2D Planar Hopping Robot Quadratic MPC

Misha Lvovsky

Abstract—The goal of this project is to implement an MPC (Model Predictive Controller) for a single legged hopping robot. The MPC will be formulated as a Quadratic Program (QP) and will be solved using an off-the-shelf solver in python. The setup of the robot is a square body with a single 1p leg and a reaction wheel that allows it to apply forces perpendicular to its leg.

I. INTRODUCTION

A. Historical Background

The control of simple hopping robots has been studied as early as 1984 when Marc Raibert built and subsequently studied the single legged Raibert Hopper [1]. The control of such robots is a reduced version of the control of legged robots with more than one leg, making it a popular field of study and testing grounds for new algorithms for legged robots.

MPCs have emerged as one of the best ways to control highly dynamic systems such as legged robots. MPCs started in other slower dynamic systems before they were used in legged robots. In fact, MPCs have been studied since the 60s when they were developed for use in the control of chemical processes such as this work by Chestnut et al. [2]. The use of MPCs for legged robots however is a more recent development that began as the processing power necessary to solve for the control inputs of a robot on-board became available, as well as the hardware necessary to make legged robots feasible started to develop. The first use of MPC to control legged robot motions online was done by Christine Azevedo in 2002 where she used an MPC to solve for control of a bipedal robot [3].

Since then this the topic has been studied in much more detail resulting in MPCs which are deployed on much faster and more dynamic robots.

B. Recent Work

More recently the use of MPCs has extended to the onboard online optimization of legged robot control sequences such as the one studied by Erez et al. [4]. These MPCs have ranged from highly nonlinear MPCs that simultaneously optimize the landing locations, times, and reaction forces, to simpler methods which eschew the more complex aspects of the problem often converting it to a convex optimization problem which can be solved at a high frequency and thus provide highly relevant trajectories that are updated to the robot's current state.

An additional direction of recent development has been in the use of various techniques to leverage prior data to speed up or stabilize the optimization process. One such example is the use of learned function approximators investigated by Melon et al. to give a warm starting guess to speed up the optimization convergence [5]. Another example is the use of heuristics to make the non-convex problem closer to a convex one preventing convergence in local minima and thus making the optimization far more robust such as the method investigated by Bledt and Kim [6].

A second direction of recent development is in the use of contact implicit solvers to merge the optimization of contact and control at the same time, allowing the discovery of more optimal control inputs and position trajectories. An example of an investigation in this direction is the one conducted by Le Cleac and Howell et al. [7] which uses complementarity to find contact implicit trajectories. Another is one led by Toussaint et al. [8] which uses a sequence of constraints to force the desired behavior.

II. BACKGROUND

MPC problems are optimization problems of the standard form that optimize a cost function $J(x)$ and given constraint functions $g(x)$ and $h(x)$ find the x that minimizes $J(x)$ subject to the constraint that $g(x) \leq 0$ and $g(x) = 0$.

$$x \qquad\qquad\qquad \text{Optimization Variable}$$
$$J \qquad\qquad\qquad \text{Cost Function}$$
$$g \text{ and } h \qquad\qquad\qquad \text{Constraint Functions}$$
$$x^* = \underset{x}{\text{argmin}} \ J(x)$$
$$\text{S.T.}$$
$$g(x) \leq 0$$
$$h(x) = 0$$

When this form is used for MPC the cost function is formulated such that some portion of x can be used to compute the optimal control input, and the constraints keep that control feasible. There exist a variety of ways to formulate such a cost function.

Naunert et al. [9] specify a collocation constraint using realistic nonlinear robot dynamics and a cost function of an arbitrary form $L(x, u, t)$ integrated over the trajectory as well as a cost on the final position $h(x_f)$. Posa et al. [10] and Le Cleac and Howell et al. [7] take a similar approach but use complementarity to implicitly solve for contacts and reaction forces simultaneously. The complementarity comes from the requirement that either contact distances are zero or contact forces are zero.

