Approximate hybrid model predictive control for multi-contact push recovery in complex environments

Approximate Hybrid Model Predictive Control for Multi-Contact Push Recovery in Complex Environments

Tobia Marcucci[1,2,3], Robin Deits[3], Marco Gabiccini[1,2,4], Antonio Bicchi[1,2], Russ Tedrake[3]

***Abstract*— Feedback control of robotic systems interacting with the environment through contacts is a central topic in legged robotics. One of the main challenges posed by this problem is the choice of a model sufficiently complex to capture the discontinuous nature of the dynamics but simple enough to allow online computations. Linear models have proved to be the most effective and reliable choice for smooth systems; we believe that piecewise affine (PWA) models represent their natural extension when contact phenomena occur. Discrete-time PWA systems have been deeply analyzed in the field of hybrid Model Predictive Control (MPC), but the straightforward application of MPC techniques to complex systems, such as a humanoid robot, leads to mixed-integer optimization problems which are not solvable at real-time rates. Explicit MPC methods can construct the entire control policy offline, but the resulting policy becomes too complex to compute for systems at the scale of a humanoid robot. In this paper we propose a novel algorithm which splits the computational burden between an offline sampling phase and a limited number of online convex optimizations, enabling the application of hybrid predictive controllers to higher-dimensional systems. In doing so we are willing to partially sacrifice feedback optimality, but we set stability of the system as an inviolable requirement. Simulation results of a simple planar humanoid that balances by making contact with its environment are presented to validate the proposed controller.**

I. Introduction

For robots governed by smooth nonlinear differential equations, local linearization and linear optimal control provide powerful tools for local stabilization of fixed points or trajectories [1]. Convenient parameterizations (e.g. LQR) allow designers to generate stabilizing controllers with tunable performance which can be applied to unconstrained systems or (via online convex optimization) to systems with linear constraints [2]. Through a natural description of the dynamics in terms of momentum, rather than generalized coordinates (i.e. joint angles), compact linear models are able to capture the fundamental features of the robot's physics within a remarkably large portion of state space [3]. Such models have been used with great success for humanoid robots with pre-planned contact locations, as in the Linear Inverted Pendulum Model (LIPM) [4], or pre-planned contact timings [5], [6].

The state of the art in multi-contact feedback control, on the other hand, is less mature: stabilization here is achieved mostly through one-step lookahead, e.g. using inverse dynamics approaches that involve solving a quadratic program (QP) online, and/or hand-designed state machines [1], [7]. Hierarchical tools can plan and stabilize complex multi-contact behaviors [1], [8] by choosing a contact sequence at low frequency and stabilizing that sequence at a higher control rate. In our case, however, we are specifically interested in creating fast online controllers which can choose to make or break contact instantaneously, without any pre-planned sequence of contacts or hybrid modes. To date, the major obstacle to achieving this is a systematic procedure for control design that can reason about stabilization across contact modes.

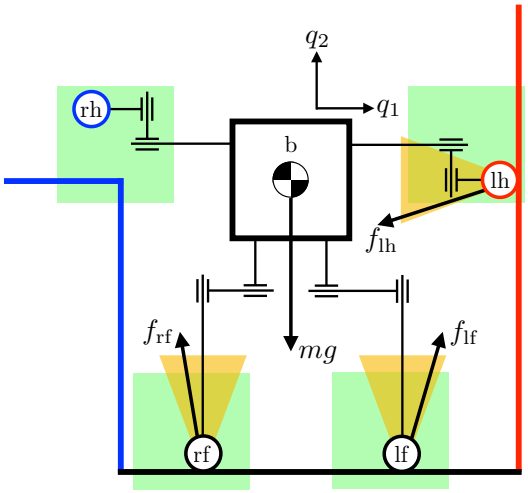

Fig. 1. Two-dimensional humanoid. Hands can interact with the surfaces that have the same color, green box-shaped sets represent the kinematic limits for the limbs, and friction cones for the contact forces are depicted in orange. The kinematics are roughly modeled after those of the NASA Valkyrie robot.

In this paper, we adopt PWA approximations of the system dynamics in floating-base coordinates as the natural and reasonable analog to linearization when local stabilization of fixed-points or trajectories requires crossing contact-mode boundaries. Piecewise affinity occurs naturally in the analysis of the centroidal dynamics of a robot subject to contact phenomena. A naïve but effective elastic contact model relates normal force to normal penetration (and, possibly, normal velocity) as well as friction force to tangential velocity directly in a PWA way. Complementarity-based contact models [9] also result in PWA dynamics, though as the result of the solution of a feasibility problem rather than a closed-form function. Moreover, it is well known that the time

[1] Research Center "E. Piaggio", University of Pisa, Pisa, Italy
[2] Department of Advanced Robotics, Istituto Italiano di Tecnologia, Genoa, Italy
[3] Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, USA
[4] Dipartimento di Ingegneria Civile e Industriale, University of Pisa, Pisa, Italy

evolution of the linear momentum of a robot is governed by a set of equations that are linear in the contact forces. The only issue with this model is angular momentum, whose dynamics are bilinear (because of dependency on the cross product of limb positions and contact forces) [10]. Here we neglect angular momentum and focus only on translational dynamics: nonetheless, recent works have shown the potential of the combined use of McCormick envelopes and binary indicators for the piecewise approximation of bilinear laws in the context of motion planning of legged robots [10]. Although it is not problematic, the integration of these techniques in the presented framework goes beyond the scope of this paper and will be object of future improvements.

It is straightforward to directly transcribe a hybrid optimal control problem using PWA dynamics into a mixed-integer optimization problem, i.e. a problem with both continuous and discrete-valued variables [10], but such models have proved difficult to solve at real-time rates. On the other hand, stability and optimal control of PWA systems have been extensively developed in the context of hybrid MPC [11], [12]: for these models, in fact, not only several stabilizing optimal controllers have been developed [13], but also the nature of the resulting feedback law and optimal-value function has been deeply investigated [14], [15], [16]. As in the straightforward mixed-integer case, however, the complexity of the optimal policy also scales poorly with the problem dimensions, making explicit solutions complex to compute, store, and evaluate even for relatively small problems.

