THE UNIVERSITY OF CALIFORNIA, LOS ANGELES

ROBOTICS DESIGN CAPSTONE

EE 183DB

Off-center spinning mass controller for Quadcopters

Author:
Lin LI
Angel JIMENEZ
Wilson CHANG
Amirali OMIDFAR

Professor:
Ankur MEHTA

June 11, 2018

Abstract

We aim to design an off-center spinning mass underactuated controller to steer flying objects. A quadcopter with a rotating arm attached to it is used to demonstrate the principle of such controller. By in depth analysis of the system dynamics and results of this project, we wanted to develop a similar model applicable for rocket control systems.

Contents

1 Introduction

Modern Rocket uses 2 DOF revolute joint to turn the nozzle to directly control the direction of thrust. Challenges are it has to resist a very high temperature and the joint need a large amount of energy to keep the nozzle in a specific direction. Instead, a precisely controlled off-center mass in the front of the rocket can create a torque that steers the Rocket.

We aim to explore an alternative way to steer flying vehicles with under-actuated controller. Taking the motivation from modern rocket control, we are going to implement such controller in a quadcopter to demonstrate such principle. We hope to extend such controller to steer rockets in a more cost and energy efficient manner.

2 Mathematical Model

2.1 Symbols

Here is a list of all symbols used in this paper:

$\boldsymbol{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$	linear position vectors
$\boldsymbol{q} = \begin{bmatrix} q_r \\ q_i \\ q_j \\ q_k \end{bmatrix}$	angular orientation in quaternion
$\boldsymbol{F_T}$	thrust force
$\boldsymbol{F_G}$	gravitational force
$\boldsymbol{F_{AB}}$	reaction force acted from A on B
$\boldsymbol{\tau_{AB}}$	reaction torque acted from A on B
$\boldsymbol{\tau_M}$	torque generated by the motor
$\boldsymbol{\tau_{RF}}$	torque generated by the reaction force
m_A	mass of A
I_A	moment of inertial of A

2.2 Appendix

The Quaternion-derived Rotation matrix is defined as follow,

$$ {}^{O}_{B}R = R(\boldsymbol{q_B}) = \begin{bmatrix} q_r^2 + q_i^2 - q_j^2 - q_k^2 & 2q_iq_j - 2q_rq_k & 2q_iq_k + 2q_rq_j \\ 2q_iq_j + 2q_rq_k & q_r^2 - q_i^2 + q_j^2 - q_k^2 & 2q_jq_k - 2q_rq_i \\ 2q_iq_k - 2q_rq_j & 2q_jq_k + 2q_rq_i & q_r^2 - q_i^2 - q_j^2 + q_k^2 \end{bmatrix} $$

2.3 Quadcopter Body Dynamics

Forces and Torques:

$$
{}^{B}\boldsymbol{F_{T}} = \begin{bmatrix} 0 \\ 0 \\ F_{TB} \end{bmatrix}
$$

$$
{}^{O}\boldsymbol{F_{GB}} = \begin{bmatrix} 0 \\ 0 \\ -m_b g \end{bmatrix}
$$

$$
{}^{O}\boldsymbol{F_{CB}} = \begin{bmatrix} F_{CBx} \\ F_{CBy} \\ F_{CBz} \end{bmatrix}
$$

$$
{}^{B}\boldsymbol{\tau_{CB}} = \begin{bmatrix} \tau_{CBx} \\ \tau_{CBy} \\ -\tau_{M} \end{bmatrix}
$$

Net Force and Torque

$$
{}^{O}\boldsymbol{F_{net,B}} = {}^{O}\boldsymbol{F_{GB}} + {}^{O}\boldsymbol{F_{T}} + {}^{O}\boldsymbol{F_{CB}} = m_B\, {}^{O}\boldsymbol{a_{B}} \tag{1}
$$

$$
{}^{O}\boldsymbol{\tau_{net,B}} = R(\boldsymbol{q_{B}})\, {}^{B}\boldsymbol{\tau_{CB}} = {}^{O}I_B\, {}^{O}\boldsymbol{\alpha_{B}} \tag{2}
$$

