Control of Single and Multiple Thrust Propelled Systems

Pedro Miguel Ótão Pereira

Department of Automatic Control,
KTH Royal Institute of Technology

October 19th

Control of TPS

Control of Single and Multiple Thrust Propelled Systems

Pedro Miguel Ótão Pereira

Department of Automatic Control,
KTH Royal Institute of Technology

October 19th

- Presentation at AEROWORKS Autumn School
- Thursday, October 19th, 2017
- 16:15-17:00
- PhD student at KTH, under the supervision of Dimos Dimarogonas

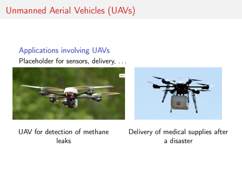
Unmanned Aerial Vehicles (UAVs)

Applications involving UAVs

Placeholder for sensors, delivery, . . .

UAV for detection of methane leaks

Delivery of medical supplies after a disaster



- Automate inspection, monitoring and delivery
- Relief support in natural (or non-natural) disasters

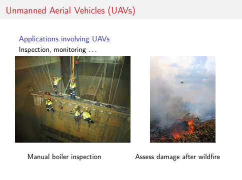
Unmanned Aerial Vehicles (UAVs)

Applications involving UAVs

Inspection, monitoring ...

Manual boiler inspection

Assess damage after wildfire



- Automate inspection, monitoring and delivery
- Relief support in natural (or non-natural) disasters

Unmanned Aerial Vehicles (UAVs)

Rotorcraft (vs fixed wing aircraft)

- Ability to hover
- Ability to take-off and land vertically

Helicopter vs Multirotors

- Helicopter: mechanically complex
- Multicopters/Multirotors: control of differential rotation (mechanically simpler)

Unmanned Aerial Vehicles (UAVs)
Rotorcraft (vs fixed wing aircraft)
Ability to hover
Ability to take-off and land vertically
Helicopter vs Multirotors
Helicopter: mechanically complex
Multicopters/Multirotors: control of differential rotation (mechanically simpler)

- Brief taxonomy
- Rotorcraft: rotary wing aircraft
- Multirotors: high maneuverability, low maintenance costs, low mechanical complexity and their ability to hover and to take-off and land vertically;
- Airplanes cannot hover or VTOL (the advantage is the large forward speed)
- Single-rotor (Helicopter): mechanically complex (swash plate is mechanically complex); Multirotors: control of differential rotation

AEROWORKS: European project

- Develop a team of aerial robotic workers

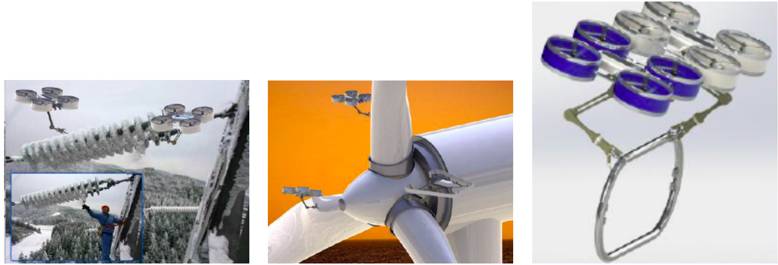
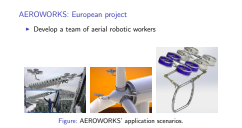


Figure: AEROWORKS' application scenarios.



- Five (six) academic partners: Lulea University of Technology – LTU, Swiss Federal Institute of Technology – ETHZ, Royal Institute of Technology – KTH, University of Twente – UT, University of Edinburgh – UEDIN, University of Patras – UPAT
- One (two) aerial robotics innovation SMEs: Ascending Technologies – ASC, Skybotix – SBX
- Two service robotics specialists: Alstom Inspection Robotics - AIR, and Skelleftea Kraft – SKL
- Removing humans from dangerous environments by performing aerial automated inspection.
- An automated fleet of aerial vehicles performing inspection and repair tasks on a wind mill.
- Cooperation between aerial vehicle when transporting a corona ring from an electrical pole.

Tethering vs Manipulator

Motivation

- Transportation of payloads in inaccessible locations
- Mechanically simplicity *vs* maneuverability
- Different useful payload capacity

Figure: Fukushima Nuclear Plant

Tethering vs Manipulator
Motivation
- Transportation of payloads in inaccessible locations
- Mechanically simplicity vs maneuverability
- Different useful payload capacity

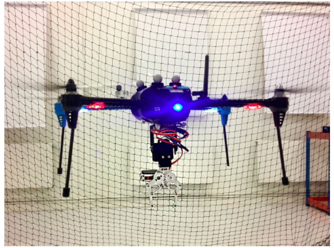
Figure: Fukushima Nuclear Plant

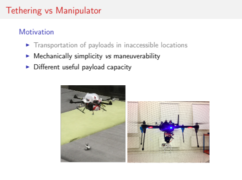
- Higher useful payload with cable (as opposed with robotic arm)
- Robotic arm provides extra degrees of freedom

Tethering vs Manipulator

Motivation

- Transportation of payloads in inaccessible locations
- Mechanically simplicity *vs* maneuverability
- Different useful payload capacity





- Higher useful payload with cable (as opposed with robotic arm)
- Robotic arm provides extra degrees of freedom

Overview

Overview

Motivation

Modeling of system of rigid bodies

Control of Thrust-propelled Systems

Conditions for local stability

Summary

Modeling: Point mass

- State

$$\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{velocity} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \end{bmatrix}$$

- Equations of motion

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} F_{net} \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \end{bmatrix}$$

- F_{net} is the net force applied on point mass
- m is the mass of the point mass

Modeling: Point mass
State

- Point mass kinematics and dynamics
- Net force = change of linear momentum
- I.F. = Inertial Frame

Modeling: Point mass

- State

$$\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{velocity} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \end{bmatrix}$$

- Equations of motion

$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} F_{net} \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \end{bmatrix}$$

- F_{net} is the net force applied on point mass
- m is the mass of the point mass

Modeling: Point mass
- State
$$\begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} \text{position} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{velocity} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \end{bmatrix}$$
- Equations of motion
$$\begin{bmatrix} \dot{p} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} F_{net} \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \end{bmatrix}$$
- F_{net} is the net force applied on point mass
- m is the mass of the point mass

- Point mass kinematics and dynamics
- Net force = change of linear momentum
- I.F. = Inertial Frame

Modeling: Rigid Body

- State

$$\begin{bmatrix} p \\ R \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} \text{linear position} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{angular position} \in \mathbb{SO}(3) \text{ w.r.t. I.F.} \\ \text{linear velocity} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{angular velocity} \in \mathbb{R}^3 \text{ w.r.t. B.F.} \end{bmatrix}$$

- Equations of motion

$$\begin{bmatrix} \dot{p} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} v \\ RS(\omega) \end{bmatrix} = \text{kinematics}$$

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{m} F_{net} \\ J^{-1}\left(-S(\omega)J\omega + \tau_{net}\right) \end{bmatrix} \Big|_{W_{net}=(F_{net},\tau_{net})} = \text{dynamics}$$

- W_{net} is the net wrench applied on rigid body
- m/J is the mass/moment-of-inertia of the rigid body

- Rigid body kinematics and dynamics
- Net wrench (net force, and net torque)
- Newton's-Euler's equations of motion
- Net force = change of linear momentum
- Net torque = change of angular momentum
- I.F. = Inertial Frame
- B.F. = Body Frame

Modeling: Rigid Body

- State

$$\begin{bmatrix} p \\ R \\ v \\ \omega \end{bmatrix} = \begin{bmatrix} \text{linear position} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{angular position} \in \mathbb{SO}(3) \text{ w.r.t. I.F.} \\ \text{linear velocity} \in \mathbb{R}^3 \text{ w.r.t. I.F.} \\ \text{angular velocity} \in \mathbb{R}^3 \text{ w.r.t. B.F.} \end{bmatrix}$$

- Equations of motion

$$\begin{bmatrix} \dot{p} \\ \dot{R} \end{bmatrix} = \begin{bmatrix} v \\ R\mathcal{S}(\omega) \end{bmatrix} = \text{kinematics}$$

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{m} F_{net} \\ J^{-1} \left(-\mathcal{S}(\omega) J \omega + \tau_{net} \right) \end{bmatrix} |_{W_{net=(F_{net}, \tau_{net})}} = \text{dynamics}$$

- W_{net} is the net wrench applied on rigid body
- m/J is the mass/moment-of-inertia of the rigid body

Modeling: Rigid Body
State
Equations of motion
W_{net} is the net wrench applied on rigid body
m/J is the mass/moment-of-inertia of the rigid body

- Rigid body kinematics and dynamics
- Net wrench (net force, and net torque)
- Newton's-Euler's equations of motion
- Net force = change of linear momentum
- Net torque = change of angular momentum
- I.F. = Inertial Frame
- B.F. = Body Frame

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass

$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$

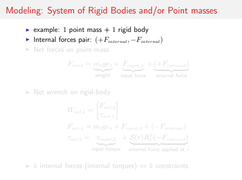
- Net wrench on rigid-body

$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$

$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$

$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{S(r)R_2^T(-F_{internal})}_{\text{internal force applied at } r}$$

- k internal forces (internal torques) $\Leftrightarrow$ k constraints



- Weight does not apply any torque on rigid-body (since point of application coincides with center-of-mass; this is only true if gravity field is assumed uniform – reasonable assumption for *small* rigid bodies)
- Actually, center-of-mass $\neq$ center-of-gravity, but center-of-mass $=$ center-of-gravity for our purposes
- r: point of application of internal force

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass

$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$

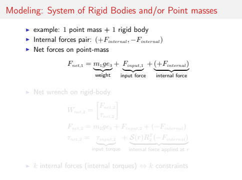
- Net wrench on rigid-body

$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$

$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$

$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{\mathcal{S}(r) R_2^T (-F_{internal})}_{\text{internal force applied at } r}$$

- k internal forces (internal torques) $\Leftrightarrow$ k constraints



- Weight does not apply any torque on rigid-body (since point of application coincides with center-of-mass; this is only true if gravity field is assumed uniform – reasonable assumption for *small* rigid bodies)
- Actually, center-of-mass $\neq$ center-of-gravity, but center-of-mass $=$ center-of-gravity for our purposes
- r: point of application of internal force

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass

$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$

- Net wrench on rigid-body

$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$

$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$

$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{\mathcal{S}(r) R_2^T (-F_{internal})}_{\text{internal force applied at } r}$$

- k internal forces (internal torques) $\Leftrightarrow$ k constraints

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass

$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$

- Net wrench on rigid-body

$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$

$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$

$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{\mathcal{S}(r) R_2^T (-F_{internal})}_{\text{internal force applied at } r}$$

