Comparison of predictive controllers for locomotion and balance recovery of quadruped robots

Thomas Corbères, Thomas Flayols, Pierre-Alexandre Léziart, Rohan Budhiraja, Nicolas Mansard

Comparison of predictive controllers for locomotion and balance recovery of quadruped robots

Thomas Corbères [a], Thomas Flayols [a], Pierre-Alexandre Léziart [a], Rohan Budhiraja [a], Nicolas Mansard [a,b]

***Abstract*— As locomotion decisions must be taken by considering the future, most existing quadruped controllers are based on a model predictive controller (MPC) with a reduced model of the dynamics to generate the motion, followed by a second whole-body controller to follow the movement. Yet the choice of the considered reduction in the MPC is often ad-hoc or decided by intuition. In this article, we focus on particular MPCs and analyze the effect of the reduced models on the robot behavior. Based on existing formulations, we offer additional controllers to better understand the influence of the reductions in the controller capabilities. Finally, we propose a robust predictive controller capable of optimizing the foot placements, gait period, center-of-mass trajectory and corresponding ground reaction forces. The behavior of these controllers is statistically evaluated in simulation. This empirical study is a basis for understanding the relative importance of the components of the optimal control problem (variables, costs, dynamics), that are sometimes arbitrarily emphasized or neglected. We also provide a qualitative study in simulation and on the real robot Solo.**

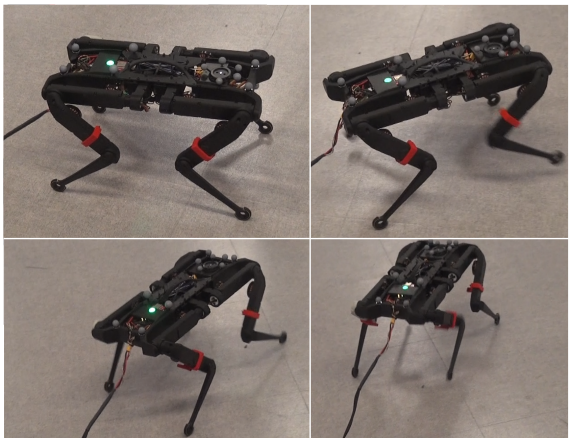

Fig. 1. Solo [9] performing trotting during the experiments. Motions from different controllers were tried on Solo. While all controllers were able to perform these simple motions, we provide comparison of their performances for more extreme cases.

I. Introduction

Locomotion with quadruped robots is challenging as well as rewarding for roboticists. With lighter feet than bipeds and more stability, quadrupeds can be used for highly dynamic motions and gaits, as already tried on the Cheetah [1], Anymal [2], HyQ [3] etc.

As a result, controllers that can generate real-time motion trajectories for online robot control are a hot research domain. Among these, non-linear model predictive controllers (MPC) [4], [5] can provide real-time computations [3], reactive control [6], [7], and because of optimal control over a time horizon, they can take care of disturbances and perturbations to the system [8].

Whole-body dynamics allows maximum utilization of our knowledge about the robot model [10]. However, the non-linearity introduced by the dynamics, and the high dimensions are prohibitive with respect to the computation times. The classical consequence is to use reduced dynamics of lower dimensions instead [11]–[13], very often with additional assumptions to simplify further the non-linearities of the problem. In such cases, the plan provided by the reduced model is followed by the low-level controller using inverse dynamics, e.g. in [6], [14]. Several reduced models can be considered to build a MPC for quadruped locomotion: table-cart [11] (or with foothold optimization [14], [15]), centroidal [16] (or with contact timings optimization [17]). For quadruped robots, a sound reduction is to approximate the angular momentum to the rotation of the rigid body, neglecting the limb dynamics [18]–[20], which we chose in this work [13].

The simplification of dynamics is mostly the result of limited resources available for online real-time MPC. In this paper, we propose to simultaneously evaluate the interest and drawbacks implied by these simplifications. After recalling the main concepts in Sec. II, we start by introducing the general optimal control problem (OCP) formulation : based on the convex problem detailed in [13], a first OCP is presented in Sec. III to clearly pose the optimization problem and serve as a point of comparison. Three other optimal problems are then proposed, by successively removing some linearization assumptions done in [13] (Sec. IV), extend the formulation to footholds optimization (Sec. V) and include the period of the walk as parameter of optimization (Sec VI). All OCPs are presented following the same template in order to better exhibit their similarities and particularities. As the two last OCP implies the simultaneous optimization of trajectories and static parameters (foothold positions and timings), we explain in Section VII how a differential dynamic programming (DDP) solver can be used to solve a parametric OCP. The MPC implementing the OCP to perform the feedback control of the robot is detailed in Sec. VIII, which is used to benchmark in Sec. IX the several OCP formulations in simulation and on the robot Solo.

