This is the post peer-review accepted manuscript of:

S. Spedicato and G. Notarstefano, "An optimal control approach to the design of periodic orbits for mechanical systems with impacts", in Nonlinear Analysis: Hybrid Systems, Volume 23, 2017, pp 111-121.

The published version is available online at:

https://doi.org/10.1016/j.nahs.2016.08.009

An optimal control approach to the design of periodic orbits for mechanical systems with impacts

Sara Spedicato[a,*], Giuseppe Notarstefano[a]

[a]Department of Engineering, Università del Salento, Via per Monteroni, 73100 Lecce, Italy

Abstract

In this paper we study the problem of designing periodic orbits for a special class of hybrid systems, namely mechanical systems with underactuated continuous dynamics and impulse events. We approach the problem by means of optimal control. Specifically, we design an optimal control based strategy that combines trajectory optimization, dynamics embedding, optimal control relaxation and root finding techniques. The proposed strategy allows us to design, in a numerically stable manner, trajectories that optimize a desired cost and satisfy boundary state constraints consistent with a periodic orbit. To show the effectiveness of the proposed strategy, we perform numerical computations on a compass biped model with torso.

Keywords: Nonlinear optimal control, Trajectory generation, Hybrid systems, Biped walking

1. Introduction

Hybrid systems, involving both continuous and discrete dynamics, arise naturally in a number of engineering applications. In particular, many robotics tasks, such as legged locomotion, (multi-finger) manipulation and load trans-

portation with aerial vehicles, can be modeled as mechanical systems with impacts, a particular class of hybrid systems. These classes of systems experience continuous dynamics until an interaction with the surrounding environment (i.e., an impact) occurs, thus causing a state discontinuity due to impulsive forces and (possibly) a switch between different dynamics.

In this paper we concentrate on a particular aspect that has important implications both for modeling and control of robots, namely trajectory design.

Motivations. The trajectory design task has gained a lot of attention both as a preliminary task for control and as a tool to explore and understand the capabilities of the system [1],[2] (see [3] for an earlier reference). Optimization techniques allow us to design reference trajectories for robot controllers by minimizing a given cost function. Typical cost functions are (i) the distance from a desired state-input curve (which does not satisfy the dynamics) and (ii) the energy injected into the system. For example, for humanoid robot design, the distance from a desired (but unfeasible) human-like walking pattern ([4], [5]) is often considered. Additional challenges arise when the trajectory generation problem is addressed for (underactuated) mechanical systems with impacts. The impact events complicate the trajectory optimization problem since discontinuous changes in the state occur. For some systems, the underactuation and the instability of the continuous dynamics render the problem even more challenging. The optimal control theory offers powerfull tools to deal with trajectory generation problems for hybrid dynamical systems.

Literature review. The literature on optimal control of hybrid systems is quite vast, thus we report only two sets of contributions relevant for our work.

Fist, a general overview on recent advances for optimal control of hybrid systems is presented. A recent survey, focusing in particular on switched systems, is [6]. In [7] the authors propose optimal control algorithms for discrete-time linear hybrid systems which "combine a dynamic programming exploration strategy with multiparametric linear programming and basic polyhedral manipulation". In [8] a set of necessary conditions is formulated and optimization algorithms

are presented for optimal control of hybrid systems with continuous nonlinear dynamics and with autonomous and controlled switchings. The algorithm described in [9] (for hybrid systems with partitioned state space and autonomous switching) is based on a version of the minimum principle for hybrid systems providing optimality conditions for intersections and corners of switching manifolds, and thus avoiding the combinatorial complexity of other algorithms (e.g., [8]). Furthermore, the work in [10] deals with optimal mode-scheduling via a gradient descent algorithm for the particular class of autonomous switched-mode hybrid dynamical systems. In [11] and [12] the design of switching laws for switched systems with linear dynamics, based on the optimization of a quadratic criterion, is addressed.

Second, we focus on trajectory optimization for the particular class of mechanical systems with impacts. Besides results regarding systems with impulsive controls, e.g. [13], [14], we focus on the trajectory design for systems controlled by inputs of the continuous dynamics. The majority of works adopt parametric optimization methods, as, e.g., [15], [16], [17]. This means that trajectories are approximated as, for example, classical, trigonometric or Bézier polynomials and the optimization is performed with polynomial coefficients as decision variables. In other works, e.g. [18], the optimization problem is addressed via dynamics equation discretization and the optimal periodic trajectory is computed by means of an approximated cost function. Available software toolboxes are used to solve trajectory generation problems addressed in [15], [16], [17] and [18]. Recently, the optimization over jump times and/or mode sequence, as e.g., in [19] and [20], has been considered. In [19] the instants of jump are optimized, but the sequence of continuous dynamics modes is fixed. In [20] the mode sequence is also optimized by sequential quadratic programming. Focusing in particular on the trajectory generation problem for legged robots (the major robotic application regarding mechanical systems with impacts), challenges arising when dealing with underactuated robots are addressed in [21] and [22] by considering an additional "virtual" input acting on the unactuated degrees of freedom. Then, an optimal approximated trajectory of the underactuated dynamics is

3

computed by dynamics inversion. Furthermore, online trajectory optimization is addressed in [23] through a method based on iterative LQG and in [24] trajectories are designed by using an auxiliary system of differential equations and then stabilizing the generated curves through a feedback on the target system.

Contributions. As main contribution of this paper, we develop an optimal control based strategy to design periodic orbits for a class of hybrid dynamical systems with impacts. First, we formulate an optimal control problem with ad-hoc boundary conditions. These ones are provided by studying how the initial conditions and the jump conditions are related in a periodic orbit with one jump per period. Second, instead of using available softwares, we develop an ad-hoc strategy based on the combination of trajectory functional optimization with three main tools: dynamics embedding, constraint relaxation and zero finding techniques. These tools enable us to (i) deal with the undeartuated nature of dynamics, (ii) consider highly non trivial constraints and (iii) avoid the tedious search for a suitable "initial (guess) trajectory" to initialize the optimization algorithm. Furthermore, optimal control problems involved in our strategy are solved by combining the penalty function approach [25] with the Projection Operator Newton method [26]. In contrast with many strategies reported in the literature, we do not resort to approximations such as considering discrete sets of motion primitives (thus ending up with the only optimization of their parameters), and/or discrete time. In fact we consider system states and inputs as optimization variables and a second-order approximation of the optimization problem is directly constructed in continuous time. In detail, the proposed strategy is based on the following steps. By adding a fictitious input, we embed the system into a completely controllable (fully actuated) one. On this system we set up, by constraint relaxation, an unconstrained optimal control problem to find a trajectory of the system (a curve satisfying the dynamics) that minimizes a weighted L_2 distance from a desired curve. The desired curve, together with the weights of the cost functional become important parameters in the designer's hand to explore the system dynamics, that is different periodic orbits

4

that the system can execute. We solve the optimal control problem to generate a trajectory of the fully actuated system with almost null fictitious input. Differently from [21] and [22], we use this trajectory to initialize the last step of our strategy, where the optimal control problem is set up on the underactuated dynamics. A Newton update rule on the final penalty of the weighted cost functional is adopted in order to hit the desired final state. It relies on the solution of an optimal state transfer problem presented in [27], where the advantages of the method are highlighted with respect to the classic ones.

