Optimal attitude control of satellites

Victorio Úbeda Sosa

Escuela Técnica Superior de Ingenieros Industriales
Universidad de Castilla - La Mancha

23 July 2019

Advisor: Pablo Pedregal Tercero

Contents

Contents

Contents

Contents

Contents

Contents

Section 1

Introduction

Aims and objectives

Aims and objectives

- To propose algorithms for the attitude control of satellites.

Aims and objectives

- To propose algorithms for the attitude control of satellites.
- Using optimal control techniques.

What is attitude control?

What is attitude control?

- **Attitude**

What is attitude control?

- **Attitude** = orientation.

What is attitude control?

- **Attitude** = orientation.
- **Attitude control**

What is attitude control?

- **Attitude** = orientation.
- **Attitude control** = Controlling the orientation (and angular velocities) of the spacecraft.

What is attitude control?

- **Attitude** = orientation.
- **Attitude control** = Controlling the orientation (and angular velocities) of the spacecraft.
    - Input

What is attitude control?

- **Attitude** = orientation.
- **Attitude control** = Controlling the orientation (and angular velocities) of the spacecraft.
  - Input: Torques exerted on the satellite.

What is attitude control?

- **Attitude** = orientation.
- **Attitude control** = Controlling the orientation (and angular velocities) of the spacecraft.
  - Input: Torques exerted on the satellite.
  - Output

What is attitude control?

- **Attitude** = orientation.
- **Attitude control** = Controlling the orientation (and angular velocities) of the spacecraft.
  - Input: Torques exerted on the satellite.
  - Output: Angular velocities and orientation (Euler angles, quaterion, etc...).

What is optimal control?

What is optimal control?

Optimal control problem

What is optimal control?

Optimal control problem

Minimize in $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^m$

$$\boldsymbol{u} \longmapsto \int_0^T F(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\ dt$$

What is optimal control?

Optimal control problem

Minimize in $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^m$

$$\boldsymbol{u} \longmapsto \int_0^T F(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\ dt$$

subject to

$$\begin{cases} \boldsymbol{x}'(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \quad \forall t \in [0, T] \\ \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ \boldsymbol{x}(T) = \boldsymbol{x}_T \\ \qquad \textit{Pointwise constraints} \end{cases}$$

Current state of the art

Current state of the art

- **Classical control theory:**

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].
- **More modern approaches:**

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].
- **More modern approaches:**
  - Wave-based control [Bellanca et al., 2017, Úbeda et al., 2018, Sherwin et al., 2018].

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].
- **More modern approaches:**
  - Wave-based control [Bellanca et al., 2017, Úbeda et al., 2018, Sherwin et al., 2018].
  - Linear-quadratic regulators [Lovera and Varga, 2005].

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].
- **More modern approaches:**
  - Wave-based control [Bellanca et al., 2017, Úbeda et al., 2018, Sherwin et al., 2018].
  - Linear-quadratic regulators [Lovera and Varga, 2005].
  - Fuzzy logic [Wiśniewski, 1996, Walker et al., 2013].

Current state of the art

- **Classical control theory:**
  - PID, PI, PD, etc: [Wiśniewski, 1996, Tudor, 2011, Sekhavat et al., 2011, Yang, 2012, Bråthen, 2013].
- **More modern approaches:**
  - Wave-based control [Bellanca et al., 2017, Úbeda et al., 2018, Sherwin et al., 2018].
  - Linear-quadratic regulators [Lovera and Varga, 2005].
  - Fuzzy logic [Wiśniewski, 1996, Walker et al., 2013].
  - Neural networks [Battipede et al., 2003].

Current state of the art

Current state of the art

- The vast majority rely on a previous linearisation of the equations of motion.

Current state of the art

- The vast majority rely on a previous linearisation of the equations of motion.
- The present work deals with the nonlinear system.

Section 2

Problem statement

Nonlinear equations of motion

Nonlinear equations of motion

Rigid body rotating in space.

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\omega}{dt} + \boldsymbol{\omega} \times J\omega = \mathbf{u}$$

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\omega}{dt} + \boldsymbol{\omega} \times J\omega = \mathbf{u}$$

where:

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\omega}{dt} + \boldsymbol{\omega} \times J\omega = \mathbf{u}$$

where:

- **Angular velocity** of the body-fixed frame with respect to the inertial frame: ω

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times J\boldsymbol{\omega} = \mathbf{u}$$

where:

- **Angular velocity** of the body-fixed frame with respect to the inertial frame: $\boldsymbol{\omega}$
- **Tensor of inertia**: J

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\omega}{dt} + \boldsymbol{\omega} \times J\omega = \mathbf{u}$$

where:

- **Angular velocity** of the body-fixed frame with respect to the inertial frame: $\boldsymbol{\omega}$
- **Tensor of inertia**: J
- **Torques** exerted on the satellite: $\boldsymbol{u}$

Nonlinear equations of motion

Rigid body rotating in space.

Dynamic equations of motion

$$\boldsymbol{J}\frac{d\omega}{dt} + \boldsymbol{\omega} \times J\omega = \mathbf{u}$$

where:

- **Angular velocity** of the body-fixed frame with respect to the inertial frame: $\boldsymbol{\omega}$
- **Tensor of inertia**: J
- **Torques** exerted on the satellite: $\boldsymbol{u}$ (This will be the control variable)

Nonlinear equations of motion

Nonlinear equations of motion

Kinematic equations of motion

$$\begin{cases} \dfrac{d\boldsymbol{\omega}}{dt} = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \dfrac{d\eta}{dt} = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases}$$

Nonlinear equations of motion

Kinematic equations of motion

$$\begin{cases} \dfrac{d\boldsymbol{\omega}}{dt} = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \dfrac{d\eta}{dt} = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases}$$

where:

Nonlinear equations of motion

Kinematic equations of motion

$$\begin{cases} \dfrac{d\boldsymbol{\omega}}{dt} = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \dfrac{d\eta}{dt} = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases}$$

where:

- **Unit quaternion** representing orientation of the body-fixed frame with respect to the inertial frame: $\boldsymbol{q} = (\epsilon, \eta)$