2.4 Controller Dynamics

Forces and Torques:

$$
\begin{aligned}
{}^{O}\boldsymbol{F_{BC}} &= \begin{bmatrix} F_{BCx} \\ F_{BCy} \\ F_{BCz} \end{bmatrix} \\
{}^{O}\boldsymbol{F_{GC}} &= \begin{bmatrix} 0 \\ 0 \\ -m_c g \end{bmatrix} \\
{}^{C}\boldsymbol{\tau_{BC}} &= \begin{bmatrix} \tau_{BCx} \\ \tau_{BCy} \\ \tau_{M} \end{bmatrix} \\
{}^{O}\boldsymbol{r_{CB}} &= R(\boldsymbol{q_C}) \begin{bmatrix} -L_{Mx} \\ 0 \\ -L_{Mz} \end{bmatrix} \\
{}^{O}\boldsymbol{\tau_{RF}} &= {}^{O}\boldsymbol{r_{CB}} \times {}^{O}\boldsymbol{F_{BC}}
\end{aligned}
$$

Net Force and Net Torque:

$$
{}^{O}\boldsymbol{F_{net,C}} = {}^{O}\boldsymbol{F_{BC}} + {}^{O}\boldsymbol{F_{GC}} = m_C \, {}^{O}\boldsymbol{a_C} \tag{3}
$$

$$
{}^{O}\boldsymbol{\tau_{net,C}} = R(\boldsymbol{q_C}) \, {}^{C}\boldsymbol{\tau_{BC}} + {}^{O}\boldsymbol{\tau_{RF}} = {}^{O}I_c \, {}^{O}\boldsymbol{\alpha_C} \tag{4}
$$

2.5 Constraints and Manipulation

In the derivation below, assume everything is in the inertial frame unless explicitly stated.

The two bodies are contrainted (attached together), there are some relationship between the states and the forces between the body and the controller,

Let $\boldsymbol{p_{sys}} = \boldsymbol{p_B}$ and $\boldsymbol{q_{sys}} = \boldsymbol{q_B}$,

$$
\begin{bmatrix} \boldsymbol{p_C} \\ \boldsymbol{q_C} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p_B} + \boldsymbol{r_{BC}} \\ \boldsymbol{q_\theta q_B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{p_{sys}} + \boldsymbol{r_{BC}} \\ \boldsymbol{q_\theta q_{sys}} \end{bmatrix} \tag{5}
$$

$$
\begin{bmatrix} \dot{\boldsymbol{p}}_{\boldsymbol{C}} \\ \dot{\boldsymbol{q}}_{\boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{p}}_{\boldsymbol{sys}} + \dot{R}(\boldsymbol{q_{sys}}) \, {}^{B}\boldsymbol{r_{BC}} \\ \boldsymbol{q_\theta} \dot{\boldsymbol{q}}_{\boldsymbol{sys}} + \dot{\boldsymbol{q}}_{\boldsymbol{\theta}} \boldsymbol{q_{sys}} \end{bmatrix} \tag{6}
$$

$$
\begin{bmatrix} \ddot{\boldsymbol{p}}_{\boldsymbol{C}} \\ \ddot{\boldsymbol{q}}_{\boldsymbol{C}} \end{bmatrix} = \begin{bmatrix} \ddot{\boldsymbol{p}}_{\boldsymbol{sys}} + \ddot{R}(\boldsymbol{q_{sys}}) \, {}^{B}\boldsymbol{r_{BC}} \\ \boldsymbol{q_\theta} \ddot{\boldsymbol{q}}_{\boldsymbol{sys}} + 2[\dot{\boldsymbol{q}}_{\boldsymbol{\theta}} \dot{\boldsymbol{q}}_{\boldsymbol{sys}}] + \ddot{\boldsymbol{q}}_{\boldsymbol{\theta}} \boldsymbol{q_{sys}} \end{bmatrix} \tag{7}
$$

Newton's Third Law

$$ {}^{O}\boldsymbol{F_{BC}} = -{}^{O}\boldsymbol{F_{CB}} \tag{8} $$
$$ {}^{O}\boldsymbol{\tau_{BC}} = -{}^{O}\boldsymbol{\tau_{CB}} \tag{9} $$

To limit our degree of freedom in the system, we have set a constraint for our quaternions, namely unit quaternion:

$$ q_r^2 + q_i^2 + q_j^2 + q_k^2 = 1 \tag{10} $$
$$ q_r\dot{q}_r + q_i\dot{q}_i + q_j\dot{q}_j + q_k\dot{q}_k = 0 \tag{11} $$
$$ q_r\ddot{q}_r + q_i\ddot{q}_i + q_j\ddot{q}_j + q_k\ddot{q}_k + \dot{q_r}^2 + \dot{q_i}^2 + \dot{q_j}^2 + \dot{q_k}^2 = 0 \tag{12} $$

