Bio-inspired and Energy-efficient Convex Model Predictive Control for a Quadruped Robot

1st Aleksei Shamraev
Faculty of Control Systems and Robotics
ITMO University
St. Petersburg, Russia
adshamraev@itmo.ru

2nd Sergey Kolyubin
Faculty of Control Systems and Robotics
ITMO University
St. Petersburg, Russia
s.kolyubin@itmo.ru

***Abstract*—Animal running has been studied for a long time, but until now robots cannot repeat the same movements with energy efficiency close to animals. There are many controllers for controlling the movement of four-legged robots. The most popular is the Convex MPC. This paper presents a bioinspirational approach to increasing the energy efficiency of the state-of-the-art Convex MPC controller. This approach is to set a reference trajectory for the Convex MPC in the form of a SLIP model. This model describes the movements of animals when running. Adding a SLIP trajectory increases the energy efficiency of the Pronk gait by 15 percent at a speed of 1 m/s.**

***Index Terms*—quadruped, model predictive control, spring-loaded inverted pendulum, bioinspiration, energy efficiency**

I. Introduction

The field in the management of four-legged robots is actively developing. Starting with Raibert [1], there is a desire to imitate four-legged animals in movement, and in particular to provide dynamic walking.

Raibert made dynamic walking possible thanks to the special design of the jumping robot, whose movement was similar to the model of a spring inverted pendulum (SLIP), and very simple control. However, the robot had low maneuverability, robustness, speed, and limited reproduction of gait types.

To solve these problems, more complex designs have been developed for working with electric drives: Cheetah 3, Mini Cheetah, ANYmal. But along with the capabilities, the complexity in managing four-legged robots has also increased.

The main goal in controlling four-legged robots (Fig. 1) is to calculate the positions of the paws p = (x, y, z) and the reaction forces of the support f at the moment of contact to achieve a given type of movement (walking, trot, gallop) at a given speed.

For static walking (there are always at least 3 legs in contact), it is enough to solve the optimization problem once, without predicting the behavior of the robot at the next moment in time. Commonly, these controllers are based on the solution of the Quadratic Programming problem. For instance, Whole Body Control [3].

For dynamic walking, it becomes necessary to predict the behavior of the robot several steps ahead. These controllers are based on Model Predictive Control (MPC).

One of the most popular controllers for controlling the movement of four-legged robots is the Convex Model Predictive Control (cMPC) [4], which is the central algorithm of many MIT works. In this controller, an optimization problem is formulated with a simplified linear model of the robot (Linear Rigid Body). The control inputs are the ground reaction forces set by each foot. The positions of the paws are set in advance by a simplified algorithm using the Raibert heuristic.

In Regulated Predictive Control (RPC) [5] MPC calculates not only the ground reaction forces but also the positions of the paws. Because of this, MPC is formulated as nonlinear and therefore a set of local minima appears, which significantly complicates calculations. To reduce the number of local minima, regularization with heuristics is added.

RPC was implemented on Cheetah 3 [6], and a framework for finding heuristics was also developed [7]. More details can be found in the Phd thesis [8].

All these controllers operate at a frequency of about 40 Hz, which is not enough for locomotion at high speeds. For example, when moving at a gallop, the controller calculates the reaction forces of the support only 4 times per contact.

To increase stability, a Whole Body Impulse Control (WBIC) has been developed [9], which operates at a frequency of 500 Hz. The essence of the algorithm is the relaxation of the found MPC ground reaction forces. This controller allowed to increase the maximum speed of the quadruped robot Mini Cheetah from 2.45 m/s to 3.7 m/s.

The paper [10] presents an algorithm based on RPC and WBIC with the addition of computer vision for detecting and avoiding obstacles.

Computer vision is also added [11] using cMPC and WBIC. And as a development, Depth-based Impulse Control (DIC) appears [12].

The paper [13] provides an algorithm for generating the type of gaits adaptively, rather than rigidly setting timings as before.

MPC is resource-demanding and usually slow. In [14], the author replaces MPC with a neural network, training it based on MPC, which allows running the controller on robots with an on-board computer with small computing capabilities.

Animal imitation is one of the main incentives in legged robotics. Today, there are works on the imitation of animals by robots using neural networks [15].

There is a well-known template describing the running of animals, Spring-Loaded Inverted Pendulum (SLIP). This template is still widely used today. For example, in [16] the author uses a SLIP model to train a complex bipedal robot Cassie to walk and run. In [17], a simple SLIP model and a Full-model are combined to reduce the MPC calculation time.

This work focuses on setting the trajectory for the Convex MPC [4] as a SLIP model so that the robot moves similarly to real animals. This will increase energy efficiency with locomotion.

In the original controller, z of CoM is constant, and the z speed is zero. The controller stabilizes the height of the robot upon contact, which is a braking element when running.

