



HAL
open science

A Simulation Framework for Simultaneous Design and Control of Passive Walkers

Guilhem Saurel, Justin Carpentier, Jean-Paul Laumond

► **To cite this version:**

Guilhem Saurel, Justin Carpentier, Jean-Paul Laumond. A Simulation Framework for Simultaneous Design and Control of Passive Walkers. 2016. hal-01360450v1

HAL Id: hal-01360450

<https://hal.science/hal-01360450v1>

Preprint submitted on 5 Sep 2016 (v1), last revised 19 Jul 2017 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License

A Simulation Framework for Simultaneous Design and Control of Passive Walkers

Guilhem Saurel¹, Justin Carpentier¹ and Jean-Paul Laumond¹

Abstract—In this paper, we propose a simulation framework which simultaneously computes both the design and the control of passive walkers. The problem of computing a design and a control is formulated as a single large-scale parametric optimal control problem on hybrid dynamics with path constraints (e.g. non sliding and non slipping contact constraints). Our framework relies on state-of-the-art numerical optimal control techniques and efficient computation of the multi-body rigid dynamics. It allows to compute both the parametrized model and the control of passive walkers on different scenarios, in only few seconds on a standard computer. The framework is illustrated by several examples which highlight the interest of the approach.

I. INTRODUCTION

Passive walkers are bipedal robots essentially powered by gravity. They exploit their natural dynamics to move forward, but in the meanwhile they are unable to exhibit quasi-static behaviors. Such mechanical systems show an excellent cost of transport (CoT). They are effective platforms, both for biomechanicians to better understand the essence of bipedal walk, and for robot designers to build efficient humanoid robots.

The purpose of this paper is to propose a generic dynamic simulation framework for optimizing both the design and the controller of such robots with respect to a given cost function. An overview of the framework is given in Fig. 1.

A. Related Work

Starting with McGeer [1], many passive walkers have been designed and crafted. A great introduction to this field is given in Collins *et al.* [2]. A complete methodology to build incrementally complex passive walkers is described by Wisse *et al.* [3]. It starts from a simple compass-like model and goes to Denise, a 3D dynamic walking robot with a bisecting mechanism for the torso, two arms rigidly coupled to the hip angle, knees that are unlocked during the swing phase, and ankles engineered to steer in the direction that prevents falling. Such incremental approach reveals the critical issues in the design of walker with an anthropomorphic shape. The first issue is the transition from the simplest compass model evolving in the sagittal plane to 3D system [4], [5], [6]. A second issue is the extension of the compass model to articulated legs [1], [4], [6], [7], [8]. The addition of ankles is addressed in [6], [9], arms in [4], [6] and even neck and head in [10], [11]. The mass distribution

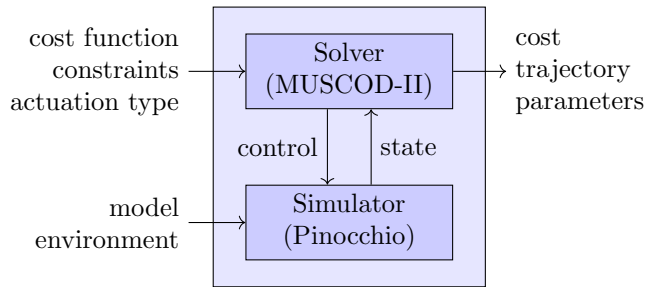


Fig. 1: Overview of the simulation framework. The simulator is described in Sec. II, and the solver is detailed in Sec. III.

for a given kinematic structure is also an important question which is addressed in [12], [13].

Beyond those mechatronical contributions, the research also focuses on the minimal energy consumption [2], [14] and on the stability properties of passive walkers [7], [10], [15], [16], [17].

By opposition to non-passive preview control approaches (e.g. Kajita [18], capture-point [19]) that require footstep planning, the control of passive walkers aims at maintaining at the best the natural steady gait that originates from the mechanical design. Passive walkers do not compute their footsteps in advance. Mechanical design and control are deeply connected and the challenge is to consider both simultaneously. The natural dynamics of passive walkers is periodic. The mechanical design tends to create limit cycles at the origin of the steady gait. The role of the controller is to maintain the system around this limit cycle in spite of environment perturbation.

