



Optimal Control of Walkers with Parallel Actuation

Ludovic de Matteis, Virgile Batto, Justin Carpentier, Nicolas Mansard

► To cite this version:

Ludovic de Matteis, Virgile Batto, Justin Carpentier, Nicolas Mansard. Optimal Control of Walkers with Parallel Actuation. 2024. hal-04716938

HAL Id: hal-04716938

<https://hal.science/hal-04716938v1>

Preprint submitted on 1 Oct 2024

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.

Optimal Control of Walkers with Parallel Actuation

Ludovic De Matteis^{1,2,*}, Virgile Batto^{1,3}, Justin Carpentier², Nicolas Mansard^{1,4}

Abstract—Legged robots with complex kinematic architectures, such as parallel linkages, offer significant advancements in mobility and efficiency. However, generating versatile movements for these robots requires accurate dynamic modeling that reflects their specific mechanical structures. Previous approaches often relied on simplified models, resulting in sub-optimal control, particularly in tasks requiring the full actuator range. Here, we present a method that fully models the dynamics of legged robots with parallel linkages, formulating their motion generation as an optimal control problem with specific contact dynamics. We introduce 6D kinematic closure constraints and derive their analytical derivatives, enabling the solver to exploit nonlinear transmission and the consequent variable actuator reduction. This approach reduces peak motor torques and expands the usable range of actuator motion and force. We empirically demonstrate that fully modeling the kinematics leads to superior performance, especially in demanding tasks such as fast walking and stair climbing. Beyond serial-parallel designs, our method also addresses motion generation for fully-parallel walkers.

I. INTRODUCTION

Recent progress in biped locomotion result from the sound combination of more mature motion generation techniques and continuous improvements in robot design and hardware [1]. Several advancements have recently demonstrated the advantages of leveraging parallel kinematic chains to boost the dynamic capabilities of robots [2]. This architecture offers benefits like lighter lower limbs and improved impact absorption [3] at the cost of introducing more complex dynamics, [4] eventually making the robot more difficult to simulate and control [5]. While robots like the ones presented in Fig. 1 have already presented such hardware, effective control methods for these systems, able of exploiting their full actuator range, remain an open question.

The main historical approaches to gaited locomotion rely on solving a reduced version of the system dynamics [6], [7], [8] combined with inverse kinematics or task-space inverse dynamics. Yet the state of the art recently shifted to whole-body methods, either based on Model Predictive Control (MPC) [9], [10] or on Reinforcement Learning (RL) [11] where the locomotion decision does not rely on any heuristic but rather on the full dynamical model of the poly-articulated system [12], [13]. A first step in that direction was done in

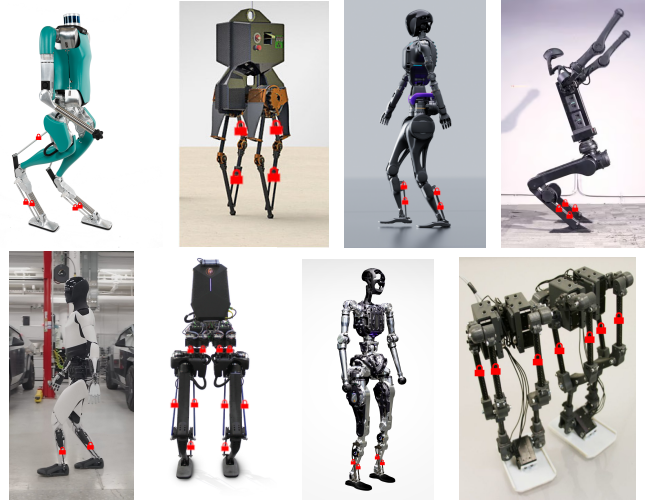


Fig. 1: From top-left to bottom-right: Digit [18], Atrias [19], Fourier GR1 [20], Unitree G1 [21], Tesla Optimus [22], Kangaroo [23], Adam [24] and Disney bipedal robot [25]. Each red lock represents a visible closure of the kinematic chain.

RL [14], but with the limiting hypothesis of a negligible transmission inertia.

The main contribution of this paper is to derive a complete modeling methodology to enable the simulation of closed kinematic chain robots, mostly targeting MPC (although these approaches are the same as those used in the simulator for RL). The dynamics of a poly-articulated robot is governed by the unconstrained equations of motion and founds efficient algorithms and their implementations in the literature [15], [16], [17]. Recent works have proposed general methods to write the dynamic of a system under contact constraints, [13] and showed a general form of its derivatives, [4], [12]. It is already accepted that closed-kinematic constraints can be casted under the same abstract scope as contact constraints, to be used in the same framework [13].