In this paper we propose a novel algorithm that allows online generation of suboptimal solutions of mixed-integer programs arising in hybrid MPC. The method we propose consists in the offline generation of a map that, given the current state of the system, returns one or more feasible binary assignments (i.e., mode sequences), reducing online computations to the solution of a small number of QPs. Through the use of a sampling approach the complexity of this map is very limited and, despite the suboptimality of the proposed control scheme, closed-loop stability of the feedback law is proved. A further contribution of this work is an algorithm for the incremental approximation of QP feasible sets; this is the key ingredient that allows the application of the overall method to large scale systems. Simulation results of a simple planar humanoid that balances by making contact with its environment are presented to validate the proposed controller. In conclusion, we remark that, even though the synthesized controller is specific to a particular partition of the domain of the PWA system (i.e., the geometry of the environment in case of the push recovery problem), it is not complicated to extend this method to consider parametric domains, for example, treating wall positions as additional fictitious states. The analysis of this aspect is left to future developments.

A. Notation

Let $a \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$. We denote with $(a, b) \in \mathbb{R}^{n+m}$ the vertical concatenation of a and b. Weighted 2-norms are denoted as $\|a\|_A^2 := a^T A a$, with $A \in \mathbb{R}^{n \times n}$. The space of positive definite (semidefinite) symmetric matrices of dimension $n \times n$ is denoted as $\mathbb{S}^n_{\succ 0}$ ($\mathbb{S}^n_{\succeq 0}$). I^n represents the $n \times n$ identity matrix. $\mathrm{int}(\mathcal{S})$ and $\mathrm{conv}(\bar{\mathcal{S}})$ denote the interior and the convex hull of the set $\mathcal{S}$, respectively. All physical units may be assumed to be in meters/kilograms/seconds.

II. Hybrid and Explicit MPC Background

Motivated by the observations presented in Section I, in this work we consider the class of discrete-time PWA dynamical systems in the form

$$x_{t+1} = A_i x_t + B_i u_t + c_i \quad \text{if} \quad (x_t, u_t) \in \mathcal{D}_i, \qquad (1)$$

where $i \in \{1, \dots, s\}$, $x_t \in \mathbb{R}^n$ denotes the state vector at the generic sampling time $t \in \mathbb{N}$, $u_t \in \mathbb{R}^m$ represents the input vector, and the sets $\mathcal{D}_i \subset \mathbb{R}^{n+m}$ are polytopic domains. The overall domain of the system $\mathcal{D} := \bigcup_{i=1}^s \mathcal{D}_i$ is assumed to be completely well-posed, i.e., $\mathrm{int}(\mathcal{D}_i) \cap \mathrm{int}(\mathcal{D}_j) = \emptyset \ \forall \ i, j \in \{1, \dots, s\}$ if $i \neq j$ [12]. In order to avoid inconsistencies, we further suppose $\mathcal{D}$ to be connected and the map $x_{t+1}(x_t, u_t)$ from (1) to be continuous on $\mathcal{D}$. Finally, we assume: $(x, u) = 0 \in \mathrm{int}(\mathcal{D}_1)$, $c_1 = 0$ (i.e., the origin is an equilibrium for the dynamics (1) in mode $i = 1$), and the pair A_1, B_1 to be stabilizable.

With the aim of regulating system (1) to the origin, given the initial state $\bar{x}$ of the system, we consider the optimal control problem

$$\min_{\boldsymbol{x}, \boldsymbol{u}} \ \|x_N\|_P^2 + \sum_{t=0}^{N-1} \|x_t\|_Q^2 + \|u_t\|_R^2 \qquad (2a)$$

$$\text{subject to } x_0 = \bar{x}, \qquad (2b)$$

$$\text{dynamics (1)} \ \forall \ t \in \{0, \dots, N-1\}, \qquad (2c)$$

$$x_N \in \mathcal{X}^{\mathrm{g}}, \qquad (2d)$$

where $\boldsymbol{x} := (x_0, \dots, x_N)$, $\boldsymbol{u} := (u_0, \dots, u_{N-1})$, $P \in \mathbb{S}^n_{\succ 0}$, $Q \in \mathbb{S}^n_{\succeq 0}$, $R \in \mathbb{S}^m_{\succ 0}$, and the polytope $\mathcal{X}^{\mathrm{g}} \subset \mathbb{R}^n$ represents a goal terminal set with $x = 0 \in \mathrm{int}(\mathcal{X}^{\mathrm{g}})$. We denote with $V^\star(x)$ the optimal value function of problem (2), and with $\boldsymbol{x}^\star(x) := (x_0^\star(x), \dots, x_N^\star(x))$, $\boldsymbol{u}^\star(x) := (u_0^\star(x), \dots, u_{N-1}^\star(x))$ the arguments that minimize it. Through the introduction of sN binary variables (one per domain per sampling time), a big-M reformulation can be employed to cast problem (2) as a Mixed-Integer Quadratic Program (MIQP) (see, e.g., the procedure presented in [12] in case of piecewise linear systems). At each sampling time the state of the system $\bar{x}$ is measured, the MIQP is solved, and the control action $u_0^\star(\bar{x})$ is applied to the system in a receding horizon fashion.

In opposition to the pure online mixed-integer approach, explicit MPC techniques can solve problem (2) offline, computing the map $\boldsymbol{u}^\star(x)$ explicitly. To do that, it is necessary to reason in terms of single mode sequences $\boldsymbol{z} := (z_0, \dots, z_{N-1}) \in \{1, \dots, s\}^N$. Once a mode sequence is fixed, system (1) becomes a time-varying affine system

$$x_{t+1} = A_{z_t} x_t + B_{z_t} u_t + c_{z_t}. \qquad (3)$$

Exploiting (3) recursively, it is now possible to explicitly express the state evolution as a function of the initial state

and the input sequence. Substituting the resulting expression in (2), we get a condensed multiparametric QP (mpQP) [14] with the characteristic dependence on the initial state $\bar{x}$

$$\min_{\boldsymbol{u}} \; q^{\boldsymbol{z}}(\boldsymbol{u}, \bar{x}) \tag{4a}$$

$$\text{subject to } G^{\boldsymbol{z}}\boldsymbol{u} \leq w^{\boldsymbol{z}} + E^{\boldsymbol{z}}\bar{x}, \tag{4b}$$

