

# 《Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control》

## 阅读笔记

2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)  
Madrid, Spain, October 1-5, 2018

### Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control

Jared Di Carlo<sup>1</sup>, Patrick M. Wensing<sup>2</sup>, Benjamin Katz<sup>3</sup>, Gerardo Blede<sup>1,3</sup>, and Sangbae Kim<sup>3</sup>

**Abstract**—This paper presents an implementation of model predictive control (MPC) to determine ground reaction forces for a torque-controlled quadruped robot. The robot dynamics are simplified to formulate the problem as convex optimization while still capturing the full 3D nature of the system. With the simplified model, ground reaction force planning problems are formulated for prediction horizons of up to 0.5 seconds, and are solved to optimality in under 1 ms at a rate of 20-30 Hz. Despite using a simplified model, the robot is capable of robust locomotion at a variety of speeds. Experimental results demonstrate control of gaits including stand, trot, flying-trot, pronk, bound, pace, a 3-legged gait, and a full 3D gallop. The robot achieved forward speeds of up to 3 m/s, lateral speeds up to 1 m/s, and angular speeds up to 180 deg/sec. Our approach is general enough to perform all these behaviors with the same set of gains and weights.

#### I. INTRODUCTION

Control of highly dynamic legged robots is a challenging problem due to the underactuation of the body during many gaits and due to constraints placed on ground reaction forces. As an example, during dynamic gaits<sup>4</sup> such as bounding or galloping, the body of the robot is always underactuated. Additionally, ground reaction forces must always remain in a friction cone to avoid slipping. Current solutions for highly dynamic locomotion include heuristic controllers for hopping and bounding [1], which are effective, but difficult to tune; two-dimensional planar simplifications [2], which are only applicable for gaits without lateral or roll dynamics; and evolutionary optimization for galloping [3], which cannot currently be solved fast enough for online use. Recent results on hardware include execution of bounding limit cycles discovered offline with HyQ [4] and learned pronking, trotting, and bounding gaits on StarLETH [5].

Predictive control can stabilize these dynamic gaits by anticipating periods of flight or underactuation, but is difficult to solve due to the nonlinear dynamics of legged robots and the large number of states and control inputs.



Fig. 1. The MIT Cheetah 3 Robot galloping at 2.5 m/s

recently, the experimental results in [12] show that whole-body nonlinear MPC can be used to stabilize trotting and jumping.

The stabilization of the quadruped robot HyQ using convex optimization discussed in [13] demonstrates the utility of convex optimization, but the approach cannot be immediately extended to dynamic gaits due to the quasi-static simplifications made to the robot model. Similarly, in bipedal locomotion, convex optimization has been used to find the best forces to satisfy instantaneous dynamics requirements [14] and to plan footsteps with the linear inverted pendulum model [15] but the latter approach does not include orientation in the predictive model.

While galloping is well studied in the field of biology [16], [17], surprisingly few hardware implementations of galloping exist. The first robot to demonstrate galloping was the underactuated quadruped Scout II [18], which reached 1.3 m/s, but had limited control of yaw. The MIT Cheetah 1 robot [19] achieved high-speed galloping, but was constrained to a plane. To the best of our knowledge, the only previous implementation of a fully 3D gallop with yaw control is on the hydraulically actuated WildCat robot [20], developed by Boston Dynamics. Unfortunately, no specific details about WildCat or its control system have been

Yuhang Guan (管宇航)

gearguan@gmail.com

广州市优普计算机有限公司

<http://www.uni-power.com.cn/>

2020年10月21日15:00

# Contents

1	简介 INTRODUCTION	3
2	控制架构 CONTROL ARCHITECTURE	4
3	简化的机器人动力学(模型) SIMPLIFIED ROBOT DYNAMICS	8
4	模型预测控制 MODEL PREDICTIVE CONTROL	13
5	结果 RESULTS	23
6	回顾和展望 CONCLUSIONS AND FUTURE WORK	24

## 摘要 *Abstract*

本文描述了一种为力矩控制四足机器人确定地面反作用力的MPC控制方法。机器人动力学(方程)被简化, 以将(控制)问题转换为凸优化的形式, 同时保留有系统的全3D特性(模型虽然被简化, 但三维空间中的动力学特性都得到了保留)。借助这个简化的动力学模型, 在20-30Hz的求解频率下, **地面反作用力规划问题**的预测时长(prediction horizon)最高可到**0.5秒**, 并且求解时间可优化到**1毫秒**以下。尽管用了简化的动力学模型, 但机器人依然可以在不同速度下稳定运动。实验结果表明 stand, trot, flying-trot, pronk, bound, pace, 3-legged gait 和全3D 奔跑步态均可实现。机器人向前运动速度可达3m/s, 横向速度1m/s, 角速度180°/s

## 1 简介 INTRODUCTION

高动态的足式机器人<sup>1</sup>运动控制是一个很有挑战性的问题, 主要有以下两个原因: 1.在很多步态下, 机器人都是欠驱动的; 2.地面反作用力的约束。欠驱动好理解, 地面反作用力约束是指地面反作用力要始终处于摩擦锥 (friction zone) 之内, 否则机器人会打滑。现在对于高动态运动的控制包括:

- 启发式控制(heuristic control): 有效, 但很难调参<sup>2</sup>;
- 二维平面简化(two-dimensional planar simplifications): 无法处理侧向和滚转运动<sup>3</sup>。
- 遗传算法(evolutionary optimization): 可以用, 但算的不够快, 无法实现在线优化<sup>4</sup>。

预测控制(predictive control)可以通过提前预知腾空/欠驱动周期对动态步态进行稳定控制<sup>5</sup>, 但由于足式机器人动力学模型的非线性和大量的状态变量、控制输入而难以求解。对于这个问题, 非线性优化已经被证明可以用来求解使用模型预测控制的单腿跳机器人, 拟人机器人和四足机器人<sup>6</sup>。另一种被广泛使用的方法是同时用两个控制器, 一个叫做 **high-level planner**<sup>7,8</sup>, 一个叫做 **lower level controller**<sup>9</sup>。另外, 最近的实验<sup>10</sup>表明全身非线性MPC (whole-body nonlinear MPC)可以用来稳定控制四足机器人 trotting 和 jumping 运动。

HyQ 机器人的稳定控制用到了凸优化(convex optimization)<sup>11</sup>, 但其中用到的方法不能直接迁移到动态步态(dynamic gaits)运动中, 因为凸优化中使用的机器人模型本身是准静态的(quasi-static)<sup>12</sup>。类似的, 在双足运动中, 凸优化也被用来寻找满足瞬时动力学(instantaneous dynamics)的(地面反作用)力<sup>13</sup>, 然后利用线性倒单摆模型做步态规划<sup>14</sup>, 但后一种方法的预测模型(predictive model)中不包含机器人的姿态。

奔跑运动(Galloping)在生物学中获得了广泛的研究, 但出人意料地很少有硬件实现过这类运动。第一个实现奔跑运

<sup>1</sup> 高动态: highly dynamic, 表示机器人运动时的状态变化起伏很大, 也可以理解为机器人状态变量 $x$ 变化很大。

<sup>2</sup> M. H. Raibert, Legged Robots That Balance. Cambridge, MA, USA:Massachusetts Institute of Technology, 1986.

<sup>3</sup> H.-W. Park, P. M. Wensing, and S. Kim, "High-speed bounding with the mit cheetah 2: Control design and experiments," The International Journal of Robotics Research, vol. 36, no. 2, pp. 167–192, 2017.

<sup>4</sup> D. P. Krasny and D. E. Orin, "Evolution of a 3d gallop in a quadrupedal model with biological characteristics," Journal of Intelligent & Robotic Systems, vol. 60, no. 1, pp. 59–82, 2010.

<sup>5</sup> Predictive control can stabilize these dynamic gaits by anticipating periods of flight or underactuation,...

<sup>6</sup> G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in 2017 IEEE International Conference on Intelligent Robots and Systems, 2017.

<sup>7</sup> . W. Winkler, D. C. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," IEEE Robotics and Automation Letters (RA-L), may 2018.

<sup>8</sup> 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 (RA-L), vol. 2, pp. 2201–2208, oct 2017.

<sup>9</sup> 之所以 high level 叫做 planner, 因为它只负责规划, 并不直接控制机器人, lower level 才直接控制机器人运动, 所以叫做 controller。

<sup>10</sup> M. Neunert, M. St"auble, M. Gifftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-Body Nonlinear Model Predictive Control Through Contacts for Quadrupeds," ArXiv e-prints, Dec. 2017.

<sup>11</sup> M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," Autonomous Robots, vol. 41, no. 1, pp. 259–272, Jan 2017.

<sup>12</sup> 这里说的准静态是指如果机器人运动足够慢, 并且没有倾倒之虞, 那么虽然机器人本身一直在动, 并非处于静态, 但每个时刻都可以视为近似静态。

<sup>13</sup> Y. Abe, M. da Silva, and J. Popovi'c, "Multiobjective control with frictional contacts," in Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ser. SCA'07. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2007, pp. 249–258.

<sup>14</sup> B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in 2010 10th IEEE/RSJ International Conference on Humanoid Robots, Dec 2010, pp. 52–59.

动的是一台欠驱动四足机器人 Scout II<sup>15</sup>，它的运动速度可达 $1.3\text{m/s}$ ，但在偏航控制上有限。MIT 的 Cheetah 1 机器人<sup>16</sup>实现了高速奔跑，但只能限制在一维平面上运动<sup>17</sup>。据我们所知，在此之前唯一具有全向3D奔跑能力的机器人是波士顿动力研发的液压机器人 WildCat，但很不幸，没有任何关于 WildCat 的技术细节或控制系统被公开过。

之前的机器人，除了 WildCat 外都无法实现三维空间快速奔跑，Scout II 和 Cheetah 1 虽然可以高速运动，但不能转弯。

这篇论文的主要贡献是给出了一个(模型)预测控制器，它可以稳定控制很多种步态，包括那些涉及到复杂的姿态动力学(orientation dynamics)的步态。在硬件上，我们实现了在三维空间中快速奔跑(gallop)时最大角速度 $180\text{ deg/sec}$ ，最大线速度 $3.0\text{ m/s}$ 。我们相信这是目前为止电驱动机器人实现的最快奔跑速度，也是跟 Cheetah 3 类似尺寸的足式机器人实现的最大角速度。我们的(模型预测)控制器可以表示为一个单独的(single)凸优化问题，这个优化问题的对象是一个三维空间中有12个 DoF (Degree of Freedom) 的机器人模型。优化问题的求解结果可以直接用来计算关节电机扭矩，无需任何额外的反馈控制器即可让机器人稳定运动。

凸优化方法可以避免非线性优化所需要的技巧，在非线性优化中，可能找不到全局最优解，而凸优化中只要找到最优解，就是全局最优解，并且非线性优化可能存在数值问题<sup>18</sup>。最近在凸优化和模型预测控制上的研究进展<sup>19</sup>带动了很多开源求解器的发展，例如 ECOS 和 qpOASES<sup>20</sup>等，这些求解器使得模型预测控制问题得以快速、稳定地求解。

本文提出的控制器在 MIT Cheetah 3 机器人上进行了测试，机器人分别在跑步机和地面无约束运动，可以 stand(站立)，trot(对角行走)，flying-trot(对角小跑)，pace(左前左后同时迈，右前右后同时迈)，bound(前前，后后，常速)，pronk(前前，后后，跳跃向前)，and gallop(快速奔跑，任意时刻只有一只脚触地，四角轮流)，急速转弯动作和全向运动也进行了测试。机器人由自带的电池供电，控制系统的全部计算也由自带的计算机完成。测试期间，机器人遇到了很多障碍和干扰，包括一个有全尺寸台阶的楼梯，并且台阶上有碎屑(树叶、木屑等)，机器人可以用对角小跑步态盲爬上去<sup>21</sup>。为了处理复杂地形干扰，机器人使用了一种触地检测算法<sup>22</sup>。在所有测试中，包括爬楼梯，机器人除了 IMU，关节位置编码器(旋转编码器)和电机控制器中的电流传感器外没有用任何其他传感器，而且电流传感器仅用于电机的电流反馈控制中。

论文组织结构如下。第二节描述了 Cheetah 3 机器人的控制架构。(机器人)动力学模型在第三节描述，第四节是模型预测控制的公式(推导)。第五部分是测试的细节和结果。

## 2 控制架构 CONTROL ARCHITECTURE

### A. MIT Cheetah3 机器人 *MIT Cheetah 3 Robot*

MIT Cheetah 3 机器人，如图 1 所示，是一台由 MIT Biomimetic Robotics Lab 建造的 45 kg 四足机器人。这台机器人使用了为 Cheetah 2 机器人特别研发的驱动技术，其中有低惯量，易回转的髌膝关节驱动器(电机)，可以有效降低(来自于地面反作用力的)冲击。Cheetah 3 机器人在髌膝和 ab/ad 关节都使用了同样的驱动技术，这样它可以完成动态转弯(dynamic turns)。

机器人有四条腿，每条腿有三个关节：外展/内收(ab/ad)，髌(hip)和膝(knee)，所有关节电机都是扭矩控制的(torque controlled)。每个关节最大可产生 $250\text{ N}\cdot\text{m}$ 的扭矩，最大转速为 $21\text{ rad/sec}$  ( $3.34\text{ cycles/sec}$ )。电机的扭矩控制是靠调整电机电流控制回路的设置值来实现的，没有扭矩或力传感器<sup>23</sup>。

<sup>15</sup>J. A. Smith and I. Poulakakis, "Rotary gallop in the untethered quadrupedal robot scout ii," in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol. 3, Sept 2004, pp. 2556–2561 vol.3.

<sup>16</sup>H.-W. Park and S. Kim, "Quadrupedal galloping control for a wide range of speed via vertical impulse scaling," Bioinspiration and Biomimetics, vol. 10, no. 2, p. 025003, 2015.

<sup>17</sup>一维平面上运动指狗无法左右转弯，只能走直线，因此狗运动的空间是一个平面。

<sup>18</sup>数值问题可能是指误差累积导致求解不准确，或者线性化时截断误差比较大。

<sup>19</sup>Y. Wang and S. Boyd, "Fast model predictive control using online optimization," IEEE Transactions on Control Systems Technology, vol. 18, no. 2, pp. 267–278, March 2010.

<sup>20</sup><https://github.com/coin-or/qpOASES>

<sup>21</sup>此处的楼梯应该是户外那种大台阶楼梯，阶数不多，而非室内的楼梯。

<sup>22</sup>G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in 2018 IEEE International Conference on Robotics and Automation, 2018.

<sup>23</sup>原文: Motor torque control is accomplished by adjusting the setpoint of the motor current control loop; there are no torque or force sensors.