This paper bridges the missing steps to enable whole-body MPC for parallel mechanisms and experimentally demonstrates the importance of accounting for the complete robot model to exploit closed-loop actuation at best. In particular, we show that, while basic flat ground locomotion can be obtained while ignoring the transmission kinematic, it is strongly limiting for reaching higher walking velocity or climbing stairs. For such movements, using a simplified serial chain model requires additional task-specific heuristics (e.g., constraining the center of mass height, limiting joints

*This work is supported by ROBOTEX 2.0 (ROBOTEX ANR-10-EQPX-44-01 and TIRREX ANR-21-ESRE-0015), ANITI (ANR-19-P3IA-0004), by the French government (INEXACT ANR-22-CE33-0007-01 and "Investissements d'avenir" ANR-19-P3IA-0001) (PRAIRIE 3IA Institute), and by the Louis Vuitton ENS Chair on Artificial Intelligence

¹ Gepetto, LAAS-CNRS, Université de Toulouse, France

² Inria, École normale Supérieure, PSL Research University, Paris, France

³ Auctus, Inria, centre de l'université de Bordeaux, Talence, France

⁴ Artificial and Natural Intelligence Toulouse Institute, France

* Corresponding author: ludovic.de-matteis@laas.fr

range...), involving dedicated tuning by an expert. Moreover, modeling the parallel kinematics allows us to obtain movements by exploiting the full dynamic capabilities of the hardware, while only modeling the serial chain implies to pessimistically clamp the actuator range. Our work also unlocks MPC for full-parallel kinematics - i.e. where no approximate serial chain exists (e.g. the Disney bipedal robot [25]) The contributions of this work can be summarized as follows:

- We provide a way to efficiently compute the derivatives of the dynamics of a robot involving closed-loop constraints (sec. III).
- We use this new dynamic to formulate and solve an optimal control problem for the walk of a robot with parallel actuation (sec. IV).
- We highlight the limitations of using a simplified model of such a robot by comparing it to the complete modeling over different motions (sec. V).

II. BACKGROUND

A. Optimal control

Optimal control of a multibody system consists in finding the control inputs that minimize a given cost function while satisfying the system dynamics and constraints. The multiple shooting approach optimizes over both the control inputs and the states at each discrete time step with the dynamics as an explicit constraint, formulated as the following non-linear program (NLP):

$$\begin{aligned} \min_{u, x} \quad & \sum_{k=0}^{N-1} l_k(x[k], u[k]) dt + l_N(x[N]) \\ \text{s.t.} \quad & \forall k \in \llbracket 0, N-1 \rrbracket \quad x[k+1] = f_k(x[k], u[k]) \\ & \forall k \in \llbracket 0, N-1 \rrbracket \quad c_k(x[k], u[k]) \leq 0 \end{aligned} \quad (1)$$

The cost function is usually defined as the sum of running costs l_k and a terminal cost l_N . The functions f_k and c_k are defining the system dynamics and constraints at time step k , respectively. The case where f models a poly-articulated system in contact is well-known and the derivations are recalled next.

B. Multibody dynamics

The dynamic of a multibody system is well described by the Lagrangian equation of motion:

$$M(q)\ddot{q} + b(q, \dot{q}) = \tau(u) + J_c(q)^T f \quad (2)$$

where M represents the generalized inertia matrix, b the non-linear terms, q the generalized coordinates of the system, \dot{q} the generalized velocities¹ and τ the torques applied on the system joints, function of the controls u (usually, the controls correspond to the motor joints torques and all other joints torques are zero). The term f represents a force applied on the system, where J_c is the Jacobian of the contact point.

¹In our implementation, q typically contains quaternions for representing basis orientation and ball-joints configurations, hence the notation \dot{q} , although classical is abusive

For systems subject to mechanical constraints (such as foot-ground contact or kinematics-closure), these forces f arise from the satisfaction of the constraints.

C. Dynamics subject to constraints

From Gauss principle of least action [26], [27], [28], the acceleration \ddot{q} of the system under contacts should be as close as possible to its free acceleration \ddot{q}_f , with respect to the kinematic metric, while satisfying the contact constraints. This can be rewritten as the following optimization problem:

$$\begin{aligned} \min_{\ddot{q}} \quad & \|\ddot{q} - \ddot{q}_f\|_M^2 \\ \text{s.t.} \quad & J_c \ddot{q} + a_0 = 0 \end{aligned} \quad (3)$$

The constraint in (3) corresponds to the second-order time derivatives of the constraint of contact, where $a(q, \dot{q}, \ddot{q}) = J_c \ddot{q} + a_0(q, \dot{q})$ describes the relative acceleration of the contact point, that we consider here - without loss of generality - to be wanted to be zero, and where $a_0(q, \dot{q}) = \dot{J}_c \dot{q}$ is the acceleration due to the velocity only. Deriving first-order optimality conditions, (3) boils down to:

$$\underbrace{\begin{bmatrix} 0 & J_c \\ J_c^T & M \end{bmatrix}}_K \underbrace{\begin{bmatrix} f \\ \ddot{q} \end{bmatrix}}_y = \underbrace{\begin{bmatrix} -a_0 \\ \tau(u) - b \end{bmatrix}}_k \quad (4)$$

where f , the dual variables of the optimisation problem, are the contact forces applied on the system. This system can be solved to give \ddot{q} and f given the state $x = (q \quad \dot{q})$ and the controls. The solution $y = K^{-1}k$ to this problem exists and is unique whenever the matrix K is invertible, which usually happens when the matrix M is positive definite and when the constraints are not redundant. Proximal resolution leads to the least-square solution when J_c is not full rank [13].

