An MPC Walking Framework With External Contact Forces

Sean Mason[1], Nicholas Rotella[1], Stefan Schaal[1,2], and Ludovic Righetti[2,3]

Abstract—In this work, we present an extension to a linear Model Predictive Control (MPC) scheme that plans external contact forces for the robot when given multiple contact locations and their corresponding friction cone. To this end, we set up a two-step optimization problem. In the first optimization, we compute the Center of Mass (CoM) trajectory, foot step locations, and introduce slack variables to account for violating the imposed constraints on the Zero Moment Point (ZMP). We then use the slack variables to trigger the second optimization, in which we calculate the optimal external force that compensates for the ZMP tracking error. This optimization considers multiple contacts positions within the environment by formulating the problem as a Mixed Integer Quadratic Program (MIQP) that can be solved at a speed between 100-300 Hz. Once contact is created, the MIQP reduces to a single Quadratic Program (QP) that can be solved in real-time (< 1kHz). Simulations show that the presented walking control scheme can withstand disturbances 2-3x larger with the additional force provided by a hand contact.

I. INTRODUCTION

In bipedal locomotion, online pattern generators have seen much success because of their ability to adapt to continually changing environment and internal states. In order to generate new plans quickly, researchers typically utilize a simplified dynamic model for fast planning of center of mass (CoM) trajectories. The plans are then tracked using a whole-body controller typically utilizing either inverse kinematics or inverse dynamics. Historically, the Linear Inverted Pendulum Model (LIPM) has been the most prominent model for biped walking because it is linear and yet effectively captures the dynamics for walking on flat surfaces. The low dimensionality and linearity of the LIPM allow the dynamics of the model to be regulated through linear control methods for compact formulations and efficient computations.

In [1], Kajita et. al introduced the preview control of the LIPM, which optimizes the CoM trajectory over a time-horizon. For a linear system with no inequality constraints, this is equivalent to solving the well known finite horizon Linear Quadratic Regulator (LQR), which yields a time-varying feed-forward and feedback policy. If the problem
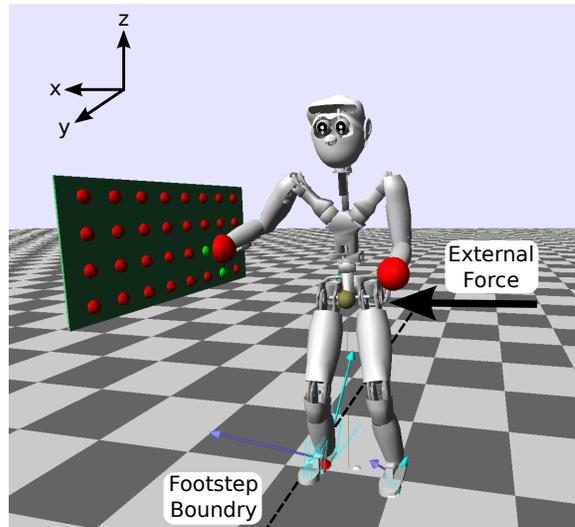
Fig. 1: Visual of a push experiment setup mid push. The spheres located on the wall mark available contact points, in which green spheres specify those considered reachable.

includes inequality constraints, the solution no longer provides a feedback policy that stabilize the trajectory. (One approach used to rectify this is) Model Predictive Control (MPC). In an MPC framework, feedback is generated by continuously resolving the constrained optimization from the measured state and applying the first feed-forward command. In [2], Center of Pressure (CoP) constraints in the form of linear inequalities are introduced and the problem is solved in a receding horizon manner to produce CoM trajectories robust to perturbations. This approach was further extended to additionally optimize over the footstep locations [3], allowing the robot to take recovery steps and track a reference velocity. Researchers have further developed MPC methods using the LIPM to adjust footstep timing [4], [5], [6], plan 3D trajectories [7], and control the Divergent Component of Motion (DCM) [8], [9]. A nonlinear extension also allows the robot to deal with walking around obstacles [10].