Nonlinear optimal control problem

Nonlinear optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$

Nonlinear optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \eta' = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases} \quad \text{in } [0, T]$$

Nonlinear optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \eta' = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases} \quad \text{in } [0, T]$$

and

$$\begin{cases} \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{q}(0) = \boldsymbol{q}_0 \\ \boldsymbol{\omega}(T) = \boldsymbol{\omega}_T, \quad \boldsymbol{q}(T) = \boldsymbol{q}_0 \end{cases}$$

Nonlinear optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \eta' = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases} \quad \text{in } [0, T]$$

and

$$\begin{cases} \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, \quad \boldsymbol{q}(0) = \boldsymbol{q}_0 \\ \boldsymbol{\omega}(T) = \boldsymbol{\omega}_T, \quad \boldsymbol{q}(T) = \boldsymbol{q}_0 \end{cases}$$

In addition to be a minimizer of a functional

$$\boldsymbol{u} \longmapsto \int_0^T F(t, \boldsymbol{\omega}, \boldsymbol{q}, \boldsymbol{u})$$

Linearised equation of motion

Linearised equation of motion

Reformulation of the nonlinear equations

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = \dfrac{1}{2} G(\boldsymbol{\epsilon})\boldsymbol{\omega} \end{cases}$$

Linearised equation of motion

Reformulation of the nonlinear equations

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = \dfrac{1}{2} G(\boldsymbol{\epsilon})\boldsymbol{\omega} \end{cases}$$

where

$$G(\boldsymbol{\epsilon}) = \begin{pmatrix} \sqrt{1-||\boldsymbol{\epsilon}||^2} & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \sqrt{1-||\boldsymbol{\epsilon}||^2} & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \sqrt{1-||\boldsymbol{\epsilon}||^2} \end{pmatrix}$$

Linearised equation of motion

Reformulation of the nonlinear equations

$$\begin{cases} J\boldsymbol{\omega}' = -\boldsymbol{\omega} \times J\boldsymbol{\omega} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = \dfrac{1}{2} G(\boldsymbol{\epsilon})\boldsymbol{\omega} \end{cases}$$

where

$$G(\boldsymbol{\epsilon}) = \begin{pmatrix} \sqrt{1-||\boldsymbol{\epsilon}||^2} & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \sqrt{1-||\boldsymbol{\epsilon}||^2} & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \sqrt{1-||\boldsymbol{\epsilon}||^2} \end{pmatrix}$$

[Yang, 2010]

Linearised equation of motion

First order taylor expansion around $\boldsymbol{\omega} = \boldsymbol{\epsilon} = \mathbf{0}$

Linearised equation of motion

First order taylor expansion around $\boldsymbol{\omega} = \boldsymbol{\epsilon} = \mathbf{0}$

Linearised equation of motion

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \boldsymbol{\epsilon}' \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ I_3 & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\epsilon} \end{pmatrix} + \begin{pmatrix} J^{-1} \\ 0_{3\times 3} \end{pmatrix} \boldsymbol{u}$$

Linearised equation of motion

First order taylor expansion around $\boldsymbol{\omega} = \boldsymbol{\epsilon} = \mathbf{0}$

Linearised equation of motion

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \boldsymbol{\epsilon}' \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ I_3 & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\epsilon} \end{pmatrix} + \begin{pmatrix} J^{-1} \\ 0_{3\times 3} \end{pmatrix} \boldsymbol{u}$$

Once $\boldsymbol{\epsilon}$ has been calculated, we use $|\boldsymbol{q}| = 1$ to compute $\boldsymbol{\eta}$.

Linearised optimal control problem

Linearised optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$

Linearised optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \boldsymbol{\epsilon}' \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ I_3 & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\epsilon} \end{pmatrix} + \begin{pmatrix} J^{-1} \\ 0_{3\times 3} \end{pmatrix} \boldsymbol{u} \quad \text{in } [0, T]$$

Linearised optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \boldsymbol{\epsilon}' \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ I_3 & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\epsilon} \end{pmatrix} + \begin{pmatrix} J^{-1} \\ 0_{3\times 3} \end{pmatrix} \boldsymbol{u} \quad \text{in } [0, T]$$

and

$$\begin{cases} \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, & \boldsymbol{\epsilon}(0) = \boldsymbol{\epsilon}_0 \\ \boldsymbol{\omega}(T) = \boldsymbol{\omega}_T, & \boldsymbol{\epsilon}(T) = \boldsymbol{\epsilon}_0 \end{cases}$$

Linearised optimal control problem

Find $\boldsymbol{u} : [0, T] \longrightarrow \mathbb{R}^3$ such that

$$\begin{pmatrix} \boldsymbol{\omega}' \\ \boldsymbol{\epsilon}' \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0_{3\times 3} & 0_{3\times 3} \\ I_3 & 0_{3\times 3} \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{\epsilon} \end{pmatrix} + \begin{pmatrix} J^{-1} \\ 0_{3\times 3} \end{pmatrix} \boldsymbol{u} \quad \text{in } [0, T]$$

and

$$\begin{cases} \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0, & \boldsymbol{\epsilon}(0) = \boldsymbol{\epsilon}_0 \\ \boldsymbol{\omega}(T) = \boldsymbol{\omega}_T, & \boldsymbol{\epsilon}(T) = \boldsymbol{\epsilon}_0 \end{cases}$$

In addition to be a minimizer of the functional

$$\boldsymbol{u} \longmapsto \frac{1}{2} \int_0^T \boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u}$$

$$\boldsymbol{x} = (\boldsymbol{\omega}, \boldsymbol{\epsilon})$$

Section 3

Numerical approximation

- Nonlinear problem

- Nonlinear problem $\rightarrow$ variational reformulation.

- Nonlinear problem $\rightarrow$ variational reformulation.
- Linearised problem

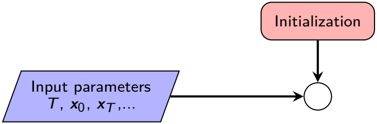
- Nonlinear problem $\rightarrow$ variational reformulation.
- Linearised problem $\rightarrow$ LQR techniques.