[a] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
[b] Artificial and Natural Intelligence Toulouse Institute, France

This work has been supported by the MEMMO European Union project within the H2020 Program under Grant Agreement No. 780684.

II. Optimal Control with Reduced Dynamics

The centroidal dynamics [21] describes exactly the dynamics of the center of mass (CoM) of the robot due to its interactions with the environment and corresponds to the under-actuated dynamics [22]. Since most quadruped robots are built with lightweight limbs, it is classically preferred to use the centroidal dynamics to approximate the whole-body dynamics, as described below:

$$m\ddot{p} = \sum_{i=1}^{n_c} f_i + mg \tag{1a}$$

$$\mathcal{I}\dot{\omega} + \omega \times (\mathcal{I}\omega) = \sum_{i=1}^{n_c} (r_i - p) \times f_i \tag{1b}$$

where $p = (x, y, z)$ is the position of the body, ω is the angular velocity of the body, m and $\mathcal{I}$ are mass and inertia of the body, and $g = (0, 0, -9.8)$ is the gravity vector. n_c 3D forces f_i are applied at the contact points r_i. Consequently, the robot state is composed by linear and angular positions and velocities, belonging to $\mathbb{R}^{12}$:

$$x = \begin{bmatrix} p & \Theta & \dot{p} & \omega \end{bmatrix}^T. \tag{2}$$

where $\Theta = (\theta, \phi, \psi)$ is the rotation of the body frame $\mathcal{R}_l$ with regards to the world frame $\mathcal{R}_0$. To ease the reading of the following, we make now the choice of representing the rotation Θ by the 3 Euler angles. The rotation matrix from body to world is then : $R_{zyx} = R_z(\psi)R_y(\phi)R_x(\theta)$.

In the next section, we will follow [13] and suppose small body inclination ϕ and θ. Consequently, i) the roll and pitch angles in the rotational matrices can be neglected : $R_z(\psi)R_y(\phi)R_x(\theta) \approx R_z(\psi)$. ii) $\omega = \dot{\Theta}$. iii) The representation of inertia matrix in world frame is simplified : $\mathcal{I} \approx R_z(\psi)\mathcal{I}_l R_z(\psi)^T$ [23], and iv) the cross-product term $\omega \times (\mathcal{I}\omega)$ is neglected leading to a linear variation of the angular momentum :

$$\frac{d}{dt}(\mathcal{I}\omega) \approx \mathcal{I}\dot{\omega} \tag{3}$$

where $\mathcal{I}$ is approximately constant in the world frame. We can now set up the generic form of the optimal control problem for locomotion. We formulate the optimization over a sequence of contacts phases s. Each phase is defined by its duration T_s and contact location r_s :

$$\min_{\{x\},\{u\}} \sum_{s \in phases} \sum_{t=0}^{T_s} \ell(x_t, u_t | r_s) + \ell_T(x_T)$$

$$\text{s.t.} \quad x_0 = \hat{x} \tag{4a}$$

$$\forall t, s \quad x_{t+1} = f(x_t, u_t | r_s) \tag{4b}$$

$$\forall t \quad u_t \in \mathcal{U} \tag{4c}$$

$$\forall t, s \quad x_t \in \mathcal{X}_s \tag{4d}$$

where ℓ_k and ℓ_N are respectively the running and terminal cost. $\{x\}$ and $\{u\}$ are the decision variables which we take as discretized at the nodes indexed by t. The control vector u contains the 3D forces at each contact point, constrained to the friction cone $\mathcal{U}$. The centroidal states x should be chosen so that there exists a valid whole body movement that can achieve x [22]. We put this last constraint under the abstract form $x \in \mathcal{X}_s$ depending on the phase s and will come back later on it. This OCP is still quite abstract. In the following, we will derive several versions of this OCP by increasing the complexity, starting from the OCP formulated in [22]. Our goal is to show how several motion features can be formulated to augment the accuracy or the range of decision of the MPC and experimentally show the consequences of these augmentations.

