Control Framework for Slung Load Transportation with Two Aerial Vehicles

Pedro Pereira and Dimos V. Dimarogonas

Department of Automatic Control,
KTH Royal Institute of Technology

December 13th

Control Framework for Slung Load Transportation with Two Aerial Vehicles

Pedro Pereira and Dimos V. Dimarogonas

Department of Automatic Control,
KTH Royal Institute of Technology

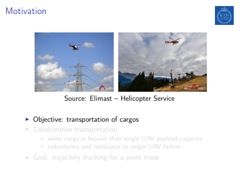
December 13th

- Submission number: 1217
- https://css.paperplaza.net/conferences/conferences/CDC17/program/CDC17_ContentListWeb_3.html
- Schedule Code: ThB17. Room: **218**
- Tentative time of presentation: Thursday December 14, 2017, 10:20–10:40 (GMT+11)
- Session: Lyapunov Methods I
- Chair: Morten Hovd
- Co-chair: Marc Jungers

Motivation

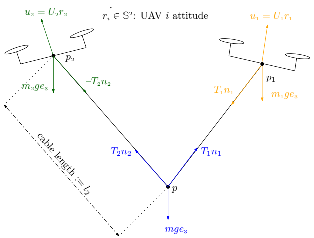
Source: Elimast – Helicopter Service

- Objective: transportation of cargos
- Collaborative transportation:
  - when cargo is heavier than single UAV payload capacity
  - redundancy and resistance to single UAV failure
- Goal: trajectory tracking for a point mass

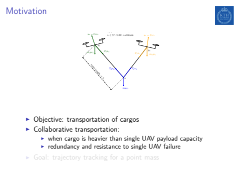


- Cargo transportation by Elimast: check https://www.elimast.it/en/
- Tethered *vs* manipulator-endowed transportation
  - goal: trajectory tracking for a point mass
  - cable is mechanically simple and light
  - manipulator provides extra degrees of freedom
- Cable does not require a power supply (if it is to be retracted, it does)
- Manipulator is mechanically complex and heavy

Motivation

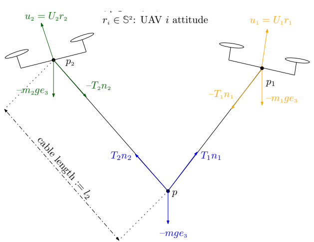


- Objective: transportation of cargos
- Collaborative transportation:
  - when cargo is heavier than single UAV payload capacity
  - redundancy and resistance to single UAV failure
- Goal: trajectory tracking for a point mass

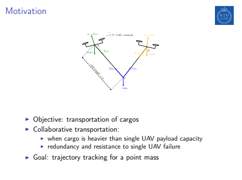


- Cargo transportation by Elimast: check https://www.elimast.it/en/
- Tethered *vs* manipulator-endowed transportation
  - goal: trajectory tracking for a point mass
  - cable is mechanically simple and light
  - manipulator provides extra degrees of freedom
- Cable does not require a power supply (if it is to be retracted, it does)
- Manipulator is mechanically complex and heavy

Motivation



- Objective: transportation of cargos
- Collaborative transportation:
  - when cargo is heavier than single UAV payload capacity
  - redundancy and resistance to single UAV failure
- Goal: trajectory tracking for a point mass

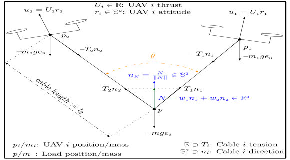


Motivation
- Objective: transportation of cargos
- Collaborative transportation:
  - when cargo is heavier than single UAV payload capacity
  - redundancy and resistance to single UAV failure
- Goal: trajectory tracking for a point mass

- Cargo transportation by Elimast: check https://www.elimast.it/en/
- Tethered *vs* manipulator-endowed transportation
  - goal: trajectory tracking for a point mass
  - cable is mechanically simple and light
  - manipulator provides extra degrees of freedom
- Cable does not require a power supply (if it is to be retracted, it does)
- Manipulator is mechanically complex and heavy