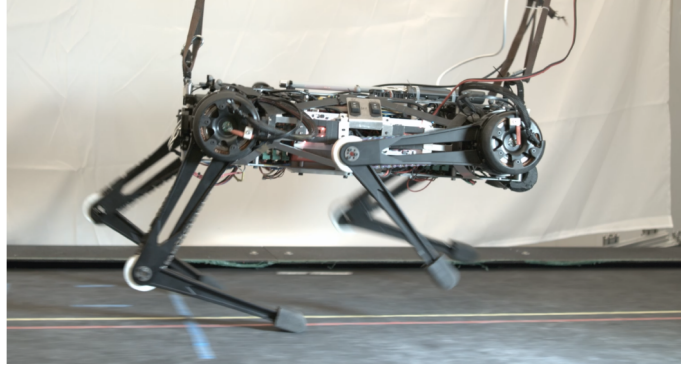


Figure 1: 以 2.5 m/s 速度奔跑的 MIT Cheetah 3 机器人

### B. (符号)约定 *Conventions*

机器人的身体坐标系和世界坐标系定义如图2所示。所有在身体坐标系中表示的量都有一个左下角标 $\mathcal{B}$ ，没有下角标的量是在世界坐标系下的。向量用粗体，竖直，小写字母表示，如 $(\mathbf{a}, \boldsymbol{\omega})$ ，矩阵用粗体，竖直，大写字母表示 $(\mathbf{A}, \boldsymbol{\Omega})$ ，标量用小写，斜体字母表示 $(a, \omega)$ 。 $\mathbf{1}_n$ 表示 $n \times n$ 的单位矩阵<sup>24</sup>。

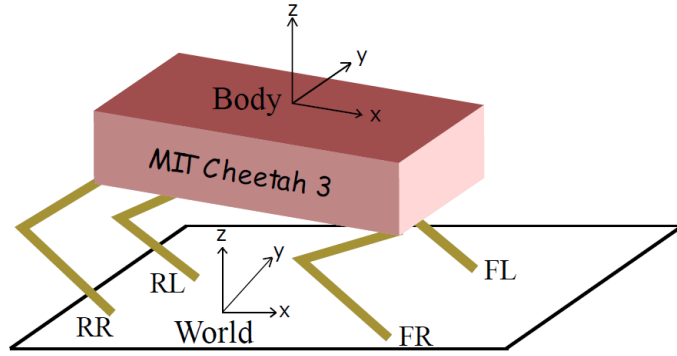


Figure 2: 机器人的世界坐标系和身体坐标系

### C. 状态机 *State Machine*

本文提出的控制器(MPC)是为固定周期的抬脚、落足运动确定三维地面反作用力的<sup>25</sup>。当一只脚按计划处于摆动相时，机器人会运行摆动腿控制器，这个控制器对于过早触地也是稳健的。如果不是处于摆动相，而是支撑相，那么机器人会运行地面反作用力控制器。这两种控制器如图3中所示，它们的细节会在下面两个小节中描述。

这里的状态机不是指运动状态之间的切换，而是某条腿在摆动相和支撑相之间的切换。随着状态机不断切换，这条腿处于“摆动腿控制器”或“地面反作用力控制器”的控制。

### D. 摆动腿控制 *Swing Leg Control*

摆动控制器计算并控制腿跟随世界坐标系下的一条轨迹。控制器在跟随这条轨迹时，使用了反馈项和前馈项的和来计算关节扭矩。计算第 $i$ 条腿关节扭矩的控制律为<sup>26</sup>

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T [\mathbf{K}_p (\mathcal{B}\mathbf{p}_{i,\text{ref}} - \mathcal{B}\mathbf{p}_i) + \mathbf{K}_d (\mathcal{B}\mathbf{v}_{i,\text{ref}} - \mathcal{B}\mathbf{v}_i)] + \boldsymbol{\tau}_{i,\text{ff}} \quad (1)$$

其中

<sup>24</sup>用 $\mathbf{1}_n$ 表示单位矩阵因为 $\mathbf{I}$ 用来表示惯性张量了。

<sup>25</sup>原文: The controller proposed in this paper determines desired 3D ground reaction forces for behaviors with fixed-timing foot placement and liftoff. 此处 fixed-timing 指的应该是踏步周期固定的运动，机器人运动快慢靠步幅调节，而非步频。

<sup>0</sup> 原文: The reference trajectory contains desired velocities and contact sequence from the operator. 这里的接触序列可能是步态种类，踏步频率等信息。

<sup>26</sup>根据Mark W. Spong著《机器人建模和控制》77页，“静态力/力矩关系”小节，机器人末端执行器处的力和力矩向量 $\mathbf{F}$ 和机器人关节力矩向量 $\boldsymbol{\tau}$ 之间的关系为 $\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F}$

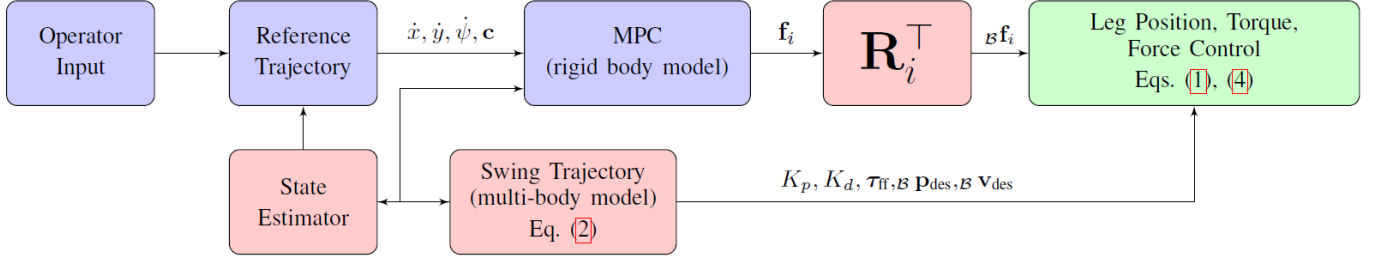


Figure 3: 机器人控制系统框图：reference trajectories 包含期望速度和接触序列<sup>0</sup>。蓝色模块运行频率为 30 Hz，红色模块运行频率为 1k Hz，绿色模块运行频率为 4.5 kHz。

- $\boldsymbol{\tau}_i \in \mathbb{R}^3$ ：第  $i$  条摆动腿的关节扭矩，因为每条腿有 3 个关节，因此  $\boldsymbol{\tau}_i$  是  $3 \times 1$  的向量
- $\mathbf{J}_i \in \mathbb{R}^{3 \times 3}$ ：足部雅可比矩阵<sup>27</sup>，即第  $i$  条摆动腿的雅可比矩阵
- $\mathbf{K}_p \in \mathbb{R}^{3 \times 3}$ ：对角正定矩阵-比例增益
- $\mathbf{K}_d \in \mathbb{R}^{3 \times 3}$ ：对角正定矩阵-微分增益
- $\mathcal{B}\mathbf{p}_{i,\text{ref}} \in \mathbb{R}^3$ ：第  $i$  条腿摆动轨迹的参考位置
- $\mathcal{B}\mathbf{v}_{i,\text{ref}} \in \mathbb{R}^3$ ：第  $i$  条腿摆动轨迹的参考速度
- $\mathcal{B}\mathbf{p}_i \in \mathbb{R}^3$ ：第  $i$  条摆动腿的实际位置
- $\mathcal{B}\mathbf{v}_i \in \mathbb{R}^3$ ：第  $i$  条摆动腿的实际速度
- $\boldsymbol{\tau}_{i,\text{ff}}$ ：前馈扭矩<sup>28</sup>

前馈扭矩  $\boldsymbol{\tau}_{i,\text{ff}}$  的计算公式为

$$\boldsymbol{\tau}_{i,\text{ff}} = \mathbf{J}^\top \boldsymbol{\Lambda}_i (\mathcal{B}\mathbf{a}_{i,\text{ref}} - \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i) + \mathbf{C}_i \dot{\mathbf{q}}_i + \mathbf{G}_i \quad (2)$$

- $\boldsymbol{\Lambda}_i \in \mathbb{R}^{3 \times 3}$ ：操作空间惯性矩阵<sup>29</sup>
- $\mathcal{B}\mathbf{a}_{i,\text{ref}} \in \mathbb{R}^3$ ：(原文没有左下角标  $\mathcal{B}$ ) 在身体坐标系中表示的参考加速度<sup>30</sup>
- $\mathbf{q}_i \in \mathbb{R}^3$ ：关节位置向量，所以  $\dot{\mathbf{q}}_i$  就是关节角速度向量了
- $\mathbf{C}_i \dot{\mathbf{q}}_i + \mathbf{G}_i$ ：由于重力和科里奥利力对腿部造成的扭矩<sup>31</sup>

合并以上两个公式，可得到第  $i$  条腿三个关节的关节扭矩计算公式为：

$$\boldsymbol{\tau}_i = \underbrace{\mathbf{J}_i^\top [\mathbf{K}_p (\mathcal{B}\mathbf{p}_{i,\text{ref}} - \mathcal{B}\mathbf{p}_i) + \mathbf{K}_d (\mathcal{B}\mathbf{v}_{i,\text{ref}} - \mathcal{B}\mathbf{v}_i)]}_{PD \text{ Controller}} + \underbrace{\mathbf{J}^\top \boldsymbol{\Lambda}_i (\mathcal{B}\mathbf{a}_{i,\text{ref}} - \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i) + \mathbf{C}_i \dot{\mathbf{q}}_i + \mathbf{G}_i}_{Feed-forward \text{ Controller}} \quad (3)$$

<sup>27</sup> 原文：foot Jacobian。

<sup>28</sup> 前馈控制是根据扰动或给定值的变化按补偿原理来工作的，其特点是当扰动产生后，被控变量还未变化以前，根据扰动作用的大小进行控制，以补偿扰动作用对被控变量的影响。前馈控制系统运用得当，可以使被控变量的扰动消灭在萌芽之中，使被控变量不会因扰动作用或给定值变化而产生偏差，它较之反馈控制能更加及时地进行控制，并且不受系统滞后的影响。前馈控制的好处是直接控制无滞后，可以提高系统的响应速率，但是需要比较准确地知道被控对象模型和系统特性。而反馈控制的优点是不知道被控对象的模型即可实现比较准确的控制，但是需要偏差发生之后才能进行调节，具有滞后性。

<sup>29</sup> 根据《Springer Handbook of Robotics》“The operational-space inertia matrix, which maps task accelerations to task forces in operational or Cartesian space.”。operational space 就是 task space 的别名。task space 是描述 task 的空间，与机器人无关，如果机器人的工作是在二维平面上画画，那么 task space 就是二维平面，如果是搬箱子，那么 task space 就是 Cartesian space。

<sup>30</sup> reference acceleration。

<sup>31</sup> 这里的  $\mathbf{C}_i$  不是重力，而是由重力引起的腿部的额外扭矩。



这里的  $\mathcal{B}\mathbf{p}_{i,\text{ref}}$ ,  $\mathcal{B}\mathbf{a}_{i,\text{ref}}$ ,  $\mathcal{B}\mathbf{v}_{i,\text{ref}}$  等都是摆动腿足端运动轨迹的信息, 例如贝塞尔曲线或者抛物线轨迹, 与 MPC 控制器中的参考轨迹不同。

为了确保腿部构型在大范围内变化时, 都保持高增益稳定性<sup>32</sup>, 必须调整  $\mathbf{K}_p$  矩阵以保持闭环系统的自然频率<sup>33</sup> 相对稳定, 以应对腿部表观质量的变化<sup>34</sup>。  $\mathbf{K}_p$  矩阵的第  $i$  个对角线元素负责维持第  $i$  根轴的自然频率  $\omega_i$  保持常量, 计算公式可以近似表达为:

$$K_{p,i} = \omega_i^2 \Lambda_{i,i} \quad (4)$$

- $\omega_i$ : (腿上)第  $i$  根轴的自然频率<sup>35</sup>
- $\Lambda_{i,i}$ : 质量矩阵的第  $i$  行第  $i$  列, 对应于腿沿着第  $i$  根轴的表现质量<sup>36</sup>

摆动腿期望的落足点位置是根据一个简单的启发式方法<sup>37</sup> 计算得到的, 细节将会在第五节中描述。此外, 一个触地检测算法<sup>38</sup> 可以检测并处理过早或过晚的触地情况。一旦检测到过早触地了, 那么控制器会立刻切换到站立模式并开始地面反作用力控制。

摆动腿的运动与地面反作用力无关, 如何快速高效地将摆动腿从地面一个位置移动到目标位置, 足端画出怎样的轨迹, 这些问题是研究的关键。在触地的一瞬间, 腿是根据摆动腿控制器命令运动的, 然后立刻切换到支撑腿控制器。

## E. 地面反作用力控制 *Ground Force Control*

在地面反作用力控制期间<sup>39</sup>, 关节扭矩计算公式为

$$\boldsymbol{\tau}_i = \mathbf{J}_i^T \mathbf{R}_i^T \mathbf{f}_i \quad (5)$$

- $\boldsymbol{\tau}_i \in \mathbb{R}^3$ : 第  $i$  条支撑腿的关节扭矩, 与摆动腿控制类似,  $\boldsymbol{\tau}_i$  也是  $3 \times 1$  的向量
- $\mathbf{J} \in \mathbb{R}^{3 \times 3}$ : 与公式 1 中的足部雅可比矩阵定义相同
- $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ : 从机器人身体坐标系变换到世界坐标系的旋转矩阵
- $\mathbf{f}_i \in \mathbb{R}^3$ : 预测控制器(MPC控制器)计算得到的世界坐标系下的(地面作用)力向量<sup>40</sup>

地面反作用力控制器就是支撑腿控制器, 与摆动腿控制器联合在一起, 形成了全部的腿控制器。腿控制器接收MPC传来的地面反作用力指令, 直接计算得到各个关节的扭矩。

从公式中可以看出, 每条腿都需要一个独立的控制器, 也就是说任意时刻都有四个腿控制器在运行着。如果是 trot 步态, 那么就是 1 和 3 号腿的控制器处于同一状态, 2 和 4 号腿的控制器处于同一状态。因为计算公式中涉及到  $\mathbf{J}_i \in \mathbb{R}^{3 \times 3}$ , 所以需要状态估计器反馈的信息,  $\dot{\mathbf{q}}_i$  是从状态估计器还是直接从编码器中获得还不知

<sup>32</sup>原文: To ensure high-gain stability in a wide range of leg configurations. 参考: Gain is a proportional value that shows the relationship between the magnitude of the input to the magnitude of the output signal at steady state. Many systems contain a method by which the gain can be altered, providing more or less "power" to the system. However, increasing gain or decreasing gain beyond a particular safety zone can cause the system to become unstable.