D. Derivatives of the constrained dynamics

MPC typically uses gradient-based solvers to get the solution of (1), which implies to evaluate the derivatives of y with respect to x and u . In [4] the authors proposed the derivatives of a robot in contact with its environment. More recent work generalize these to arbitrary contacts [12], [16]. Following these works, the gradient of y with respect to $z \in \{q, \dot{q}, u\}$ can be reduced to:

$$\nabla_z y = K^{-1} \begin{bmatrix} \frac{\partial a_0}{\partial z} \\ \frac{\partial \tau}{\partial z} \end{bmatrix} \quad (5)$$

where the Inverse Dynamics (ID) function outputs the joint torques creating acceleration \ddot{q} under contact forces f .

$$ID(q, \dot{q}, \ddot{q}, f) = M\ddot{q} + J_c^T f + b \quad (6)$$

The derivatives of these terms with respect to the controls u will not be explicitated in this paper as the first term is independent on u and the second depends on the chosen actuation model. The derivatives of ID when J_c are the joint jacobians have been established [29] and are sufficient for the case of a contact between the robot and its environment, yet we will see that they are not sufficient in our case.

We can now introduce the main technical contribution of

this work, by proposing an effective form for the constraint $a = J_c \ddot{q} + a_0$ representing the kinematic closure and by exhibiting the corresponding $\frac{\partial a_0}{\partial z}$ and $\frac{\partial ID}{\partial z}$.

III. A CONSTRAINT MODELLING THE KINEMATIC CLOSURE

A. Formulation of the constraint in acceleration

We consider the mechanical linkage between two bodies of the robot, characterized by frames \mathcal{F}_1 and \mathcal{F}_2 rigidly attached to each of them. We choose to write the 6D contact constraint on the relative placement as:

$$\text{Log}({}^1M_2) = 0 \quad (7)$$

where ${}^1M_2 \in \text{SE}(3)$ is the rigid transformation between the frames and Log is the retraction from $\text{SE}(3)$ to \mathbb{R}^6 [30]. The first-order time derivative of this constraint can be expressed as the difference in spatial velocities expressed in a common frame. We choose as a convention to express all quantities in the frame \mathcal{F}_1 , giving the constraint:

$$\nu_c = {}^1\nu_1 - {}^1X_2 {}^2\nu_2 = 0 \quad (8)$$

where, following the notations introduced by Featherstone [15], ${}^1\nu_1$ and ${}^2\nu_2$ are the spatial velocities of the two bodies expressed in their respective frames, 1X_2 is the Plücker coordinate transform, i.e. the adjoint matrix of $\text{SE}(3)$, from \mathcal{F}_2 coordinates to \mathcal{F}_1 coordinates. In the same way, the second-order time derivative of (7) can be expressed as the derivative of the relative spatial velocity ν_c .

$$\begin{aligned} a_c &= {}^1a_1 - {}^1X_2 {}^2a_2 - [{}^1\nu_2 - {}^1\nu_1] \times {}^1\nu_2 \\ &= \underbrace{{}^1a_2}_{\gamma_1} - \underbrace{{}^1X_2 {}^2a_2}_{\gamma_2} + \underbrace{[{}^1\nu_1] \times {}^1\nu_2}_{\gamma_3} \end{aligned} \quad (9)$$

where the spatial cross-product $[\nu]_\times$ (*small adjoint*) is given by

$$[\nu]_\times \triangleq \begin{bmatrix} [\omega]_\times & [\mathbf{v}]_\times \\ \mathbf{0} & [\omega]_\times \end{bmatrix} \quad (10)$$

B. Differentiation of the acceleration constraint

As shown in (5), we need the derivatives of the terms γ_1 , γ_2 and γ_3 with respect to the configuration vector q and velocity \dot{q} . The first of these derivatives yields directly:

$$\frac{\partial \gamma_1}{\partial q} = \frac{\partial {}^1a_1}{\partial q} = \frac{{}^1\partial a_1}{\partial q} \quad (11)$$

Note that we make here a clear distinction between the terms $\frac{\partial A_{AB}}{\partial q}$ and $\frac{{}^A\partial a_B}{\partial q}$ as explained in [15] (section 2.10). To compute the derivative of the second term, we can use the method proposed in [31] - i.e. we search the time derivatives of γ_2 under the form $\dot{\gamma}_2 = G\dot{q}$ to deduce $\frac{\partial \gamma_2}{\partial q} = G$.

$$\begin{aligned} \frac{\partial \gamma_2}{\partial t} &= \frac{\partial {}^1X_2 {}^2a_2}{\partial t} \\ &= \frac{\partial {}^1X_2 {}^2a_2}{\partial t} + {}^1X_2 \frac{\partial {}^2a_2}{\partial t} \\ &= [{}^1(\nu_2 - \nu_1)]_\times {}^1X_2 {}^2a_2 + {}^1X_2 \frac{\partial {}^2a_2}{\partial t} \\ &= -[{}^1X_2 {}^2a_2]_\times {}^1(\nu_2 - \nu_1) + {}^1X_2 \frac{\partial {}^2a_2}{\partial t} \end{aligned} \quad (12)$$

which yields

$$\frac{\partial \gamma_2}{\partial q} = -[{}^1X_2 {}^2a_2]_\times ({}^1J_2 - {}^1J_1) + {}^1X_2 \frac{\partial {}^2a_2}{\partial q} \quad (13)$$