Problem and Strategy

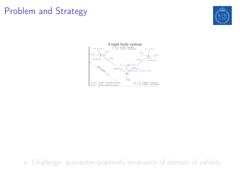
3 rigid body system



- Challenge: guarantee positively invariance of domain of validity

2017-12-11

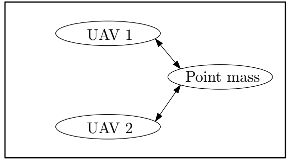
Slung Load Transportation
└Problem and Strategy
└Problem and Strategy



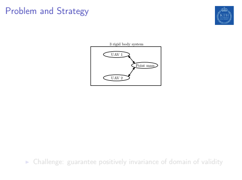
- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

3 rigid body system

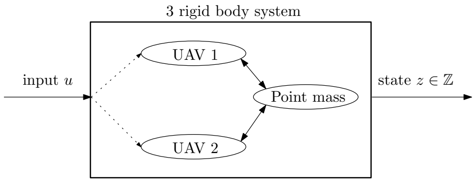


- Challenge: guarantee positively invariance of domain of validity

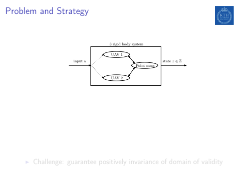


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

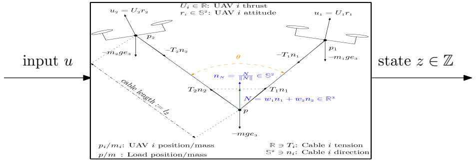


- Challenge: guarantee positively invariance of domain of validity

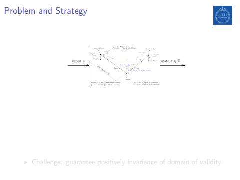


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

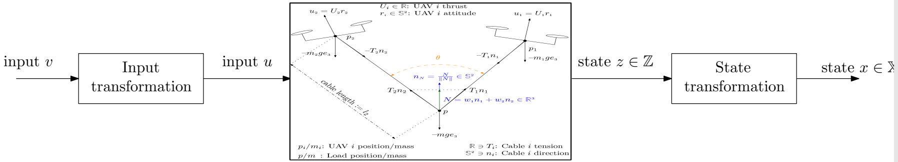


- Challenge: guarantee positively invariance of domain of validity

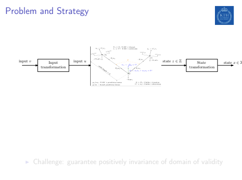


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

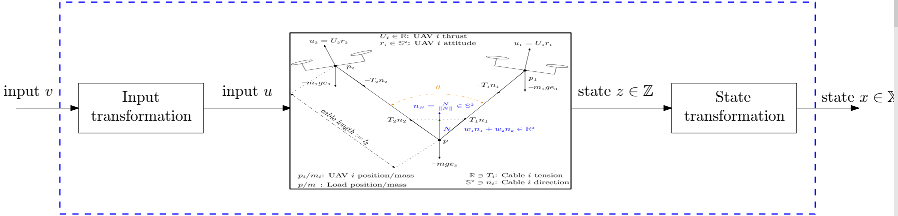


- Challenge: guarantee positively invariance of domain of validity

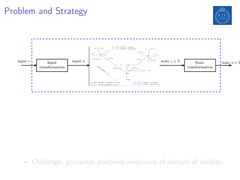


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy



- Challenge: guarantee positively invariance of domain of validity



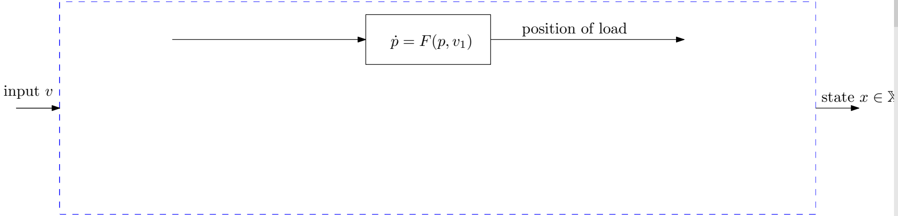
- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