The problem of designing receding horizon controllers that handle rough terrain and allow the use of hand contacts remain a largely open area. More general models such as the centroidal momentum dynamics have gained popularity to generate patterns allowing multi-contact behaviors [11], [12] but they are not convex and are computationally more demanding. While significant progress has been made with these algorithms, they have not been applied for receding horizon control of legged robots.

Simpler models still allowing hand contacts have also

been explored. In [13], hand contacts with the environment were introduced in an MPC problem by deriving a non-linear dynamical model of the CoM that included an external wrench. To solve the non-linear optimization problem a Newton scheme was presented in which each step of the optimization was bounded such that the intermediate solutions were always feasible. While external hand contacts were used for stability, the contact locations and timing were predetermined. Additionally, the contact considered was a grasp and rather than a push, thus neglecting friction cone constraints. In [14], an external wrench is also included in the MPC scheme to design a walking pattern generator for physical collaboration. To drive cooperation with a second agent, the robot assumes the role of either a leader or follower in the task of collectively carrying an object. Contact forces are included in the dynamics and cost function such that the problem remains linear in the constraints and quadratic in the cost.

In this paper, we extend simplified walking pattern generators to exploit hand contacts when stability cannot be maintained by stepping alone. We treat the hand contact as a complement to footstep planning. To limit computational complexity and allow receding horizon control, we propose a two-part optimization that automatically calculates how to use hand contacts when the CoM and footstep adjustment alone are not sufficient. The contact locations are chosen among multiple contact points, considering both the time to reach the contact and friction cone constraints. Our decomposition maintains the computational simplicity of preview controllers based on the LIPM dynamics in the first optimization. The second part then selects a hand contact location and contact force to stabilize the robot when necessary by solving a low-dimensional mixed integer quadratic program (MIQP). The result is a controller that reactively decides when and how to use the hands of the robot to further stabilize the walking controller. Push recovery simulation experiments demonstrate that our approach can significantly improve the stability of a walking robot under strong perturbations.

II. Newton-Euler Dynamics

We begin by deriving the LIPM dynamics with the addition of an external contact force $f_c = [f_c^x, f_c^y, f_c^z]$, corresponding to a hand contact located at point $p = [p^x, p^y, p^z]$. Similar to [14], we separate foot contact forces, f_i, from all other forces in the Newton-Euler Equations (1), (2).

$$m(\ddot{c} + g) = f_c + \sum f_i \tag{1}$$

$$c \times m(\ddot{c} + g) + \dot{L} = p \times f_c + \sum s_i \times f_i \tag{2}$$

, where c is the CoM, L is the angular momentum taken about the CoM, s_i is the location of the co-planar foot contacts, and the coordinate frame has the z axis aligned with the gravity vector. Dividing the Euler equation by the foot contact forces in the z direction, f_i^z, of the Newton equation and rearranging yields:

$$\frac{\sum s_i \times f_i}{\sum f_i^z} = \frac{c \times m(\ddot{c} + g) - (p \times f_c) + \dot{L}}{m(\ddot{c}^z + g) - f_c^z}.$$

Under the LIPM assumptions, the angular momentum and the CoM height remain constant ($\dot{L} = 0$, $\ddot{c}^z = 0$), with the ground plan at zero ($s_i^z = 0$). We can then write the equations for the Zero Moment Point (ZMP) which we will refer to as Z_{hand} (i.e. ZMP with external hand contact).

$$Z_{\text{hand}}^{x,y} = \frac{\sum s_i^{x,y} f_i^z}{\sum f_i^z} = -\frac{mc^z \ddot{c}^{x,y} - mgc^{x,y} - p^z f_c^{x,y} + p^{x,y} f_c^z}{mg - f_c^z} \tag{3}$$