The works that this project is adapting to the planar 2D monopod case are the work of of Kim et al. [11] and Di Carlo et al. [12] on optimizing reaction forces for the control of fast legged robots. These works further distill the problem down from the other formulations mentioned to reduce all the

way to a QP optimization problem subject to linear constraints that can be solved extremely quickly on-board a small mobile robot. By using an abstraction which captures enough of the robot's dynamics to generate useful control trajectories in a form that is easy for optimization engines to solve, these formulations are able to update extremely quickly and thus reject errors more effectively.

III. METHODOLOGY

To begin formulating the optimization problem we will stack the optimization variables into vectors

$$x_t = \begin{bmatrix} \theta_t \ \vec{r}_t^T \ \dot{\theta}_t \ \dot{\vec{r}}_t^T \end{bmatrix}^T$$

$$u_t = \begin{bmatrix} f_x \ f_y \end{bmatrix}^T$$

where $\vec{r}_t$ is the position of the robot's body at time t, θ_t is the angle, and f_x and f_y are the reaction forces from the ground.
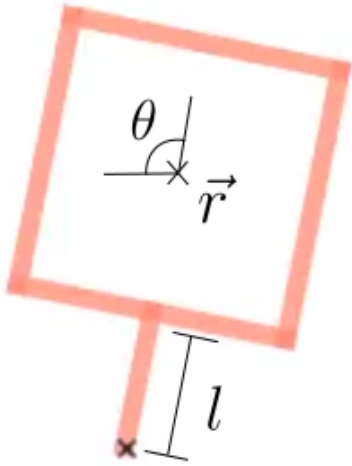

Fig. 1. This is the 2D planar biped being controlled. This figure shows the θ, $\vec{r}$, and l state elements on the system.

Now we can begin to make simplifications to the dynamics of the robot in order to get it into the right form for QP.

$$\frac{d}{dt}x = \begin{bmatrix} \mathbf{I} & \mathbf{I}dt \\ \mathbf{0} & \mathbf{I} \end{bmatrix} x + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ J_t \end{bmatrix} u = Ax + Bu$$

Where $\mathbf{M}$ is the mass matrix and A and B are calculated from $\mathbf{M}$ and dt. This simplification ignores the coriolis effects, assuming that in the absence of control force there will be no acceleration whatsoever. This is not correct in our case since our system has coriolis forces linking the rotation of the body to the length of the leg. However, this model is a good approximation when the speed is relatively low and the leg is short.

The friction constraint enforces that $\|f_x\| \leq \mu f_y$. The contact constraint enforces that $f_y > \epsilon$. Where ϵ is some

small value to ensure that the foot remains planted on the ground. Expressing this as a matrix for the QP solver gives the following.

$$\begin{bmatrix} -1.0 & -\mu \\ 1.0 & -\mu \\ 0 & -1.0 \end{bmatrix} u \leq \begin{bmatrix} 0.0 \\ 0.0 \\ -\epsilon \end{bmatrix}$$

Now to simplify the QP we will further coalesce these representations into one which is easily representable by a few matrix products. First we will reformulate the dynamics in terms of two matrices $\mathbf{A}_{qp}$ and $\mathbf{B}_{qp}$ which have the form:

$$\mathbf{A}_{qp} = \begin{bmatrix} \mathbf{I} \ A^T \ A^{2T} \cdots A^{nT} \end{bmatrix}$$

$$\mathbf{B}_{qp} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ b_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ Ab_0 & b_1 & 0 & \cdots & 0 \\ A^2b_0 & Ab_1 & b_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A^nb_0 & A^{n-1}b_1 & A^{n-2}b_2 & \cdots & b_n \end{bmatrix}$$

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_0^T \ \mathbf{x}_1^T \cdots \mathbf{x}_n^T \end{bmatrix}^T$$

Given these matrices we can now reformulate the dynamics as follows.