<sup>33</sup>闭环系统的自然频率: 这里说的自然频率应该是指系统的固有频率, 结构的固有频率只受刚度分布和质量分布的影响, 如果腿在小范围内改变构型, 那么固有频率基本不发生变化; 但如果腿从伸直变为全弯曲, 那么固有频率就会发生相对大的改变。固有频率就是不同阶的共振频率, 它们的组合决定了系统的频域响应。

<sup>34</sup>apparent mass: 表观质量。当物体在非惯性系中运动时, 除了重力等非惯性力, 还会受到科里奥利力等惯性力的作用。所有这些力在惯性系中的和, 称为表观重量, 除以重力加速度  $\mathbf{g}$ , 即为表观质量。因此, 当物体以不同的加速度移动或旋转时, 也会具有不同的表观质量。参考: Apparent weight

<sup>35</sup> $\omega_i$  是如何得到的?

<sup>36</sup>这里的  $\Lambda_i$  矩阵与上面公式中的不一样, 而且要注意,  $\Lambda_i$  不是对角矩阵。

<sup>37</sup>M. H. Raibert, Legged Robots That Balance. Cambridge, MA, USA: Massachusetts Institute of Technology, 1986.

<sup>38</sup>G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in 2018 IEEE International Conference on Robotics and Automation, 2018.

<sup>39</sup>腿处于支撑相, 由地面反作用力控制器控制。

<sup>40</sup> $\mathbf{R}_i^T$  用来将世界坐标系下的足底反力转换到身体坐标系中, 然后根据  $\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q})\mathbf{F}$  计算得到关节扭矩。这里没有考虑重力、摩擦力等因素, 全部留给 MPC 控制器处理。

道。 $\mathcal{B}\mathbf{a}_{i,\text{ref}}$  如何计算也不清楚，可能跟规划的足端轨迹有关。

### 3 简化的机器人动力学(模型) SIMPLIFIED ROBOT DYNAMICS

预测控制器(MPC控制器)将机器人建模为一个单刚体，受力区域为机器人足端与地面的接触区域<sup>41</sup>。虽然忽略腿部动力学特性是一个很大的简化，但该控制器依然能够稳定控制一个多自由度系统<sup>42</sup>，对(四条腿引起的)多体(动力学)效应应具有(不错的)鲁棒性<sup>43</sup>。对于 Cheetah 3 机器人来说，这个简化是可以接受的，因为腿部的质量大概只占机器人总质量的 10%。对于每条腿受到的地面反作用力  $\mathbf{f}_i \in \mathbb{R}^3$ ，都对应着一个从质心<sup>44</sup> 指向力作用点<sup>45</sup> 的向量  $\mathbf{r}_i \in \mathbb{R}^3$ <sup>46</sup>。世界坐标系下的刚体动力学方程为

$$\begin{aligned}\ddot{\mathbf{p}} &= \frac{\sum_{i=1}^n \mathbf{f}_i}{m} - \mathbf{g} \\ \frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) &= \sum_{i=1}^n \mathbf{r}_i \times \mathbf{f}_i \\ \dot{\mathbf{R}} &= [\boldsymbol{\omega}]_{\times} \mathbf{R}\end{aligned}\tag{6}$$

- $\mathbf{p} \in \mathbb{R}^3$ : 单刚体模型的位置坐标
- $m \in \mathbb{R}$ : 单刚体模型的质量(机器人质量)
- $\mathbf{g} \in \mathbb{R}^3$ : 重力加速度
- $\mathbf{I} \in \mathbb{R}^3$ : 单刚体模型的惯性张量<sup>47</sup>
- $\boldsymbol{\omega} \in \mathbb{R}^3$ : 机器人角速度<sup>48</sup>
- $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ : 从身体坐标系到世界坐标系的旋转变换矩阵(只是旋转矩阵，不是其次变换矩阵)
- $[\mathbf{x}]_{\times} \in \mathbb{R}^{3 \times 3}$  表示反对称矩阵，使得  $[\mathbf{x}]_{\times} \mathbf{y} = \mathbf{x} \times \mathbf{y}$  对于所有  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^3$  成立

第一个方程描述了单刚体模型的平动，依据牛顿第二定律；第二个方程描述了单刚体模型的转动，依据角动量定理<sup>a</sup> 第三个方程是单刚体模型的角速度在世界坐标系下的表示<sup>b</sup>。

<sup>a</sup> 角动量定义为:  $\mathbf{L} = \mathbf{I}\boldsymbol{\omega}$  求导即为角动量定理:  $\frac{d\mathbf{L}}{dt} = \frac{d\mathbf{I}}{dt}\boldsymbol{\omega} + \mathbf{I}\frac{d\boldsymbol{\omega}}{dt} = \mathbf{r} \times \mathbf{f}$ 。

<sup>b</sup> 参考 Mark.W.Spong 著《机器人建模和控制》66页,  $\mathbf{R}_n^0 = S(\boldsymbol{\omega}_{0,n}^0)\mathbf{R}_n^0$  表示坐标系  $n$  相对于坐标系 0 的总角速度在 0 系中的表示，这个角速度是由旋转矩阵  $\mathbf{R}_n^0$  导致的， $S(\boldsymbol{\omega}_{0,n}^0)$  意为将角速度  $\boldsymbol{\omega}_{0,n}^0$  写成反对称矩阵形式，与公式 6 中的  $[\boldsymbol{\omega}]_{\times}$  含义相同。

因为公式 6 中后面两个方程是非线性的，因此直接使用这个单刚体模型的动力学方程可能会导致后面的模型预测控制中的优化问题非凸，所以在下面的两个小节中会对公式 6 中后面两个方程做一些近似处理，使其线性化<sup>49</sup>。

#### A. 近似的角速度动力学 *Approximated Angular Velocity Dynamics*

机器人的姿态用一组 Z-Y-X 欧拉角表示<sup>50</sup>，写作向量形式为  $\boldsymbol{\Theta} = [\phi \ \theta \ \psi]^T$ ，其中  $\psi$  表示偏航(Yaw Angle)， $\theta$  表示俯仰(Pitch Angle)， $\phi$  表示滚转(Roll Angle)。这些角度对应着一系列的旋转变换，使得从身体坐标系到世界坐标系的

<sup>41</sup>contact patches: 谷歌图片搜索显示，contact patches 多用于表示汽车轮胎与地面的接触区域，因此单刚体模型中机器人与地面采用的是面接触，而非点接触。

<sup>42</sup>high-DoF system.

<sup>43</sup>robust to these multi-body effects.

<sup>44</sup>center of mass (COM).

<sup>45</sup>足端与地面的接触点。

<sup>46</sup>向量  $\mathbf{r}_i \in \mathbb{R}^3$  用来计算地面反作用力施加到单刚体模型上的扭矩。

<sup>47</sup>robot's inertia tensor.

<sup>48</sup>从公式形式可以看出，这个角速度是机器人相对于身体坐标系的角速度，所以需要旋转矩阵  $\mathbf{R}$  将这个角速度映射到世界坐标系下。

<sup>50</sup>欧拉角就是从某一参考坐标出发，经过三次基本旋转变换，实现三维空间中任意姿态的一组旋转角，其中相邻两次旋转不能绕同一根轴或者平行轴。如果第一次和第三次旋转绕着同一根轴或者平行轴(例如 X-Y-X, Y-Z-Y 等)，这种欧拉角叫做经典欧拉角(proper Euler angle)。如果三次旋转的旋转轴都不同(例如 X-Y-Z, Z-Y-X 等)，叫做 Tait-Bryan 角/万向角(Cardan angles)/RPY 角(roll-pitch-yaw)，机器人学中经常使用后者。一般来说，Z-Y-X 欧拉角叫做 Tait-Bryan 角因为经常用在航空器领域，因此又叫做 RPY 角，X-Y-Z 欧拉角叫做万向角。需要注意的是，无论哪种形式的欧拉角，下一次的旋转都是基于上一次旋转得到的坐标系进行的，换句话说，是基于“动系”进行的旋转，旋转矩阵依次右乘。如果每次旋转都基于一个给定的初始坐标系，那么最终的姿态是旋转矩阵依次左乘。假设有一基础坐标系(base frame)  $o_0x_0y_0z_0$  和任一不同姿态的坐标系  $o_0x_ny_nz_n$ ，无论基于动系还是  $o_0x_0y_0z_0$ ，都可以用一组欧拉角表示它们之间的关系，并且最终的结果都表示  $o_0x_ny_nz_n$  在  $o_0x_0y_0z_0$  中的姿



旋转矩阵可以写作

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi) \quad (7)$$

- $\mathbf{R}_z(\psi)$ : 绕  $Z$  轴正向旋转  $\psi$  角
- $\mathbf{R}_y(\theta)$ : 绕  $Y$  轴正向旋转  $\theta$  角
- $\mathbf{R}_x(\phi)$ : 绕  $X$  轴正向旋转  $\phi$  角

补充推导: 因为将机器人视为单刚体, 所以忽略了腿部运动, 那么机器人在世界坐标系下的角速度就是 Body 的角速度。从 IMU 中可以读出来 Body 坐标系相对于世界坐标系的 RPY 角数据, 计算时间导数  $\dot{\Theta} = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T$ , 但这只是欧拉角的变化率, 而非实际的角速度。接下来, 推导从欧拉角时间导数到坐标系角速度的映射关系:  $f: \dot{\Theta} \rightarrow \omega$

对比欧拉角的向量形式和公式 7 可以看出, 从世界坐标系到身体坐标系的旋转变换顺序为 X-Y-Z, 参考系均是世界坐标系, 因此左乘旋转矩阵<sup>a</sup>。逆着旋转顺序看, 最后一个旋转的是  $Z$  轴, 所以  $Z$  对应的旋转角的时间导数, 就是身体坐标系在世界坐标系中沿着  $Z$  轴方向的角速度, 所以映射的最后一列为

$$\mathbf{1}_{3 \times 3} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

第二列经过了一次绕  $Z$  轴的变换, 所以

$$\begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin \psi \\ \cos \psi \\ 0 \end{bmatrix}$$

映射矩阵的第一列经过了  $Z$ - $Y$  两次旋转, 所以

$$\begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi \\ \cos \theta \sin \psi \\ -\sin \theta \end{bmatrix}$$

所以最终的变换矩阵为<sup>b</sup>

$$\omega = \begin{bmatrix} \cos \theta \cos \psi & -\sin \psi & 0 \\ \cos \theta \sin \psi & \cos \psi & 0 \\ -\sin \theta & 0 & 1 \end{bmatrix} \dot{\Theta}$$

<sup>a</sup> 变换顺序也可以为  $Z$ - $Y$ - $X$ , 旋转轴基于“动系”, 旋转矩阵依次右乘, 效果、表达式和结论是一样的。

<sup>b</sup> 论文中(3,1)元素为 0 是错的, 应该如推导中所述, 为  $-\sin \theta$ , 但不影响后续计算结果, 因为原文中对矩阵求逆的结果是对的(已经在 Maple 中验证)。

(刚体)在世界坐标系下的角速度可以用欧拉角的时间导数计算, 映射关系为

$$\omega = \begin{bmatrix} \cos(\theta) \cos(\psi) & -\sin(\psi) & 0 \\ \cos(\theta) \sin(\psi) & \cos(\psi) & 0 \\ -\sin(\theta) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (8)$$

如果机器人不是竖直向上的状态( $\cos(\theta) \neq 0$ )<sup>51</sup>, 公式 8 可以变换为求解欧拉角时间导数的形式

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi)/\cos(\theta) & \sin(\psi)/\cos(\theta) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ \cos(\psi)\tan(\theta) & \sin(\psi)\tan(\theta) & 1 \end{bmatrix} \omega \quad (9)$$

态, 两种变换得到的结果是相同的。

如果  $o_0x_0y_0z_0$  和  $o_0x_ny_nz_n$  之间的变换关系为  $\mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$ , 那么可以解释成按照  $Z$ - $Y$ - $X$  的顺序基于动系旋转, 也可以解释成基于“定系”分别绕  $X_0$ - $Y_0$ - $Z_0$  旋转。

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

<sup>51</sup> 按照最后一行展开, 映射矩阵的行列式值为  $\cos \theta$ , 为了保证映射有效, 所以  $\cos(\theta) \neq 0$ 。

当俯仰角和滚转角  $(\phi, \theta)$  比较小时，公式 9 可以近似为

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \omega \quad (10)$$

公式 9 中的矩阵就是绕 Z 轴的基本旋转矩阵，所以可以写做

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \mathbf{R}_z(\psi) \omega \quad (11)$$

$(\phi, \theta) \approx 0$  时， $\cos(\theta) \approx 1$ ， $\tan(\theta) \approx 0$ ，因为公式 9 中的矩阵与  $\phi$  无关，所以这里的近似只需要  $\theta$  足够小即可。换句话说，只要满足俯仰角不大，这里的近似关系就成立，滚转和偏航角没关系。公式 10 中旋转矩阵退化为 Z 轴的基本旋转矩阵是理所当然的，因为  $\theta \approx 0$ ， $\psi$ ，与映射矩阵无关，所以就只剩下绕 Z 轴的旋转了。

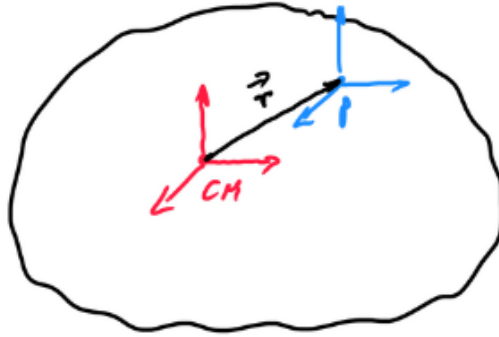
注意，欧拉角旋转顺序的定义是很重要的，如果是其他的旋转顺序(X-Y-Z, Z-X-Y等)，那么对于有意义的姿态，上述的近似也是不对的。

公式 6 中第二个方程可以近似处理为

$$\frac{d}{dt}(\mathbf{I}\omega) = \mathbf{I}\dot{\omega} + \omega \times (\mathbf{I}\omega) \approx \mathbf{I}\dot{\omega} \quad (12)$$

关于  $\frac{d}{dt}(\mathbf{I}\omega) = \frac{d\mathbf{I}}{dt}\omega + \mathbf{I}\frac{d\omega}{dt} = \mathbf{I}\dot{\omega} + \omega \times (\mathbf{I}\omega)$  的解释:

$\mathbf{I}\frac{d\omega}{dt} = \mathbf{I}\dot{\omega}$  是很直观的，下面解释  $\frac{d\mathbf{I}}{dt}\omega = \omega \times (\mathbf{I}\omega)$  <sup>a</sup>:



1. 平行轴定理：假设有一刚体，绕通过其质心  $C$  的轴  $Z$  的惯性张量为  $\mathbf{J}_C$ ，那么绕着通过  $P$  点且平行于  $Z$  的轴  $Z'$  的惯性张量为<sup>b</sup>

$$\mathbf{J}_P = \mathbf{J}_C + m\tilde{\mathbf{r}}\tilde{\mathbf{r}}^T$$

$$\tilde{\mathbf{r}}\tilde{\mathbf{r}}^T = \tilde{\mathbf{r}}\tilde{\mathbf{r}}^T - \tilde{\mathbf{r}}^T\tilde{\mathbf{r}}\mathbf{1}_3$$

- $m$ : 物体的质量
- $\tilde{\mathbf{r}}$ : 从质心  $C$  指向点  $P$  的向量

2. 实际上  $\tilde{\mathbf{r}}$  是  $\mathbf{r}$  向量的反对称矩阵形式，验证如下:

$$\tilde{\mathbf{r}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \tilde{\mathbf{r}} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

那么

$$\vec{\mathbf{r}}\vec{\mathbf{r}}^T = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} x & y & z \end{bmatrix} = \begin{bmatrix} x^2 & xy & xz \\ yx & y^2 & yz \\ zx & zy & z^2 \end{bmatrix}$$

$$\vec{\mathbf{r}}^T \vec{\mathbf{r}} \mathbf{1}_3 = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x^2 + y^2 + z^2 & 0 & 0 \\ 0 & x^2 + y^2 + z^2 & 0 \\ 0 & 0 & x^2 + y^2 + z^2 \end{bmatrix}$$

所以:

$$\vec{\mathbf{r}}\vec{\mathbf{r}}^T - \vec{\mathbf{r}}^T \vec{\mathbf{r}} \mathbf{1}_3 = \begin{bmatrix} -y^2 - z^2 & xy & xz \\ yx & -x^2 - z^2 & yz \\ zx & zy & -x^2 - y^2 \end{bmatrix}$$

计算:

$$\tilde{\mathbf{r}}\tilde{\mathbf{r}} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} = \begin{bmatrix} -y^2 - z^2 & xy & xz \\ yx & -x^2 - z^2 & yz \\ zx & zy & -x^2 - y^2 \end{bmatrix} = \vec{\mathbf{r}}\vec{\mathbf{r}}^T - \vec{\mathbf{r}}^T \vec{\mathbf{r}} \mathbf{1}_3$$

3. 现在想要在一个惯性系(世界坐标系)中得到刚体的角动量  $\vec{\mathbf{L}} = \mathbf{J}_I \boldsymbol{\omega}$ , 那就必须将非惯性系中的惯性张量  $\mathbf{J}_P$  变换到惯性系, 即世界坐标系中, 记为  $\mathbf{J}_I$ 。假设从身体坐标系到世界坐标系的变换矩阵为  $\mathbf{R}$ , 那么在世界坐标系中的惯性张量为

$$\mathbf{J}_I = \mathbf{R} \mathbf{J}_P \mathbf{R}^T$$

世界坐标系中刚体角动量为:

$$\vec{\mathbf{L}} = \mathbf{J}_I \boldsymbol{\omega} = \mathbf{R} \mathbf{J}_P \mathbf{R}^T \boldsymbol{\omega}$$

4. 角动量对时间求导

$$\frac{d}{dt} \vec{\mathbf{L}} = \frac{d}{dt} (\mathbf{J}_I \boldsymbol{\omega}) = \mathbf{J}_I \dot{\boldsymbol{\omega}} + \frac{d}{dt} (\mathbf{R} \mathbf{J}_P \mathbf{R}^T) \boldsymbol{\omega}$$

$$\frac{d}{dt} (\mathbf{R} \mathbf{J}_P \mathbf{R}^T) \boldsymbol{\omega} = (\dot{\mathbf{R}} \mathbf{J}_P \mathbf{R}^T + \mathbf{R} \mathbf{J}_P \dot{\mathbf{R}}^T) \boldsymbol{\omega}$$

因为:

$$\dot{\mathbf{R}} = \tilde{\boldsymbol{\omega}} \mathbf{R}, \quad \dot{\mathbf{R}}^T = -\mathbf{R}^T \tilde{\boldsymbol{\omega}}$$

代入有:

$$\begin{aligned} \frac{d}{dt} (\mathbf{R} \mathbf{J}_P \mathbf{R}^T) \boldsymbol{\omega} &= (\tilde{\boldsymbol{\omega}} \mathbf{R} \mathbf{J}_P \mathbf{R}^T - \mathbf{R} \mathbf{J}_P \mathbf{R}^T \tilde{\boldsymbol{\omega}}) \boldsymbol{\omega} = (\tilde{\boldsymbol{\omega}} \mathbf{J}_I - \mathbf{J}_I \tilde{\boldsymbol{\omega}}) \boldsymbol{\omega} = \tilde{\boldsymbol{\omega}} (\mathbf{J}_I \boldsymbol{\omega}) - \mathbf{J}_I \tilde{\boldsymbol{\omega}} \boldsymbol{\omega} = \boldsymbol{\omega} \times (\mathbf{J}_I \boldsymbol{\omega}) - \mathbf{0} \\ &= \boldsymbol{\omega} \times (\mathbf{J}_I \boldsymbol{\omega}) \end{aligned}$$

所以

$$\frac{d}{dt} \vec{\mathbf{L}} = \mathbf{J}_I \dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times (\mathbf{J}_I \boldsymbol{\omega})$$

<sup>a</sup> 参考stackexchange上的回答, 网址: <https://physics.stackexchange.com/questions/530248/first-derivative-of-the-moment-of-inertia-in-time-as-a-physical-parameter> 图片比较难看, 因为在两边加线的区域内无法正常使用 figure 功能, 只能用 includegraphics。

<sup>b</sup> 这里的平行轴定理比较抽象, 因为三维空间中的转动惯量是惯性张量, 计算上比一维情况复杂些。

这种近似已经在控制器中被使用过了(公式 12 中的近似)<sup>52</sup>, 其忽略了进动(precession)<sup>53</sup> 和章动(nutation)<sup>54</sup> 的影响。当刚体角速度很小时,  $\boldsymbol{\omega} \times (\mathbf{I} \boldsymbol{\omega})$  项也很小, 对机器人动力学没有明显贡献。惯性张量在世界坐标系中的描述为

$$\mathbf{I} = \mathbf{R} \mathbf{I}_B \mathbf{R}^T \quad (13)$$

<sup>52</sup> M. Focchi, A. del Prete, I. Havoutis, R. Featherstone, D. G. Caldwell, and C. Semini, "High-slope terrain locomotion for torque-controlled quadruped robots," Autonomous Robots, vol. 41, no. 1, pp. 259–272, Jan 2017.

<sup>53</sup> <https://en.wikipedia.org/wiki/Precession>

<sup>54</sup> [https://en.wikipedia.org/wiki/Astronomical\\_nutation](https://en.wikipedia.org/wiki/Astronomical_nutation)

从公式 7 和公式 11 中可以看出，对于小的滚转角和俯仰角，公式 13 可以近似为

$$\hat{\mathbf{I}} = \mathbf{R}_z(\psi) {}_B\mathbf{I} \mathbf{R}_z(\psi)^T \quad (14)$$

其中  ${}_B\mathbf{I}$  表示惯性张量在身体坐标系下的表示。

## B. 简化的机器人动力学 *Simplified Robot Dynamics*

经过近似处理的姿态动力学方程和平动动力学方程可以结合起来，写成矩阵的形式

$$\frac{d}{dt} \begin{bmatrix} \hat{\Theta} \\ \hat{\mathbf{p}} \\ \hat{\omega} \\ \hat{\dot{\mathbf{p}}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \begin{bmatrix} \hat{\Theta} \\ \hat{\mathbf{p}} \\ \hat{\omega} \\ \hat{\dot{\mathbf{p}}} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_{\times} & \dots & \hat{\mathbf{I}}^{-1}[\mathbf{r}_n]_{\times} \\ \mathbf{1}_3/m & \dots & \mathbf{1}_3/m \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{g} \end{bmatrix} \quad (15)$$

整理成矩阵形式因为要得到系统状态方程(state equation, 形如  $\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t)$ )。从表达式看，状态向量(state vector)  $\mathbf{x}$  是  $\mathbf{x} = [\hat{\Theta} \ \hat{\mathbf{p}} \ \hat{\omega} \ \hat{\dot{\mathbf{p}}}]^T$ ，物理含义分别为：身体坐标系在世界坐标系下的欧拉角；身体坐标系在世界坐标系下的位置；世界坐标系下身体坐标系的角速度；身体坐标系在世界坐标系下的速度。 $\hat{\omega}$ ， $\hat{\mathbf{p}}$  和  $\hat{\dot{\mathbf{p}}}$  在前文中没有出现，按照公式 14 中的记法，加上^记号表示近似值，但实际上此处并未做近似处理。四个方程分别如下：

1.

$$\frac{d}{dt} \hat{\Theta} = \mathbf{R}_z(\psi) \cdot \hat{\omega}$$

这个方程来自于公式 11，世界坐标系下身体坐标系(即刚体)的欧拉角的时间导数与角速度之间的变换关系。

2.

$$\frac{d}{dt} \hat{\mathbf{p}} = \hat{\dot{\mathbf{p}}}$$

来自于公式 6 中的第一个方程的降阶处理，将一个二阶微分方程拆成了两个一阶微分方程，分别是矩阵表达式中的第二和第四个方程。

3.

$$\frac{d}{dt} \hat{\omega} = \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_{\times} \mathbf{f}_1 + \dots + \hat{\mathbf{I}}^{-1}[\mathbf{r}_n]_{\times} \mathbf{f}_n$$

这个方程来自于公式 6 中第二个方程，公式 12 和公式 14。公式 12 中忽略了  $\omega \times (\mathbf{I}\omega)$  项，但并没有对  $\mathbf{I}$  取近似，将公式 12 中的  $\mathbf{I}$  替换为  $\hat{\mathbf{I}}$ ，再代入 6 的第二个方程有

$$\begin{aligned} \frac{d}{dt}(\hat{\mathbf{I}}\omega) &= \hat{\mathbf{I}}\dot{\omega} = \sum_{i=1}^n \mathbf{r}_i \times \mathbf{f}_i \\ \dot{\omega} &= \frac{d}{dt} \hat{\omega} = \hat{\mathbf{I}}^{-1} \left( \sum_{i=1}^n \mathbf{r}_i \times \mathbf{f}_i \right) = \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_{\times} \mathbf{f}_1 + \dots + \hat{\mathbf{I}}^{-1}[\mathbf{r}_n]_{\times} \mathbf{f}_n \end{aligned}$$

4.

$$\frac{d}{dt} \hat{\dot{\mathbf{p}}} = \mathbf{1}_3/m \cdot \mathbf{f}_1 + \dots + \mathbf{1}_3/m \cdot \mathbf{f}_n + \mathbf{g}$$

第四个方程计算如下，令：

$$\begin{aligned} \hat{\mathbf{p}} &= \mathbf{p} \\ \hat{\dot{\mathbf{p}}} &= \dot{\mathbf{p}} \end{aligned}$$

可以获得第二个方程及  $\dot{\mathbf{p}}$  的表达式：

$$\begin{aligned} \frac{d}{dt} \hat{\dot{\mathbf{p}}} &= \dot{\mathbf{p}} \\ \frac{d}{dt} \hat{\mathbf{p}} &= \ddot{\mathbf{p}} = \frac{\sum_{i=1}^n \mathbf{f}_i}{m} - \mathbf{g} \\ &= \frac{\sum_{i=1}^n \mathbf{1}_3 \cdot \mathbf{f}_i}{m} - \mathbf{g} \\ &= \mathbf{1}_3/m \cdot \mathbf{f}_1 + \dots + \mathbf{1}_3/m \cdot \mathbf{f}_n - \mathbf{g} \end{aligned}$$

上式中  $\mathbf{g}$  的符号与论文中不一致，因为论文中公式 15 与公式 6 中第一个方程的符号就不一致，应该是输入有误，不影响实际计算。

如果将重力项合并到状态向量中，那方程 15 可以写的更紧凑些，也是我们熟悉的状态空间方程的样子：

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c(\psi)\mathbf{x}(t) + \mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)\mathbf{u}(t) \quad (16)$$

其中  $\mathbf{A}_c \in \mathbb{R}^{13 \times 13}$ ， $\mathbf{B}_c \in \mathbb{R}^{13 \times 3n}$ 。这种经过近似简化处理的方程，系统变量只与偏航角和落足位置有关<sup>55</sup>。如果可以提前计算<sup>56</sup> 上面的方程，动力学(模型)就会变成线性时变的，很适合凸模型预测控制。

根据公式 16 的形式，可以计算出新的状态向量  $\mathbf{x}$ ，矩阵  $\mathbf{A}_c$  和  $\mathbf{B}_c$  如下：

$$\mathbf{x} = \begin{bmatrix} \hat{\Theta} \\ \hat{\mathbf{p}} \\ \hat{\omega} \\ \hat{\mathbf{p}} \\ 1 \end{bmatrix}_{13 \times 1} \quad \mathbf{A}_c = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & g \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{13 \times 13}$$

$$\mathbf{B}_c = \begin{bmatrix} \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_{\times} & \dots & \hat{\mathbf{I}}^{-1}[\mathbf{r}_n]_{\times} \\ \mathbf{1}_3/m & \dots & \mathbf{1}_3/m \\ 0 & \dots & 0 \end{bmatrix}_{13 \times 3n}^a$$

所以

$$\frac{d}{dt} \begin{bmatrix} \hat{\Theta} \\ \hat{\mathbf{p}} \\ \hat{\omega} \\ \hat{\mathbf{p}} \\ 1 \end{bmatrix}_{13 \times 1} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{R}_z(\psi) & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{1}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & 0 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & g \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{13 \times 13} \begin{bmatrix} \hat{\Theta} \\ \hat{\mathbf{p}} \\ \hat{\omega} \\ \hat{\mathbf{p}} \\ 1 \end{bmatrix}_{13 \times 1} +$$

$$\begin{bmatrix} \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \mathbf{0}_3 & \dots & \mathbf{0}_3 \\ \hat{\mathbf{I}}^{-1}[\mathbf{r}_1]_{\times} & \dots & \hat{\mathbf{I}}^{-1}[\mathbf{r}_n]_{\times} \\ \mathbf{1}_3/m & \dots & \mathbf{1}_3/m \\ 0 & \dots & 0 \end{bmatrix}_{13 \times 3n} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_n \end{bmatrix}_{3n \times 1}$$

<sup>a</sup> n 是外力数量，四足机器人中 n=4。

## 4 模型预测控制 MODEL PREDICTIVE CONTROL

期望的地面反作用力是由一个**离散时间有限时域模型预测控制器**<sup>57</sup> 计算得到的。因为我们考虑的是地面反作用力，而非机器人关节扭矩，所以(模型)预测控制器不需要知道腿部的构型(configuration)或者运动学(kinematics)情况<sup>58</sup>。一般来说，模型预测控制器中包含有一个被控系统的模型<sup>59</sup>，在每次迭代时，(控制器)从当前状态开始，在有限的预测时域内寻找最优的控制输入序列<sup>60</sup> 和对应的(系统)状态轨迹<sup>61</sup>，同时要满足(系统)状态轨迹约束<sup>62</sup> 和控制输入约

<sup>55</sup> This form depends only on yaw and footstep locations.