Notice that if there is no contact force (i.e. $f_c = 0$) this term reduces to the familiar ZMP equations for the LIPM, Z_{lipm}, [1] :

$$Z_{\text{lipm}}^{x,y} = \frac{\sum s_i^{x,y} f_i^z}{\sum f_i^z} = c^{x,y} - \frac{c^z \ddot{c}^{x,y}}{g}. \tag{4}$$

III. Shifting the ZMP Support Polygon

To express the difference between the ZMP with and without a hand contact, we introduce the variable ΔZ such that:

$$\Delta Z = Z_{\text{hand}} - Z_{\text{lipm}} \tag{5}$$

or equivalently

$$Z_{\text{hand}} = Z_{\text{lipm}} + \Delta Z. \tag{6}$$

By expressing the ZMP this way, one can see how the traditional ZMP from the LIPM shifts with the addition of an external force. As mentioned in [14], we can additionally think about the implication of this force on the ZMP bounds. That is, we can rewrite

$$\underline{Z} \leq Z_{\text{lipm}} + \Delta Z \leq \overline{Z} \tag{7}$$

as

$$\underline{Z} - \Delta Z \leq Z_{\text{lipm}} \leq \overline{Z} - \Delta Z. \tag{8}$$

Eq. (8) implies that it is possible to shift the effective support polygon that bounds Z_{lipm} through the addition of an external force. Solving for ΔZ yields the nonlinear expression

$$\Delta Z^{x,y} = \left(\frac{p^z}{mg - f_c^z} \right) f_c^{x,y} + \left(\frac{-p^{x,y} - \frac{c^z \ddot{c}^{x,y}}{g} + c^{x,y}}{mg - f_c^z} \right) f_c^z. \tag{9}$$

IV. Optimization Formulation

Due to constraints on footstep locations (e.g. kinematic limits and environmental obstructions) and CoM motion (e.g. acceleration limits), it is not always feasible to keep the ZMP within the specified bounds. We remedy these situations by shifting the ZMP bounds by specifying an external force with the hand. One approach is to embed Eq. (9) into the optimization resulting in nonlinear dynamics. It also introduces the variables f_{ci}, $i = 1, \ldots, N$, and p for a total
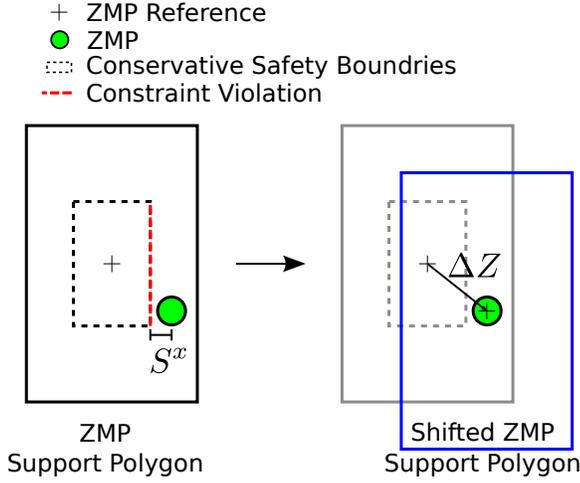
Fig. 2: When the conservative ZMP bounds are violated the robot seeks to create a contact force among the available options such that the "Shifted ZMP Support Polygon", Eq. (8) is centered at the current ZMP location.

---

1 while *Robot Has Not Fallen* **do**
2   Find reachable contact points from list.
3   Constrain slack variables based on time to reach contact
4   Solve LIPM based MPC (Sec. IV-A) to generate CoM, footstep, and slack trajectory.
5   **if** $\|Slack\ Trajectory\| > 0$ **then**
6   **if** *Hand In Contact* **then**
7   Solve QP to calculate the external force that compensates for the ZMP error (Sec. IV-C)
8   **else**
9   Solve MIQP to decide the best contact location (Sec. IV-D).
10   Track solution with hand.
11   **end**
12   **end**
13 end

