $\boldsymbol{f}^{\text{ext}}$ causes $\boldsymbol{f}^c$ to exit $\mathscr{K}$, the mass will start to slide, resulting in an overall contact-switching motion

- The normal component $p_i \stackrel{\text{def}}{=} (\boldsymbol{n}_i \cdot \boldsymbol{f}_i^c)$ is called *pressure force*, and
- The tangential component $\boldsymbol{f}_i^t \stackrel{\text{def}}{=} \boldsymbol{f}_i^c - (\boldsymbol{n}_i \cdot \boldsymbol{f}_i^c)\boldsymbol{n}_i$ is the *friction force* at C_i.

Definition 3 (Coulomb friction). A point contact remains in the fixed contact mode as long as its contact force $\boldsymbol{f}_i^c$ lies inside the *friction cone* directed by $\boldsymbol{n}_i$:

$$\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i \; > \; 0, \quad \text{and} \quad \left\| \boldsymbol{f}_i^t \right\|_2 \; \leq \; \mu_i (\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i), \tag{1}$$

where μ_i is the static friction coefficient at contact C_i.

The Euclidean norm $\|\cdot\|_2$ in this definition represents friction cones with circular sections, which models the isotropy of friction. Although more realistic, this model presents some computational challenges down the control pipeline. A common practice in motion planning [17,23,35,46] and control [3,26,49,57] is to consider its linear approximation:

Definition 4 (Linearized Coulomb friction [88]). A point contact remains in the fixed contact mode, while its contact force $\boldsymbol{f}_i^c$ lies inside the *linearized friction cone* directed by $\boldsymbol{n}_i$:

$$\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i \; > \; 0, \quad \text{and} \quad \frac{\boldsymbol{f}_i^t}{\tilde{\mu}_i (\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i)} \; \in \; \mathscr{P}_n, \tag{2}$$

where $\mathscr{P}_n$ is a regular n-sided polygon inscribed in the 2D unit circle.

This approximation can be made as close as desired to the original model by increasing the number of edges n of the section polygon $\mathscr{P}_n$. Equation (2) provides a set of linear inequality constraints. For example, the four-sided friction pyramid obtained for $n = 4$ can be written:

$$\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i > 0 \tag{3}$$

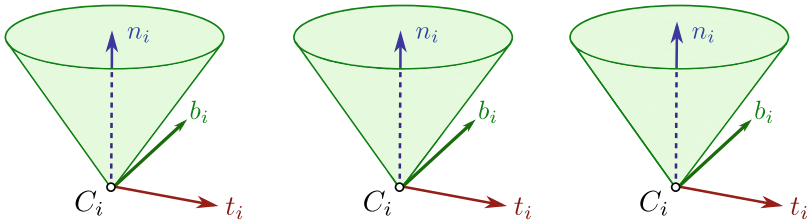

Fig. 4 Friction cone at the contact point C_i with local contact frame $(\boldsymbol{t}_i, \boldsymbol{b}_i, \boldsymbol{n}_i)$, represented, respectively, alone (*left*), with the outer (*middle*) and inner (*right*) linear approximations (This figure is adapted from Figure 1 in [88])

$$|\boldsymbol{f}_i^c \cdot \boldsymbol{t}_i| \leq \tilde{\mu}_i (\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i), \tag{4}$$

$$|\boldsymbol{f}_i^c \cdot \boldsymbol{b}_i| \leq \tilde{\mu}_i (\boldsymbol{f}_i^c \cdot \boldsymbol{n}_i), \tag{5}$$

with $(\boldsymbol{t}_i, \boldsymbol{b}_i)$ any basis of the tangential contact plane such that $(\boldsymbol{t}_i, \boldsymbol{b}_i, \boldsymbol{n}_i)$ is a direct frame. For $\tilde{\mu}_i = \mu_i$, the linearized Coulomb cone is an outer approximation of the circular one, while it is an inner approximation for $\tilde{\mu}_i = \mu_i / \sqrt{2}$, as depicted in Fig. 4.

2.1.1 Surface Contacts

In motion planning, surface contacts are commonly modeled by taking contact points at the vertices of polygonal contact patches [26, 35, 44]. This practice is tightly related to the goal of achieving *weak* contact stability:

Definition 5 (Weak contact stability [73]). A rigid body in contact is *weak contact stable* if there exists a set of point contact forces distributed on the contact area both satisfying Coulomb friction and summing up to its resultant contact wrench.

The "there exists" in this definition is important. Rather than computing force trajectories explicitly for each contact point, which would be intractable when the number of points is infinite, this condition only ensures the feasibility of the resultant wrench exerted on the rigid body in contact:

Proposition 1 (Contact Stability for a Surface Patch (Continuous)). *A surface patch in the fixed contact mode is (weak) contact stable if and only if there exists a continuous distribution of pressure and mechanical shear satisfying Coulomb friction and summing up to the resultant contact wrench.*

Forces at the vertices of the contact polygon are sufficient for this criterion:

Proposition 2 (Contact Stability for a Surface Patch (Discrete) [24]). *Equivalently, a surface patch in the fixed contact mode is (weak) contact stable if and only if there exists a set of contact forces* $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N\}$ *applied at its vertices* $\{C_1, \ldots, C_N\}$ *satisfying Coulomb friction and summing up to its resultant contact wrench.*

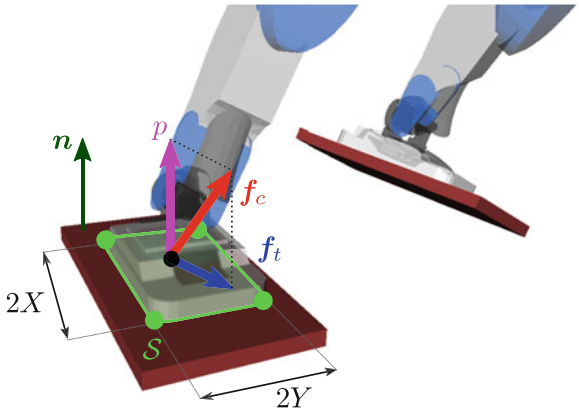

Fig. 5 Polygonal contact patch for a single contact. Forces are applied at the vertices of the contact area. In the rectangular case here, there are four vertices, and we denote by X (resp. Y) the half-length (resp. half-width) of the area

Thus, planners that rely on weak contact stability [28, 29, 35, 46, 63] can use a discrete set of contact points. However, the resulting planned forces $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N\}$ at these points should be considered as no more than a certificate, as nothing guarantees that they can be realized in practice. For instance, toe forces on a foot may point backward and heel forces forward, which cannot be realized when the foot is a single rigid body.

Let us now consider the practical case of a humanoid foot shaped as a rectangular contact patch (C_1, C_2, C_3, C_4) of center O and dimensions $2X \times 2Y$ (see Fig. 5). Let $\boldsymbol{w}_O = (\boldsymbol{f}, \boldsymbol{\tau}_O)$ denote the contact wrench resulting from the contact forces $(\boldsymbol{f}_1^c, \ldots, \boldsymbol{f}_4^c)$ applied at corner vertices.

Proposition 3 (Contact Stability for a Surface Patch (CWC) [24]). *A rectangular surface patch in the fixed contact mode is (weak) contact stable if and only if its resultant contact wrench $\boldsymbol{w}_O = (\boldsymbol{f}, \boldsymbol{\tau}_O)$ lies inside the Contact Wrench Cone (CWC) given by*

$$|f_x|, |f_y| \leq \mu f_z \tag{6}$$

$$|\tau_{O,x}| \leq Y f_z \tag{7}$$

$$|\tau_{O,y}| \leq X f_z \tag{8}$$

$$\tau_{O,z}^{\min} \leq \tau_{O,z} \leq \tau_{O,z}^{\max} \tag{9}$$

where $\tau_{O,z}^{\min}$ and $\tau_{O,z}^{\max}$ are defined by

$$\tau_{O,z}^{\min} \stackrel{\text{def}}{=} -\mu(X+Y) f^z + |Y f^x - \mu \tau_O^x| + |X f^y - \mu \tau_O^y|,$$

$$\tau_{O,z}^{\max} \stackrel{\text{def}}{=} +\mu(X+Y) f^z - |Y f^x + \mu \tau_O^x| - |X f^y + \mu \tau_O^y|.$$

Note that the minimum yaw torque $\tau_{O,z}^{\min}$ is *not* the opposite of $\tau_{O,z}^{\max}$. Also, observe how the latter relation (9) is more different from "no rotation occurs while τ_O^z is

small enough." To gain insight on this phenomenon, let us reformulate the condition at the center of pressure (COP):

Proposition 4 (Contact Stability for a Surface Patch (at COP) [20]). *A rectangular surface patch in the fixed contact mode is (weak) contact stable if and only if its resultant force* $\boldsymbol{f}$ *and resultant yaw moment* $\boldsymbol{\tau}_{C,z}$ *at its center of pressure* (x_C, y_C) *satisfy*

$$|f_x|, |f_y| \leq \mu f_z \tag{10}$$

$$|x_C|, |y_C| \leq X, Y \tag{11}$$

$$|\tau_C^z - \tau_{C,\text{safe}}^z| \leq \mu \, d_{\text{vert}}(x_C, y_C) \, f_z, \tag{12}$$

where $\tau_{C,\text{safe}}^z = \max_{i=1}^4 \overrightarrow{CC_i} \times \boldsymbol{f}_t$ *and* $d_{\text{vert}}(x_C, y_C)$ *is the distance from the COP to the nearest vertex of the contact surface.*

We see that the CWC, i.e., the friction cone applying to the resultant contact wrench, has a separable structure:

- The first inequalities (10) correspond to Coulomb friction applied to the resultant force. (Note how they also imply that $f^z > 0$.)
- Inequalities (11) constraint the center of pressure (COP) [85] to lie inside of the support polygon.
- The last inequality (12) provides a "safe" yaw torque value, from which the actual yaw torque should not deviate by more than $\mu \, d_{\text{vert}}(x_C, y_C) \, f_z$.

Note how zero may not be a solution to the yaw constraint (12), meaning that sometimes a nonzero yaw-compensating moment is required at the COP to prevent undesired yaw rotations. This behavior goes against the assumption, made in some yaw compensation methods [30, 89], that zero is the most desired yaw moment to achieve.

2.2 Multi-body Contact Stability

2.2.1 Newton-Euler Equations

Let m and G represent the total mass and center of mass (COM) of the robot. We write $\boldsymbol{p}_A$ the vector of absolute coordinates of a point A. For a link k, define:

- m_k the total mass of the link;
- $\boldsymbol{p}_{G_k}$ the vector of absolute coordinates of its COM G_k;
- $\mathbf{R}_k$ its orientation matrix in the absolute frame;
- $\boldsymbol{\omega}_k$ its angular velocity in the link frame;
- $\mathbf{I}_k$ its inertia matrix in the link frame.

The linear momentum $\mathbf{P}$ and angular momentum $\mathbf{L}_G$ of the robot, taken at the COM G, are defined by

$$\mathbf{P} \overset{\text{def}}{=} \sum_{\text{link } k} m_k \dot{\boldsymbol{p}}_{G_k}, \tag{13}$$

$$\mathbf{L}_G \overset{\text{def}}{=} \sum_{\text{link } k} m_k \overrightarrow{GG_k} \times \dot{\boldsymbol{p}}_{G_k} + \mathbf{R}_k \mathbf{I}_k \boldsymbol{\omega}_k. \tag{14}$$

The fundamental principle of dynamics states that the rate of change of the momentum is equal to the total wrench of forces acting on the system:

$$\begin{bmatrix} \dot{\mathbf{P}} \\ \dot{\mathbf{L}}_G \end{bmatrix} = \begin{bmatrix} m\boldsymbol{g} \\ \mathbf{0} \end{bmatrix} + \sum_{\text{contact } i} \begin{bmatrix} \boldsymbol{f}_i \\ \overrightarrow{GC}_i \times \boldsymbol{f}_i \end{bmatrix}, \tag{15}$$

This equation is called the Newton-Euler equations of the system (Fig. 6). It can be equivalently derived from Gauss's principle of least constraint and corresponds to the six unactuated components in the equations of motion of the system robot + environment [96].