- k internal forces (internal torques) $\Leftrightarrow$ k constraints

- Weight does not apply any torque on rigid-body (since point of application coincides with center-of-mass; this is only true if gravity field is assumed uniform – reasonable assumption for *small* rigid bodies)
- Actually, center-of-mass $\neq$ center-of-gravity, but center-of-mass $=$ center-of-gravity for our purposes
- r: point of application of internal force

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass

$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$

- Net wrench on rigid-body

$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$

$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$

$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{\mathcal{S}(r) R_2^T (-F_{internal})}_{\text{internal force applied at } r}$$

- k internal forces (internal torques) $\Leftrightarrow$ k constraints

Modeling: System of Rigid Bodies and/or Point masses
- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- Net forces on point-mass
$$F_{net,1} = \underbrace{m_1 g e_3}_{\text{weight}} + \underbrace{F_{input,1}}_{\text{input force}} + \underbrace{(+F_{internal})}_{\text{internal force}}$$
- Net wrench on rigid-body
$$W_{net,2} = \begin{bmatrix} F_{net,2} \\ \tau_{net,2} \end{bmatrix}$$
$$F_{net,2} = m_2 g e_3 + F_{input,2} + (-F_{internal})$$
$$\tau_{net,2} = \underbrace{\tau_{input,2}}_{\text{input torque}} + \underbrace{\mathcal{S}(r) R_2^T (-F_{internal})}_{\text{internal force applied at } r}$$
- k internal forces (internal torques) $\Leftrightarrow$ k constraints

- Weight does not apply any torque on rigid-body (since point of application coincides with center-of-mass; this is only true if gravity field is assumed uniform – reasonable assumption for *small* rigid bodies)
- Actually, center-of-mass $\neq$ center-of-gravity, but center-of-mass $=$ center-of-gravity for our purposes
- r: point of application of internal force

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- 3 (holonomic) constraints

$$\{(p_1, p_2, R_2) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{SO}(3) : C(p_1, p_2, R_2) = 0_3\} =: \mathbb{X}_{kinematic}$$

- Internal forces $F_{internal}$ are such that

$$\frac{d^2}{dt^2} C(p_1, p_2, R_2) = 0_3$$

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- 3 (holonomic) constraints

$\{(p_1, p_2, R_2) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{SO}(3) : C(p_1, p_2, R_2) = 0_3\} =: \mathbb{X}_{kinematic}$

- Internal forces $F_{internal}$ are such that

$\frac{d^2}{dt^2} C(p_1, p_2, R_2) = 0_3$

- Suppose in our example (1 point mass + 1 rigid body) we have 3 internal forces
- holonomic constraints: only depend on kinematic variables (i.e., don't depend on twists, or accelerations, ...)
- $\mathbb{X}_{kinematic}$: domain which all the kinematic variables belong to

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- 3 (holonomic) constraints

$$\left\{(p_1, p_2, R_2) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{SO}(3) : C(p_1, p_2, R_2) = 0_3\right\} =: \mathbb{X}_{kinematic}$$

- Internal forces $F_{internal}$ are such that

$$\frac{d^2}{dt^2} C(p_1, p_2, R_2) = 0_3$$

Modeling: System of Rigid Bodies and/or Point masses

- example: 1 point mass + 1 rigid body
- Internal forces pair: $(+F_{internal}, -F_{internal})$
- 3 (holonomic) constraints

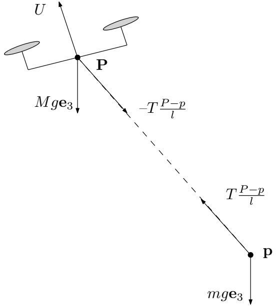
$$\left\{(p_1, p_2, R_2) \in \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{SO}(3) : C(p_1, p_2, R_2) = 0_3\right\} =: \mathbb{X}_{kinematic}$$

- Internal forces $F_{internal}$ are such that

$$\frac{d^2}{dt^2} C(p_1, p_2, R_2) = 0_3$$

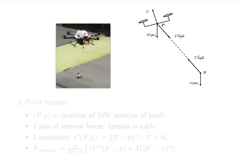
- Suppose in our example (1 point mass + 1 rigid body) we have 3 internal forces
- holonomic constraints: only depend on kinematic variables (i.e., don't depend on twists, or accelerations, ...)
- $\mathbb{X}_{kinematic}$: domain which all the kinematic variables belong to

Example: Slung-load



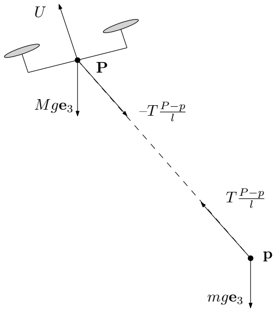
2 Point-masses

- $(P, p) =$ (position of UAV, position of load)
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0_1$
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}(U^T(P-p) + M\|V - v\|^2)$



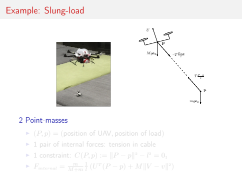
- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

Example: Slung-load



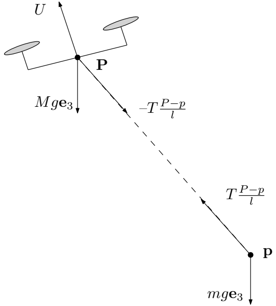
2 Point-masses

- $(P, p) =$ (position of UAV, position of load)
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0$,
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}(U^T(P-p) + M\|V - v\|^2)$



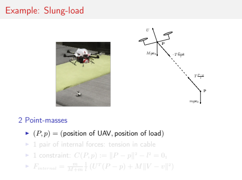
- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

Example: Slung-load



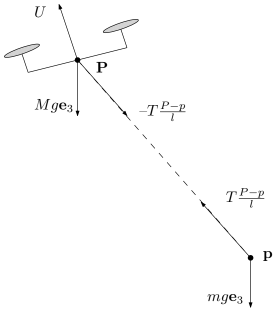
2 Point-masses

- $(P, p) = $ (position of UAV, position of load)
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0_1$
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}\left(U^T(P-p) + M\|V - v\|^2\right)$



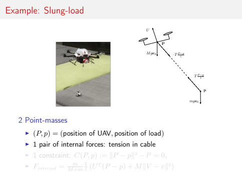
- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

Example: Slung-load



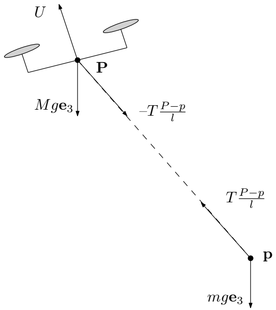
2 Point-masses

- $(P, p) =$ (position of UAV, position of load)
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0_1$
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}(U^T(P-p) + M\|V - v\|^2)$



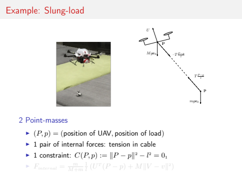
- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

Example: Slung-load



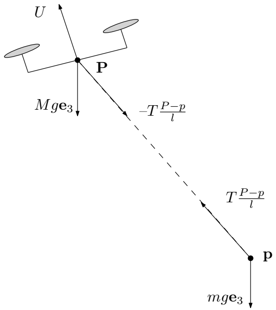
2 Point-masses

- $(P, p) = (\text{position of UAV}, \text{position of load})$
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0_1$
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}\left(U^T(P-p) + M\|V - v\|^2\right)$



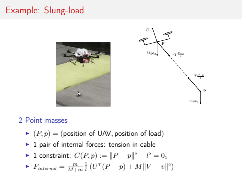
- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

Example: Slung-load



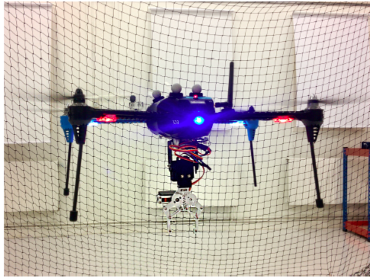
2 Point-masses

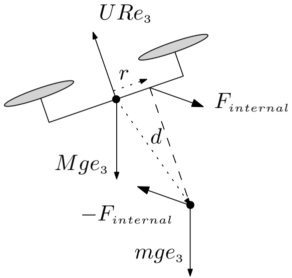
- $(P, p) =$ (position of UAV, position of load)
- 1 pair of internal forces: tension in cable
- 1 constraint: $C(P, p) := \|P - p\|^2 - l^2 = 0_1$
- $F_{internal} = \frac{m}{M+m}\frac{1}{l}\left(U^T(P-p) + M\|V - v\|^2\right)$



- Massless cable
- Cable imposes 1 holonomic constraint
- $U \in \mathbb{R}^3$ is taken as an input force on UAV

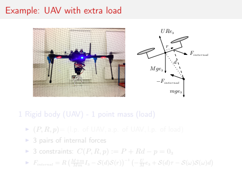
Example: UAV with extra load





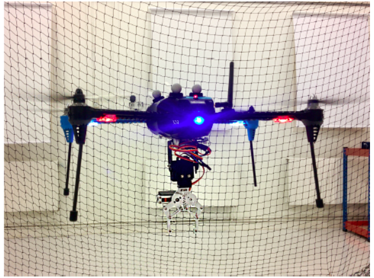
1 Rigid body (UAV) - 1 point mass (load)

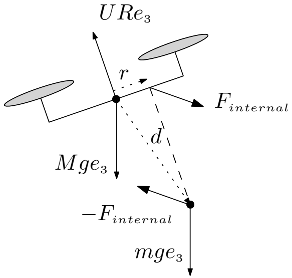
- (P, R, p)= (l.p. of UAV, a.p. of UAV, l.p. of load)
- 3 pairs of internal forces
- 3 constraints: $C(P, R, p) := P + Rd - p = 0_3$
- $F_{internal} = R\left(\frac{M+m}{Mm} I_3 - \mathcal{S}(d)\mathcal{S}(r)\right)^{-1}\left(-\frac{U}{M}e_3 + \mathcal{S}(d)\tau - \mathcal{S}(\omega)\mathcal{S}(\omega)d\right)$



- l.p. = linear position; a.p. = angular position
- For brevity, I took $J = I_3$: in mathematica the internal forces for generic moment of inertia are found.

