Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot

Ewen Louis Dantec, Maximilien Naveau, Nicolas Mansard, Pierre Fernbach, Nahuel Villa, Guilhem Saurel, Olivier Stasse, Michel Taïx

Whole-Body Model Predictive Control for Biped Locomotion on a Torque-Controlled Humanoid Robot

Ewen Dantec [a,b,*], Maximilien Naveau [a], Pierre Fernbach [c], Nahuel Villa [a], Guilhem Saurel [a], Olivier Stasse [a], Michel Taïx [a], Nicolas Mansard [a,b]

***Abstract*— Locomotion of biped robots requires predictive controllers due to its unstable dynamics and physical limitations of contact forces. A real-time controller designed to perform complex motions while maintaining balance over feet must generate whole-body trajectories, predicting a few seconds in the future with a high enough updating rate to reduce model errors. Due to the huge computational power demanded by such solvers, future trajectories are usually generated using a reduced order model that contains the unstable dynamics. However, this simplification introduces feasibility problems on many edge cases. Considering the permanent improvement of computers and algorithms, whole-body locomotion in real-time is becoming a viable option for humanoids, and this article aims at illustrating this point. We propose a whole-body model predictive control scheme based on differential dynamic programming that takes into account the full dynamics of the system and decides the optimal actuation for the robot's lower body (20 degrees of freedom) along a preview horizon of 1.5 s. Our experimental validation on the torque-controlled robot Talos shows good and promising results for dynamic locomotion at different gaits as well as 10 cm height stairstep crossing.**

I. Introduction

Generalized locomotion for legged robots is a complex problem that requires predicting the future in order to maintain balance over feet while coping with the inertial effects. Model Predictive Control (MPC) schemes have thus become a popular tool to perform locomotion as they consider future trajectory predictions to generate optimal controls. However, deciding optimal torques for every motor of the robot along a preview horizon of up to two steps in the future [1] makes the whole-body locomotion control quite computationally expensive.

Locomotion strategies can roughly be divided into three approaches. The first one aims at decoupling the centroidal part of the dynamics and the whole-body control [2], so that future trajectories are predicted using only the centroidal dynamics to reduce the computational burden; then an instantaneous controller produces the whole-body actuation required to follow the desired centroidal motion. Early works on this approach proposed position control, based on inverse kinematics, to generate the whole-body motion [3], [4], [5]. However, this approach is limited as position control does not account for the inertia of the limbs, which matters for robots with heavy arms and legs (e.g. Talos [6]). Additionally, the compliance, i.e. the ability to smoothly handle impacts and external disturbances, is not intrinsic to the controller [7] and needs to be explicitly designed. On the other hand, torque control allows to directly optimize contact forces and is more suited for heavy robots dealing with multi-contact scenarios. In order to fully exploit the potential of torque control, researchers started to use inverse dynamics as a whole-body instantaneous controller [8], [9], [10] despite the need for a very precise dynamic model, often difficult to estimate [11]. To mitigate this issue, combinations between inverse kinematics and inverse dynamics were also studied [12], [13].

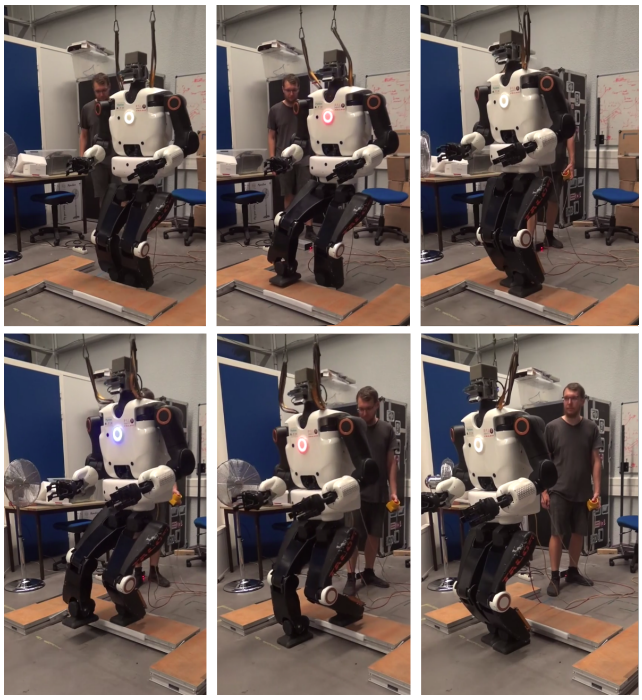

Fig. 1. Snapshots of the stairstep crossing experiment.

Another approach involves decoupling the centroidal dynamics over the end of the preview horizon while only using the whole-body dynamics for the first time-steps [14], [15]. This has the advantage to propagate the terminal cost or constraint of the dynamics into the whole-body motion, while keeping low the overall computational load.

Recent efficient solvers [16], [17] allow for a third approach which consists in a complete whole-body MPC. Particular implementations involve solving iteratively the centroidal dynamics and whole-body kinematics over a preview horizon until both solutions are consistent [18], [19],

This work is supported by the European project MEMMO (GA-780684) and ANITI (ANR-19-P3IA-0004).