The *resultant contact wrench* exerted on the system is

$$\boldsymbol{w}_O^c \overset{\text{def}}{=} \begin{bmatrix} \boldsymbol{f}^c \\ \boldsymbol{\tau}_O^c \end{bmatrix} \overset{\text{def}}{=} \sum_{\text{contact } i} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{p}_{C_i} \times \boldsymbol{f}_i \end{bmatrix} \tag{16}$$

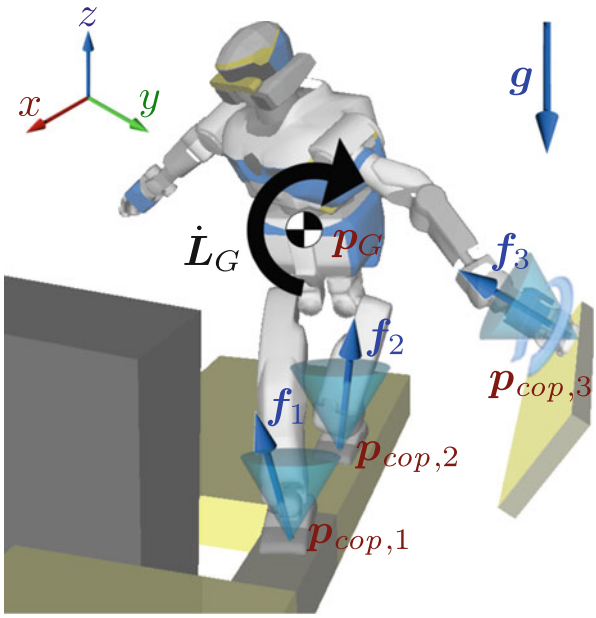

Fig. 6 The Newton-Euler Eq. (15) describes how contact wrenches sum to the resultant wrench acting on the robot's free-floating coordinates (Illustration adapted from [3])

The *gravito-inertial wrench* of the robot is

$$\boldsymbol{w}_O^{gi} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{f}^{gi} \\ \boldsymbol{\tau}_O^{gi} \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} m(\boldsymbol{g} - \ddot{\mathbf{p}}_G) \\ \boldsymbol{p}_G \times m(\boldsymbol{g} - \ddot{\mathbf{p}}_G) - \dot{\mathbf{L}}_G \end{bmatrix}. \tag{17}$$

These two wrenches are opposite from the Newton-Euler equation (15). The second one is computed from the robot's motion, while the first one results from contact forces, yet interestingly they characterize the same underlying quantity.

2.2.2 Multi-contact Wrench Cone

Coulomb friction models how a local motion constraint, a fixed contact point, yields a set of dual inequality constraints, the complementary contact force lying in the friction cone (Motion and force vectors are always dual in the sense of linear algebra; see, e.g., Chapter 2 in [39].). In the presence of multiple contact points on a robot body, these forces add up into a resultant contact wrench on the body. In turn, when multiple links are in contact, these wrenches add up into the resultant contact wrench exerted on the whole robot. Interestingly, it is possible to compute an equivalent friction for this wrench:

Proposition 5 (Multi-contact Wrench Cone [79]). *A motion is (weak) contact stable if the resultant contact wrench $\boldsymbol{w}_O^c$ of all contact forces exerted on the robot lies in the multi-contact wrench cone (MCWC), written*

$$\mathbf{A}_O \boldsymbol{w}_O^c \leq \mathbf{0} \Leftrightarrow \mathbf{A}_O \begin{bmatrix} m(\ddot{\mathbf{p}}_G - \boldsymbol{g}) \\ \boldsymbol{p}_G \times m(\ddot{\mathbf{p}}_G - \boldsymbol{g}) + \dot{\mathbf{L}}_G \end{bmatrix} \leq \mathbf{0}. \tag{18}$$

Note the subscript O in $\mathbf{A}_O$: the matrix depends on the point where the resultant contact wrench is taken.

This contact-stability criterion was initially introduced in 2003 by Saida et al. [83] in an early attempt to generalize the ZMP support areas to non-coplanar contacts. It was later extended by Hirukawa et al. [49], but without the explicit computation of the criterion's matrix $\mathbf{A}_O$ which Qiu et al. [79] introduced 5 years later. This computation can be carried out using algorithms from polyhedral geometry, which we will now review briefly.

Recall that the convex hull of a set of vectors is $conv(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n) = \sum_i \alpha_i \boldsymbol{v}_i$ where all $\alpha_i > 0$ and $\sum_i \alpha_i = 1$. Similarly, the positive span of a set of vectors is $nonneg(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_s) = \sum_i \lambda_i \boldsymbol{r}_i$ where all $\lambda_i > 0$.

Theorem 1 (Minkowski-Weyl). *For a subset P of $\mathbb{R}^d$, the following statements are equivalent:*

1. *P is a polyhedron: $P = \{\boldsymbol{x} : \mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ for $\mathbf{A} \in \mathbb{R}^{m \times d}$ and $\boldsymbol{b} \in \mathbb{R}^m$*
2. *There are finite real vectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ and $\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_s$ in $\mathbb{R}^d$ such that*

$$P = conv(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n) + nonneg(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_s). \tag{19}$$

Polyhedra have thus two representations: the half-space (or H-)representation $\mathbf{A}\boldsymbol{x} \leq \boldsymbol{b}$, and the vertex (or V-)representation as convex hull of base vertices plus positive combinations of rays. Each has its own advantages. For instance, checking whether a vector $\boldsymbol{x}$ belongs to a cone is fast in H-representation (check each inequality) but amounts to solve a linear program (LP) in V-representation. Converting from H- to V-representation is known as the vertex enumeration problem, while the reciprocal from V- to H-representation is the facet enumeration problem. The *double-description method* [41] is an algorithm that allows the conversion from one representation to the other, solving for both problems at once. It has been readily applied in [7,23,35].

Let us consider the vertex representation of the linearized friction cones depicted by Fig. 4:

$$\boldsymbol{f}_i = \sum_{\text{ray } j} \lambda_{ij}\, \boldsymbol{u}_{ij}, \quad \lambda_{ij} \geq 0, \quad \boldsymbol{u}_{ij} \in \boldsymbol{n}_i + \mathscr{P}_4 = \left\{ \boldsymbol{n}_i \pm \frac{\mu_i}{\sqrt{2}}\, \boldsymbol{t}_i \pm \frac{\mu_i}{\sqrt{2}}\, \boldsymbol{b}_i \right\}. \tag{20}$$

(Replace $\mathscr{P}_4$ by $\mathscr{P}_n$ for finer approximations.) From Eq. 16, one can then write the resultant contact wrench as

$$\boldsymbol{w}_O^c = \sum_{\text{contact } i} \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{p}_{C_i} \times \boldsymbol{f}_i \end{bmatrix} = \sum_{\text{rays } i,j} \lambda_{ij} \begin{bmatrix} \boldsymbol{u}_{ij} \\ \boldsymbol{p}_{C_i} \times \boldsymbol{u}_{ij} \end{bmatrix}. \tag{21}$$

It is therefore straightforward to compute the V-representation $\boldsymbol{w}_O^c \in nonneg(\{\boldsymbol{v}_{O,ij}\})$ of the MCWC using the vectors $\boldsymbol{v}_{O,ij}$ defined as $[\boldsymbol{u}_{ij} \;\; \boldsymbol{p}_{C_i} \times \boldsymbol{u}_{ij}]$. The matrix $\mathbf{A}_O$ given in Proposition 5 can finally be computed [23] by facet enumeration:

$$\mathbf{A}_O = \textsc{EnumerateFacets}(\{\boldsymbol{v}_{O,ij}\}). \tag{22}$$

2.2.3 Contact-Stability Areas and Volumes

The MCWC encodes support areas and volumes for a variety of behaviors, three of which are depicted in Fig. 7:

- **Static equilibrium:** Bretl and Lall [18] showed how immobility, i.e., static equilibrium, can only be achieved if the horizontal projection of the COM lies in a polygon whose shape depends only on contact locations (Fig. 1, left). They also introduced a ray casting algorithm that leverages the small output dimension (2D, versus 6D for the complete MCWC). This algorithm has since been extended [2,32] and applied to compute similar polygons related to time-optimal retiming of humanoid motions [44,77].
- **ZMP support areas:** A common mode used in locomotion controllers is the Linear-Inverted Pendulum Mode (LIPM) [52], where the angular momentum and COM height are kept constant ($\dot{\mathbf{L}}_G = \mathbf{0}$ and $\ddot{z}_G = 0$). In this mode, COM dynamics are related to the ZMP, the point defined from the resultant wrench by [85]

Fig. 7 Three types of support areas and volumes encoded in the contact wrench cone. *Left*: the right cylinder of COM positions that characterizes static equilibrium [18]. *Middle*: support areas for the ZMP taken in virtual planes, which can be used for multi-contact locomotion in the Linear-Pendulum Mode [25]. *Right*: cone of feasible COM accelerations, used for multi-contact locomotion in the Pendulum Mode [21]

$$\boldsymbol{p}_Z = \frac{\boldsymbol{n} \times \boldsymbol{\tau}_O^c}{\boldsymbol{n} \cdot \boldsymbol{f}} + \boldsymbol{p}_O, \tag{23}$$

with $(O, \boldsymbol{n})$ the coordinates of the ZMP plane (As pointed out by Sardain and Bessonnet, the ZMP is actually a non-central axis of the resultant wrench. Intersection by a plane yields a point.). COM and ZMP are coupled via the differential equation $\ddot{\mathbf{p}}_G = \frac{g}{h}(\boldsymbol{p}_Z - \boldsymbol{p}_G)$, with g the gravity constant and h the algebraic height $\boldsymbol{n} \cdot (\boldsymbol{p}_G - \boldsymbol{p}_O)$ of the COM with respect to the ZMP plane. A well-known condition to avoid tilting while walking on horizontal floors is that the ZMP lies within the convex hull of ground contact points [1]. It generalizes to multi-contact locomotion by applying the double description to the MCWC with the additional constraints of the LIPM [25] (Fig. 2, middle).

- **COM acceleration cone:** maintaining the COM in a fixed plane was reasonable for walking on a single surface but becomes a stringent constraint in multi-contact. It can actually be removed altogether by only considering the angular-momentum condition ($\dot{\mathbf{L}}_G = \mathbf{0}$) of the Pendulum Mode. Reducing this constraint again into the MCWC yields a 3D cone of feasible COM accelerations [21] which can be used to control the COM position in space, as we will detail in Sect. 5.5.

These criteria have been applied in both planning and control. For instance, static-equilibrium polygons have been used in planning where trajectories are joined at statically stable postures [17, 35, 46], while ZMP support areas have been used for walking pattern generation [97] and online footstep planning on horizontal floors [47].

3 Contact Planning for Multi-contact Motions

The motion planning problem in robotics has long been formalized as a path planning problem for a point in the C-space, i.e., the problem of connecting two points with a continuous path [59, 61]. This formulation is particularly suitable for

obstacle collision-avoidance purposes, for either mobile robots (wheeled or legged, generally non-holonomic) evolving in the 2D plane or for fixed-base manipulators evolving in a 3D cluttered environment. Motion planning problems for mobile manipulators are solved by decoupling the two components (mobile 2D base and holonomic 3D manipulator). In this section we present a method where the whole "mobile motion" is planned at once, including the motion of the locomotion support limbs, based on 3D obstacle avoidance and whole-body dynamics going through changing contact states. The resulting planning framework displays the following three key features:

- Autonomous planning of the contact-state switching sequence for any motion (including basic walking);
- Collision avoidance (including self-collision) for the whole body of the robot;
- Whole-body dynamics satisfied throughout the motion, including contact-physics consistency and handling uncertainties and model errors.

Whole-body dynamics-consistent motion on fixed contact states have been addressed in [80, 81, 84, 86]. They achieve motions that are accomplished either on fixed (multi-)contact stances or according to a user-prescribed contact sequence or to a reduced-model (e.g., LIPM)-based cyclic walking pattern. In what follows, we will see an approach that adds autonomous contact-sequence planning to this feature, resulting in a *contact-first* motion planning framework. The approach presented in this section is the one mainly established by the authors in [7, 8, 10, 12, 33–36]. Other similar contact-first approaches include the works [17, 45, 46].

Let $\mathscr{C} = SE(3) \times \mathscr{C}^{\text{joints}}$ denote the configuration space of the robot, including the 6D free-floating base which evolves in $SE(3)$. For every configuration $\boldsymbol{q}$ of the robot, let $\sigma(\boldsymbol{q})$ denote the *stance* (contact state) of the robot when put in that configuration. $\sigma(\boldsymbol{q})$ is a set of contacts, e.g., $\sigma(\boldsymbol{q}) = \{c_1, c_2\}$ if the configuration $\boldsymbol{q}$ puts the robot in a double support stance with both feet on the floor. Each contact c_i encodes the two robot/environment surfaces that are in contact (e.g., foot surface/floor surface) as well as their relative position/orientation. See Table 1 for typical surfaces that can be designated on the robot and the environment. The *bilateral* tag on some of these surfaces will be explained thereafter.

Table 1 Typical examples of surfaces considered in humanoid contact planning

Robot surfaces	Environment surfaces
Foot surfaces (soles)	Ground surfaces (flat, slopped, or uneven)
Hand surfaces	Stair surfaces
Gripper surfaces (*bilateral*)	Desk surfaces
Knee, elbow, and forearm surfaces (for crawling)	Sitting chair surfaces
Sitting surface	Ladder rung surfaces (*bilateral*)
(Bottom of the waist or back of the legs)	Steering wheel of a car (*bilateral*)

3.1 Guide Path Planning

Let $\boldsymbol{q}_s$ and $\boldsymbol{q}_g$ denote, respectively, the start and goal configurations of the humanoid motion planning problem. In a preprocessing stage, we first plan a *guide path* from $\boldsymbol{q}_s$ to $\boldsymbol{q}_g$ in $\mathscr{C}_{\text{free}}$ using off-the-shelf robotics collision-free path planning software. Along this guide path, the humanoid is freely floating without any contact with the environment but keeps some of its limbs close enough to the environment components so as to be able to enforce static equilibrium if contacts were to be established. This is done by adequate biasing of the random configuration sampling method in a reduced-dimensional C-space [7]. We denote this guide path $p_{\text{guide}} : [0,1] \rightarrow \mathscr{C}_{\text{free}}$, and its image in $\mathscr{C}$, $\mathscr{P}_{\text{guide}} = \{p_{\text{guide}}(t) \mid t \in [0,1]\}$. Using probabilistic roadmap [53] or rapidly exploring random tree (RRT) [60] techniques without smoothing, the guide $\mathscr{P}_{\text{guide}}$ turns out to be made of linear segments interpolating a finite sequence of *milestone* configurations $(\boldsymbol{q}_k^{\text{milestone}})_k$, as schematically illustrated in Fig. 8. Alternatively, this sequence of milestone configurations can be specified manually by the user, if needed, by roughly selecting a set of basic postures of the robot (standing, sitting, four-legged, etc.) and then roughly positioning them on selected key waypoints along the expected motion path in the 3D workspace.