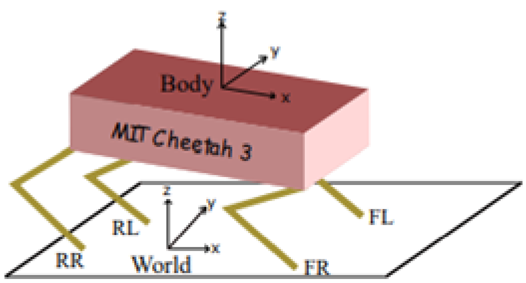

Fig. 1. Model of a quadruped robot [2].

II. Model Predictive Control

Let's briefly describe the controller [4]. The model in the MPC controller [4] is a simplified, linear Space Potato.

The robot's orientation is expressed as a vector of Z-Y-X Euler angles [18] $\Theta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{\boldsymbol{T}}$ where ψ is the yaw, θ is the pitch, and ϕ is the roll.

The robot's position is expressed as a $\mathbf{p} \in \mathbb{R}^3$. The control inputs are the ground reaction forces $\mathbf{f_i}$. For each ground reaction force $\mathbf{f_i} \in \mathbb{R}^3$, the vector from the center of mass (COM) to the point where the force is applied is $\mathbf{r_i} \in \mathbb{R}^3$.

${}_{\mathcal{G}}\boldsymbol{I}$ and ${}_{\mathcal{B}}\boldsymbol{I}$ are the inertia tensor seen from the global and local (body) frame, respectively, such as

$$
{}_{\mathcal{G}}\boldsymbol{I} \approx \boldsymbol{R}_z(\psi)\, {}_{\mathcal{B}}\boldsymbol{I} \boldsymbol{R}_z(\psi)^{\top}, \tag{1}
$$

where $\boldsymbol{R}_z(\psi)$ is a rotation matrix translating angular velocity in the global frame.

The discrete dynamics of the system can be expressed as

$$
\mathbf{x}(k+1) = \boldsymbol{A}_k \mathbf{x}(k) + \boldsymbol{B}_k \hat{\mathbf{f}}(k) + \hat{\mathbf{g}} \tag{2}
$$

where,

$$
\begin{aligned}
\mathbf{x} &= \begin{bmatrix} \boldsymbol{\Theta}^{\top} & \mathbf{p}^{\top} & \boldsymbol{\omega}^{\top} & \dot{\mathbf{p}}^{\top} \end{bmatrix}^{\top} \\
\hat{\mathbf{f}} &= \begin{bmatrix} \mathbf{f}_1 & \cdots & \mathbf{f}_n \end{bmatrix}^{\top}, \\
\hat{\mathbf{g}} &= \begin{bmatrix} \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & \mathbf{g}^{\top} \end{bmatrix}^{\top},
\end{aligned}
$$

$$
\boldsymbol{A} = \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 3} & \boldsymbol{R}_z(\psi_k)\Delta t & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{1}_{3\times 3}\Delta t \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{1}_{3\times 3} \end{bmatrix},
$$

$$
\boldsymbol{B} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \cdots & \mathbf{0}_{3\times 3} \\ {}_{\mathcal{G}}\boldsymbol{I}^{-1}[\mathbf{r}_1]_{\times}\Delta t & \cdots & {}_{\mathcal{G}}\boldsymbol{I}^{-1}[\mathbf{r}_n]_{\times}\Delta t \\ \mathbf{1}_{3\times 3}\Delta t/m & \cdots & \mathbf{1}_{3\times 3}\Delta t/m \end{bmatrix}
$$

The MPC formulated as a QP, which minimizes

$$
\min_{\mathbf{x},\mathbf{f}} \sum_{k=0}^{m} \left\| \mathbf{x}(k+1) - \mathbf{x}^{\text{ref}}(k+1) \right\|_{\boldsymbol{Q}} + \|\mathbf{f}(k)\|_{\boldsymbol{R}} \tag{3}
$$

subject to dynamics and initial condition constraints.

In the original, $\mathbf{x}^{\text{ref}}$ is set by the velocity in x, y, rate roll, and constant z. In this work, the trajectory is formulated by z, $\dot{z}$ from the SLIP model.

III. Spring-Loaded Inverted Pendulum Trajectory

At the abstract level, the robot body can be represented as a SLIP model (Fig. 2) with mass m, spring stiffness k, spring length l, the resting length of the spring l_0.

Then COM will be described by the SLIP model as

$$
\begin{bmatrix} m\ddot{l} \\ ml^2\ddot{\theta} \end{bmatrix} = \begin{bmatrix} ml\dot{\theta}^2 - k(l - l_0) - mg\cos\theta \\ -2ml\dot{\theta} + mgl\sin\theta \end{bmatrix} \tag{4}
$$

Since in the cMPC controller the greatest braking effect when running occurs along the z axis, let's take only this part of (4).