<sup>56</sup> 在实际发生前计算完成。

<sup>57</sup> a discrete-time finite-horizon model predictive controller.

<sup>58</sup> MPC 控制器决定着机器人踏步的频率，但不知道机器人腿到底在如何运动，给它的所有参数都是为了计算足底反力的。

<sup>59</sup> 这个模型可以写成很多种不同的形式，最常用的是状态空间模型(state space model)，也有人用阶跃响应模型，脉冲响应模型和传递函数模型的，参考《模型预测控制》第 6 页，陈虹著，科学出版社。Cheetah 3 中用的就是前文推导得到的单刚体状态空间模型。

<sup>60</sup> optimal sequence of control inputs. 这里的 control inputs 不是状态向量，而是系统的输入信号，即公式 15 中的  $[\mathbf{f}_1 \dots \mathbf{f}_n]^T$  足底反力向量。

<sup>61</sup> state trajectory. 系统状态轨迹指的是系统输出，一般的状态空间模型除了前面推导的状态方程外，还有一个输出方程  $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t)$ ，其中  $\mathbf{C}$  叫做输出矩阵， $\mathbf{D}$  叫做前馈矩阵。一般来说，系统没有直接从输入到输出的路径，所以前馈矩阵为零矩阵。输出矩阵是系统状态向量到输出变量的映射，在论文中没有明确定义系统的输出，只讨论了状态变量，所以可以将  $\mathbf{C}$  视为单位矩阵，那么系统状态向量就是系统的输出了。

<sup>62</sup> 系统的输出应该满足一定的约束条件。

束<sup>63</sup>。因为这个过程<sup>64</sup>在每次迭代时都会被重复，所以在控制器再次运行起来并计算出下一批最优控制序列前<sup>65</sup>只有(上次迭代计算出的最优控制序列中的)第一个时间步长上的控制输入信号会被使用。在这一节中，我们考虑一个标准形式的MPC(模型预测控制)问题，预测时域长度(horizon length)为 $k$ ：

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{i=0}^{k-1} \|\mathbf{x}_{i+1} - \mathbf{x}_{i+1, \text{ref}}\|_{\mathbf{Q}_i} + \|\mathbf{u}_i\|_{\mathbf{R}_i} \quad (17)$$

$$\text{subject to } \mathbf{x}_{i+1} = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i \mathbf{u}_i, i = 0 \dots k-1 \quad (18)$$

$$\underline{\mathbf{c}}_i \leq \mathbf{C}_i \mathbf{u}_i \leq \bar{\mathbf{c}}_i, i = 0 \dots k-1 \quad (19)$$

$$\mathbf{D}_i \mathbf{u}_i = 0, i = 0 \dots k-1 \quad (20)$$

关于公式17的说明：

式中 $\mathbf{Q}_i$ 和 $\mathbf{R}_i$ 写的像下标，而非矩阵，因为原文就是如此，正常的写法应该是：

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{i=0}^{k-1} \|\mathbf{x}_{i+1} - \mathbf{x}_{i+1, \text{ref}}\|^T \mathbf{Q}_i \|\mathbf{x}_{i+1} - \mathbf{x}_{i+1, \text{ref}}\| + \|\mathbf{u}_i\|^T \mathbf{R}_i \|\mathbf{u}_i\|$$

因为 $\|\mathbf{x}_{i+1} - \mathbf{x}_{i+1, \text{ref}}\|$ 与 $\mathbf{Q}_i$ 之间不是相乘的关系，因此故意写的像下标，这是向量加权范数的标准写法<sup>a</sup>，后文也明确说了 $\mathbf{Q}_i$ 和 $\mathbf{R}_i$ 是对角半正定的权重矩阵(矩阵所有特征值非负)。可能因为作者都来自于MIT的同一个实验室，因此使用了同样的latex模板，MIT另外两篇关于Cheetah的论文也用了同样偏数学化的写法<sup>b c</sup>，不过几乎所有讨论MPC的文献都用的是标准写法<sup>d</sup>

$$\bar{\mathbf{x}}(t)^T \mathbf{Q} \bar{\mathbf{x}}(t) + \bar{\mathbf{u}}(t)^T \mathbf{R} \bar{\mathbf{u}}(t)$$

其中 $\bar{\mathbf{x}}(t)$ 和 $\bar{\mathbf{u}}(t)$ 表示实际系统状态和输入与期望的系统状态和输入之间的偏差， $\mathbf{Q}$ 和 $\mathbf{R}$ 是权重矩阵。

- <sup>a</sup> <https://math.stackexchange.com/questions/394237/understanding-weighted-inner-product-and-weighted-norms?noredirect=1&lq=1>  
<sup>b</sup> M. Chignoli, D. Kim, E. Stanger-Jones, S. Kim. The MIT Humanoid Robot: Design, Motion Planning, and Control For Acrobatic Behaviors. ArXiv e-prints, April 2021.  
<sup>c</sup> 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 preprint arXiv:1909.06586, 2019.  
<sup>d</sup> M. Neunert, F. Farshidian, A. W. Winkler, and J. Buchli. Trajectory Optimization Through Contacts and Automatic Gait Discovery for Quadrupeds. ArXiv e-prints, July 2016.

- $\mathbf{x}_i$ ：系统在 $i$ 时刻的状态(向量)
- $\mathbf{u}_i$ ： $i$ 时刻的控制输入信号，即足底反力向量 $\mathbf{f}$
- $\mathbf{Q}_i, \mathbf{R}_i$ ：对角半正定权重矩阵<sup>66</sup>
- $\mathbf{A}_i, \mathbf{B}_i$ ：离散时间系统动力学(特性)<sup>67</sup>
- $\mathbf{C}_i, \underline{\mathbf{c}}_i$ 和 $\bar{\mathbf{c}}_i$ ：控制输入信号的不等式约束(足底反力约束)
- $\mathbf{D}_i$ ：矩阵 $\mathbf{D}_i$ 负责在时刻 $i$ 为没有与地面接触的脚选择足底反力<sup>68</sup>

记法 $\|\mathbf{a}\|_{\mathbf{S}}$ 表示加权范数 $\mathbf{a}^T \mathbf{S} \mathbf{a}$ <sup>69</sup>。这种(数学表达)形式的控制器会试图寻找一系列控制输入信号<sup>70</sup>，引导系统(的状态

<sup>63</sup> 系统的输入信号应该满足一定的约束条件。

<sup>64</sup> 在预测时域内寻找最优控制序列和对应的系统状态轨迹的过程——求解二次凸优化。

<sup>65</sup> 即下一次迭代开始前。

<sup>66</sup> 关于半正定矩阵的有趣解释：对任意非零向量做完变换后，新向量与原向量之间的夹角小于等于 $\pi/2$ 。参考：<https://zhuanlan.zhihu.com/p/44860862>

<sup>67</sup>  $\mathbf{A}_i, \mathbf{B}_i$ 就是公式16中的 $\mathbf{A}_c$ 和 $\mathbf{B}_c$ 矩阵，一般来说这个矩阵是不随时间变化的，此时系统叫做“线性时不变系统”(linear time invariant, LTI)。但很明显在 $\mathbf{A}_c$ 和 $\mathbf{B}_c$ 中的 $\psi$ 和 $\mathbf{r}$ (落足点)都是时变的，因此这个状态方程是“线性时变方程”(linear time-variant)。

<sup>68</sup> 从上一个约束可以看出 $\underline{\mathbf{c}}_i$ 和 $\bar{\mathbf{c}}_i$ 表示的是机器人可以输出的最小和最大足底反力。最小足底反力由摩擦锥确定，防止打滑；最大足底由电机确定，防止电机长时间过载。类似的，此处等式约束表示处于摆动相的腿足底反力为零。

<sup>69</sup> 加权范数是一种特殊的向量范数，也称椭圆范数。设 $\mathbf{A}$ 是任意一个 $n$ 阶对称正定矩阵，列向量 $\mathbf{x} \in \mathbb{R}^3$ ，则加权范数定义为： $\|\mathbf{x}\|_{\mathbf{A}} = (\mathbf{x}^T \mathbf{A} \mathbf{x})^{1/2}$ ，参考《矩阵论》116页，程云鹏等，西北工业大学出版社。

<sup>70</sup> 这一系列控制输入信号，也就是足底反力，是按时间排列的。假设MPC预测时域长度为 $k$ ，那么这一系列的控制输入信号为 $(t = i + 0, \mathbf{u}_0), (t = i + 1, \mathbf{u}_1), (t = i + 2, \mathbf{u}_2), \dots, (t = i + k - 1, \mathbf{u}_{k-1})$ 。



向量)沿着轨迹  $\mathbf{x}_{\text{ref}}$  前进, 用追踪精度换取控制消耗<sup>71</sup>, 同时遵从约束限制<sup>72</sup>。当系统(机器人)误差准确追踪参考轨迹时, 这通常是由于欠驱动状态下不可控制的动力学(特性)或(公式 19 和 20)约束导致的, (模型)预测控制器会在预测时域内寻找最小二乘意义上的最优解<sup>73</sup>。在考虑到未来约束<sup>74</sup>的条件下在预测时域内进行的优化, 使得(模型)预测控制器可以提前对摆动相(的运动)进行规划, 并且当身体在某些步态下始终处于欠驱动时, 例如飞奔步态(bounding), 调整身体的状态<sup>75</sup>。

注记:  $\mathbf{x}_i$  是  $13 \times 1$  的向量,  $\mathbf{u}_i$  包括四条腿的足底反力和重力项, 所以也是  $13 \times 1$  的向量,  $\mathbf{C}_i$  和  $\mathbf{D}_i$  可以是  $12 \times 13$  的矩阵, 也可以是  $13 \times 13$  的矩阵, 取决于  $\mathbf{c}_i$  和  $\mathbf{d}_i$  是 12 还是 13。

$i = 0 \dots k-1$  表示预测时域内每个时间点上都要满足约束条件, 当前时刻为  $i = 0$ 。

### A. 力约束 Force Constraints

等式约束 20 用来将离开地面的脚的足底反力置为零, (强制)执行期望的步态<sup>76</sup>。不等式约束 19 用来为每只脚在触地时添加以下 10 个不等式约束<sup>77</sup>:

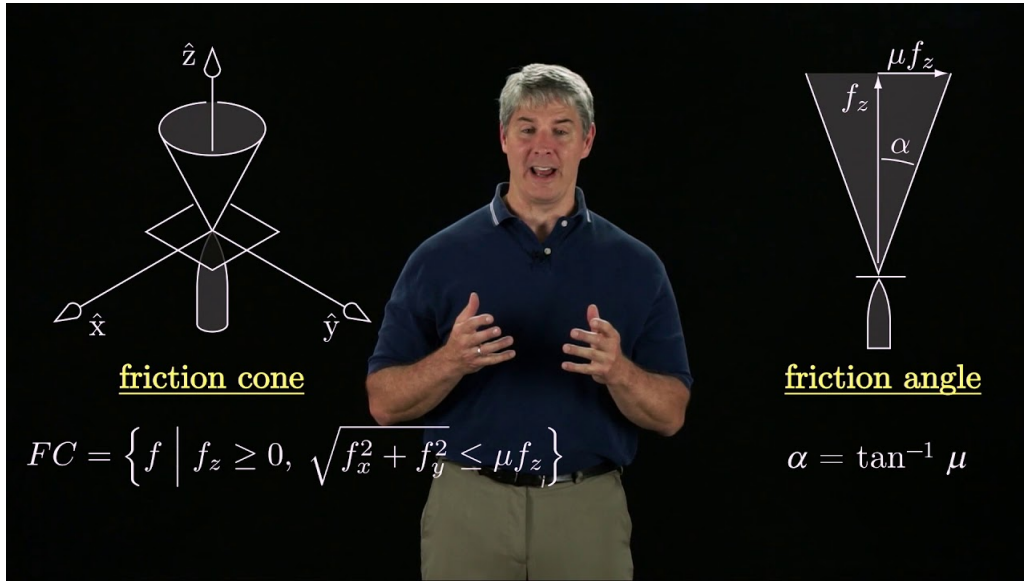
$$f_{\min} \leq f_z \leq f_{\max} \quad (21)$$

$$-\mu f_z \leq \pm f_y \leq \mu f_z \quad (22)$$

$$-\mu f_z \leq \pm f_x \leq \mu f_z \quad (23)$$

这些约束限制了  $z$  方向力的最小值和最大值, 以及摩擦锥的一个正方形金字塔的近似<sup>78</sup>。

关于摩擦锥,如下图所示:



$\alpha = \tan^{-1} \mu$  是摩擦角, 摩擦锥就是以  $\alpha$  为圆锥素线角,  $\mu f_z$  为底面半径的圆锥。  $f_x, f_y, f_z$  指的是地面反作用力的三个分量, 当地面反作用力位于摩擦锥之内时就不会发生滑动。

### B. 生成参考轨迹 Reference Trajectory Generation

期望机器人做出的行为动作构成了参考轨迹。在我们的应用中参考轨迹很简单, 只包含了非零的  $xy$ -速度,  $xy$ -位

<sup>71</sup>Control effort: The control effort is the amount of energy or power necessary for the controller to perform its duty. 参考: [https://en.wikibooks.org/wiki/Control\\_Systems/System\\_Specifications](https://en.wikibooks.org/wiki/Control_Systems/System_Specifications)

<sup>72</sup>The controller in this form attempts to find a sequence of control inputs that will guide the system along the trajectory  $\mathbf{x}_{\text{ref}}$ , trading off tracking accuracy for control effort, while obeying constraints.

<sup>73</sup>Predictive controller finds the best solution, in the least squares sense, over the prediction horizon.最小二乘意义上的最优解是什么意思?

<sup>74</sup>future constraints.

<sup>75</sup>The optimization over the horizon while taking into account future constraints enables the predictive controller to plan ahead for periods of flight and regulate the states of the body during a gait when the body is always underactuated, such as bounding.

<sup>76</sup>意思是通过令哪只脚的足底反力为零来控制脚处于支撑相还是摆动相?

