# Input-Output Nonlinear Model Predictive Control for Dynamic Stair Climbing

Grant A Gibson, Advisor: Jessy W Grizzle

## I. Introduction

Stair climbing (ascent or descent) is a task that most able-bodied humans complete with ease on a regular basis. While a variety of legged robots have demonstrated walking on flat terrain, stair climbing remains a challenge requiring a rigorous solution. Quadrupedal and wheeled configurations have been used to successfully navigate stairs, however these configurations are not robust to steep steps or narrow landing platforms [1],[2]. Bipedal leg configurations are a natural choice for these conditions as they emulate the anatomy for which stairs have been designed.

Stair climbing solutions have already been developed for bipedal robots based on periodic motions (infinite stairs) and static climbing solutions (so-called ZMP walkers) [3],[4],[5]. In addition to the deficiencies already noted, most of the current solutions are not based on active perception, but rather assume a known, perfect geometry and perform blind walking. Just as humans use their perception to alter swing leg trajectories based on stair height, width, depth, and asymmetry, a Model Predictive Control (MPC) based controller can be used to improve real-world performance on bipedal robots.

This work formulates and analyzes a Nonlinear MPC (NMPC) controller for a planar five-link biped robot as a foundation for formulating *environmentally aware* control policies. Periodic stair trajectories (ascent and descent) are computed offline, and the controller performs trajectory tracking. State, output, and control constraints are used to evaluate the controller's robustness to environmental and actuator disturbances within a specified prediction horizon.

Computational inefficiency and intractability have plagued these constraint based methods from wide-use in the past, however improvements to nonlinear program optimization and hardware have sparked a renewed interest in this field [6]. With added advancements in mapping and vision, we anticipate that similar controller methods will be very useful for navigating dynamic and diverse terrain.

### A. Background

ZMP-based controllers can generate provably safe trajectories, however dynamic walking controllers can outperform these methods when analyzing certain metrics. ZMP controllers compute torques by constraining the center of pressure of the robot to lie within its support polygon. For multi-legged configurations the solutions are easily calculated and non-unique, however as the number of leg contacts decreases, greater torque inputs and slower movements are needed in order to maintain stability. Dynamic walking controllers have no such support polygon constraint. This class of controllers is designed to utilize the energy of impact in order to stabilize walking gaits, typically resulting in more agile and energy efficient motions. For this reason we have implemented a dynamic walking controller.

Both error-based and model-based trajectory tracking controllers have found success in bipedal locomotion [7],[3],[8]. Error-based controllers like Proportional-Derivative (PD) control are reactive and can handle model uncertainties, while model-based controllers, such as feedback linearization, can achieve significantly lower tracking error but are prone to failure as model uncertainty dominates. Model Predictive control can give the best of both worlds as the model dynamics are used to predict future trajectories, while the current tracking error is computed in a cost which is minimized to generate control commands. For these reasons we use a predictive control method.

### B. Contributions

We propose an Input-Output Nonlinear Model Predictive Control (IO-NMPC) method for achieving stable periodic stair climbing motions. Previous works implement a form of input-output NMPC, however their optimization problem is formulated using the linear dynamics of the transformed system [9],[10]. Conversely, our method uses the original nonlinear dynamics in order to predict future states. Another NMPC method was used to implement ZMP-based constraints for stair climbing in [11]. However, our reference trajectories are generated using the Hybrid Zero Dynamics (HZD) framework for dynamic walking [3]. We also implement IO-NMPC on a hybrid system to show periodic convergence in lieu of impacts.

## II. Planar Five-Link Walker

A planar five-link biped robot (RABBIT) is used to analyze the NMPC controller performance for various trajectories (schematic shown in Figure 1). The stair climbing task imposes a hybrid system and so we must consider both continuous and discrete dynamics. The continuous dynamics of the robot are derived from the Euler-Lagrange equations (1) and are parametrized by floating base coordinates.