Algorithm 1: MPC Walking Scheme With Hand Contacts

---

of $(3N+3)$ new decision variables, where N is the number of time-steps in the MPC preview horizon. This problem grows even further if one would like to consider points on different surfaces with different constraints (i.e. friction cone and boundary constraints). Because of the increased dimensionality, complexity, and thus the time needed to solve this larger nonlinear optimization in an online manner, we propose a more light-wight approach that remains linear by breaking the problem into two separate optimizations. In the first optimization, we decide both when an external force is necessary and what value $\Delta Z^{x,y}$ is desired to shift the support polygon. In the second optimization, we decide both the location of the hand contact p and the force f_c. While separating the optimization into two stages will result in a sub-optimal solution, we would argue that in the context of a quick recovery motion, feasibility is more important than optimality. The result is a walking controller that hand contacts in a reactive manner when the control over the CoM and footstep locations alone is deemed insufficient. An outline of the algorithm logic is shown in Alg. 1.

A. Stage 1: MPC of CoM Trajectory and Footstep Locations

In the first stage of the optimization, we formulate an MPC scheme to minimize CoM jerk, ZMP tracking error, and CoM velocity tracking error. The scheme is based on the well known approach from [3] so we omit the details for brevity. The only difference in our approach is that we write the ZMP constraints with the addition of a trajectory of slack variables, $S_k, k = 1 \ldots N$. We can write this constraint in the same form as Eq. (8), omitting the subscript k, as:

$$\underline{Z} - S \leq Z_{\text{lipm}} \leq \overline{Z} - S, \quad (10)$$

where $\overline{Z}$ and $\underline{Z}$ correspond to the upper and lower bounds of a conservative support polygon (see Fig. 2). The amount we scale the support polygon boundaries is directly linked to

how early the hand contact will be triggered to help. When the ZMP leaves the conservative safety boundaries, $\|S\| > 0$, we treat this as an indication that additional support is necessary to maintain balance, triggering the use of a hand contact. In order to account for the fact that it will take some time for the hand to reach the contact point location, we constrain the first N_D time steps of the slack variable S to be zero, where N_D is representative of the time-steps it will take for the hand to make contact. Furthermore, we add a weighted quadratic term to minimize S in the cost function to favor solutions that only use footstep and CoM adjust for balance.

B. Stage 2: Hand Contact

Once the first optimization signifies the need for a hand contact, $\|S\| > 0$, a second optimization calculates the necessary contact force and location to shift the ZMP boundaries by ΔZ such that the ZMP is centered within the support polygon (see Fig. 2). While the hand has not yet created a contact, we formulate a MIQP that includes all the reachable contact point locations, along with the corresponding friction cones, to determine which, if any, is the optimal contact location to enact the shift specified by the ΔZ trajectory. Once a contact has been made, we remove all other contacts from the optimization and solve for the contact force as single QP that can be solved in real-time. We begin by formulating the problem for when the hand is already in contact.

C. Determining the Contact Force

In Eq. (9), we can see that with the values of $\Delta Z^{x,y}$ determined, we have two equations and three unknowns. Geometrically, the solutions for the 3D force vector form a line can be rewritten in parametric form as:

$$f_c^{x,y} = \frac{p^{x,y} + \frac{c^z \ddot{c}^{x,y}}{g} - c^{x,y} - \Delta Z^{x,y}}{p^z} f_c^z + \frac{mg\Delta Z^{x,y}}{p^z}. \tag{11}$$

Note that $f_c^{x,y}$ is a linear function of f_c^z and thus the only decision variable is f_c^z. We include linearized friction cone constraints, by first changing the frame of our variables to be aligned with the surface normal of the contact:

$$\begin{bmatrix} f^b \\ f^t \\ f^n \end{bmatrix} = R \begin{bmatrix} f^x \\ f^y \\ f^z \end{bmatrix} \tag{12}$$

and write the following constraints

$$\begin{aligned}
f^b &\le u f^n &,& \quad f^b \ge -\mu f^n \\
f^t &\le u f^n &,& \quad f^t \ge -\mu f^n \\
f^n &\le f_{\max}^n &,& \quad f^n \ge 0 \quad ,
\end{aligned} \tag{13}$$