In model-based optimization, the problem of finding a good controller is expressed as a constrained numerical optimization problem that benefits from solid theoretical analysis of stability [20]. This general numerical framework has been applied to passive bipedal walkers for the optimization of mass distribution [12]. It has been also instantiated to the study of passive quadrupeds [21] and then to passive walkers for the creation and analysis of efficient gaits [13]. In this last paper, the authors propose a dynamic simulation package whose scope is illustrated by various examples including passive walkers and running robots. Our paper falls in the same framework.

B. Problem statement

In this paper, we want to simultaneously solve the two following objectives:

- 1) for a given kinematic chain architecture, we want

¹ The authors are with CNRS, LAAS, Univ. de Toulouse, UPS, INSA, 7 avenue du Colonel Roche, F-31400 Toulouse, France, {saurel,carpentier,jpl}@laas.fr

to optimize the parameters of a robot (*e.g.* mass distribution, body segments lengths, slope of the ground, mean forward velocity, *etc.*),

- 2) for a given robot, we want to find the best controller with respect to a cost function (*e.g.* the cost of transport, the minimal time, *etc.*).

In short, for a given kinematic structure, we propose a generic approach to design all robot parameters together with the controller that optimizes a given cost function.

C. Contribution

The first originality of this approach is its ability to deal with complex architectures like human-like walkers, both in 2D and 3D. We do not restrict the motion to only lie in the sagittal plane. Moreover and contrary to similar works on this topic, our framework automatically computes the full dynamics. Therefore, there is no need of writing down the complex dynamic equations of polyarticulated systems. Hence, many passive walkers can be efficiently designed, optimized and compared.

Secondly, control can be chosen to be active or passive. This means that we are able to evaluate the sub-optimality of a passive controller (composed of spring-dampers) compared to an active one. We also handle periodic as well as non-periodic gaits.

Finally, we can optimize every single parameter of a given walker (slope, lengths, masses, speeds, *etc.*) with respect to a given cost function. Fig. 1 illustrates the global architecture of our framework.

D. Outline of the paper

In Sec. II, we first introduce the dynamic contact simulator at the root of the framework. We highlight how the various problem parameters are distributed either in the mechanical model or in the controller. In Sec. III, we set up the generic optimal control formulation which allows the computation of both design and control parameters. Finally, experimental results are presented in Sec. IV.

II. DYNAMIC CONTACT SIMULATION

Passive walkers are intrinsically hybrid systems. They are submitted to a continuous dynamics when the stance leg is in contact with the ground, and they are also subject to impacts when the swing foot hits the ground. In addition to this hybrid dynamics, some contact constraints must be satisfied to ensure the feasibility of the entire motion.

In the following of this section, we expose the notations, the parametric model of the walkers and the contact formulation used inside the framework.

A. Notations

In this paper, we assimilate a passive walker to a free-floating base system. We denote by $\mathbf{q} \in SE(3) \times \mathbb{R}^n$ its configuration vector, with $SE(3)$ the special Euclidian group of dimension 3 encoding the placement of the robot base and n the number of degrees of freedom (DoF). The tangent velocity and acceleration of the configuration vector are denoted by $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ respectively and live in \mathbb{R}^{6+n} . Finally, the torque applied at each joint is denoted by $\boldsymbol{\tau} \in \mathbb{R}^n$.

B. The parametric model

A passive walker is primarily a kinematic tree, *i.e.* a tree of joints where each joint has its own topology (*e.g.* revolute, free-flyer, spherical, *etc.*) and a particular placement regarding to its parent joint. The joints are the nodes of the tree. In addition, each joint supports a body, which is defined by its mass, its center of mass (CoM) and its inertia matrix. All the bodies together define the mass distribution of the model. The tree structure with the mass distribution correspond to the structural parameters of the system. The model of the passive walker is then parametrized by those two sets of parameters:

$$\text{model}(\text{tree}, \text{mass_distribution})$$

C. The parametric controller

A gait is characterized by its controller which is represented by a set of real parameters. For instance, a controller may be a set of splines which encode the torque trajectories or just the PID gains (stiffness and damping of the spring attached to the joint) in the case of pure passive controller.

$$\text{controller}(\text{control_parameters})$$

D. Continuous contact dynamics

For the continuous dynamics, we make the hypothesis of punctual rigid contact with Coulomb friction cone. The dynamic equation of the polyarticulated system with constraints can be stated as:

$$M(\mathbf{q})\ddot{\mathbf{q}} + b(\mathbf{q}, \dot{\mathbf{q}}) = S^\top \boldsymbol{\tau} + J_c(\mathbf{q})^\top \mathbf{f}_c \quad (1)$$

$$J_c(\mathbf{q})\dot{\mathbf{q}} + \dot{J}_c(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0} \quad (2)$$

where $M(\mathbf{q})$ is the joint space inertia matrix, $b(\mathbf{q}, \dot{\mathbf{q}})$ corresponds to the Coriolis, centrifugal and gravitational effects, S is a selection matrix encoding the underactuation, $J_c(\mathbf{q})$ is the contact Jacobian with \mathbf{f}_c the contact forces and \cdot^\top denotes the transpose operator.