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=Bu+J_c^T\lambda \tag{1}$$

Where $q=[\bar{x},\bar{z},\psi,q_{1R},q_{2R},q_{1L},q_{2L}]^T$ are the generalized coordinates. Since this is an underactuated system, the control motor torques $u\in\mathbb{R}^4$ are only applied at the $q_i$ joints. We assume that the robot has point feet and therefore $\lambda\in\mathbb{R}^2$ is

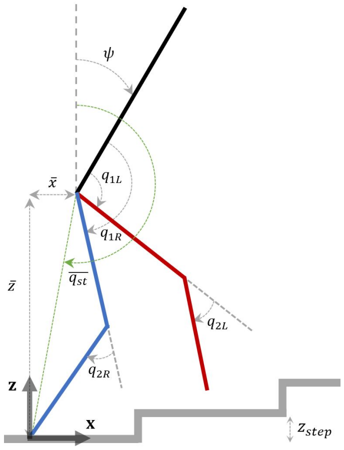

Fig. 1: Five-Link Walker (RABBIT) Schematic. The configuration is represented with floating base coordinates. The absolute stance leg angle $\bar{q}_{st}$ (shown in green) is used for phase-based control because its value monotonically changes.

composed of horizontal and vertical ground reaction forces (no moments).

An additional contact constraint is required for the continuous dynamics of the floating base model. This constraint sets the acceleration of the foot position to zero (2) and relates generalized coordinates and velocities to this quantity. $p_{stancefoot}$ is the position of the stance foot and $J_c$ is the foot contact jacobian.

$$\begin{aligned} 0 &= \ddot{p}_{stancefoot} \\ &= J_c\ddot{q} + \dot{J}_c\dot{q} \end{aligned} \tag{2}$$

Lastly, we describe the impact model (3) that is applied when the swing leg makes contact with the ground. The minus (-) and plus (+) symbols represent instantaneous quantities before and after impact, respectively. Note that the block matrix is always invertible for positive definite $D(q)$ and for full rank $J_{swingfoot}$.

$$\begin{bmatrix} \dot{q}^+ \\ \lambda^+ \end{bmatrix} = \begin{bmatrix} D(q^-) & -J^T_{swingfoot}(q^-) \\ J^T_{swingfoot}(q) & 0^{-1}_{2\times 2} \end{bmatrix}^{-1} \begin{bmatrix} D(q^-)\dot{q}^- \\ 0_{2\times 1} \end{bmatrix} \tag{3}$$

For this model to hold we make the following assumptions: (i) pure elastic collision, (ii) ground impact forces on the swing leg are treated as impulses, (iii) there is no slipping or rebounding at impact ($q^- = q^+$), and (iv) the former stance leg foot releases from the ground. These assumptions should be modified when operating on slippery or uneven terrain.

Lastly, we update the states after impact with a reset map. Because we are using a symmetric model, the corresponding thigh and knee joint coordinates can be switched.

$$x^+ = \begin{bmatrix} H_{reset}q^- \\ \dot{q}^+ \end{bmatrix} \tag{4}$$

$$\text{where } H_{reset} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 2} & 0_{3\times 2} \\ 0_{2\times 3} & 0_{2\times 2} & I_{2\times 2} \\ 0_{2\times 3} & I_{2\times 2} & 0_{2\times 2} \end{bmatrix}$$

## III. Trajectory/Gait Generation

There are many different stair climbing tasks (i.e. stopping on step, aperiodic stair climbing,etc.), however for the scope of this research we only analyze periodic climbing motions. These reference trajectories are generated using the Fast Robotics Optimization and Simulation Toolbox (FROST) [12]. This toolbox formulates a nonlinear program with direct collocation methods based on the Hybrid Zero Dynamics (HZD) framework [3],[8],[13].

The HZD principle guarantees low-dimensional periodic stability for prescribed virtual constraints that satisfy a hybrid invariance condition. For the case of the RABBIT model, the torso pitch angle $\psi$ and the four relative joint coordinates $q_i$ mus be equivalent at the beginning of the step and after impact and reset. For simplicity we have chosen the virtual constraints (5) to equal the thigh and knee joints (i.e. $y \in \mathbb{R}^4$). The desired trajectories $h^d$ are chosen as bezier polynomials (6) whose shape is governed by the coefficients $\alpha$. These polynomials exist for $s \in [0,1]$. $M$ refers to the degree of the polynomial and is user-defined.