[a] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[b] Artificial and Natural Intelligence Toulouse Institute, France
[c] TOWARD, Toulouse, France
[*] corresponding author: edantec@laas.fr

[20]. The other possibility is to directly optimize the full motion [16] through a direct shooting approach [21], either by using iterative Linear Quadratic Regulator (iLQR) [22], [23] methods or Differential Dynamics Programming (DDP) [24], [25]. These algorithms feature a linear complexity in the time horizon, which make them especially efficient for large horizon and complex models; moreover, they produce a reliable state-feedback policy at no extra cost through the Riccati gains [25], [26].

In this work, a DDP-based MPC scheme taking into account the whole-body model of the robot is used to achieve dynamic locomotion and stairstep crossing. Given the current state of the robot measured at 2 kHz, the control block directly computes the optimal torque and feedback policy to be played on the robot, without any other refinement or estimation needed. The method is fully generic, open-source and can be applied to any legged robots on various terrains. It corresponds to the first demonstration of the capabilities of whole-body MPC for the locomotion of a torque-controlled humanoid robot on flat and non-flat terrains.

The optimization problem is presented in Sec. II, while the formulation of the resulting MPC is given in Sec. III. We demonstrate the validity of the framework on the torque-controlled humanoid robot Talos [27] through a series of locomotion experiments, presented in Sec. IV.

II. Whole Body Trajectory Optimization Problem

A. Whole-body dynamics modelling

We consider a floating-base rigid-body system with n_j joints, in contact with the environment at n_p places. The configuration space of such a system can be described by a vector $\boldsymbol{q} \in \mathrm{SE}(3) \times \mathbb{R}^{n_j}$ composed by its global position, orientation and the angle of each joint. When we command torques $\boldsymbol{\tau} \in \mathbb{R}^{n_j}$ on the joint motors, the robot configuration evolves as [28]:

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{J}_c^\top \\ \boldsymbol{J}_c & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ -\boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}^\top \boldsymbol{\tau} - \boldsymbol{b} \\ -\dot{\boldsymbol{J}}_c \dot{\boldsymbol{q}} \end{bmatrix}, \tag{1}$$

where $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{b}$ the generalized Coriolis, centripetal and gravity effects, $\boldsymbol{S}$ a selection matrix mapping over the actuated joints, $\boldsymbol{J}_c = (\boldsymbol{J}_1 \cdots \boldsymbol{J}_{n_p})$ the concatenation of contact Jacobian matrices and $\boldsymbol{\lambda} = (\boldsymbol{\lambda}_1 \cdots \boldsymbol{\lambda}_{n_p}) \in \mathbb{R}^{6 \times n_p}$ represents all contact wrenches.

B. Problem formulation

We choose the state to contain the robot configuration and its rate of change $\boldsymbol{x} = (\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\dot{\boldsymbol{x}} = (\dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})$, and we use the vector of joint torques as our control signal $\boldsymbol{u} = \boldsymbol{\tau}$.

The control problem is formulated in discrete time by rewriting the dynamics (1) as a constraint in our next-state integration scheme according to [29]. At time t, we have

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t). \tag{2}$$

We then obtain the optimal state and control trajectories along a preview horizon of T knots by solving the control problem:

$$\begin{aligned} \min_{\underline{\boldsymbol{x}}, \underline{\boldsymbol{u}}} \sum_{t=0}^{T-1} & \ell(\boldsymbol{x}_t, \boldsymbol{u}_t, t) + \ell_T(\boldsymbol{x}_T) \\ s.t. \quad & \boldsymbol{x}_0 = \boldsymbol{f}_0 \\ \forall t \in [0 \,..\, T-1], \quad & \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t), \end{aligned} \tag{3}$$

where $\underline{\boldsymbol{x}} = (\boldsymbol{x}_t)_{t=0..T}$ and $\underline{\boldsymbol{u}} = (\boldsymbol{u}_t)_{t=0..T-1}$ are the state and control trajectories, ℓ and ℓ_T are the running and terminal cost functions which define the locomotion task, and $\boldsymbol{f}_0$ is the initial state. In order to be able to solve this problem in real time, we do not consider any other state and control constraints, but instead choose to use penalty costs.

C. Solving the OCP with DDP

DDP is a very efficient tool to solve (3) based on Bellman's optimality principle [30]. The algorithm leverages the sparsity of the Markovian nature of the dynamics constraints and provides a descent step $(\boldsymbol{\Delta x}_t, \boldsymbol{\Delta u}_t)_{t=0..T}$ through a backward-forward pass formulation followed by a line search heuristics. Moreover, it naturally computes the optimal state feedback gains $\boldsymbol{K}_t$, also known as Riccati gains, such that $\boldsymbol{\Delta u}_t = \boldsymbol{K}_t \boldsymbol{\Delta x}_t + \boldsymbol{k}_t$. We demonstrated before that these gains can be used to produce a first order approximation of the optimal torque at the frequency of the low-level control [26]. In classical DDP scheme [31], [32], the second-order term of the Taylor approximation of the quality function is taken into account; here we neglect this term to lower the computational load of the backward pass, making us able to solve the OCP in real time. Such an approach is classically called iLQR [33]. Note that this approximation does not affect the computation of the optimal feedback gains.