Constrained minimization

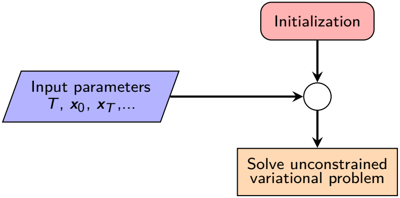
Constrained minimization

Initialization

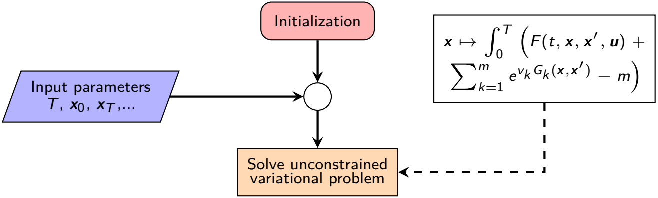
Constrained minimization



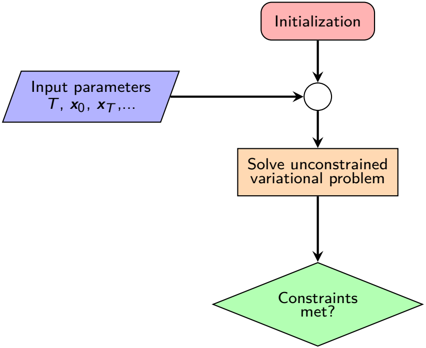
Constrained minimization



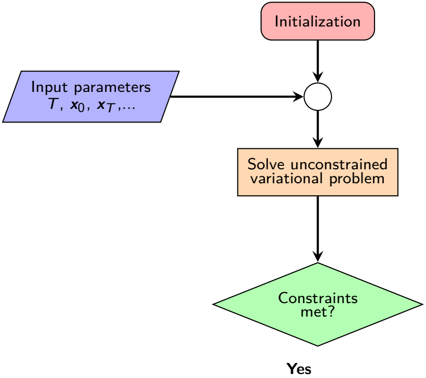
Constrained minimization



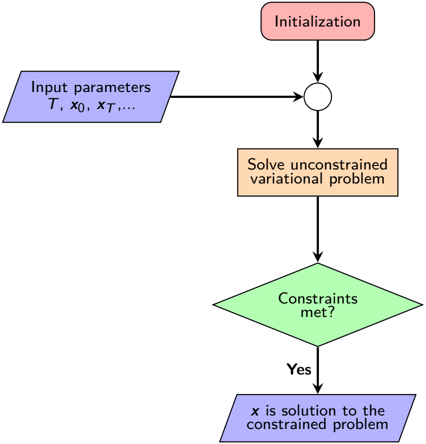
Constrained minimization



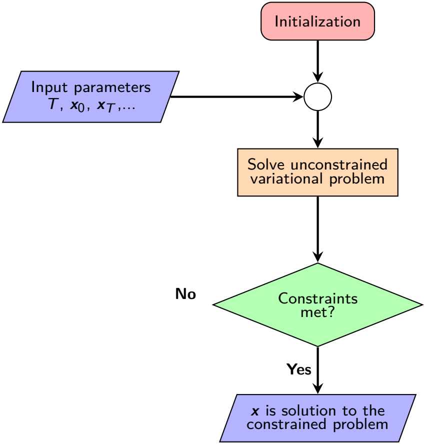
Constrained minimization



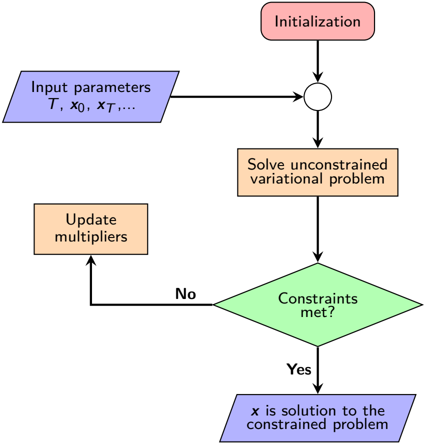
Constrained minimization



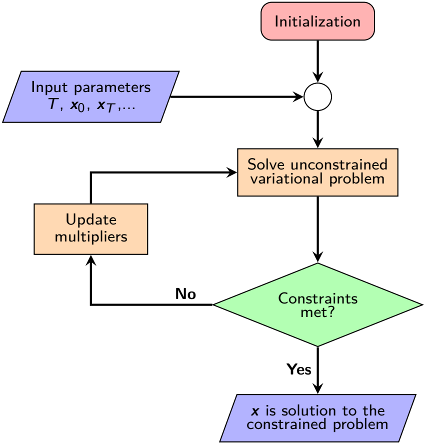
Constrained minimization



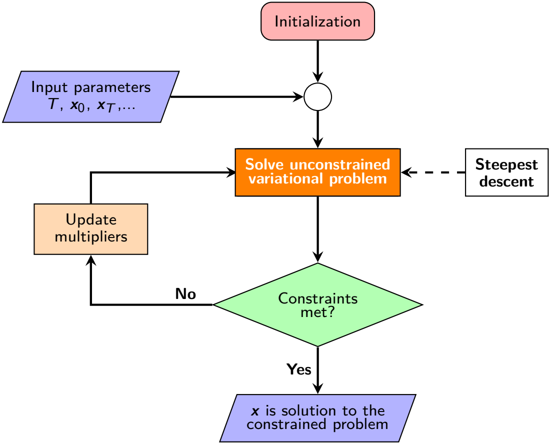
Constrained minimization



Constrained minimization



Constrained minimization

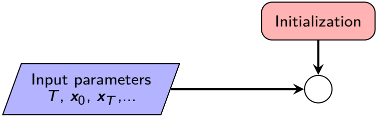


Algorithm for the solution of the LQR

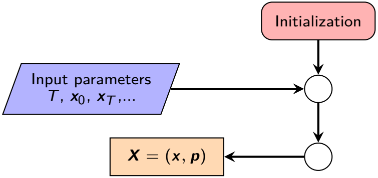
Algorithm for the solution of the LQR

Initialization

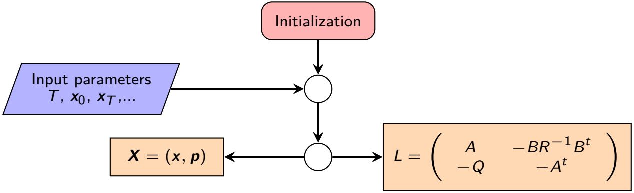
Algorithm for the solution of the LQR



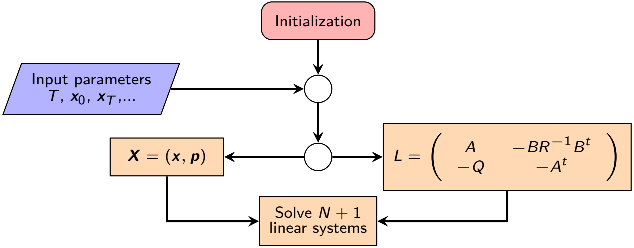
Algorithm for the solution of the LQR



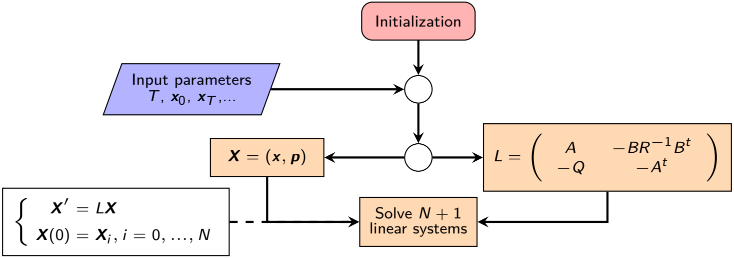
Algorithm for the solution of the LQR



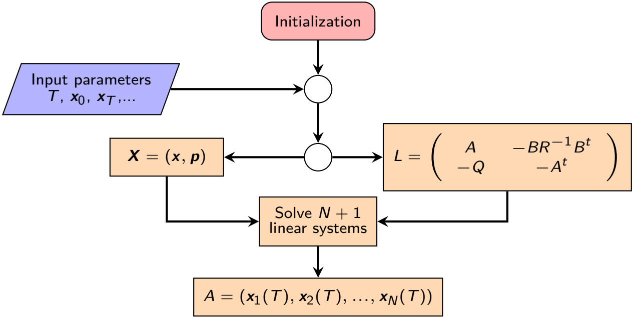
Algorithm for the solution of the LQR



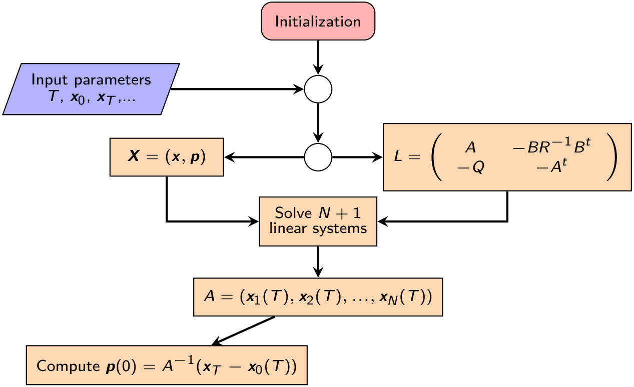
Algorithm for the solution of the LQR



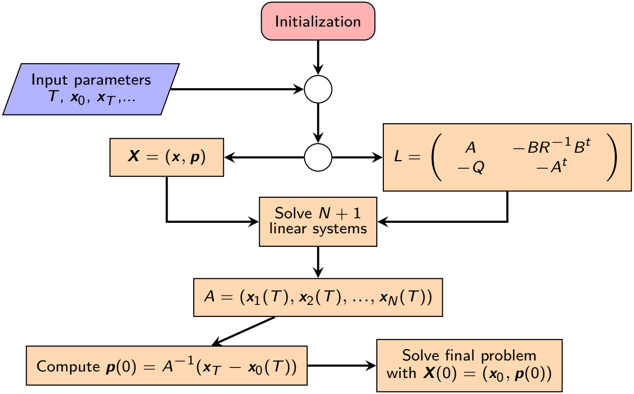
Algorithm for the solution of the LQR



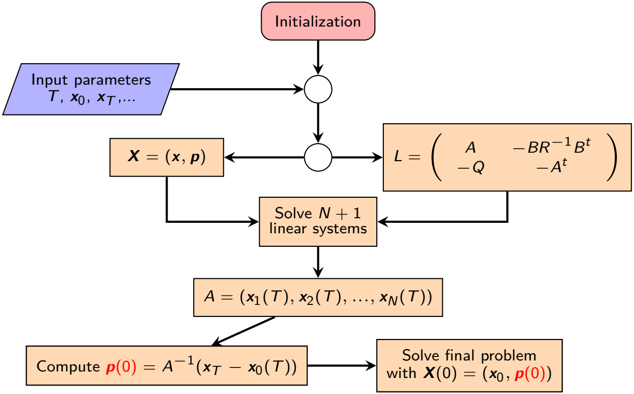
Algorithm for the solution of the LQR



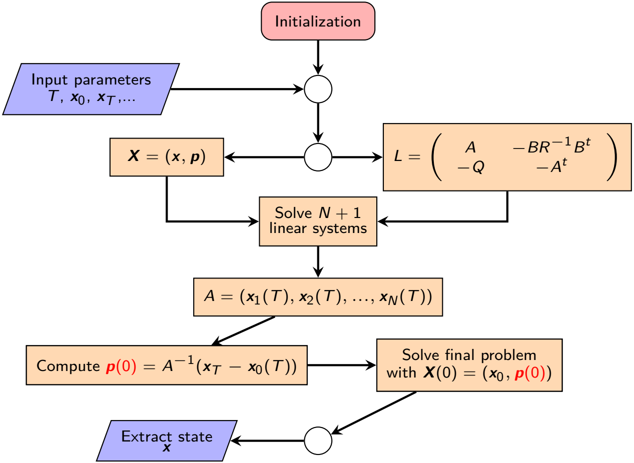
Algorithm for the solution of the LQR



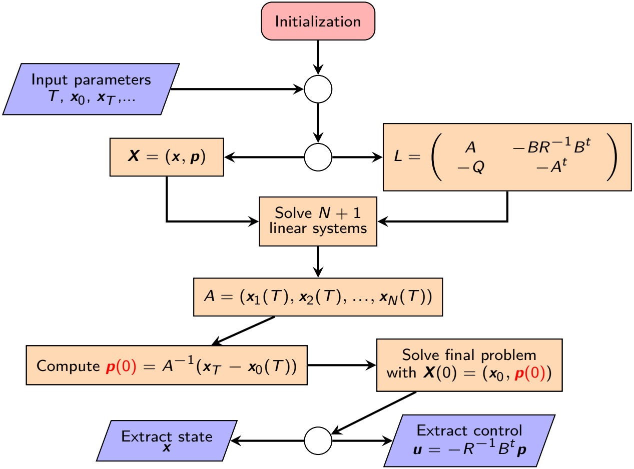
Algorithm for the solution of the LQR



Algorithm for the solution of the LQR



Algorithm for the solution of the LQR



Section 4

Results

All simulations were performed with a functional of the type

All simulations were performed with a functional of the type

Functional