<sup>77</sup>第一个不等式约束最小和最大地面正压力; 第二个和第三个不等式约束保证地面反力在摩擦锥内。

<sup>78</sup>用一个底部是正方形的金字塔近似摩擦锥。

置,  $z$  位置, 偏航角和偏航角速度<sup>79</sup>。除了偏航角和  $xy$ -位置外, 所有参数都直接由机器人操作者给予命令, 偏航角和  $xy$ -位置由适当的速度积分得到。其他的状态变量(滚转, 俯仰, 滚转角速度, 俯仰角速度和  $z$ -速度)都始终置为零。参考轨迹还用于确定动力学约束<sup>80</sup>和未来落点的位置。在实践(测试)中, 参考轨迹很短(大概0.5到0.3秒), 并且被快速重新计算(每0.05到0.03秒, 即20 Hz到33 Hz)以确保机器人受到干扰时小节3中简化的动力学(模型)依然有效。

### C. 线性离散时间系统 *Linear Discrete Time Dynamics*

对于参考轨迹中的每个点  $n$ <sup>81</sup>, 都需要根据参考轨迹和落足点控制器<sup>82</sup>中的期望值  $\psi$  和  $\mathbf{r}_i$ , 计算公式16中定义的矩阵  $\mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)$  的近似矩阵  $\hat{\mathbf{B}}_c[n] \in \mathbb{R}^{13 \times 3n}$ 。类似地, 对整条参考轨迹, 需要根据参考轨迹中  $\psi$  的平均值计算一个  $\hat{\mathbf{A}}_c \in \mathbb{R}^{13 \times 13}$  矩阵<sup>83</sup>。通过使用下面的扩展线性系统<sup>84</sup> 的状态转移矩阵, 可以将  $\mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)$  和  $\mathbf{A}_c$  矩阵转换成一个零阶保持离散时间模型<sup>85</sup>:

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \quad (24)$$

注: 公式24不是零阶保持离散时间模型, 而是将要进行离散处理的扩展线性系统模型。将公式24展开:

$$\begin{aligned} \frac{d}{dt} \mathbf{x} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} = \mathbf{A}_c(\psi)\mathbf{x}(t) + \mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)\mathbf{u}(t) \\ \frac{d}{dt} \mathbf{u} &= \mathbf{0}\mathbf{x} + \mathbf{0}\mathbf{u} \end{aligned}$$

$\mathbf{u}(t)$  是时间的函数, 那么  $\frac{d}{dt} \mathbf{u} = \mathbf{0}\mathbf{x} + \mathbf{0}\mathbf{u}$  为何能成立? 输入信号的一阶导数始终为零?

Comment by Yang 可能是因为离散信号, 都是平的。其实首先它就可不可导, 其次可导的地方都是0。只是一个猜测。

上面的简化<sup>86</sup> 使得我们可以将(机器人)动力学(状态方程)表示成离散时间形式:

$$\mathbf{x}[n+1] = \hat{\mathbf{A}}\mathbf{x}[n] + \hat{\mathbf{B}}[n]\mathbf{u}[n] \quad (25)$$

公式25中的近似<sup>87</sup> 只有当机器人可以跟随参考轨迹 ( $\mathbf{x}_{i+1, \text{ref}}$ ) 的时候才成立。当距离参考轨迹偏差较大时, 可能由外部或地形干扰引起, 会导致  $\hat{\mathbf{B}}[n]$  不准确。然而, 在第一个时间节点<sup>88</sup>,  $\hat{\mathbf{B}}[n]$  是直接根据当前机器人状态计算得到

<sup>79</sup>意思是机器人参考轨迹即是由这些信息组成的, 需要注意, 公式17中的  $\mathbf{x}_{i+1, \text{ref}}$  是系统状态向量, 包含的信息要多于参考轨迹中的信息。

<sup>80</sup>dynamics constraints.

<sup>81</sup>这里  $n$  表示预测时域中的采样时间点, 而不是公式15中那样表示腿的数量, 更好的方式是延续公式17里面的写法, 采样时间点用  $k$  表示。

<sup>82</sup>foot placement controller.

<sup>83</sup>以MIT的MPC控制器为例, 当预测时域长度为10时, 每个采样时间点都需要计算  $\hat{\mathbf{B}}_c[n]$  矩阵, 但整个预测时域内只需要计算一个  $\hat{\mathbf{A}}_c$  矩阵。

<sup>84</sup>extended linear system: 公式15中描述的是线性连续时间模型, 其中状态向量为  $\mathbf{x}$ 。在公式24描述的是扩展的线性连续时间模型, 状态向量为  $[\mathbf{x} \ \mathbf{u}]^T$ 。Yang: 这个地方我的确没见过。我想去查, 但是也没找到。我也不知道有什么好处, 除了写一块儿, 好算。Guan: 可能因为状态向量  $\mathbf{x}$  和控制输入信号  $\mathbf{u}$  都是需要优化的变量, 所以写在一起, 统称为 optimization variables。

<sup>85</sup>零阶保持模型 (Zero Order Hold, ZOH) 是由传统的数模转换器 (DAC) 完成的实际信号重建的数学模型, 也就是说, 它描述了通过将每个采样值保持一个采样间隔来将离散时间信号转换为连续时间信号的效果。其实就是连续时间信号经过采样保持电路后得到的信号, 叫零阶保持, 是指信号恢复为原信号时, 信号之间用直线连接, 即用DAC直接转换得到原信号; 类似地, 一阶保持就是在两个采样之间用插值复原, 对应的实际电路是DAC加积分电路。所以零阶保持模型就是采样后直接用DAC恢复成原始信号的离散时间模型, 一节系统就是采样后用DAC和积分电路恢复成原始信号的离散时间模型。参考: <https://www.scimall.org.cn/article/detail?id=4812400> 和维基百科: <https://zh.wikipedia.org/wiki/%E9%9B%B6%E9%9A%8E%E4%BF%9D%E6%8C%81>

零阶保持离散时间模型如公式25所示。Yang: 这里我是这么理解的, 这个话我好像也经常写。我们会说, ZOH是exact method。这个exact应该是指, 变换前后, 在离散的时间点, 两者数值是一样的。相对应的应该是tustins method, MATLAB里的话就是'c2d()'。所谓在频域里如何如何, 这个我学的也不好, 你可以自己去了解一下, 但是相对于ZOH, 在这些离散时间点上, 不保证数值相等。所以说exact是指, 至少时间点上数值保真。

<sup>86</sup>简化(simplification)是指extended linear system? Yang: 不排除是, 但是我觉得可能也是对A和B矩阵的简化。比如它们可能两个都时变, 但是现在好像只有B了。只是我的猜测。

<sup>87</sup>这里的“approximation”和上一段的“simplification”应该是不同的, 这里的近似指离散化处理, 用的是“forward difference”。Yang: 对, 这儿又来了个近似, 也不知道具体指啥。另外我想理解的是, 为什么只有跟随了才成立, 不成立会怎样? Guan: 因为  $\hat{\mathbf{A}}$  矩阵只跟偏航角  $\psi$  有关, 在一次MPC迭代(至多40ms)中, 偏航角改变很小, 所以  $\hat{\mathbf{A}}$  可视为不变;  $\hat{\mathbf{B}}$  除了偏航角, 还跟  $(\mathbf{r}_1, \dots, \mathbf{r}_n)$  有关,  $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$  分别是质心指向四只脚所受的地面反力的作用点, 在一次MPC迭代中, 地面反力作用点应该是不变的, 但机器人质心会变化, 因此在预测时域的每个time step中,  $(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4)$  都会变化。

<sup>88</sup>MPC预测时域内的第一个时间节点, 也就是当前时刻,  $i=0$ 。

的，并且总是正确的。假设在某个时刻，机器人追踪参考轨迹时受到了干扰，那么下一次 MPC 的迭代时（这发生在距离机器人受到干扰后至多 40 ms），将基于受到干扰的机器人状态重新计算参考轨迹，从而补偿受到的干扰<sup>89</sup>。

计算  $\hat{\mathbf{A}}$  和  $\hat{\mathbf{B}}[n]$ ，考虑方程：

$$\frac{d}{dt}\mathbf{x} = \dot{\mathbf{x}}(t) = \mathbf{A}_c(\psi)\mathbf{x}(t) + \mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)\mathbf{u}(t)$$

因为前文说过，在预测时域内，假设  $\mathbf{A}_c(\psi)$  矩阵是常数，不随时间变化，而  $\mathbf{B}_c(\mathbf{r}_1, \dots, \mathbf{r}_n, \psi)\mathbf{u}(t)$  矩阵在每一步迭代时都会发生变化。因此，可以重写方程为：

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c\mathbf{x}(t) + \mathbf{B}_c(t)\mathbf{u}(t)$$

假设  $t = nT$ ，使用前向差分法(forward difference)代替时间导数：

$$\begin{aligned}\dot{\mathbf{x}}(t) \Big|_{t=nT} &= \frac{1}{T} \{ \mathbf{x}[[n+1]T] - \mathbf{x}[nT] \} \\ \Rightarrow \frac{1}{T} \{ \mathbf{x}[[n+1]T] - \mathbf{x}[nT] \} &= \mathbf{A}_c\mathbf{x}[nT] + \mathbf{B}_c[nT]\mathbf{u}[nT] \\ \Rightarrow \mathbf{x}[[n+1]T] &= \mathbf{x}[nT] + T \cdot \mathbf{A}_c\mathbf{x}[nT] + T \cdot \mathbf{B}_c[nT]\mathbf{u}[nT] \\ &= (\mathbf{1} + T \cdot \mathbf{A}_c)\mathbf{x}[nT] + T \cdot \mathbf{B}_c[nT]\mathbf{u}[nT]\end{aligned}$$

所以有：

$$\begin{aligned}\hat{\mathbf{A}} &= \mathbf{1} + T \cdot \mathbf{A}_c \\ \hat{\mathbf{B}}[nT] &= T \cdot \mathbf{B}_c[nT]\end{aligned}$$

- $T$ ：采样周期
- $\mathbf{1}$ ：单位矩阵，此处跟上文保持一致， $\mathbf{I}$  表示惯性张量，所以用  $\mathbf{1}$  表示单位矩阵。另外，因为  $\mathbf{A}_c$  是  $13 \times 13$  的矩阵，因此这里用单位矩阵没问题

当采样周期  $T = 1$  时，有：

$$\begin{aligned}\mathbf{x}[n+1] &= (\mathbf{1} + \mathbf{A}_c)\mathbf{x}[n] + \mathbf{B}_c[n]\mathbf{u}[n] \\ \hat{\mathbf{A}} &= \mathbf{1} + \mathbf{A}_c \\ \hat{\mathbf{B}}[n] &= \mathbf{B}_c[n]\end{aligned}$$

!!!注意这里的  $n$  是预测时域中的采样周期数，即 time step 的步数，而小节 3 里面公式 15 的  $n$  表示足底反力数量。

为何  $\hat{\mathbf{A}}$  在一次 MPC 计算周期内不需要变化，而  $\hat{\mathbf{B}}[n]$  在预测时域的每个采样时间点都需要变化？如果不变化，是不是就是令梯度为零，求出的 patch solution？如果每次都变化，就是用 Riccati iterations 求解？

by Yang: 中间有段“当采样周期  $T=1$  时”，真的是这样吗？我觉得这个推导不该是关于某个特殊采样周期吧。我的理解是  $[n]$  只是个缩略写法，或者在离散里，再写  $T$  就没什么必要了。这个  $n$  实际上是第  $n$  步，对应的时间点是  $t = nT$ 。以上我说的这些未必跟你现在写的冲突，但是我只是觉得  $[]$  里边可能就不该有  $T$ ，可能  $()$  里边是可以有  $nT$  的。

Guan: 对的，是这样。

#### D. 二次规划形式 QP Formulation

对公式 17 中的优化问题做重新整理，以降低问题的规模<sup>90</sup>。虽然(直接)搞一个或找一个可以利用我们问题稀疏性和结构(特点)的求解器可能是最快的解决方案，但忽略(问题的)稀疏性和结构(特点)而致力于降低问题的规模，会是更直接和足够快的办法<sup>91</sup>。对于一个没有利用问题稀疏性的求解器，公式 17 的求解相对于预测时域长度( $k$ )，状态数(包

<sup>89</sup> 这一句话的意思是，如果距离参考轨迹偏差太大，会导致  $\hat{\mathbf{B}}[n]$  不准确，但因为起始采样时间点的  $\hat{\mathbf{B}}[n]$  一定是准确的，所以后续计算出来的  $\hat{\mathbf{B}}[n]$  矩阵也一定是准确的。假设在某次迭代计算 MPC 后机器人受到了干扰，那么机器人是无法及时察觉出来的，要等到下一次迭代时才能根据受到干扰后的状态重新规划轨迹，因为迭代间隔很短，不到 40 ms，所以可以补偿扰动。

<sup>90</sup> 在这一小节进行重整？因为前文并没有对公式 17 做过什么改变

<sup>91</sup> “直接”是指直观的想法，“足够快”指搞一个求解器太慢了，很浪费时间，而降低问题规模会快很多，更容易搞定。

括控制输入和状态轨迹)<sup>92</sup> 和约束数量(包括力约束和动力学约束)具有立方时间复杂度( $\mathcal{O}^3$ ), 因此, 我们可以藉由移除约束中的动力学约束, 和优化变量<sup>93</sup>中的状态轨迹来显著提升求解速度, 动力学约束和状态轨迹可以放到罚函数<sup>94</sup>中。这个(重新)整理的方法还允许我们忽略一些已经由公式 20 约束为零的次要的优化变量, 这样我们只需要对处于支撑相的脚的足底反力进行优化即可。上面这些讨论都是基于文献 29<sup>95</sup> 中讨论的紧凑形式的公式, 但(在我们这里它还)包含了时变动力学和状态参考(轨迹)<sup>96</sup>。

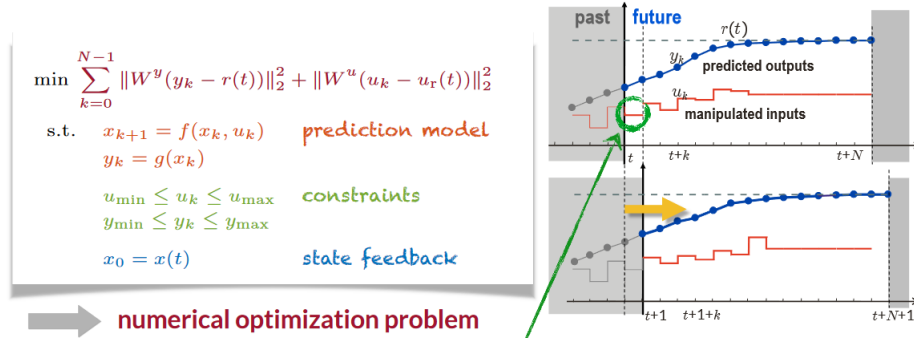
将动力学方程写成紧凑的形式:

$$\mathbf{X} = \mathbf{A}_{qp}\mathbf{x}_0 + \mathbf{B}_{qp}\mathbf{U} \quad (26)$$