A necessary and sufficient condition for non-sliding and non-slipping of the contact is described by the constraint that \mathbf{f}_c must remain inside the Coulomb friction cone \mathcal{K}_c . This cone reflects the fact that the normal component of the contact force is positive (the ground cannot pull), the norm of its tangential components and the normal torque are limited by the normal component.

Several LCP-based techniques exist to handle such contact modelling [22], [23]. However, they are sensitive to rounding errors. This issue may be due to the linearization of the friction cones for instance and the use of active set methods to handle the constraints. Thus, LCP-based techniques may be difficult to exploit in the context of optimal control where solvers need to precisely evaluate the derivatives of the dynamic of the system regarding to all the parameters, especially when doing convenient finite-difference approaches (as done in this paper).

We make the choice to solve (1)-(2) together, and to add the conic constraint inside the optimal control problem to enforce the Coulomb contact model [23].

The joint acceleration and contact force vectors are then given by:

$$\ddot{\mathbf{q}} = M^{-1}(I_n - J_c^t \Lambda_c^{-1} J_c M^{-1})(S^T \boldsymbol{\tau} - \mathbf{b}) - M^{-1} J_c^T \Lambda_c^{-1} J_c \dot{\mathbf{q}} \quad (3)$$

$$\mathbf{f}_c = \Lambda_c^{-1} \left(J_c M^{-1} (\mathbf{b} - S^t \boldsymbol{\tau}) - J_c \dot{\mathbf{q}} \right) \quad (4)$$

with $\Lambda_c \stackrel{\text{def}}{=} J_c M^{-1} J_c^T$ the so-called operational-space inertia matrix and I_n the identity matrix of dimension n . The dependences on \mathbf{q} and $\dot{\mathbf{q}}$ have been omitted for simplicity of notation.

E. Impact dynamics

Passive walkers are also subject to impacts. Here, we make the assumption of instantaneous inelastic impacts with a restitution coefficient sets to zero, *i.e.* the post-impact velocity of the contact point is null. The impact dynamics then leads to a discontinuity in the joint velocity space, which is described by the two following equations:

$$\dot{\mathbf{q}}^+ = (I_n - M^{-1} J_c^T \Lambda_c^{-1} J_c) \dot{\mathbf{q}}^- \quad (5)$$

$$\boldsymbol{\lambda}_c = \Lambda_c^{-1} J_c \dot{\mathbf{q}}^- \quad (6)$$

with $\dot{\mathbf{q}}^-$, $\dot{\mathbf{q}}^+$ the pre-impact and post-impact generalized velocities and $\boldsymbol{\lambda}_c$ is the impulse resulting from the impact [23]. Other impact model (*e.g.* elastic) could also be introduced without loss of generality.

F. Efficient rigid-body dynamic computation

The optimal control solver must evaluate thousands of times the multi-body dynamics either inside the numerical integration procedure or for the evaluation of the dynamic sensitivities regarding to the model and control parameters. For that purpose, we used Pinocchio [24], a whole new, open-source and efficient C++ library to model and compute the forward and the inverse dynamics of polyarticulated systems in contact. Pinocchio is written on top of the efficient Eigen C++ library [25] for Linear Algebra. It is based on Featherstone's algorithms [26] but they have been implemented in a way to take profit of branch prediction and cache mechanisms of modern processors [27].

III. OPTIMAL CONTROL APPROACH FOR DESIGN AND CONTROL

Optimal control is a powerful and generic mathematical tool which allows the exhibition of a particular solution among an infinite number of candidates. While geometric results only exist for a very limited class of systems, the past few years have seen the emergence of efficient and reliable numerical optimal control frameworks working on high dimensional and complex systems [28], [29].

