Adaptive Quaternion Control of a Quadrotor

Bernhard Paus Graesdal

Massachusetts Institute of Technology

12th May 2020

Introduction

Scope of project

- Cascaded controller design for both attitude and position control
- Quaternion based controller to avoid singularities
- Attitude controller augmented with indirect MRAC design

Goal

Implement project on hardware.

- Physical parameters of the an actual quadrotor are used
- Actuator constraints of real DC motors taken into account

Sources

Design is very much based on [1] and [2].

Table of Contents

Attitude tracking

Error signal

$$\boldsymbol{q_e} = \bar{\boldsymbol{q}}_c \otimes \boldsymbol{q} \tag{1}$$

$$^{B}\boldsymbol{\omega}_{bc} = {}^{B}\boldsymbol{\omega}_b - {}^{B}\boldsymbol{\omega}_c \tag{2}$$

- $\boldsymbol{q}_c$ - Unit quaternion describing commanded attitude
- $\boldsymbol{q}_e$ - Rotation from current attitude to commanded attitude
- $\boldsymbol{\omega}_{bc}$ - Angular velocity of B frame wrt. C frame
- $\boldsymbol{\omega}_c$ - Commanded angular velocity

Attitude Tracking

The controller design will stabilize the *error dynamics*.

Rotational error dynamics

$$\dot{\boldsymbol{q}}_e = \frac{1}{2}\boldsymbol{q}_e \otimes \left(\begin{bmatrix} 0 \\ {}^{B}\boldsymbol{\omega}_{bc} \end{bmatrix}\right) \tag{3}$$

Angular velocity error dynamics

$${}^{B}\dot{\boldsymbol{\omega}}_{bc} = \boldsymbol{J}^{-1}[-{}^{B}\boldsymbol{\omega}_b \times \boldsymbol{J}\,{}^{B}\boldsymbol{\omega}_b + \boldsymbol{\tau}_{ext}] - {}^{B}\dot{\boldsymbol{\omega}}_c - {}^{B}\boldsymbol{\omega}_b \times {}^{B}\boldsymbol{\omega}_c \tag{4}$$

- $\boldsymbol{J}$ - Inertia matrix
- $\boldsymbol{\tau}_{ext}$ - Sum of external torques applied to the body

Attitude controller design

Desired closed-loop error dynamics:

$$\dot{\boldsymbol{q}}_e = \frac{1}{2}\boldsymbol{q}_e \otimes \left(\begin{bmatrix} 0 \\ \boldsymbol{\omega}_{bc} \end{bmatrix}\right) \tag{5}$$

$$\dot{\boldsymbol{\omega}}_{bc} = -k_q \log_v\left(\boldsymbol{q}_e^+\right) - k_\omega \boldsymbol{\omega}_{bc} \tag{6}$$

Stability and tracking

$$V = \frac{1}{2}k_q\|\boldsymbol{q}_e\|^2_{SO(3)} + \frac{1}{2}\boldsymbol{\omega}_{bc}^T\boldsymbol{\omega}_{bc} \implies \dot{V} = -k_\omega \boldsymbol{\omega}_{bc}^T\boldsymbol{\omega}_{bc} \leq 0 \tag{7}$$

Global Invariant set theorem: The fixed point $(\boldsymbol{q}_e, \boldsymbol{\omega}_{bc}) = (\pm\boldsymbol{q}_{id}, \mathbf{0})$ is asymptotically stable [3]. $\rightarrow$ tracking is achieved!

Attitude controller design

Two-fold controller design

$$\boldsymbol{\tau}_c = \boldsymbol{\tau}_b + \boldsymbol{\tau}_a \tag{8}$$

- Baseline controller: achieve desired closed-loop dynamics in the nominal case
- Adaptive controller: restore nominal behaviour in presence of uncertainties

Attitude controller design - Baseline controller

Baseline controller

Assume all parameters perfectly known: Use feedback linearization to achieve desired closed loop error dynamics:

$$\begin{aligned}\boldsymbol{\tau}_b = {}^B\boldsymbol{\omega}_b \times \boldsymbol{J}\,{}^B\boldsymbol{\omega}_b + \boldsymbol{J}({}^B\dot{\boldsymbol{\omega}}_c + {}^B\boldsymbol{\omega}_b \times {}^B\boldsymbol{\omega}_c \\ - k_q \log_v(\boldsymbol{q}_e^+) - k_w\,{}^B\boldsymbol{\omega}_{bc})\end{aligned} \tag{9}$$