D. Formulation of the costs

The walking task is described in our OCP by a running cost function ℓ consisting of 6 terms, and a terminal cost function ℓ_T reduced to only 3 terms (**(1)**, **(3)** and **(4)**):

(1) State regularization:
$\ell_1(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_d)^T \boldsymbol{R}_x (\boldsymbol{x} - \boldsymbol{x}_d)$ with $\boldsymbol{R}_x$ a positive definite weight matrix and $\boldsymbol{x}_d$ the initial state of the robot, usually half-sitting position with null velocity. This cost prevents the optimized state trajectory from being too far away from the default position of the robot.

(2) Control regularization:
$\ell_2(\boldsymbol{u}) = (\boldsymbol{u} - \boldsymbol{u_d})^T \boldsymbol{R}_u (\boldsymbol{u} - \boldsymbol{u_d})$ with $\boldsymbol{R}_u$ a positive definite weight matrix and $\boldsymbol{u}_d$ the gravity-compensating torque in default state. This cost prevents the solver to cross torque limits, although it does only act as a penalization.

(3) Feet-tracking:
$\ell_3(\boldsymbol{x}, t) = a(\boldsymbol{p}(\boldsymbol{x}) - \boldsymbol{p}_d(t))$ with $a : \boldsymbol{r} \mapsto log(1 + \frac{||\boldsymbol{r}||}{\alpha})$ a logarithmic activation function, $\boldsymbol{p}(\boldsymbol{x})$ and $\boldsymbol{p}_d(t)$ current and desired foot placement in $SE(3)$, and $\alpha = 0.2$. This cost, associated with a user-tuned desired foot trajectory $\boldsymbol{p}_d(t)_{t=0..T}$, ensures that the foot in swing phase goes to the next pre-defined contact. It stays inactive for the

supporting foot during swing phases, but is active during double support phases.

(4) Kinematic limit:
$\ell_4(\boldsymbol{x}) = \frac{1}{2}||\max(\boldsymbol{x} - \boldsymbol{x}_u, \boldsymbol{0})||^2 + \frac{1}{2}||\min(\boldsymbol{x} - \boldsymbol{x}_l, \boldsymbol{0})||^2$ with $\boldsymbol{x}_u$ the upper bound and $\boldsymbol{x}_l$ the lower bound of the joint positions and velocities. This cost prevents the solver from crossing the kinematics bounds of the robot. Although the kinematics constraint takes the form of a penalization, it efficiently rejects unfeasible trajectories when the associated weight is high.

The two last costs terms are **(5) wrench tracking** and **(6) local Center of Pressure (CoP) regularization** costs. They ensure that the robot weight efficiently switches from one contact to the next, and that the contact forces remain dynamically consistent during the motion. In the following, we only consider contacts between rigid bodies. A foot in contact with the ground exerts a wrench $\boldsymbol{\lambda} = (\boldsymbol{f}, \boldsymbol{\tau}) = (f_x, f_y, f_z, \tau_x, \tau_y, \tau_z)$ composed by linear $\boldsymbol{f}$ and angular $\boldsymbol{\tau}$ parts. Let us denote L, W the half-length and half-width of the foot, and μ a friction coefficient. Ideally, we want $\boldsymbol{\lambda}$ to remain inside the wrench cone [34], [35] defined by the following inequalities:

$$f_z > 0 \tag{4a}$$
$$|f_x| \leq \mu f_z \tag{4b}$$
$$|f_y| \leq \mu f_z \tag{4c}$$
$$|\tau_x| \leq W f_z \tag{4d}$$
$$|\tau_y| \leq L f_z \tag{4e}$$
$$\tau_{min} \leq \tau_z \leq \tau_{max}, \tag{4f}$$

with

$$\tau_{min} = -\mu(W+L)f_z + |W f_x - \mu\tau_x| + |L f_y - \mu\tau_y|$$
$$\tau_{max} = \mu(W+L)f_z - |W f_x + \mu\tau_x| - |L f_y + \mu\tau_y|.$$

These inequalities describe a linearized wrench cone with 4 facets. While one can opt for a better approximation of the true wrench cone by adding more facets, we chose to use a simple model for the sake of time computation. As already mentioned, DDP does not allow yet to take into account such inequality constraints, although some recent works may solve this issue in the near future [36], [37]. Using quadratic barrier costs to approximate those constraints makes the problem too hard to solve in real time, as the solver struggles to discover the optimal contact forces in just one iteration. For this reason, we introduce a reference wrench trajectory $\boldsymbol{\lambda}_d(t) = (0, 0, f_d(t), 0, 0, 0)$ where $f_d(t)$ is a normal force reference equal to $M_r g$, the weight of the robot during single support phase, and switching continuously from $M_r g - f_{min}$ to f_{min} during double support phase. Here, $f_{min} = 150$ N is a security margin and should be read as the minimum force reference at contact just before take-off. This margin is implemented so that cost **(5)** doesn't become ill-conditioned when f_d tends toward 0.