III. Linearization of the dynamics

The first problem is a simple reformulation of the convex formulation proposed in [13]. The CoM is approximated by the desired CoM p^* in the Euler equation (1b). With this assumption, we can reduce the complexity and make the system convex as the dynamic matrix no longer induces coupling between control and state vector :

$$\mathcal{I}\dot{\omega} = \sum_{i=1}^{n_c} (r_i - p) \times f_i \approx \sum_{i=1}^{n_c} (r_i - p^*) \times f_i \tag{5}$$

A. Convex Dynamic Model

The chosen dynamic (1a) - (5) is discretized with an implicit scheme of integration such as $P_{t+1} = P_t + \Delta t V_{t+1}$ to estimate with more accuracy the position, bringing Δt^2 terms in the B matrix.

$$x_{t+1} = f_t(x_t, u_t) = Ax_t + Bu_t \tag{6a}$$

$$A = \begin{bmatrix} I_6 & \Delta t I_6 \\ 0_6 & I_6 \end{bmatrix} \tag{6b}$$

$$B = \begin{bmatrix} \frac{\Delta t^2}{m} I_3 & \dots & \frac{\Delta t^2}{m} I_3 \\ \Delta t^2 \mathcal{I}^{-1} [r_1 - p*]_\times & \dots & \Delta t^2 \mathcal{I}^{-1} [r_n - p*]_\times \\ \frac{\Delta t}{m} I_3 & \dots & \frac{\Delta t}{m} I_3 \\ \Delta t \mathcal{I}^{-1} [r_1 - p*]_\times & \dots & \Delta t \mathcal{I}^{-1} [r_n - p*]_\times \end{bmatrix} \tag{6c}$$

where Δt is the integration time between the nodes, I_6 the identity matrix of size 6 and $[\dots]_\times$ a 3x3 skew-symmetric matrix representing the cross products as matrix multiplications.

B. Cost function

The cost function are three cost terms. The first one is a quadratic cost which penalizes the state vector with regards to the desired state. The second quadratic cost penalizes the relative command vector. We use the relative reference force $f_z^{ref} = f_z - \frac{mg}{n_c}$, where f_z is the normal ground reaction force, since it brings the forces to equilibrium and also leads to a faster convergence. Finally, we use a penalty cost to implement the friction cone inequality constraints. The solver used works by penalization and inequalities constraints can not be formulated as such. Thus, to keep the force inside the

friction cone, constraints are translated in a penalisation term using a discrete 4-facet cone approximation and for each foot :

$$\ell_{cone,i}(f) = \frac{1}{2}||(f_x - \mu f_z)^+||^2 + \frac{1}{2}||(-f_x - \mu f_z)^+||^2 + \frac{1}{2}||(f_y - \mu f_z)^+||^2 + \frac{1}{2}||(-f_y - \mu f_z)^+||^2 + \frac{1}{2}||(f_z)^+||^2 \tag{7}$$

where $y^+ = \max(y, 0)$. The cost is thus activated and strongly penalized only if u does not respect the inequality constraints. However, this penalisation formulation does not guarantee that the constraints will be respected. Some margin for the coefficient friction is needed. We experimentally show that this approximation works very well.

IV. Keeping the bilinear dynamics

A. Non Linear model

In this second OCP, the assumption on the lever arm (5), is withdrawn. We no longer use the desired CoM position p^* but the predicted CoM p in the cross product. The B matrix (6c) therefore varies with the state. As a result, the coupling between state and control is re-introduced. Moreover, the derivatives of the dynamics with respect to state F_x are not invariant anymore :

$$F_x = \frac{\partial f_t}{\partial x_t} = A + u_t \frac{\partial B}{\partial x_t} \tag{8}$$

B. Kinematic constraint

In the previous controller, no precaution are taken to ensure that x is admissible (i.e. $x \in \mathcal{X}_s$). In particular we cannot guarantee that the CoM or orientation move too far away from the contact. As we now accept to have a non-linear formulation, we take it as an opportunity to add a term to enforce it. This kinematic limit is approximated by penalizing the distance between the shoulder and the contact point when it reaches 80% of the leg limit. The position of the shoulder in R_0 can be formulated such as :

$$\begin{cases} x_{sh} = x + p_x - p_y\psi \\ y_{sh} = y + p_y + p_x\psi \\ z_{sh} = z + p_y\phi - p_x\theta \end{cases} \tag{9}$$

where $\begin{pmatrix} p_x & p_y & 0 \end{pmatrix}^T$ is the position of the shoulder in $\mathcal{R}_l$ and $\begin{pmatrix} x & y & z \end{pmatrix}^T$ the position of the CoM in $\mathcal{R}_0$. The shoulder-to-cost penalization is thus the following :

$$\ell_{sh}(x_t) = ||((x_{sh}-x_c)^2+(y_{sh}-y_c)^2+(z_{sh}-z_c)^2-d_{lim}^2)^+||^2$$

where $x_c = \begin{pmatrix} x_c, y_c, z_c \end{pmatrix}^T$ is the contact placement in R_0.