We provide a set of numerical computations showing the effectiveness of the proposed strategy. In particular, we generate a periodic gait for a three-link biped robot (compass model with torso). We perform two computations by choosing two different sets of weights. In the first scenario we try to generate a trajectory that is as close as possible to the guessed state curve. Vice-versa, in the second one, we compute a trajectory that minimizes the input effort (i.e., some sort of minimum-energy trajectory). It is worth noting that, although the three links model is relatively simple, its underactuation represents a significant challenge. Also, it is well known that for many applications, even such a reduced model of the biped dynamics is instrumental for analyzing and controlling the actual system, see, e.g., [24].

Paper organization. The paper is organized as follows. In Section 2 we introduce the model of the particular class of hybrid system we study in this paper and we present the problem formulation for the generation of periodic orbits. In Section 3 we describe the proposed strategy and in Section 4 we provide numerical computations (on a biped walking model) showing its effectiveness.

2. Problem formulation for the generation of periodic orbits

In this section we provide the optimal control formulation for solving the problem of periodic orbit generation.

2.1. Hybrid system model: underactuated systems with impacts

In this subsection we introduce the particular class of hybrid systems studied in the paper. The hybrid model consists of a continuous dynamics phase and a discrete (impulsive) jump event, occurred when the system state reaches a jump set. The mechanical system dynamics is modeled by the ordinary differential equation

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = Y_u(q)\,u, \tag{1}$$

where q, $M(q)$, $C(q,\dot{q})$, $G(q)$, u, are respectively the generalized coordinate vector, the mass matrix, the Coriolis vector, the gravity vector, and the input vector. The dynamics (1) can be written in the state space form as the continuous and control affine dynamics

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \tag{2}$$

with state $x \in \mathbb{R}^n$ defined as $x := [q^T\ \dot{q}^T]^T$ and input $u \in \mathbb{R}^m$. In particular, we assume that (i) the system is underactuated, i.e., we let $m < n$, and (ii) functions $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are (at least) C^3. When the system state, evolving as modeled in (2), reaches a jump set $S \subset \mathbb{R}^n$, a discrete impulsive event occurs, causing a discontinuity in the system state evolution. The discrete event is modeled by the jump map $\Delta : \mathbb{R}^n \to \mathbb{R}^n$, which we assume to be invertible, such that

$$x^+(t) = \Delta(x^-(t)), \tag{3}$$

where $x^-(t) := \lim_{\tau \to t^-} x(\tau)$ and $x^+(t) := \lim_{\tau \to t^+} x(\tau)$ denote the left and right limits of system trajectories satisfying (2). Finally, the overall hybrid model is:

$$\Sigma := \begin{cases} \dot{x}(t) = f(x(t)) + g(x(t))u(t), & x^-(t) \notin S \\ x^+(t) = \Delta(x^-(t)), & x^-(t) \in S. \end{cases} \tag{4}$$

2.2. Optimal control problem set-up

We aim at designing periodic trajectories, minimizing a weighted L_2 distance from a desired curve, whose period T is fixed. We approach the task

by formulating an optimal control problem over the continuous dynamics (2). We impose a boundary constraint $x(T) = x_f$, where x_f is a final state that allows the system to jump exactly to the initial condition x_0, i.e., $x_0 = \Delta(x_f)$. In particular, we choose the initial state x_0 and we compute the final state by inverting the jump map, i.e., $x_f = \Delta^{-1}(x_0)$.

Let a desired curve $(x_d(\cdot), u_d(\cdot))$, satisfying $x_d(0) = x_0$, $x_d(T) = x_f$ and thus $x_d(T) = \Delta^{-1}(x_d(0))$, be given. Furthermore, let us assume that the desired curve does not reach the jump set S in the open time interval $(0, T)$. We are ready now to present the optimal control problem we aim to solve for the generation of periodic orbits,

$$
\min_{x(\cdot), u(\cdot)} \frac{1}{2} \int_0^T (\|x(\tau) - x_d(\tau)\|_Q^2 + \|u(\tau) - u_d(\tau)\|_R^2) \, d\tau
$$

$$
\text{subj. to } \dot{x}(t) = f(x(t)) + g(x(t)) \, u(t) \tag{5}
$$

$$
x(0) = x_0
$$

$$
x(T) = x_f,
$$

where $x(\cdot)$ is an absolutely continuous state trajectory, $u(\cdot)$ is a bounded (measurable) input, Q and R are positive definite weighting matrices and for some vector $z \in \mathbb{R}^k$ and matrix $W \in \mathbb{R}^{k \times k}$ we denote $\|z\|_W^2 = z^T W z$.

The time-horizon T is a design parameter together with the desired curve, the initial (or final) conditions and the weighting matrices Q and R. We stress the fact that the desired curve is not a trajectory of the system, i.e., even if it satisfies the boundary conditions, it does not satisfy the dynamics. Thus, the desired curves can be seen as "design tools" to parameterize actual trajectories of the system. Furthermore, by choosing the desired trajectory $(x_d(\cdot), u_d(\cdot))$ to satisfy the jump condition only at time T, we expect to compute an optimal (state) trajectory satisfying the same property. Otherwise, this constraint can be (implicitly) enforced by properly choosing the weight matrices Q and R.

Remark (Final state constraint and controllability). *It is worth noticing that for problem (5) to be feasible the state x_f must belong to the reachable space of the system.* □

3. Optimal control based strategy for periodic orbit design

As highlighted in the literature review, several software toolboxes are available to solve nonlinear constrained optimal control problems. Nevertheless, optimal control problems are optimization problems in the infinite dimensional space of state-input curves subject to highly non trivial constraints as the dynamics and the fixed final state constraint. This means that the solution of an algorithm is highly influenced by the desired curve and by the "initial curve" (possibly a trajectory) that initializes the algorithm. For this reason, instead of attacking the problem directly by simply applying one of the solvers available in the literature, we develop a strategy based on suitable embedding and relaxation ideas that gives a systematic method to compute periodic trajectories.