$$y = h(x) - h^d(\alpha, s) \tag{5}$$

$$\text{where } h(x) = \begin{bmatrix} 0_{4\times 3} & I_{4\times 4} & 0_{4\times 7} \end{bmatrix} x$$

$$h^d(\alpha, s) = \sum_{k=0}^{M} \alpha_k^i \frac{M!}{k!(M-k)!} s^k (1-s)^{M-k} \tag{6}$$

When using virtual constraints it is important to delineate between phase-based and time-based desired trajectories. We use a phase-based implementation and thus $h^d$ is parametrized by the phase variable $s$. As previously stated, $s \in [0,1]$ and must be strictly monotonically increasing. The absolute stance leg angle $\bar{q}_{st}$ has been used previously for parametrization of $s$ [14] and so we define $s$ in the following equation (7). $\bar{q}_{st}^{begin}$ and $\bar{q}_{st}^{end}$ are constants and correspond to $\bar{q}_{st}$ at the beginning and end of each step. These constants can remain unchanged throughout trajectory or they can be updated after each impact. For simplicity we keep them constant.

$$s = \frac{\bar{q}_{st}(x) - \bar{q}_{st}^{begin}}{\bar{q}_{st}^{end} - \bar{q}_{st}^{begin}} \tag{7}$$

$$\text{where } \bar{q}_{st} = \psi + q_{1R} + \frac{1}{2} q_{2R}$$

## IV. Input-Output Linearization

Input-Output Linearization (IO-L) feedback control is an established control policy which has been used to successfully stabilize bipedal walking gaits [3]. The basic principle is that a control law, $u$, is chosen such that the zero dynamics of the system are linear and exponentially stable. The continuous dynamical system described by (1) and (2) can be written in a modified control affine form (8), where $x = [q, \dot{q}]^T$.

$$\begin{aligned} \dot{x} &= f(x) + g(x)u + j(x)\lambda \\ y &= h(x) - h^d(\alpha, s) \end{aligned} \tag{8}$$

By inspection, $y$ has relative degree 2 and therefore the virtual constraint acceleration ($\ddot{y}$) can be written as a function of the state, wrench, and phase ($s$). The control $u$ is computed (10) be cancelling all previous dynamics of $\ddot{y}$ and inserting user-defined linear dynamics, $v_L$. The equation can be further reduced by substituting in $\lambda$, which is uniquely found by combining (1) and (2). For simplicity, we use a PD control law (11) to achieve these stabilizing dynamics (LQR is another acceptable method and will give a similar output feedback law for the integrator system).

$$\ddot{y} = L_f^2h(x) + L_jL_fh(x)\lambda + L_gL_fh(x)u - \ddot{h}^d(s) \tag{9}$$

$$u = (L_gL_fh(x))^{-1}(v^L - L_f^2h(x) - L_jL_fh(x)\lambda - \ddot{h}^d(\alpha, s)) \tag{10}$$

$$v^L = -K_d\dot{y} - K_py \tag{11}$$

The lie derivative notation is used for brevity. An important observation is that the decoupling matrix ($L_gL_fh(x)$) is not well-conditioned for all choices of $h(x)$ and so the virtual constraints must be chosen carefully. As long as $K_p, K_d > 0$ the error dynamics will be stabilizing. A more structured approach for computes the gains as a function of desired settling time and damping ratio.

## V. Input-Output NMPC Feedback Control

The IO-NMPC optimization problem is described below in (12).

$$\begin{aligned}
\min_{\mathbf{Z}}\ J = \sum_{k=0}^{N-1}[(x_k - \hat{x}_k)^TQ(x_k - \hat{x}_k) + (u_k - \hat{u}_k)^TR(u_k - \hat{u}_k)] +& \\
\left\lVert \ddot{y}_k - v_k^L \right\rVert_2^2 + [y_N;\ \dot{y}_N]^TQ_{term}[y_N;\ \dot{y}_N]& \\
\textbf{subject to}& \\
x_{k+1} = x_k + \Delta T f(x_k, v_k)& \\
v_k = [L_gL_fh(x_k)]^{-1}[u_k + v^L{}_k - L_f^2h(x_k) - L_jL_fh(x)\lambda_k - \ddot{h}^d(\alpha_k, s_k)]& \\
\lambda_k = [J_cD^{-1}J_c^T]^{-1}[J_cD^{-1}(C\dot{q}_k + G - Bu_k) - \dot{J}_c\dot{q}_k]& \\
v_k^L \rightarrow \text{see Equation (11)}& \\
y_k, \dot{y}_k, \ddot{y}_k \rightarrow \text{see Equation (5,9)}& \\
\ddot{h}^d(\alpha_k, s_k) \rightarrow \text{see Equation (6,7)}& \\
x_0, x_k \in \mathbf{X},\ u_k \in \mathbf{U}&
\end{aligned} \tag{12}$$