Let's take into account that the angle θ when running is not so large, so it can be ignored and take $l = z$.

Then the dynamics along the z axis will have the form

$$
m\ddot{z} = -mg + k(z_0 - z) \tag{5}
$$

We can formulate the dynamic (5) into a state space equation as,

$$
\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} \left(z_0 - \frac{g}{\omega_z^2} \right), \tag{6}
$$

where the states of CoM in vertical direction are defined as $Z = [z, \dot{z}]^T$, and we define $\omega_z = \sqrt{\frac{k}{m}}$. Further, we can denote $z_0 - \frac{g}{\omega_z^2}$ as u_z, and (5) can be written into a linear state space equation as,

$$
\dot{Z} = \begin{bmatrix} 0 & 1 \\ -\omega_z^2 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 \\ \omega_z^2 \end{bmatrix} u_z \tag{7}
$$

We discretize (7) with sampling time Δt and can obtain

$$
Z_{k+1} = A_z(\Delta t) Z_k + B_z(\Delta t) u_{z,k} \tag{8}
$$

where,

$$
A_z = \begin{bmatrix} \cos(\omega_z \Delta t) & \sin(\omega_z \Delta t)/\omega \\ -\omega_z \cdot \sin(\omega_z \Delta t) & \cos(\omega_z \Delta t) \end{bmatrix}
$$

$$
B_z = \begin{bmatrix} 1 - \cos(\omega_z \Delta t) \\ \omega_z \cdot \sin(\omega_z \Delta t) \end{bmatrix}
$$

Initial conditions Z_0 is taken at each contact with the surface during the jump.

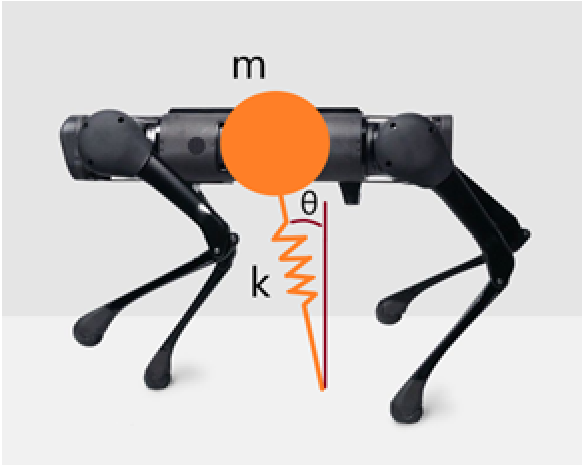

Fig. 2. The Spring-Loaded Inverted Pendulum (SLIP) model.

IV. Results

This approach was tested in the Pybullet simulator with a model of a four-legged Unitree A1 robot with a pronk gait. The stiffness was taken k = 4000, the mass m = 11 kg.

Fig. 3 shows the results without SLIP trajectory. With pronk gait at a speed of 1m/s, the energy efficiency value COT = 1.27.

Fig. 4 shows the results with SLIP trajectory. With pronk gait at a speed of 1m/s, the energy efficiency value COT = 1.10. The peak ground reaction forces decreased from 17 to 15.

The desired trajectory is not fully tracked, but it still gives an increase in energy efficiency of 15 percent.

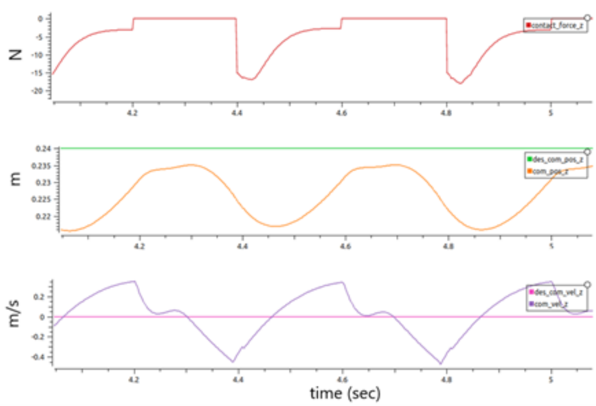

Fig. 3. Results of pronk running at a speed 1 m/s without SLIP trajectory in vertical directory.

V. Conclusion and Discussion

Despite the fact that MPC does not fully track the trajectory in vertical direction, the energy efficiency of running has increased by 15 percent.

Since the dynamics in the vertical direction is very fast, we can try to set this trajectory in the MPC itself on the prediction horizon. This can increase the accuracy of tracking the trajectory.

In future work, we plan to experiment with other gaits and implement this approach on the Unitree A1 robot.