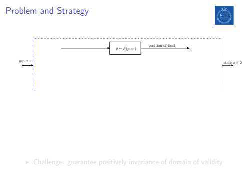
- Challenge: guarantee positively invariance of domain of validity

- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

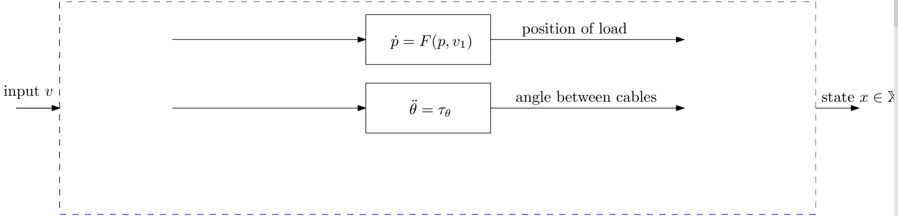


- Challenge: guarantee positively invariance of domain of validity

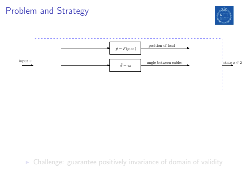


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

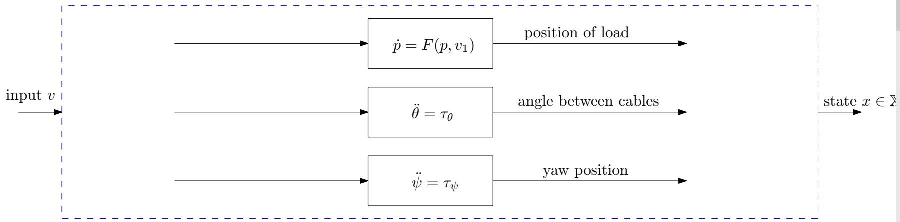


▶ Challenge: guarantee positively invariance of domain of validity

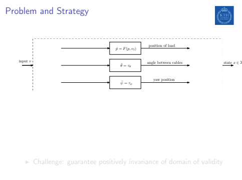


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

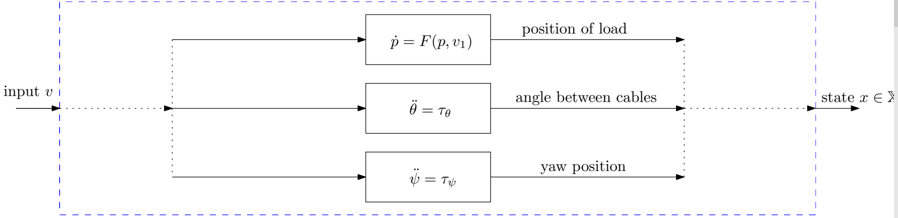


- Challenge: guarantee positively invariance of domain of validity

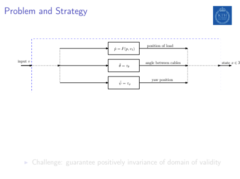


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