The contact wrench constraints are summarized in a matrix $\boldsymbol{A}$ such that (4) is equivalent to $\boldsymbol{b}_l \leq \boldsymbol{A}\boldsymbol{\lambda} \leq \boldsymbol{b}_u$, with $\boldsymbol{b}_l$ and $\boldsymbol{b}_u$ the corresponding lower and upper bounds. The final wrench tracking cost is then defined by:

$$\ell_5(\boldsymbol{x}, \boldsymbol{u}, t) = ||\boldsymbol{A}(\boldsymbol{\lambda} - \boldsymbol{\lambda}_d(t))||^2, \tag{5}$$

with every wrench expressed in the local frame of the contact foot. One alternative formulation of the cost consists in replacing the matrix $\boldsymbol{A}$ with a diagonal weight matrix to regularize individually each component of the wrench. This alternative has been tested on the robot but has not yet resulted in stable motions.

The wrench tracking cost alone does not prevent occasional breaking of the non-tipping constraints during contact transition. Practically speaking, it has been observed in simulation and on the robot that the local CoP constraints described by inequalities 4d and 4e are violated during contact switches, although the wrench tracking cost **(5)** should regularize the contact torque to zero. To address this behavior, a local CoP regularization cost has been added to the formulation:

$$\ell_6(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{2W^2}\left(|\frac{\tau_y}{f_z}|^2 + |\frac{\tau_x}{f_z}|^2\right). \tag{6}$$

The idea behind this cost is to force the inequalities $|\frac{\tau_y}{Lf_z}| < |\frac{\tau_y}{Wf_z}| \leq 1$ and $|\frac{\tau_x}{Wf_z}| \leq 1$ by penalizing the corresponding quantities. In order to be more conservative, W can be set smaller than the half-width of the foot, which is equivalent to increasing the CoP cost weight.

The effect of this cost is illustrated in Fig. 2, where a walking motion was performed in simulation with and without the CoP regularization.

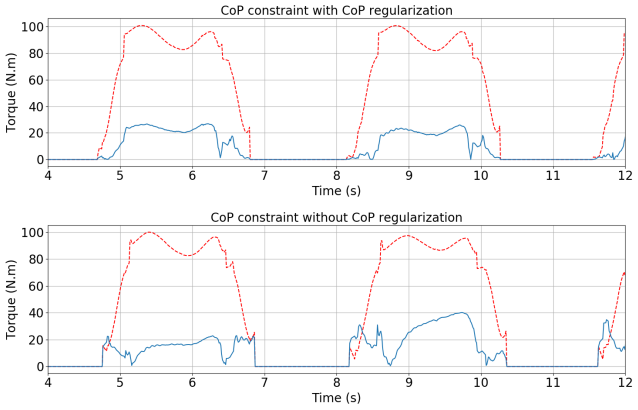

Fig. 2. Left foot CoP constraint $|\tau_y| \leq L f_z$ for simulated walk with and without CoP regularization. Blue plain line is $|\tau_y|$, red dotted line is $L f_z$. Bottom plot shows violations of the constraint during contact switch, which translate into oscillating ankles in simulation. On the real robot, those constraints violations lead to falling.

III. Model Predictive Control Scheme

A. Timing and model

The complexity of the DDP increases as the cube of the state dimension. In order to cut the computation load as much as possible, it is interesting to use a reduced robot model including a total of 14 torque-controlled joints: 6 for each leg and 2 for the torso. The arm joints are controlled in

position and remain fixed during the walking motion. From the viewpoint of the high-level MPC, the arm joints do not exist. It has been demonstrated that including the arms in the model allows for a better control of the centroidal angular momentum of the robot [12]; using only a half-body model is thus penalizing to the walking motion. Nevertheless, our scheme is able to handle this issue. As for the contact phases and timings, they are decided offline by the user, which could be replaced in the future by a high-level pattern generator in order to perform step adaptation. The DDP problem is formulated using the open-source Crocoddyl library [17], which is based on Pinocchio [38], a state-of-the-art library of rigid body algorithms. Although the derivatives evaluation is parallelized, the backward pass is not particularly optimized and we observed computational burden (likely in the cache flow) that we don't understand properly yet. It is likely that optimizing our implementation could lead to 30-50% discount to the current version.

The horizon of our OCP has 150 knots separated by a time-step of 10 ms, allowing the solver to predict the behavior of the robot 1.5 s into the future. Before starting the motion, the OCP is first solved until convergence with a horizon only composed of double support models; then, at each control cycle, the previous optimal trajectory is used as a warm-start to bootstrap the problem and one DDP iteration is performed. As the time goes by, contact switches and desired wrench trajectory are gradually inserted at the end of the horizon. This scheme allows us to solve the whole-body problem online in about 15 ms.

B. Reference foot trajectory

During double support phase, at each control cycle, the desired swing foot trajectories are updated over the entire horizon using a minimum jerk Bezier curve, as in [39]. The final placement of the swing foot is set to be the placement of the support foot, plus a desired 2D translation $(\delta x, \delta y)$. These curves are tuned so that during landing and take-off, foot velocity and acceleration are colinear to the normal of the contact surface; as a consequence, the entire foot area is making contact with the ground at the same time.

C. Cost parametrization

The following weight distribution has been used on the robot to make it walk.