where $q^{\boldsymbol{z}}(\boldsymbol{u}, \bar{x})$ is a quadratic function of $(\boldsymbol{u}, \bar{x})$, and $G^{\boldsymbol{z}}$, $w^{\boldsymbol{z}}$, and $E^{\boldsymbol{z}}$ are properly assembled matrices that enforce stage and terminal constraints. We will denote with $V^{\boldsymbol{z}\star}(x)$ the (convex piecewise quadratic) optimal value function of problem (4) and with $\boldsymbol{u}^{\boldsymbol{z}\star}(x)$ the related (PWA continuous) optimal input sequence [14]. For problem (4), we define the feasible set as $\mathcal{X}^{\boldsymbol{z}\mathrm{f}} := \{x \mid \exists \boldsymbol{u} : G^{\boldsymbol{z}}\boldsymbol{u} \leq w^{\boldsymbol{z}} + E^{\boldsymbol{z}}x\}$. By definition, $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ is the orthogonal projection of the polytopic constraint set (4b) from $\mathbb{R}^{Nm+n}$ (the space of $(\boldsymbol{u}, x)$) onto $\mathbb{R}^n$ (the space of x) and, hence, is a polytope itself. We then define the feasible set of the hybrid MPC problem (2) as $\mathcal{X}^{\mathrm{f}} := \{x \mid \exists \boldsymbol{z} : x \in \mathcal{X}^{\boldsymbol{z}\mathrm{f}}\}$ and we denote the set of feasible mode sequences for a state x as $\mathcal{Z}^{\mathrm{f}}(x) := \{\boldsymbol{z} \mid x \in \mathcal{X}^{\boldsymbol{z}\mathrm{f}}\}$.

Typically, explicit hybrid MPC algorithms share the following structure [15], [16]:

- Enumeration of all the feasible mode sequences $\{\boldsymbol{z} \mid \mathcal{X}^{\boldsymbol{z}\mathrm{f}} \neq \emptyset\}$. This is done through a backwards reachability analysis and, especially for long horizons N, it might return in a huge number of sequences (see, e.g., Section V-A).
- Solution of a mpQP for each feasible mode sequence. The results of this step are the laws $\boldsymbol{u}^{\boldsymbol{z}\star}(x)$ and $V^{\boldsymbol{z}\star}(x)$, whose complexity is exponential in both the number of constraints and optimization variables.
- Comparison of the optimal-value functions $V^{\boldsymbol{z}\star}(x)$ to determine the pointwise optimal binary assignments. This requires non-convex programming techniques and results in a discontinuous PWA control policy defined on non-convex domains [17]. (This step is sometimes substituted by a runtime comparison.)

The computational complexity of these steps limits the applicability of explicit hybrid MPC to systems of very small dimension [16].

III. Stable MPC for High-Dimensional Hybrid Systems

Interestingly, the gray area between a pure online approach and the explicit offline solution of the MPC problem has not received much attention in the MPC literature. In this work we propose a method which aims to split the computational burden between the offline and the online phases in a way that is more effective than the two extreme approaches individually. The key observation of the algorithm we propose is that the stabilizing properties of the feedback law from (2) are not directly related to the optimality of the MIQP, but just to its feasibility (see Theorem 1). Once that feasibility of the MIQP is guaranteed, a Lyapunov argument can be made just assuming optimality of the condensed QP (4). In light of this observation, we propose to generate a map offline that, given the state of the system, returns a set of feasible mode sequences; reducing the online phase of the controller to the solution of a limited number of QPs. Beside the clear online advantage, the algorithm we propose uses the relatively large feasible sets $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ to cover the state space instead of tiny critical regions, as it is done in explicit MPC [14].

In the following we will assume the availability of the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x) \subseteq \mathcal{Z}^{\mathrm{f}}(x)$ that for all $x \in \mathcal{X}^{\mathrm{f}}$ returns a nonempty set of feasible mode sequences; details about the construction of this map are given in next section. Denoting with $\tau \in \mathbb{N}$ the absolute time step from the instant ($\tau = 0$) where the controller is turned on, Algorithm 1 describes how the feedback control is derived from the measured state $\bar{x}$.

Algorithm 1: Feedback control u_0^τ for state $\bar{x}$ at time τ

```
Result: Feedback u_0^τ
if τ = 0 then
    Pick any feasible mode sequence z ∈ Z̃^f(x̄)
else
    z := (z_1^{τ-1}, ..., z_{N-1}^{τ-1}, 1)
end
Solve QP (4) to get u^{z⋆}(x̄), V^{z⋆}(x̄)
Set u^τ = u^{z⋆}(x̄), z^τ = z, V^τ = V^{z⋆}(x̄)
while Additional computation time remains do
    Pick a new z ∈ Z̃^f(x̄)
    Solve QP (4) to get u^{z⋆}(x̄), V^{z⋆}(x̄)
    if V^{z⋆}(x̄) < V^τ then
        Set u^τ = u^{z⋆}(x̄), z^τ = z, V^τ = V^{z⋆}(x̄)
    end
end
```

In order to guarantee stability, Algorithm 1 requires the solution of only one QP. Nonetheless, the performance, in terms of cost (2a), of the feedback can be improved by solving additional (feasible) QPs until the the allocated time for the feedback computation is over. Moreover, the solution of these QPs can be parallelized, keeping the overall solution time potentially equal to the time necessary to solve a single QP. The online evaluation of the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ consists in checking membership of $\bar{x}$ to a set of polytopes in halfspace representation; this requires a negligible amount of time and can be parallelized as well.

The following theorem addresses the closed-loop stability of system (1) with the feedback from Algorithm 1.

Theorem 1: Assume that:

- H1 P in (2a) is the solution of the discrete algebraic Riccati equation for the system $x_{t+1} = A_1 x_t + B_1 u_t$ with the related optimal feedback $K \in \mathbb{R}^{m \times n}$.
- H2 $\mathcal{X}^{\mathrm{g}}$ in (2d) is an invariant constraint-admissible set for the closed-loop system $x_{t+1} = \Lambda x_t$ with $\Lambda := A_1 + B_1 K$ (i.e., $x \in \mathcal{X}^{\mathrm{g}} \Rightarrow (\Lambda^t x, K\Lambda^t x) \in \mathcal{D}_1 \forall t \in \mathbb{N}$ [18]).

Then system (1) in closed loop with the feedback control from Algorithm 1 is asymptotically stable with domain of attraction $\mathcal{X}^{\mathrm{f}}$.