V. Simultaneous foothold optimization

The third controller additionally optimizes the foot placements. The contact placements are therefore a decision variable and the contact phase S only depends on the contact timing. The new OCP can be rewritten :

$$\min_{\{x\},\{u\},\{r\}} \sum_{s \in phases} \sum_{t=0}^{T_s} \ell(x_t, u_t|r_s) + \ell_T(x_T) \tag{10a}$$

$$\text{s.t.} \quad x_0 = \hat{x} \tag{10b}$$

$$\forall t, s \quad x_{t+1} = f(x_t, u_t|r_s) \tag{10c}$$

In addition to the four cost terms described earlier in Sections III and IV, namely the state regularization, the relative force regularization, the friction cone violation penalty and the kinematic limit penalty, three other quadratic costs are proposed to reduce the search space : i) The distance of the footstep is penalized. ii) The contact placement is penalised to keep the position of the foot around a certain heuristic. The same heuristic terms given to the previous OCP model as contact point serve here to lead the optimization. iii) A last cost term is proposed which allows to stop the optimization of the contact placement when the foot is approaching the ground. It avoids destabilisation at the end of the flying phase by preventing lateral velocities. This last cost term nicely disappear in the next controller.

VI. OCP with period optimization

This last OCP formulation includes contact timing as a decision variable in addition to the decision variables of (10). Thus, we evaluate the behaviour of the OCP while optimizing the gait period, the footholds position, the trajectory of the CoM and the reaction forces at the same time :

$$\min_{\{x\},\{u\},\{r\},\{T\}} \sum_{s \in phases} \sum_{t=0}^{T_s} \ell(x_t, u_t|r_s) + \ell_T(x_T)$$

$$\text{s.t.} \quad x_0 = \hat{x} \tag{11a}$$

$$\forall t, s \quad x_{t+1} = f(x_t, u_t|r_s) \tag{11b}$$

Looking only at the cendroidal dynamics, the solver will always take advantage of reducing the duration of the contact phase. Yet, short duration of the phase implies high velocity of the flying feet, in particular if the solver decides to also takes large step. We, then, have to add a cost term to carry this information. The trajectory of the flying feet are represented by polynomial functions of degree 5 chosen to nullify the speed and acceleration at the impact point and to take into account the current speed and acceleration of the feet which are already flying. Only the lateral velocities are penalized since they are higher than the speed along the Z axis during the movement. The maximum speed along this trajectory is only a function of the start and the end location, and of the current velocity of the foot if any. For predicted phases that are not started, maximum speed is reached at the middle of the polynomials and then easily evaluated. The maximum of a 4 degree polynomial function with non null initial conditions cannot be computed in a reasonable time. Thus, the velocity of the foot for the first phase is evaluated and discretized at $\alpha\Delta T$, in a much simpler manner :

$$V_x(\alpha\Delta T) = V_0 b_0(\alpha) + \Delta T acc_0 b_2(\alpha) + \frac{\Delta x}{\Delta T} b_3(\alpha) \tag{12}$$

where V_x is the speed of the foot along the X axis, Δx the distance done during the phase along the X axis, v_{lim} the speed limit, b constant coefficients depending on α, V_0 the current speed of the foot and acc_0 the current acceleration. For one foot, the penalization is thus :

$$\ell_{speed}(\Delta T, \Delta x, \Delta y) = \sum_{\alpha}[(\Delta T b_1 + \Delta T^2 b_2 + \Delta x b_3)^2 + (\Delta T b_1 + \Delta T^2 b_2 + \Delta y b_3)^2 - \Delta T V_{lim}^2]^+ \tag{13}$$

For the next phases, it is the same expression with $\alpha = \frac{1}{2}$ holds.

VII. Differential dynamic programming solver

To solve the OCP formulations described above, the differential dynamic algorithm programming (DDP) is used from the library Crocoddyl [24]. The discrete dynamic model is inserted into a multiple shooting problem. For the two first OCP, the implementation is quite straightforward since the state and the control vector does not change along the temporal horizon. OCP proposed in (10) and (11) are actually parameterized OCP and to handle them with DDP the state variables are augmented with the new decision variables.

A. OCP with foothold optimization

For the foothold OCP, with r_t the position of the foot in $\mathcal{R}_0$: $y_{t+1} = (x_t, r_t)$. A first shooting node is introduced to handle the dynamics (1a), (1b) with the forces as control vector. The second one is used to determine the next position of the feet and as the contact timing is predefined, this node is inserted before the modification of the contact point.

$$\forall t \quad \dot{y_{t+1}} = \begin{pmatrix} f_t(x_t, u_t, p_t) \\ g_t(p_t, u_t) \end{pmatrix} \tag{14}$$

where f corresponds to the discretized dynamic model, and g is function which allocates the new position of the feet when the contact switch occurs.