In our framework, the simultaneous search of model and control parameters is set up as an optimal control problem with a prescribed cost function. This cost function reflects the objective of the gait and can be any real value function. Some examples of cost functions in the context of passive walkers are the cost of transport or even the minimal time. In addition, we add the possibility to set the duration of the motion as a

free parameter of the problem. Other free variables, like the gait stride length for example or the slope of the ground, can be stacked to the list of parameters.

A. Notations

We denote by $\mathbf{x} \stackrel{\text{def}}{=} (\mathbf{q}, \dot{\mathbf{q}})$ the state of the systems, \mathbf{u} is the control vector and \mathbf{p} is the vector of parameters, composed of both the model parameters and the aforementioned free variables. State and control trajectories are denoted by $\underline{\mathbf{x}}$ and $\underline{\mathbf{u}}$ respectively. The cost function and dynamics of the system are then written as $\ell(t, \mathbf{x}, \mathbf{u}, \mathbf{p})$ and $\frac{d\mathbf{x}}{dt} = f(t, \mathbf{x}, \mathbf{u}, \mathbf{p})$ respectively. Here, we use a slight abuse of notations to denote with the same notation both the continuous and the impact dynamics. Finally, $g(t, \mathbf{x}, \mathbf{u}, \mathbf{p})$ corresponds to the inequality constraints that must be satisfied along the path (equality constraint can be also considered without loss of generality). As mentioned in Sec. II-D, the function g is first and foremost composed of the Coulomb conic constraints.

B. The Optimal Control Problem formulation

The hybrid dynamics of passive walkers can be seen as a multi-phase system, each stage corresponding either to the single, double support or impact phases. Thereafter, the integer s refers to the index of the s^{th} stage.

The generic optimal control problem for simultaneously computing model and control parameters with multi-phase dynamics can be written as:

$$\min_{\underline{\mathbf{x}}, \underline{\mathbf{u}}, \mathbf{p}} \sum_{s=1}^S \int_{t_s}^{t_s + \Delta t_s} \ell_s(t, \mathbf{x}, \mathbf{u}, \mathbf{p}) dt \quad (7a)$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f_s(t, \mathbf{x}, \mathbf{u}, \mathbf{p}), \forall t \in [t_s, t_s + \Delta t_s] \quad (7b)$$

$$g_s(t, \mathbf{x}, \mathbf{u}, \mathbf{p}) \geq \mathbf{0}, \forall t \in [t_s, t_s + \Delta t_s] \quad (7c)$$

$$\pi(\underline{\mathbf{x}}, \underline{\mathbf{u}}, \mathbf{p}) \geq \mathbf{0} \quad (7d)$$

where π in (7d) is a function which acts both on the state and control trajectories to enforce periodicity constraints. Δt_s is the duration of the phase s , and $T = \sum \Delta t_s$ is the total duration of the motion. In case of impacts, the phase duration is reduced to 0.

C. Solving the Optimal Control Problem

Two major directions exist to solve the infinite dimensional problem (7). The first direction belongs to the so-called indirect methods. It consists in exploiting the necessary conditions for optimality, namely the Pontryagin's maximum principle [30], which transforms the problem (7) into a boundary value problem working on ordinary differential equations. However, such methods are currently unable to track path constraints.

The second direction corresponds to direct methods. Direct methods first discretize the original problem into a finite dimensional nonlinear programming problem (NLP), which is then solved with standard NLP strategies. Among NLP strategies, three of them are now popular: (i) single shooting, (ii) collocation and (iii) multiple shooting. In what follows, we briefly survey these three methods. For further details, we refer the reader to the general overview on numerical optimal control methods written by Diehl *et al.* [31].

1) *Single shooting*: discretizes the control and constraints according to a temporal grid. The state trajectory is recovered by integration of the discrete control trajectory along this grid. As single shooting method reduces the NLP to the search of a control trajectory, the optimization problem is of low dimension. However, the solver is hard to initialize if only an initial guess on the state trajectory is available or it may not converge at all in the context of unstable systems.

2) *Collocation*: discretizes both the control and the state trajectories according to a temporal grid. In addition to the classic discretized constraints, the state trajectory is enforced to match the dynamics equation (7b) at each grid node. The problem can then be easily initialized from a given state trajectory and collocation handles well unstable dynamics. However, a very fine grid is required to make the state trajectory closer to the true dynamics of the system.