$$\boldsymbol{u} \longmapsto \frac{1}{2}\int_0^T \boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u}$$

Linearised model

Linearised model – Detumbling

Linearised model – Detumbling

- Objective:
  - Dissipate rotational energy.
- Initial state:
  - Tumbling ($\boldsymbol{\omega} \neq \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.

Linearised model – Detumbling

- Objective:
  - Dissipate rotational energy.
- Initial state:
  - Tumbling ($\omega \neq \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)rad$.
- Final state:
  - At rest ($\omega = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)rad$.

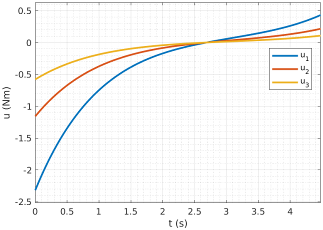


Figure: Control

Linearised model – Detumbling

- Objective:
  - Dissipate rotational energy.
- Initial state:
  - Tumbling ($\boldsymbol{\omega} \neq \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)rad$.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)rad$.

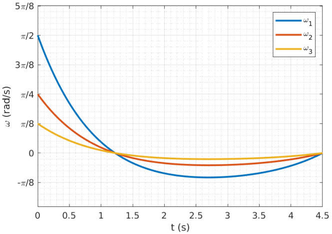


Figure: Angular velocities

Linearised model – Detumbling

- Objective:
  - Dissipate rotational energy.
- Initial state:
  - Tumbling ($\boldsymbol{\omega} \neq \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.

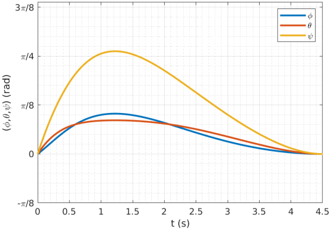


Figure: Euler angles

Simulation parameters

Name	Symbol	Value	Unit
Final time	T	4.5	s
Tensor of inertia	J	I_3	$kg\ m^2$
Integrand $F = \frac{1}{2}(\boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u})$	Q	I_6	
	R	I_3	
Initial angular velocity	$\boldsymbol{\omega}_0$	$(\pi/2, \pi/4, \pi/8)$	rad/s
Final angular velocity	$\boldsymbol{\omega}_T$	$(0, 0, 0)$	rad/s
Initial orientation (Euler angles)	Φ_0	$(0, 0, 0)$	rad
Final orientation (Euler angles)	Φ_T	$(0, 0, 0)$	rad

Table: Simulation parameters

Simulation results

Name	Symbol	Value	Unit
Achieved final angular velocity	$\boldsymbol{\omega}(T)$	$(4.22, 2.11, 2.14)10^{-3}$	rad/s