3.2 Inverse Kinematics-and-Statics Solver

For a given set of contact $\{c_i\}_i$, the role of the inverse kinematics-and-statics solver [8, 9] is to find a configuration of the robot $\boldsymbol{q}$ that satisfies these contacts, i.e., such that $\sigma(\boldsymbol{q}) = \{c_i\}_i$, with additional constraints that model the physical limitations of the robot. These constraints are listed as follows:

1. The robot should be in static equilibrium under the action of (a) contact forces $\{(\boldsymbol{f}_{ij})_j\}_i$ applied at contacts $\{c_i\}_i$, (In this setting, we consider that each contact area corresponding to a contact c_i is a polygon with vertices $(C_{ij})_j$ and on each vertex C_{ij} a contact force $\boldsymbol{f}_{ij}$ is applied. Recall from Sect. 2.1 how this model suffices to described surface contacts by including one contact point for each vertex of the contact area (Proposition 2).), (b) its actuators' torques, and (c) the gravity.
2. Actuation torques are limited to their nominal maximal values.
3. Joint positions are limited to the robot's range of motion.
4. Contact forces are restricted to lie inside their Coulomb friction cones (Proposition 2). In the case of *bilateral contacts* (between two *bilateral*-tagged surfaces, e.g., between the hand gripper and a ladder rung), this constraint is dropped.
5. All the bodies of the robot, except the ones contributing with a surface to one of the contacts $\{c_i\}_i$, must not be in collision with the environment nor with the other bodies of the robot.

To find a configuration satisfying all of these constraints, a nonlinear constrained optimization problem is usually expressed on the configuration and contact force

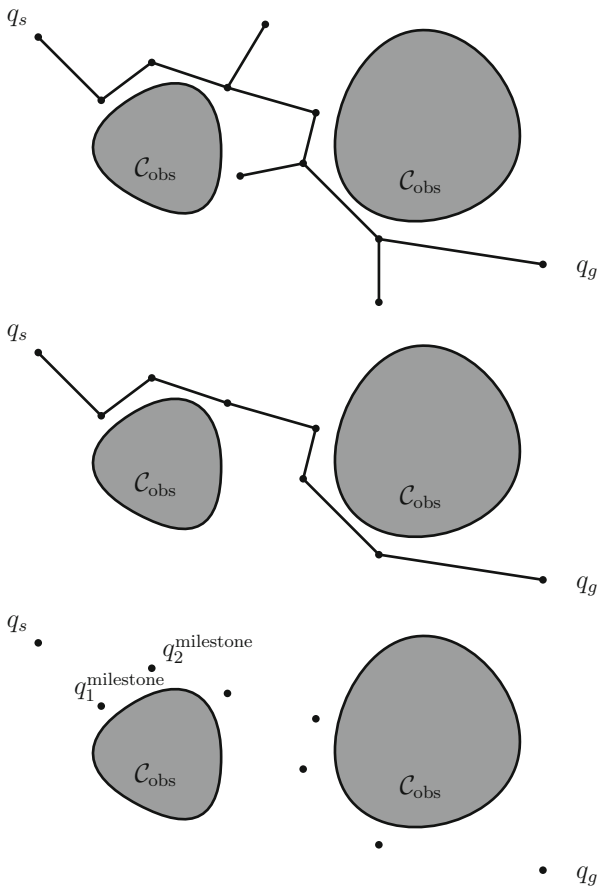

Fig. 8 *Top*: guide path planning using RRT staying close to the environment. *Middle*: the guide path is the raw (unsmoothed) result of the RRT. *Bottom*: milestone configurations extracted from the resulting path (vertex configurations of the path); the milestone configurations can alternatively be manually input by the user

variables simultaneously, which minimizes a cost (objective) function $\text{obj}(\boldsymbol{q}, \boldsymbol{f}_{ij})$ under the constraint $\sigma(\boldsymbol{q}) = \{c_i\}_i$ (geometric contact constraint) along with Constraints 1 to 5. In the next subsection, we detail this objective function. The IPOPT software [69, 94] can be used as optimization solver for this class of problems. Recently, a more dedicated solver was proposed to take into account that the optimization problem is more naturally formulated on non-Euclidean manifolds [19].

The contact points $\{c_i\}_i$ described here are all fixed in the 3D workspace, and contact forces $\boldsymbol{f}_{ij}$ are applied at all of them. We additionally consider contacts that are fixed but not used for maintaining static equilibrium, i.e., contacts that are positioned on the environment but with zero contact forces applied at them

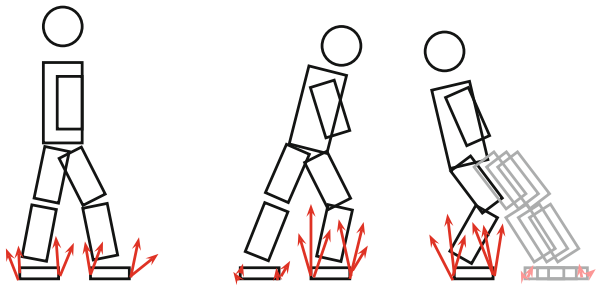

Fig. 9 Different types of contact configurations. *Left*: $\{c_1, c_2\}$. *Middle*: $\{\mathring{c}_1, c_2\}$. *Right*: $\{c_1, \bar{c}_2\}$. In the *middle figure*, the contact with the rear foot is geometrically established, but no forces are applied on it (we represent this situation in the figure with almost zero force). Same with the contact of the front foot in the *right figure*, for which the relative $SE(2)$ position/orientation between the floor and floor surface is not a priori decided

$(\forall j \ \boldsymbol{f}_{ij} = 0)$. Let us denote such contacts $\mathring{c}_i$. Finally, we also consider contacts that are prescribed between a pair of surfaces but without specifying their relative position/orientation and without contact forces applied at them. We denote such contacts $\bar{c}_i$. This type of contacts corresponds to situations in which the robot wants to put a surface in contact with an environment surface but without a priori knowing where to position its surface on the environment surface. The objective function of the optimization will let the solver decide this position orientation, as depicted on Fig. 9.

As for Constraint 5, a geometric model of the robot with strictly convex Sphere-Torus-Patch Bounding Volumes (STP-BV) [5] ensures the continuity of the gradient of distance to non-strictly convex environment bodies and, using the GJK algorithm [42], allows the definition of interpenetration distance that is required by the optimization algorithm to compute constraint gradients. The environment also needs to be decomposed into convex components to allow for collision checking between all pairs of robot and environment bodies.

3.3 Computing the Contact-Switching Sequence

To go from $\boldsymbol{q}_s$ to $\boldsymbol{q}_g$ in the presented approach, it is necessary to first plan the contact-switching sequence that will make the motion possible and the goal reachable [10]. To do so, we consider the set of contacts of these two configurations, respectively, $\sigma(\boldsymbol{q}_s)$ and $\sigma(\boldsymbol{q}_g)$, and we search for a sequence of intermediate stances

$$\big(\sigma_0 = \sigma(\boldsymbol{q}_s),\ \sigma_1,\ \sigma_2,\ \ldots,\ \sigma_{N-1},\ \sigma_N = \sigma(\boldsymbol{q}_g)\big),$$

such that :

1. Every stance either adds or removes one contact from the previous one

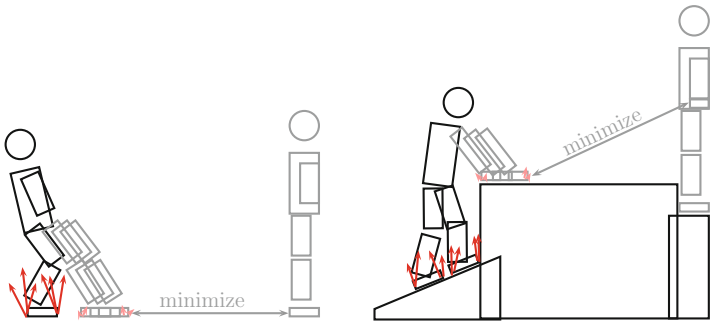

Fig. 10 Body contact position minimization component of the optimization objective function when adding a contact (i.e., necessarily of type $\bar{C}$ according to Algorithm 1). In both examples the *gray configuration* in the right is the next milestone configuration in the search process

2. There exists a common transition configuration that simultaneously satisfies any two consecutive stances in the sequence.

Algorithm 1 is an example of best-first search approach that outputs such a sequence, ensuring that the two conditions above are realized.

When adding a non-fixed contact $\bar{c}$ between a body $\mathscr{B}$ of the robot and the environment (Fig. 10), the objective function is

Algorithm 1 Contact-Switching Planning Algorithm

initialize a priority queue Q and a search tree T with the initial stance $\sigma(\boldsymbol{q}_s)$
repeat
  pop best stance σ_* from Q
  for all possible pairs of robot/environment surfaces (the ones that are not already forming a contact in σ_*) **do**
    define a non-fixed zero-force contact $\bar{c}$ between the two surfaces
    call the inverse kinematics-and-statics solver...
    ...on the set of contacts $\sigma_* \cup \{\bar{c}\}$ with the objective function obj_1
    if a configuration $\boldsymbol{q}$ is found **then**
      push $\sigma(\boldsymbol{q})$ into Q and add the node $\sigma(\boldsymbol{q})$ to T with an edge drawn from σ_* to $\sigma(\boldsymbol{q})$
    end if
  end for
  for all contacts c that are already in σ_* **do**
    define a contact $\mathring{c}$ that is fixed at the same position as c but with zero contact force
    call the inverse kinematics-and-statics solver...
    ...on the set of contacts $\sigma_* \setminus \{c\} \cup \{\mathring{c}\}$ with the objective function obj_2
    if a configuration $\boldsymbol{q}$ is found **then**
      push $\sigma(\boldsymbol{q})$ into Q and add the node $\sigma(\boldsymbol{q})$ to T with an edge drawn from σ_* to $\sigma(\boldsymbol{q})$
    end if
  end for
until a stance close enough to $\sigma(\boldsymbol{q}_g)$ is found

$$\text{obj}_1(\boldsymbol{q}, \boldsymbol{f}) = \alpha \|\boldsymbol{q} - \mathscr{P}_{\text{guide}}\|^2_{\mathscr{C}} + \beta \|\mathscr{B}(\boldsymbol{q}) - \mathscr{B}(\boldsymbol{q}_\kappa^{\text{milestone}})\|^2_{\mathbb{R}^3} + \gamma \|\boldsymbol{f}\|^2_{\mathbb{R}^{\dim(\boldsymbol{f})}}, \quad (24)$$

while when removing a contact c (i.e., turning it into $\mathring{c}$), it becomes

$$\text{obj}_2(\boldsymbol{q}, \boldsymbol{f}) = \alpha \|\boldsymbol{q} - \mathscr{P}_{\text{guide}}\|^2_{\mathscr{C}} + \gamma \|\boldsymbol{f}\|^2_{\mathbb{R}^{\dim(\boldsymbol{f})}}. \quad (25)$$

(where $\|X - Y\|_Z$ denotes the multidimensional Euclidean distance between two entities X and Y in the space Z). In both cases, we can use the sequence of milestone configurations $(\boldsymbol{q}_k^{\text{milesone}})_k$ instead of $\mathscr{P}_{\text{guide}}$ and replace the term $\|\boldsymbol{q} - \mathscr{P}_{\text{guide}}\|^2_{\mathscr{C}}$ with $\|\boldsymbol{q} - \boldsymbol{q}_\kappa^{\text{milestone}}\|^2_{\mathscr{C}}$ where κ keeps track of the currently targeted milestone ($\kappa - 1$ being the last reached milestone).

Algorithm 1 produces a contact-switching sequence $(\sigma_0, \ldots, \sigma_N)$ going from $\sigma(\boldsymbol{q}_s)$ to $\sigma(\boldsymbol{q}_g)$, as well as the sequence of configurations that were found upon calls to the inverse kinematics-and-statics solver $(\boldsymbol{q}_0, \ldots, \boldsymbol{q}_N)$, with $\boldsymbol{q}_0 = \boldsymbol{q}_s$ and where, for all $i \geq 1$, $\boldsymbol{q}_i$ is the transition configuration found between σ_{i-1} and σ_i (see Fig. 11). Another algorithm is presented following a sampling-based approach in [46]. A comparison between the two algorithms can be found in [10].