Resulting error dynamics

$${}^B\dot{\boldsymbol{\omega}}_{bc} = \boldsymbol{J}^{-1}[-{}^B\boldsymbol{\omega}_b \times \boldsymbol{J}\,{}^B\boldsymbol{\omega}_b + \boldsymbol{\tau}_b] - {}^B\dot{\boldsymbol{\omega}}_c - {}^B\boldsymbol{\omega}_b \times {}^B\boldsymbol{\omega}_c \tag{10}$$

$$\implies \dot{\boldsymbol{\omega}}_{bc} = -k_q \log_v(\boldsymbol{q}_e^+) - k_\omega \boldsymbol{\omega}_{bc} \tag{11}$$

Works perfectly in the nominal case!

Attitude controller design - Adaptive controller

Purpose of adaptive controller

Restore nominal behaviour in face of uncertainties and disturbances.

Design approach

1. Find linear reference model which describes nominal case
2. Define adaptive parameters
3. Find adaptive control law
4. Choose adaptive laws to restore reference model

Attitude controller design - Adaptive controller

1. Find reference model to describe nominal case

Insert baseline controller into *real* dynamics (not error dynamics):

$$
\begin{aligned}
{}^{B}\dot{\boldsymbol{\omega}}_b &= [-k_w \boldsymbol{i} - [{}^{b}\boldsymbol{\omega}_c]_\times]{}^{B}\boldsymbol{\omega}_b + {}^{B}\dot{\boldsymbol{\omega}}_c - k_q \log_v(\boldsymbol{q}_e^+) + k_w \boldsymbol{\omega}_c \\
&:= \boldsymbol{A}_m {}^{B}\boldsymbol{\omega}_b + \boldsymbol{r}_m
\end{aligned} \tag{12}
$$

- $\boldsymbol{A}_m = -(k_w \boldsymbol{I} + [{}^{B}\boldsymbol{\omega}_c]_\times)$. Always stable: $\boldsymbol{A}_m^T + \boldsymbol{A}_m = -k_w \boldsymbol{I}$
- $\boldsymbol{r}_m = {}^{B}\dot{\boldsymbol{\omega}}_c - k_q \log_v(\boldsymbol{q}_e^+) + k_w \boldsymbol{\omega}_c$.

Attitude controller design - Adaptive controller

2. Define adaptive parameters

$$-\boldsymbol{J}^{-1}[\boldsymbol{\omega}_b \times \boldsymbol{J}\boldsymbol{\omega}_b] = \begin{bmatrix} \frac{J_z - J_y}{J_x}\omega_z\omega_y \\ \frac{J_x - J_z}{J_y}\omega_x\omega_z \\ \frac{J_y - J_x}{J_z}\omega_x\omega_y \end{bmatrix} := \boldsymbol{\Theta}\phi(\boldsymbol{\omega}_b) \tag{13}$$

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) =: \boldsymbol{J}^{-1} \succ 0 \tag{14}$$

New dynamics:

$$\boldsymbol{J}^B\dot{\boldsymbol{\omega}}_b = -{}^B\boldsymbol{\omega}_b \times \boldsymbol{J}^B\boldsymbol{\omega}_b + {}^B\boldsymbol{\tau}_{ext} \tag{15}$$

$$\implies \dot{\boldsymbol{\omega}}_b = \boldsymbol{\Theta}\phi(\boldsymbol{\omega}_b) + \boldsymbol{\Lambda}\boldsymbol{\tau}_c + \boldsymbol{\tau}_d \tag{16}$$

- $\boldsymbol{\Theta}$, $\boldsymbol{\Lambda}$ - Unknown parameters
- $\phi(\boldsymbol{\omega}_b)$ - Known regressor vector