We divide our strategy into three steps that will be presented in detail in the next subsections. Here we provide an informal idea. First, we add a fictitious input to the underactuated dynamics, thus getting a fully actuated system. Second, we consider an *unconstrained* relaxation of the optimal control problem (5) by substituting the final state constraint with a penalty in the cost function. We solve the relaxed problem by using also the fictitious input, but with a high penalty. Third and final, we get rid of the fictitious input and enforce the final state constraint exactly by means of a suitable iterative method (i.e., a zero finding Newton iteration on the final state penalty).

3.1. Dynamics embedding

The first part of the strategy is called dynamics embedding and is inspired by [28]. It consists of embedding the underactuated system into a fully actuated one by adding a fictitious input $u_{emb} \in \mathbb{R}^p$, with $p = n - m$. Thus, embedding the mechanical system (1) into a fully actuated one, we get

$$M(q)\ddot{q} + C(q, \dot{q}) + G(q) = Y(q)\, u^e, \qquad (6)$$

where $u^e := [u^T\ u_{emb}^T]^T$ and $Y(q) := [Y_u(q)\ Y_{emb}(q)]$ is assumed to be invertible $\forall q$. Accordingly, the state space equations of the (embedding) fully-actuated

8

dynamics are

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + g_{emb}(x(t))u_{emb}(t), \tag{7}$$

where g_{emb} is a $\mathcal{C}^3$ vector field taking into account the action of the embedding input. The role of this additional input is to allow for the tracking of any (sufficiently regular) desired curve. This provides a useful tool to initialize the optimal control problem solved in the next part of the strategy.

3.2. Optimal control problem relaxation and continuation with respect to parameters

Next, we design an optimal control relaxation of problem (5). The relaxation involves two aspects. First, according to the dynamics embedding introduced in the previous step we consider the fully actuated version of the system. We add a penalty in the cost function for the embedding input. Second, we relax the final state constraint by adding a penalty in the cost functional. The relaxed problem is

$$\min_{x(\cdot),u(\cdot)} \frac{1}{2} \int_0^T (\|x(\tau) - x_d(\tau)\|_Q^2 + \|u(\tau) - u_d(\tau)\|_R^2$$
$$+ \rho_{emb}^2 \|u_{emb}(\tau)\|^2) \, d\tau + \frac{1}{2}\rho_f^2 \|x(T) - x_T\|^2 \tag{8}$$
$$\text{subj. to } \dot{x}(t) = f(x(t)) + g(x(t))u(t) + g_{emb}(x(t))u_{emb}(t)$$
$$x(0) = x_0,$$

where ρ_{emb} and ρ_f in $\mathbb{R}_{>0}$ are respectively the weights on the embedding input and the final state error. The target state x_T is a strategy parameter that is set to $x_d(T)$ at beginning of the strategy, but will be modified during the strategy evolution according to the root finding procedure described in the next step. The idea is to weight the embedding input and the final state error much more than the other inputs and the states. In order to avoid a conflict between the two objectives we propose a continuation procedure on the two penalties.

Problem (8) is an unconstrained optimal control problem that could be solved by means of several available tools in the literature. We use the *PRo-*

9

jection Operator Newton method for Trajectory Optimization (PRONTO) introduced in [26] and described in Appendix A. We adopt the PRONTO tool because it shows two main appealing features for our trajectory design task. First, it allows one to handle unstable dynamics in a numerically stable manner and with a "reasonable" computational effort. Second, it guarantees recursive feasibility during the algorithm evolution. That is, at each step a system trajectory is available.

We are now ready to present the second step of the strategy. First, given a desired state curve $x_d(\cdot)$ such that $x_d = [q_d^T \ \dot{q}_d^T]^T$, we can easily compute input trajectories of the fully actuated system such that the desired state curve is a system trajectory. By inverting the fully actuated dynamics model (6), we compute $u_d^e(t)$, $\forall t \in [0, T]$ as

$$u_d^e = Y(q_d)^{-1}[M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d) + G(q_d)].$$

Note that, $u_d^e(\cdot)$ depends on the choice of $Y_{emb}(\cdot)$ but once $Y_{emb}(\cdot)$ is defined, the input $u_d^e(\cdot)$ is unique.

Thus, the second step of the strategy can be informally described as follows. We denote by $\xi^e = (x(\cdot), [u^T(\cdot) \ u_{emb}^T(\cdot)]^T)$ a generic state-input curve for the fully actuated system. We set the desired curve to $\xi_d^e = (x_d(\cdot), [u_d^T(\cdot) \ 0^T(\cdot)]^T)$ and let $\xi_0^e = (x_d(\cdot), u_d^e(\cdot))$ be the initial trajectory (of the fully actuated system) for the Projection Operator Newton method (the optimal control solver). Then, following an integral penalty function approach [25], we iteratively solve problem (8) increasing the weight ρ_{emb} at each step, by means of a suitable heuristic. When the norm of the embedding input is sufficiently small (i.e., when $\|u_{emb}(\cdot)\| < \epsilon_{emb}$, where ϵ_{emb} is a given tolerance) we stop the procedure, thus obtaining an approximated trajectory of the underactuated system.

A pseudo-code description of the strategy is given in the next table (Algorithm 1). We denote $\mathcal{P}^e$ the projection operator (A.3) acting on the fully actuated system, so that $\eta^e = \mathcal{P}^e(\xi^e)$ is a trajectory of the fully-actuated system. For given ρ_{emb} and ρ_f, we denote $\texttt{PRONTO}^e$ a routine such that $\xi_{opt^e}^e = \texttt{PRONTO}^e(\xi_0^e, \xi_d^e; \rho_{emb}, \rho_f)$, i.e., it takes as inputs an initial trajectory ξ_0^e and a

desired (state-control) curve ξ_d^e, and computes an embedding optimal trajectory, which is denoted as $\xi_{opt^e}^e = (x_{opt^e}(\cdot), [u_{opt^e}^T(\cdot)\ u_{opt^e,emb}^T(\cdot)]^T)$, solving the nonlinear optimal control relaxation in (8).

Algorithm 1 Periodic orbit design strategy: step 2

Require: Initial condition x_0 and embedding desired curve ξ_d^e

   compute initial trajectory $\xi_0^e = \mathcal{P}^e(\xi_0^e)$

  for $i = 0, 1, \ldots$ **do**

     % solve relaxed optimal control for given ρ_{emb}

     $\xi_{opt^e,i}^e = \mathtt{PRONTO}^e(\xi_i^e, \xi_d^e; \rho_{emb}, \rho_f)$

     if $\|u_{emb}(\cdot)\| < \epsilon_{emb}$ exit for **end if**

     increase ρ_{emb}

     set $\xi_{i+1}^e = \xi_{opt^e,i}^e$

  end for

Ensure: $\xi_{opt^e}^e = \xi_{opt^e,i}^e$

Remark. *(Convergence) Provided that a solution to (8) with $u_{emb}(\cdot) = 0$ exists, the existence of the solution is also guaranteed, by continuity, when u_{emb} is in a neighbourhood of the origin (with suitable radius $r_{emb} \geq \epsilon_{emb}$). Thus, a solution to (8) with $\|u_{emb}(\cdot)\| < \epsilon_{emb}$ exists and the convergence to that solution follows since we are using an integral penalty function approach (see, e.g. [25]). Nevertheless, dealing with a non-convex problem, only a local convergence is ensured, provided that the algorithm is initialized with a trajectory lying inside the basin of attraction.*

3.3. Enforcing the final state constraint

We are ready to present the third and final step of the strategy. This part relies on the idea, proposed in [27], of enforcing the final state constraint of problem (5) by combining two actions: (i) increase the penalty ρ_f on the final state error, and (ii) vary the target state x_T until the candidate optimal trajectory "practically" meets the final state constraint.