We proceed in a similar way for the third and last term:

$$\begin{aligned} \frac{d\gamma_3}{dt} &= \frac{\partial [{}^1\nu_1]_\times {}^1X_2 {}^2\nu_2}{\partial t} \\ &= [\frac{\partial {}^1\nu_1}{\partial t}]_\times {}^1\nu_2 + [{}^1\nu_1]_\times (\frac{\partial {}^1X_2 {}^2\nu_2}{\partial t}) + [{}^1\nu_1]_\times ({}^1X_2 \frac{\partial {}^2\nu_2}{\partial t}) \end{aligned} \quad (14)$$

which gives after development

$$\begin{aligned} \frac{\partial \gamma_3}{\partial q} &= -[{}^1\nu_2]_\times \frac{{}^1\partial \nu_1}{\partial q} - [{}^1\nu_1]_\times ([{}^1X_2 {}^2\nu_2]_\times ({}^1J_2 - {}^1J_1)) \\ &\quad + [{}^1\nu_1]_\times {}^1X_2 \frac{\partial {}^2\nu_2}{\partial q} \end{aligned} \quad (15)$$

All these terms can be accessed using rigid body dynamics algorithms such as those included in Pinocchio [16]. As the action matrices do not depend on \dot{q} , the derivatives of a_c with respect to \dot{q} are direct and left to the reader.

C. Derivatives of Inverse Dynamic

We will now look at the derivatives of ID with respect to q and \dot{q} . Let us write the expression of ID as a function of the forces applied on the joints. We will denote these forces by ϕ_k and the corresponding joint Jacobians J_k for joint j_k .

$$ID(q, \dot{q}, \ddot{q}, f) = M(q)\ddot{q} + b(q, \dot{q}) + \sum_k J_k(q)^T \phi_k(q, f) \quad (16)$$

where the ϕ_k are typically expressed in the reference frame of joint j_k denoted by \mathcal{F}_{j_k} . We will now omit the dependences to simplify the notation. The derivative with respect to the q is given by:

$$\frac{\partial ID}{\partial q} = \frac{\partial (M\ddot{q} + b)}{\partial q} + \sum_k \frac{\partial J_k^T}{\partial q} \phi_k + \sum_k J_k^T \frac{\partial \phi_k}{\partial q} \quad (17)$$

The first two terms are the classical Recursive Newton-Euler Algorithm (RNEA) derivatives, which are well established and implemented in the Pinocchio library [29], [16].

To compute the third term, we denote by j_1 and j_2 the parent joints of \mathcal{F}_1 and \mathcal{F}_2 . Following the choice of (9), the force f arising from the constraint (7) is expressed in \mathcal{F}_1 . Considering that only f acts on the system, then all forces ϕ_k are null except ϕ_1 and ϕ_2 which are:

$$\begin{aligned} \phi_1 &= {}^{j_1}X_{c_1}^* f \\ \phi_2 &= -{}^{j_2}X_{c_2}^* {}^{c_2}X_{c_1}^*(q) f \end{aligned} \quad (18)$$

where X^* is the Plücker transform on forces (dual adjoint), ${}^{j_1}X_{c_1}$ is the fixed placement of the contact frame \mathcal{F}_1 with respect to the joint frame \mathcal{F}_{j_1} (respectively ${}^{j_2}X_{c_2}$) and ${}^{c_2}X_{c_1}$ is function of q . We can see that ϕ_1 is independent of q while ϕ_2 is not. Its derivative with respect to q is given by:

$$\begin{aligned} \frac{\partial \phi_{j_2}}{\partial q} &= -\frac{\partial {}^{j_2}X_{c_1}^*}{\partial q} f \\ &= [{}^{j_2}X_{c_1}^* f]_\times^* (J_2 - {}^{j_2}X_{j_1} J_1) \end{aligned} \quad (19)$$

where, J_1, J_2 are the joint jacobians respectively expressed in \mathcal{F}_1 and \mathcal{F}_2 and the term $[f]_{\times}^*$ can be defined as follows:

$$[f]_{\times}^* \triangleq \begin{bmatrix} 0 & [f_{linear}]_{\times} \\ [f_{linear}]_{\times} & [f_{angular}]_{\times} \end{bmatrix} \quad (20)$$

With the existing derivatives of RNEA [29], this completes the computation of the derivative of the ID with respect to q .

D. Implementation

We presented the computation of the derivatives of the contact acceleration in the case of a 6D contact constraint, which is sufficient for defining the closed loop dynamic of a robot. An example of this application is proposed in the following section and applied to solve OCPs on a closed loop robot. Other definition of the constraint could be adopted such as a 3D contact constraint acting only the position of the contact frames only. The same method can be applied for these other cases, with the only difference being the definition of the constraint and its explicit differentiation.