Last but not least, in the derivation below we use $\boldsymbol{q_\theta}$ directly for ease of typsetting, however, q_θ is not our state variable but θ, their relationship is defined below

$$ \boldsymbol{q_\theta} = \cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2})R(\boldsymbol{q_{sys}})\,{}^{B}\boldsymbol{\hat{z_B}} $$
$$ \boldsymbol{\dot{q}_\theta} = -\frac{1}{2}\sin(\frac{\theta}{2})\dot{\theta} + \frac{1}{2}\cos(\frac{\theta}{2})\dot{\theta}R(\boldsymbol{q_{sys}})\,{}^{B}\boldsymbol{\hat{z_B}} + \sin(\frac{\theta}{2})R(\dot{\boldsymbol{q_{sys}}})\,{}^{B}\boldsymbol{\hat{z_B}} $$

where ${}^{B}\boldsymbol{\hat{z_B}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

2.5.1 Combining the Force equations

From (1),

$$ {}^{O}\boldsymbol{F_{CB}} = m_B\,{}^{O}\boldsymbol{a_B} - {}^{O}\boldsymbol{F_{GB}} - {}^{O}\boldsymbol{F_T} $$

From (3),

$$ {}^{O}\boldsymbol{F_{BC}} = m_C\,{}^{O}\boldsymbol{a_C} - {}^{O}\boldsymbol{F_{GC}} $$

Using (6),

$$ m_B\,{}^{O}\boldsymbol{a_B} + m_C\,{}^{O}\boldsymbol{a_C} = {}^{O}\boldsymbol{F_{GC}} + {}^{O}\boldsymbol{F_{GB}} + {}^{O}\boldsymbol{F_T} $$

Simplifying the above expression, we get

$$ (m_b + m_c)\boldsymbol{\ddot{p}_{sys}} + m_c\ddot{R}(\boldsymbol{q_{sys}})\,{}^{B}\boldsymbol{r_{BC}} = \boldsymbol{F_{GC}} + \boldsymbol{F_{GB}} + \boldsymbol{F_T} \tag{13} $$

2.5.2 Combining the Torqe equations

From (2),

$$^{O}\boldsymbol{\tau_{CB}} = {}^{O}I_B\,{}^{O}\boldsymbol{\alpha_B}$$

From (4),

$$^{O}\boldsymbol{\tau_{BC}} = {}^{O}I_c\,{}^{O}\boldsymbol{\alpha_C} - {}^{O}\boldsymbol{\tau_{RF}}$$

Using (7),

$$^{O}I_B\,{}^{O}\boldsymbol{\alpha_B} + {}^{O}I_c\,{}^{O}\boldsymbol{\alpha_C} = {}^{O}\boldsymbol{\tau_{RF}}$$

Assuming all the vectors are represented in the inertial O frame, using the quaternion representation for angular acceleration,

$$I_B 2\left[\boldsymbol{\ddot{q}_B q_B^*} - (\boldsymbol{\dot{q}_B q_B^*})^2\right] + I_c 2\left[\boldsymbol{\ddot{q}_C q_C^*} - (\boldsymbol{\dot{q}_C q_C^*})^2\right] = {}^{O}\boldsymbol{\tau_{RF}} \tag{14}$$

Substituting (5)-(7) in the above expression and isolating second derivative on the left, we have

$$2I_B[\boldsymbol{\ddot{q}_{sys} q_{sys}^*}] + 2I_C[\boldsymbol{q_\theta \ddot{q}_{sys}}(\boldsymbol{q_\theta q_{sys}})^*] + 2I_C[\boldsymbol{\ddot{q}_\theta q_{sys}}](\boldsymbol{q_\theta q_{sys}})^* - \boldsymbol{r_{CB}} \times \boldsymbol{F_{BC}} = \zeta \tag{15}$$

where

$$\zeta = 2I_B(\boldsymbol{\dot{q}_{sys} q_{sys}^*})^2 + 2I_C[(\boldsymbol{q_\theta \dot{q}_{sys}} + \boldsymbol{\dot{q}_\theta q_{sys}})(\boldsymbol{q_\theta q_{sys}})^*]^2 - 4I_C(\boldsymbol{\dot{q}_\theta \dot{q}_{sys}})(\boldsymbol{q_\theta q_{sys}})^*$$