For the model representing the dynamic, f is the same as the non linear OCP. The position of the feet is constant :

$$g_t(p_t, u_t) = p_t \tag{15}$$

Here is the model inserted between the dynamic models. The control corresponds to the distance between the previous and the new contact point and the size of the control thus depends of the number of foot placement modified. In this node, the state x_t is constant $f(x_t, u_t, p_t) = x_t$ and here is the variation of the foot position :

$$g_t(p_t, u_t) = p_t + u_t \tag{16}$$

B. OCP with period optimization

The same method is used and the state is augmented with the integration time between the nodes : $\Delta t_s = \frac{T_s}{N_s}$ where N_s is the number of nodes in the phase s. A third type of node, where the control is size one and equal to the integration time, is inserted before each flying phase and stance phase and by modifying the integration time between the nodes, the period of the phase is thus modified. The cost proposed on the state vector is no longer quadratic and depends on the integration time :

$$\ell_{state}(x_t|s) = \frac{\Delta t_s}{2}(x_t - x^*)^T W_s (x_t - x^*) \tag{17}$$

where W_s the weight vector, x_t the state and x^* the desired state. Thus to minimize the cost, the optimization leads to minimize the period of the phase by reducing the integration time if no additional constraints are formulated. This phenomena is highly sought since first it avoid the fall in local optimum for any understandable reasons. Then it allows to lead the optimizing toward a faster solution. To finish, we assume that the behaviour of the robot is more stable with a small gait period since the step between the foot is lower and the quadrupedal support are more frequent. A minimal integration time is required.

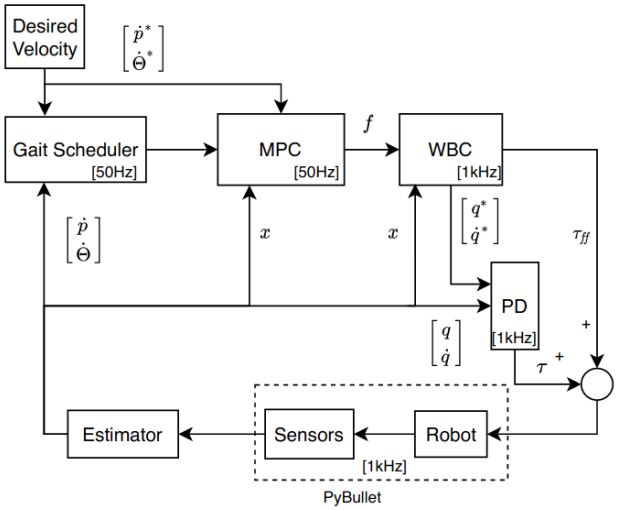

Fig. 2. Architecture of the controller. x is the state vector, q the joint configuration, τ the torque from the PD controller and τ_{ff} the feed-forward torques, f the contact forces.

VIII. Implementation details

The four OCP formulations described above are tested, evaluated and compared in simulation as a model predictive controller. The Solo12 robot is tested in simulation. They are integrated inside a state of the art architecture of quadrupedal robots for cyclic locomotion. It is closed to the hybrid architecture described in [1]. The MPC bloc, running at 50Hz, is coupled with a whole body controller at 1kHz. The environment and the simulated feedback of the sensors are generated using pyBullet simulator [25]. We have chosen to run our test in simulation as each controller will be extensively tested until it reaches its limit, which we could not afford to do on the real robot.

A. Model predictive controller

The OCP formulations are used as MPC and receive the estimated state, the desired state coming from the reference velocity target and the position of the foot in local frame given by the footstep generator. The heuristic to choose

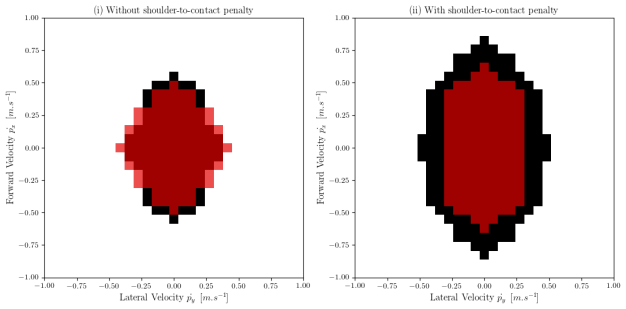

Fig. 3. Viable Operating Region. Comparison of linear and non linear MPC with the shoulder-to-contact point penalization on the right (ii). On the left (i), the non-linear solver tends to elevate the body, that the WBC then fails to track, although both solves behave quite similar. The benefits of completely modeling the lever arm dynamics is highlight by the larger area on the right.