The following optimal control problem is the relaxation of problem (5) subject to the original system, i.e., without dynamics embedding

$$\min_{x(\cdot),u(\cdot)} \frac{1}{2} \int_0^T (\|x(\tau) - x_d(\tau)\|_Q^2 + \|u(\tau) - u_d(\tau)\|_R^2) \, d\tau$$
$$+ \frac{1}{2}\rho_f^2\|x(T) - x_T\|^2$$
$$\text{subj. to } \dot{x}(t) = f(x(t)) + g(x(t))u(t)$$
$$x(0) = x_0.$$

(9)

Before presenting the theorem that establishes a connection between the original problem (5) and its relaxed version (9), let us state the second order sufficient condition for optimality satisfied by a local minimum of the original problem (5).

Definition 1. *(Second order sufficient condition [27]) The second order sufficient condition for a local minimum is*

$$\int_0^T \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix}^T \begin{bmatrix} H_{xx}(\tau) & H_{xu}(\tau) \\ H_{xu}^T(\tau) & H_{uu}(\tau) \end{bmatrix} \begin{bmatrix} z(\tau) \\ v(\tau) \end{bmatrix} d\tau > 0, \quad \forall z \in \mathbb{R}^n, \forall v \in \mathbb{R}^m,$$

where H is the control Hamiltonian and $H_{uu} = \frac{\partial^2 H}{\partial u^2}$, $H_{xx} = \frac{\partial^2 H}{\partial x^2}$, $H_{xu} = \frac{\partial^2 H}{\partial x \partial u}$.

The following result is at the basis of the third step of the proposed strategy.

Theorem 1. *(Equivalence of constrained and relaxed minimizers [27]) Suppose that $\xi^* = (x^*(\cdot), u^*(\cdot))$ is a local minimum of (5) satisfying the second order sufficiency condition for optimality. Then for each $\rho_f > 0$, there is a target state $x_T = x_T(\rho_f)$ such that ξ^* is a local minimizer of problem (9) satisfying the corresponding second order sufficiency condition.* $\square$

Thus, the third step of the strategy can be informally described as follows. Given an embedding optimal trajectory $\xi_{opt^e}^e$ from the previous step, we obtain the initial trajectory $\xi_0 = \mathcal{P}((x_{opt^e}(\cdot), u_{opt^e}(\cdot))$ of the original system, where $\mathcal{P}$ is the projection operator acting on the original system. Furthermore, we reduce the embedding desired curve ξ_d^e, in order to get the desired curve for the original

system as $\xi_d = (x_d(\cdot), u_d(\cdot))$. Then, adopting a penalty function approach, we iteratively solve problem (9) increasing the weight ρ_f at each step. The use of a large ρ_f can easily result in a poorly conditioned problem where the terminal cost overwhelms the remaining one. Thus, we start with a reasonably small ρ_f and we gradually increase it by means of a suitable heuristic. Note that our penalty function approach generates an optimal trajectory which approximately satisfies the final state constraint. For this reason, when the final state $x_{opt,i}(T)$ of the temporary optimal trajectory is "sufficiently close" to the desired final state (i.e., $\|x_{opt,i}(T) - x_d(T)\| \leq \delta_{tol}^f$, where δ_{tol}^f is tolerance guaranteeing the root finding convergence), we apply a Newton method for root finding on x_T to meet exactly the final state constraint. The update rule we use on x_T is the one proposed in [27]. According to the Implicit Function Theorem, (i) there exists a mapping $\beta : \mathbb{R}^n \to \mathbb{R}^n$ such that $x_{opt,i}(T) = \beta(x_{T,i})$, where $\xi_{opt,i}(\cdot) = (x_{opt,i}(\cdot), u_{opt,i}(\cdot))$ is the solution to (9) with target state $x_{T,i}$ and (ii) the first Fréchet differential of $\beta(\cdot)$ at $x_{T,i}$, i.e., $z_f \mapsto D\beta(x_{T,i}) \cdot z_f$, exists. Furthermore, provided that the linearization of $\dot{x}(t) = f(x(t)) + g(x(t))u(t)$ about $\xi_{opt,i}$ is controllable on $[0, T]$, the mapping $\beta(\cdot)$ is invertible. Thus, a Newton method for root finding is applied on the final state constraint equation $\beta(x_T) - x_f = 0$ in order to find the value of x_T such that the solution to (9) is equivalent to the solution to (5). In particular, at the iteration $i + 1$, we update the value of x_T according to the following rule: $x_{T,i+1} = x_{T,i} + D\beta(x_{T,i})^{-1}(x_f - \beta(x_{T,i}))$. The final state constraint is met when $\|x_{opt,i}(T) - x_d(T)\| < \epsilon_{tol}^f$, where ϵ_{tol}^f is the desired tolerance on the final state error.

A pseudo code description of the third step is given in the following table (Algorithm 2). We use the following notation. For a given ρ_f and x_T, we denote PRONTO the Projection Operator Newton method for the original system, so that we have $\xi_{opt} = \text{PRONTO}(\xi_0, \xi_d; \rho_f, x_T)$. Note that, as for Algorithm 1, convergence of Algorithm 2 directly follows since it is a properly formulated penalty function approach.

Algorithm 2 Periodic orbit design strategy: step 3

Require: $\xi_{opt^e}^e$ and ξ_d^e

  reduce embedding desired curve

  $\xi_d = (x_d(\cdot), u_d(\cdot))$

  % project $(x_{opt^e}(\cdot), u_{opt^e}(\cdot))$ to get an initial trajectory for `PRONTO`

  $\xi_0 = \mathcal{P}((x_{opt^e}(\cdot), u_{opt^e}(\cdot)))$

  for $i = 0, 1, \ldots$ **do**

    compute: $\xi_{opt,i} = \texttt{PRONTO}(\xi_i, \xi_d; \rho_f, x_{T,i})$

    if $\|x_{opt,i}(T) - x_d(T)\| < \epsilon_{tol}^f$ **then** exit **end if**

    if $\|x_{opt,i}(T) - x_d(T)\| > \delta_{tol}^f$ **then**

      increase ρ_f

    else

      $x_{T,i+1} = x_{T,i} + D\beta(x_{T,i})^{-1}(x_d(T) - \beta(x_{T,i}))$

    end if

    set: $\xi_{i+1} = \xi_{opt,i}$

  end for

Ensure: $\xi_{opt} = \xi_{opt,i}$

---

4. An illustrative example: three link planar biped robot

As an example, we apply the proposed technique to a planar biped walking model. The model adopted in this paper was introduced in [29].