- $\mathbf{X} \in \mathbb{R}^{13k}$ : 预测时域内所有状态向量的集合
- $\mathbf{U} \in \mathbb{R}^{3nk}$ : 预测时域内所有控制输入向量的集合

关于  $\mathbf{X} \in \mathbb{R}^{13k}$  和  $\mathbf{U} \in \mathbb{R}^{3nk}$  的解释, 如下图所示:

- **Goal:** find the best control sequence over a future horizon of  $N$  steps



- **At each time  $t$ :**
  - get new measurements to update the estimate of the current state  $x(t)$
  - solve the optimization problem with respect to  $\{u_0, \dots, u_{N-1}\}$
  - apply only the first optimal move  $u(t) = u_0^*$ , discard the remaining samples

假设预测时域长度为  $k = 10$ , 那么当 MPC 在  $\delta t$  内完成求解后, 会得到 10 个控制输入向量 ( $\mathbf{u}_0 \dots \mathbf{u}_9$ ) 和 10 个系统状态向量 ( $\mathbf{x}_1 \dots \mathbf{x}_{10}$ )。之所以  $\mathbf{u}$  从 0 开始, 而  $\mathbf{x}$  从 1 开始, 因为当前时刻系统状态向量  $\mathbf{x}_0$  是已知的, 但还没有输入信号  $\mathbf{u}_0$ 。实际上  $\mathbf{u}_0$  就是这次求解的最终目标, 也是执行完本次 MPC 计算后真正执行的控制输入。当执行完  $\mathbf{u}_0$  后, MPC 就会继续迭代一次, 重新计算出 10 个控制输入向量, 然后再取新的  $\mathbf{u}_0$  来执行... 这样持续下去。很明显, 求解 MPC 耗时  $\delta t$  必须足够短才行, 至少要远远短于一个 time step ( $\Delta t$ ) 的长度, 因为每隔一个 time step, MPC 就会迭代一次, 所以实际上 MPC 的迭代周期  $T$  就是  $\Delta t$ 。另外, 因为  $\hat{\mathbf{A}}(\psi)$  矩阵在一次 MPC 计算中保持不变, 所以迭代周期  $T(\Delta t)$  也必须足够短才行。总结成关系式:

$$\text{MPC 求解时间 } \delta t \ll \text{time step } \Delta t = T$$

上面的  $\mathbf{X}$ , 就是一次 MPC 求解后所有状态向量的集合, 合并成了一个向量, 所以  $\mathbf{X}$  的长度为  $13k$ 。类似地, 预测时域内所有控制输入向量合并起来的维度是: 时域长度  $k \times$  每个控制输入向量包含的足底反力数量  $n \times$  每个

<sup>92</sup>根据 MPC 理论, 在预测时域内的每个采样时间点, 都需要计算系统的控制输入  $\mathbf{u}$ ,  $\mathbf{x}$  状态向量和系统输出。在这里我们不考虑系统输出, 因此只需要计算控制输入和状态向量即可。

<sup>93</sup>待补充: 何为将状态轨迹从优化变量中移除?

<sup>94</sup>so we achieve a significant speedup by removing the dynamics constraints and state trajectory from the constraints and optimization variables and include them in the cost function instead.

<sup>95</sup>J. L. Jerez, E. C. Kerrigan, and G. A. Constantinides, "A condensed and sparse qp formulation for predictive control," in 2011 50th IEEE Conference on Decision and Control and European Control Conference, Dec 2011, pp. 5217–5222.

<sup>96</sup>This is based on the condensed formulation discussed in [29], but includes time-varying dynamics and state reference.

足底反力的维度 3，即  $3nk$ 。

$$\mathbf{X}_{13k \times 1} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_k \end{bmatrix} \quad \mathbf{U}_{3nk \times 1} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{k-1} \end{bmatrix}$$

所以公式 26 中各矩阵的维度为：

$$\mathbf{X}_{13k \times 1} = \mathbf{A}_{\text{qp}13k \times 13k} \mathbf{x}_{013k \times 1} + \mathbf{B}_{\text{qp}13k \times 3nk} \mathbf{U}_{3nk \times 1}$$

现在尝试计算  $\mathbf{A}_{\text{qp}13k \times 13k}$  和  $\mathbf{B}_{\text{qp}13k \times 3nk}$  的表达式，根据 25 的递推关系式有：

$$\begin{aligned} \mathbf{x}[1] &= \hat{\mathbf{A}}\mathbf{x}[0] + \hat{\mathbf{B}}[0]\mathbf{u}[0] \\ \mathbf{x}[2] &= \hat{\mathbf{A}}\mathbf{x}[1] + \hat{\mathbf{B}}[1]\mathbf{u}[1] \\ &= \hat{\mathbf{A}}^2\mathbf{x}[0] + \hat{\mathbf{A}}\hat{\mathbf{B}}[0]\mathbf{u}[0] + \hat{\mathbf{B}}[1]\mathbf{u}[1] \\ \mathbf{x}[3] &= \hat{\mathbf{A}}\mathbf{x}[2] + \hat{\mathbf{B}}[2]\mathbf{u}[2] \\ &= \hat{\mathbf{A}}^3\mathbf{x}[0] + \hat{\mathbf{A}}^2\hat{\mathbf{B}}[0]\mathbf{u}[0] + \hat{\mathbf{A}}\hat{\mathbf{B}}[1]\mathbf{u}[1] + \hat{\mathbf{B}}[2]\mathbf{u}[2] \\ &\dots \\ \mathbf{x}[k-1] &= \hat{\mathbf{A}}\mathbf{x}[k-2] + \hat{\mathbf{B}}[k-2]\mathbf{u}[k-2] \\ &= \hat{\mathbf{A}}^{k-1}\mathbf{x}[0] + \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[0]\mathbf{u}[0] + \hat{\mathbf{A}}^{k-3}\hat{\mathbf{B}}[1]\mathbf{u}[1] + \dots + \hat{\mathbf{B}}[k-2]\mathbf{u}[k-2] \\ \mathbf{x}[k] &= \hat{\mathbf{A}}\mathbf{x}[k-1] + \hat{\mathbf{B}}[k-1]\mathbf{u}[k-1] \\ &= \hat{\mathbf{A}}^k\mathbf{x}[0] + \hat{\mathbf{A}}^{k-1}\hat{\mathbf{B}}[0]\mathbf{u}[0] + \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[1]\mathbf{u}[1] + \dots + \hat{\mathbf{A}}\hat{\mathbf{B}}[k-2]\mathbf{u}[k-2] + \hat{\mathbf{B}}[k-1]\mathbf{u}[k-1] \end{aligned}$$

将上面所有公式合并在一起，参考公式 26 写成矩阵形式：

$$\begin{bmatrix} \mathbf{x}[1] \\ \mathbf{x}[2] \\ \mathbf{x}[3] \\ \vdots \\ \mathbf{x}[k-1] \\ \mathbf{x}[k] \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}}^2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{A}}^3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \hat{\mathbf{A}}^{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \hat{\mathbf{A}}^k \end{bmatrix} \begin{bmatrix} \mathbf{x}[0] \\ \mathbf{x}[0] \\ \mathbf{x}[0] \\ \vdots \\ \mathbf{x}[0] \\ \mathbf{x}[0] \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{B}}[0] & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{A}}\hat{\mathbf{B}}[0] & \hat{\mathbf{B}}[1] & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{A}}^2\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}\hat{\mathbf{B}}[1] & \hat{\mathbf{B}}[2] & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}^{k-3}\hat{\mathbf{B}}[1] & \hat{\mathbf{A}}^{k-4}\hat{\mathbf{B}}[2] & \dots & \hat{\mathbf{B}}[k-2] & \mathbf{0} \\ \hat{\mathbf{A}}^{k-1}\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[1] & \hat{\mathbf{A}}^{k-3}\hat{\mathbf{B}}[2] & \dots & \hat{\mathbf{A}}\hat{\mathbf{B}}[k-2] & \hat{\mathbf{B}}[k-1] \end{bmatrix} \begin{bmatrix} \mathbf{u}[0] \\ \mathbf{u}[1] \\ \mathbf{u}[2] \\ \vdots \\ \mathbf{u}[k-2] \\ \mathbf{u}[k-1] \end{bmatrix}$$

$$\mathbf{X}_{13k \times 1} = \mathbf{A}_{\text{qp}13k \times 13k} \mathbf{x}_{013k \times 1} + \mathbf{B}_{\text{qp}13k \times 3nk} \mathbf{U}_{3nk \times 1}$$

所以:

$$\mathbf{A}_{\text{qp}13k \times 13k} = \begin{bmatrix} \hat{\mathbf{A}} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{A}}^2 & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \hat{\mathbf{A}}^3 & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \hat{\mathbf{A}}^{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \hat{\mathbf{A}}^k \end{bmatrix}$$

$$\mathbf{B}_{\text{qp}13k \times 3nk} = \begin{bmatrix} \hat{\mathbf{B}}[0] & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{A}}\hat{\mathbf{B}}[0] & \hat{\mathbf{B}}[1] & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ \hat{\mathbf{A}}^2\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}\hat{\mathbf{B}}[1] & \hat{\mathbf{B}}[2] & \dots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}^{k-3}\hat{\mathbf{B}}[1] & \hat{\mathbf{A}}^{k-4}\hat{\mathbf{B}}[2] & \dots & \hat{\mathbf{B}}[k-2] & \mathbf{0} \\ \hat{\mathbf{A}}^{k-1}\hat{\mathbf{B}}[0] & \hat{\mathbf{A}}^{k-2}\hat{\mathbf{B}}[1] & \hat{\mathbf{A}}^{k-3}\hat{\mathbf{B}}[2] & \dots & \hat{\mathbf{A}}\hat{\mathbf{B}}[k-2] & \hat{\mathbf{B}}[k-1] \end{bmatrix}$$

最小化参考轨迹偏离量的加权最小二乘<sup>97</sup> 和加权(地面反作用)力幅值(大小)<sup>98</sup> 的目标函数为:

$$J(\mathbf{U}) = \|\mathbf{A}_{\text{qp}}\mathbf{x}_0 + \mathbf{B}_{\text{qp}}\mathbf{U} - \mathbf{x}_{\text{ref}}\|_{\mathbf{L}} + \|\mathbf{U}\|_{\mathbf{K}} \quad (27)$$

- $\mathbf{L} \in \mathbb{R}^{13k \times 13k}$ : 状态偏移量(实际状态与参考状态的偏差)对角权重矩阵
- $\mathbf{K} \in \mathbb{R}^{3nk \times 3nk}$ : 力幅值(足底反力大小)对角权重矩阵
- $\mathbf{U}, \mathbf{X}$ : 与公式 26 中相同<sup>99</sup>

在机器人上, 我们使用的是平权的力幅值对角权重矩阵  $\mathbf{K} = \alpha \mathbf{1}_{3nk}^{100}$ 。

公式 27 使用了跟公式 17 一样的向量加权范数写法, 全部展开是:

$$J(\mathbf{U}) = \|\mathbf{A}_{\text{qp}}\mathbf{x}_0 + \mathbf{B}_{\text{qp}}\mathbf{U} - \mathbf{x}_{\text{ref}}\|_{1 \times 13k}^T \mathbf{L}_{13k \times 13k} \|\mathbf{A}_{\text{qp}}\mathbf{x}_0 + \mathbf{B}_{\text{qp}}\mathbf{U} - \mathbf{x}_{\text{ref}}\|_{13k \times 1} + \|\mathbf{U}\|_{1 \times 3nk}^T \mathbf{K}_{3nk \times 3nk} \|\mathbf{U}\|_{3nk \times 1}$$

结合公式 15 和 26, 将上述表达式写出来为:

$$\|\mathbf{A}_{\text{qp}}\mathbf{x}_0 + \mathbf{B}_{\text{qp}}\mathbf{U} - \mathbf{x}_{\text{ref}}\|_{13k \times 1} = \begin{bmatrix} \hat{\Theta}_1 - \hat{\Theta}_{1,\text{ref}} \\ \hat{\mathbf{p}}_1 - \hat{\mathbf{p}}_{1,\text{ref}} \\ \hat{\omega}_1 - \hat{\omega}_{1,\text{ref}} \\ \hat{\mathbf{p}}_1 - \hat{\mathbf{p}}_{1,\text{ref}} \\ 1 - 1 \\ \hat{\Theta}_2 - \hat{\Theta}_{2,\text{ref}} \\ \hat{\mathbf{p}}_2 - \hat{\mathbf{p}}_{2,\text{ref}} \\ \hat{\omega}_2 - \hat{\omega}_{2,\text{ref}} \\ \hat{\mathbf{p}}_2 - \hat{\mathbf{p}}_{2,\text{ref}} \\ 1 - 1 \\ \vdots \\ \hat{\Theta}_k - \hat{\Theta}_{k,\text{ref}} \\ \hat{\mathbf{p}}_k - \hat{\mathbf{p}}_{k,\text{ref}} \\ \hat{\omega}_k - \hat{\omega}_{k,\text{ref}} \\ \hat{\mathbf{p}}_k - \hat{\mathbf{p}}_{k,\text{ref}} \\ 1 - 1 \end{bmatrix}_{13k \times 1}$$

<sup>97</sup>weighted least-squares deviation from the reference trajectory.

<sup>98</sup>weighted force magnitude.