Note that we put $\boldsymbol{\tau_{RF}}$ on the left hand side, this is because we can express $\boldsymbol{F_{BC}}$ in terms of $\boldsymbol{\ddot{p}_{sys}}$ from (1), a second derivative of positional state

$$\boldsymbol{F_{BC}} = m_B \boldsymbol{\ddot{p}_{sys}} - \boldsymbol{F_{GB}} - \boldsymbol{F_T}$$

2.6 System of equations

From equation (12), (14), and (15), we have the function that relates our state variables together,

$$f(\boldsymbol{\ddot{p}}, \boldsymbol{\ddot{q}}, \ddot{\theta}, \boldsymbol{\dot{p}}, \boldsymbol{\dot{q}}, \dot{\theta}, \boldsymbol{p}, \boldsymbol{q}, \theta) = 0 \tag{16}$$

Assuming we can solve for $\ddot{\boldsymbol{p}}, \ddot{\boldsymbol{q}}, \ddot{\theta}$ given $\dot{\boldsymbol{p}}, \dot{\boldsymbol{q}}, \dot{\theta}, \boldsymbol{p}, \boldsymbol{q}, \theta$, let the state of our system to be

$$\boldsymbol{s_{sys}} = \begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \\ \dot{\theta} \\ \boldsymbol{p} \\ \boldsymbol{q} \\ \theta \end{bmatrix} \quad \text{so that} \quad \dot{\boldsymbol{s}}_{\boldsymbol{sys}} = \begin{bmatrix} \ddot{\boldsymbol{p}} \\ \ddot{\boldsymbol{q}} \\ \ddot{\theta} \\ \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{q}} \\ \dot{\theta} \end{bmatrix}$$

We have our state evolution equations as

$$s_{t+1} = s_t + \dot{s}_t \Delta t \tag{17}$$

2.7 Matlab Implementation

Implementing the systems of equations in (16), and solve for $\ddot{\boldsymbol{p}}, \ddot{\boldsymbol{q}}, \ddot{\theta}$ given $\dot{\boldsymbol{p}}, \dot{\boldsymbol{q}}, \dot{\theta}, \boldsymbol{p}, \boldsymbol{q}, \theta$ in Matlab doesn't yield a solution. There must be something wrong with the equations / the implementation.

3 Simulation

4 Off center spinning module design

In order to clarify the steps taken in designing the spinning mass module, we start with reviewing some main specifications of the quad-copter used for the purpose in the project.

4.1 Quad-copter Specifications

Crazyflie 2.0 is a 27 gram nano quad-copter used for this project. As for designing our spinning mass module, the main constraint was the maximum payload that Crazyflie could handle. According to Crazyflie 2.0 hardware specifications maximum recommended payload is 15 grams. Therefore the spinning mass module initially assumed to have maximum weight of 15 grams.

Figure 1: Crazyflie 2.0

4.2 Spinning mass considerations

Our design of spinning consists of selecting below components:

1. DC motor
2. Rotating arm
3. Actual mass
4. Rotary encoder

4.2.1 DC motor

As mentioned, mainly constrained by maximum payload for spinning mass module, we needed small, light motor with control feedback. Next important constraint was voltage and current criteria of the motor. Ideally we aimed to power the module from Crazyflie power supply which was 3.7V (250 mAh). Adding external battery was not ideal for our weight limit, so before implementation we tested our candidate motor along with the quad-copter 4 motors and measured voltage and current drawn by candidate motor. Consequently we selected the same type of DC motor as those utilized in quad-copter. The selected 7x16 mm Core less DC motor, weighs only 2.7 grams and was spinning controllable spinning rate drawing 0.4 A of current while powered by 3.7 battery. According to our power calculation, we estimated the complete system (quad-copter and spinning mass module) to last

Figure 2: 7mm Core less DC motor

for roughly 4 minutes, which we agreed would have been enough for demonstrations purposes at this point. After finalizing the ideal type of motor, we

4.2.2 Rotating arm

4.3 given summary

In order to controll the the direction - How we come up with the off-center mass design? + why use the motor we chose? Light, pwm controllable speed - What are some challenges in designing the off center mass? + maximum load + the hole that goes in the motor + mount it stably on the quad

5 Motors and Control circuit

WORK ON THIS ANGEL!!! WORK ON THIS AMir!!! - The more technical part of the controller + PWM + SMD soldering + Parallel battery source

6 Quad-copter Hacking

not successful

7 Results

- Simulation and Mathematical Model suggest it may work - Limitation in physical implementation may be the cause of unideal results

8 Further Work

9 Conclusion

- Our implementation has a lot of limitation: given the short amount of time we had - Proof of concept in Math Model / Simulation

10 Reference