3) *Multiple shooting*: takes profit of both previous methods. It works on a coarser time grid, which are called multiple shooting intervals. On each interval, the control is discretized as well as the initial state value. The final state value on the interval is then obtained by integration of the system dynamics (7b). In this way, each interval is set independent from its neighbours. And the dependencies between successive intervals is shifted as equality constraints of the NLP. The NLP remains a low dimensional problem and can be easily warm-started with an initial guess on the state trajectory. Furthermore, multiple shooting is really suited for multi-phase dynamics, as each phase is set independent to the others.

After this brief summary of NLP methods, it appears that multiple shooting is the most appropriate solution to solve optimal control problems on multi-phase dynamics with path and periodic constraints. It has been successfully applied for the modelling of human running [32]. We chose this strategy to solve the problem (7).

D. Efficient optimal control solver

Our framework relies on the 20 years old optimization package MUSCOD-II [28], a multiple shooting solver for highly nonlinear systems submitted to path equality and inequality constraints, developed inside the Optimization and Simulation group at the University of Heidelberg. MUSCOD-II handles multi-phase systems with discontinuous dynamics and periodic constraints with efficiency. While MUSCOD-II is a closed-source framework, one can depend on ACADO [29] which implements similar features.

IV. EXPERIMENTAL RESULTS

The generality of our simulation framework is illustrated by various examples depicted in Table I. Those examples include different kinematic structures and different control schemes. Doing so, we may compare different polyarticulated topologies and, for a same topology, different control schemes, including either active or passive actuators.

Once a topology is chosen, we show that it is possible to replace the active actuation by a passive spring damper

system. The motions of the first five robots are constrained to lie in the sagittal plane. Such restriction is removed for the robot with arms which is in 3D.

A. Inputs and Outputs

Fig.1 shows the general inputs and outputs of our framework. In the experiments of Tab.I, the inputs of the system are the kinematic structure of the robot with the anthropometric parameters of the body segments (see Sec.IV-C), and the cost function, constraints and actuation type (see Sec.IV-B).

In those experiments, we choose to set at a fixed value both the step duration (0.8 s) and the slope (0.05 rad) for each scenario, and let the solver optimize the step length. We limit the step length to the bounds [0.4; 1.] m.

The outputs of the system are the optimal cost of transport, together with the associated step length of the gait and the state and control trajectories of the joints during one step.

We also collect the number of iterations needed by the solver to converge and the total computation time for each experiment.

B. Cost function, constraints and actuation

1) *Cost function*: we use the same cost function in each scenario. This cost function corresponds to the classic cost of transport (CoT). The CoT is a non-dimensional quantity which reflects the energy efficiency of a locomotion pattern. By definition, the CoT is the ratio between the energy consumed by the system (E) and its weight (mg) multiplied by the traveled distance (d):



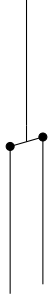
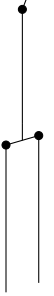
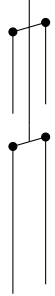
$$\text{CoT} \stackrel{\text{def}}{=} \frac{E}{mgd} \quad (8)$$

with g the gravity field value and m the mass of the system. The energy (E) consumed by the system is the sum of the potential energy mgh (with h the variation in altitude of the center of mass) with the integral of the power input $\int_{t=0}^T \|\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-\|_M \delta + \boldsymbol{\tau} \cdot \dot{\mathbf{q}} dt$. In this last formula, δ is the Dirac impulsion corresponding to the impact instant, \cdot is the dot product operator and $\|\mathbf{x}\|_M \stackrel{\text{def}}{=} \sqrt{\mathbf{x}^\top \mathbf{M} \mathbf{x}}$. Finally, on a slope with angle α , the CoT is given by:

$$\text{CoT} = \sin \alpha + \frac{\int_{t=0}^T \|\dot{\mathbf{q}}^+ - \dot{\mathbf{q}}^-\|_M \delta + \boldsymbol{\tau} \cdot \dot{\mathbf{q}} dt}{mgd} \quad (9)$$

2) *Constraints*: to reduce the dimension of the NLP, we compute a solution only for a half-step due to the periodicity and the symmetry between right and left segments. We then constrain the swing leg to be in contact with the ground only at the beginning and the end of the simulation. Then, the cyclicity and the symmetry of the motion is enforced by periodic constraints on both the initial and final configuration, velocity and torque of each joint trajectory.