4.1. Biped model

The three rigid links biped with four lumped masses consists of a torso and two legs of equal length connected at the hip. The hybrid walking model consists of a continuous swing phase model and a discrete jump event model.

The swing phase model describes the motion of the swing leg to develop a walking step. Let θ_1, θ_2, θ_3 denote the orientation of the stance leg, the swing leg and the torso, respectively. Furthermore, let u_1 be the torque applied between the stance leg and the torso while the torque u_2 acts between the swing

leg and the torso. The swing phase model via Euler-Lagrange equations is given by

$$M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}) + G(\theta) = U \tag{10}$$

where $\theta = [\theta_1\ \theta_2\ \theta_3]^T$ and $M(\theta)$, $C(\theta,\dot{\theta})$, $G(\theta)$ and U (defined in Appendix B) are respectively the mass matrix, the Coriolis vector, the gravity vector and the generalized torques vector. Defining the state $x = [\theta_1\ \theta_2\ \theta_3\ \dot{\theta}_1\ \dot{\theta}_2\ \dot{\theta}_3]^T$, and the input $u = [u_1\ u_2]^T$, we can write the dynamics (10) in state-space form (2).

The jump event model takes into account (i) the impulse force arising when the swing leg touches the ground and (ii) the switch of the leg being in contact with the ground. See [29] for more details. The jump event model is given by (3) where

$$\Delta([\theta^{-T}\ \dot{\theta}^{-T}]) = \begin{bmatrix} R & 0_{3\times3} \\ 0_{3\times3} & A(\theta^-) \end{bmatrix} \begin{bmatrix} \theta^- \\ \dot{\theta}^- \end{bmatrix} \tag{11}$$

with $A(\theta^-)$ reported in Appendix C and

$$R = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The jump event occurs when the system state reaches the jump set

$$S := \{x \in \mathbb{R}^6 | \theta_1 = \theta_1^{jmp}\} \tag{12}$$

where θ_1^{jmp} is set according to physical considerations.

4.2. Numerical computations

According to [29], we consider the following model parameters: mass of the legs $m = 5$ kg; mass of the hip $M_H = 15$ kg; mass of the torso $M_T = 10$ kg; length of the legs $r = 1$ m, and the length of the torso $l = 0.5$ m. We use as jump angle $\theta_1^{jmp} = \pi/8$. We choose as time horizon $T = 1.53$ and as initial condition $x_0 = [-22.5 \text{ deg}\ 22.5 \text{ deg}\ 20 \text{ deg}\ 50 \text{ deg/s}\ 0 \text{ deg/s}\ 90 \text{ deg/s}]^T$.

The first step of the proposed strategy requires the definition of a fully actuated dynamics and the design of a desired curve. In order to obtain a

15

fully actuated dynamics, we consider the dynamics (10) and add a (generalized) torque acting directly on the torso angle velocity $\dot{\theta}_3$. That is, the swing phase model becomes

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = U^e, \tag{13}$$

where $U^e = Y u^e$,

$$Y = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & 1 \end{bmatrix}, \tag{14}$$

and $u^e = [u_1 \ u_2 \ u_{emb}]^T$. To design a desired curve that satisfies constraints on the initial and final states, we choose as desired angles $\theta_{d,k}(\cdot)$ with $k = 1, 2, 3$ spline curves so that the initial and final point and their relative slopes, i.e., $\theta_{d,k}(0)$, $\dot{\theta}_{d,k}(0)$, $\theta_{d,k}(T)$ and $\dot{\theta}_{d,k}(T)$, can be assigned a priori. Then, the desired velocity and acceleration curves, respectively $\dot{\theta}_d(\cdot)$ and $\ddot{\theta}_d(\cdot)$, are obtained by *symbolic* time differentiation. According to the fully-actuated dynamic model (13), we compute $\forall t \in [0, T]$

$$u_d^e(t) = Y^{-1}(M(\theta_d(t))\ddot{\theta}_d(t) + C(\theta_d(t), \dot{\theta}_d(t)) + G(\theta_d(t))),$$

where $u_d^e = [u_0^T \ u_{0,emb}]^T$, with $u_0 \in \mathbb{R}^2$ and $u_{0,emb} \in \mathbb{R}$. Furthermore, as regards the update rule for the penalty parameters, at iteration $i + 1$, we set $\rho_{emb,i+1} = 2\,\rho_{emb,i}$. The same update rule is adopted for ρ_f. Numerical computations characterized by two different design objectives are presented in the following. We also invite the reader to watch the attached video showing optimal walking gaits obtained by means of the proposed strategy.

In the first computation, we choose $x_d(t) = [\theta_d(t)^T \ \dot{\theta}_d(t)^T]^T$ and $u_d(t) = u_0(t)$, $\forall t \in [0, T]$. We find the optimal trajectory choosing diagonal Q and R matrices and penalizing the angles 10^4 times the input, while the velocities 10^3 times the input. That is, $Q = \text{diag}\,[100\ 100\ 100\ 10\ 10\ 10]$ and $R = \text{diag}\,[0.01\ 0.01]$. The optimal trajectory (blue) is depicted on the left column of Figure 1 and Figure 2 together with the desired curve (red), the embedding optimal trajectory (green) and temporary trajectories of the underactuated system (black). The optimization strategy shapes the desired curve in order to obtain

a trajectory (a curve satisfying the dynamics), thus showing that in this case the desired curve is just a rough guess of the biped gait.

In the second computation, we look for a trajectory that minimizes the "energy" injected in the system. Indeed, we choose $x_d(t) = [\theta_d(t)^T \ \dot{\theta}_d(t)^T]^T$, $u_d(t) = [0 \ 0]^T$, $\forall t \in [0, T]$, and penalize the inputs 10^3 times the angles and 10^2 times the velocities. That is, we choose $Q = \text{diag} \, [0.01 \ 0.01 \ 0.01 \ 0.1 \ 0.1 \ 0.1]$ and $R = \text{diag} \, [10 \ 10]$. The optimal trajectory (blue) is depicted on the right column of Figure 1 and Figure 2 together with the desired curve (red), the embedding optimal trajectory (green) and temporary trajectories of the underactuated system (black). As expected, the desired curve and the optimal trajectory coincide at T, while the embedding optimal trajectory has a nonzero final state error. This shows the effectiveness of the third step of the strategy that enforces the final state constraint. This optimal trajectory is different from the previous one, thus showing another possible gait. In order to minimize the injected energy, the swing leg first goes behind the stance one, it gains potential energy and then, it passes in front of the stance leg. This gait requires a less amount of torque, with respect to the first gait. Approaching the final time T, the torque u_2 of the first gait (Figure 2c) is greater than $100 \ Nm$ while it approaches zero in the second gait (Figure 2d).