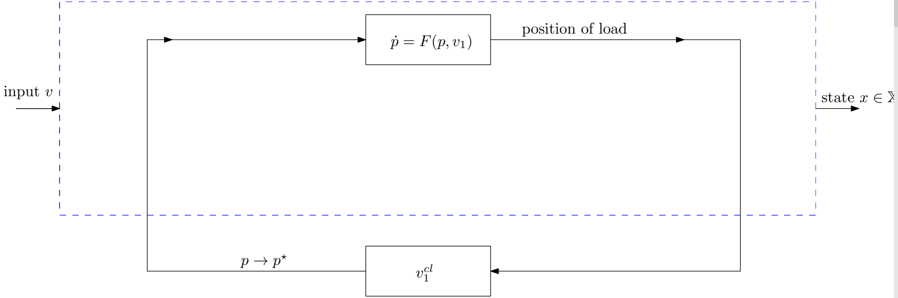


- Challenge: guarantee positively invariance of domain of validity

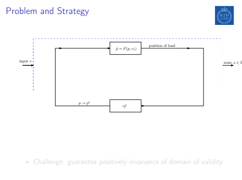


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

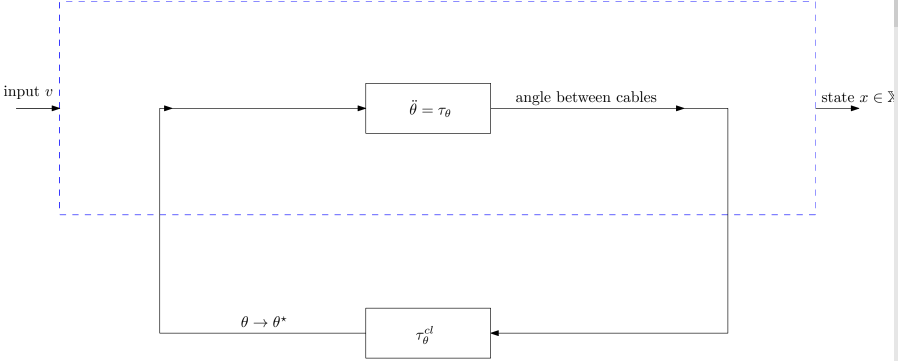


- Challenge: guarantee positively invariance of domain of validity

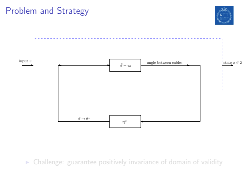


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

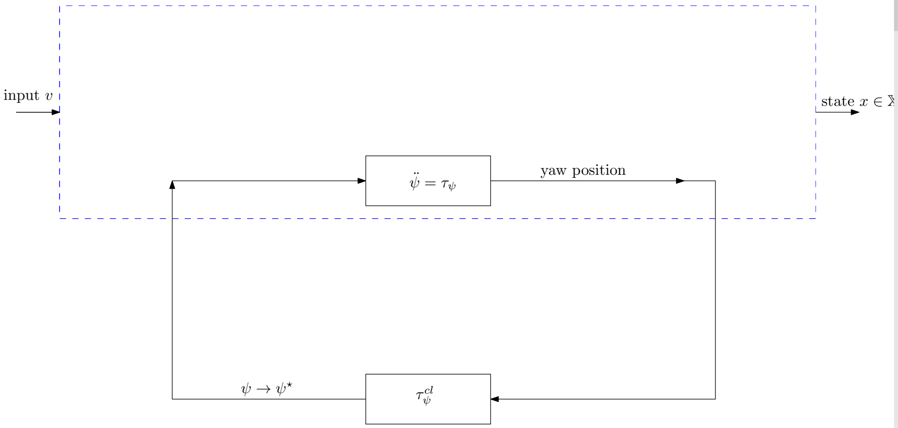


- Challenge: guarantee positively invariance of domain of validity

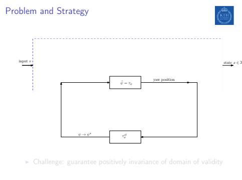


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

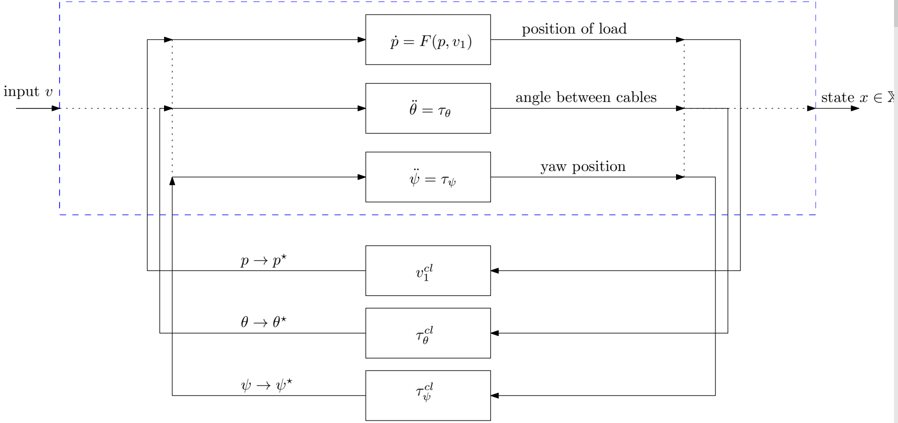


- Challenge: guarantee positively invariance of domain of validity

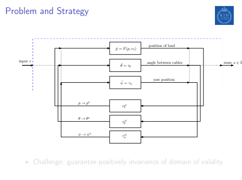