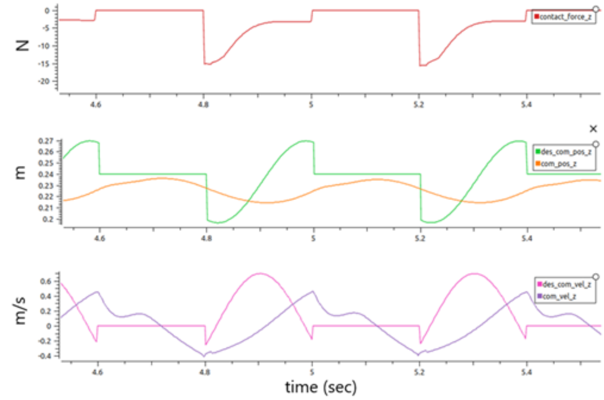

Fig. 4. Results of pronk running at a speed 1 m/s with SLIP trajectory in vertical directory.

Acknowledgment

We would like to thank SBER Robotics lab for their aid and support in our studies.

References

[1] Marc R. Legged Robots That Balance. The MIT Press, 1986. 205 pp.

[2] Di Carlo J. Software and control design for the MIT Cheetah quadruped robots. Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2020. 101 pp.

[3] Kim D., Lee J., Ahn J., Campbell O., Hwang H., Sentis L. Computationally-Robust and Efficient Prioritized Whole-Body Controller with Contact Constraints // IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2018. pp. 5987-5994.

[4] Di Carlo, J.,Wensing, P. M., Katz, B., Bledt, G., Kim, S. Dynamic locomotion in the MIT cheetah 3 through convex model-predictive control // IEEE International Conference on Intelligent Robots and Systems, 2018. pp. 7440-7447.

[5] Bledt, G., Wensing, P. M., Kim, S. Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the MIT cheetah // IEEE International Conference on Intelligent Robots and Systems, 2017. pp. 4102-4109.

[6] Bledt, G., Kim, S. Implementing regularized predictive control for simultaneous real-time footstep and ground reaction force optimization // IEEE International Conference on Intelligent Robots and Systems, 2019. pp. 6316-6323.

[7] Bledt, G., Kim, S. Extracting legged locomotion heuristics with regularized predictive control // IEEE International Conference on Robotics and Automation, 2020. pp. 406-412.

[8] Bledt G. Regularized predictive control framework for robust dynamic legged locomotion. Massachusetts Institute of Technology, 2020. 160 pp.

[9] 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 // arXiv, 2019. DOI: 10.48550/arXiv.1909.06586.

[10] Dudzik, T., Chignoli, M., Bledt, G. Kim, S. Robust autonomous navigation of a small-scale quadruped robot in real-world environments // IEEE International Conference on Intelligent Robots and Systems, 2020. pp. 3664-3671.

[11] Kim, D., Carballo, D., Di Carlo, J., Katz, B., Bledt, G., Lim, B., Kim, S. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot // IEEE International Conference on Robotics and Automation, 2020. pp. 2464-2470.

[12] Gabriel B. Margolis, Tao Chen. Learning to Jump from Pixels, 2021. DOI: 10.48550/arXiv.2110.15344.

[13] Boussema, C., Powell, M. J., Bledt, G., Ijspeert, A. J., Wensing, P. M., Kim, S. Online gait transitions and disturbance recovery for legged robots via the feasible impulse set // IEEE Robotics and Automation Letters, 2019. pp. 1611-1618.

[14] Carius J., Farshidian F., Hutter M. MPC-Net: A First Principles Guided Policy Search // IEEE Robotics and Automation Letters, 2020. pp. 2897-2904.

[15] Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, Sergey Levine. Learning Agile Robotic Locomotion Skills by Imitating Animals // arxiv, 2020. DOI: 10.48550/arXiv.2004.00784.

[16] Kevin Green, Yesh Godse, Jeremy Dao, Ross L. Hatton, Alan Fern, and Jonathan Hurst. Learning Spring Mass Locomotion: Guiding Policies with a Reduced-Order Model // IEEE Robotics and Automation Letters, 2021. pp. 3926 - 3932.

[17] He Li, Robert J. Frei, Patrick M. Wensing. Model Hierarchy Predictive Control of Robotic Systems // IEEE Robotics and Automation Letters, 2021. pp. 3373-3380.

[18] J. Craig, Introduction to Robotics: Mechanics and Control, ser. Addison-Wesley series in electrical and computer engineering: control engineering. Pearson/Prentice Hall, 2005.

[19] Bledt, G., Powell, M. J., Katz, B., Di Carlo, J., Wensing, P. M., Kim, S. MIT cheetah 3: Design and control of a robust, dynamic quadruped robot // IEEE International Conference on Intelligent Robots and Systems, 2018. pp. 2245-2252.