5. Conclusion

In this paper we have developed an optimal control based strategy to compute periodic orbits for underactuated mechanical systems with impacts. The strategy combines trajectory optimization with dynamics embedding, optimal control relaxation and root finding techniques. The proposed strategy provides a systematic and numerically robust methodology to design periodic orbits for a particular class of hybrid systems.
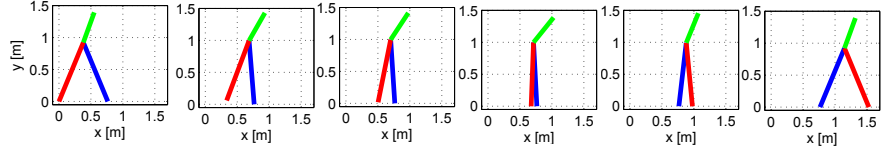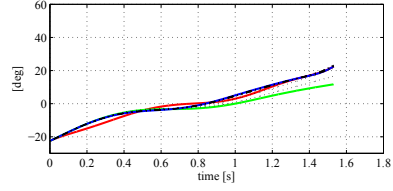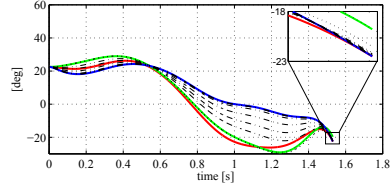
(a) Snapshots of the optimal trajectory (Gait 1)
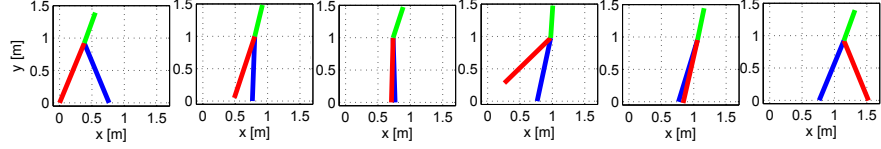
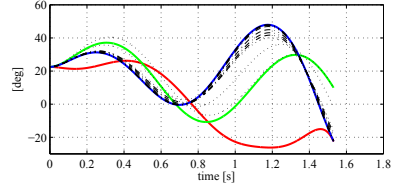(b) Snapshots of the optimal trajectory (Gait 2)
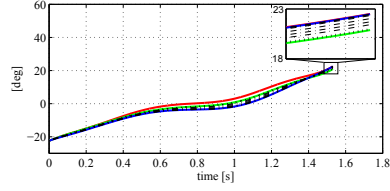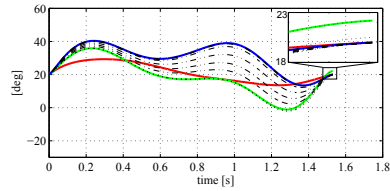

(c) Stance leg position θ_1 (Gait 1)

(d) Stance leg position θ_1 (Gait 2)

(e) Swing leg position θ_2 (Gait 1)

(f) Swing leg position θ_2 (Gait 2)

(g) Torso position θ_3 (Gait 1)
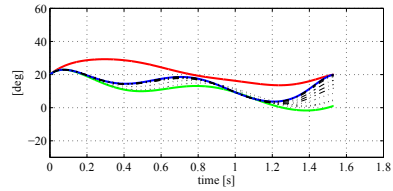
(h) Torso position θ_3 (Gait 2)

Figure 1: Gait 1 (left column) and gait 2 (right column): optimal trajectory (solid blue) compared to the desired infeasible curve (solid red) and the temporary optimal trajectory of the embedded system (solid green). Temporary optimal trajectories of the uderactuated system are depicted in black: trajectories varying ρ_f (dotted line) and updating the target state x_T (dot-dashed line).

18

(a) Input u_1 (Gait 1)

(b) Input u_1 (Gait 2)
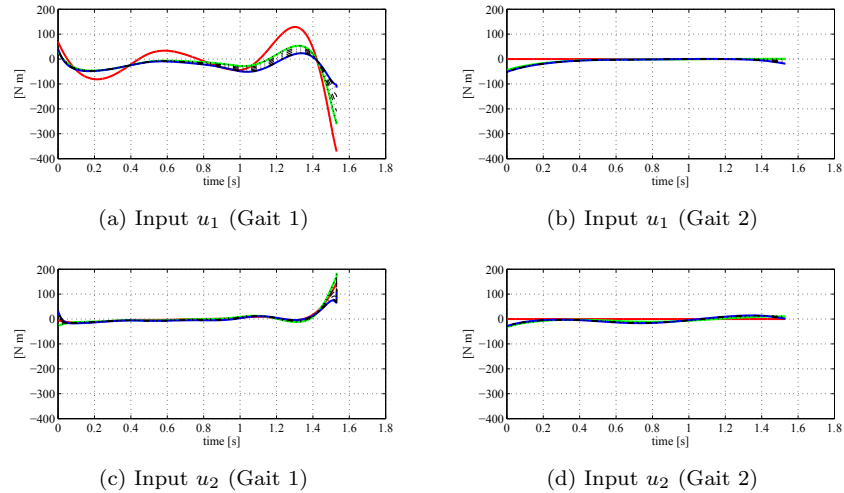
(c) Input u_2 (Gait 1)

(d) Input u_2 (Gait 2)

Figure 2: Inputs of gait 1 (left column) and gait 2 (right column): optimal trajectory (solid blue) compared to the desired infeasible curve (solid red) and the temporary optimal trajectory of the embedded system (solid green). Temporary optimal trajectories of the underactuated system are depicted in black: trajectories varying ρ_f (dotted line) and updating the target state x_T (dot-dashed line).

References

[1] M. Egerstedt, C. F. Martin, Optimal trajectory planning and smoothing splines, Automatica 37 (7) (2001) 1057–1064.

[2] M. S. Shaikh, P. E. Caines, On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules, in: Hybrid systems: Computation and control, Springer, 2003, pp. 466–481.

[3] J. T. Betts, Survey of numerical methods for trajectory optimization, Journal of guidance, control, and dynamics 21 (2) (1998) 193–207.

[4] R. Vasudevan, A. Ames, R. Bajcsy, Persistent homology for automatic determination of human-data based cost of bipedal walking, Nonlinear Analysis: Hybrid Systems 7 (1) (2013) 101–115.