where μ is the static coefficient of friction that defines the linearized fiction cone. The cost function to optimize is then:

$$\underset{f^z}{\text{minimize}} \quad (f^b)^2 + (f^t)^2 + \kappa(f^n)^2 \tag{14}$$

$$\text{subject to} \quad \text{Friction Cone Constraints (13)}.$$

Reducing the gain κ favors solutions that are closer to the center of the friction cone and thus more robust to slipping. This optimization formulation specified by Eq. (13,11,14) will be referred to as QP1. Because the formulation of QP1 shifts the support polygon by exactly ΔZ, the optimization does not consider that the ZMP does not have to be centered in the support polygon. To address this, we include the addition of the **new** slack variable $S_z = [S_z^x, S_z^y]$ that is bounded as follows:

$$\underline{S_z} \le S_z \le \overline{S_z}. \tag{15}$$

Rewriting Eq. (11) as a function of S_z and f_z results in:

$$\begin{aligned}
f_c^{x,y} = &\left(\frac{p^{x,y} + \frac{c^z \ddot{c}^{x,y}}{g} - c^{x,y} - \Delta Z^{x,y}}{p^z} \right) f_c^z \\
&+ \left(\frac{1}{p^z} \right) S_z^{x,y} f_c^z + \left(\frac{mg}{p^z} \right) S_z^{x,y} + \left(\frac{mg\Delta Z^{x,y}}{p^z} \right).
\end{aligned} \tag{16}$$

Unfortunately, by introducing the new decision variable S_z we have added the bilinear term $S_z f_c^z$. In order to keep the problem linear and solvable with a QP, we introduce a convex relaxation of the bilinear term by means of a McCormick envelope [15]. In doing so, we create the new variables $W^{x,y} = S_z^{x,y} f_c^z$, more compactly written as W, and add the following new constraints to the system:

$$\begin{aligned}
\underline{S_z} f_c^z &+ S_z \underline{f_c^z} &- \underline{S_z}\,\underline{f_c^z} &\le W \\
\overline{S_z} f_c^z &+ S_z \overline{f_c^z} &- \overline{S_z}\,\overline{f_c^z} &\le W \\
\overline{S_z} f_c^z &+ S_z \underline{f_c^z} &- \overline{S_z}\,\underline{f_c^z} &\ge W \\
S_z \overline{f_c^z} &+ \underline{S_z} f_c^z &- \underline{S_z}\,\overline{f_c^z} &\ge W.
\end{aligned} \tag{17}$$
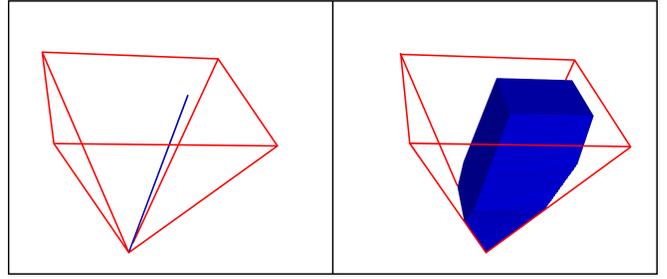
This new formulation will be referred to as QP2.

Fig. 3: The left shows the solution space when we solve for the exact ZMP shift (QP1) and the right for when we formulate the optimization as QP2.