Where $\mathbf{Z} = \{x_i, u_j, \lambda_j \ : \ i = 0,...,N, j = 0,...,N-1\}$ is the decision variable set, and $x_i$, $u_j$, and $\lambda_j$ are the state, control input, and wrench, respectively, over the prediction horizon, $N$. $Q \succcurlyeq 0$ is the running stage penalty, $R \succ 0$ is the control input penalty, and $Q_{term} \succcurlyeq 0$ is the terminal cost. $\hat{\mathbf{X}} = \{\hat{x}_i, \hat{u}_j, \alpha_i, s_i : \ i = 0,...,N, j = 0,...,N-1\}$ is the reference trajectory set defined prior to optimization. The state and control constraint sets $\mathbf{X}$ and $\mathbf{U}$ form convex polytopes via the union of box constraints. The construction of $\mathbf{U}$ in the current problem formulation is not intuitive, therefore a new constraint set $\mathbf{V}$ can be added which imposes torque saturation box constraints on $v_k$, while $\mathbf{U}$ remains large. As shown, the nonlinear dynamics are updated with a simple forward Euler integration method.

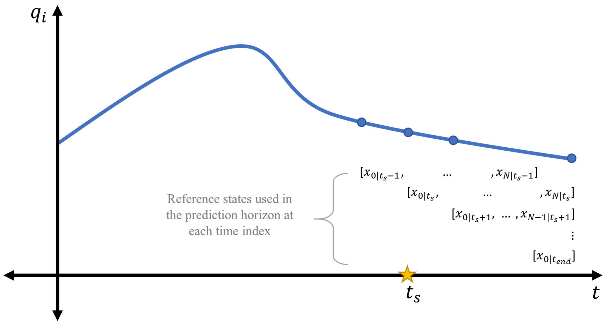
$q_i$
$[x_{0|t_s-1},\ \dots\ ,x_{N|t_s-1}]$
$[x_{0|t_s},\ \dots\ ,x_{N|t_s}]$
$[x_{0|t_s+1},\ \dots\ ,x_{N-1|t_s+1}]$
$\vdots$
$[x_{0|t_{end}}]$
Reference states used in the prediction horizon at each time index
$t_s$
$t$

Fig. 2: Shrinking Horizon Schematic. $t_s$ represents the shrinking horizon time. All subsequent prediction horizons will be shortened to make sure that the last state in the prediction horizon is the final reference state.

### A. Cost Function Intuition

The IO-L solution is equivalent to the unique point-wise solution of (13), subject to continuous dynamics shown in (1) and (2). The minimization in (12) can be viewed as an approximation of (13) over a desired prediction horizon subject to state and control constraints.

$$\begin{aligned}
u^*_{IO-L} = \arg\min_u \left\lVert \ddot{y} + K_d\dot{y} + K_py \right\rVert^2 \\
\textbf{subject to}\ \text{Equations}\ (1,2)
\end{aligned} \tag{13}$$

To guarantee stability and recursive feasibility of the optimization problem, a terminal cost is required to vanish as the reference error approaches zero when the current states and control inputs are admissible [15]. We give no verification of admissibility but it is a reasonable assumption that state and control values within a local region of the reference trajectory are admissible. The terminal penalty matrix $Q_{term}$ is set as the solution to the continuous-time algebraic Riccati equation by solving the infinite-horizon Linear Quadratic Regulator (LQR) problem for the IO-L output dynamics (with prespecified gains $K_p, K_d$).