[5] I. Lim, O. Kwon, J. H. Park, Gait optimization of biped robots based on human motion analysis, Robotics and Autonomous Systems 62 (2) (2014) 229 – 240.

[6] F. Zhu, P. J. Antsaklis, Optimal control of hybrid switched systems: A brief survey, Discrete Event Dynamic Systems 25 (3) (2015) 345–364.

[7] M. Baotic, F. J. Christophersen, M. Morari, Constrained optimal control of hybrid systems with a linear performance index, IEEE Transactions on Automatic Control 51 (12) (2006) 1903–1919.

[8] M. S. Shaikh, P. E. Caines, On the hybrid optimal control problem: theory and algorithms, IEEE Transactions on Automatic Control 52 (9) (2007) 1587–1603.

[9] B. Passenberg, P. E. Caines, M. Leibold, O. Stursberg, M. Buss, Optimal control for hybrid systems with partitioned state space, IEEE Transactions on Automatic Control 58 (8) (2013) 2131–2136.

[10] Y. Wardi, M. Egerstedt, M. Hale, Switched-mode systems: gradient-descent algorithms with armijo step sizes, Discrete Event Dynamic Systems 25 (4) (2015) 571–599.

[11] P. Riedinger, J.-C. Vivalda, An LQ sub-optimal stabilizing feedback law for switched linear systems, in: International Conference on Hybrid systems: computation and control, 2014, pp. 21–30.

[12] D. Corona, A. Giua, C. Seatzu, Stabilization of switched systems via optimal control, Nonlinear Analysis: Hybrid Systems 11 (2014) 1–10.

[13] Y. Huang, Y. Zhang, H. Xia, T. Liang, D. Cheng, LQ-based optimization for linear impulsive control systems mixed with continuous-time controls and fixed-time impulses, in: IEEE Conference on Decision and Control, 2012, pp. 6144–6150.

[14] A. W. Long, T. D. Murphey, K. M. Lynch, Optimal motion planning for a class of hybrid dynamical systems with impacts, in: IEEE International Conference on Robotics and Automation, 2011, pp. 4220–4226.

[15] D. Djoudi, C. Chevallereau, Y. Aoustin, Optimal reference motions for walking of a biped robot, in: IEEE International Conference on Robotics and Automation, 2005, pp. 2002–2007.

[16] G. Bätz, U. Mettin, A. Schmidts, M. Scheint, D. Wollherr, A. S. Shiriaev, Ball dribbling with an underactuated continuous-time control phase: Theory & experiments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 2890–2895.

[17] S. S. Pchelkin, A. S. Shiriaev, U. Mettin, L. B. Freidovich, L. V. Paramonov, S. V. Gusev, Algorithms for finding gaits of locomotive mechanisms: case studies for gorilla robot brachiation, Autonomous Robots (2015) 1–17.

[18] C. Liu, C. G. Atkeson, J. Su, Biped walking control using a trajectory library, Robotica 31 (02) (2013) 311–322.

[19] J. Nakanishi, A. Radulescu, S. Vijayakumar, Spatio-temporal optimization of multi-phase movements: Dealing with contacts and switching dynamics, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 5100–5107.

[20] M. Posa, C. Cantu, R. Tedrake, A direct method for trajectory optimization of rigid bodies through contact, The International Journal of Robotics Research 33 (1) (2014) 69–81.

[21] H. Sugianto, D. Lau, C. Burvill, P. Lee, D. Oetomo, Motion planning for underactuated bipedal mechanisms with kinematic constraints, in: International Conference on Advanced Robotics, 2013, pp. 1–6.

[22] T. Erez, E. Todorov, Trajectory optimization for domains with contacts using inverse dynamics, in: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, 2012, pp. 4914–4919.

[23] Y. Tassa, T. Erez, E. Todorov, Synthesis and stabilization of complex behaviors through online trajectory optimization, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4906–4913.

[24] P. La Hera, A. Shiriaev, L. Freidovich, U. Mettin, S. Gusev, Stable walking gaits for a three-link planar biped robot with one actuator, IEEE Transactions on Robotics 29 (3) (2013) 589–601.

[25] A. E. Bryson, Y.-C. Ho, Applied optimal control: optimization, estimation and control, CRC Press, 1975.

[26] J. Hauser, A projection operator approach to the optimization of trajectory functionals, in: IFAC World Congress, 2002.

[27] J. Hauser, On the computation of optimal state transfers with application to the control of quantum spin systems, in: American Control Conference, 2003, pp. 2169–2174.

[28] J. Hauser, A. Saccon, R. Frezza, Aggressive motorcycle trajectories, in: IFAC Symposium on Nonlinear Control Systems, 2004.

[29] J. Grizzle, G. Abba, F. Plestan, Asymptotically stable walking for biped robots: analysis via systems with impulse effects, IEEE Transactions on Automatic Control 46 (1) (2001) 51–64.

[30] L. Cesari, Optimization — theory and applications: problems with ordinary differential equations, Springer-Verlag, 1983.

[31] E. B. Lee, L. Markus, Foundations of Optimal Control Theory, John Wiley & Sons, 1989.

[32] J. Hauser, A. Saccon, A barrier function method for the optimization of trajectory functionals with constraints, in: IEEE Conference on Decision and Control, 2006, pp. 864–869.

Appendix A. The Projection Operator approach for the optimization of trajectory functionals

The PRojection Operator based Newton method for Trajectory Optimization (PRONTO) [26] is suitable for solving optimal control problems in the form

$$
\min \int_0^T l(x(\tau), u(\tau)) \, d\tau + m(x(T)) \tag{A.1}
$$
$$
\text{subj. to} \quad \dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0,
$$

where the initial condition x_0 is fixed and $l(x, u)$, $m(x)$ and $f(x, u)$ are (at least) $\mathcal{C}^3$ in x and u. Sufficient conditions on f, l and m ([30], [31]) guarantee the existence of optimal trajectories. In order to deal with state-input constraints, a strategy combining PRONTO with a barrier function approach is proposed in [32]. The key idea of PRONTO is that a properly designed *projection operator* $\mathcal{P}$, mapping state-control curves into system trajectories (curves satisfying the dynamics), is used to convert the dynamically constrained optimization problem (A.1) into an essentially unconstrained one. Let $\xi = (\alpha(\cdot), \mu(\cdot))$ be a bounded curve and let $\eta = (x(\cdot), u(\cdot))$ be a trajectory of the nonlinear feedback system

$$
\dot{x}(t) = f(x(t), u(t)), \qquad x(0) = x_0, \tag{A.2}
$$
$$
u(t) = \mu(t) + K(t)(\alpha(t) - x(t)),
$$

where the initial condition x_0 is given in (A.1) and the feedback gain $K(\cdot)$ is designed, e.g., by solving a suitable linear quadratic optimal control problem, in order to guarantee (local) exponential stability of the trajectory η. The feedback system (A.2) defines the nonlinear *projection operator*

$$
\mathcal{P} : \xi \mapsto \eta, \tag{A.3}
$$

mapping the curve ξ to the trajectory η. Using the projection operator to locally parameterize the trajectory manifold, problem (A.1) is equivalent to the one of minimizing the unconstrained functional $g(\xi) = h(\mathcal{P}(\xi))$, where $h(\xi) := \int_0^T l(x(\tau), u(\tau)) \, d\tau + m(x(T))$. Then, using an (infinite dimensional)