D. Determining the Contact Location

In order to determine the contact location, we reason about reachable contact locations and their corresponding friction cone in the form of a MIQP. To define whether a contact has been chosen, we introduce the binary variables $b_i \in 0, 1$, $i = 1, \dots, N_c$. We then optimize over the time steps $\hat{k}$ corresponding to the non-zero elements in the ΔZ trajectory.

$$\underset{f_{i\hat{k}}^z, b_i}{\text{minimize}} \quad (f^b)_{i\hat{k}}^2 + (f^t)_{i\hat{k}}^2 + \kappa(f^n)_{i\hat{k}}^2$$

subject to

$$\sum_1^{N_c} b_i = 1 \tag{18}$$

$$b_i \implies \begin{cases} 0, & f_{i\hat{k}}^z = 0 \\ 1, & \text{(Friction Cone Constraints)}_i \end{cases} \tag{19}$$

V. EXPERIMENTS

A. Optimization Benchmarks

The experiments were conducted in this section using the optimization library Gurobi [16]. The average solve while searching among contacts, shown in Table I, was found to vary linearly with the length of the ΔZ trajectory and the number of contacts points. Once in contact, the optimization problem collapses down to a single QP which can be solved in 0.11 ms.

ΔZ Time-steps	Number of Contact Points (N_c)			
	5	10	20	40
16	5.2 ms	9.7 ms	18.7 ms	35.9 ms
8	3.2 ms	5.2 ms	9.8 ms	19.8 ms
4	1.8 ms	2.9 ms	5.4 ms	10.2 ms

TABLE I: Solving speed for multiple contact points considering different horizon lengths.

B. QP1 vs QP2

As shown in Fig. 3, by solving for contact forces using the formulation of QP2 as opposed to QP1 we expand the feasible solution space from a line to a volume. To compare how the solutions of QP1 and QP2, we performed a parameter sweep over μ and ΔZ and evaluated 5,000

random friction cones orientations and CoM accelerations for the following ranges: $\mu = [.1, 1]$ and $\Delta Z = [-.05, .05]$, $S_z = [-.02, .02]$. In total, there were 2,000,000 test cases evaluated, resulting in the following statistics:

	Success Rate	Average Force (N)
Force QP1	36.1%	28.0
Force QP2	73.6%	11.7

On average, formulation of QP2 favorably found optimal solutions for approximately twice the number test cases at approximately half the force. Because of the convex relaxation of the bilinear term used in QP2, we additionally need to check that the solutions satisfy the original problem constraints. Over the values tested, the maximum force violation was 0.037 N, which is negligible given the accuracy of force tracking capabilities of the robot seen in experiments.

C. Maximum Push Experiments

We evaluated the proposed control scheme in simulation through a series of push recovery tests on a 31 DoF torque controlled humanoid. We are particularly interested in the relationship between the max impulse the robot can recover from with and without an external force through hand contact with the environment. Additionally, we are interested in how the number of recovery steps before hand contact affects this maximum impulse value. To test this, the robot was pushed towards a wall while walking in place from three separate distances that allow for zero, one, and two recovery steps before hand contact. Additionally, the robot footstep locations were constrained to remain at a minimum distance from the wall (0.35m) so that the upper-body would not make contact unless the arm was purposefully extended (see Fig. 1). In order to track the CoM and footstep trajectory, we use an optimization-based whole-body inverse dynamics controller formulated as a QP, similar to [17], [18], [19], to resolve for joint torques at 1kHz. To track the desired forces, we implemented a simple force feedback policy that was added to the resulting joint torques from the whole-body QP solver (τ_{qp}):

$$\tau = \tau_{qp} + J_c^T(f_c + P(f_c - \hat{f}_c)) \qquad (20)$$

, where J_c is the Jacobian from the CoM to the hand in contact, $\hat{f}_c$ is the measured force at the end-effector in global coordinates, and P is a force-feedback gain. In the future, we plan to include these desired forces and friction cone constraints directly in the whole-body QP. The following parameters were used in these experiments: MPC horizon = 1.6 s, MPC Δt = 0.1 s, robot mass = 62.5 kg, CoM height = 0.78m, single support duration = 1.0 s, double support duration = 0.1 s, $\mu = 1$, and the support polygon boundary used in Sec. IV-A scaled the foot dimensions by a factor of 0.1.