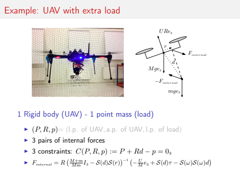
Example: UAV with extra load





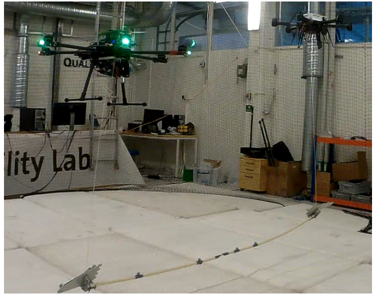
1 Rigid body (UAV) - 1 point mass (load)

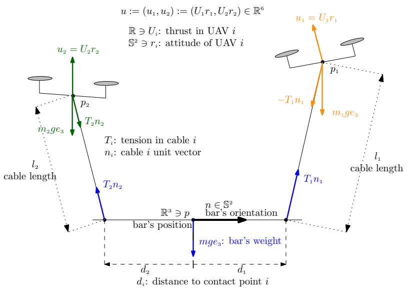
- (P, R, p)= (l.p. of UAV, a.p. of UAV, l.p. of load)
- 3 pairs of internal forces
- 3 constraints: $C(P, R, p) := P + Rd - p = 0_3$
- $F_{internal} = R\left(\frac{M+m}{Mm} I_3 - \mathcal{S}(d)\mathcal{S}(r)\right)^{-1}\left(-\frac{U}{M}e_3 + \mathcal{S}(d)\tau - \mathcal{S}(\omega)\mathcal{S}(\omega)d\right)$



- l.p. = linear position; a.p. = angular position
- For brevity, I took $J = I_3$: in mathematica the internal forces for generic moment of inertia are found.

Example: 2 UAVs + 1 bar

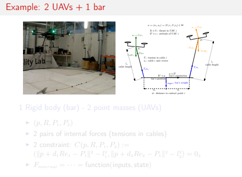




1 Rigid body (bar) - 2 point masses (UAVs)

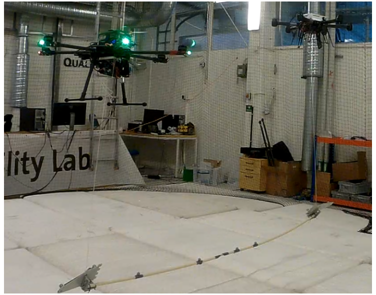
- (p, R, P_1, P_2)
- 2 pairs of internal forces (tensions in cables)
- 2 constraint: $C(p, R, P_1, P_2) := (\|p + d_1 R e_1 - P_1\|^2 - l_1^2, \|p + d_2 R e_2 - P_1\|^2 - l_2^2) = 0_2$
- $F_{internal} = \cdots =$ function(inputs, state)

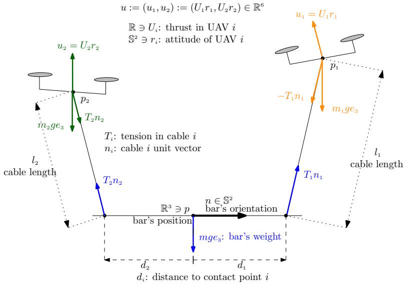
Pedro



- Pereira, P. and Dimarogonas, D.V. *Collaborative Transportation of a Bar by Two Aerial Vehicles with Attitude Inner Loop and Experimental Validation* CDC, 2017 (to appear).

Example: 2 UAVs + 1 bar

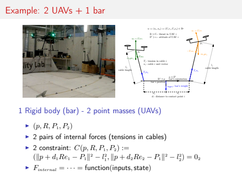




1 Rigid body (bar) - 2 point masses (UAVs)

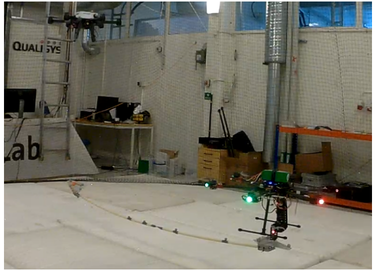
- (p, R, P_1, P_2)
- 2 pairs of internal forces (tensions in cables)
- 2 constraint: $C(p, R, P_1, P_2) :=$
  $(\|p + d_1 R e_1 - P_1\|^2 - l_1^2, \|p + d_2 R e_2 - P_1\|^2 - l_2^2) = 0_2$
- $F_{internal} = \cdots = \text{function}(\text{inputs}, \text{state})$

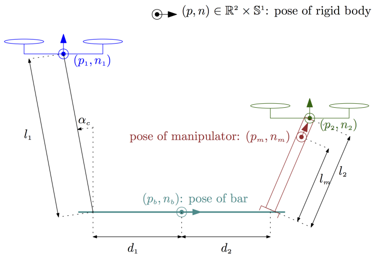
Pedro



- Pereira, P. and Dimarogonas, D.V. *Collaborative Transportation of a Bar by Two Aerial Vehicles with Attitude Inner Loop and Experimental Validation* CDC, 2017 (to appear).

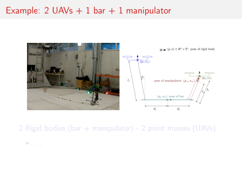
Example: 2 UAVs + 1 bar + 1 manipulator





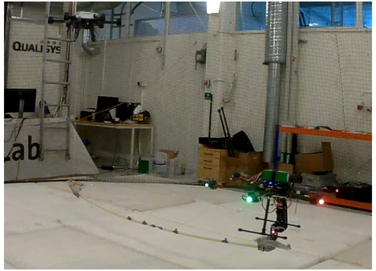
2 Rigid bodies (bar + manipulator) - 2 point masses (UAVs)

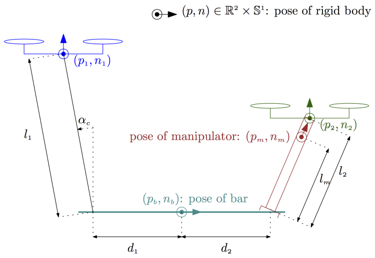
- ...



- Roque, P., Pereira, P. and Dimarogonas, D.V. *Heterogeneous Collaborative Pose Stabilization of a Bar Via Aerial Manipulation and Tethering* ICRA, 2018 (submitted).

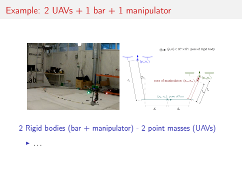
Example: 2 UAVs + 1 bar + 1 manipulator





2 Rigid bodies (bar + manipulator) - 2 point masses (UAVs)

- ...



- Roque, P., Pereira, P. and Dimarogonas, D.V. *Heterogeneous Collaborative Pose Stabilization of a Bar Via Aerial Manipulation and Tethering* ICRA, 2018 (submitted).

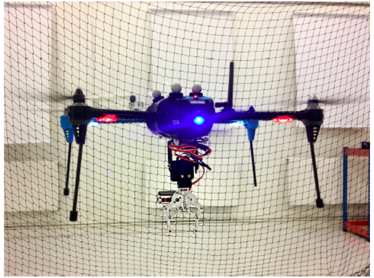
Final remarks

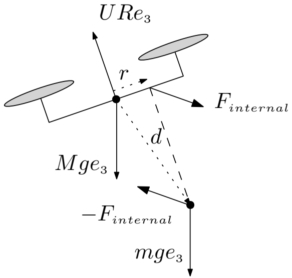
- Model necessary for design/analysis
- Parameterizations: smaller set of coordinates

Final remarks
Final remarks
- Model necessary for design/analysis
- Parameterizations: smaller set of coordinates

- Other parameterizations: Euler-angles – see Professor Tzes' talk
- Other parameterizations: unit-quaternion – see Emil Fresk's talk
- I chose the $\mathbb{SO}(3)$ group to be parametrized by 3 by 3 matrices (orthonormal matrices)

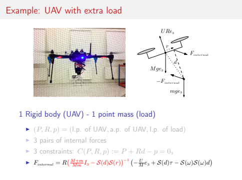
Example: UAV with extra load





1 Rigid body (UAV) - 1 point mass (load)

- $(P, R, p) =$ (l.p. of UAV, a.p. of UAV, l.p. of load)
- 3 pairs of internal forces
- 3 constraints: $C(P, R, p) := P + Rd - p = 0_3$
- $F_{internal} = R\left(\frac{M+m}{Mm} I_3 - \mathcal{S}(d)\mathcal{S}(r)\right)^{-1}\left(-\frac{U}{M} e_3 + \mathcal{S}(d)\tau - \mathcal{S}(\omega)\mathcal{S}(\omega)d\right)$



- Complexity of internal forces
- If we have k internal forces we have to invert a $k \times k$ matrix
- In this example we have to invert a 3×3 matrix
- See Professor Tzes' talk on modeling complexity

Overview

Overview

Motivation

Modeling of system of rigid bodies

Control of Thrust-propelled Systems

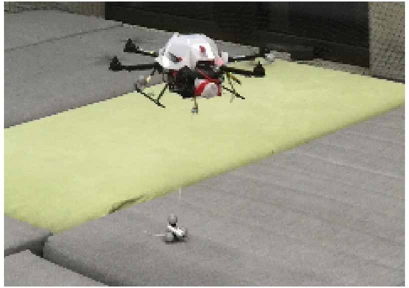
Conditions for local stability

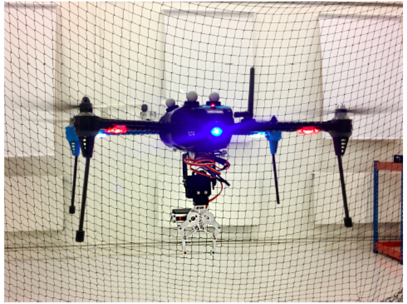
Summary

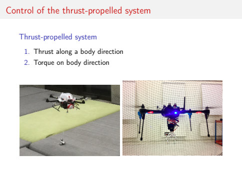
Control of the thrust-propelled system

Thrust-propelled system

1. Thrust along a body direction
2. Torque on body direction

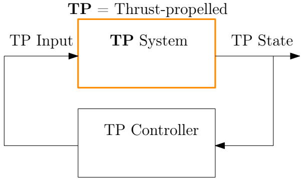




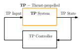


- We wish to design a controller that can be used for different systems
- These systems have something in common, which is what we explore in the control design
- Thrust-propelled systems, for example, trajectory tracking by quadrotor, trajectory tracking by load tethered to UAV, trajectory tracking with manipulator coupled with UAV