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

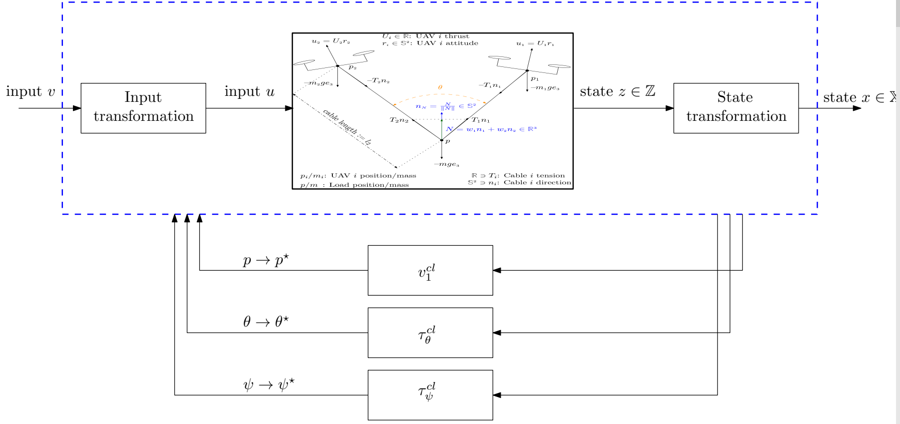


- Challenge: guarantee positively invariance of domain of validity

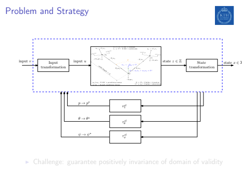


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

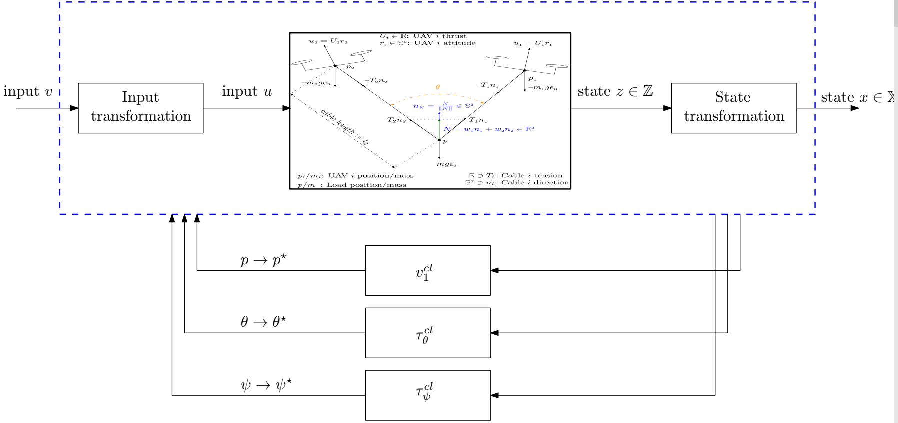


- Challenge: guarantee positively invariance of domain of validity

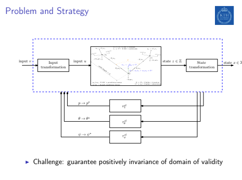


- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Problem and Strategy

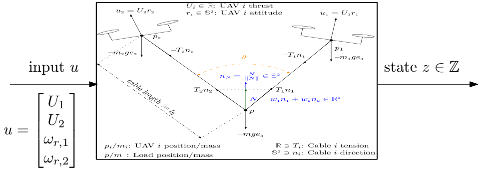


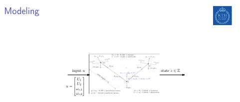
- Challenge: guarantee positively invariance of domain of validity



- steps for solving the problem
- model system as coupling between 3 bodies
- Find input and state transformation, such that composition yields three decouple vector fields