### B. Shrinking Prediction Horizon Modification

A shrinking prediction horizon modification can be made to simplify formulation modifications or when there is uncertainty in the impact model. Rather than propagate unknown dynamics, the prediction horizon can be decreased relative to your previous knowledge of when the impact should occur. For simplicity we implement this shrinking horizon method (Figure 2). A more accurate formulation would forego the shrinking horizon to instead predict and switch dynamics based on the estimated time to impact. A more complex formulation of the disturbance preview problem discussed in the next problem could be used.

### C. Disturbance Preview Formulation

As previously mentioned, one motivation for this work is to incorporate perception information in the formulation of a stable walking and climbing controllers. Therefore, we reformulate (12) so that a new reference trajectory can be used mid-step without recompiling the solver. This new

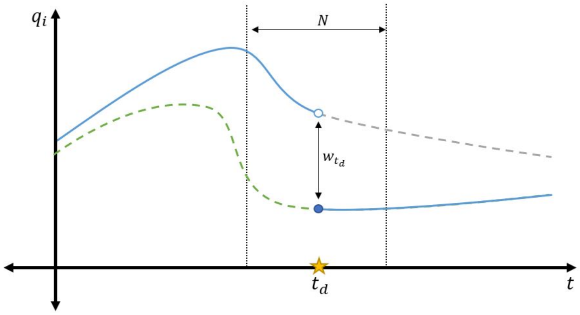

Fig. 3: New Reference Trajectory for Incorporating Disturbance Preview

task is a disturbance preview problem, with the disturbance defined below (14).

$$w_{t_d} = \tilde{x}_{t_d} - \hat{x}_{t_d} \tag{14}$$

Where $\hat{x}_{t_d}$ and $\tilde{x}_{t_d}$ are the states of previous and new reference trajectories. It is important to note, that even though the formulation developed below is for a stair height change, with slight modifications this formulation can be utilized to capture other disturbances such as impact. In this case, an additional impact update and reset map equality constraint must be inserted at the appropriate time step.

To simplify the problem, we assume that we know, a priori, the time at which a new trajectory is set, allowing us to include new reference errors in the prediction horizon. A simple diagram of this preview setup is depicted in Figure 3. As a design choice, the new reference trajectory is right continuous at the switching time $t_d$. The new desired state at time of reference switch is derived below (15).

$$\tilde{x}_{t_d} = \hat{x}_{t_d - 1} + \Delta T f + w_{t_d} \tag{15}$$

The disturbance preview problem can then be formulated with slight modifications to (12), by modifying the forward Euler integration matrix inequality constraints (16).

$$\begin{aligned}
&\min_{Z} \; J_N \\
&\textbf{subject to} \\
&\begin{bmatrix} x_0 - \hat{x}_0 \\ x_1 - (x_0 + \Delta T f(x_0, u_0)) \\ \vdots \\ x_{k_{t_d}} - (x_{k_{t_d}-1} + \Delta T f(x_{k_{t_d}-1}, u_{k_{t_d}-1})) \\ \vdots \\ x_N - (x_{N-1} + \Delta T f(x_{N-1}, u_{N-1})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ w_{t_d} \\ \vdots \\ 0 \end{bmatrix} \\
&\rightarrow \text{(Include previous constraints)}
\end{aligned} \tag{16}$$

## VI. Preliminary Results

A direct multi-shooting approach is used to solve (12) by increasing the sparsity, and thus, efficiency of the optimization solver. The problem is pre-processed using the Casadi algorithmic differentiation package and is solved with IPOPT [16],[17]. A variety of stair climbing trajectories are generated using FROST (0, 5, 7.5, and 10 cm, ascent and descent, with average step velocities of 0.5 and 0.75 m/s). An example animation is shown in Figure 4 for a 5 cm stair ascent with average velocity 0.50 m/s. The unconstrained, 0-step prediction horizon version of the IO-NMPC controller generates comparable periodic trajectories to the IO-L solutions. Figure 5 shows the the state position and control trajectories over a ten step sequence.