Error in final angular velocity	$\lvert\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T\rvert$	$8.39 \cdot 10^{-3}$	rad/s
Achieved final orientation	$\Phi(T)$	$(8.48, 1.63, 3.29)10^{-3}$	rad
Error in final orientation	$\lvert\Phi(T) - \Phi_T\rvert$	$3.77 \cdot 10^{-2}$	rad
Cost	$I(\boldsymbol{u})$	4.95	
CPU time		$1.68 \cdot 10^{-3}$	s

Table: Simulation results

Simulation results

Name	Symbol	Value	Unit
Achieved final angular velocity	$\boldsymbol{\omega}(T)$	$(4.22, 2.11, 2.14)10^{-3}$	rad/s
Error in final angular velocity	$\lvert\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T\rvert$	$8.39 \cdot 10^{-3}$	rad/s
Achieved final orientation	$\Phi(T)$	$(8.48, 1.63, 3.29)10^{-3}$	rad
Error in final orientation	$\lvert\Phi(T) - \Phi_T\rvert$	$3.77 \cdot 10^{-2}$	rad
Cost	$I(\boldsymbol{u})$	4.95	
CPU time		$1.68 \cdot 10^{-3}$	s

Table: Simulation results

Simulation parameters

Name	Symbol	Value	Unit
Final time	T	4.5	s
Tensor of inertia	J	I_3	$kg\ m^2$
Integrand $F = \frac{1}{2}(\boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u})$	Q	I_6	
	R	I_3	
Initial angular velocity	$\boldsymbol{\omega}_0$	$(\pi/2, \pi/4, \pi/8)$	rad/s
Final angular velocity	$\boldsymbol{\omega}_T$	$(0, 0, 0)$	rad/s
Initial orientation (Euler angles)	Φ_0	$(0, 0, 0)$	rad
Final orientation (Euler angles)	Φ_T	$(0, 0, 0)$	rad

Table: Simulation parameters.

Simulation parameters

Name	Symbol	Value	Unit
Final time	T	4.5	s
Tensor of inertia	J	I_3	$kg\ m^2$
Integrand $F = \frac{1}{2}(\boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u})$	Q R	I_6 I_3	
Initial angular velocity	$\boldsymbol{\omega}_0$	$(\pi/2, \pi/4, \pi/8)$	rad/s
Final angular velocity	$\boldsymbol{\omega}_T$	$(0, 0, 0)$	rad/s
Initial orientation (Euler angles)	Φ_0	$(0, 0, 0)$	rad
Final orientation (Euler angles)	Φ_T	$(0, 0, 0)$	rad

Table: Simulation parameters.

