Kinematics, Dynamics and Control of robotic systems in PyBullet

Deepak Raina
IIT Delhi

July 31, 2020

This document gives details regarding the dynamics and control techniques used to simulate the robotic systems in PyBullet [1].

1 Inverse dynamics

$$\boldsymbol{\tau} = \boldsymbol{I}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{G}(\boldsymbol{\theta}) \tag{1}$$

2 Computed torque control

$$\boldsymbol{\tau} = \boldsymbol{I}(\boldsymbol{\theta})\boldsymbol{u} + \boldsymbol{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \boldsymbol{G}(\boldsymbol{\theta}) \tag{2}$$

where,

$$\boldsymbol{u} = \ddot{\boldsymbol{\theta}}_d + \boldsymbol{K}_p(\boldsymbol{\theta}_d - \boldsymbol{\theta}) + \boldsymbol{K}_d(\dot{\boldsymbol{\theta}}_d - \dot{\boldsymbol{\theta}})$$

3 Impdedence Control

$$\boldsymbol{\tau}_\theta = \boldsymbol{G}(\boldsymbol{\theta}) + \boldsymbol{J}^T \boldsymbol{F}_{\boldsymbol{x}} \tag{3}$$

Since the goal is given in task space usually, thus dynamic equations will be transformed to task space coordinates. Thus,

$$\boldsymbol{F}_x = \boldsymbol{I}_d^{-1}\boldsymbol{I}_x(\boldsymbol{K}_p(\boldsymbol{x}_d - \boldsymbol{x}) + \boldsymbol{K}_d(\dot{\boldsymbol{x}}_d - \dot{\boldsymbol{x}})) + (\boldsymbol{I}_d^{-1}\boldsymbol{I}_x - \boldsymbol{1})\boldsymbol{F}^{\boldsymbol{e}} \tag{4}$$

where,

$$\boldsymbol{I}_x = \boldsymbol{J}^{-T}\boldsymbol{I}_\theta \boldsymbol{J}^{-1} \tag{5}$$

The matrices $\boldsymbol{I}_d$, $\boldsymbol{K}_p$ and $\boldsymbol{K}_d$ are the desired inertia, stiffness and damping matrices.

References

[1] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.