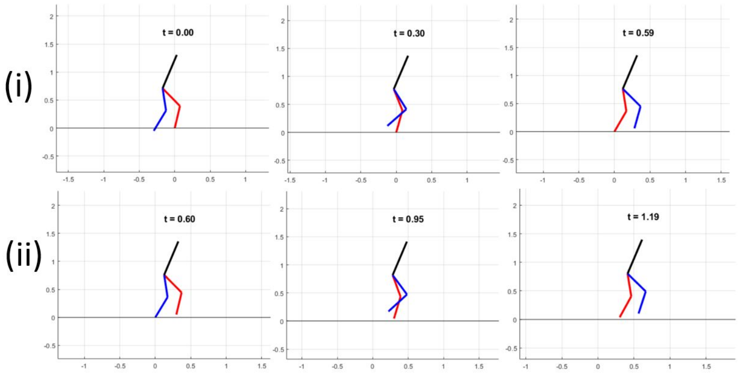

Fig. 4: 2-step animation result for stair ascent with IO-NMPC control. The first step is shown in the top row and the second step is shown in the second row. The stance leg (red) swaps after impact and becomes the swing leg (blue).

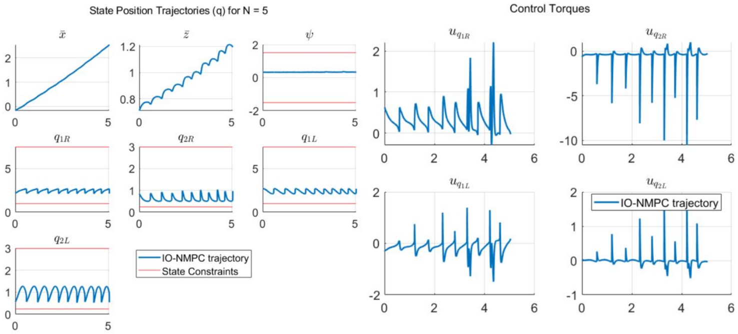

Fig. 5: State and control trajectories corresponding to the IO-NMPC solution of Figure 4. A 5 cm stair ascent, 0.50 m/s velocity reference trajectory is used to drive the phase-based control method.

In order to highlight NMPC capabilities we add output constraints (17).

$$\textbf{(1)} \left|\frac{\lambda_k^x}{\lambda_k^z}\right| \le \mu_s \qquad \textbf{(2)} \; |v_k| \le u_{max} \tag{17}$$

These *natural* constraints represent real world restrictions that would be enforced on the robot via surface conditions and hardware limitations. (17.1) represents the friction cone at the stance foot that must be below the coefficient of friction ($\mu_s$) to avoid slipping. (17.2) represents motor torque saturation limits. A solution trajectory for both IO-L and IO-NMPC with 1-step prediction is shown in Figure 6. As shown, the IO-NMPC controller satisfies ground reaction force constraints while the IO-L controller greatly violates them.

Swing foot height constraints are also simulated to test obstacle avoidance. The normalized height of the swing foot $z_{sw}$ is derived as a function of phase. Before each update the bounds of $z_{sw}$ are updated based on current and predicted values of phase (18). A plot of the results are shown in

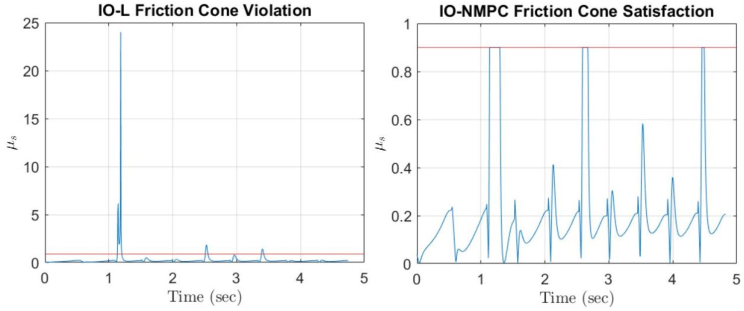

Fig. 6: Friction cone violation and satisfaction for the IO-L and IO-NMPC solution trajectories. A 10 cm ascent (with velocity = 0.75 m/s) reference trajectory was used and additional 10 N-m torque constraints were imposed. The static coefficient of friction is set to 0.9 which is comparable with rubber and concrete surfaces.