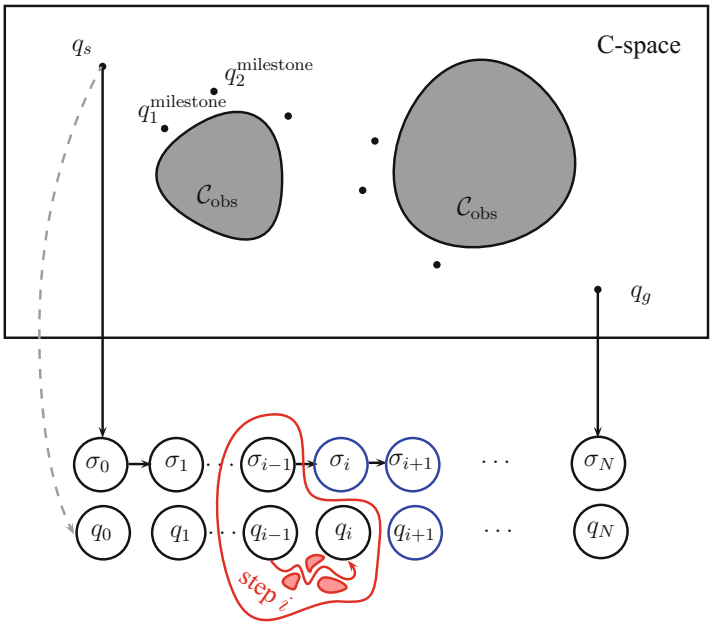

Fig. 11 A schematic representation of the contact-switching sequence. In *blue*: Configuration $\boldsymbol{q}_{i+1}$ is generated as a transition configuration (removing a contact or adding a contact) from the contact set σ_i to the contact set σ_{i+1}. In *red*, the motion of step i that "interpolates" the configurations $\boldsymbol{q}_{i-1}$ and $\boldsymbol{q}_i$ with controlled contact forces applied at the contacts of the set σ_{i-1}. The motion of step i occurs in the contact submanifold of the C-space that corresponds to the fixed placements of the contacts σ_i

4 Control of Multi-contact Motions

Multi-contact control allows to convert the off-line-computed sequences $(\sigma_0, \ldots, \sigma_N)$ and $(q_0, \ldots, q_N)$ of stances and robot postures into a whole-body motion. The conversion will be done online by coupling a multitask whole-body controller with a finite-state machine that follows the discrete logic of the motion. The approach presented in this section is the one mainly proposed by the authors [11, 13, 90, 91] among other similar approaches in the field.

4.1 Finite-State Machine

The finite-state machine (FSM) keeps track of the current step $i \in \{1, \ldots, N\}$ in the sequence (step i is between the configurations $\boldsymbol{q}_{i-1}$ and $\boldsymbol{q}_i$), as well as the nature of the step, i.e., whether it is a step that removes a contact or a step that adds a contact. Overall, the FSM consists of two meta-states:

1. State "Move the COM": when the current step is removing a contact between σ_{i-1} and σ_i. We define a COM task to track the 3D position of the COM at configuration $\boldsymbol{q}_i$ and a configuration task to track the configuration $\boldsymbol{q}_i$. The resulting motion is meant to zero the contact forces on the contact being removed (the contact stays fixed at its position throughout the step).
2. State "Move the Contact Body": when the current step is adding a contact between σ_{i-1} and σ_i. Additionally to the two tasks above, we define a 6D position and orientation task on the body that is going to make contact to track its position and orientation at configuration $\boldsymbol{q}_i$, possibly going on the way through a user-input *waypoint* at the middle of the step.

The COM task allows the robot to keep balance, the body position and orientation task allow steering of the moving body to make contact, and the posture tasks solves for the redundancy and the remaining joints motion. When the tasks are achieved for the current step i, the FSM moves to step $i + 1$, until the final configuration $\boldsymbol{q}_N$ is reached.

4.2 Weighted Multitask Whole-Body Controller

At every control time step t, the multi-contact motion controller is given a set of tasks $(\tau_j)_j$ with Jacobians $(\mathbf{J}_{\tau_j})_j$, weights $(w_j)_j$, and stiffnesses $(k_j)_j$. A task is typically a residual $\boldsymbol{\tau}_j(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{p}_{\text{target}} - \boldsymbol{p}_{\text{robot}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ over a desired robot behavior such as end-effector position, COM position, angular momentum, etc. The controller is fed back with the current state of the robot $(\boldsymbol{q}, \dot{\boldsymbol{q}})$. It solves for the acceleration $\ddot{\boldsymbol{q}}$, the stacked vector of contact forces $\boldsymbol{f}$, and the actuation torques $\boldsymbol{u}$, through the following quadratic program (QP):

$$\min_{(\ddot{\boldsymbol{q}},\boldsymbol{f},\boldsymbol{u})} \sum_j w_j \| \overbrace{\mathbf{J}_{\tau_j}\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}_{\tau_j}\dot{\boldsymbol{q}}}^{\ddot{\tau}_j} + 2\sqrt{k_j}\dot{\tau}_j + k_j\tau_j \|^2 \tag{26}$$

under a number of linear equality and inequality constraints on the optimization variables $(\ddot{\boldsymbol{q}}, \boldsymbol{f}, \boldsymbol{u})$:

- Whole-body equations of motion of the robot:

$$\mathbf{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{N}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{S}^T \boldsymbol{u} + \mathbf{J}^T \boldsymbol{f}, \tag{27}$$

  where $\mathbf{S}$ is a selection matrix for the actuated joints in $\boldsymbol{q}$ (i.e., excluding the 6D free-flying base) and $\mathbf{J}$ is the stacked contact Jacobian (That is to say: if $(C_1, \ldots, C_K)$ denotes the sequence of (environment) contact points in the order in which contact forces are stacked in $\boldsymbol{f}^T = [\boldsymbol{f}_1^T \cdots \boldsymbol{f}_K^T]$, then J stacks the corresponding contact Jacobians $\mathbf{J}_{C_i}(q)$ in this order $J^T = [\mathbf{J}_{C_1}(q)^T \cdots \mathbf{J}_{C_K}(q)^T]$, where $\mathbf{J}_{C_i}(q)$ is the translation Jacobian of the point of the robot body that needs to come into contact with C_i.) (those of the set σ_{i-1} during the execution of step i; see Fig. 11).
- Kinematic (fixed) contact condition:

$$\mathbf{J}\ddot{\boldsymbol{q}} + \dot{\mathbf{J}}\dot{\boldsymbol{q}} = 0 \tag{28}$$

- Actuation limits:

$$-\boldsymbol{u}_{\max} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \tag{29}$$

- Contact stability: forces $\boldsymbol{f}$ lie within their linearized friction cones (As in the contact planning phase, we drop this constraint as a first approximation in case of a bilateral contact (involving a grasp with the gripper; see Fig. 13).)

$$\boldsymbol{f} = \lambda^T K, \ \lambda \geq 0, \tag{30}$$

  where K is the stacked matrix of V-representation of the local contact wrench cones and λ are a set of positive multipliers, one for each friction ray. This constraint can be equivalently cast using the H-representation of local contact wrenches:

$$\mathbf{F}\boldsymbol{f} \leq 0, \tag{31}$$

  where $\mathbf{F} = \mathrm{diag}(\mathbf{A}\mathbf{R}_1^T, \ldots, \mathbf{A}\mathbf{R}_K^T)$ is the H-representation matrix of the (product) friction cone for $\boldsymbol{f}$, $\mathbf{A}$ denoting the matrix of the single-body contact wrench cone given in Proposition 3 and $\mathbf{R}_i$ the rotation matrix of the surface at contact C_i.

As an output we get both $\ddot{\boldsymbol{q}}$ and $\boldsymbol{u}$. In the case of motion generation, joint accelerations $\ddot{\boldsymbol{q}}$ can be simply integrated to produce a continuous trajectory $\boldsymbol{q}(t)$. For control, the choice will depend on the ensuing hardware or simulation software: for torque control, the vector $\boldsymbol{u}$ can be used straight away as a reference, while for position control, joint accelerations $\ddot{\boldsymbol{q}}$ can be integrated into a reference position $\boldsymbol{q}_{\mathrm{ref}}$.

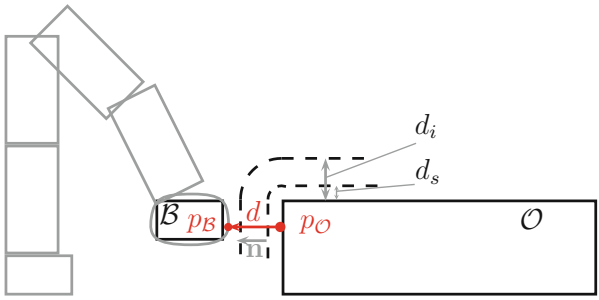

Fig. 12 Velocity damper. d_s represents the security distance around the obstacle $\mathcal{O}$, that the robot body $\mathcal{B}$ will not penetrate. d_i represents the influence distance around the obstacle $\mathcal{B}$, when the body $\mathcal{B}$ enters the influence zone, the velocity-damper mechanism starts to act and to deviate the trajectory of the body $\mathcal{B}$

4.2.1 Collision Avoidance

To integrate collision avoidance into a QP formulation, one can follow the method of Faverjon and Tournassoud [38] (Fig. 12). A collision-avoidance constraint between a robot body $\mathcal{B}$ and an environment obstacle $\mathcal{O}$ is expressed as follows:

$$\dot{d} + \bar{t}\,\ddot{d} \;\geq\; \xi \frac{d - d_s}{d_i - d_s} \tag{32}$$

where:

- d is the distance between $\mathcal{B}$ and $\mathcal{O}$. This distance is computed using the GJK algorithm on STP-BV bounding volume of $\mathcal{B}$ and assuming $\mathcal{O}$ is convex (recall that in Sect. 3.2, we needed to decompose the environment into convex components). This ensures that the witness points, i.e., the points such that $d = \| p_{\mathcal{B}} - p_{\mathcal{O}} \|$, vary continuously on the surfaces of the bodies.
- d_s is the security distance, under which we consider that collision happens.
- d_i is the influence distance, a threshold under which the damping in activated.
- ξ is a damping coefficient.
- $\bar{t}$ is the control time step.

This constraint is linear in $\ddot{q}$ by noting that

$$\dot{d} = \mathbf{n}^T \mathbf{J}_{p_{\mathcal{B}}}\, \dot{\boldsymbol{q}} \tag{33}$$

$$\ddot{d} = \mathbf{n}^T \left(\dot{\mathbf{J}}_{p_{\mathcal{B}}}\, \dot{\boldsymbol{q}} + \mathbf{J}_{p_{\mathcal{B}}}\, \ddot{\boldsymbol{q}} \right) \tag{34}$$

where $\mathbf{n}$ is the unit vector from $p_{\mathcal{O}}$ to $p_{\mathcal{B}}$. $\mathbf{J}_{p_{\mathcal{B}}}$ is the Jacobian at the point *fixed in the body frame* that coincides with $p_{\mathcal{B}}$ at the current instant [62]. Self-collision between two bodies $\mathcal{B}_1$ and $\mathcal{B}_2$ is handled similarly, only replacing $\mathbf{J}_{p_{\mathcal{B}}}$ with

$\mathbf{J}_{p_{\mathscr{B}_1}} - \mathbf{J}_{p_{\mathscr{B}_2}}$ in the expression above. Joint positions and velocities limits are handled with similar velocity-damper formulations.

4.2.2 Waypoints

In order to avoid local minima in which the obstacle-avoiding body might get trapped, we additionally define waypoints along the motion of the body going to make contact. The meta-state "Move Contact Body" is subdivided into two consecutive states:

1. "Go to waypoint"
2. "Go to goal contact location"

These waypoints are also crucial in resolving the apparent contradiction between moving a body to a desired contact spot and at the same time avoiding collision between that body and the environment component on which the contact is defined. For example, in a stair-climbing motion, the foot that goes to contact location on the top of the stair must at the same time not collide with the edge of the stair while executing its movement. The solution to that is, for the body going to make the contact, to activate the collision-avoidance constraint (velocity damper) in the first phase of the motion "go to waypoint," and once the waypoint is reached to deactivate this constraint in the second phase, "go to goal contact location," thus allowing the body to make the landing on the environment component. Waypoints can be either defined automatically using user-defined parameters of the motion (e.g., step height) or simply given manually by the user.