(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
1	0.005	1000	1000	0.001	10

The weight matrix $\boldsymbol{R}_u$ is the identity matrix, while the weight matrix $\boldsymbol{R}_x$ has been carefully tuned to penalize critical behaviors like tilting torso or swinging base. Here are the weights used on the robot.

	Base pose	Base angle	Leg	Torso
Position	0	10000	10	100
Velocity	10	10	10	10

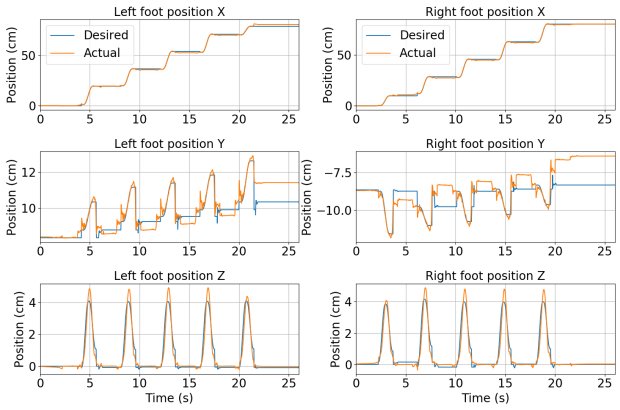

Fig. 3. Measured and desired feet position during locomotion on flat floor.

D. Robotic Operating System (ROS) architecture

Our MPC is embedded inside a ROS node which subscribes to the actual state of the robot measured at 2 kHz, and which publishes a feedforward optimal torque $\boldsymbol{u}_0$, Riccati gains $\boldsymbol{K}_0$ and last computed initial state $\boldsymbol{x}_0$ at 70 Hz. The final torque sent to the low-level control combines the feedforward control with a state feedback based on the Riccati gains and last measured state $\boldsymbol{x}_m$ (see [26] for more details):

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{K}_0(\boldsymbol{x}_0 - \boldsymbol{x}_m). \tag{7}$$

At last, the desired intensity of current at each controlled joint motor is computed through a proportional-derivative feedback on the joint torque measurement. Because the actuator dynamics (motor inertia, frictions, flexibility...) are not considered in the whole-body model, a feedforward term on the current is added to compensate for the model discrepancy. From the point of view of the high-level MPC, every joint is behaving as an ideal joint. The architecture is separated between two parallel processes running on different CPUs. This publisher-subscriber architecture has been successfully tested in previous works [26], [40].

IV. Experimental results

We performed experiments with the humanoid robot Talos [27] to test our whole-body MPC framework. Due to the critical problem of computation time, the MPC runs on a powerful external computer (AMD Ryzen 5950X, 16 cores and 4.9GHz with 64 GB of RAM), whereas the low-level control runs on the embedded CPU of the robot.

The proposed control scheme has been evaluated in two locomotion scenarios: in the first one, the robot performed a straight walk on flat floor with two sets of gait timings; in the second one, the robot successfully climbed up and down a 10 cm-high step. In both experiments, the exact same cost weights have been used on the robot, and the only varying parameters were the feet trajectory and gait timings. This demonstrates the adaptability of the MPC framework, which can produce relevant stable trajectories over a wide range of dynamic motions. The videos of the experiments, as well as the associated raw data, are available at

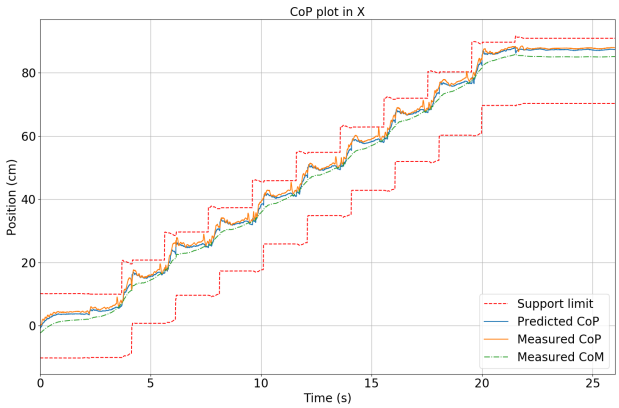

Fig. 4. CoP trajectory in X axis compared to actual feet position during locomotion on flat floor. The foot of the robot is 20 cm long.

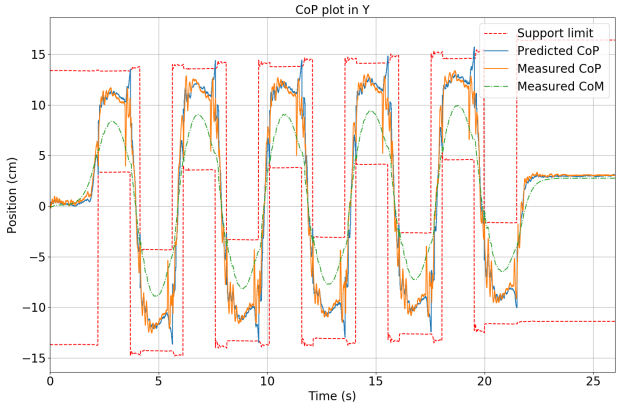

Fig. 5. CoP trajectory in Y axis compared to actual feet position during locomotion on flat floor. The foot of the robot is 10 cm wide.