The results of the push experiments are shown in Fig. 4. On average, using an external hand contact increased the maximum disturbance impulse by 233%, 156%, and 13% for the experiments where the robot was allowed zero, one, and two recovery steps respectively. While addition of a hand

contact greatly increased the walking controllers robustness to external impulses for zero and one recovery steps, in the 2-step recovery situation, the robot had a very high acceleration by the time the wall became close enough to make contact. This resulted in high desired contact forces that were difficult to track. The experiments confirm that because of the short optimization horizon and the reactive nature of this formulation, that approach presented is most effective in close proximity of contacts or when there is limited control authority through stepping.
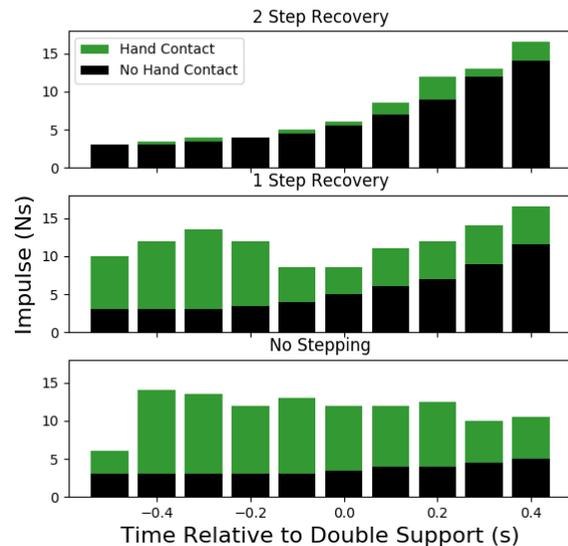

Fig. 4: Maximum impulse at different times in the walking gait that the robot was able to recover from for zero, one, and two step recoveries with and without hand contacts.

Additionally, it is important to note that perfect force tracking was not achieved. Fig. 5 shows the tracking of the external hand force in one of the experiments. This is important because it is very improbable that on a real humanoid perfect force tracking while making and breaking contacts with the environment will be achieved. The robustness of the approach presented comes from constantly replanning the contact force from the measured robot state.

Further demonstrations of the behavior of this control scheme is shown in the accompanying video [1]. The video shows the response to pushes with and without hand contact, choosing between surfaces with different friction properties, and utilizing contacts on surfaces that are rotated at different angles.
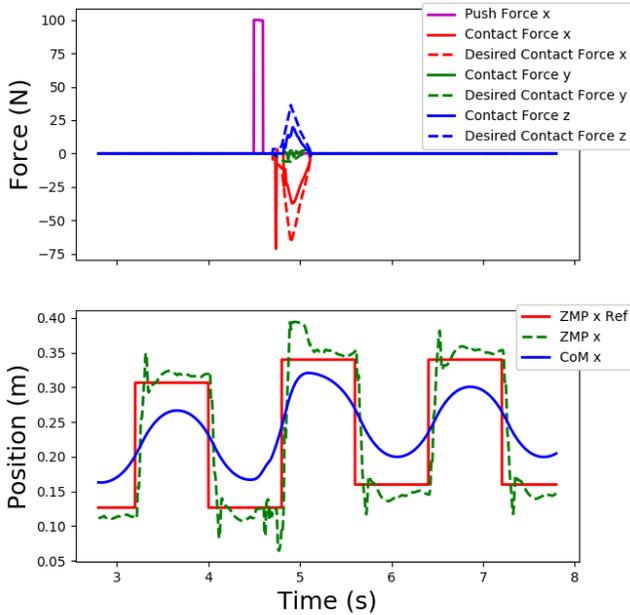
---

[1]http://bit.ly/masonIcra2018

Fig. 5: Top: External impulse applied to the robot (10 Ns) and the force tracking of the hand. Bottom: ZMP tracking and CoM trajectory as a result of the push. Despite imperfect force tracking, the robot is still able to recover from the impulse.