the footstep location is the same than described in the [1]. The MPC generates a predictive state trajectory and ground reaction forces to apply. One period of gait is chosen for the temporal horizon, around 0.5s. The number of nodes is chosen depending on the integration time between them. 15 to 30 nodes are usually taken. To warm-start the MPC, the predicted state and forces computed at the previous control cycle and slipped of one iteration in the timeline is used for the new control cycle. For the last node, the values at the equilibrium point are chosen.

B. Whole body controller (WBC)

The foot in swing phase follow a polynomial trajectory of degree 5 in X,Y and Z axis to ensure negative acceleration and velocity at the impact point. The trajectory is updated each iteration with current acceleration/velocities of the foot and the new contact point. To avoid lateral velocities at the impact, the contact point is locked when 90% of the flying phase is done.

To generate the inverse dynamic control, we use a task space inverse dynamics (TSID) [26]. Using the estimated state, contact forces from the MPC and position of the feet, the WBC computes the torques, position and velocities of joints. The computed torques are then tracked using a proportional derivative controller with feed-forward. In order to highlight the behavior of the MPC rather than the WBC, we voluntarily choose a high frequency for it at 1Khz.

IX. Results

This section concerns the results obtained in simulation. The objective is to compare the different MPC to understand the impact of proposed augmentations done. To do so, a statistical approach to determine its viable operating region has been set up. The results obtained are also shown in the video[1].

[1]https://peertube.laas.fr/videos/watch/d6d90690-3262-4e61-9d03-08a2623726e3

A. Checking the force-cone penalization

To first validate the DDP solver, the first linear controller has been compared to the QP formulation proposed in [13], solved with the OSQP solver [27]. We checked by an empirical validation that 3 iterations of DDP solver are enough to solve the problem in every situations. We have chosen 16 nodes and a time horizon of 0.32s. On a forward velocity $V^* = 0.3m.s^{-1}$, the root means square error (RMSE) between the forces computed by [13] and the forces computed by DDP solver using the force cone penalization is only $\Delta f = 0.3N$. The next results concerning the computation time are obtained with 5000 trials for one iteration of the DDP algorithm.

TABLE I
Benchmark of the DDP for OCPs

OCP	Mean [ms]	Min - Max [ms]
Linear [ms]	0.327	0.302 - 0.882
Non Linear [ms]	0.351	0.305 - 0.938
Footstep [ms]	0.676	0.618 - 1.271
Dt Optim [ms]	0.769	0.725 - 1.511

B. Influence of the lever arm assumption

In the remaining, we compare the formulation by running a complete MPC simulation with a given reference basis velocity. The simulation is validated if the robot reaches steady cycle with the commanded velocity, and invalidated if the robot falls or does not reach it. We then plot the viable operating region $V_x - V_y$ and $V_x - \omega$. We first compare the linear OCP in section III with the non-linear OCP of section IV. In Fig. 3 on the right, we compare the two MPC with the cost term introduced in section IV that penalizes the distance between the shoulder and the contact point. Without the kinematic constraint, convex and non-linear MPC are quite equivalent. The area are almost similar in Fig. 3-left. The non-linear MPC slightly improves the behaviour for forward velocities whereas it is less efficient for lateral velocities. Indeed, it takes advantage of the cross-product relative to the lever arm (1b) by elevating the robot body for such lateral velocities. It results in movements not achievable by the robot with the CoM too high.

The shoulder-to-contact penalty cost allows the OCP to handle better this issue. On the opposite, this term degrades the behaviour of the linear MPC for lateral velocities as highlighted by the Fig.3-right although we do not get an intuition of this effect. In total the non-linear OCP performs much better it particular at high speed. This non-linear problem does not need a complex warm-started to converge toward an acceptable solution. The number of iteration needed to converge or the computation time are equals.

C. Optimization of the footsteps

Fig. 4 summarizes the performances of the three first OCP with shoulder-to-contact penalty. The non linear model with optimization of the footholds improves first the behaviour of the walk of the robot by producing more stable locomotion.