Proof: Consider a generic time step τ. Call $\boldsymbol{x}^\tau$ the trajectory obtained by applying $\boldsymbol{u}^\tau$ from the initial state

$x_0^\tau \in \mathcal{X}^{\mathrm{f}}$. At time $\tau+1$, starting from the new initial state $x_0^{\tau+1} = x_1^\tau$, we exploit H2 and the fact that $x_N^\tau \in \mathcal{X}^{\mathrm{g}}$ to construct a feasible input sequence $\tilde{\boldsymbol{u}}^{\tau+1} := (u_1^\tau, \dots, u_{N-1}^\tau, Kx_N^\tau)$ that results in a feasible state trajectory $\tilde{\boldsymbol{x}}^{\tau+1} := (x_1^\tau, \dots, x_N^\tau, \Lambda x_N^\tau)$ for the mode sequence $\tilde{\boldsymbol{z}}^{\tau+1} := (z_1^\tau, \dots, z_{N-1}^\tau, 1)$. The cost of the latter control action is $\tilde{V}^{\tau+1} = V^\tau - \|x_0^\tau\|_Q^2 - \|u_0^\tau\|_R^2 - \|x_N^\tau\|_P^2 + \|x_N^\tau\|_Q^2 + \|Kx_N^\tau\|_R^2 + \|\Lambda x_N^\tau\|_P^2 = V^\tau - \|x_0^\tau\|_Q^2 - \|u_0^\tau\|_R^2$, where the second equality follows from H1. Since $\tilde{\boldsymbol{u}}^{\tau+1}$ is a feasible control, we have that $\tilde{V}^{\tau+1} \geq V^{\tau+1}$ (which holds even if a "cheaper" mode sequence is found in the allocated time in Algorithm 1) and hence $V^\tau - V^{\tau+1} \geq \|x_0^\tau\|_Q^2 + \|u_0^\tau\|_R^2 \geq 0$. Since $\{V^\tau\}_{\tau=0}^\infty$ is a nonnegative decreasing sequence, there exists a limit $\lim_{\tau\to\infty} V^\tau$ and, consequently, $\lim_{\tau\to\infty} V^{\tau+1} - V^\tau = 0$. This in turn implies $\lim_{\tau\to\infty} x_0^\tau = 0$ and $\lim_{\tau\to\infty} u_0^\tau \in \ker(R)$, hence the thesis. ■

Remark 1: Assumptions in Theorem 1, together with $(x,u) = 0 \in \mathrm{int}(\mathcal{D}_1)$, can be relaxed with a more complex formulation of problem (2), see [13]. These assumptions are made here for simplicity and also because they are generally fulfilled by the family of systems we are interested in.

IV. Offline Generation of Feasible Mode Sequences

In the developments of the previous section we have assumed the availability of a map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ that, for each feasible state $x \in \mathcal{X}^{\mathrm{f}}$, returns a nonempty set of feasible mode sequences. In this section we propose an algorithm for its computation that allows generation of such a map even for systems of dimensions that would be prohibitive for both online solution of the MIQP and explicit hybrid MPC.

A straightforward approach to the generation of $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ consists in the enumeration of all the feasible mode sequences, as done in hybrid explicit MPC, and then in the computation of the orthogonal projection of (4b) to derive the feasible set for each mode sequence. (Note that this strategy would actually generate the complete map $\mathcal{Z}^{\mathrm{f}}(x)$.) Unfortunately this naïve approach has two limitations:

- It results in an excessive number of QPs to be solved online. In fact, only a small subset of the feasible mode sequences is actually optimal for some $x \in \mathcal{X}^{\mathrm{f}}$ and, since we do not want to derive the explicit expression of the optimal value functions $V^{\boldsymbol{z}\star}(x)$, we cannot eliminate never-optimal sequences as done in [15].
- Orthogonal projection of polytopes is a very complex operation (see [19] and the references therein). Choosing the right algorithm is a very case-dependent process and, in general, it is a tradeoff between speed and numerical robustness. In practice, unless some assumptions (that typically do not hold in the MPC context) on the orientation of the polytope are made, this operation becomes prohibitive for system with roughly $n > 5$, $m > 2$, and $N > 5$. Some algorithms tailored for the computation of MPC feasible sets have been proposed [20], but they proved to be applicable only to small scale problems (e.g., $n = 4$, $m = 2$, and $N = 10$).

In order to overcome these difficulties, we adopt a sampling approach; this choice entails multiple advantages:

- Feasible mode sequences do not have to be enumerated; instead they are discovered automatically by the sampling process, which, at the same time, filters never-optimal mode sequences.
- As shown in the previous subsection, stability of hybrid MPC is a consequence of feasibility rather than optimality. In this sense a sampling rejection strategy (i.e., discard a sample if it already belongs to a feasible set) can be adopted to greatly reduce the number of mode sequences to take into account.
- Sampling allows the development of tailored algorithms for the approximation of feasible sets with low-complexity polytopes, enabling the application of these methods to high-dimensional spaces.

Algorithm 2 illustrates the generation of the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$. $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} \subseteq \mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ denotes an inner approximation of the feasible set for the mode sequence $\boldsymbol{z}$, whereas $\boldsymbol{z}^\star(\bar{x})$ represents the optimal mode sequence derived by solving (2) for the initial condition $\bar{x}$. Once the collection of polytopes $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ is generated, the evaluation of the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(\bar{x})$ returns the set $\{\boldsymbol{z} \mid \bar{x} \in \tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}\}$.

Algorithm 2: Coverage of the feasible set $\mathcal{X}^{\mathrm{f}}$

```
Result: Collection of polytopes X̃^{zf}
Initialize X̃^{zf} = ∅, ∀z ∈ {1,...,s}^N
while Samples x̄ are generated do
    if ∃z | x̄ ∈ X̃^{zf} then
        Reject x̄
    else
        Solve MIQP (2) to get z*(x̄)
        if MIQP (2) is infeasible then
            Reject x̄
        else
            Expand X̃^{z*(x̄)f} to include x̄ (Algorithm 3)
        end
    end
end
```

Algorithm 3 illustrates the details about the expansion process in Algorithm 2. This procedure is based on the Convex-Hull method for orthogonal projections, originally proposed in [21]. Given a point $\bar{x}$ which is known to belong to the feasible set $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$, the polytopic inner approximation $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} := \{x \mid Dx \leq d\}$ is expanded until $\bar{x} \in \tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$. The initialization of $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ is performed finding, through the solution of $n+1$ linear programs, a set of vertices of $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ that generate a full-dimensional simplex (see Algorithm 4). The method then consists in an iterative expansion in the direction normal to the facet of $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ whose inequality is most violated by $\bar{x}$. At each step a new vertex of $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ is found and added to the vertex representation of $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$. The latter is the most expensive operation of the algorithm but it can be easily performed even for high-dimensional polytopes (the free library Qhull, for example, implements

such a progressive construction of a convex hull [22]). For brevity, the constraint set from (4b) is here denoted as $\mathcal{C}^{\boldsymbol{z}} := \{(\boldsymbol{u}, x) \mid G^{\boldsymbol{z}} \leq w^{\boldsymbol{z}} + E^{\boldsymbol{z}} x\}$. Figure 2 illustrates the various steps of Algorithm 3 and Algorithm 4 in case of a simple two-dimensional example.