The constrained dynamic with 6D contact constraints and 3D contact constraints have been implemented in C++ in the Crocoddyl library. The current implementation can be found in a fork on github [32] and will be merge to the main library. The complete code for generating the examples presented in the following section can also be found in Github [33].

IV. COMPLETE DESCRIPTION OF THE OCP

In this section, we give the complete description of the locomotion OCP we used in our benchmarks, including the description of the robot model.

A. Presentation of the robot

We have chosen to use in our benchmarks a mechanical design featuring a main serial chain with non-trivial actuator transmission. This corresponds to the main features of several new biped robots unveiled in early 2024 (including Unitree H1 and G1, electric Atlas, Adam and Kangaroo). On the opposite, it does not fully matches the models of other robots such as Digit or Kangaroo for which an approximate serial chain can be defined but does not correspond to a perfect equivalence (in particular the placement of the DoF in the serial chain must be changed with respect to the real robot). As the models of the robots of interest are not easily available, we introduce a novel design, combining advantages of H1 and Atlas, previously described. Our method is generic and applies to other designs, including those for which an equivalent serial chain does not exist, as shown in the companion video.

This robot is available on GitHub [34] following the work of [3]. An visual of the robot is shown in Fig. 2, revealing the different kinematic closures of the knees and ankles.

For each leg, the first 3 motors are serial motors controlling the hips degrees of freedom, next a motor controls the knee through a 4-bars linkage and 2 motors on the calf control the ankle. Each parallel actuation creates a reduction ratio between the motors and the joints that depends on the

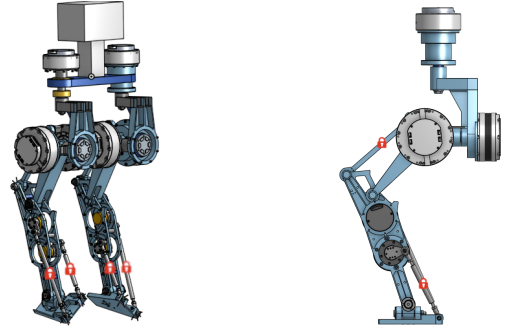


Fig. 2: Robot model used for our benchmark. Each red lock represents a closure of the kinematic chain. In our model, we represent the chain as a tree-like structure with added contact constraints by splitting the bar in two at the lock position and adding 6D contact constraints

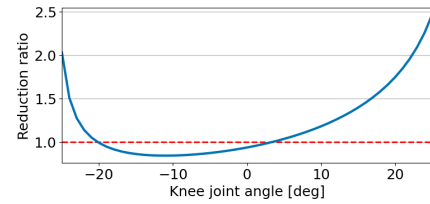


Fig. 3: Variation of the reduction ratio of the knee actuation with respect to the knee angle. The 0 angle corresponds to a nominal configuration of the robot while positive angles correspond to a stretched leg..

robot configuration. We show in Fig. 3 the variation of the reduction ratio of the knee actuation with respect to the knee angle.

The complete model including parallel linkages is compared in the benchmark against a simplified model considering only the serial joints of the robot (hip, knee, and ankle joints) and freezing the other joints in place. The non-serial motors are fixed and fictive actuation is added on serial joints, yielding a fully serial model with 6 actuated revolute joints per leg.

B. Experimental protocol

For this study, we are interested in comparing trajectories obtained by first, solving the OCP problem for the simplified model of the robot and extending the obtained optimal trajectory to the complete model (as explained in sec. IV-D), and second by solving directly the OCP for the complete model, with the dynamic under 6D closed-kinematic constraints. These trajectory are referred to as "simplified" and "complete" respectively. The former can be seen as a trajectory obtained for a simplified model, that we would want to directly apply to the robot by only solving for the motor controls.

C. Optimal control problem

Both trajectories are optimized using the same optimal control problem written in the form described in (1) using the

Crocodyl library. The OCP is based on a predefined contact pattern consisting 4 steps of alternating double support and single support phases with costs taken from [10].

a) *Regularisation costs*: We regularize the controls and the states around zero and a reference configuration respectively, and the ground contact forces around 0 when the foot is in the air, the weight of the robot when the foot is in contact with the ground in single support phases and half of it during double support phases.

b) *Impact costs*: As the dynamics of the robot under contact only constraint the relative accelerations of the contact points, we add soft constraints on the velocity and placement of the foot at the end of single support phases to ensure it lands flat, still, and at the correct height.

c) *Target costs*: The motion is defined by a COM velocity target defined in a running cost and in the terminal state regularization that accounts for the expected displacement.

d) *Additional costs*: Other costs are added to improve the realism of the trajectory (foot fly-high, center of pressure, forces in friction cones...). We do not present them here but one can find their definition in the example codes. Lastly, a cost penalizes the drift of the COM from its initial height. The weight of the cost will be changed to emphasize differences of behaviors between models.