VI. CONCLUSION

In this work, we presented an extension to an MPC walking control scheme that reactively plans a desired contact location and force without any predetermined timing. In doing so, the formulation considers friction properties of the contact surfaces and the time it takes for the robot to move its hand to the contact location. To this end, we introduced a two part optimization scheme in which the first optimization plans the CoM and footsteps trajectories while the second optimization reasons about the hand contacts. The second optimization is initiated when the first optimization violates some conservative ZMP constraints. This separation into two optimization problems allowed us to reach real-time computation while maintaining low computational complexity. As shown in our experiments, there is a great benefit of using hand contact together with a walking controller when within a one recovery step proximity. Because this control scheme separates walking from hand contacts, there is the possibility to pair the second stage with different LIPM based walking pattern generators and other methods to trigger the need for hand contact. Potential directions for future research would be to use the DCM of the LIPM or incorporate an external force estimator [20] to trigger the need for hand contact.

REFERENCES

[1] 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," vol. 2, pp. 1620–1626 vol.2, Sept 2003.

[2] P. b. Wieber, "Trajectory free linear model predictive control for stable walking in the presence of strong perturbations," in *2006 6th IEEE-RAS International Conference on Humanoid Robots*, Dec 2006, pp. 137–142.

[3] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online Walking Motion Generation with Automatic Foot Step Placement," ser. Special Issue: Section Focused on Cutting Edge of Robotics in Japan 2010, vol. 24, no. 5-6. Taylor & Francis, 2010, pp. 719–737. [Online]. Available: https://hal.inria.fr/inria-00391408

[4] M. Khadiv, A. Herzog, S. A. a. Moosavian, and L. Righetti, "Step Timing Adjustment : A Step toward Generating Robust Gaits *," 2016.

[5] M. R. O. A. Maximo, C. H. C. Ribeiro, and R. J. M. Afonso, "Mixed-Integer Programming for Automatic Walking Step Duration," no. 3, pp. 5399–5404, 2016.

[6] S. Caron and Q. C. Pham, "When to make a step? tackling the timing problem in multi-contact locomotion by topp-mpc," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Nov 2017, pp. 522–528.

[7] C. Brasseur, A. Sherikov, C. Collette, D. Dimitrov, and P. B. Wieber, "A robust linear MPC approach to online generation of 3D biped walking motion," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Decem, pp. 595–601, 2015.

[8] J. Englsberger, T. Koolen, S. Bertrand, J. Pratt, C. Ott, and A. Albu-Schäffer, "Trajectory generation for continuous leg forces during double support and heel-to-toe shift based on divergent component of motion," pp. 4022–4029, Sept 2014.

[9] M. A. Hopkins, D. W. Hong, and A. Leonessa, "Humanoid locomotion on uneven terrain using the time-varying divergent component of motion," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, pp. 266–272, 2015.

[10] 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, Jan 2017.

[11] J. Carpentier, S. Tonneau, M. Naveau, O. Stasse, and N. Mansard, "A versatile and efficient pattern generator for generalized legged locomotion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3555–3561.

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

[13] D. Serra, C. Brasseur, A. Sherikov, D. Dimitrov, and P. B. Wieber, "A newton method with always feasible iterates for nonlinear model predictive control of walking in a multi-contact situation," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, Nov 2016, pp. 932–937.

[14] D. J. Agravante, A. Sherikov, P. B. Wieber, A. Cherubini, and A. Kheddar, "Walking pattern generators designed for physical collaboration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1573–1578.

[15] P. Castro and F. de Cincias, "Tightening piecewise mccormick relaxations through partition-dependent bounds for non-partitioned variables," 11 2014.

[16] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

[17] S. Feng, E. Whitman, X. Xinjilefu, and C. G. Atkeson, "Optimization based full body control for the atlas robot," *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-February, no. Id, pp. 120–127, 2015.

[18] 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, no. 3, pp. 473–491, Mar. 2016.

[19] R. Tedrake *et al.*, "A summary of team mit's approach to the virtual robotics challenge," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 2087–2087.

[20] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 556–563.