Algorithm 3: Inner approximation of the set $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$

Result: Inner approximation $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$
if $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} = \emptyset$ **then**
  Initialize $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ (Algorithm 4)
end
while $\bar{x} \notin \tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ **do**
  Call i the index of the greatest element of $D\bar{x} - d$ and D_i the ith row of D
  $(\boldsymbol{u}^\star, x^\star) := \arg\max_{(\boldsymbol{u},x)\in\mathcal{C}^{\boldsymbol{z}}} D_i x$
  Update $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} \leftarrow \mathrm{conv}(\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} \cup \{x = x^\star\})$
end

Algorithm 4: Initialization of $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$

Result: Inner approximation $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$
Pick a random direction $a \in \mathbb{R}^n$
$(\boldsymbol{u}_1, x_1) := \arg\max_{(\boldsymbol{u},x)\in\mathcal{C}^{\boldsymbol{z}}} a^T x$
$(\boldsymbol{u}_2, x_2) := \arg\min_{(\boldsymbol{u},x)\in\mathcal{C}^{\boldsymbol{z}}} a^T x$
for $i = 3, \ldots, n+1$ **do**
  Pick any hyperplane $\mathcal{H} := \{x \mid a^T x = b \in \mathbb{R}\}$ such that $x_j \in \mathcal{H}$, $\forall j \in \{1, \ldots, i-1\}$
  $(\boldsymbol{u}_i, x_i) := \arg\max_{(\boldsymbol{u},x)\in\mathcal{C}^{\boldsymbol{z}}} \mathrm{sign}(a^T\bar{x} - b) a^T x$
end
Initialize $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}} = \mathrm{conv}(\bigcup_{i=1}^{n+1} \{x = x_i\})$

A fundamental property of the algorithm we propose is that coverage of the feasible set $\mathcal{X}^{\mathrm{f}}$ can be achieved after a finite number of samples. A sketch of proof for this claim is the following: Let us denote with $\mathcal{X}^{\boldsymbol{z}\star} \subseteq \mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ the region of state space where the mode sequence $\boldsymbol{z}$ is optimal. As long as there exists a full-dimensional subset of $\mathcal{X}^{\boldsymbol{z}\star}$ that has not been included in any $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$, the event of a sample (drawn from a uniform distribution) ending up in this region has a finite probability. Every time that this event happens, a vertex of $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ is added to the vertex-representation of $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$. Since $\mathcal{X}^{\boldsymbol{z}\mathrm{f}}$ has a finite number of vertices and $\mathcal{X}^{\boldsymbol{z}\star}$ is contained in their convex hull, there is a finite upper bound to the number of events that can happen. The finiteness of the event probability and the bound on the maximum number of events, are together sufficient conditions to claim that after a sufficiently high number of samples an event (i.e., a sample in a full-dimensional uncovered region of $\mathcal{X}^{\mathrm{f}}$) cannot occur.

The previous observation implies that the number of samples after which the map from Algorithm 2 is such that the feedback controller is stabilizing is finite. Moreover, note that, due to the rejection sampling in Algorithm 2, our algorithm does not necessarily converge to the optimal MIQP solution, but rather to a feasible (and stabilizing) solution.

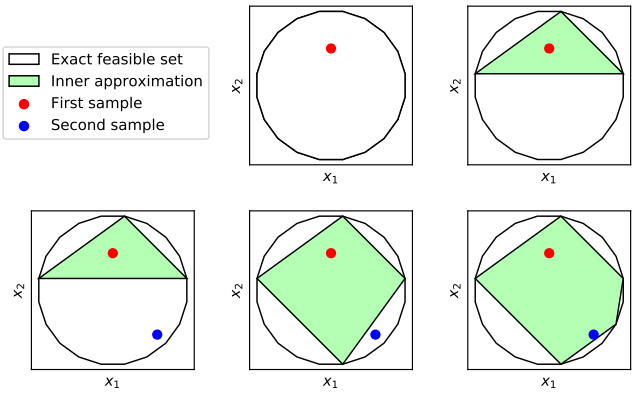

Fig. 2. Result of Algorithm 3 for a two-dimensional example. From left to right: first sample inside the orthogonal projection, inclusion of the first sample in the inner approximation of the projection, second sample, first expansion for the inclusion of the second sample, inclusion of the second sample.

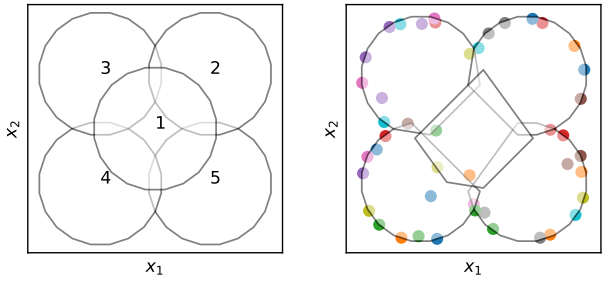

Fig. 3. Result of the application of Algorithm 2 to a synthetic two-dimensional example. Left: five overlapping feasible sets numbered by their cost (e.g., whenever feasible, set 1 is always preferable). Right: inner approximation after 10^4 samples (only samples that required the expansion of a set are shown). The proposed algorithm limits automatically the complexity of the representation of the feasible sets in the regions where there is overlapping, whereas it converges to the exact shape in the regions where only a single mode sequence is feasible.

The main advantage of the method we propose is that it automatically adapts the complexity of the approximated feasible sets $\tilde{\mathcal{X}}^{\boldsymbol{z}\mathrm{f}}$ to our needs. Generally speaking, feasible sets might be very complex polytopes defined by a large number of facets; however, if our goal is closed-loop stability, we actually do not need a detailed representation of these sets: we only care about coverage of the state space. In this sense, in the regions where multiple feasible sets overlap, it is not necessary to derive a detailed description of their boundaries. Algorithm 2 performs this reduction of complexity automatically: simplifying the coverage process, reducing memory requirements, and speeding up online evaluations. Figure 3 depicts this effect for a simple example. This coverage requirement resembles the one in the LQR Trees approach [23] which also covers the state with locally stable policies, rather than seeking global optimality.

V. Simulation Results

In this section we validate the proposed controller with two systems. First we consider a simple inverted pendulum that is allowed to interact with an elastic wall in order to keep the

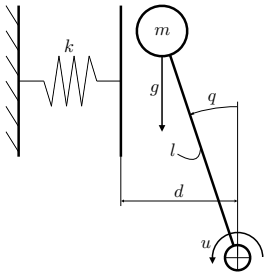

Fig. 4. Linear inverted pendulum with elastic wall.

vertical position. Then we test the controller with a simple two-dimensional humanoid that can make or break contacts with a complex environment in order to recover from a push.