D. From simplified model to complete model

As the leg is fully actuated and outside of singularities of the parallel linkage, an trajectory of the simplified model corresponds to a trajectory of the complete model whose motor torques can be evaluated. The simplified model trajectory is defined by a sequence of states $x_s^{[k]} = (q_s^{*[k]}, \dot{q}_s^{*[k]})$ and of controls $u_s^{[k]}$. The complete model trajectory is similarly defined by the sequences $x_c^{[k]} = (q_c^{[k]}, \dot{q}_c^{[k]})$ and $u_c^{[k]}$, where $q_c^{[k]} = \begin{pmatrix} q_s^{[k]} & q_l^{[k]} \end{pmatrix}$ (respectively \dot{q}_c), describing that the complete model state includes the simplified model state. We propose to lift the simplified model trajectory into a trajectory of the complete model including the motor torques, by solving:

$$\begin{aligned} \min_{u^{[k]}, x_c^{[k+1]}} \quad & \frac{1}{2} \|q_s^{[k+1]} - q_s^{*[k+1]}\|^2 + \frac{1}{2} \|\dot{q}_s^{[k+1]} - \dot{q}_s^{*[k+1]}\|^2 \\ \text{s.t.} \quad & x_c^{[k+1]} = f_k(x_c^{[k]}, u^{[k]}) \end{aligned} \quad (21)$$

where the state $x_c^{[k]}$ is known from the previous iteration (assuming $x^{[0]}$ is known) and the targets are the expected values for the serial part of the state, given by the simplified trajectory. We can observe that problem (21) takes the form of a 1-step optimal control problem and can therefore be solved using the same solver.

V. COMPARISON OF THE MODELS

A. Flat ground walk

We consider a walking motion on a flat terrain with a constant speed at $0.5m/s$ with an initial robot configuration that places its base at $0.575m$ above the ground. In this

situation, we compare the simplified trajectory to the complete trajectory, taking into consideration the center of mass trajectory, the foot trajectories and the motor controls.

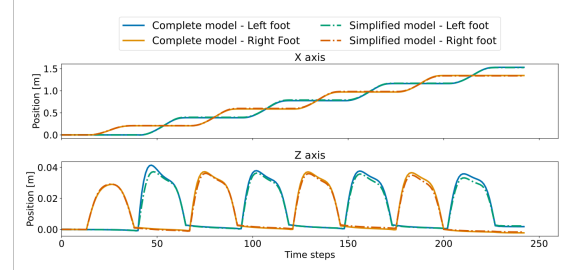


Fig. 4: Feet trajectory for the simplified and complete trajectories during the reference walk motion

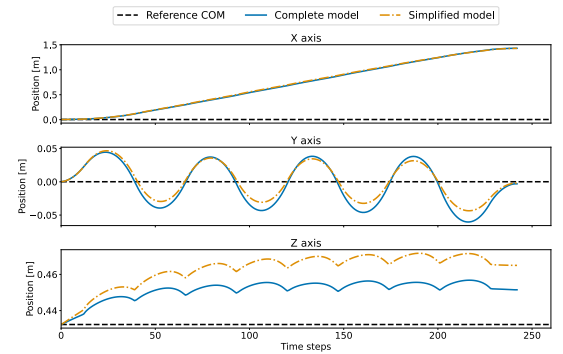


Fig. 5: COM position trajectory for the simplified and complete trajectories during the reference walk motion

Figures 4 and 5 show the comparison of the foot and the COM trajectories for the simplified and complete trajectories. The complex kinematics raises no difference in the foot dynamics. The COM trajectories significantly differ with the simplified model leading to undesirable elevation while the complete modeling account for the knee variable reduction and keep the COM low.

B. Variation of the COM penalization weight

Indeed, the parallel actuation prevents a full leg extension, that the simplified model ignores. The COM elevation must then be controlled with a dedicated cost. Varying this cost leads to a range of robot performance for the simplified model while the complete model is robust to it, as shown in Fig. 6. Consequently, the simplified model may inconsiderately extend the leg leading to peak torque of the knee motor, which we can not observe when only considering the knee joint, as shown in Fig. 7.

C. Variation of the velocity command

Yet, tuning the COM elevation cost is difficult as its importance must be adapted to the situation. We exemplify it by changing velocity command of the OCP from $0.1m/s$ to $0.7m/s$ (keeping the contact pattern unchanged and with constant COM-elevation cost of 500). Figure 8 shows that

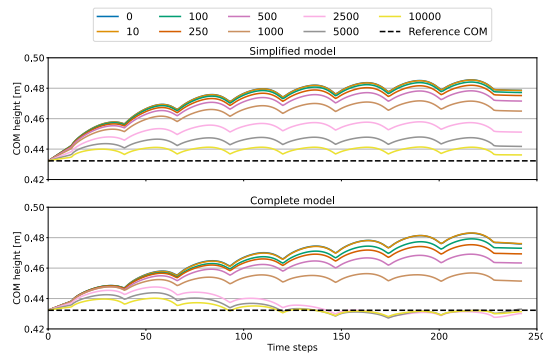


Fig. 6: COM position for the simplified and the complete trajectory at different COM height weights. It is necessary to carefully tune the COM-elevation cost when using the simplified model