Tensor of inertia J

Tensor of inertia J proportional to the identity

Tensor of inertia J proportional to the identity $\Rightarrow$

Tensor of inertia J proportional to the identity $\Rightarrow \boldsymbol{\omega} \times J\boldsymbol{\omega} = \mathbf{0}$

Tensor of inertia J proportional to the identity $\Rightarrow \boldsymbol{\omega} \times J\boldsymbol{\omega} = \mathbf{0}$

$$\begin{cases} J\boldsymbol{\omega}' = -\cancel{\boldsymbol{\omega} \times J\boldsymbol{\omega}} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \eta' = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases}$$

Tensor of inertia J proportional to the identity $\Rightarrow \boldsymbol{\omega} \times J\boldsymbol{\omega} = \mathbf{0}$

$$\begin{cases} J\boldsymbol{\omega}' = -\cancel{\boldsymbol{\omega} \times J\boldsymbol{\omega}} + \boldsymbol{u} \\ \boldsymbol{\epsilon}' = -\dfrac{1}{2}\boldsymbol{\omega} \times \boldsymbol{\epsilon} + \dfrac{1}{2}\eta\boldsymbol{\omega} \\ \eta' = -\dfrac{1}{2}\boldsymbol{\omega}^t\boldsymbol{\epsilon} \end{cases}$$

Nonlinear effects are less noticeable.

If J is not proportional to the identity

If J is not proportional to the identity

Identical configuration, except:

$$J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} kg\ m^2$$

If J is not proportional to the identity

Identical configuration, except:

$$J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} kg\ \ m^2$$

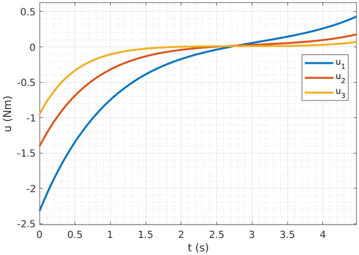


Figure: Control

If J is not proportional to the identity

Identical configuration, except:

$$J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} kg\ m^2$$

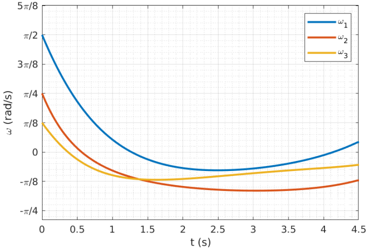


Figure: Angular velocities

If J is not proportional to the identity

Identical configuration, except:

$$J = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.5 \end{pmatrix} kg\ m^2$$

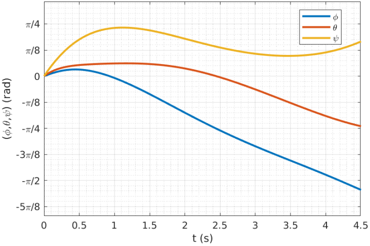


Figure: Euler angles

If J is not proportional to the identity – Simulation results

Name	Symbol	Value	Unit
Achieved final angular velocity	$\boldsymbol{\omega}(T)$	$(1.33, -3.81, -1.71)10^{-1}$	*rad/s*
Error in final angular velocity	$\lvert\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T\rvert$	$4.38 \cdot 10^{-1}$	*rad/s*
Achieved final orientation	$\Phi(T)$	$(-1.71, -0.75, 0.52)$	*rad*
Error in final orientation	$\lvert\Phi(T) - \Phi_T\rvert$	1.94	*rad*
Cost	$I(\boldsymbol{u})$	4.52	
CPU time		$2.56 \cdot 10^{-3}$	s

Table: Simulation results.

If J is not proportional to the identity – Simulation results

Name	Symbol	Value	Unit
Achieved final angular velocity	$\boldsymbol{\omega}(T)$	$(1.33, -3.81, -1.71)10^{-1}$	rad/s
Error in final angular velocity	$\lvert\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T\rvert$	$4.38 \cdot 10^{-1}$	rad/s
Achieved final orientation	$\Phi(T)$	$(-1.71, -0.75, 0.52)$	rad
Error in final orientation	$\lvert\Phi(T) - \Phi_T\rvert$	1.94	rad
Cost	$I(\boldsymbol{u})$	4.52	
CPU time		$2.56 \cdot 10^{-3}$	s

Table: Simulation results.

If J is not proportional to the identity – Simulation results

- Error in angular velocity:

$$|\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T| = 0.44\ rad/s \approx 4\ rpm$$

- Error in orientation (Euler angles):

$$|\phi(T) - \phi_T| = 1.94\ rad \approx 111^\circ$$

If J is not proportional to the identity – Simulation results

The linearised model is not able to steer the system to the desired state.

Nonlinear model

Nonlinear model – Detumbling

Nonlinear model – Detumbling

- Objective:
  - Dissipate rotational energy.
- Initial state:
  - Tumbling ($\boldsymbol{\omega} \neq \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0) rad$.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0) rad$.

Simulation parameters

Simulation parameters

Parameters are set as in the previous case:

Simulation parameters

Parameters are set as in the previous case:

Name	Symbol	Value	Unit
Final time	T	4.5	s
Integrand $F = \frac{1}{2}(\boldsymbol{x}^t Q \boldsymbol{x} + \boldsymbol{u}^t R \boldsymbol{u})$	Q	I_7	
	R	I_3	
Initial angular velocity	$\boldsymbol{\omega}_0$	$(\pi/2, \pi/4, \pi/8)$	rad/s
Final angular velocity	$\boldsymbol{\omega}_T$	$(0, 0, 0)$	rad/s
Initial orientation (Euler angles)	Φ_0	$(0, 0, 0)$	rad
Final orientation (Euler angles)	Φ_T	$(0, 0, 0)$	rad

Table: Simulation parameters

Simulation results

Simulation results

The nonlinear model behaves well for any choice of J.