<sup>99</sup>这里并未直接涉及到  $\mathbf{X}$ 。

<sup>100</sup> $\alpha$  的含义是? 力缩放系数? 还是保证对角线元素和为 1?



$$\mathbf{L}_{13k \times 13k} = \begin{bmatrix} l_{\Theta_1} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & l_{\mathbf{p}_1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & l_{\omega_1} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & l_{\dot{\mathbf{p}}_1} & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ l_{\Theta_1} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & l_{\Theta_k} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & l_{\mathbf{p}_k} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & l_{\omega_k} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & l_{\dot{\mathbf{p}}_k} & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{13k \times 13k}$$

很明显,  $J(\mathbf{U})$  是一个从向量到标量的映射。

现在有了罚函数, 问题就可以写成二次规划的形式了:

$$\min_{\mathbf{U}} \quad \frac{1}{2} \mathbf{U}^T \mathbf{H} \mathbf{U} + \mathbf{U}^T \mathbf{g} \quad (28)$$

$$\text{s. t.} \quad \underline{\mathbf{c}} \leq \mathbf{C} \mathbf{U} \leq \bar{\mathbf{c}} \quad (29)$$

$$\mathbf{H} = 2 (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} + \mathbf{K}) \quad (30)$$

$$\mathbf{g} = 2 \mathbf{B}_{qp}^T \mathbf{L} (\mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{y}) \quad (31)$$

推导  $\mathbf{H}$  和  $\mathbf{g}$  的表达式:

$$\begin{aligned} J(\mathbf{U}) &= \|\mathbf{A}_{qp} \mathbf{x}_0 + \mathbf{B}_{qp} \mathbf{U} - \mathbf{x}_{ref}\|_{1 \times 13k}^T \mathbf{L}_{13k \times 13k} \|\mathbf{A}_{qp} \mathbf{x}_0 + \mathbf{B}_{qp} \mathbf{U} - \mathbf{x}_{ref}\|_{13k \times 1} + \|\mathbf{U}\|_{1 \times 3nk}^T \mathbf{K}_{3nk \times 3nk} \|\mathbf{U}\|_{3nk \times 1} \\ &= (\mathbf{x}_0^T \mathbf{A}_{qp}^T + \mathbf{U}^T \mathbf{B}_{qp}^T - \mathbf{x}_{ref}^T) \mathbf{L} (\mathbf{A}_{qp} \mathbf{x}_0 + \mathbf{B}_{qp} \mathbf{U} - \mathbf{x}_{ref}) + \mathbf{U}^T \mathbf{K} \mathbf{U} \\ &= \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 + \cancel{\mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{B}_{qp} \mathbf{U}} - \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{x}_{ref} + \cancel{\mathbf{U}^T \mathbf{B}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0} + \cancel{\mathbf{U}^T \mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} \mathbf{U}} - \cancel{\mathbf{U}^T \mathbf{B}_{qp}^T \mathbf{L} \mathbf{x}_{ref}} \\ &\quad - \mathbf{x}_{ref}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 - \cancel{\mathbf{x}_{ref}^T \mathbf{L} \mathbf{B}_{qp} \mathbf{U}} + \mathbf{x}_{ref}^T \mathbf{L} \mathbf{x}_{ref} + \mathbf{U}^T \mathbf{K} \mathbf{U} \\ &= \frac{1}{2} \cdot 2 \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} + \mathbf{K}) \mathbf{U} + \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{B}_{qp}^T \mathbf{L} \mathbf{x}_{ref}) + (\mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{B}_{qp} - \mathbf{x}_{ref}^T \mathbf{L} \mathbf{B}_{qp}) \mathbf{U} \\ &\quad + \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{x}_{ref} - \mathbf{x}_{ref}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 + \mathbf{x}_{ref}^T \mathbf{L} \mathbf{x}_{ref} \end{aligned}$$

证明:

$$\begin{aligned} &[(\mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{B}_{qp} - \mathbf{x}_{ref}^T \mathbf{L} \mathbf{B}_{qp}) \mathbf{U}]^T \\ &= \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L}^T \mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{B}_{qp}^T \mathbf{L}^T \mathbf{x}_{ref}) \\ &\Rightarrow \mathbf{L} = \mathbf{L}^T \\ &= \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{B}_{qp}^T \mathbf{L} \mathbf{x}_{ref}) \end{aligned}$$

所以:

$$\begin{aligned} J(\mathbf{U}) &= \frac{1}{2} \cdot 2 \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} + \mathbf{K}) \mathbf{U} + \mathbf{U}^T 2 \mathbf{B}_{qp}^T \mathbf{L} (\mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{x}_{ref}) \\ &\quad + \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{x}_0^T \mathbf{A}_{qp}^T \mathbf{L} \mathbf{x}_{ref} - \mathbf{x}_{ref}^T \mathbf{L} \mathbf{A}_{qp} \mathbf{x}_0 + \mathbf{x}_{ref}^T \mathbf{L} \mathbf{x}_{ref} \end{aligned}$$

因为进行二次规划求解前  $\mathbf{x}_0$ ,  $\mathbf{A}_{qp}$ ,  $\mathbf{L}$  和  $\mathbf{x}_{ref}$  均已知, 所以蓝色部分是常数项, 不影响最终的极值, 因此可以省略:

$$J(\mathbf{U}) = \frac{1}{2} \cdot 2 \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} + \mathbf{K}) \mathbf{U} + \mathbf{U}^T 2 \mathbf{B}_{qp}^T \mathbf{L} (\mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{x}_{ref})$$

上式中的  $2 \mathbf{U}^T (\mathbf{B}_{qp}^T \mathbf{L} \mathbf{B}_{qp} + \mathbf{K})$  即是公式 30 中的表达式。  $2 \mathbf{B}_{qp}^T \mathbf{L} (\mathbf{A}_{qp} \mathbf{x}_0 - \mathbf{x}_{ref})$  应该是公式 31 中的表达式, 但论文中用了一个没有任何说明的符号  $\mathbf{y}$ , 可以合理猜测是作者笔误,  $\mathbf{y}$  即是  $\mathbf{x}_{ref}$ 。

期望的足底反力即是矩阵  $\mathbf{U}$  中的前  $3n$  个元素。另外注意到  $\mathbf{H} \in \mathbb{R}^{3nk \times 3nk}$  和  $\mathbf{g} \in \mathbb{R}^{3nk \times 1}$  的所有维度都与状态(向量)的数量无关, 只与脚的数量  $n$  和预测时域的长度  $k$  有关。

公式 28 符号的说明:

因为  $\mathbf{U}$  是  $3nk \times 1$  的向量, 并且含义是预测时域内每个采样时刻的控制输入, 所以不难猜测  $\underline{\mathbf{c}}$  和  $\bar{\mathbf{c}}$  是每个采样时刻的力约束。公式 20-22 这 10 个不等式可以让一只脚处在摩擦锥内, 将其扩展到整个预测时域内时, 其维度等于: 每只脚约束的个数  $\times$  脚的数量  $(n) \times$  预测时域长度  $(k)$ 。对 10 个不等式进行整理有:

$$\begin{aligned}
 & f_{\min} \leq f_z \leq f_{\max} \\
 & \begin{aligned}
 & f_{\min} \leq f_z \leq f_{\max} & -\mu f_z \leq f_y \leq \mu f_z & f_{\min} \leq f_z \leq f_{\max} \\
 & -\mu f_z \leq \pm f_y \leq \mu f_z \Rightarrow & -\mu f_z \leq -f_y \leq \mu f_z \Rightarrow & -\mu f_z \leq f_y \leq \mu f_z \\
 & -\mu f_z \leq \pm f_x \leq \mu f_z & -\mu f_z \leq f_x \leq \mu f_z & -\mu f_z \leq f_x \leq \mu f_z \\
 & & -\mu f_z \leq -f_x \leq \mu f_z & 
 \end{aligned}
 \end{aligned}$$

显然, 上面不等式中两个红色的不等式是一样的, 两个蓝色不等式也是一样的, 所以可以将 10 个约束缩减为 6 个。把预测时域内所有约束写在一起, 如下所示, 第一个脚标是预测时域内的采样时间点, 第二个脚标是腿编号:

$$\begin{bmatrix} f_{1,1,\min} \\ -\mu f_{1,1,z} \\ -\mu f_{1,1,z} \\ f_{1,2,\min} \\ -\mu f_{1,2,z} \\ -\mu f_{1,2,z} \\ \vdots \\ f_{1,n,\min} \\ -\mu f_{1,n,z} \\ -\mu f_{1,n,z} \\ \vdots \\ f_{k,n,\min} \\ -\mu f_{k,n,z} \\ -\mu f_{k,n,z} \end{bmatrix}_{3nk \times 1} \leq \begin{bmatrix} f_{1,1,z} \\ f_{1,1,y} \\ f_{1,1,x} \\ f_{1,2,z} \\ f_{1,2,y} \\ f_{1,2,x} \\ \vdots \\ f_{1,n,z} \\ f_{1,n,y} \\ f_{1,n,x} \\ \vdots \\ f_{k,n,z} \\ f_{k,n,y} \\ f_{k,n,x} \end{bmatrix}_{3nk \times 1} \leq \begin{bmatrix} f_{1,1,\max} \\ \mu f_{1,1,z} \\ \mu f_{1,1,z} \\ f_{1,2,\max} \\ \mu f_{1,2,z} \\ \mu f_{1,2,z} \\ \vdots \\ f_{1,n,\max} \\ \mu f_{1,n,z} \\ \mu f_{1,n,z} \\ \vdots \\ f_{k,n,\max} \\ \mu f_{k,n,y} \\ \mu f_{k,n,x} \end{bmatrix}_{3nk \times 1}$$

因此公式 29 的维度为:

$$\underline{\mathbf{c}}_{3nk \times 1} \leq \mathbf{C}_{3nk \times 3nk} \mathbf{U}_{3nk \times 1} \leq \bar{\mathbf{c}}_{3nk \times 1}$$

写成矩阵形式为:

$$\begin{bmatrix} f_{1,1,\min} \\ -\mu f_{1,1,z} \\ -\mu f_{1,1,z} \\ f_{1,2,\min} \\ -\mu f_{1,2,z} \\ -\mu f_{1,2,z} \\ \vdots \\ f_{1,n,\min} \\ -\mu f_{1,n,z} \\ -\mu f_{1,n,z} \\ \vdots \\ f_{k,n,\min} \\ -\mu f_{k,n,z} \\ -\mu f_{k,n,z} \end{bmatrix}_{3nk \times 1} \leq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3nk \times 3nk} \begin{bmatrix} f_{1,1,z} \\ f_{1,1,y} \\ f_{1,1,x} \\ f_{1,2,z} \\ f_{1,2,y} \\ f_{1,2,x} \\ \vdots \\ f_{1,n,z} \\ f_{1,n,y} \\ f_{1,n,x} \\ \vdots \\ f_{k,n,z} \\ f_{k,n,y} \\ f_{k,n,x} \end{bmatrix}_{3nk \times 1} \leq \begin{bmatrix} f_{1,1,\max} \\ \mu f_{1,1,z} \\ \mu f_{1,1,z} \\ f_{1,2,\max} \\ \mu f_{1,2,z} \\ \mu f_{1,2,z} \\ \vdots \\ f_{1,n,\max} \\ \mu f_{1,n,z} \\ \mu f_{1,n,z} \\ \vdots \\ f_{k,n,\max} \\ \mu f_{k,n,y} \\ \mu f_{k,n,x} \end{bmatrix}_{3nk \times 1}$$

## 5 结果 RESULTS

Table 1: 控制器设置和机器人数据

$m$ (机器人质量)	43 kg	$\Theta$ 权重	1
$I_{xx}$ (X轴转动惯量)	0.41 kgm <sup>2</sup>	$z$ 权重	50
$I_{yy}$ (Y轴转动惯量)	2.1 kgm <sup>2</sup>	偏航角速度(yaw rate) 权重	1
$I_{zz}$ (Z轴转动惯量)	2.1 kgm <sup>2</sup>	$\mathbf{v}$ 权重	1
$\mu$ (地面摩擦系数)	0.6	力权重 $\alpha$	$1 \times 10^{-6}$
$g_z$	-9.8 m/s <sup>2</sup>	$\mathbf{f}_{\min}$ (最小足底正压力)	10 N
$\tau_{\max}$ (关节极限扭矩)	250 Nm	$\mathbf{f}_{\max}$ (最大足底正压力)	666 N

### A. 实验设置 *Experimental Setup*

上面的控制器已经成功部署在 MIT Cheetah 3 机器人上，预测时域长度等于一个步态周期<sup>101</sup>，大概 0.33 到 0.5 秒，根据不同的步态类型预测时域中的时间步长数为 10 到 16 个<sup>102</sup>。根据选择的步态，新的预测(MPC迭代)计算频率为 25 到 50 Hz。

因为每执行完一个时间步长，MPC 就需要重新迭代计算，因此 MPC 迭代频率有以下四种情况：

$$\begin{aligned} \text{步态周期} = 0.33 \text{ s} & \begin{cases} \text{① } 10 \text{ time steps} = 0.033 \text{ s} \Rightarrow 30.303 \text{ Hz} \\ \text{② } 16 \text{ time steps} = \frac{0.33}{16} \text{ s} \Rightarrow \mathbf{48.48 \text{ Hz}} \end{cases} \\ \text{步态周期} = 0.5 \text{ s} & \begin{cases} \text{③ } 10 \text{ time steps} = 0.05 \text{ s} \Rightarrow \mathbf{20 \text{ Hz}} \\ \text{④ } 16 \text{ time steps} = \frac{0.5}{16} \text{ s} \Rightarrow 32 \text{ Hz} \end{cases} \end{aligned}$$

参考轨迹和期望步态是根据一个视频游戏控制手柄的输入信号生成的。上表中包括了(模型)预测控制器使用的所有参数。

优化计算得到的结果不会经过滤波，而是直接根据公式 5 计算关节扭矩。状态估计，摆动腿规划<sup>103</sup>和腿的阻抗控制<sup>104</sup>运行频率均为 1 kHz。

落足点的位置在世界坐标系的  $xy$  平面上，由一个简单的启发式算法计算得到，灵感来自于 M.H.Raibert<sup>105</sup>：

$$\mathbf{p}^{\text{des}} = \mathbf{p}^{\text{ref}} + \mathbf{v}^{\text{CoM}} \Delta t / 2 \quad (32)$$

- $\Delta t$ ：脚在地面上的时间(脚处于支撑相的时长)
- $\mathbf{p}^{\text{ref}}$ ：髋关节在地面的投影点<sup>106</sup>
- $\mathbf{v}^{\text{CoM}}$ ：机器人质心速度在  $xy$  平面上的投影

状态估计器用来确定当前状态下的这些值<sup>107</sup>，参考轨迹用来确定将来这量未来的值<sup>108</sup>。 $\mathbf{p}^{\text{des}}$  既被用于摆动腿控制器，也用来为模型预测控制器计算  $\mathbf{r}_i$ 。

<sup>101</sup> one gait cycle. 一个摆动相和一个支撑相。

<sup>102</sup> 10~16 time steps.

<sup>103</sup> swing leg planning.

<sup>104</sup> leg impedance control.

<sup>105</sup> M. H. Raibert, Legged Robots That Balance. Cambridge, MA, USA: Massachusetts Institute of Technology, 1986.

<sup>106</sup> location on the ground beneath the hip of the robot.

<sup>107</sup> 猜测是指  $\mathbf{v}^{\text{CoM}}$  和  $\mathbf{p}^{\text{ref}}$ 。

<sup>108</sup> The state estimator is used to determine these values for the current time step, and the reference trajectory is used for future values. 意思是  $\mathbf{v}^{\text{CoM}}$  和  $\mathbf{p}^{\text{ref}}$  会跟 MPC 同频率迭代更新，当前值用状态估计器的结果计算，未来值用预测结果计算。

B. 对角踏步 *Trotting*

C. 扰动和楼梯 *Disturbances and Stairs*

D. 跳跃(类似于瞪羚那种) *Pronking/Jumping*

E. 飞奔 *Bounding and Galloping*

F. 高速运动时的限制 *Limitations at High Speed*

## 6 回顾和展望 CONCLUSIONS AND FUTURE WORK

我们发现，在MPC中精准的动力学模型不如瞬时的动力学模型重要，即是说在算力有限的情况下，算的快比算得准更重要。