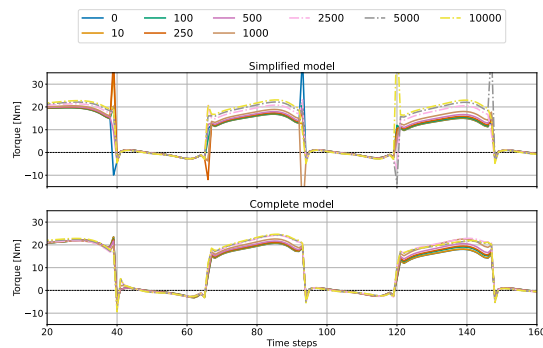


Fig. 7: Knee controls for the simplified and complete trajectory at different COM height weights. The uncontrolled COM elevation with the simplified model leads to peak torque when the knee linkage approaches to singularity, while the complete model enables the OCP to rather take advantage of it

the COM height varies depending on the velocity command for both models. On both models, the robot tends to walk with a lower COM for higher velocities, allowing wider steps. Yet the equilibrium height of the COM is again different for the two models, with an overall higher COM with using the simplified model. In addition of the previous observation, this also reveals that the simplified model does not account for the actual joints limits of the complete model. Then, for higher velocities, the simplified model stretches the leg too much and exceed the limit of feasibility for the knee joint, yielding a trajectory that cannot be transferred to the complete model (this occurs at 0.7 m/s and above for our contact pattern). More generally, the complete model is able to take advantage of the actuator variable reduction due to the parallel linkage, hence leading to reasonable motor effort independently of the walk speed (see Fig. 9).

D. Variation of the steps height

For a stairs climbing task, we observe similar results as before, with the complete model using in a favorable

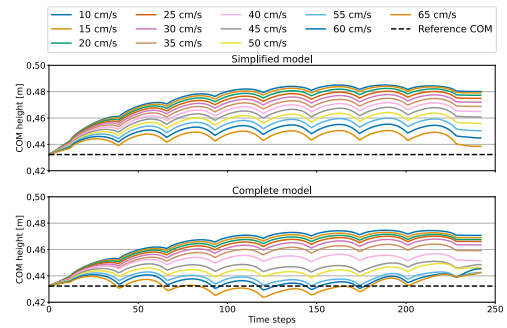


Fig. 8: COM position for the simplified and the complete trajectory at different velocity commands

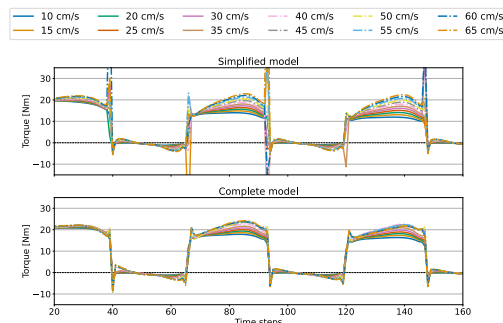


Fig. 9: Knee controls for the simplified and complete trajectory at different velocity commands

way the capabilities of its actuator by working with higher reduction ratio whenever high torques are needed. For higher steps, the simplified model stretches its back leg too much, yielding unfeasible trajectories for the complete model. We do not present figures here as it would be redundant with the previous analysis and refer the reader to the companion video where various other movements are shown that emphasize the capabilities of the OCP using the complete model.

VI. CONCLUSION

This paper presented the derivatives of the contact dynamic of a multi-body system with closed-loop constraints. It allowed us to control robots with closed kinematic chains by considering an underlying serial/tree-like chain and adding bilateral contact constraints to closed the loop. The task is written in the form of an Optimal Control Problem (OCP) that we apply to control a biped in some walking tasks. We also compared optimal trajectories obtained with a simplified model, defined by the underlying serial chain of the robot, to the complete model of the robot. We observed that the simplified model can provide a good approximation of the complete model and can be used to easily create a complete model trajectory by solving several one-step OCPs. However, we also emphasized that it can overlook some aspects of the complete model of the robot, and yield sub-optimal solutions. This occurs especially when the robot uses a wider range of its capabilities, pushing the parallel actuation in non-

linear regions where their reduction ratio varies more. Future work will focus on refining the simplified model to take into consideration the actuators specificities and on generalizing the method to more tasks and robots.