A. Inverted Pendulum with Elastic Wall

Consider the inverted pendulum depicted in Figure 4 with $m = 1$, $l = 1$, $d = 0.1$, $g = 10$, and $k = 1000$. The state of the continuous time system is $x := (q, \dot{q})$ and the dynamics are linearized around the vertical configuration $q = 0$. The system has two modes: not in contact with the wall (mode 1), in contact with the wall (mode 2). Discretizing the model with the explicit Euler scheme with a sampling time $h = 0.01$, we get a model in the form (1), with

$$A_1 = \begin{bmatrix} 1 & 0.01 \\ 0.1 & 1 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}, \quad c_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$
$$\mathcal{D}_2 = \{(x, u) \mid (-0.12, -1) \leq x \leq (0.1, 1), -4 \leq u \leq 4\},$$
$$A_2 = \begin{bmatrix} 1 & 0.01 \\ -9.9 & 1 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}, \quad c_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$
$$\mathcal{D}_2 = \{(x, u) \mid (0.1, -1) \leq x \leq (0.12, 1), -4 \leq u \leq 4\}.$$

We then synthesize the MPC controller with $N = 10$, $Q = I^2$, $R = 1$, and $\mathcal{X}^{\mathrm{g}}$ equal to the maximal invariant constraint-admissible set for the system in mode 1 [18].

Figure 5 shows the coverage of the feasible set $\mathcal{X}^{\mathrm{f}}$ obtained by applying Algorithm 2 with $5 \cdot 10^4$ samples. The result is compared with the plot of the feasible sets of all the feasible mode sequences. In this case, all the possible $s^N = 1024$ mode sequences are actually feasible, resulting in a huge map $\mathcal{Z}^{\mathrm{f}}(x)$; on the other hand, through the sampling process only 17 mode sequences are selected, resulting in a very compact map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ that covers the entire feasible set. The maximum number of overlapping feasible sets for $\mathcal{Z}^{\mathrm{f}}(x)$ is 257 whereas for $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ it is only 6, resulting in a dramatic reduction in the number of QPs to be solved online.

Figure 6 shows the state-space trajectories obtained by simulating the closed-loop system from a set of initial states $(\bar{x} = (0.095, -0.5 + i0.05)$ for $i = 0, \ldots, 16)$ chosen to force the system to switch between modes. The trajectories derived using the feedback from Algorithm 1 are blue in the foreground, whereas the ones obtained solving the MIQP (2) are depicted in green in the background. Here we assumed to have enough time to solve all the feasible QPs that the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ returns at a given sampling time. In this case, it is not possible to distinguish between the two controllers: this

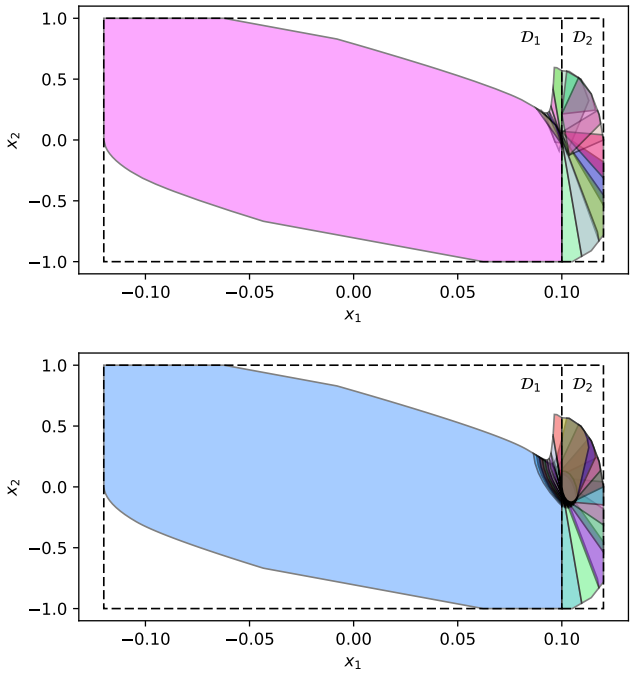

Fig. 5. Coverage of the feasible set of the MIQP (2) for the inverted pendulum. Top: inner approximations of the feasible sets from Algorithm 2. Bottom: feasible sets of all the mode sequences. The proposed algorithm requires only 17 feasible mode sequences out of a total of 1024.

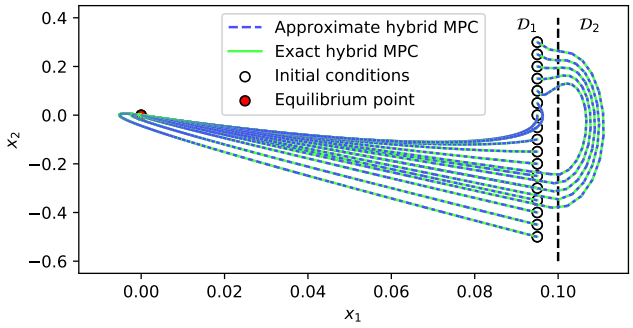

Fig. 6. Closed loop trajectories for the inverted pendulum with the feedback from Algorithm 1 (blue) and from the hybrid MPC controller (2) (green).

equivalence is due to the relatively long horizon N. In fact, it is known that with the proposed formulation of the MPC problem, if the value of N is sufficiently large to make the terminal constraint redundant, the trajectory planned at the initial time will coincide with the closed-loop trajectory of the system. Consequently, the optimal mode sequence at each time step will be the time-shifted copy of the one from the previous time; which is exactly the "first try" in Algorithm 1. Hence, if the map $\tilde{\mathcal{Z}}^{\mathrm{f}}(x)$ contains the optimal mode sequence for the initial state $\bar{x}$, the two controllers will behave exactly the same on the nominal system.

B. Two-Dimensional Humanoid in Complex Environment

In this subsection we evaluate the performance of the controller from Algorithm 1 with the two-dimensional humanoid

robot shown in Figure 1. In this simple model we consider only the translational dynamics of the body.