Control of the thrust-propelled system

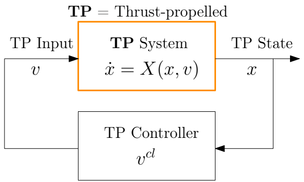


Control of the thrust-propelled system

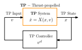


- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

Control of the thrust-propelled system

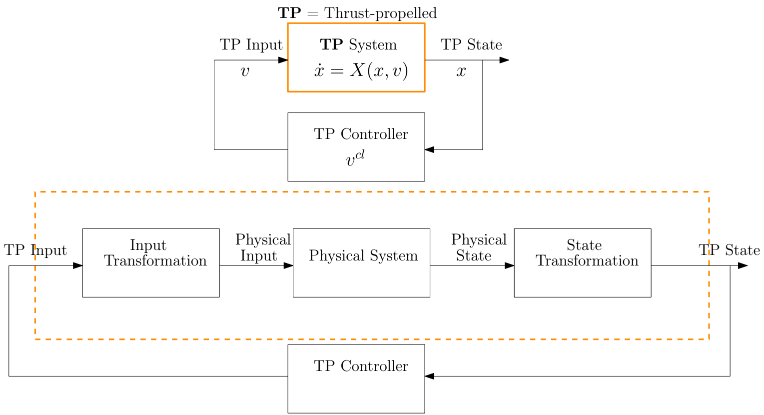


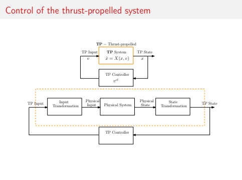
Control of the thrust-propelled system



- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

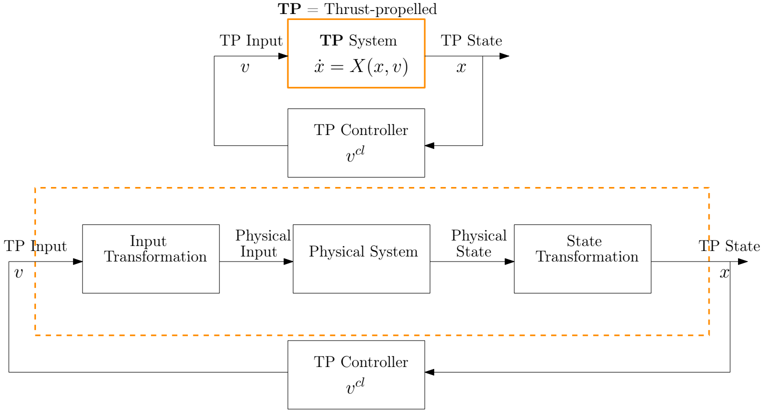
Control of the thrust-propelled system

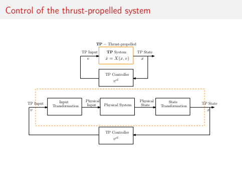




- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

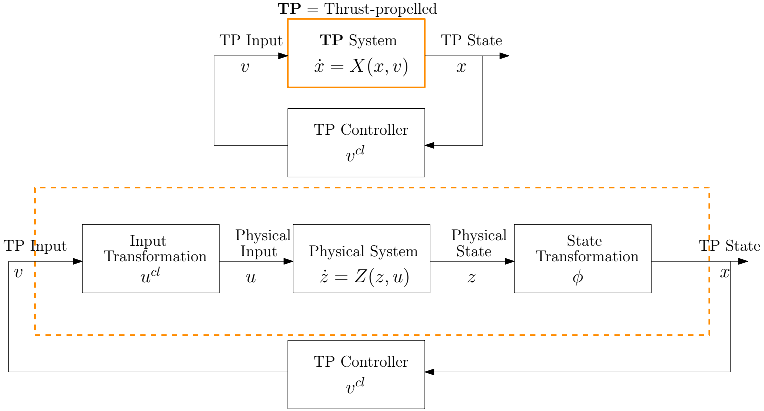
Control of the thrust-propelled system

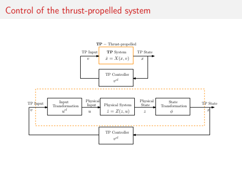




- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

Control of the thrust-propelled system

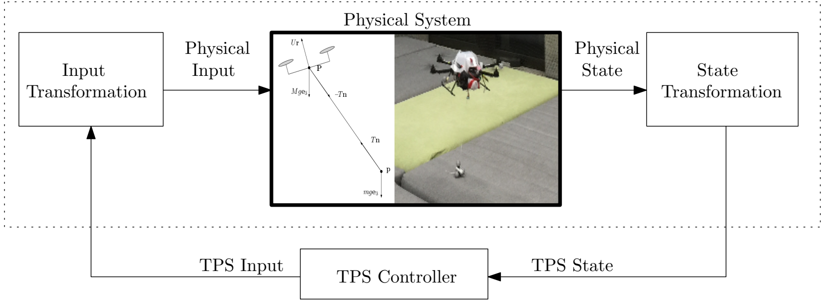


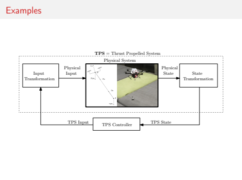


- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

Examples

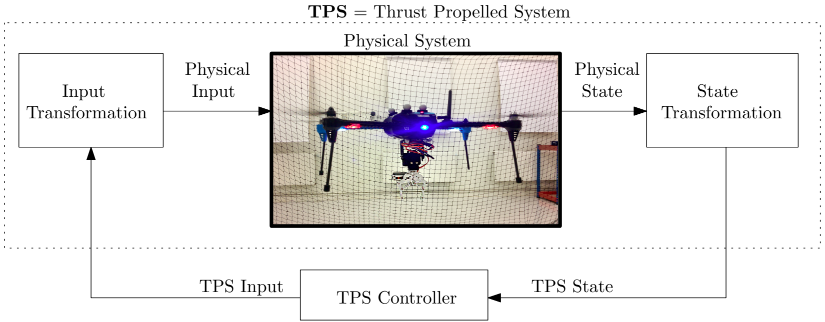
TPS = Thrust Propelled System

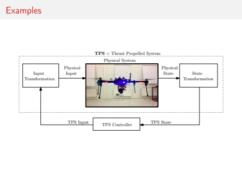




- Slung-load as a thrust-propelled system
- UAV with robotic arm as two decoupled thrust-propelled systems
- Bar tethered to two UAVs as an extended thrust-propelled system

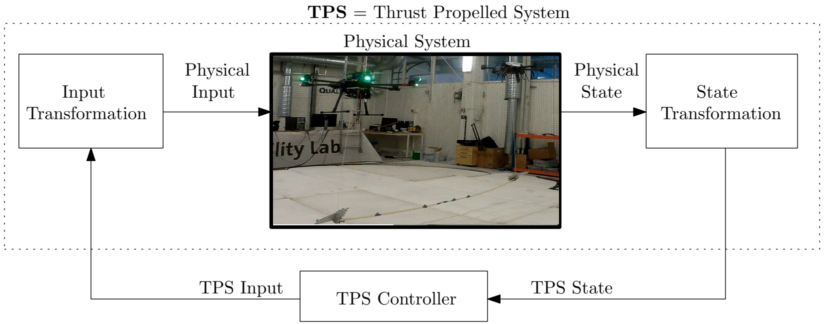
Examples

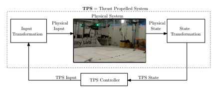




- Slung-load as a thrust-propelled system
- UAV with robotic arm as two decoupled thrust-propelled systems
- Bar tethered to two UAVs as an extended thrust-propelled system

Examples



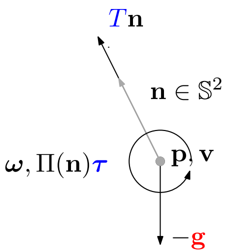


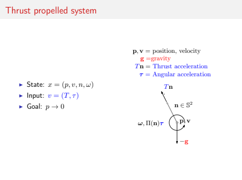
- Slung-load as a thrust-propelled system
- UAV with robotic arm as two decoupled thrust-propelled systems
- Bar tethered to two UAVs as an extended thrust-propelled system

Thrust propelled system

- State: $x = (p, v, n, \omega)$
- Input: $v = (T, \tau)$
- Goal: $p \to 0$

$\mathbf{p}, \mathbf{v}$ = position, velocity
$\mathbf{g}$ =gravity
$T\mathbf{n}$ = Thrust acceleration
$\boldsymbol{\tau}$ = Angular acceleration





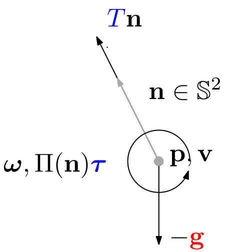
- State is composed of position, velocity, unit vector and angular velocity of unit vector
- Input: thrust and torque
- Time-varying gravity
- Pereira, P. and Dimarogonas, D.V. *Lyapunov-based Generic Controller Design for Thrust-Propelled Underactuated Systems.* IEEE European Control Conference, pp. 594–599, 2016.

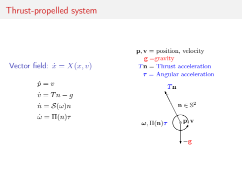
Thrust-propelled system

Vector field: $\dot{x} = X(x, v)$

$$\dot{p} = v$$
$$\dot{v} = Tn - g$$
$$\dot{n} = \mathcal{S}(\omega)n$$
$$\dot{\omega} = \Pi(n)\tau$$

$\mathbf{p}, \mathbf{v}$ = position, velocity
$\mathbf{g}$ = gravity
$T\mathbf{n}$ = Thrust acceleration
$\boldsymbol{\tau}$ = Angular acceleration





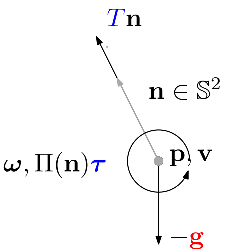
- Notation in this slide can be confusing: v (in blue) used as input to system; v (in black) used as velocity of the system.
- Unit vector along which thrust can be provided
- Torque: torque along unit vector makes unit vector spin around itself, and that is irrelevant, this is why we only use the torque orthogonal to the unit vector
- $\mathcal{S}$: skew operator; Π orthogonal projection operator

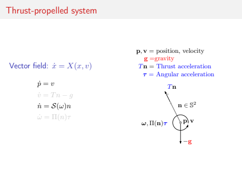
Thrust-propelled system

Vector field: $\dot{x} = X(x, v)$

$$\dot{p} = v$$
$$\dot{v} = Tn - g$$
$$\dot{n} = \mathcal{S}(\omega)n$$
$$\dot{\omega} = \Pi(n)\tau$$

$\mathbf{p}, \mathbf{v}$ = position, velocity
$\mathbf{g}$ =gravity
$T\mathbf{n}$ = Thrust acceleration
$\boldsymbol{\tau}$ = Angular acceleration





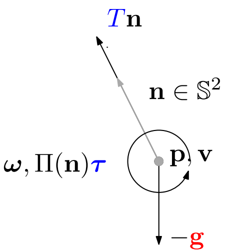
- Two kinematic equations

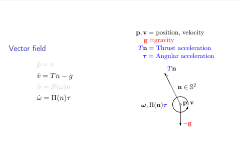
Thrust-propelled system

Vector field

$$\dot{p} = v$$
$$\dot{v} = Tn - g$$
$$\dot{n} = S(\omega)n$$
$$\dot{\omega} = \Pi(n)\tau$$

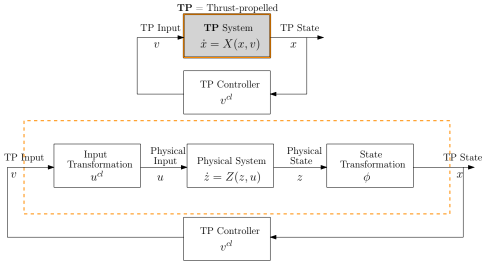
$\mathbf{p}, \mathbf{v}$ = position, velocity
$\mathbf{g}$ =gravity
$T\mathbf{n}$ = Thrust acceleration
$\boldsymbol{\tau}$ = Angular acceleration





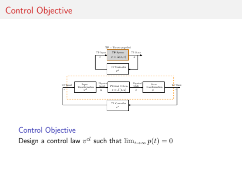
- Two dynamic equations

Control Objective



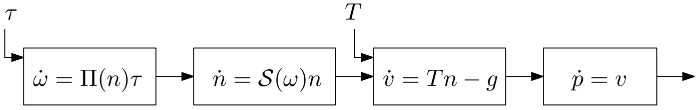
Control Objective

Design a control law v^{cl} such that $\lim_{t\to\infty} p(t) = 0$



- Stabilizing controller
- Steer position of TPS to origin (in physical systems, this will correspond to steering a position tracking error to zero)
- Pereira, P. and Dimarogonas, D.V. *Lyapunov-based Generic Controller Design for Thrust-Propelled Underactuated Systems.* IEEE European Control Conference, pp. 594–599, 2016.