Newton descent method, a local minimizer is computed. The strength of this approach is that the local minimizer is obtained as the limit of a sequence of trajectories, i.e., curves satisfying the dynamics. The feedback system (A.2), defining the projection operator, allows us to generate trajectories in a numerically stable manner. In other words, the choice of the feedback in (A.2) is convenient from a numerical point of view. Furthermore, note that (projected) trajectories $(x_i(\cdot), u_i(\cdot))$ satisfy $x_i(0) = x_0$, according to the definition of the projection operator.

A pseudo-code of the Projection Operator Newton method is shown in the table (Algorithm 3). Let ξ_0 be an initial trajectory. Minimization of the cost functional $g(\xi)$ is accomplished iteratively. Given the current trajectory iterate ξ_i, the search direction ζ_i is obtained by solving a linear quadratic optimal control problem with cost $Dg(\xi_i) \cdot \zeta + \frac{1}{2} D^2 g(\xi_i)(\zeta, \zeta)$, where $\zeta \mapsto Dg(\xi_i) \cdot \zeta$ and $\zeta \mapsto D^2 g(\xi_i)(\zeta, \zeta)$ are respectively the first and second Fréchet differentials of the functional $g(\xi) = h(\mathcal{P}(\xi))$ at ξ_i. Then, the curve $\xi_i + \gamma_i \zeta_i$, where γ_i is a step size obtained through a standard backtracking line search, is projected, by means of the projection operator, in order to get a new trajectory ξ_{i+1}.

Algorithm 3 Projection Operator Newton method

Require: initial trajectory $\xi_0 \in \mathcal{T}$

  for $i = 0, 1, 2 \ldots$ **do**

    design K defining $\mathcal{P}$ about ξ_i

    search for descent direction

    $\zeta_i = \arg\min_{\zeta \in T_{\xi_i} \mathcal{T}} Dg(\xi_i) \cdot \zeta + \frac{1}{2} D^2 g(\xi_i)(\zeta, \zeta)$

    step size $\gamma_i = \arg\min_{\gamma \in (0,1]} g(\xi_i + \gamma \zeta_i)$;

    project $\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i)$.

  end for

Remark. *The algorithm has the structure of a standard Newton method for the minimization of an unconstrained function. The key points are the design of K defining the projection operator and the computation of the derivatives of g to "search for descent direction". It is worth noting that these two steps involve*

the solution of suitable (well known) linear quadratic optimal control problems, [26]. □

Appendix B. Biped walking model: swing phase

The mass matrix, the Coriolis vector, the gravity vector and the generalized torques vector of equation (10) are respectively

$$M(\theta) = \begin{bmatrix} (\frac{5}{4}m + M_H + M_T)r^2 & -\frac{1}{2}mr^2c_{12} & M_Trlc_{13} \\ -\frac{1}{2}mr^2c_{12} & \frac{1}{4}mr^2 & 0 \\ M_Trlc_{13} & 0 & M_Tl^2 \end{bmatrix} \tag{B.1}$$

$$C(\theta, \dot{\theta}) = \begin{bmatrix} -\frac{1}{2}mr^2s_{12}\dot{\theta}_2^2 + M_Trls_{13}\dot{\theta}_3^2 \\ \frac{1}{2}mr^2s_{12}\dot{\theta}_1^2 \\ -M_Trls_{13}\dot{\theta}_1^2 \end{bmatrix} \tag{B.2}$$

$$G(\theta) = \begin{bmatrix} -\frac{1}{2}g(2M_H + 3m + 2M_T)r\sin\theta_1 \\ \frac{1}{2}gmr\sin\theta_2 \\ -gM_Tl\sin\theta_3 \end{bmatrix} \tag{B.3}$$

$$U = \begin{bmatrix} -u_1 \\ -u_2 \\ u_1 + u_2 \end{bmatrix} \tag{B.4}$$

where

$$\begin{aligned} c_{12} &:= \cos(\theta_1 - \theta_2) & c_{13} &:= \cos(\theta_1 - \theta_3) \\ s_{12} &:= \sin(\theta_1 - \theta_2) & s_{13} &:= \sin(\theta_1 - \theta_3), \end{aligned} \tag{B.5}$$

and the model parameters are: the mass of the legs m, the mass of the hip M_H, the mass of the torso M_T, the length of the legs r, and the length of the torso l.

Appendix C. Biped walking model: relation between velocities just after and before the impact

The relation between velocities just after and before the impact is represented by $\dot{\theta}^+ = A(\theta^-)\dot{\theta}^-$. The terms of the matrix $A(\theta^-)$ are reported below.

$$A_{11} = \frac{1}{den}[2M_T\cos(-\theta_1^- - \theta_2^- + 2\theta_3^-)+$$

$$-(2m + 4M_H + 2M_T)\cos(\theta_1^- - \theta_2^-),$$

$$A_{12} = \frac{1}{\text{den}}m, \quad A_{13} = 0,$$

$$A_{21} = \frac{1}{\text{den}}[m - (4m + 4M_H + 2M_T)\cos(2\theta_1^- - 2\theta_2^-)+$$

$$2M_T\cos(2\theta_1^- - 2\theta_3^-)],$$

$$A_{22} = \frac{1}{\text{den}}[2m\cos(\theta_1^- - \theta_2^-)], \quad A_{23} = 0,$$

$$A_{31} = \frac{1}{l\text{den}}[(2mr + 2M_Hr + 2M_Tr)\cos(\theta_1^- - 2\theta_2^- + \theta_3^-)+$$

$$-2M_Hr\cos(-\theta_1^- + \theta_3^-) - (2mr + 2M_Tr)$$

$$\cos(-\theta_1^- + \theta_3^-) + mr\cos(-3\theta_1^- + 2\theta_2^- + \theta_3^-),$$

$$A_{32} = -\frac{1}{l\text{den}}rm\cos(-\theta_2^- + \theta_3^-), \quad A_{33} = 1,$$

$$\text{den} = -3m - 4M_H - 2M_T + 2m\cos(2\theta_1^- - 2\theta_2^-)+$$

$$2M_T\cos(-2\theta_2^- + 2\theta_3^-).$$