Attitude controller design - Adaptive controller

3. Find adaptive control law

Letting the parameters deviate from their nominal values, and reformulating dynamics in terms of reference model, yields:

$$\begin{aligned}\dot{\boldsymbol{\omega}}_b = (\boldsymbol{\Theta} - \boldsymbol{\Theta}_D)\phi + \boldsymbol{A}_m\boldsymbol{\omega}_b + \boldsymbol{r}_m + \boldsymbol{\tau}_d \\ + \boldsymbol{\Lambda}(\boldsymbol{\tau}_a + (\boldsymbol{I} - \boldsymbol{\Lambda}^{-1}\boldsymbol{J}_D^{-1})\boldsymbol{\tau}_b)\end{aligned} \tag{17}$$

Choosing the adaptive controller:

$$\boldsymbol{\tau}_a = \hat{\boldsymbol{\Lambda}}^{-1}\left[-(\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_D)\phi - \hat{\boldsymbol{\tau}}_d\right] - (\boldsymbol{I} - \hat{\boldsymbol{\Lambda}}^{-1}\boldsymbol{J}_D^{-1})\boldsymbol{\tau}_b \tag{18}$$

yields the reference model when parameters equal nominal values (denoted by $_D$).

Attitude controller design - Adaptive controller

4. Choose adaptive laws to restore reference model

Error model: Letting $\boldsymbol{e} := \boldsymbol{\omega}_m - \boldsymbol{\omega}_b$ yields:

$$\dot{\boldsymbol{e}} = \boldsymbol{A}_m \boldsymbol{e} + \tilde{\boldsymbol{\Theta}} \phi + \tilde{\boldsymbol{\tau}}_d + \tilde{\boldsymbol{\Lambda}} \boldsymbol{\tau}_c \tag{19}$$

Adaptive laws:

$$\dot{\hat{\boldsymbol{\tau}}}_d = -\gamma_\tau \boldsymbol{P} \boldsymbol{e} \tag{20}$$

$$\dot{\hat{\boldsymbol{\Theta}}} = -\boldsymbol{\Gamma}_\Theta \boldsymbol{P} \boldsymbol{e} \phi^T \tag{21}$$

$$\dot{\hat{\boldsymbol{\Lambda}}} = -\boldsymbol{\Gamma}_\Lambda \boldsymbol{P} \boldsymbol{e} \boldsymbol{\tau}_c^T \tag{22}$$

Attitude controller design - Adaptive controller

Stability and tracking

$$V = \boldsymbol{e}^T \boldsymbol{P} \boldsymbol{e} + \gamma_\tau^{-1} \tilde{\boldsymbol{\tau}}_d^T \tilde{\boldsymbol{\tau}}_d + \mathrm{Tr}(\tilde{\boldsymbol{\Theta}}^T \boldsymbol{\Gamma}_\Theta^{-1} \tilde{\boldsymbol{\Theta}} + \tilde{\boldsymbol{\Lambda}}^T \boldsymbol{\Gamma}_\Lambda^{-1} \tilde{\boldsymbol{\Lambda}}) \tag{23}$$

$$\implies \dot{V} = -\boldsymbol{e}^T \boldsymbol{Q} \boldsymbol{e} = -k_\omega \boldsymbol{e}^T \boldsymbol{e} \leq 0 \tag{24}$$

Barbalats Lemma: Gives $\boldsymbol{e} \to \mathbf{0}$. Tracking is ensured!

Position controller

Position dynamics

$$\dot{\boldsymbol{x}}_{pos} := \frac{d}{dt}\begin{bmatrix}\boldsymbol{p}\\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{p}}\\ \boldsymbol{q} \otimes \frac{{}^{B}\boldsymbol{F}_{th}}{m} \otimes \bar{\boldsymbol{q}} + \boldsymbol{g}\end{bmatrix} := \begin{bmatrix}\dot{\boldsymbol{p}}\\ \boldsymbol{u}_{pd} + \boldsymbol{g}\end{bmatrix} \tag{25}$$

- $\boldsymbol{F}_{th}$ - Total thrust vector in body frame
- $\boldsymbol{g}$ - Gravity vector

Controller

- Assume attitude controller sufficiently fast: $\boldsymbol{q} \cong \boldsymbol{q}_c$
- Let $\boldsymbol{u}_{pd} := \boldsymbol{q} \otimes \frac{{}^{B}\boldsymbol{F}_{th}}{m} \otimes \bar{\boldsymbol{q}}$
- Use feedback linearization