INPUTS

					
Model reference	M_A	M_B	M_C	M_D	M_E
Model description	Compass	M_A with knees	M_A with torso	M_C with neck	M_C with arms
Total mass	33.1 kg	33.1 kg	60.3 kg	60.3 kg	65.9 kg
Actuation type	active	active	active	passive	active

OUTPUTS


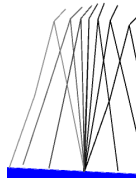
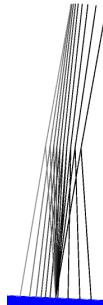
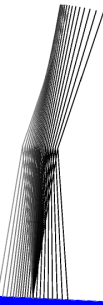
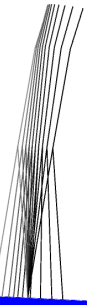
					
CoT	0.1007	0.0544	0.0618	0.2796	0.0621
Step length	0.56 m	0.85 m	0.40 m	0.40 m	0.40 m
Iterations	56	40	76	9	66
Time	2.8 s	2.9 s	2.6 s	1.2 s	4.1 s

TABLE I: Cases studies are based on six walkers models to measure the impact of the knees, torso, neck, arms and actuation type. For each example, the considered output are the cost of transport and the step length. The two last lines give the performance of the algorithm. In all examples the duration of a step is fixed (0.8s) as well as the slope of the ground (0.05rad).

3) *Actuation*: in a first stage, we propose an active actuation. The torque trajectory of each joint is modeled by piecewise cubic polynomials. An hardware implementation would then need an external source of power for the system, as a battery or an air canister, and an intelligent controller to deliver the desired torque.

In a second stage, we compare this active actuation with a passive one, which is simply a proportional derivative (PD) controller. In this case, we apply the torque τ_j to the joint j :

$$\tau_j = -K_{p_j}(q_j - q_{0_j}) - K_{d_j}\dot{q}_j \quad (10)$$

where q_j is the position of the joint and \dot{q}_j its velocity. K_{p_j} is the spring constant, q_{0_j} the free length of the spring, and K_{d_j} the damping coefficient. Those three parameters are optimized by the solver. From those spring-damper parameters, one may craft a purely passive walker. In this case, the single source of power is the gravity field. Of course, another constraint for the numerical solver is to use the same spring and damper for each symmetrical joint.

C. Body parameters

All the robots have an anthropometric mass distribution, *i.e.* segment length, mass, center of mass position and inertia tensor are scaled according to a reference height and a reference mass. Those inertial parameters follow the anthropometric table proposed in [33].

Our minimal model is composed of a pelvis, two thighs and two legs. Ankles (and knees on the model M_B) are actuated. On top of that, in the models M_C , M_D and M_E , we add a torso and a head. This head is only actuated in the model M_D . At last, we add two arms and forearms in the model M_E with actuated shoulders. All actuated joints are revolute along parallel axes.

D. Case studies

In the following, all the experiments have 9 nodes for the single support phase and 1 node for the impact phase.

1) *Influence of the knees*: the influence of the knees is highlighted by comparing models M_A and M_B in Tab. I.

The experimental results show that adding knees to a compass walker roughly divides by two the CoT while increases the optimal step length. The computational cost of adding knees is negligible.

2) *Influence of the torso*: here, we study the impact of adding a fixed torso-head segment attached to the pelvis. This situation corresponds to the models M_A and M_C of Tab. I respectively, with a non-passive actuation.

Then, adding a fixed torso on top of the pelvis leads to a 40 percent decrease of the CoT and also reduces the optimal step length down to 40 cm, which is the lower bound we allowed. In the meanwhile, the computation times remain the same.

3) *Influence of the neck*: we consider the influence of actuating the neck by comparing the models M_C and M_D while keeping the actuation type.

The experiments on those models show that the increase of the CoT is lower than one percent even if we add an actuator. However, the solver needs more time to converge.

4) *Comparison of actuation type*: this experiment considers the model M_C only and study the differences between active and passive controllers.

If we confront the results of the third and fourth columns of Tab. I, it clearly appears that the passive controller is suboptimal with respect to the active one: the optimal CoT is indeed multiplied by 4.5. Nevertheless, the computational time is lower in the passive case. This can be explained by the dimensionality of the problems: on one side, the dimensionality of the joint actuation is 4 polynomial parameters times the number of shooting nodes v.s. three scalar parameters corresponding to the spring-damper model.

5) *Beyond the 2D sagittal plane*: in this last case, we study a 3D model. In order to let the robot keep its balance, we add two actuated arms on the model M_E .