Simulation results

$$J = I_{3\times 3}$$

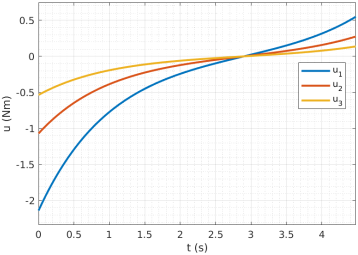


Figure: Control.

$$J = diag(1, 0.75, 0.5)$$

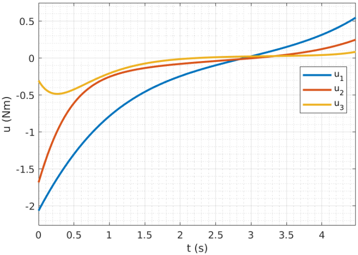


Figure: Control.

Simulation results

$J = I_{3\times3}$

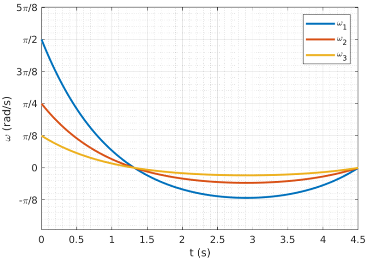


Figure: Angular velocities.

$J = diag(1, 0.75, 0.5)$

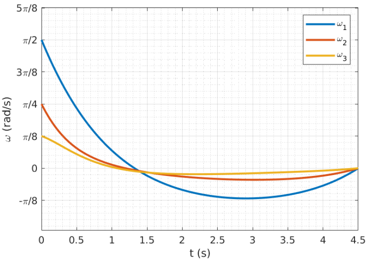


Figure: Angular velocities.

Simulation results

$$J = I_{3\times 3}$$

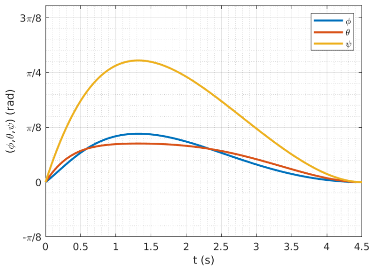


Figure: Euler angles.

$$J = diag(1, 0.75, 0.5)$$

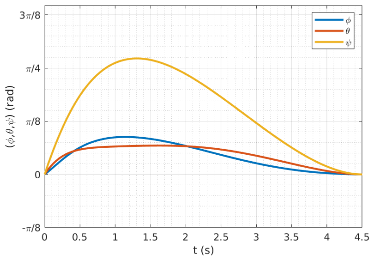


Figure: Euler angles

J proportional to the identity – Simulation results

- Error in angular velocity:
$$|\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T| = 7 \cdot 10^{-3} \ rad/s \approx 0.07 \ rpm$$
- Error in orientation (Euler angles):
$$|\phi(T) - \phi_T| = 1.59 \cdot 10^{-4} \ rad \approx 0.009^\circ$$

J not proportional to the identity – Simulation results

- Error in angular velocity:

$$|\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T| = 6.1 \cdot 10^{-3}\ rad/s \approx 0.06\ rpm$$

- Error in orientation (Euler angles):

$$|\phi(T) - \phi_T| = 1.55 \cdot 10^{-4}\ rad \approx 0.009^\circ$$

J not proportional to the identity – Simulation results

The nonlinear model is able to steer the system to the desired state.

Nonlinear model

Nonlinear model – Single axis, rest to rest

Nonlinear model – Single axis, rest to rest

- Objective:
  - Rotate $\pi/2$ *rad* around the z axis.
- Initial state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (\pi/2, 0, 0)$*rad*.
- Inequality constraints on control: $|u_i| \leq 1Nm\ i = 1, 2, 3$.

Nonlinear model – Single axis, rest to rest

- Objective:
  - Rotate $\pi/2$ *rad* around the *z* axis.
- Initial state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0,0,0)rad$.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (\pi/2, 0, 0)rad$.
- Inequality constraints on control: $|u_i| \leq 1Nm \; i = 1,2,3$.

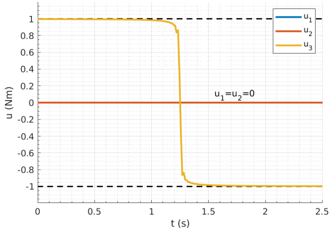


Figure: Control

Nonlinear model – Single axis, rest to rest

- Objective:
  - Rotate $\pi/2$ *rad* around the *z* axis.
- Initial state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $=$ $(\pi/2, 0, 0)$*rad*.
- Inequality constraints on control: $|u_i| \leq 1Nm \ i = 1, 2, 3$.

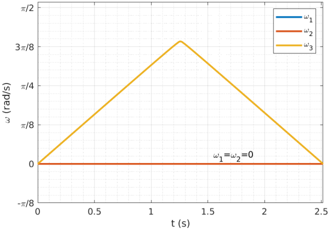


Figure: Angular velocities

Nonlinear model – Single axis, rest to rest

- Objective:
  - Rotate $\pi/2$ *rad* around the *z* axis.
- Initial state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $= (0, 0, 0)$*rad*.
- Final state:
  - At rest ($\boldsymbol{\omega} = \mathbf{0}$).
  - (pitch,roll,yaw) $=$ $(\pi/2, 0, 0)$*rad*.
- Inequality constraints on control: $|u_i| \leq 1Nm \; i = 1, 2, 3$.