4.2.3 Bilateral Contacts: Case of Grasping Contacts with a Gripper

We have assumed so far that the motion was made using only unilateral contacts. Contacts made with special kind of end effectors of the robot, such as grippers, can be dealt with as follows. First, define a regular contact surface on the palm of the gripper. When this surface comes in contact with a "graspable" surface of the environment (e.g., tagged as such by the user), then the contact is assumed to be bilateral (the two surfaces are bilaterally stuck together) by dropping the constraints on the friction cones that we consider for regular contact. See Fig. 13. Using this approximated information, precise movement of the end effector to realize the grasp is subsequently handled by special chain of states in which the FSM goes whenever the current step i adds or removes a contact involving these grippers:

1. "Add gripper"
2. "Close gripper"
3. "Open gripper"
4. "Remove gripper"
5. "Move gripper to waypoint"

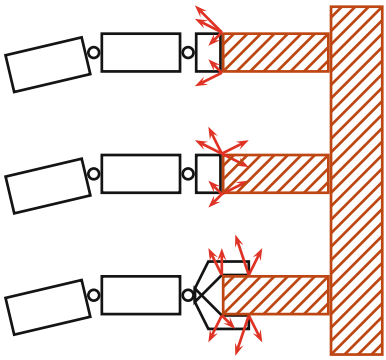

Fig. 13 *Top*: regular contact with friction cone constraint. *Middle*: approximated grasp contact without constraint when the search tree reaches a "graspable"-tagged surface. *Bottom*: Fine control of the gripper with regular contacts

Fig. 14 Example multi-contact planning and control for a car ingress scenario

4.3 Summary and Prospects on Contact Planning

Given the initial and goal configurations that differ in contact states, we saw a complete motion planning pipeline that goes all the way from planning the necessary changes in contact states, encoded as a sequence of stances and intermediate postures, to the full dynamically simulated motion traversing these changing contact states, among obstacles, and avoiding self-collision (Fig. 14). Yet, full autonomy is not yet achieved. Let us enumerate a number of points that still require manual user intervention:

1. Specifying the milestone configurations if the guide path planner is not used.
2. A full 3D model of the environment is required.
3. Listing candidate contact surfaces both on the robot and the environment. In particular, the user manually extracts candidate surfaces on the environment. The automation of this process is known as *surface segmentation*.
4. The user may additionally restrict matches between robot and environment surfaces (e.g., "do not use hand surfaces on the floor"). This reduces the branching factor of the exploration tree in the contact-switching sequence planning.
5. Tag some surfaces as being appropriate for grasping. This additional domain knowledge should be automatically inferred by adequate analysis of the shape of the environment component.
6. Decomposing the environment into convex components. As in item 3, this part can be automated by dedicated algorithms.

7. Specifying waypoints for the moving contact body in situations that are trapped in local minima for collision avoidance.
8. Retuning some parameters (e.g., COM weight) in hindsight if the simulation leads to falling down of the robot.

Like classical randomized planners, this approach also suffers from the resolution incompleteness issue: it cannot report in finite time the nonexistence of a solution. It is therefore more of a practical tool for motion design and control.

5 Multi-contact Predictive Control

In the previous section, boundary configurations in static equilibrium were connected by a QP-based whole-body controller with PID tasks under critical damping (i.e., the damping coefficient was set to twice the square root of the stiffness). This approach succeeds in connecting states that are reasonably close to each other but may fail at some more complex tasks where going straight to the target does not work. Walking gives a typical example of this, as gaited COM trajectories need to be carefully synchronized with footsteps and accelerations. Model predictive control provides tools to implement these more complex tasks.

Figure 15 illustrates the main components of a complete architecture completing the framework from the previous sections with a replanning loop. In this loop, a predictive control looks ahead for a few seconds on future trajectories (usually on a reduced model of the dynamics of the task of interest, as whole-body trajectories would be too expensive to survey) before forwarding its outputs to the tasks of the multitask whole-body controller.

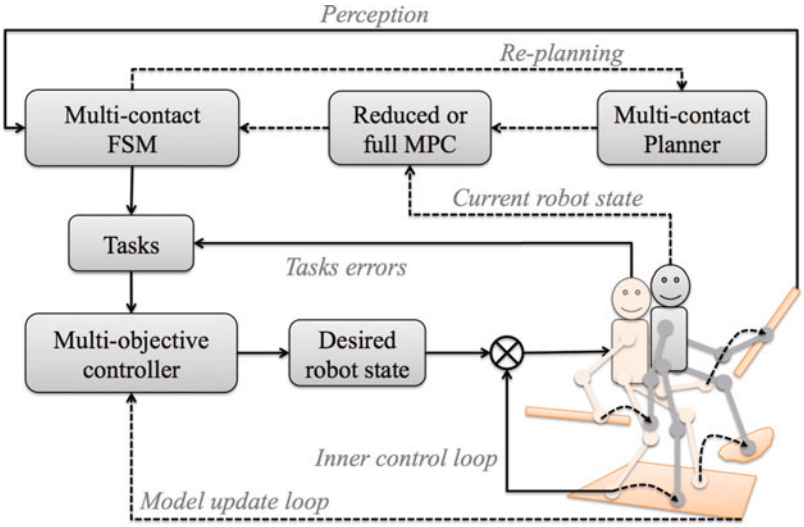

Fig. 15 Multi-contact planning and control integration scheme

5.1 Reduced-Model Predictive Control

The equations of motion of an n-DOF humanoid robot can be decomposed [96] in two parts: the Newton-Euler equations (15) corresponding to the six unactuated coordinates locating the robot in space and the n equations for each actuated DOF. In locomotion, a common working pipeline consists in treating the unactuated coordinates first, then delegate n remaining equations to the whole-body controller [3,21,26,48,54], or, if time permits, treat them in a second pass known as *dynamics filter* [51,99].

Let $\boldsymbol{x} = [\boldsymbol{p}_G \ \dot{\mathbf{p}}_G \ \mathbf{L}_G]$ denote the *state* of the center of mass. One way to perform predictive control on Newton-Euler equations is to take the vector $\boldsymbol{u} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f}_m]$ of stacked contact forces as control variable. Then, discretizing dynamics with sampling ΔT, Eq. (15) rewrites to

$$\boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}(\boldsymbol{x}(k))\boldsymbol{u}(k) + \boldsymbol{c} \tag{35}$$

denoting by $\mathbf{E}_3$ the 3×3 identity matrix and where

$$\mathbf{A} = \begin{bmatrix} \mathbf{E}_3 & \Delta T \mathbf{E}_3 \\ \mathbf{0}_{3\times 3} & \mathbf{E}_3 \\ \mathbf{0}_{3\times 3} & \mathbf{E}_3 \end{bmatrix} \quad \mathbf{B}(\boldsymbol{x}(k)) = \begin{bmatrix} \frac{1}{2}\Delta T^2 \mathbf{E}_3 \cdots & \frac{1}{2}\Delta T^2 \mathbf{E}_3 \\ \Delta T \mathbf{E}_3 \cdots & \Delta T \mathbf{E}_3 \\ \Delta T[\overrightarrow{GC_1}\times] \cdots & \Delta T[\overrightarrow{GC_m}\times] \end{bmatrix} \quad \boldsymbol{c} = \begin{bmatrix} m\boldsymbol{g} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{36}$$

At the beginning of a control step, the predictive controller receives the current state $\boldsymbol{x}_0 = (\boldsymbol{p}_0, \dot{\mathbf{p}}_0)$ estimated from sensor readings and computes a sequence of controls $\boldsymbol{u}(0), \ldots, \boldsymbol{u}(N)$ driving the system from $\boldsymbol{x}_0$ to $\boldsymbol{x}(N)$ at the end of the time horizon $T = N\Delta T$. The difficulty of this general formulation of Newton-Euler dynamics lies in the dependency of the matrix $\mathbf{B}$ upon the position $\boldsymbol{p}_G$ of the center of mass, highlighted by the notation $\mathbf{B}(\boldsymbol{x}(k))$.

Suppose that a planner provides a reference trajectory $\boldsymbol{x}_{\text{ref}}(t)$, including both COM $\boldsymbol{p}_G(t)$ and angular momentum $\mathbf{L}(t)$. Tracking this reference can be formulated as an optimization problem [48,54]:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \ \sum_{k=0}^{N} \ell_x(\boldsymbol{x}(k) - \boldsymbol{x}_{\text{ref}}(k)) + \ell_u(\boldsymbol{u}(k)) \tag{37}$$

$$\text{s.t.} \ \ \forall k, \ \boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}(\boldsymbol{x}(k))\boldsymbol{u}(k) + \boldsymbol{c} \tag{38}$$

$$\forall k, \ \boldsymbol{u}(k) \in \mathscr{C}_k \tag{39}$$

where ℓ_x and ℓ_u are generic quadratic functions and $\mathscr{C}_k$ denotes the linearized friction cone at the k^{th} contact. Different resolution strategies can be applied from there.

5.2 Nonlinear Multi-contact Predictive Control

Once a contact sequence has been decided, the optimal-control problem (OCP) (37), (38), and (39) can be attacked using a general OCP solver, a direction explored in [26, 54]. A damping behavior is implemented by setting $\boldsymbol{x}_{\text{ref}} = \mathbf{0}$, making ℓ_x ignore COM position and angular momentum so that solutions try to minimize their derivatives instead. Taking $\boldsymbol{u} = [\boldsymbol{w}_1 \cdots \boldsymbol{w}_m]$ the stacked vector of contact wrenches, control constraints can be written as linear inequalities using Proposition 3. With this approach, locomotion results from optimization under contact constraints and phase timings. That is, contact switches force the robot to move forward:

$$\min_{\boldsymbol{x},\boldsymbol{u}} \sum_{k=0}^{N} \ell_{\text{lim}}(\boldsymbol{p}_G(k)) + \ell_{\text{damp}}(\dot{\mathbf{p}}_G(k)) + \ell_u(\boldsymbol{u}(k)) \tag{40}$$

$$\text{s.t. } \forall k,\ \boldsymbol{x}(k+1) = \mathbf{A}\boldsymbol{x}(k) + \mathbf{B}(\boldsymbol{x}(k))\boldsymbol{u}(k) + \boldsymbol{c} \tag{41}$$

$$\forall k,\ \boldsymbol{u}(k) \in \mathscr{C}_k \tag{42}$$

$$\boldsymbol{x}(N) = \boldsymbol{x}_{\text{goal}} \tag{43}$$

Two questions arise in when solving such an OCP: forward integration and the choice of a shooting method. The former arises from the dependency of $\mathbf{B}(\boldsymbol{x}(k))$ upon the COM position: with a small number of preview steps, Eq. (41) overapproximates system dynamics, while a large number of preview steps will harm performance. To palliate this, a *forward integrator* (e.g., using the Runge-Kutta method) is used to compute $\boldsymbol{x}(k+1)$ from the pair $(\boldsymbol{x}(k), \boldsymbol{u}(k))$, using an integration step smaller than the overall discretization step of the predictive controller. Tuning the numbers of integration and discretization steps is part of controller design.

The second design choice lies in the shooting method. There are three main choices: single shooting, multiple shooting, and collocation. In multiple shooting, optimization variables are defined for both controls $\boldsymbol{u}(k)$ and states $\boldsymbol{x}(k)$, with pairs of consecutive states constrained by (41). As a result, the problem has more variables but is also sparser. This approach has been followed in [26, 54]. In [26], finding an initial solution to the problem takes several seconds; then further iterations are performed around 100 times faster. Figure 16 illustrates a stand-up motion generated using this approach.

The main motivation for applying nonlinear solvers to the general predictive control problem lies in the angular momentum $\dot{\mathbf{L}}_G$ at the COM, which yields a cross-product between the COM position $\boldsymbol{p}_G$ and contact forces $\boldsymbol{f}_i$. Non-convexity of the cross-product operator makes this problem uneligible for common quadratically constrained quadratic programming (QCQP) solvers, which usually work on *convex* inequality constraints. Ongoing works include a study of the structure of this cross-product to design better-suited solvers [78].

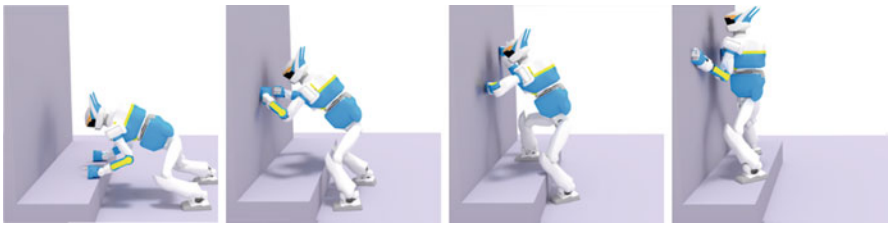

Fig. 16 HRP-2 standing up using wall contacts. Preview trajectory generated by applying an optimal-control solver to the predictive control problem (37) (Illustration adapted from [26])

5.3 Linearized Multi-contact Predictive Control

A general technique to linearize the COM cross-product in the equations of motion is to fix the COM trajectory along one coordinate, for instance, the vertical one [3, 50, 70]. Suppose that a feed-forward trajectory $z_G(t)$ is given to the predictive controller. By differentiating twice, one can equivalently rewrite it as a vertical force profile:

$$f_z(k) \;=\; \sum_{\text{contact } i} f_{iz}(k) \tag{44}$$

Considering all $f_z(k)$ as constants, Nagasaka et al. [70] noticed that the Euler equation (15) expands in the form (35) but where this time the matrix $\mathbf{B}$ does not depend on the COM position $\boldsymbol{p}_G(k)$. By recursively applying these new discretized dynamics, $\boldsymbol{x}(k)$ can be written as a function of the current state $\boldsymbol{x}_0$ and the stacked vector of preview controls $\mathbf{U}(k) \stackrel{\text{def}}{=} [\boldsymbol{u}(0) \;\cdots\; \boldsymbol{u}(k)]$:

$$\tilde{\boldsymbol{x}}(k) \;=\; \boldsymbol{\Phi}_k \tilde{\boldsymbol{x}}_0 + \boldsymbol{\Psi}_k \mathbf{U}(k-1) \tag{45}$$

where $\tilde{\boldsymbol{x}}$ is the projection of $\boldsymbol{x}$ eliminating components along the z-coordinate. To reduce the number of contact forces stacked in $\boldsymbol{u}$, [3, 70] considered a single force applied at the COP of each contact, using the COP contact-stability criterion from Proposition 4:

$$\boldsymbol{u} \;=\; [\boldsymbol{f}_1 \;\boldsymbol{p}_{\text{cop},1} \cdots \boldsymbol{f}_m \;\boldsymbol{p}_{\text{cop},m}] \tag{46}$$

$$\forall k,\; \boldsymbol{f}_k \in \mathscr{C}_{f,k} \tag{47}$$

$$\forall k,\; \boldsymbol{p}_{\text{cop},k} \in \mathscr{S}_i \tag{48}$$

with $\mathscr{C}_{f,k}$ and $\mathscr{S}_k$ the kth Coulomb friction cone and contact polygon, respectively. The resulting criterion is incomplete as only Eqs. (10)–(11) from Proposition 4 are taken into account (Eq. (12) is missing), meaning yaw slippage is not considered. Using the feed-forward trajectory $z_G(t)$ to compute the fixed sequence $[f_{z0} \ldots f_{zN}]$