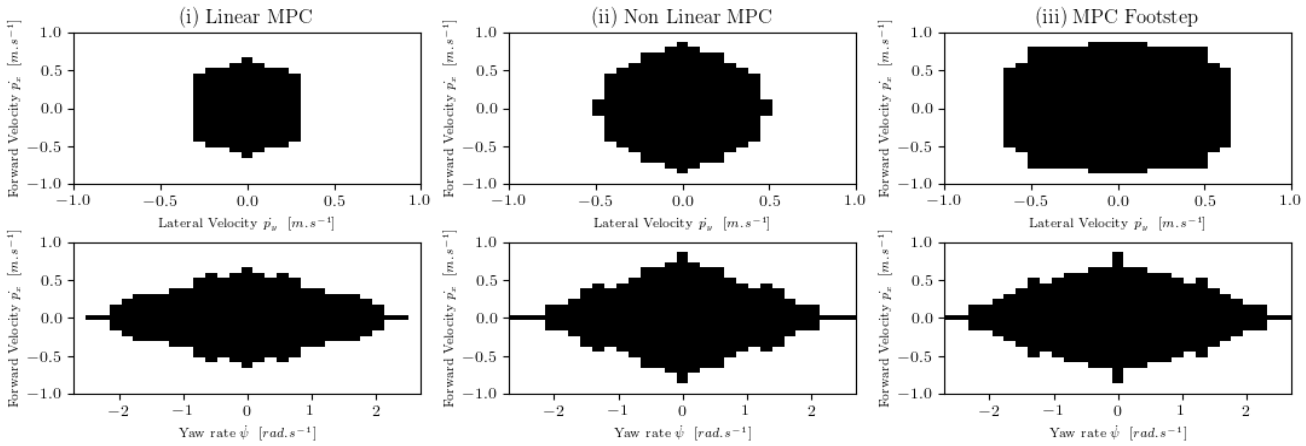

Fig. 4. Comparing the 3 OCP : (i) linear OCP, (ii) non-linear OCP with fixed footholds, (iii) non-linear OCP with foothold as decision variable. In the three cases, the shoulder-to-cost penalty is activated. Optimally deciding the footholds greatly helps to stabilize the system.

However, the computation time is higher : one iteration of the DDP solver, takes approximately 0.6ms. It also requires more iterations, up to 5 with a proper warm-start.

D. Optimization of the gait period

In nominal situation, the MPC tends to minimize the period as explained in VII-B. In nominal walk, even at high speed, the timings are marginally changed by the solver and the viable operating region is not modified. Optimizing the gait period during the walk is mostly interesting to handle a perturbation. When a perturbation occurs, the controller is able to modify the duration of the flying phase and the duration of the stance phase. We have tried to quantify this augmented robustness, yet in simulation, all controllers resists to unrealistic perturbations (up to $\Delta_V = 1m.s^{-1}$). As an example, the system is perturbed by a linear velocity during a double support phase in Fig. 5. The stance phase is then elongated as shown in Fig. 6 to allow a better rejection of the perturbation on the roll, pitch and yaw angles since it is more stable during quadrupedal support.

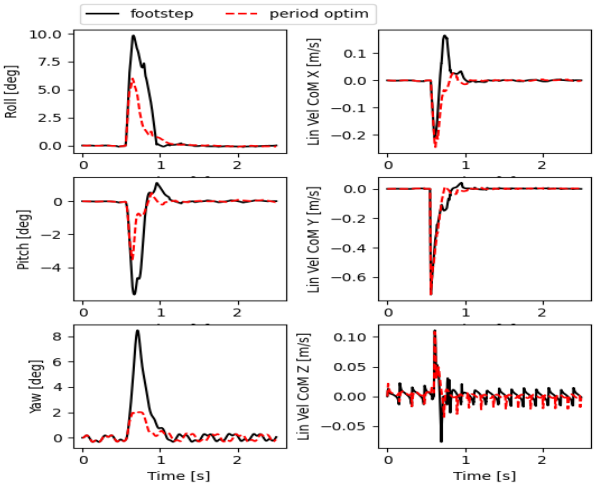

Fig. 5. Angular position and linear velocities of the robot in the world frame. Perturbation of the lateral velocity of 0.7 $m.s^{-1}$ during a double contact phase.

X. Conclusion and Perspective

We have proposed several formulations of locomotion MPC and evaluated statistically and qualitatively the impact of their formulation on the control.

Our first observation is that linearity of the dynamics or of the cost is not a mandatory feature for computational efficiency. Actually, we would be more interested by strict convexity, for which linearity is only a proxy [28].

We empirically demonstrate the importance to consider in the MPC the exact dynamics of the lever arm, the kinematic limit, the position and timing of the contacts. We showed that these non-linear features only marginally increase the computational burden. On the other hand, some approximations of the dynamics, or some constraints that are discarded from the problem because of their non-linearity, might have a strong impact on the quality of the controller. We believed the same comparative methodology should be continued, and will integrate next the other missing terms.