Position controller

Controller

Choose $\boldsymbol{u}_{pd} = \boldsymbol{u}_{pos} - \boldsymbol{g}$. This yields the linear, time-invariant system:

$$\dot{\boldsymbol{x}}_{pos} = \begin{bmatrix} 0 & \boldsymbol{I} \\ 0 & 0 \end{bmatrix} \boldsymbol{x}_{pos} + \begin{bmatrix} 0 \\ \boldsymbol{I} \end{bmatrix} \boldsymbol{u}_{pos} \tag{26}$$

Linear control law:

$$\boldsymbol{u}_{pos} = -\boldsymbol{K}_{pos}(\boldsymbol{x}_{pos} - \boldsymbol{x}_{pos,d}) + \ddot{\boldsymbol{p}}_d \tag{27}$$

$$\implies \ddot{\boldsymbol{e}}_{pos} + \boldsymbol{k}_1 \boldsymbol{e}_{pos} + \boldsymbol{k}_2 \dot{\boldsymbol{e}}_{pos} = 0 \tag{28}$$

- $\boldsymbol{x}_{pos,d}$ is the desired position and velocity.
- $\ddot{\boldsymbol{p}}_d$ is the desired acceleration.
- Choose $\boldsymbol{K}_{pos}$ with LQR.

Experimantal Results

Implementation

- Implemented in C++ with the open-source library Eigen [4].
- Forward Euler with a step-size of 0.0001 seconds.

Physical parameters

$$m = 2.856 \quad \text{kg} \tag{29}$$

$$l_a = 0.2 \quad \text{m} \tag{30}$$

$$\boldsymbol{J} = \begin{bmatrix} 0.07 & 0 & 0 \\ 0 & 0.08 & 0 \\ 0 & 0 & 0.12 \end{bmatrix} \tag{31}$$

Experimantal Results

Controller parameters

$$k_q = 60.0, \quad k_\omega = 10.0 \quad k_e = 25.0 \tag{32}$$

$$\mathbf{\Gamma}_\Theta = 2\boldsymbol{I}, \quad \mathbf{\Gamma}_\Lambda = 2\boldsymbol{I}, \quad \gamma_\tau = 350 \tag{33}$$

$$\boldsymbol{K}_{pos} = \begin{bmatrix} 3.16 & 0 & 0 & 2.71 & 0 & 0 \\ 0 & 3.16 & 0 & 0 & 2.71 & 0 \\ 0 & 0 & 3.16 & 0 & 0 & 2.71 \end{bmatrix} \tag{34}$$

Initialization of adaptive parameters

- *Poor initial estimates:* Between 180% – 350% of the true value.
- The initial adaptive parameter values are calculated from this.

Flight test 1 - Aggressive attitude maneuvers

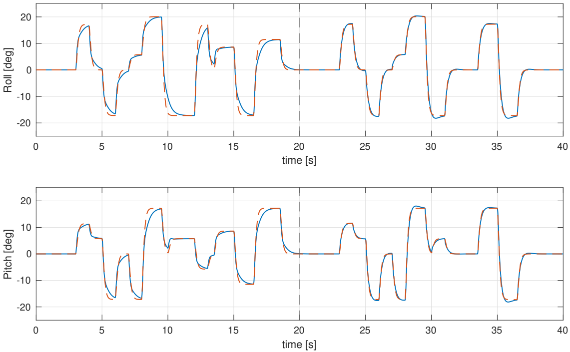

Figure 1: Attitude tracking of the quadrotor. The dashed line marks the point where the adaptive controller is switched on

Flight test 1 - Actuator inputs

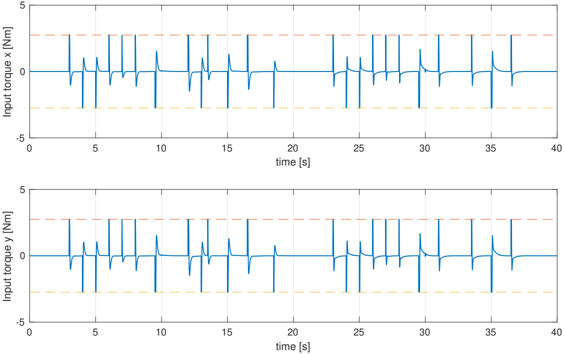

Figure 2: Applied input torques during attitude tracking. Dashed lines show the actual actuator limits, where the input is saturated.