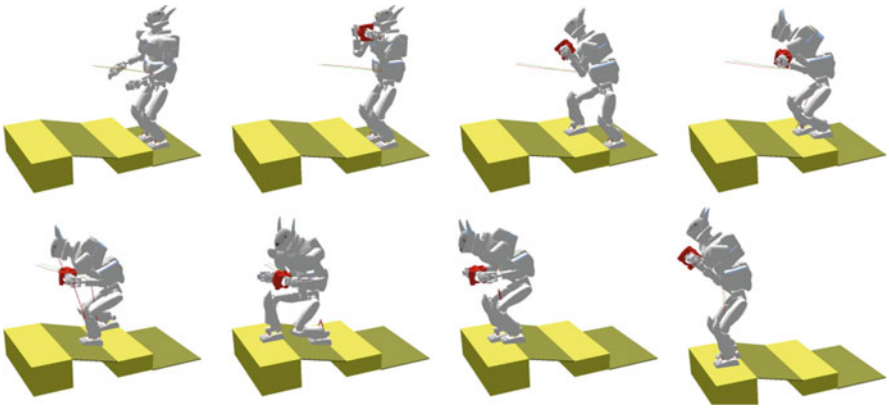

Fig. 17 HRP-2 walking on uneven ground while carrying an object. Preview trajectory generated using a QP reformulation of the predictive control problem (37) (Illustration adapted from [3])

of resultant vertical contact forces, the optimization cost (37) can be formulated quadratically on the stacked state vector $\mathbf{X} = [\boldsymbol{x}(0) \cdots \boldsymbol{x}(N)]$:

$$(\mathbf{X} - \mathbf{X}_{\text{ref}})^\top \mathbf{W}_x (\mathbf{X} - \mathbf{X}_{\text{ref}}) + \mathbf{U}^\top \mathbf{W}_u \mathbf{U} = \mathbf{U}^\top \mathbf{Q} \mathbf{U} + 2\boldsymbol{r}^\top \mathbf{U} + \boldsymbol{s}, \tag{49}$$

where the matrix $\mathbf{Q}$ and vectors $\boldsymbol{r}, \boldsymbol{s}$ are computed from (45). The resulting optimization problem is finally written:

$$\min_{\mathbf{U}} \quad \mathbf{U}^\top \mathbf{Q} \mathbf{U} + 2\boldsymbol{r}^\top \mathbf{U} + \boldsymbol{s} \tag{50}$$

$$\text{s.t.} \quad \mathbf{F}\mathbf{U} \leq \mathbf{0} \tag{51}$$

From the standpoint of numerical optimization, this reformulation in terms of $\mathbf{U}$ is a single shooting method. It relies on the feed-forward reference trajectory $z_G(t)$ provided by the planner and can be run fast enough for the control loop (around 20 ms to compute a cold-start CoM trajectory reported in [3]). Figure 17 illustrates a constrained walking motion generated using this approach.

5.4 Centroidal Predictive Control

The previous approaches have in common that a small-dimensional state $\boldsymbol{x}$ (positions and velocities at the COM) is controlled by a high-dimensional $\boldsymbol{u}$ (contact wrenches), a design choice that can be traced back to the seminal walking pattern generator from Hirukawa et al. [49]. From a polyhedral point of view, these methods are based on the V-representation of the CWC, which is straightforward to compute from individual friction cones. At the time of [49], polyhedral algorithms to compute the MCWC (Proposition 5) were not known yet to roboticists. Let

us see now how these tools can be applied to formulate predictive control using the H-representation of the CWC, i.e., eliminating the redundant force variables so that the final problem only contains COM positions and velocities (states) and COM accelerations (controls). In this case, the *centroidal dynamics* [72] of the system are decoupled from the set of contact forces, opening the way for the application of simplified dynamical models such as the spring-loaded inverted pendulum (SLIP) [67,95].

Let $\boldsymbol{u}$ denotes the acceleration $\ddot{\mathbf{p}}_G$ of the COM. Enforcing conservation of the angular momentum $\dot{\mathbf{L}}_G = \mathbf{0}$, the state $\boldsymbol{x}$ reduces to $[\boldsymbol{p}_G \ \dot{\mathbf{p}}_G]$ and matrices in the discretized equations of motion (35) become

$$\mathbf{A} = \begin{bmatrix} \mathbf{E}_3 & \Delta T \mathbf{E}_3 \\ \mathbf{0}_{3\times 3} & \mathbf{E}_3 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \frac{1}{2}\Delta T^2 \mathbf{E}_3 \\ \Delta T \mathbf{E}_3 \end{bmatrix} \quad \boldsymbol{c} = \mathbf{0} \tag{52}$$

The MCWC condition (Proposition 5) can then be written as a bilinear inequality constraint between states and controls:

$$\boldsymbol{x}(k)^\top \mathbb{A}_\times (\boldsymbol{u}(k) - \boldsymbol{g}) + \mathbf{A}'_O (\boldsymbol{u}(k) - \boldsymbol{g}) \ \leq \ \mathbf{0} \tag{53}$$

with $\mathbb{A}_\times$ is a $3 \times L \times 3$ tensor and $\mathbf{A}'_O$ the first three columns of $\mathbf{A}_O$. Applying the QP reduction (45), (46), (47), (48), (49), (50), and (51), the predictive control problem (37), (38), and (39) can finally be written as a quadratically constrained quadratic program (QCQP):

$$\min_{\mathbf{U}} \ \mathbf{U}^\top \mathbf{Q} \mathbf{U} + 2\boldsymbol{r}^\top \mathbf{U} + \boldsymbol{s} \tag{54}$$

$$\text{s.t.} \ \ \forall k, \ \mathbf{U}(k)^\top \mathbb{P}_k \mathbf{U}(k) + \mathbf{Q}_k \mathbf{U}(k) + \boldsymbol{l}_k \leq \mathbf{0} \tag{55}$$

where $\mathbb{P}_k$ is a $3(k+1) \times L \times 3(k+1)$ tensor, $\mathbf{Q}_k$ is a $L \times 3(k+1)$, and matrix $\boldsymbol{l}_k$ is an L-dimensional vector. From there, one can apply a QCQP solver [71,92] to find optimal solutions. Real-time constraints imply however that only a small number of solver iterations can be run in the control loop. Van Heerden [92] also observed that, even in a simplified setting with coplanar surfaces, inequality constraints (53) are not positive semidefinite, which points out that the QCQP problem is not convex.

5.5 Linearized Centroidal Predictive Control

A generic method to transform QCQP with bilinear inequalities into QP with linear ones is to bound one of the two bilinear terms (either state or control). In [16], the nonlinear pendulum frequency term is bounded to linearize constraints on vertical COM motions. Similarly, polytopes of robust static-equilibrium COM positions $\boldsymbol{p}_G$ were obtained in [23,32] by defining polyhedral bounds on force disturbances $\boldsymbol{f}_\epsilon$, thus eliminating the bilinear coupling between $\boldsymbol{p}_G$ and $\boldsymbol{f}_\epsilon$. The same method can be applied to Eq. (53) by bounding the COM position in a polytope [21,31]. In

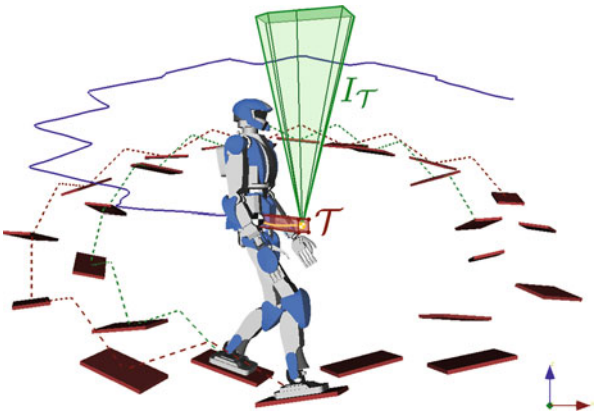

Fig. 18 Example where a model of the HRP-4 humanoid walks a circular staircase with tilted steps using predictive control on its COM. Future COM positions are restricted to a polytope $\mathcal{T}$ (*red*), yielding a dual acceleration cone $I_{\mathcal{T}}$ (*green*) that can be computed fast enough for the control loop. Scale and position of $I_{\mathcal{T}}$ were chosen arbitrarily in this figure

the derivation of the latter, boxing the COM in a polytope $\mathbf{P}\boldsymbol{x}(k) \leq \mathbf{1}$ yields a linearized dual constraint $\mathbf{C}\boldsymbol{u}(k) \leq \mathbf{0}$ on its acceleration, so that the optimization problem becomes

$$\min_{\mathbf{U}} \quad \mathbf{U}^\top \mathbf{Q}\mathbf{U} + 2\boldsymbol{r}^\top \mathbf{U} + \boldsymbol{s} \tag{56}$$

$$\text{s.t.} : \forall k,\ \mathbf{C}\boldsymbol{u}(k) \leq \mathbf{0} \tag{57}$$

$$\forall k,\ \mathbf{P}\boldsymbol{x}(k) \leq \mathbf{1} \tag{58}$$

This linearization has a cost: problem (56), (57), and (58) is a strict subproblem of (54)–(55), with the same cost function but more restrictive constraints. Its optimal solutions are thus suboptimal to the initial problem, or it may have no solution, while the initial one was feasible. Applications suggest however that these scenari are not as frequent as may be feared [16, 21], while the latter reference reported sub-millisecond computation times to solve the QP (56), (57), and (58). Figure 18 shows locomotion in multi-contact generated using this approach.

6 Applications

We have been recently witnessing a sustained evolution of humanoid technology in terms of hardware and embedded software. The demonstrations made by Honda's Asimo or Boston Dynamics' humanoids, for example, were particularly impressive. These efforts are presumably sustained, and we may expect more advanced hardware and skills in future humanoid robots. Hence, it is time to concretely consider what applications are possible for the anthropomorphic robots and what are their requirements in terms of functionalities and performances.

Our particular concern in this chapter is multi-contact planning and control. There are three emerging fields of applications where multi-contact technology

is a must-implement behavior in order to exploit humanoids to their fullest performances. These are, listing them by reverse chronological appearance:

1. Large-scale manufacturing in confined or cluttered spaces as revealed by airplane assembly lines requirements (http://www.comanoid.eu).
2. Rescue and disaster operations as recently revealed by the DARPA robotic challenge (http://www.theroboticschallenge.org/);
3. Domotic assistance for frail persons as targeted by SoftBank Robotics, e.g., [15, 58, 64].

In the following we detail as illustrative examples the large-scale manufacturing and rescue operations applications and open problems.

6.1 Large-Scale Manufacturing

We exemplify the applications in manufacturing with our two ongoing research projects with AIRBUS. At the AIRBUS airliners assembly lines, most of the operations dealing with assembly or system installation are not automated and are currently achieved manually. This is even more true for those operations that have to be performed inside the aircraft. Robotic solutions have not found their way into all areas of aeronautical assembly lines because of safety, accessibility, weight, and the complexity of the operations to be achieved.

Figure 19 shows a set of shop floors where any automation/robotic solution shall operate. The cargo area, the cockpit, and the aile are particularly not flat, and these are precisely where legged robots may be able to provide a more complete manufacturing solution. Furthermore, if one considers wheeled cobotic solutions, they will be confined to a given floor because they are not capable of autonomously moving to another platform within the same aircraft. While the installation of elevators could be a solution, it is costly and cannot be deployed everywhere. On the other hand, a humanoid platform would be free of this constraint since it can use the stairs already in place for human workers.

If we want to increase the level of automation, in order to increase the shop floor productivity, we have to devise robots capable of:

- Working in such confined, cluttered, and rugged areas;
- Performing varying operations with standard manual tools (like humans);
- Moving autonomously through the manufacturing environment.