In comparison to the 2D model without arms M_C , the CoT is a bit higher, but remains lower than the first compass walker M_A . We also notice that both the computation time and the number of iterations are higher, but the solver still converges in a reasonable time, *i.e.* lower than 10 seconds.

V. DISCUSSIONS AND FUTURE WORKS

While our framework shares some common features with the Matlab package developed by Remy *et al.* [13], it differs on various aspects. The first difference lies at the optimal control method level. In [13], the authors use either collocation or shooting methods. Still, as mentioned in [31] and recalls in Sec. III-C, the multiple shooting approach is the most suited in the context of multi-phase dynamical systems submitted to impacts and constraints. In addition, the choice of the C++ programming language to write both our dynamic simulation and optimal control problem is fundamental for efficient and fast computation.

Currently, we do not check the stability of the resulting motions. As future extension, we plan to implement an approach similar to [34] to converge on optimal and stable open-loop motions.

Another extension of this framework can be achieved at the level of contact modelling. Currently, punctual contacts are the standard for passive walkers. They are easy to simulate and to mechanically integrate as feet for passive walkers. However, rolling contacts may lead to more efficient gaits, as suggested by Kuo *et al.* [35]. We recently proposed in [36] an analytical formulation of the rolling contact which is compatible with our dynamic formulation presented in Sec. II-D. The addition of this contact model inside our framework can largely improve the quality of the generated motions and enable us to deal with more complex scenarios.

The examples of Sec. IV are purely illustrative. We do not try yet to demonstrate that a kinematic structure is better than another one. On contrast, we want to emphasise the generality of the approach.

Finally, our framework is not restricted to passive walkers. On a wider scale, it can evaluate some paradigms in the design and the control of new humanoid robots [37], [38] and exoskeletons [39].

ACKNOWLEDGEMENT

We warmly thank the Simulation and Optimization group at the Interdisciplinary Center for Scientific Computing (IWR) of Heidelberg University in Germany for providing the optimal control framework MUSCOD-II. We also give thanks to Nicolas Mansard from LAAS-CNRS for his comments and suggestions.

REFERENCES

- [1] T. McGeer, "Passive dynamic walking," *The international journal of robotics research*, vol. 9, no. 2, pp. 62–82, 1990.
- [2] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [3] M. Wisse and R. Q. Van der Linde, *Delft pneumatic bipeds*. Springer Science & Business Media, 2007, vol. 34.
- [4] S. H. Collins, M. Wisse, and A. Ruina, "A three-dimensional passive-dynamic walking robot with two legs and knees," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 607–615, 2001.
- [5] R. Tedrake, T. W. Zhang, M.-f. Fong, and H. S. Seung, "Actuating a simple 3d passive dynamic walker," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5. IEEE, April 2004, pp. 4656–4661.
- [6] D. Hobbelen, T. De Boer, and M. Wisse, "System overview of bipedal robots flame and tulip: Tailor-made for limit cycle walking," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, Sept 2008, pp. 2486–2491.
- [7] Y. Ikemata, A. Sano, and H. Fujimoto, "A physical principle of gait generation and its stabilization derived from mechanism of fixed point," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*. IEEE, May 2006, pp. 836–841.
- [8] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007, vol. 28.
- [9] M. Wisse, D. G. Hobbelen, R. J. Rotteveel, S. O. Anderson, and G. J. Zeglin, "Ankle springs instead of arc-shaped feet for passive dynamic walkers," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*. IEEE, Dec 2006, pp. 110–116.
- [10] M. Benallegue, J.-P. Laumond, and A. Berthoz, "Contribution of actuated head and trunk to passive walkers stabilization," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5638–5643.
- [11] E. Falotico, N. Cauli, P. Kryczka, K. Hashimoto, A. Berthoz, A. Takanishi, P. Dario, and C. Laschi, "Head stabilization in a humanoid robot: models and implementations," *Autonomous Robots*, pp. 1–17, 2016.