REFERENCES

- [1] T. Mikolajczyk, E. Mikolajewska, H. F. Al-Shuka, T. Malinowski, A. Klodowski, D. Y. Pimenov, T. Paczkowski, F. Hu, K. Giasin, D. Mikolajewski *et al.*, “Recent advances in bipedal walking robots: Review of gait, drive, sensors and control systems,” *Sensors*, vol. 22, no. 12, p. 4440, 2022.
- [2] J.-P. Merlet, *Parallel robots*. Springer Science & Business Media, 2006, vol. 128.
- [3] Virgile Batto, T. Flayols, N. Mansard, and Margot Vulliez, “Comparative Metrics of Advanced Serial/Parallel Biped Design and Characterization of the Main Contemporary Architectures,” *IEEE-RAS International Conference on Humanoid Robots*, 2023.
- [4] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, “Crocodyl: An efficient and versatile framework for multi-contact optimal control,” *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
- [5] G. Liu, Y. Wu, X. Wu, Y. Kuen, and Z. Li, “Analysis and control of redundant parallel manipulators,” *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, pp. 3748–3754, 2001.
- [6] 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,” *IEEE international conference on robotics and automation (ICRA)*, 2003.
- [7] Victor C. Paredes and Ayonga Hereid, “Resolved Motion Control for 3D Underactuated Bipedal Walking using Linear Inverted Pendulum Dynamics and Neural Adaptation,” *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, 2022.
- [8] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” *IEEE-RAS International Conference on Humanoid Robots*, pp. 295–302, 2014.
- [9] J. B. Rawlings, D. Q. Mayne, M. Diehl *et al.*, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.
- [10] E. Dantec, M. Taix, and N. Mansard, “First order approximation of model predictive control solutions for high frequency feedback,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, pp. 4448–4455, 2022.
- [11] E. Chane-Sane, P.-A. Leziart, T. Flayols, O. Stasse, P. Souères, and N. Mansard, “Cat: Constraints as terminations for legged locomotion reinforcement learning,” *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2024.
- [12] Shubham Singh, Ryan P. Russell, and Patrick M. Wensing, “Analytical Second-Order Derivatives of Rigid-Body Contact Dynamics: Application to Multi-Shooting DDP,” *IEEE-RAS International Conference on Humanoid Robots*, 2023.
- [13] J. Carpentier, R. Budhiraja, and N. Mansard, “Proximal and sparse resolution of constrained dynamic equations,” *Robotics: Science and Systems*, 2021.
- [14] Guillermo A. Castillo, Bowen Weng, Wei Zhang, and Ayonga Hereid, “Robust Feedback Motion Policy Design Using Reinforcement Learning on a 3D Digit Bipedal Robot,” *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2021.
- [15] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.
- [16] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiroux, O. Stasse, and N. Mansard, “The Pinocchio C++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives,” *IEEE/SICE International Symposium on System Integration (SII)*, pp. 614–619, 2019.
- [17] M. L. Felis, “RBDL: an efficient rigid-body dynamics library using recursive algorithms,” *Autonomous Robots*, vol. 41, no. 2, pp. 495–511, 2017.
- [18] AgilityProducts. Agility Robotics. Accessed on 2024-09-12. [Online]. Available: <https://agilityrobotics.com/products>
- [19] C. Hubicki, J. Grimes, M. Jones, D. Renjewski, A. Spröwitz, A. Abate, and J. Hurst, “Atrias: Design and validation of a tether-free 3d-capable spring-mass bipedal robot,” *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1497–1521, 2016.
- [20] Fourier Intelligence. Fourier Intelligence. Accessed on 2024-09-12. [Online]. Available: <https://fourierintelligence.com/>
- [21] Humanoid robot G1 — Unitree Robotics. Accessed: 2024-09-12. [Online]. Available: <https://www.unitree.com/g1/>
- [22] Tesla AI & robotics. Tesla. Accessed on 2024-09-12. [Online]. Available: <https://www.tesla.com/AI>
- [23] A. Roig, S. K. Kothakota, N. Miguel, P. Fernbach, E. M. Hoffman, and L. Marchionni, “On the Hardware Design and Control Architecture of the Humanoid Robot Kangaroo,” in *6th Workshop on Legged Robots during the International Conference on Robotics and Automation (ICRA 2022)*, 2022.
- [24] PNDbotics. PNDbotics. Accessed on 2024-09-12. [Online]. Available: <https://www.pndbotics.com/humanoid>
- [25] K. G. Gim, J. Kim, and K. Yamane, “Design and fabrication of a bipedal robot using serial-parallel hybrid leg mechanism,” in *IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [26] S. Redon, A. Kheddar, and S. Coquillart, “Gauss’ least constraints principle and rigid body simulations,” *IEEE international conference on robotics and automation (ICRA)*, 2002.
- [27] D. Baraff, “Fast contact force computation for nonpenetrating rigid bodies,” *International Conference on Computer Graphics and Interactive Techniques*, pp. 23–34, 1994.
- [28] F. E. Udawadia, “Equations of motion for mechanical systems: A unified approach,” *International Journal of Non-linear Mechanics*, vol. 31, no. 6, pp. 951–958, 1996.
- [29] J. Carpentier and N. Mansard, “Analytical derivatives of rigid body dynamics algorithms,” in *Robotics: Science and systems (RSS)*, 2018.
- [30] J. Sola, J. Deray, and D. Atchuthan, “A micro lie theory for state estimation in robotics,” *arXiv:1812.01537*, 2018.
- [31] S. Kleff, J. Carpentier, N. Mansard, and L. Righetti, “On the derivation of the contact dynamics in arbitrary frames: Application to polishing with talos,” *IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, 2022.
- [32] Github - crocodyl - temporary fork. [Online]. Available: <https://github.com/LudovicDeMatteis/crocodyl/tree/topic/contact-6D-closed-loop>
- [33] Locomotion generation for parallel robots: a collection of ocp problems used to generate the examples of the paper - on github. [Online]. Available: <https://github.com/LudovicDeMatteis/ClosedLoopMotion>
- [34] Example parallel robots: a repository containing several urdf models of legged robots with parallel kinematics - on github. [Online]. Available: <https://github.com/Gepetto/example-parallel-robots>