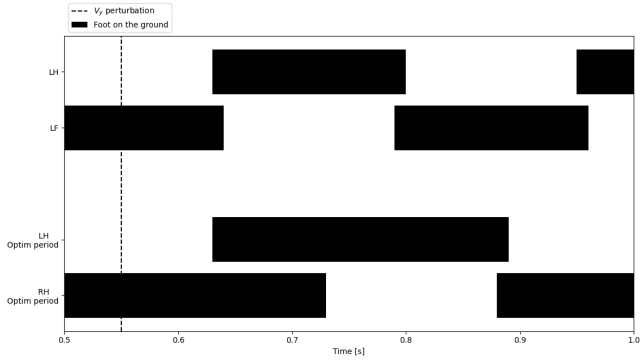

Fig. 6. Stance phase in dark for the left hind and front foot, depending on which controller is used (optimization of the period or not). The gait, predefined, is symmetric for the right feet.

We are also looking at a way to extend the systematic validation on the real hardware without damaging the robot.

REFERENCES

[1] D. Kim, J. D. Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," 2019.

[2] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[3] M. Neunert, M. Stauble, 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, no. 3, pp. 1458–1465, jul 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8276298/

[4] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, oct 2012, pp. 5026–5033. [Online]. Available: http://ieeexplore.ieee.org/document/6386109/

[5] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the HRP-2 humanoid," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, sep 2015, pp. 3346–3351. [Online]. Available: http://ieeexplore.ieee.org/document/7353843/

[6] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, sep 2017, pp. 3359–3365. [Online]. Available: http://ieeexplore.ieee.org/document/8206174/

[7] M. V. Minniti, F. Farshidian, R. Grandia, and M. Hutter, "Whole-body mpc for a dynamically stable mobile manipulator," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3687–3694, 2019.

[8] H. Kim, H. Seo, S. Choi, C. J. Tomlin, and H. J. Kim, "Incorporating Safety Into Parametric Dynamic Movement Primitives," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2260–2267, jul 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8648156/

[9] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols, J. Fiene, A. Badri-Spröwitz, and L. Righetti, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.

[10] M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli, "Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1502–1509, jul 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7845678/

[11] 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 *IEEE International Conference on Robotics and Automation*. IEEE, 2003, pp. 1620–1626. [Online]. Available: http://ieeexplore.ieee.org/document/1241826/

[12] A. W. Winkler, F. Farshidian, D. Pardo, M. Neunert, and J. Buchli, "Fast Trajectory Optimization for Legged Robots Using Vertex-Based ZMP Constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2201–2208, oct 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7970144/

[13] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8594448/

[14] A. W. Winkler, F. Farshidian, M. Neunert, D. Pardo, and J. Buchli, "Online walking motion and foothold optimization for quadruped locomotion," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5308–5313.

[15] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Footstep Placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, jan 2010. [Online]. Available: https://www.tandfonline.com/doi/full/10.1163/016918610X493552

[16] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, nov 2016, pp. 842–849. [Online]. Available: http://ieeexplore.ieee.org/document/7803371/

[17] J. Carpentier and N. Mansard, "Multicontact Locomotion of Legged Robots," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1441–1460, dec 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8558661/https://hal.archives-ouvertes.fr/hal-01520248/

[18] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–8, jul 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2185520.2185539

[19] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, jul 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8283570/

[20] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[21] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, oct 2013. [Online]. Available: http://link.springer.com/10.1007/s10514-013-9341-4

[22] J. Carpentier, R. Budhiraja, and N. Mansard, "Learning Feasibility Constraints for Multicontact Locomotion of Legged Robots," in *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, jul 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p31.pdf

[23] R. Featherstone, *Rigid Body Dynamics Algorithms*. Boston, MA: Springer US, 2008. [Online]. Available: http://link.springer.com/10.1007/978-0-387-74315-8

[24] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *ICRA 2020 - IEEE International Conference on Robotics and Automation*, 2020. [Online]. Available: http://arxiv.org/abs/1909.04947

[25] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2019.

[26] A. Del Prete, N. Mansard, O. E. Ramos, O. Stasse, and F. Nori, "Implementing Torque Control with High-Ratio Gear Boxes and Without Joint-Torque Sensors," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1550044, mar 2016. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219843615500449

[27] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, dec 2020. [Online]. Available: http://link.springer.com/10.1007/s12532-020-00179-2

[28] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004. [Online]. Available: https://dl.acm.org/citation.cfm?id=993483