- [12] J. Hass, J. M. Herrmann, and T. Geisel, "Optimal mass distribution for passivity-based bipedal robots," *The International Journal of Robotics Research*, vol. 25, no. 11, pp. 1087–1098, 2006.
- [13] C. D. Remy, K. Buffinton, and R. Siegwart, "A matlab framework for efficient gait creation," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 190–196.
- [14] P. A. Bhounsule, J. Cortell, A. Grewal, B. Hendriksen, J. D. Karssen, C. Paul, and A. Ruina, "Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1305–1321, 2014.
- [15] K. D. Mombaur, "Stability optimization of open-loop controlled walking robots," Ph.D. dissertation, University of Heidelberg, 2001.
- [16] J. W. Grizzle, G. Abba, and F. Plestan, "Asymptotically stable walking for biped robots: Analysis via systems with impulse effects," *IEEE Transactions on automatic control*, vol. 46, no. 1, pp. 51–64, 2001.
- [17] K. Byl and R. Tedrake, "Metastable walking machines," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 1040–1064, 2009.
- [18] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, Sept 2003, pp. 1620–1626.
- [19] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, Dec 2006, pp. 200–207.
- [20] M. Diehl, K. Mombaur, and D. Noll, "Stability optimization of hybrid periodic systems via a smooth criterion," *Transactions on Automatic Control*, vol. 54, no. 8, August 2009.
- [21] C. D. Remy, K. W. Buffinton, and R. Siegwart, "Stability analysis of passive dynamic walking of quadrupeds," *I. J. Robotic Res.*, vol. 29, pp. 1173–1185, 2010.
- [22] D. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with coulomb friction," in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 1, 2000, pp. 162–169 vol.1.
- [23] B. Brogliato, *Nonsmooth mechanics*. Springer, 1999.
- [24] J. Carpentier, F. Valenza, N. Mansard *et al.*, "Pinocchio: fast forward and inverse dynamics for poly-articulated systems," <https://stack-of-tasks.github.io/pinocchio>, 2015.
- [25] G. Guennebaud, B. Jacob *et al.*, "Eigen v3," <http://eigen.tuxfamily.org>, 2010.
- [26] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2008.
- [27] M. Naveau, J. Carpentier, S. Barthelemy, O. Stasse, and P. Souères, "METAPOD Template META-programming applied to dynamics: CoP-CoM trajectories filtering," in *International Conference on Humanoid Robotics*, Madrid, Spain, Nov. 2014.
- [28] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization. part 1: theoretical aspects," *Computers & Chemical Engineering*, vol. 27, no. 2, pp. 157–166, 2003.
- [29] B. Houska, H. Ferreau, and M. Diehl, "ACADO Toolkit – An Open Source Framework for Automatic Control and Dynamic Optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [30] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2012.
- [31] M. Diehl, H. Bock, H. Diedam, and P.-B. Wieber, *Fast Direct Multiple Shooting Algorithms for Optimal Robot Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 65–93.
- [32] G. Schultz and K. Mombaur, "Modeling and optimal control of human-like running," *IEEE/ASME Transactions on mechatronics*, vol. 15, no. 5, pp. 783–792, 2010.
- [33] R. Dumas, L. Cheze, and J.-P. Verriest, "Adjustments to McConville *et al.* and Young *et al.* body segment inertial parameters," *Journal of biomechanics*, vol. 40, no. 3, pp. 543–553, 2007.
- [34] K. D. Mombaur, H. G. Bock, J. P. Schlöder, and R. W. Longman, "Open-loop stable solutions of periodic optimal control problems in robotics," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 85, no. 7, pp. 499–515, 2005.
- [35] P. G. Adamczyk and A. D. Kuo, "Mechanical and energetic consequences of rolling foot shape in human walking," *Journal of Experimental Biology*, vol. 216, no. 14, pp. 2722–2731, 2013.
- [36] J. Carpentier, A. Del Prete, N. Mansard, and J.-P. Laumond, "An analytical model of rolling contact and its application to the modeling of bipedal locomotion," in *IMA Conference on Mathematics of Robotics*, Oxford, United Kingdom, Sep. 2015.
- [37] J.-P. Laumond, M. Benallegue, J. Carpentier, and A. Berthoz, "The Yoyo-Man," in *International Symposium on Robotics Research (ISRR)*, Sestri Levante, Italy, September 2015.
- [38] J. W. Grizzle, C. Chevallereau, A. D. Ames, and R. W. Sinnet, "3d bipedal robotic walking: models, feedback control, and open problems," *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 505–532, 2010.
- [39] K. H. Koch and K. Mombaur, "Exoopt – a framework for patient cetered design optimization of lower limb exoskeletons," in *IEEE International Conference on Rehabilitation Robotics (ICORR 2015)*, 2015.