$$\mathbf{X} = \mathbf{A}_{qp}\mathbf{x}_0 + \mathbf{B}U$$

Now that we have simplified the dynamics we can forumlate cost function.

$$\|\mathbf{A}_{qp}\mathbf{x}_0 + \mathbf{B}_{qp}\mathbf{U} - \mathbf{x}_{\text{ref}}\|_{\mathbf{L}} + \|U\|_{\mathbf{K}}$$

Where $\|P\|_Q$ represents $Q^T P Q$. Rearranging this into standard QP form gives:

$$\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^T\mathbf{H}\mathbf{U} + \mathbf{U}^T g \min_{\mathbf{U}} \quad \frac{1}{2}\mathbf{U}^T\mathbf{H}\mathbf{U} + \mathbf{U}^T g$$
$$\text{s. t.} \quad \mathbf{C}\mathbf{U} \leq \bar{c}$$

Where $\mathbf{C}$ is defined as the constraint matrix defined earlier expanded to all time steps

$$\mathbf{C} = \begin{bmatrix} -1.0 & -\mu \\ 1.0 & -\mu \\ 0 & -1.0 \end{bmatrix}^{\otimes n}$$

$$\bar{c} = \begin{bmatrix} 0.0 \\ 0.0 \\ -\epsilon \end{bmatrix}^{\otimes n}$$

Optimizing this quadratic objective and linear constraint results in a set of control inputs that make the robot track the reference trajectory.
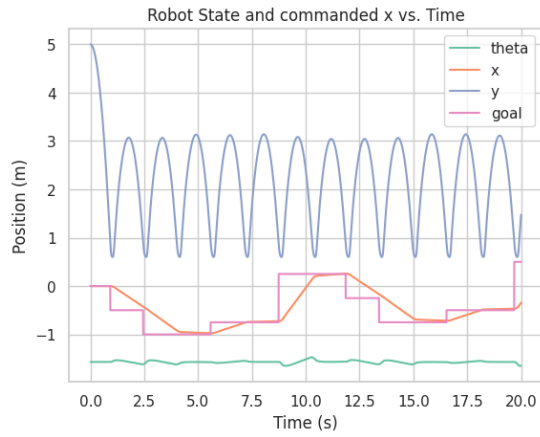
Fig. 2. The x axis of this plot corresponds to the time of the state that is plotted on the y axis. The y axis is the value of the variable in question, which is different for each colored line. The blue line shows the y height of the robot, the orange shows the x position, green shows the theta at that time, and red shows the target x pos for the robot to end up in.
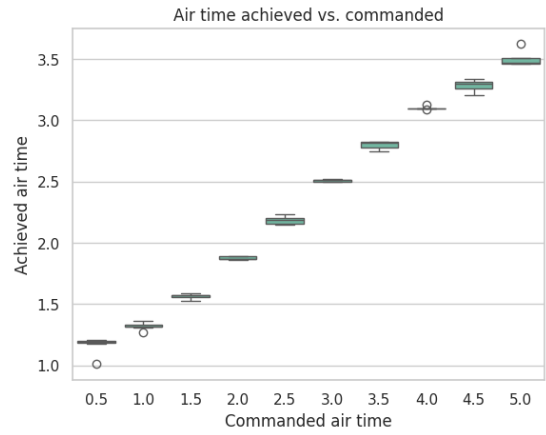

Fig. 3. The x axis of this plot corresponds to the commanded air time. Each column contains a box plot but they are very dense because the variation in height is very small. The y axis is the air time that the robot actually achieved in simulation.

IV. RESULTS

The first set of results is from a simulated experiment of running the robot attempting to make a few different size hops one after another. The results of this experiment are in figure 2. From the results of this plot we can see that based on the convergence of the x position to the target we have a successful controller. The x position command changes and then we can see that the robot arrives at that position after one bounce.