The massless arms are free to move in the plane and they are velocity-controlled, while the feet have fixed positions in contact with the ground. Using the acronyms "l" left, "r" right, "h" hand, "f" foot, and "b" body, we denote with $q \in \mathbb{R}^2$ the position of a part of the robot and with $f \in \mathbb{R}^2$ the contact force applied to a limb. The hands are allowed to interact with the environment, generating forces $f_{\rm lh}$ and $f_{\rm rh}$, that are used to control the position of the body (colors in Figure 1 indicate the surfaces that each hand can touch). The resulting dynamics of the body is

$$m\ddot{q}_{\rm b} = f_{\rm lh} + f_{\rm rh} + f_{\rm lf} + f_{\rm rf} + mg,$$

with $m = 1$ the mass of the body and gravitational acceleration $g = (0, -10)$. Denoting with $f^{\rm n}$ and $f^{\rm t}$ the normal and tangential contact forces applied to a moving limb, and with $\delta^{\rm n}$ and $\dot{\delta}^{\rm t}$ the related normal penetration and relative tangential velocity, we employ the linear contact models $f^{\rm n} = -\kappa\delta^{\rm n}$ and $f^{\rm t} = -\beta\dot{\delta}^{\rm t}$, with $\kappa = 200$ and $\beta = 1000$.

Considering a frame with horizontal axis q_1 and vertical axis q_2, the wall for the left hand (red in Figure 1) is in position $q_1 = 0.5$, the floor (black) is at $q_2 = -0.5$, and the walls for the right hand (blue) are at $q_1 = -0.35$ and $q_2 = 0$. The relative position of each limb with respect to the body has to lie inside a 0.4 by 0.4 square set (green in Figure 1). The centers of these regions (in body frame) are in the position $\hat{q}_{\rm lh} = (0.3, 0)$, $\hat{q}_{\rm rh} = (-0.3, 0)$, $\hat{q}_{\rm lf} = (0.2, -0.5)$, and $\hat{q}_{\rm rf} = (-0.2, -0.5)$. The equilibrium state of the system is such that the body is in position $q_1 = q_2 = 0$, the limbs are in the center of their box-shaped domains, the weight is equally distributed between the feet, and velocities and tangential forces are zero. Note that, in this configuration, the hands are not in contact with the walls. The state and the input vector are then

$$x := (q_{\rm lh} - \hat{q}_{\rm lh}, q_{\rm rh} - \hat{q}_{\rm rh}, q_{\rm b}, \dot{q}_{\rm b}) \in \mathbb{R}^8,$$
$$u := (\dot{q}_{\rm lh}, \dot{q}_{\rm rh}, f_{\rm lf} + mg/2, f_{\rm rf} + mg/2) \in \mathbb{R}^8,$$

Velocity bounds are the same for each part of the robot $\|\dot{q}\|_\infty \leq 1$ whereas normal contact forces have to lie in the interval $[0, 2m\|g\|]$ for the feet and $[0, m\|g\|]$ for the hands. Tangential contact forces have to lie in the friction cone (orange in Figure 1), with friction coefficient $\mu = 0.5$. Equations of motion are discretized with the explicit Euler scheme with a sampling time $h = 0.1$.

In order to stabilize the robot in the equilibrium state, we set the controller parameters to $Q = M^T\bar{Q}M$, $\bar{Q} = R = I^8$, $N = 10$, and for $\mathcal{X}^{\rm g}$ we employ the maximal invariant constraint-admissible set for the system in the equilibrium mode (the matrix $M \in \mathbb{R}^{8\times 8}$ is introduced to penalize the relative position of the limbs with respect to the body). Dividing the domain of the right hand position in four convex sets (in contact with the top of the table, in contact with the side of the table, over the table, on the side of the table) and the domain of the left hand in two sets (in contact with the wall, not in contact with the wall), the number of modes for the PWA dynamics is $s = 5$ (modes in which both the hands are in contact at the same time can be removed automatically since they generate empty domains $\mathcal{D}_i$).

Running Algorithm 2 with 10^5 samples, we get a total number of feasible sets $\tilde{\mathcal{X}}^{\mathbf{z}{\rm f}}$ equal to 593, which is satisfactorily low compared to all the potential $s^N = 5^{10} \approx 10^7$ mode sequences. In Figure 7 we show the trajectories of the robot in closed loop with the controller from Algorithm 1 starting from three different initial states. These are chosen in such a way that, in each motion, the robot has to interact with a different surface of the environment in order to recover the equilibrium state. Comparing the costs obtained applying the exact feedback from the MIQP (2) with the results in Figure 7, we have that only for the first motion (contact with the top of the table) is there a difference in cost, and even in this case the loss in optimality is just 1.11%; the reason for this is once again related to the relatively long horizon N as in Section V-A. For these three initial configurations, the solution of the MIQP (2) required 2.3 s, 9.6 s, and 3.0 s whereas the solution of one of the feasible QPs (4) required on average 26 ms, 24 ms, and 27 ms (using Gurobi 7.0.2 on a 2.4 GHz Intel Core i7). In conclusion, to quantify the level of optimality of the proposed control scheme, we considered 611 feasible random initial conditions $\bar{x} \in \mathcal{X}^{\rm f}$. In only 11 of 611 cases was there no set $\tilde{\mathcal{X}}^{\mathbf{z}{\rm f}}$ covering the sample, meaning that the coverage of the feasible set is almost complete after the 10^5 samples. For the remaining samples, on average 6.71 and at most 39 feasible mode sequences were provided by the map $\tilde{\mathcal{Z}}^{\rm f}(x)$. In these cases, the loss of optimality as a function of the maximum number of QPs we decide to solve online in Algorithm 1 was 39.0% for 1 QP, 13.3% for 2 QPs, 3.30% for 5 QPs, 2.00% for 10 QPs, and 1.86% if we solve all the feasible QPs returned by $\tilde{\mathcal{Z}}^{\rm f}(x)$.

VI. Conclusions and Future Works

In this work we have presented a framework for multi-contact feedback stabilization of robotic systems. We adopted a PWA model for the description of the robot's dynamics that is able to capture the non-smooth nature of the problem but, at the same time, is still tractable from a computational point of view. Leveraging recent theoretical advances in the field of hybrid MPC, we have developed a control algorithm that is able to generate nearly-optimal feedback in real time for systems of high dimensions. We proved the effectiveness of the method on a two-dimensional model of humanoid, showing that the proposed algorithm behaves almost identically to the exact hybrid MPC controller.

Future works will be focused on developing a high-performance implementation of the algorithm and testing it on systems with higher dimension, parameterized environments, and real hardware.

Source Code

The source code for all the simulations presented in this work can be found at `https://github.com/TobiaMarcucci/py-mpc/tree/humanoids2017`.