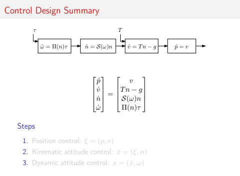
Control Design Summary



$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ Tn - g \\ \mathcal{S}(\omega)n \\ \Pi(n)\tau \end{bmatrix}$$

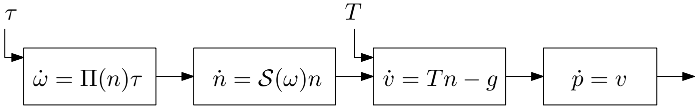
Steps

1. Position control: $\xi = (p, v)$
2. Kinematic attitude control: $\bar{x} = (\xi, n)$
3. Dynamic attitude control: $x = (\bar{x}, \omega)$



- Notice cascaded structure
- First step, we make system behave like a stable second order system
- However, an error is left, and in the next step we steer this error to zero
- Same logic in third step

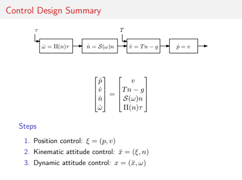
Control Design Summary



$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ Tn - g \\ \mathcal{S}(\omega)n \\ \Pi(n)\tau \end{bmatrix}$$

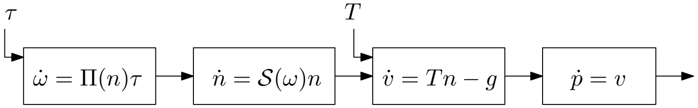
Steps

1. Position control: $\xi = (p, v)$
2. Kinematic attitude control: $\bar{x} = (\xi, n)$
3. Dynamic attitude control: $x = (\bar{x}, \omega)$



- First step, we make system behave like a stable second order system
- However, an error is left, and in the next step we steer this error to zero
- Same logic in third step

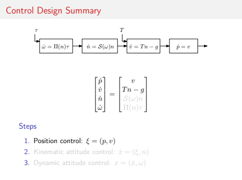
Control Design Summary



$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ Tn - g \\ \mathcal{S}(\omega)n \\ \Pi(n)\tau \end{bmatrix}$$

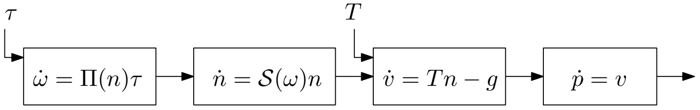
Steps

1. Position control: $\xi = (p, v)$
2. Kinematic attitude control: $\bar{x} = (\xi, n)$
3. Dynamic attitude control: $x = (\bar{x}, \omega)$



- First step, we make system behave like a stable second order system
- However, an error is left, and in the next step we steer this error to zero
- Same logic in third step

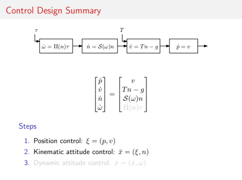
Control Design Summary



$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ Tn - g \\ \mathcal{S}(\omega)n \\ \Pi(n)\tau \end{bmatrix}$$

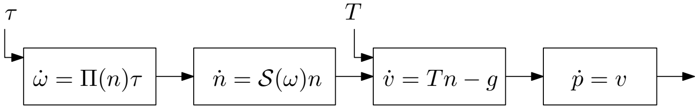
Steps

1. Position control: $\xi = (p, v)$
2. Kinematic attitude control: $\bar{x} = (\xi, n)$
3. Dynamic attitude control: $x = (\bar{x}, \omega)$



- First step, we make system behave like a stable second order system
- However, an error is left, and in the next step we steer this error to zero
- Same logic in third step

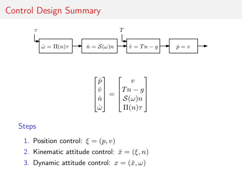
Control Design Summary



$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \\ Tn - g \\ \mathcal{S}(\omega)n \\ \Pi(n)\tau \end{bmatrix}$$

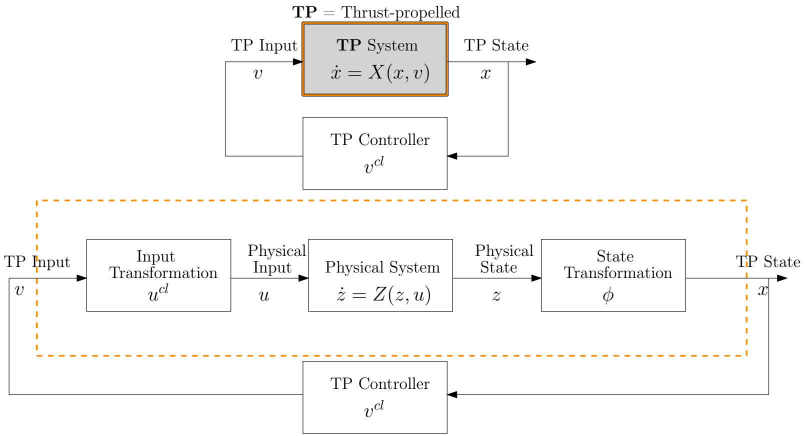
Steps

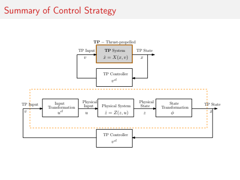
1. Position control: $\xi = (p, v)$
2. Kinematic attitude control: $\bar{x} = (\xi, n)$
3. Dynamic attitude control: $x = (\bar{x}, \omega)$



- First step, we make system behave like a stable second order system
- However, an error is left, and in the next step we steer this error to zero
- Same logic in third step

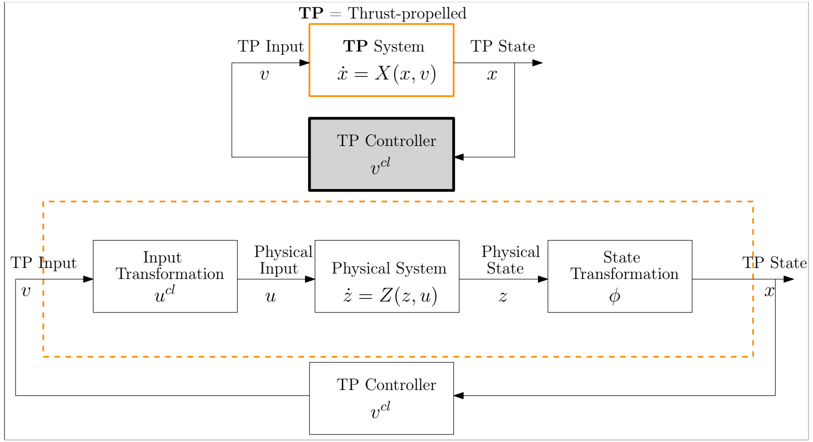
Summary of Control Strategy

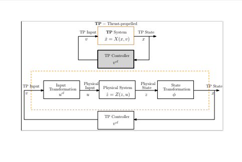




- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

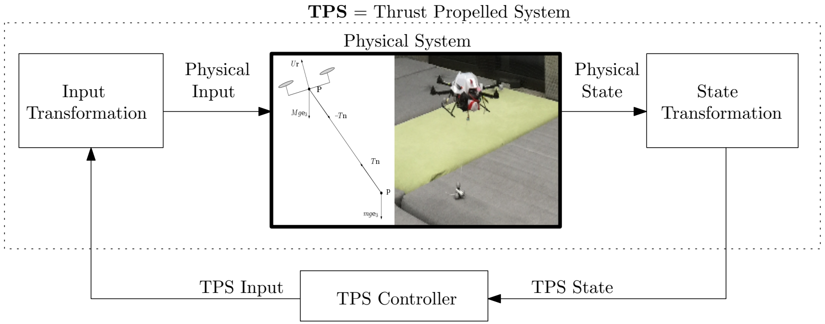
Summary of Control Strategy



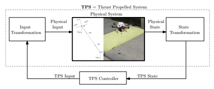


- Design state and input transformations
- Change original vector field, by finding maps that change the input and the state
- Transform physical vector field into vector field of thrust propelled sytem

Example: Slung-load

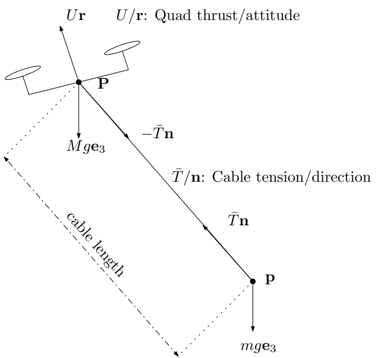


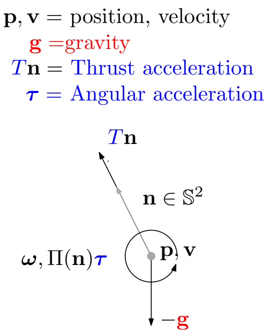
Example: Slung-load



- Pereira, P. and Herzog, M. and Dimarogonas, D. V. *Slung Load Transportation with Single Aerial Vehicle and Disturbance Removal.* IEEE Mediterranean Conference on Control and Automation, pp. 671–676, 2016.