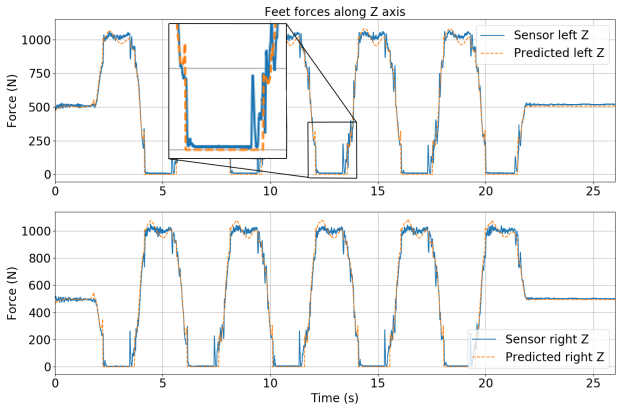

Fig. 6. Predicted vs measured normal forces in left and right feet during locomotion on flat floor. The foot is touching the ground before landing as can be seen in the zoomed window.

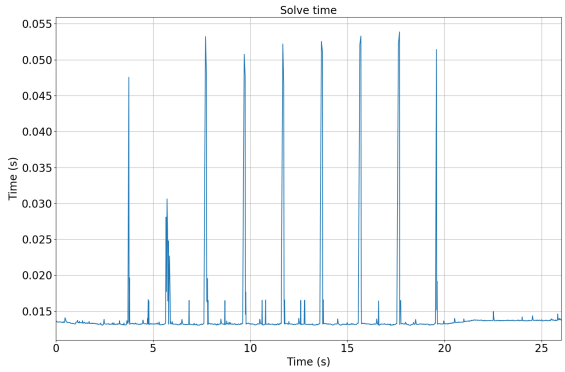

Fig. 7. Time computation of one DDP iteration during locomotion on flat floor.

https://gepettoweb.laas.fr/articles/dantechumanoid22.html. The following table presents the different gait parameters used during experimental validation.

	Gait 1	Gait 2	Stairstep
Forward step length δx	10 cm	20 cm	30 cm
Width btw. feet δy	20 cm	20 cm	20 cm
Single support time	1.1 s	1.5 s	3 s
Double support time	0.3 s	0.75 s	2 s

A. *Walking on flat floor*

For the sake of conciseness, only the first gait tested on the robot will be discussed in this paper. Both gaits are nonetheless displayed in the attached video.

The results of the walking experiment are presented in figures 3 to 7. Fig. 3 shows the tracking of the desired feet trajectories over time. The Bezier curves used to generate the reference in position are only followed during swing phases, hence are not relevant during support phases. Although the reference in X-axis is precisely tracked, the actual foot roll and position in Y-axis suffer from noisy oscillations at take-off and landing. Empirically, we found out that the walking motion improves when the contact switch happens while the foot reference is 1 cm above ground. Doing this makes the robot force the contact into the ground with non-zero forces, creating a stable contact and an impact. Because the control scheme is intrinsically compliant, it damps the impact and limits the resulting forces, preventing hardware damage.

Additionally, Fig. 6 shows that a normal force is measured just before the contact switch, which means that the foot impacted the ground before the solver predicted it. As no step or timing adaptation has been introduced in our framework yet, such behavior is for now unavoidable; nevertheless, the MPC is able to cope with such unknown disturbances and produce a stable walking trajectory.

Figures 4 and 5 shows that the predicted CoP, computed from the forces predicted by the solver, matches the CoP based on the sensor's forces. During the motion, the solver defines its own CoP trajectory to minimize the user-tuned cost function. It is interesting to notice that when the foot lands, the predicted CoP in Y axis may sometimes go outside the feet support area, although the real CoP remains inside. In this case, the low-level torque feedback loop is acting as a low-pass filter which smooths the control's sharp edges.

Finally, Fig. 7 indicates that the MPC is computed in less than 15ms, but features brutal peaks of 50 ms which happen around the instants of contact switch, where the structure

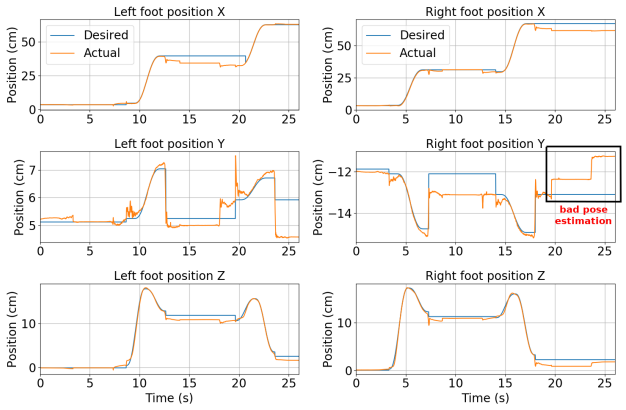

Fig. 8. Measured and desired feet position during stairstep crossing. Bad pose estimation is highlighted at the end of the right foot trajectory.