An additional metric by which the performance of this controller can be measured is by the success with which the controller adhered to the commanded air time. In figure 2 I used an air time of 1.5 seconds for all data since this seems to be the ideal point for the studied MPC. Figure 3 shows the distribution of the achieved air times for a given commanded air time. The reason for the inaccuracy to the trajectory at the higher end is due to the simplification made to not use effects dependent on velocity. Once a large enough velocity is commanded these effects become relevant and accuracy is lost.

V. CONCLUSION

This project successfuly adapter the trajectory optimization algorithm to the 2D monopod system and controlled it to follow some reference trajectories. This success despite the limitations of the simplicity of the model and the sub-optimality of the trajectory generator highlight the positive properties of MPC control, namely that because of the predictive control it is reasonably robust to dynamics immprecision.

The challenges enountered were primarily in generating feasible reference trajectories that lead to feasible contact sequences.

An interesting area of future investigation would be improvement upon the achieved air times either by switching to some variety of nonlinear model or improving the formulation of the QP model. Another interesting area of

future investigation would be to try various more advanced trajectory generators than the simple heuristic one used for this investigation.

In conclusion, this project was great for getting more familiar with the MPC algorithm. Though I was aware of this algorithm prior to doing this assignment I still found myself referncing various papers often. I learned a lot throught the process of implementing this final project and am happy with the reasonably successful result.

REFERENCES

[1] M. H. Raibert, H. B. Brown Jr., and M. Chepponis, "Experiments in Balance with a 3D One-Legged Hopping Machine." *International Journal of Robotics Research*, vol. 3, no. 2, p. 75, 1984. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true&AuthType=ip, sso&db=bth&AN=4707675&site=eds-live&scope=site&custid=umaah

[2] H. Chestnut, W. E. Sollecito, and P. H. Troutman, "Predictive-control system application," *Transactions of the American Institute of Electrical Engineers, Part II: Applications and Industry*, vol. 80, no. 3, pp. 128–139, Jul. 1961. [Online]. Available: https://ieeexplore.ieee.org/document/6371731/

[3] C. Azevedo, P. Poignet, and B. Espiau, "ON LINE OPTIMAL CONTROL FOR BIPED ROBOTS," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 199–204, 2002. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1474667015392661

[4] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. Atlanta, GA: IEEE, Oct. 2013, pp. 292–299. [Online]. Available: http://ieeexplore.ieee.org/document/7029990/

[5] O. Melon, R. Orsolino, D. Surovik, M. Geisert, I. Havoutis, and M. Fallon, "Receding-Horizon Perceptive Trajectory Optimization for Dynamic Legged Locomotion with Learned Initialization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE, May 2021, pp. 9805–9811. [Online]. Available: https://ieeexplore.ieee.org/document/9560794/

[6] G. Bledt and S. Kim, "Extracting Legged Locomotion Heuristics with Regularized Predictive Control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France: IEEE, May 2020, pp. 406–412. [Online]. Available: https://ieeexplore.ieee.org/document/9197488/

[7] S. L. Cleac'h, T. Howell, S. Yang, C.-Y. Lee, J. Zhang, A. Bishop, M. Schwager, and Z. Manchester, "Fast Contact-Implicit Model-Predictive Control," Jan. 2023, arXiv:2107.05616 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2107.05616

[8] M. Toussaint, J. Harris, J.-S. Ha, D. Driess, and W. Hönig, "Sequence-of-Constraints MPC: Reactive Timing-Optimal Control of Sequential Manipulation," Sep. 2022, arXiv:2203.05390 [cs]. [Online]. Available: http://arxiv.org/abs/2203.05390

[9] 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: https://ieeexplore.ieee.org/document/8276298/

[10] M. Posa, C. Cantu, and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, Jan. 2014, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/0278364913506757

[11] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control," Sep. 2019, arXiv:1909.06586 [cs]. [Online]. Available: http://arxiv.org/abs/1909.06586

[12] 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)*. Madrid: IEEE, Oct. 2018, pp. 1–9. [Online]. Available: https://ieeexplore.ieee.org/document/8594448/