The motivation behind using humanoid technology is not solely driven by a wish to increase the level of automation. In fact, we mainly aim at addressing human worker's health and safety issues. The goal is to use humanoids to perform "non-added value tasks": repetitive tasks where the experience and intelligence of the operator are not put to use and where boredom can lead to mistakes, tasks presenting health risks such as anticorrosion painting (highly toxic) or cleaning

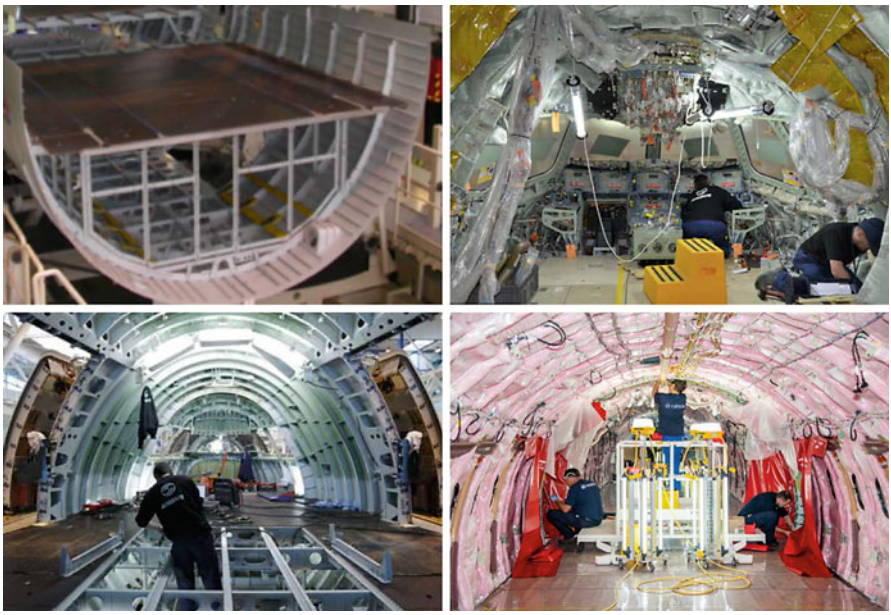

Fig. 19 Example of shop floors and environments (from *left* to *right*, *top* to *down*): the cargo area, the cockpit, the upper-area in two different phases of the assembly/installation. Notice the postures of the workers. A humanoid is expected to have such accessibility capabilities

(after use of solvents, metallic dirt after drilling, etc.) that are usually performed in confined spaces.

If safety requirements are met, a humanoid could perform a number of operations, such as:

- Accurate assembly operations, e.g., riveting, drilling, and screwing using manual or semiautomatic hand tools;
- Cleaning and painting operations;
- System installation: electrical harness installation, connector plugging, etc.

Humanoids could also be used to perform other non-added value tasks such as conveying equipment or tools to highly qualified operators: hundreds of thousands of parts and tools are transported and manipulated by the operators for each aircraft. By unburdening highly qualified operators from such boring tasks, a robot (even a costly humanoid) is socially and economically viable.

It is important to highlight that the introduction of humanoids in aeronautic plants should be facilitated because the aeronautic shop floor is a very well-mastered environment:

- 3D models of the aircraft and the shop floors exist, enabling model-based reference and localization of the robot,

- The Airbus operators are trained: they follow strict rules and assembly sequences, which will simplify human-humanoid interaction.

In that example of the Airbus project that is focused on using multi-contact technology, the technological bottleneck concerns (i) navigation that is yet to be demonstrated with real humanoid in a 1:1 scale airplane mock-up, and (ii) once near the area of assembly or system installation, achieve the tasks and eventually report any encountered problem. Multi-contact planning and control is now being used for specific tasks and mockups provided by Airbus; see a preliminary result in [6] where task-aware multi-contact planning and multimodal multi-robot multi-contact QP controller is experimented in a circuit-breaker check use case and results in [76] for nut-onto-bolt fastening operations.

6.2 Rescue and Disaster Operations

Rescue and disaster operations were best illustrated recently by the DARPA robotic challenge. Several teams worldwide gathered with various robotic platforms to achieve eight challenging tasks: (i) drive a utility vehicle, (ii) egress from the vehicle, (iii) open and traverse a door, (iv) rotate a valve, (v) use a tool to break through a wall, (vi) unplug and re-plug a connector (vii) traverse rubble, and (viii) climb stairs. The contest brought into light the gap that remains to overcome in order to efficiently deploy the humanoid technology outdoor and achieve complex tasks such as driving [74] or ladder climbing [91].

There were a number of situations where the DARPA challenge robots could have exploited multi-contact motions in order to enhance, make more robust, or simply sustain their equilibrium and the performance of the task. If we take all the challenging tasks in their chronological order, we noticed that:

1. For the driving (first) task, none of the robots ingressed autonomously. Egress from the car is nearly impossible without considering additional contact supports; the environment is so cluttered that avoiding multi-contact planning and control would require to position the robot in a ready-to-egress posture, which all teams did to secure the task. The successful teams have exploited non-controlled (i.e., open-loop) sliding, others bet on a robust attitude and posture stabilization, etc. While the commonsense would have suggested to grasp the car door rail and egress in multi-contact safe whole-body motion, none of the robots did it. This is because controlling and planning whole-body sliding in multi-contact are not readily solved problems and are still an open research challenge. In addition, some robots had too weak-powered grippers to risk such a strategy.
2. For the door opening and entering the building, the valve, the hose, and all the surprise tasks operations, taking an additional (task-aware) contact as suggested recently in [6], would have been beneficial. None of the teams did it. There were several cited reasons for this depending on each team strategy: not readily available multi-contact “spirit”; using both hands in operations: e.g., some teams

embedded a camera-in-gripper system to monitor closely the tasks and could not use the hand to lean for an additional contact; not readily available force control; risk conservatism; etc.

3. Apparently none of the teams who had chosen to challenge the bricks uneven terrain task (instead of debris removal) had "noticed" the presence of a wall that they can exploit for a better stable walking strategy! The main reason stands probably in the lack of readily available technology to walk dynamically in multi-contact. Only recently did we witness new development in multi-contact predictive control for uneven walking eventually with hand contact; see examples in [3, 16, 21, 22, 31, 54, 71]. None of the developments reached robust and repeatable experiments with real humanoid robots.
4. Finally, more unexpected was the absence of use of multi-contact motion for the stair-climbing task. Whereas the common sense would have suggested to exploit the handrail and climb the stairs stably, none of the robots, neither the humanoids nor the non-humanoid ones, did it. This is probably the most disappointing observation in the challenge. Here again, the cited reason is that multi-contact technology dealing with such tasks was not readily available, and the teams preferred to avoid using the hand contact since not only is it not easy to plan, but it is also difficult to control. This was highlighted in papers that tried multi-contact motions in climbing ladders [91] or stairs using handrails [2, 26].

To sum up, although attempts outside the frame of challenge have been made or are ongoing, the question of why none of the teams used multi-contact technology during the challenge itself was raised and is open for debate. We believe that if such a challenge is to be renewed in the future, mastering multi-contact technology will be the key for robustness and efficiency of the operations. Nevertheless, using multi-contact is subject to the ability to detect contact, close the loop on contact motions, and control contact forces. This requires additional functionalities such as having whole-body contact detection through dedicated sensing or using a mixed of integrated sensing and observers; see recent work in [40, 65, 66, 93].

7 Conclusion

Multi-contact planning and control is a key technology for the development and effective deployment of humanoid robots in real application scenarios: disaster interventions, high-scale manufacturing, companion assistive robot, etc. Non-gaited motions not only increase humanoid capabilities beyond walking and their manipulation skills beyond bimanual schemes, it also allows them to increase their stability through contact-aware behaviors. We have surveyed the overall framework used in ongoing efforts to realize such awareness. First, contact events are decided in a discrete fashion by a contact planner. A predictive controller determines then how to make use of them in a preview of the robot's future so as to achieve a desired goal. Finally, these rough predictions are turned into precise motor controls by a whole-body controller that takes into account the full kinematic and dynamic model of the

system. Examples of applications where these components are integrated exhibit the feasibility of this pipeline while highlighting the road ahead.

This being said, state-of-the-art multi-contact planning and control requires additional research efforts as the problem is far from being solved. Although impressive demonstrators are achieved, these are mostly made in well-calibrated environments; for specific tasks, they hardly repeat, and they are not robust to few changes and perturbations in most cases. The planning does not generate what one would expect without guidance (even in simulation), the generated contact stances plans often do not result in a feasible control (even without perturbation and with very reliable models), and replanning in case of failure is not yet tackled. Therefore, new methodologies that bring a flavor of semantics and learning use more sophisticated contact behaviors (sliding, deforming, etc.) that are robust and could tighten the planning, and the control parts are to be investigated.

References

1. T. Arakawa, T. Fukuda, Natural motion generation of biped locomotion robot using hierarchical trajectory generation method consisting of GA, EP layers, in *IEEE International Conference on Robotics and Automation*, vol. 1, 1997, pp. 211–216
2. H. Audren, A. Kheddar, P. Gergondet, Stability polygons reshaping and morphing for smooth multi-contact transitions and force control of humanoid robots, in *IEEE-RAS International Conference on Humanoid Robots*, Cancun, 2016, pp. 1037–1044
3. H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, E. Yoshida, Model preview control in multi-contact motion – application to a humanoid robot, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 2014, pp. 4030–4035
4. D.J. Balkcom, J.C. Trinkle, Computing wrench cones for planar rigid body contact tasks. Int. J. Robot. Res. **21**(12), 1053–1066 (2002)
5. M. Benallegue, A. Escande, S. Miossec, A. Kheddar, Fast C^1 proximity queries using support mapping of sphere-torus-patches bounding volumes, in *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 483–488
6. A. Bolotnikova, K. Chappellet, A. Paolillo, A. Escande, G. Anbarjafari, A. Suarez-Roos, P. Rabaté, A. Kheddar, A circuit-breaker use-case operated by a humanoid in aircraft manufacturing, in *IEEE Conference on Automation Science and Engineering*, Xi'an, 2017
7. K. Bouyarmane, A. Escande, F. Lamiraux, A. Kheddar, Potential field guide for humanoid multicontacts acyclic motion planning, in *2009 IEEE International Conference on Robotics and Automation*, Kobe, 2009, pp. 1165–1170
8. K. Bouyarmane, A. Kheddar, Static multi-contact inverse problem for multiple Humanoid robots and manipulated objects, in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, Nashville, 2010, pp. 8–13
9. K. Bouyarmane, A. Kheddar, Fem-based static posture planning for a humanoid robot on deformable contact support, in *2011 11th IEEE-RAS International Conference on Humanoid Robots*, Bled, 2011, pp. 487–492
10. K. Bouyarmane, A. Kheddar, Multi-contact stances planning for multiple agents, in *IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 5246–5253
11. K. Bouyarmane, A. Kheddar, Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, 2011, pp. 4414–4419
12. K. Bouyarmane, A. Kheddar, Humanoid robot locomotion and manipulation step planning. Adv. Robot. **26**(10), 1099–1126 (2012)

13. K. Bouyarmane, A. Kheddar, On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control, in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2012), pp. 36–42
14. K. Bouyarmane, A. Kheddar, Non-decoupled locomotion and manipulation planning for low-dimensional systems. Int. J. Intell. Robot. Syst. (to appear). https://doi.org/10.1007/s10846-017-0692-5
15. K. Bouyarmane, J. Vaillant, N. Sugimoto, F. Keith, J.I. Furukawa, J. Morimoto, Brain-machine interfacing control of whole-body humanoid motion. Front. Syst. Neurosci. **8**, 138 (2014)
16. C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, P.B. Wieber, A robust linear MPC approach to online generation of 3d biped walking motion, in *IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2015), pp. 595–601
17. T. Bretl, Motion planning of multi-limbed robots subject to equilibrium constraints: the free-climbing robot problem. Int. J. Robot. Res. **25**(4), 317–342 (2006)
18. T. Bretl, S. Lall, Testing static equilibrium for legged robots. IEEE Trans. Robot. **24**(4), 794–807 (2008)
19. S. Brossette, A. Escande, G. Duchemin, B. Chretien, A. Kheddar, Humanoid posture generation on non-Euclidean manifolds, in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 352–358. DOI 10.1109/HUMANOIDS.2015.7363574
20. S. Caron, Computational foundation for planner-in-the-loop multi-contact whole-body control of humanoid robots. Ph.D. thesis, The University of Tokyo, 2016. See p. 81 for a proof of Proposition 4
21. S. Caron, A. Kheddar, Multi-contact walking pattern generation based on model preview control of 3D COM accelerations, in *IEEE-RAS International Conference on Humanoid Robots*, Cancun, 2016, pp. 550–557
22. S. Caron, A. Kheddar, Dynamic walking over rough terrains by nonlinear predictive control of the floating-base inverted pendulum, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, 2017
23. S. Caron, Q.C. Pham, Y. Nakamura, Leveraging cone double description for multi-contact stability of humanoids with applications to statics and dynamics, in *Robotics: Science and System*, 2015
24. S. Caron, Q.C. Pham, Y. Nakamura, Stability of surface contacts for humanoid robots: closed-form formulae of the contact wrench for rectangular support areas, in *IEEE International Conference on Robotics and Automation*, 2015, pp. 5107–5112
25. S. Caron, Q.C. Pham, Y. Nakamura, ZMP support areas for multi-contact mobility under frictional constraints. IEEE Trans. Robot. **33**, 67–80 (2017). DOI 10.1109/TRO.2016.2623338
26. J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, N. Mansard, A versatile and efficient pattern generator for generalized legged locomotion, in *IEEE International Conference on Robotics and Automation*, Stockholm, 2016
27. J. Chestnutt, J. Kuffner, K. Nishiwaki, S. Kagami, Planning biped navigation strategies in complex environments, in *IEEE-RAS International Conference on Humanoid Robots*, 2003
28. B. Chrétien, A. Escande, A. Kheddar, Continuously satisfying constraints with contact forces in trajectory optimization for humanoid robots, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 3956–3961. DOI 10.1109/IROS.2015.7353934
29. B. Chrétien, A. Escande, A. Kheddar, GPU robot motion planning using semi-infinite nonlinear programming. IEEE Trans. Parallel Distrib. Syst. **27**(10), 2926–2939 (2016)
30. R. Cisneros, K. Yokoi, E. Yoshida, Yaw moment compensation by using full body motion, in *2014 IEEE International Conference on Mechatronics and Automation (ICMA)* (IEEE, 2014), pp. 119–125
31. H. Dai, R. Tedrake, Planning robust walking motion on uneven terrain via convex optimization, in *IEEE-RAS International Conference on Humanoid Robots*, Cancun, 2016, pp. 579–586