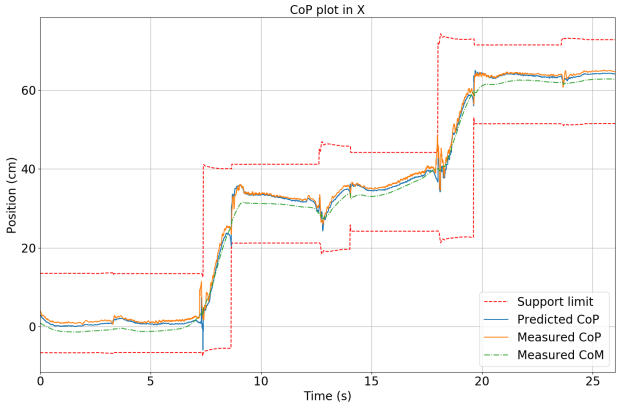

Fig. 9. CoP trajectory in X axis compared to actual feet position during stairstep crossing. The foot of the robot is 20 cm long.

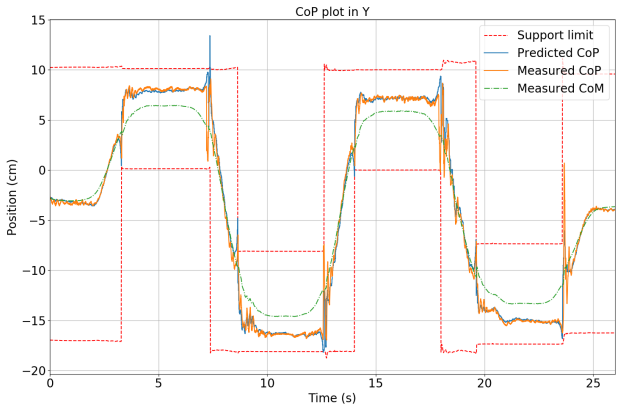

Fig. 10. CoP trajectory in Y axis compared to actual feet position during stairstep crossing. The foot of the robot is 10 cm wide.

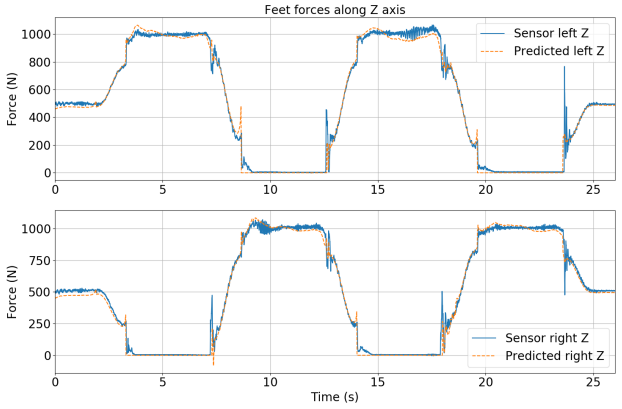

Fig. 11. Predicted vs measured normal forces in left and right feet during stairstep crossing.

of the problem suddenly changes. Despite these unexpected time overflows, the motions produced by our framework remain stable because the approximated Riccati policy takes over while the high-level control is computing.

B. Stair up and down

The main objective of the stair experiment is to demonstrate the versatility of the proposed framework. Compared to the walking experiment, only the feet trajectories and contact timings have been modified, while the wrench trajectory remains the same. Figures 8 to 11 illustrate the results of this experiment.

Figure 8 highlights an essential issue caused by a bad estimation of the base of the robot: during the last 5 seconds of the motion, whereas the right foot is in support phase and not moving, the actual foot position in Y shows brutal discontinuities of 1 cm when the left foot is creating and breaking contact. The estimated base of the robot shows the exact same discontinuities at the same time. Fortunately, the solver is robust enough to handle such disturbances.

Similarly to the flat floor experiment, the predicted CoP trajectory briefly goes outside the foot area when landing occurs, as seen in Fig. 10. This issue is also noticeable in Fig. 11 where the lateral forces during contact switches feature several peaks corresponding to contact mismatches caused by blind time-scheduled transitions and model uncertainties.

V. Conclusion

This paper shows the first application of whole-body MPC for the locomotion of an industrial biped robot. The proposed framework can be adapted to cross obstacles like stairstep by simply adjusting the feet reference trajectories and locomotion gaits. It has proven to be robust against model uncertainties, imprecise state estimation and unexpected latency in data transmission. The key to implementing this MPC on a real platform lies in the approximation of the wrench cone constraints treated as regularization costs and the use of Riccati-based feedback policy inside the low-level control of the robot. The latter improvement allows us to increase the frequency of the control scheme to 2 kHz. Our single-block framework can be extended to any given task at the (small) price of designing and tuning specific costs adapted to the goal.

Although this whole-body MPC architecture produces promising results, we still need to prove that it can out-

perform state-of-the-art centroidal walking controllers. The proposed method used to cope with wrench cone constraints results in dynamically unfeasible predicted CoP trajectories, even if the filtering effect of the low-level control allows for such solutions to be executed on the robot. The behavior of the robot at contact transitions needs to be studied more thoroughly, so as to eliminate those brutal discontinuities in torques and forces. Our main perspective of work is to get rid of the feet reference trajectories in order to make the solver able to perform push recovery. Following this logic, we plan to introduce step adaptation and online optimization of the contact timings, with the goal of increasing control smoothness, walk speed and step length.