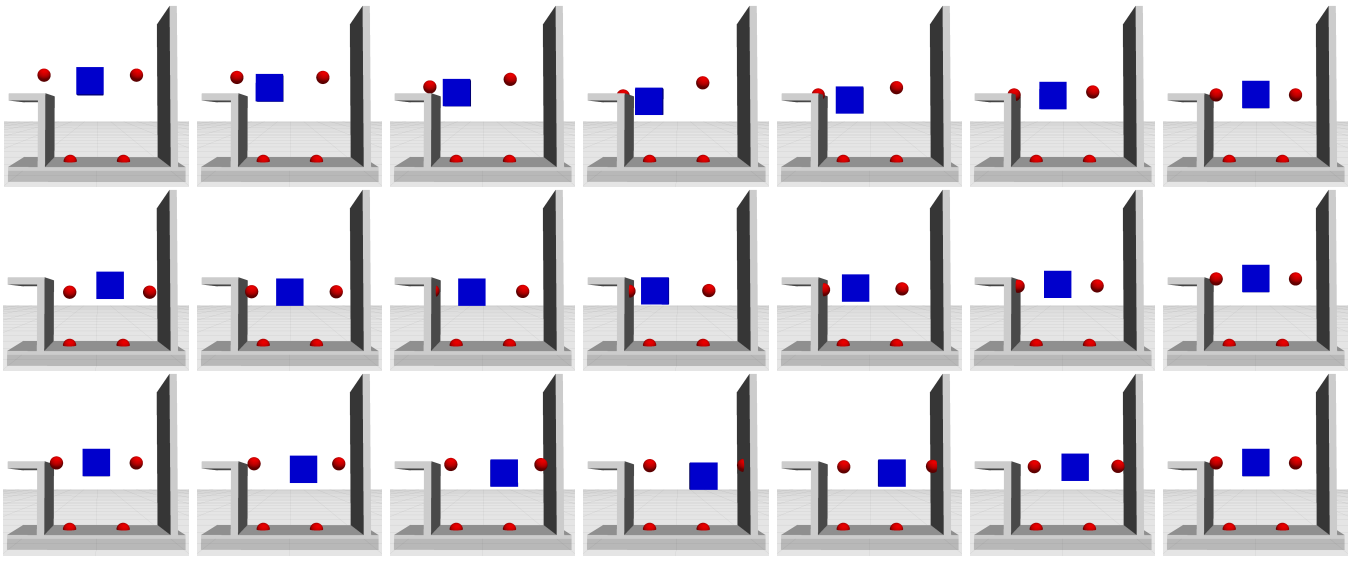

Fig. 7. Trajectories of the two-dimensional humanoid in feedback with the controller from Algorithm 1 for three different initial states. From top to bottom: contact with the top of the table for $\bar{x} = (0, 0.15, -0.1, 0.15, -0.05, 0.1, -1, -0.5)$, contact with the side of the table for $\bar{x} = (0.1, -0.1, 0.1, -0.1, 0.1, -0.05, -1, -0.5)$, and contact with the wall for $\bar{x} = (0, 0, 0, 0, 0, 0, 1, -0.5)$. From left to right: frame at time $\tau = 0, 1, 2, 4, 8, 16, 100$. Configurations at time $\tau = 100$ are practically indistinguishable from the nominal configuration.

ACKNOWLEDGEMENT

The authors would like to acknowledge the support from MIT-UNIPI Project: Fast Multi-Contact Dynamic Planning, coordinated by M. Gabiccini, COAN CA 09.01.04.0; NASA award NNX16AC49A; and the Fannie and John Hertz Foundation. The views expressed in this paper are those of the authors themselves and are not endorsed by NASA. The authors also thank Twan Koolen for many helpful discussions and insights.

REFERENCES

[1] S. Kuindersma *et al.*, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, no. 3, pp. 429–455, 2016.

[2] T. A. Johansen, I. Petersen, and O. Slupphaug, "On explicit suboptimal LQR with state and input constraints," in *Proceedings of the 39th IEEE Conference on Decision and Control*, IEEE, 2000.

[3] P.-B. Wieber, R. Tedrake, and S. Kuindersma, "Modeling and control of legged robots," in *Springer Handbook of Robotics*, pp. 1203–1234, Springer, 2016.

[4] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001.

[5] T. Koolen, T. de Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, Part 1: Theory and application to three simple gait models," *International Journal of Robotics Research*, 2012.

[6] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.

[7] M. Johnson *et al.*, "Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials," *Journal of Field Robotics*, vol. 32, pp. 192–208, Mar. 2015.

[8] J. Carpentier and N. Mansard, "Multi-contact Locomotion of Legged Robots," *Submitted to: IEEE Transactions on Robotics*, 2017.

[9] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *IEEE International Conference on Robotics and Automation*, IEEE, 2000.

[10] A. K. Valenzuela, *Mixed-integer convex optimization for planning aggressive motions of legged robots over rough terrain*. PhD thesis, Massachusetts Institute of Technology, 2016.

[11] E. Sontag, "Nonlinear regulation: The piecewise linear approach," *IEEE Transactions on automatic control*, vol. 26, no. 2, pp. 346–358, 1981.

[12] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[13] M. Lazar, W. Heemels, S. Weiland, and A. Bemporad, "Stabilizing model predictive control of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 11, pp. 1813–1818, 2006.

[14] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[15] A. Alessio and A. Bemporad, "Feasible mode enumeration and cost comparison for explicit quadratic model predictive control of hybrid systems," *IFAC Proceedings Volumes*, vol. 39, no. 5, pp. 302–308, 2006.

[16] R. Oberdieck and E. N. Pistikopoulos, "Explicit hybrid model-predictive control: The exact solution," *Automatica*, vol. 58, pp. 152–159, 2015.

[17] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," *Automatica*, vol. 41, no. 10, pp. 1709–1721, 2005.

[18] E. G. Gilbert and K. T. Tan, "Linear systems with state and control constraints: The theory and application of maximal output admissible sets," *IEEE Transactions on Automatic control*, vol. 36, no. 9, pp. 1008–1020, 1991.

[19] C. Jones, E. C. Kerrigan, and J. Maciejowski, "Equality set projection: A new algorithm for the projection of polytopes in halfspace representation," tech. rep., Cambridge University Engineering Dept, 2004.

[20] F. Scibilia, S. Olaru, and M. Hovd, "On feasible sets for mpc and their approximations," *Automatica*, vol. 47, no. 1, pp. 133–139, 2011.

[21] C. Lassez, "Quantifier elimination for conjunctions of linear constraints via a convex hull algorithm," *Symbolic and Numerical Computation for Artificial Intelligence*, pp. 103–122, 1992.

[22] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[23] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-trees: Feedback Motion Planning via Sums-of-Squares Verification," *The International Journal of Robotics Research*, vol. 29, pp. 1038–1052, July 2010.