32. A. Del Prete, S. Tonneau, N. Mansard, Fast algorithms to test robust static equilibrium for legged robots, in *IEEE International Conference on Robotics and Automation*, Stockholm, 2016
33. A. Escande, A. Kheddar, Contact planning for acyclic motion with tasks constraints, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009
34. A. Escande, A. Kheddar, S. Miossec, Planning support contact-points for humanoid robots and experiments on HRP-2, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006
35. A. Escande, A. Kheddar, S. Miossec, Planning contact points for humanoid robots. Robot. Auton. Syst. **61**(5), 428–442 (2013)
36. A. Escande, A. Kheddar, S. Miossec, S. Garsault, Planning support contact-points for acyclic motions and experiments on HRP-2, in *International Symposium on Experimental Robotics*, 2008
37. C. Esteves, G. Arechavelata, J. Pettré, J.P. Laumond, Animation planning for virtual characters cooperation. ACM Trans. Graph. **25**(2), 319–339 (2006)
38. B. Faverjon, P. Tournassoud, Planning of manipulators with a high number of degrees of freedom, in *IEEE International Conference on Robotics and Automation*, 1987
39. R. Featherstone, *Rigid Body Dynamics Algorithms* (Springer, US, 2014)
40. F. Flacco, A. Paolillo, A. Kheddar, Residual-based contacts estimation for humanoid robots, in *IEEE-RAS International Conference on Humanoid Robots*, Cancun, 2016, pp. 409–415
41. K. Fukuda, A. Prodon, Double description method revisited, in *Combinatorics and Computer Science* ed. by M. Deza, R. Euler, I. Manoussakis. Lecture Notes in Computer Science, vol 1120 (Springer, Berlin/Heidelberg, 1996), pp. 91–111
42. E.G. Gilbert, D.W. Johnson, S.S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space. IEEE Trans. Robot. Autom. **4**, 193–203 (1988)
43. K. Harada, S. Kajita, F. Kanehiro, K. Fujiwara, K. Kaneko, K. Yokoi, H. Hirukawa, Real-time planning of humanoid robot's gait for force-controlled manipulation. IEEE/ASME Trans. Mechatron. **12**(1), 53–62 (2007)
44. K. Hauser, Fast interpolation and time-optimization with contact. Int. J. Robot. Res. **33**(9), 1231–1250 (2014)
45. K. Hauser, T. Bretl, J.C. Latombe, Non-gaited humanoid locomotion planning, in *IEEE-RAS International Conference on Humanoid Robots*, 2005
46. K. Hauser, T. Bretl, J.C. Latombe, K. Harada, B. Wilcox, Motion planning for legged robots on varied terrain. Int. J. Robot. Res. **27**(11–12), 1325–1349 (2008)
47. A. Herdt, H. Diedam, P.B. Wieber, D. Dimitrov, K. Mombaur, M. Diehl, Online walking motion generation with automatic footstep placement. Adv. Robot. **24**(5–6), 719–737 (2010)
48. A. Herzog, N. Rotella, S. Schaal, L. Righetti, Trajectory generation for multi-contact momentum control, in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 874–880
49. 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
50. M.A. Hopkins, D.W. Hong, A. Leonessa, Humanoid locomotion on uneven terrain using the time-varying divergent component of motion, in *2014 14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2014), pp. 266–272
51. 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03)*, vol. 2 (IEEE, 2003), pp. 1620–1626
52. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, H. Hirukawa, The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation, in *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1 (IEEE, 2001), pp. 239–246

53. L.E. Kavraki, P. Svestka, J. Claude Latombe, M.H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**, 566–580 (1996)
54. 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 *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2015), pp. 684–689
55. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, H. Inoue, Dynamically-stable motion planning for humanoid robots. Auton. Robot. **12**, 105–118 (2002)
56. J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, H. Inoue, Footstep planning among obstacles for biped robots, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001
57. S. Kuindersma, F. Permenter, R. Tedrake, An efficiently solvable quadratic program for stabilizing dynamic locomotion, in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2014), pp. 2589–2594
58. A. Kumar Pandey, R. Gelin, R. Alami, R. Viry, A. Buendia, R. Meertens, M. Chetouani, L. Devilliers, M. Tahon, D. Filliat, Y. Grenier, M. Maazaoui, A. Kheddar, F. Lerasle, L.F. Duval, Romeo2 project: humanoid robot assistant and companion for everyday life: I. Situation assessment for social intelligence, in *International Workshop on Artificial Intelligence and Cognition*, Torino, 2014, pp. 140–147
59. J.C. Latombe, *Robot Motion Planning* (Kluwer Academic Publishers, Boston, 1991)
60. S. LaValle, J. Kuffner, Rapidly-exploring random trees: progress and prospects, in *Algorithmic and Computational Robotics: New Direction*, ed. by B. Donald, K. Lynch, D. Rus (A. K. Peters, Wellesley, 2001), pp. 293–308
61. S.M. LaValle, *Planning Algorithms* (Cambridge University Press, New York, 2006)
62. O. Lefebvre, F. Lamiraux, D. Bonnafous, Fast computation of robot-obstacle interactions in nonholonomic trajectory deformation, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005
63. S. Lengagne, J. Vaillant, E. Yoshida, A. Kheddar, Generation of whole-body optimal dynamic multi-contact motions. Int. J. Robot. Res. **32**(9–10), 1104–1119 (2013)
64. A.M. López, J. Vaillant, F. Keith, P. Fraisse, A. Kheddar, Compliant control of a humanoid robot helping a person stand up from a seated position, in *14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2014), pp. 817–822
65. L. Manuelli, R. Tedrake, Localizing external contact using proprioceptive sensors: the contact particle filter, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Daejon, 2016, pp. 5062–5069
66. T. Mattioli, M. Vendittelli, Interaction force reconstruction for humanoid robots. IEEE Robot. Autom. Lett. **2**(1), 282–289 (2017)
67. I. Mordatch, M. De Lasa, A. Hertzmann, Robust physics-based locomotion using low-dimensional planning. ACM Trans. Graph. (SIGGRAPH)) **29**(4), 71 (2010)
68. I. Mordatch, E. Todorov, Z. Popović, Discovery of complex behaviors through contact-invariant optimization. ACM Trans. Graph. (TOG) **31**(4), 43 (2012)
69. T. Moulard, F. Lamiraux, K. Bouyarmane, E. Yoshida, Roboptim: an optimization framework for robotics, in *Robomec*, 2013, p. 4p
70. K. Nagasaka, T. Fukushima, H. Shimomura, Whole-body control of a humanoid robot based on generalized inverse dynamics and multi-contact stabilizer that can take acount of contact constraints, in *Robotics Symposium (In Japanese)*, vol. 17, 2012
71. M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, P. Souères, A reactive walking pattern generator based on nonlinear model predictive control. IEEE Robot. Autom. Lett. **2**(1), 10–17 (2017)
72. D.E. Orin, A. Goswami, S.H. Lee, Centroidal dynamics of a humanoid robot. Auton. Robot. **35**(2–3), 161–176 (2013)
73. J.S. Pang, J. Trinkle, Stability characterizations of rigid body contact problems with coulomb friction. ZAMM J. Appl. Math. Mech./Z. Angew. Math. Mech. **80**(10), 643–663 (2000)

74. A. Paolillo, P. Gergondet, A. Cherubini, M. Vendittelli, A. Kheddar, Autonomous car driving by a humanoid robot. J. Field Robot. (to appear). https://doi.org/10.1002/rob.21731
75. J. Pettré, J.P. Laumond, T. Siméon, A 2-stages locomotion planner for digital actors, in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003
76. K. Pfeiffer, A. Escande, A. Kheddar, Nut fastening with a humanoid robot, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, 2017
77. Q.C. Pham, O. Stasse, Time-optimal path parameterization for redundantly actuated robots: a numerical integration approach. IEEE/ASME Trans. Mechatron. **20**(6), 3257–3263 (2015)
78. B. Ponton, A. Herzog, S. Schaal, L. Righetti, A convex model of humanoid momentum dynamics for multi-contact motion generation, in *IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2016), pp. 842–849
79. Z. Qiu, A. Escande, A. Micaelli, T. Robert, Human motions analysis and simulation based on a general criterion of stability, in *International Symposium on Digital Human Modeling*, 2011
80. L. Righetti, M. Mistry, J. Buchli, S. Schaal, Inverse dynamics control of floating-base robots with external contraints: an unified view, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011
81. L. Saab, O. Ramos, N. Mansard, P. Soueres, J.Y. Fourquet, Generic dynamic motion generation with multiple unilateral constraints, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011
82. L. Saab, O.E. Ramos, F. Keith, N. Mansard, P. Souères, J.Y. Fourquet, Dynamic whole-body motion generation under rigid contacts and other unilateral constraints. IEEE Trans. Robot. **29**(2), 346–362 (2013)
83. T. Saida, Y. Yokokohji, T. Yoshikawa, Fsw (feasible solution of wrench) for multi-legged robots, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03)*, vol. 3 (IEEE, 2003), pp. 3815–3820
84. J. Salini, V. Padois, P. Bidaud, Synthesis of complex humanoid whole-body behavior: a focus on sequencing and tasks transitions, in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011
85. P. Sardain, G. Bessonnet, Forces acting on a biped robot. Center of pressure-zero moment point. IEEE Trans. Syst. Man Cybern. Part A Syst. Hum. **34**(5), 630–637 (2004)
86. L. Sentis, J. Park, O. Khatib, Compliant control of multi-contact and center of mass behaviors in humanoid robots. IEEE Trans. Robot. **26**(3), 483–501 (2010)
87. T. Siméon, J.P. Laumond, J. Cortès, A. Sahbani, Manipulation planning with probabilistic roadmaps. Int. J. Robot. Res. **23**(7–8), 729–746 (2004)
88. J.C. Trinkle, J.S. Pang, S. Sudarsky, G. Lo, On dynamic multi-rigid-body contact problems with coulomb friction. ZAMM J. Appl. Math. Mech./Z. Angew. Math. Mech. **77**(4), 267–279 (1997)
89. B. Ugurlu, J.A. Saglia, N.G. Tsagarakis, D.G. Caldwell, Yaw moment compensation for bipedal robots via intrinsic angular momentum constraint. Int. J. Humanoid Robot. **9**(04) (2012)
90. J. Vaillant, K. Bouyarmane, A. Kheddar, Multi-character physical and behavioral interactions controller. IEEE Trans. Vis. Comput. Graph. **23**(6), 1650–1662 (2017)
91. 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. Auton. Robot. **40**(3), 561–580 (2016)
92. K. Van Heerden, Real-time variable center of mass height trajectory planning for humanoids robots. IEEE Robot. Autom. Lett. **2**(1), 135–142 (2017)
93. J. Vorndamme, M. Schappler, S. Haddadin, Collision detection, isolation and identification for humanoids, in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4754–4761
94. A. Wachter, L.T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)

95. P.M. Wensing, D.E. Orin, High-speed humanoid running through control with a 3D-slip model, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 5134–5140
96. P.B. Wieber, Holonomy and nonholonomy in the dynamics of articulated motion, in *Fast Motions in Biomechanics and Robotics*, 2006, pp. 411–425
97. P.B. Wieber, Trajectory free linear model predictive control for stable walking in the presence of strong perturbations, in *6th IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2006), pp. 137–142
98. K. Yamane, J. Kuffner, J.K. Hodgins, Synthesizing animations of human manipulation tasks. ACM Trans. Grap. (Proc. SIGGRAPH 2004) **23**(3), 530–537 (2004)
99. K. Yamane, Y. Nakamura, Dynamics filter-concept and implementation of online motion generator for human figures. IEEE Trans. Robot. Autom. **19**(3), 421–432 (2003)
100. E. Yoshida, C. Esteves, I. Belousov, J.P. Laumond, K. Yokoi, Planning 3D collision-free dynamic robotic motion through iterative reshaping. IEEE Trans. Robot. **24**(5), 1186–1198 (2008)