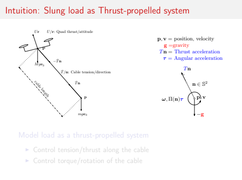
Intuition: Slung load as Thrust-propelled system





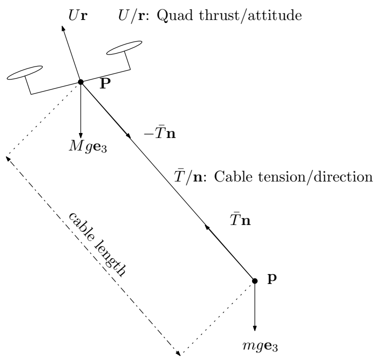
Model load as a thrust-propelled system

- Control tension/thrust along the cable
- Control torque/rotation of the cable

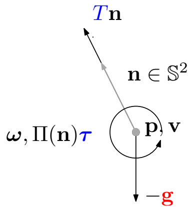


- thrust-propelled system: one where we can provide force along body axis, and where we can rotate that body axis
- load and cable form a thrust propelled system
- control uav actuation along cable direction, to control cable tension (thrust)
- control uav actuation orthogonal to cable to control cable rotation (torque)

Intuition: Slung load as Thrust-propelled system

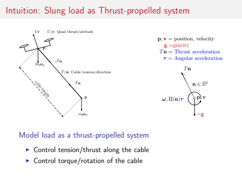


$\mathbf{p}, \mathbf{v}$ = position, velocity
$\mathbf{g}$ = gravity
$T\mathbf{n}$ = Thrust acceleration
$\boldsymbol{\tau}$ = Angular acceleration



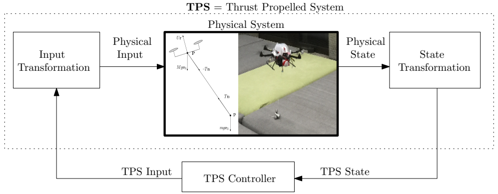
Model load as a thrust-propelled system

- Control tension/thrust along the cable
- Control torque/rotation of the cable



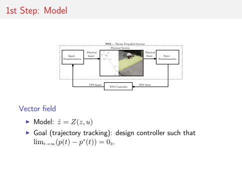
- thrust-propelled system: one where we can provide force along body axis, and where we can rotate that body axis
- load and cable form a thrust propelled system
- control uav actuation along cable direction, to control cable tension (thrust)
- control uav actuation orthogonal to cable to control cable rotation (torque)

1st Step: Model



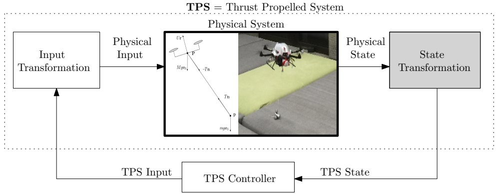
Vector field

- Model: $\dot{z} = Z(z, u)$
- Goal (trajectory tracking): design controller such that $\lim_{t\to\infty}(p(t) - p^\star(t)) = 0_3$.



- Vector field is composed of the velocities and accelerations of the load and uav
- Hybrid system (in general): vector field changes according to state and input (there may be discontinuities in the state as well)
- We must guarantee that with the control law we design that the tension remains positive
- Tracking objective: we want load to track a desired trajectory
- Trajectory needs to be sufficiently smooth ($\mathcal{C}^4$)
- There is an extra constraint on the trajectories we can track (related to the fact that the tension must remain positive)

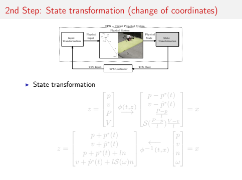
2nd Step: State transformation (change of coordinates)



- State transformation

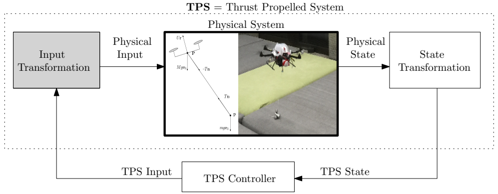
$$z = \begin{bmatrix} p \\ v \\ P \\ V \end{bmatrix} \xrightarrow{\phi(t,z)} \begin{bmatrix} p - p^\star(t) \\ v - \dot{p}^\star(t) \\ \frac{P-p}{l} \\ S(\frac{P-p}{l})\frac{V-v}{l} \end{bmatrix} = x$$

$$z = \begin{bmatrix} p + p^\star(t) \\ v + \dot{p}^\star(t) \\ p + p^\star(t) + ln \\ v + \dot{p}^\star(t) + lS(\omega)n \end{bmatrix} \xleftarrow{\phi^{-1}(t,x)} \begin{bmatrix} p \\ v \\ n \\ \omega \end{bmatrix} = x$$



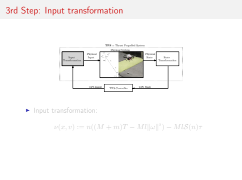
- State transformation = coordinate change
- Design this map using intution I provided before
- Map (for state transformation) must be invertible
- Thrust-propelled system: $\mathbf{p} \to \mathbf{0}$
- Slung load system: $\mathbf{p} - \mathbf{p}^\star(t) \to \mathbf{0}$

3rd Step: Input transformation



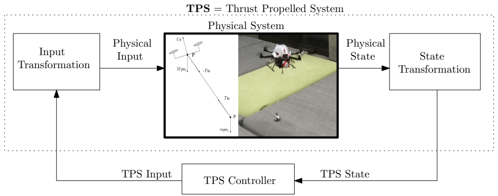
- Input transformation:

$$\nu(x, v) := n((M+m)T - Ml\|\omega\|^2) - MlS(n)\tau$$

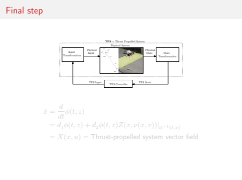


- Input transformation (from $v = (T, \tau)$ to physical input; recall that $v = (T, \tau)$ is the input to the thrust-propelled system)
- Design this map using intution I provided before

Final step

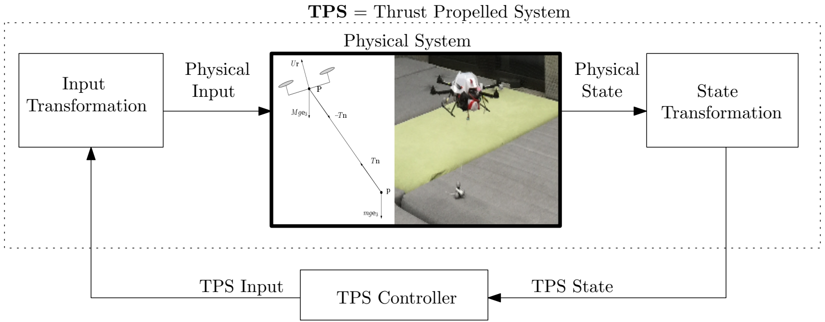


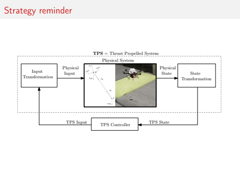
$$
\begin{aligned}
\dot{x} &= \frac{d}{dt}\phi(t,z) \\
&= d_1\phi(t,z) + d_2\phi(t,z)Z(z,\nu(x,v))|_{\phi^{-1}(t,x)} \\
&= X(x,u) = \text{Thrust-propelled system vector field}
\end{aligned}
$$



- Composing the three components: input transformation + vector field + state transformation
- d_1 = derivative w.r.t. 1st argument; d_2 = derivative w.r.t. 2nd argument
- We obtain the TPS vector field, thus we can use any controller for a TPS system for trajectory tracking of a load tethered to a UAV

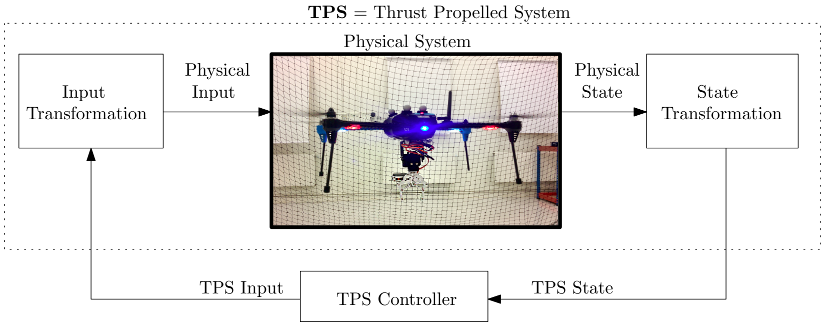
Strategy reminder





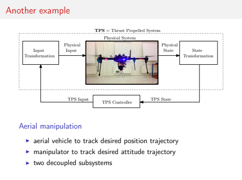
- Reminder of control scheme

Another example



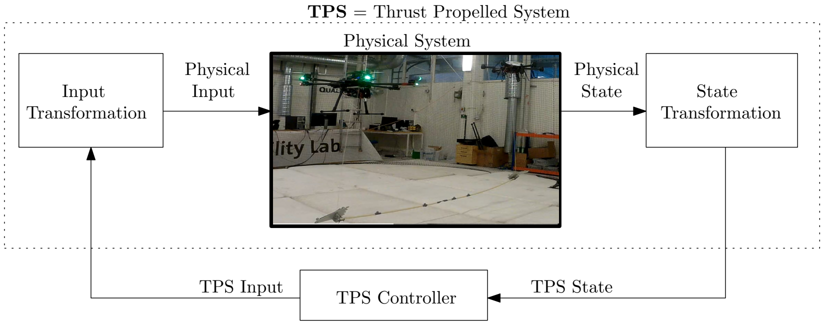
Aerial manipulation

- aerial vehicle to track desired position trajectory
- manipulator to track desired attitude trajectory
- two decoupled subsystems

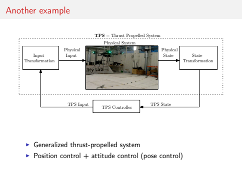


- Control of a system composed of a rigid manipulator and an aerial vehicle
- Disturbance removal: consider a constant thrust input disturbance (we design a disturbance estimator, and penalizes the position tracking error and the manipulation tracking error differently)
- Aerial vehicle: IRIS+; rigid manipulator: aluminum arm with a 1DOF revolute joint and a clamp gripper
- Pereira, P. and Zanella, R. and Dimarogonas, D.V. *Decoupled Design of Controllers for Aerial Manipulation with Quadrotors.* IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4849–4855, 2016.

Another example



- Generalized thrust-propelled system
- Position control + attitude control (pose control)



- Pereira, P. and Dimarogonas, D.V. *Nonlinear Pose Tracking Controller for Bar Tethered to Two Aerial Vehicles with Bounded Linear and Angular Accelerations* CDC, 2017 (to appear).