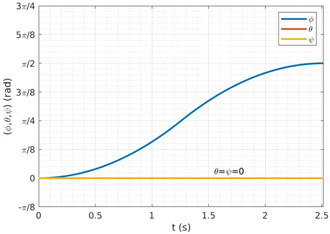


Figure: Euler angles

Simulation parameters

Simulation parameters

Name	Symbol	Value	Unit
Final time	T	$2.55, 2.525, 2.52$	s
Tensor of inertia	J	$diag(0.5, 0.75, 1)$	$Kg\ m^2$
Integrand $F = \frac{1}{2}(\boldsymbol{x}^t Q\boldsymbol{x} + \boldsymbol{u}^t R\boldsymbol{u})$	Q R	0_6 0_3	
Initial angular velocity	$\boldsymbol{\omega}_0$	$(0, 0, 0)$	rad/s
Final angular velocity	$\boldsymbol{\omega}_T$	$(0, 0, 0)$	rad/s
Initial orientation (Euler angles)	Φ_0	$(0, 0, 0)$	rad
Final orientation (Euler angles)	Φ_T	$(\pi/2, 0, 0)$	rad

Table: Simulation parameters

Bang-bang effect

Bang-bang effect

When inequality constraints on control are imposed

Bang-bang effect

When inequality constraints on control are imposed

For example: $|u_i| \leq 1\ Nm \quad i = 1, 2, 3$

Bang-bang effect

When inequality constraints on control are imposed

For example: $|u_i| \leq 1\ Nm \quad i = 1, 2, 3$

The control approaches a *bang-bang* strategy as the time interval is reduced.

Bang-bang effect

Bang-bang effect

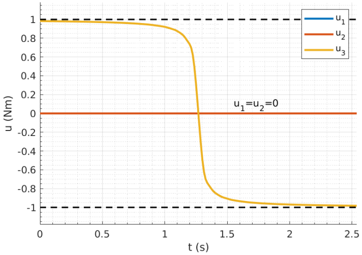


Figure: Control for $T = 2.55s$.

Bang-bang effect

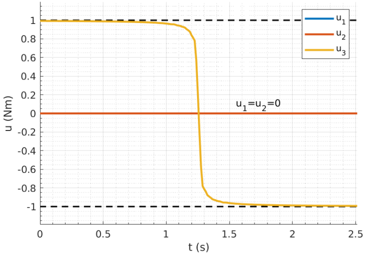


Figure: Control for $T = 2.525s$.

Bang-bang effect

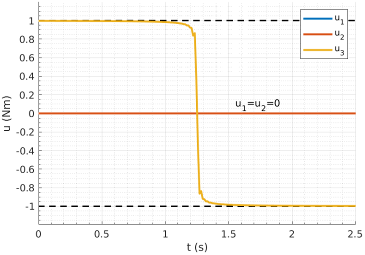
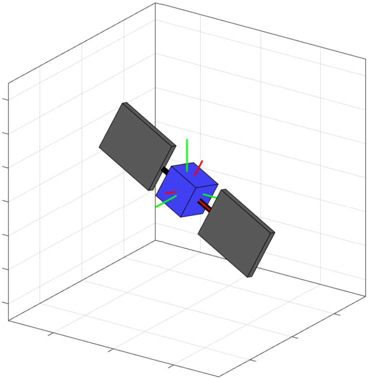


Figure: Control for $T = 2.52s$.

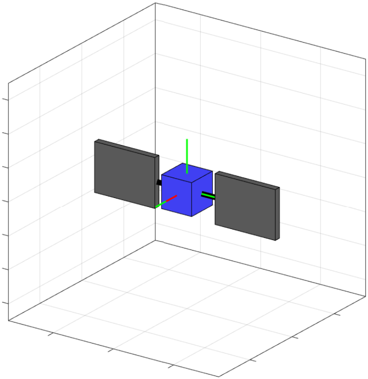
Simulation results

- Error in angular velocity:
$$|\boldsymbol{\omega}(T) - \boldsymbol{\omega}_T| = 2.84 \cdot 10^{-3} \ rad/s \approx 0.03 \ rpm$$
- Error in orientation (Euler angles):
$$|\phi(T) - \phi_T| = 3.8 \cdot 10^{-3} \ rad \approx 0.2^\circ$$

Animation – Detumbling



Animation – Single axis, rest to rest



Section 5

Conclusions and future work

Conclusions

Conclusions

- Two different approaches have been employed:

Conclusions

- Two different approaches have been employed:
  - Linearised model.

Conclusions

- Two different approaches have been employed:
    - Linearised model.
    - Nonlinear model (Main aim of the work).

Conclusions

- Two different approaches have been employed:
  - Linearised model.
  - Nonlinear model (Main aim of the work).
- Both have proven successful in certain situations.

Comparison

Advantages of the nonlinear approach:

Advantages of the linear approach:

Comparison

Advantages of the nonlinear approach:

- Accuracy.

Advantages of the linear approach:

Comparison

Advantages of the nonlinear approach:

- Accuracy.
- Constraints.

Advantages of the linear approach:

Comparison

Advantages of the nonlinear approach:

- Accuracy.
- Constraints.
- Larger class of integrands.

Advantages of the linear approach:

Comparison

Advantages of the nonlinear approach:

- Accuracy.
- Constraints.
- Larger class of integrands.
- Complex manoeuvres.

Advantages of the linear approach:

Comparison

Advantages of the nonlinear approach:

- Accuracy.
- Constraints.
- Larger class of integrands.
- Complex manoeuvres.

Advantages of the linear approach:

- Computation time.

Comparison

Advantages of the nonlinear approach:

- Accuracy.
- Constraints.
- Larger class of integrands.
- Complex manoeuvres.

Advantages of the linear approach:

- Computation time.
  - Linearised model: $CPU\ time \sim 10^{-3} s$.
  - Nonlinear model: $CPU\ time \sim 10 s$.

Simulations

Simulations

- Fortran 95.

Simulations

- Fortran 95.
- Compiled with gfortran.

Simulations

- Fortran 95.
- Compiled with gfortran.
- Linux machine.

Simulations

- Fortran 95.
- Compiled with gfortran.
- Linux machine.
- Intel Core i7 4510U.

Future work

Future work

- Controllability.

Future work

- Controllability.
- Influence of:

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.
- Underactuated satellites.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.
- Underactuated satellites.
- Uncertainty.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.
- Underactuated satellites.
- Uncertainty.
- Incomplete measurements.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.
- Underactuated satellites.
- Uncertainty.
- Incomplete measurements.
- Improvement of numerical procedures.

Future work

- Controllability.
- Influence of:
  - Magnetic actuation.
  - Disturbances, such as atmospheric drag.
- Underactuated satellites.
- Uncertainty.
- Incomplete measurements.
- Improvement of numerical procedures.
- Implementation and testing.

Thank you.