REFERENCES

[1] P. Zaytsev, S. J. Hasaneini, and A. Ruina, "Two steps is enough: no need to plan far ahead for walking balance," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] 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.

[3] G. Romualdi, S. Dafarra, Y. Hu, and D. Pucci, "A benchmarking of dcm based architectures for position and velocity controlled walking of humanoid robots," *IEEE-RAS International Conference on Humanoid Robots*, 2019.

[4] G. Romualdi, S. Dafarra, G. L'Erario, I. Sorrentino, S. Traversaro, and D. Pucci, "Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment," *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[5] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, 2017.

[6] O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. del Prete, P. Souères, N. Mansard, F. Lamiraux, J.-P. Laumond, L. Marchionni, H. Tome, and F. Ferro, "Talos: A new humanoid research platform targeted for industrial applications," *IEEE-RAS Humanoids*, 2017.

[7] N. Ramuzat, G. Buondonno, S. Boria, and O. Stasse, "Comparison of position and torque whole body control schemes on the humanoid robot talos," *20th International Conference on Advanced Robotics (ICAR)*, 2021.

[8] S. Wang, G. Mesesan, J. Englsberger, D. Lee, and C. Ott, "Online virtual repellent point adaptation for biped walking using iterative learning control," *IEEE-RAS Int. Conf. Humanoid Robots*, 2021.

[9] R. Schuller, G. Mesesan, J. Englsberger, J. Lee, and C. Ott, "Online learning of centroidal angular momentum towards enhancing dcm-based locomotion," *IEEE International Conference on Robotics and Automation*, 2022.

[10] M. Hutter, M. A. Hoepflinger, C. Gehring, M. Bloesch, C. D. Remy, and R. Siegwart, "Hybrid operational space control for compliant legged systems," *Robotics:Science and Systems (R:SS)*, 2012.

[11] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, p. 473–491, 2014.

[12] C. Khazoom and S. Kim, "Humanoid arm motion planning for improved disturbance recovery using model hierarchy predictive control," *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[13] T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englesberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, pp. 1–34, 2016.

[14] A. Sherikov, D. Dimitrov, and P.-B. Wieber, "Whole body motion controller with long-term balance constraints," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, p. 444–450, 2014.

[15] H. Li, R. J. Frei, and P. M. Wensing, "Model hierarchy predictive control of robotic systems," *IEEE Robotics and Automation Letters*, vol. 6, 2021.

[16] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, 2018.

[17] C. Mastalli, R. Budhiraja *et al.*, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[18] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimization for kino-dynamic motion generation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 2703–2710, 2016.

[19] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *ArXiv*, vol. abs/2201.07601, 2022.

[20] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," *IEEE-RAS International Conference on Humanoid Robots*, 2015.

[21] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[22] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," *IEEE American Control Conference*, 2005.

[23] S. Mason, N. Rotella, S. Schaal, and L. Righetti, "Balancing and walking using full dynamics lqr control with contact constraints," *IEEE-RAS Humanoids*, 2016.

[24] Y. Tassa, T. Erez, and W. D. Smart, "Receding horizon differential dynamic programming," *NeurIPS*, 2008.

[25] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," *IEEE/RSJ IROS*, 2019.

[26] E. Dantec, M. Taix, and N. Mansard, "First order approximation of model predictive control solutions for high frequency feedback," *IEEE Robotics and Automation Letters*, vol. 7, 2022.

[27] O. Stasse, T. Flayols *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," *IEEE-RAS Humanoids*, 2017.

[28] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and sparse resolution of constrained dynamic equations," in *R:SS*, 2021.

[29] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "DDP for multi-phase rigid contact dynamics," *IEEE-RAS Humanoids*, 2018.

[30] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[31] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[32] D. Q. Mayne, "Differential dynamic programming–a unified approach to the optimization of dynamic systems," *Control and Dynamics Systems*, vol. 10, pp. 179–254, 1973.

[33] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems," *Int.Conf. on Informatics in Control, Automation and Robotics*, 2004.

[34] S. Caron, Q.-C. Pham, and Y. Nakamura, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[35] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, and M. Morisawa, "A universal stability criterion of the foot contact of legged robots - adios zmp," *IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[36] W. Jallet, A. Bambade, N. Mansard, and J. Carpentier, "Proxnlp: a primal-dual augmented lagrangian solver for nonlinear programming in robotics and beyond," *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[37] T. A. Howell, B. E. Jackson, and Z. Manchester, "Altro: A fast solver for constrained trajectory optimization," *IEEE/RSJ IROS*, 2019.

[38] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, 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 Int. Symp. on System Integration*, 2019.

[39] S. Tonneau, P. Fernbach, J. Chemin, and G. Saurel, "ndcurves," 2013. [Online]. Available: https://github.com/loco-3d/ndcurves

[40] E. Dantec, R. Budhiraja, A. Roig, T. Lembono, G. Saurel, O. Stasse, P. Fernbach, S. Tonneau, S. Vijayakumar, S. Calinon, M. Taix, and N. Mansard, "Whole body model predictive control with a memory of motion: Experiments on a torque-controlled talos," *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.