Disturbance removal

- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

Disturbance removal
- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

- If vector field is affine w.r.t. disturbance, *standard* disturbance estimation techniques may be applied
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine
- check also: Slotine and Li, *Applied nonlinear control*, 1991
- check also (bounded estimators): Cai, Queiroz, Dawson. *A sufficiently smooth projection operator*. TAC 2006

Disturbance removal

- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

Disturbance removal
- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

- If vector field is affine w.r.t. disturbance, *standard* disturbance estimation techniques may be applied
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine
- check also: Slotine and Li, *Applied nonlinear control*, 1991
- check also (bounded estimators): Cai, Queiroz, Dawson. *A sufficiently smooth projection operator*. TAC 2006

Disturbance removal

- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

Disturbance removal
- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

- If vector field is affine w.r.t. disturbance, *standard* disturbance estimation techniques may be applied
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine
- check also: Slotine and Li, *Applied nonlinear control*, 1991
- check also (bounded estimators): Cai, Queiroz, Dawson. *A sufficiently smooth projection operator*. TAC 2006

Disturbance removal

- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

Disturbance removal

- Standard solutions for affine disturbances
- Input disturbance for input affine systems ($\dot{z} = f(z) + g(z)u$)
- Disturbance estimator and (dynamic) control law
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine

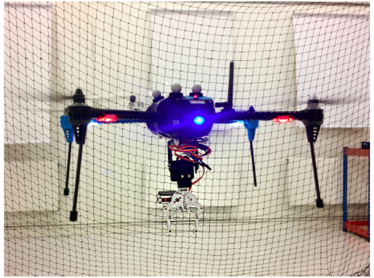
- If vector field is affine w.r.t. disturbance, *standard* disturbance estimation techniques may be applied
- Helicopters dynamics are not input affine
- Multirotors dynamics are input affine
- check also: Slotine and Li, *Applied nonlinear control*, 1991
- check also (bounded estimators): Cai, Queiroz, Dawson. *A sufficiently smooth projection operator*. TAC 2006

Overview

Overview

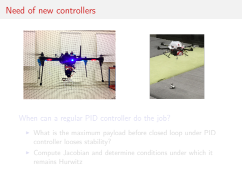
Motivation
Modeling of system of rigid bodies
Control of Thrust-propelled Systems
Conditions for local stability
Summary

Need of new controllers



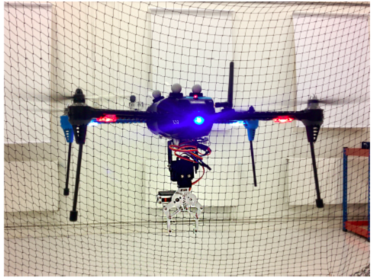
When can a regular PID controller do the job?

- What is the maximum payload before closed loop under PID controller looses stability?
- Compute Jacobian and determine conditions under which it remains Hurwitz



- Linearization around equilibrium point
- Determine conditions on gains or other parameters using Routh-Hurwitz criterion such that Jacobian is Hurwitz
- example condition: if proportinal gain is bigger than . . . , then Jacobian is Hurwitz
- example condition: if extra load is no further than . . . away from CoM, then Jacobian is Hurwitz

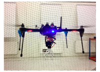
Need of new controllers

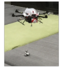


When can a regular PID controller do the job?

- What is the maximum payload before closed loop under PID controller looses stability?
- Compute Jacobian and determine conditions under which it remains Hurwitz

Need of new controllers

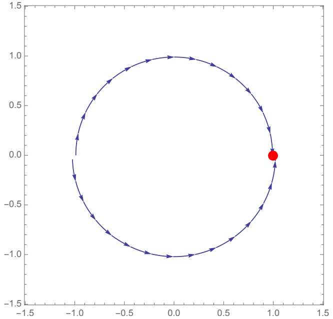




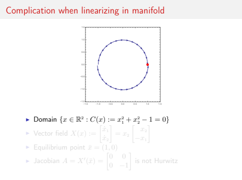
When can a regular PID controller do the job?
- What is the maximum payload before closed loop under PID controller looses stability?
- Compute Jacobian and determine conditions under which it remains Hurwitz

- Linearization around equilibrium point
- Determine conditions on gains or other parameters using Routh-Hurwitz criterion such that Jacobian is Hurwitz
- example condition: if proportinal gain is bigger than . . . , then Jacobian is Hurwitz
- example condition: if extra load is no further than . . . away from CoM, then Jacobian is Hurwitz

Complication when linearizing in manifold

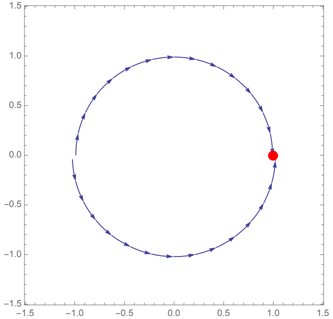


- Domain $\{x \in \mathbb{R}^2 : C(x) := x_1^2 + x_2^2 - 1 = 0\}$
- Vector field $X(x) := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = x_2 \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$
- Equilibrium point $\bar{x} = (1, 0)$
- Jacobian $A = X'(\bar{x}) = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$ is not Hurwitz

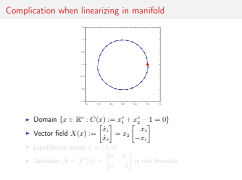


- Recall that physical system (system of rigid bodies and point masses) is constrained in some set (defined, in our case, by holonomic constraints)
- Let us illustrate the problem with a constraint that defines a circle in $\mathbb{R}^2$.

Complication when linearizing in manifold

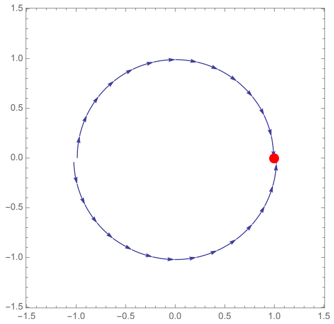


- Domain $\{x \in \mathbb{R}^2 : C(x) := x_1^2 + x_2^2 - 1 = 0\}$
- Vector field $X(x) := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = x_2 \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$
- Equilibrium point $\bar{x} = (1, 0)$
- Jacobian $A = X'(\bar{x}) = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$ is not Hurwitz

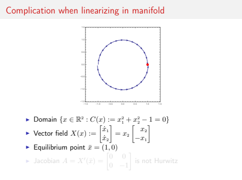


- Recall that physical system (system of rigid bodies and point masses) is constrained in some set (defined, in our case, by holonomic constraints)
- Let us illustrate the problem with a constraint that defines a circle in $\mathbb{R}^2$.

Complication when linearizing in manifold

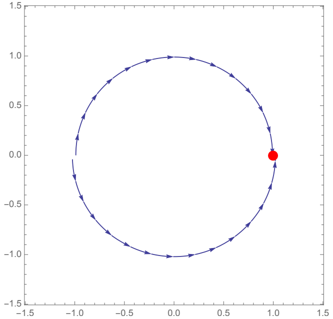


- Domain $\{x \in \mathbb{R}^2 : C(x) := x_1^2 + x_2^2 - 1 = 0\}$
- Vector field $X(x) := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = x_2 \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$
- Equilibrium point $\bar{x} = (1, 0)$
- Jacobian $A = X'(\bar{x}) = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$ is not Hurwitz

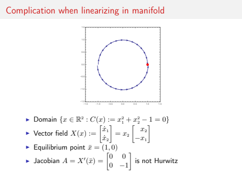


- Recall that physical system (system of rigid bodies and point masses) is constrained in some set (defined, in our case, by holonomic constraints)
- Let us illustrate the problem with a constraint that defines a circle in $\mathbb{R}^2$.

Complication when linearizing in manifold

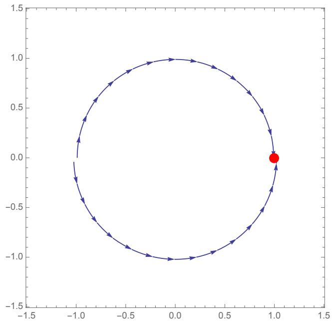


- Domain $\{x \in \mathbb{R}^2 : C(x) := x_1^2 + x_2^2 - 1 = 0\}$
- Vector field $X(x) := \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = x_2 \begin{bmatrix} x_2 \\ -x_1 \end{bmatrix}$
- Equilibrium point $\bar{x} = (1, 0)$
- Jacobian $A = X'(\bar{x}) = \begin{bmatrix} 0 & 0 \\ 0 & -1 \end{bmatrix}$ is not Hurwitz



- Recall that physical system (system of rigid bodies and point masses) is constrained in some set (defined, in our case, by holonomic constraints)
- Let us illustrate the problem with a constraint that defines a circle in $\mathbb{R}^2$.

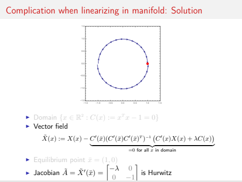
Complication when linearizing in manifold: Solution



- Domain $\{x \in \mathbb{R}^2 : C(x) := x^T x - 1 = 0\}$
- Vector field

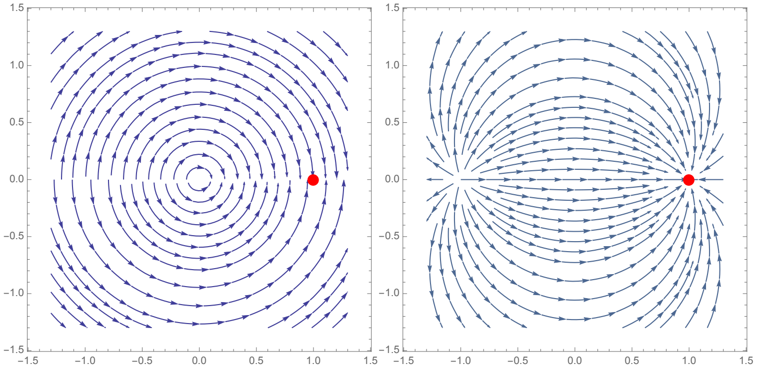
$$\tilde{X}(x) := X(x) - \underbrace{C'(\bar{x})(C'(\bar{x})C'(\bar{x})^T)^{-1}\left(C'(x)X(x) + \lambda C(x)\right)}_{=0 \text{ for all } x \text{ in domain}}$$

- Equilibrium point $\bar{x} = (1, 0)$
- Jacobian $\tilde{A} = \tilde{X}'(\bar{x}) = \begin{bmatrix} -\lambda & 0 \\ 0 & -1 \end{bmatrix}$ is Hurwitz

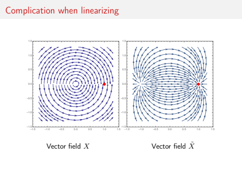


- λ is a positive number
- This method is an artifact/gimmick that we use, so that we can invoke results from linearization theorems
- See also: Palis and de Melo. *Geometric theory of dynamical systems*, 1982 (Chapter 2).

Complication when linearizing

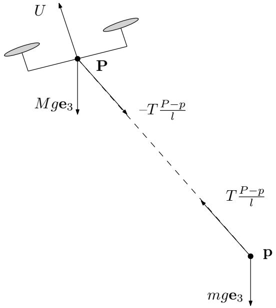


Vector field X Vector field $\tilde{X}$

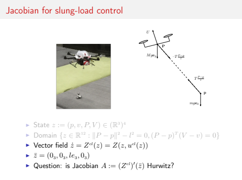


- On the circle, vector fiels are exactly the same
- Outside the circle, they are not
- Another option: parametrize circle with angle, and study stability of equilibrium with the vector field in the "angle coordinate" (however, more complicated sets, require more complicated parametrizations)

Jacobian for slung-load control

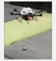


- State $z := (p, v, P, V) \in (\mathbb{R}^3)^4$
- Domain $\{z \in \mathbb{R}^{12} : \|P-p\|^2 - l^2 = 0, (P-p)^T(V-v) = 0\}$
- Vector field $\dot{z} = Z^{cl}(z) = Z(z, u^{cl}(z))$
- $\bar{z} = (0_3, 0_3, le_3, 0_3)$
- Question: is Jacobian $A := (Z^{cl})'(\bar{z})$ Hurwitz?