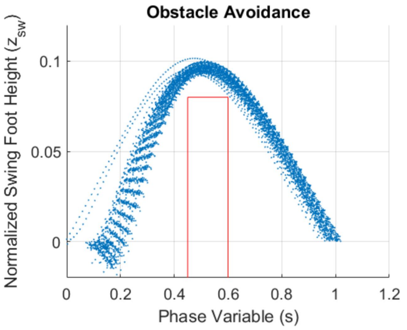

Fig. 7: Normalized swing foot height $z_{sw}$ avoiding obstacle highlighted by the red box

Figure 7 for flat ground walking

$$\textbf{if} \quad 0.45 \leq s_k \leq 0.6 \quad (k = 0, \ldots N-1) \\ z_{swk} \geq 0.08 \quad \textbf{end} \tag{18}$$

Additional disturbance rejection tests were also performed on both controllers by applying external forces at the hip position of RABBIT. The number of failures (fall over) and successes was approximately equal under the same conditions (i.e. PD gains, speed, stair height, and varied initial condition).

## VII. Discussion

When using unconstrained conditions, the two controllers performed equally well. However, the IO-NMPC was able to outperform IO-L after adding practical (friction and torque limits) and preferential (swing foot path) output constraints. Supplemental disturbance preview and perturbation rejection tests were performed however there were no significant differences in performance between the two controllers.

As currently simulated, the IO-NMPC controller is not built for on-line implementation. The IPOPT nonlinear program solver is bulky for a problem of this size and large prediction horizons become intractable.

At first glance, the current IO-NMPC controller might seem redundant and over-defined. However, from an optimization standpoint, the dynamics equality constraints are highly nonlinear and make the problem non-convex. We argue that the input-output structure of the control input improves feasibility and helps avoid undesirable local minima since IO-L is a solution to the unconstrained minimization problem. Future work will extend this method to more complex models in order to evaluate feasibility on real world robots.

## References

[1] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2245–2252.

[2] M. Eich, F. Grimminger, and F. Kirchner, "A versatile stair-climbing robot for search and rescue applications," in *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*. IEEE, 2008, pp. 35–40.

[3] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, and B. Morris, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2018.

[4] 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 *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2. IEEE, 2003, pp. 1620–1626.

[5] J.-Y. Kim, I.-W. Park, and J.-H. Oh, "Experimental realization of dynamic stair climbing and descending of biped humanoid robot, hubo," *International Journal of Humanoid Robotics*, vol. 6, no. 02, pp. 205–240, 2009.

[6] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[7] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4559–4566.

[8] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[9] X.-B. Kong, Y.-j. Chen, and X.-J. Liu, "Nonlinear model predictive control with input-output linearization," in *2012 24th Chinese Control and Decision Conference (CCDC)*. IEEE, 2012, pp. 688–693.

[10] M. Soroush and H. M. Soroush, "Input-output linearizing nonlinear model predictive control," *International Journal of Control*, vol. 68, no. 6, pp. 1449–1474, 1997.

[11] R. Heydari and M. Farrokhi, "Model predictive control for biped robots in climbing stairs," in *2014 22nd Iranian Conference on Electrical Engineering (ICEE)*. IEEE, 2014, pp. 1209–1214.

[12] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, "3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1447–1454.

[13] A. D. Ames, E. A. Cousineau, and M. J. Powell, "Dynamically stable bipedal robotic walking with nao via human-inspired hybrid zero dynamics," in *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, 2012, pp. 135–144.

[14] C. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. Westervelt, C. C. De Wit, and J. Grizzle, "Rabbit: A testbed for advanced control theory," 2003.

[15] L. Grüne and J. Pannek, "Nonlinear model predictive control," in *Nonlinear Model Predictive Control*. Springer, 2017.

[16] J. Andersson, J. Åkesson, and M. Diehl, "Casadi: A symbolic package for automatic differentiation and optimal control," in *Recent advances in algorithmic differentiation*. Springer, 2012, pp. 297–307.

[17] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[18] K. A. Hamed, J. Kim, and A. Pandala, "Quadrupedal locomotion via event-based predictive control and qp-based virtual constraints," *arXiv preprint arXiv:2004.06858*, 2020.

[19] X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, "From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking," *IEEE Access*, vol. 4, pp. 3469–3478, 2016.

[20] T. Koolen, T. De Boer, J. Rebula, A. Goswami, and J. Pratt, "Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models," *The international journal of robotics research*, vol. 31, no. 9, pp. 1094–1113, 2012.

[21] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.