Flight test 2 - Aggressive attitude maneuvers w/ unknown payload

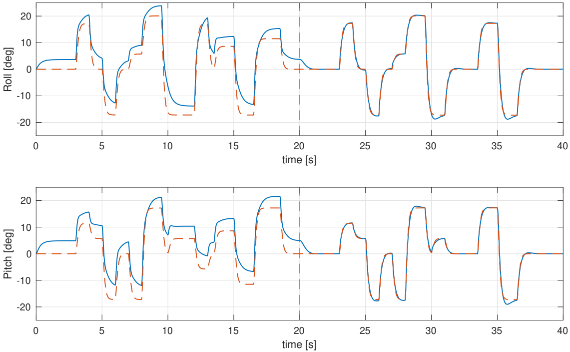

Figure 3: Attitude tracking with an unknown payload attached to two of the arms.

Flight test 3 - Position trajectory tracking

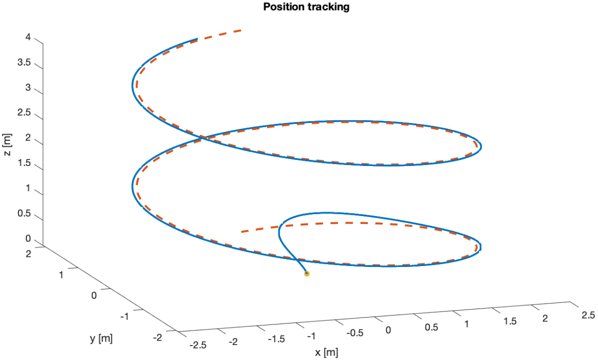

Figure 4: Simple position tracking using only the baseline controller

Flight test 4 - Position trajectory tracking w/ unknown payload

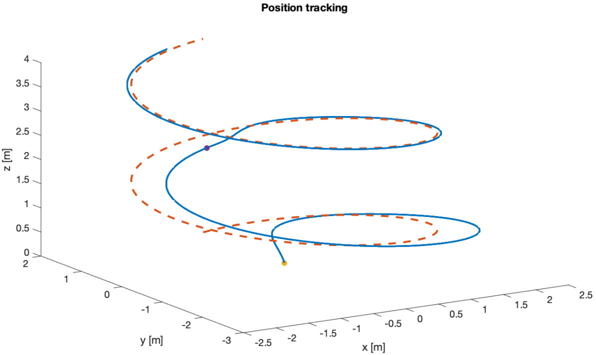

Figure 5: Position tracking with an unknown payload added to two of the quadrotor arms. The adaptive controller is switched on at the point indicated by a red dot.

Flight test 4 - Position trajectory tracking w/ unknown payload

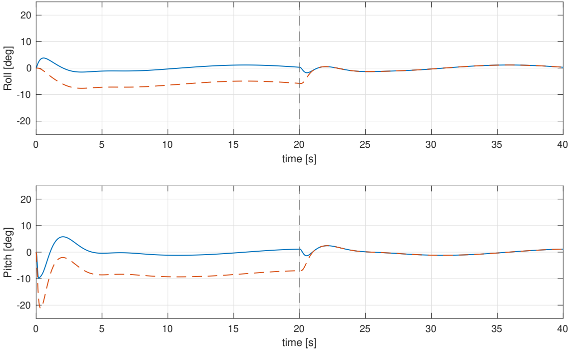

Figure 6: The attitude of the quadrotor when tracking a position reference with an unknown payload added to two of the arms. The adaptive augmentation is switched on at $t = 20$.

Future work

- Simulate motor dynamics
- Robustness modifications to adaptive controller:
  - Projection operator
  - e-modification
  - ++ [5]
- Input prioritization to deal with saturation [1].
- Adaptive mass parameter in position controller

Bibliography

- D. Mihailescu-Stoica, R. Acuña, and J. Adamy, "High performance adaptive attitude control of a quadrotor," 06 2019.
- J. Carino, H. Abaunza, and P. Castillo, "Quadrotor quaternion control," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 825–831, 2015.
- J.-J. E. Slotine and W. Li, "Applied nonlinear control," 1991.
- G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.
- E. Lavretsky and K. A. Wise, "Robust and adaptive control: With aerospace applications," 2012.