- Equilibrium $\bar{z}$
- u^{cl}: control law – in this case a PID control law based on position and velocity of UAV
- Jacobian is unstructured (find similarity transformation that exposes its structure)
- $(Z^{cl})' =$ “$\frac{dZ^{cl}}{dz}$”
- P. O. Pereira and D. V. Dimarogonas, *Stability of load lifting by a quadrotor under attitude control delay*, 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3287-3292.

Jacobian for slung-load control



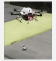
$$PAP^{-1} = \begin{bmatrix} A_z & 0 & 0 & \star \\ 0 & A_x & 0 & \star \\ 0 & 0 & A_y & \star \\ 0 & 0 & 0 & -\lambda I_2 \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{M}{M+m}k_{p,z} & -\frac{M}{M+m}k_{d,z} \end{bmatrix}}_{=:A_z}\begin{bmatrix} z \\ \dot{z} \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g}{l}k_{p,x} & -\frac{g}{l}k_{d,x} & -k_{p,x} - \frac{g}{l}\frac{M+m}{M} & -k_{d,x} \end{bmatrix}}_{=:A_x}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix}$$

- Jacobian is similar to a block diagonal matrix (thus, if each block diagonal matrix is Hurwitz, then the Jacobian is Hurwitz)
- z is associated to the z motion of the load (which behaves as a second order system – this is the case for a PD control law)
- x is associated to the x motion of the load (which behaves as a fourth order system)
- By the Routh-Hurwitz's criterion, both A_z and A_x are Hurwitz
- Lesson: PD control law does the job (local result though)

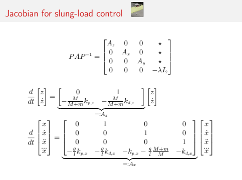
Jacobian for slung-load control



$$PAP^{-1} = \begin{bmatrix} A_z & 0 & 0 & \star \\ 0 & A_x & 0 & \star \\ 0 & 0 & A_y & \star \\ 0 & 0 & 0 & -\lambda I_2 \end{bmatrix}$$

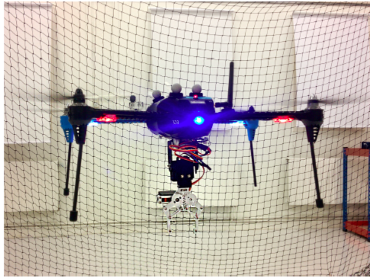
$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{M}{M+m}k_{p,z} & -\frac{M}{M+m}k_{d,z} \end{bmatrix}}_{=:A_z} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}$$

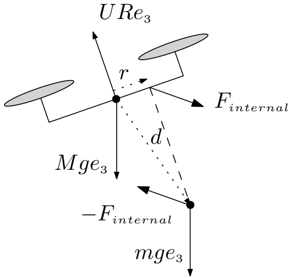
$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{g}{l}k_{p,x} & -\frac{g}{l}k_{d,x} & -k_{p,x} - \frac{g}{l}\frac{M+m}{M} & -k_{d,x} \end{bmatrix}}_{=:A_x} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix}$$

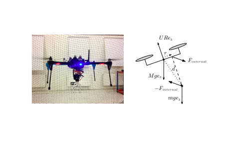


- Jacobian is similar to a block diagonal matrix (thus, if each block diagonal matrix is Hurwitz, then the Jacobian is Hurwitz)
- z is associated to the z motion of the load (which behaves as a second order system – this is the case for a PD control law)
- x is associated to the x motion of the load (which behaves as a fourth order system)
- By the Routh-Hurwitz's criterion, both A_z and A_x are Hurwitz
- Lesson: PD control law does the job (local result though)

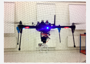
Jacobian for UAV with off-balanced load







- Same idea applied to a different system
- Does a *standard* PID control law does the job when the UAV is rigidly attached to some load (fixed in the body frame of the UAV)
- $d = (d_1, d_2, d_3)$: position of load w.r.t. UAV and expressed in the UAV body frame
- $r = (r_1, r_2, r_3)$: where internal force is applied – $r = (0, 0, 0)$ if arm is exactly placed on UAV CoM
- For simplicity, in the linearization, we assume that $r = (0, 0, 0)$.

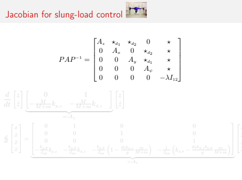
Jacobian for slung-load control 

$$PAP^{-1} = \begin{bmatrix} A_z & \star_{d_1} & \star_{d_2} & 0 & \star \\ 0 & A_x & 0 & \star_{d_2} & \star \\ 0 & 0 & A_y & \star_{d_1} & \star \\ 0 & 0 & 0 & A_\psi & \star \\ 0 & 0 & 0 & 0 & -\lambda I_{12} \end{bmatrix}$$

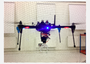
$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{M}{M+m}k_{p,z} & -\frac{M}{M+m}k_{d,z} \end{bmatrix}}_{=:A_z} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_{p,\theta}}{J_{yy}}k_{p,x} & -\frac{k_{p,\theta}}{J_{yy}}k_{d,x} & -\frac{k_{p,\theta}}{J_{yy}}\left(1 - \frac{d_3 k_{p,z}}{g}\frac{m}{M+m}\right) & -\frac{1}{J_{yy}}\left(k_{d,\theta} - \frac{d_3 k_{d,z} k_{p,\theta}}{g}\frac{m}{M+m}\right) \end{bmatrix}}_{=:A_x}$$



- Jacobian is similar to block triangular matrix (thus, if each block diagonal is Hurwitz, then the Jacobian is Hurwitz)
- $\star_{d_1}$ means that there are non-zero terms, which vanish if $d_1 = 0$ (same for d_2)
- z is related to the z motion of the system's CoM (which behaves as a second order system)
- x (and y) is related to the x (and y) motion of the system's CoM (which behaves as a fourth order system)
- z motion is coupled to x (y) motion via d_1 (d_2).
- We can find precise conditions on d_3 such that A_z is Hurwitz

Jacobian for slung-load control 

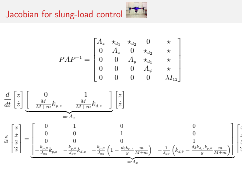
$$PAP^{-1} = \begin{bmatrix} A_z & \star_{d_1} & \star_{d_2} & 0 & \star \\ 0 & A_x & 0 & \star_{d_2} & \star \\ 0 & 0 & A_y & \star_{d_1} & \star \\ 0 & 0 & 0 & A_\psi & \star \\ 0 & 0 & 0 & 0 & -\lambda I_{12} \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} z \\ \dot{z} \end{bmatrix} \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{M}{M+m}k_{p,z} & -\frac{M}{M+m}k_{d,z} \end{bmatrix}}_{=:A_z} \begin{bmatrix} z \\ \dot{z} \end{bmatrix}$$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ \dddot{x} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_{p,\theta}}{J_{yy}}k_{p,x} & -\frac{k_{p,\theta}}{J_{yy}}k_{d,x} & -\frac{k_{p,\theta}}{J_{yy}}\left(1 - \frac{d_3 k_{p,z}}{g}\frac{m}{M+m}\right) & -\frac{1}{J_{yy}}\left(k_{d,\theta} - \frac{d_3 k_{d,z} k_{p,\theta}}{g}\frac{m}{M+m}\right) \end{bmatrix}}_{=:A_x}$$

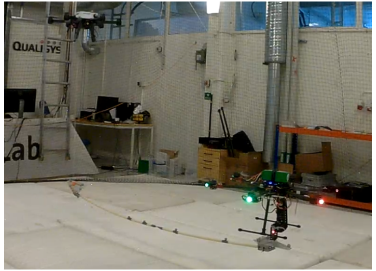
Pedro

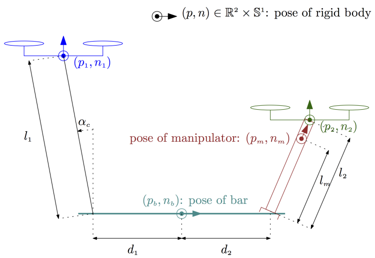
 

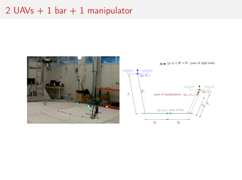


- Jacobian is similar to block triangular matrix (thus, if each block diagonal is Hurwitz, then the Jacobian is Hurwitz)
- $\star_{d_1}$ means that there are non-zero terms, which vanish if $d_1 = 0$ (same for d_2)
- z is related to the z motion of the system's CoM (which behaves as a second order system)
- x (and y) is related to the x (and y) motion of the system's CoM (which behaves as a fourth order system)
- z motion is coupled to x (y) motion via d_1 (d_2).
- We can find precise conditions on d_3 such that A_z is Hurwitz

2 UAVs + 1 bar + 1 manipulator

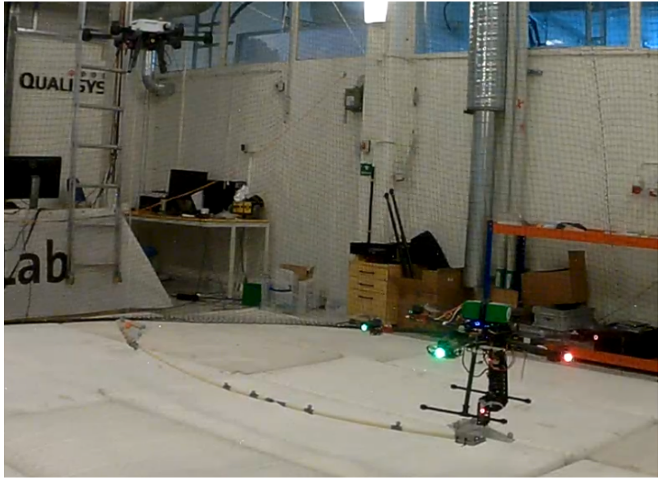






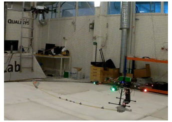
- Similar analysis
- Roque, P., Pereira, P. and Dimarogonas, D.V. *Heterogeneous Collaborative Pose Stabilization of a Bar Via Aerial Manipulation and Tethering* ICRA, 2018 (submitted).

Experiment



https://youtu.be/NADR9_VffBk

Experiment



https://youtu.be/NADR9_VffBk

- See video at https://youtu.be/NADR9_VffBk

Summary & Final remarks

Summary

- Modeling
- Control of thrust propelled systems
- Conditions for local stability

Summary & Final remarks

Summary

- Modeling
- Control of thrust propelled systems
- Conditions for local stability

Thank you!

Thank you!