PR2 Controller – Gravity Compensation Callibration

Danny

July 6, 2016

1 Problem Definition

The PR2 controller greatly relies on a precise dynamics model. Especially the gravity compensation is crucial for compliant, yet accurate control performance. Unfortunately, the counterbalance system of the PR2 generates joint state dependent forces that are *not* part of the usual analytic dynamics model. Therefore, the goal is to use machine learning to obtain a better dynamics model. Generally, the dynamics of a (rigid) robot manipulator are given by

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = u. \tag{1}$$

The term $G(q)$ is the force generated by gravity. More compactly, this reads as

$$M\ddot{q} + F(q,\dot{q}) = u. \tag{2}$$

for $F(q,\dot{q}) = C(q,\dot{q})\dot{q} + G(q)$.

Given the state $q, \dot{q}$, the Featherstone algorithm (`world.equationOfMotion(M,F);`) provides a method to calculate M and F efficiently. Whereas M and the C in F are considered to be precise (they are based on CAD-data), the $G(q)$ is completely wrong, as it does not contain the springs of the PR2. Usually, it is even turned off. Therefore, instead of learning the whole dynamics model, we are only interested in the gravity compensation. As a consequence, we think that the static case $\ddot{q}, \dot{q} = 0$ is sufficient. If the robot is at a static equilibrium,

$$F(q,0) = u \tag{3}$$

must hold true. As this is certainly not the case in the presence of the springs, the goal is to learn the *residual* $\widehat{F}(q)$ such that

$$F(q,0) + \widehat{F}(q) = u. \tag{4}$$

This is to learn which motor signals $u = \widehat{F}(q)$ are necessary to hold the robot at every joint state q. The necessary motor commands u can be obtained quite easily with a P(I)D controller that holds the robot at a specific q.

2 Method

Collect data $D = \left\{q^{(i)}, u^{(i)}\right\}_{i=1}^{w}$ of random poses $q^{(i)} \in \mathbb{R}^n$ of the robot. The $u^{(i)} \in \mathbb{R}^n$ are the commanded torques to the motors, which were calculated by the realtime P(I)D controller. Before reading the $u^{(i)}$, it should be waited for a few seconds in order to ensure that the controller has been converged. Averaging the $u^{(i)}$ is also reasonable.

2.1 Pose Generation

2.2 Model

After having sampled the data, the residual $\widehat{F}(q)$ is considered to be linear in some features

$$\widehat{F}(q) = \phi(q)^T \beta. \tag{5}$$

The optimal β is calculated with standard ridge regression regularization.

3 Features

Of course, one could use Gaussian kernels (or RBFs in the feature view case). However, we think that in the 7 dimensional joint space the generalization capabilities for unseen configurations are considered to be limited. In addition, it should be possible, in principle, to express the counter balance system analytically. Therefore, finding features that could be parts of the analytic expressions of the gravity compensations seems to be more reasonable. Possible features are

- 1.
- q.
- q^2 (component wise).
- q^p (component wise) for some $p \in \mathbb{N}$.
- $\sin$ / $\cos(q)$ (component wise).
- $\sin$ / $\cos\left(\sum_{i \in T} q_i\right)$ for some index set T.
- $J(q)q$. Jacobian J of something interesting, for example the Jacobian of the center of mass of a link, multiplied with the joint state q. Maybe evaluated only at a subset T of joint states.
- $F(q, 0)$ analytic model of the gravity compensation. Then the method described here will learn the influence of the springs only.