- Design controllers independently
- Major road block: design controllers that guarantee we stay in domain of validity (where coordinate change is valid)

Modeling

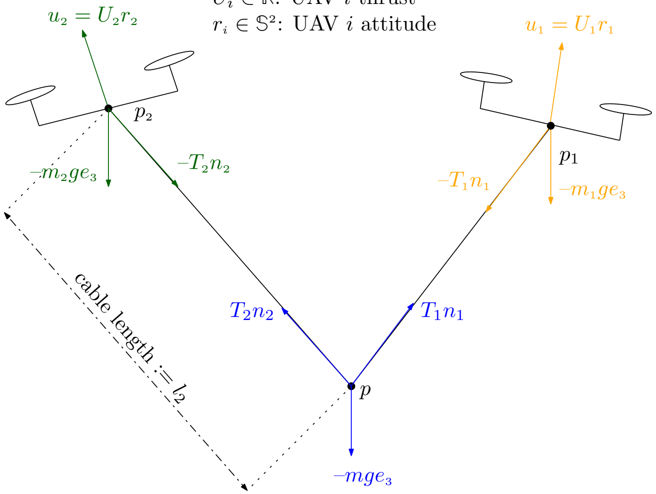




- Define state, input and state equations

Modeling

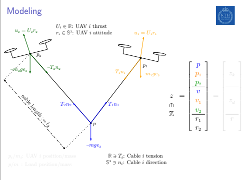
$U_i \in \mathbb{R}$: UAV i thrust
$r_i \in \mathbb{S}^2$: UAV i attitude



$$z \in \mathbb{Z} = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}$$

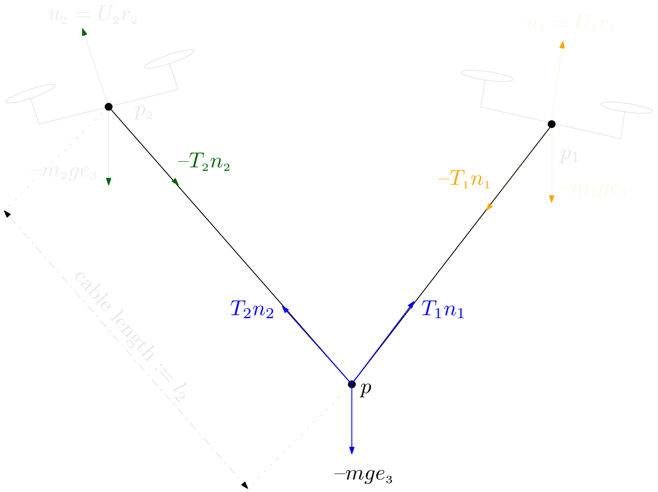
p_i/m_i: UAV i position/mass
p/m : Load position/mass

$\mathbb{R} \ni T_i$: Cable i tension
$\mathbb{S}^2 \ni n_i$: Cable i direction



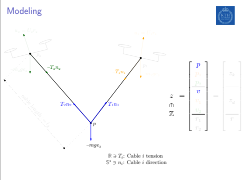
- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Modeling



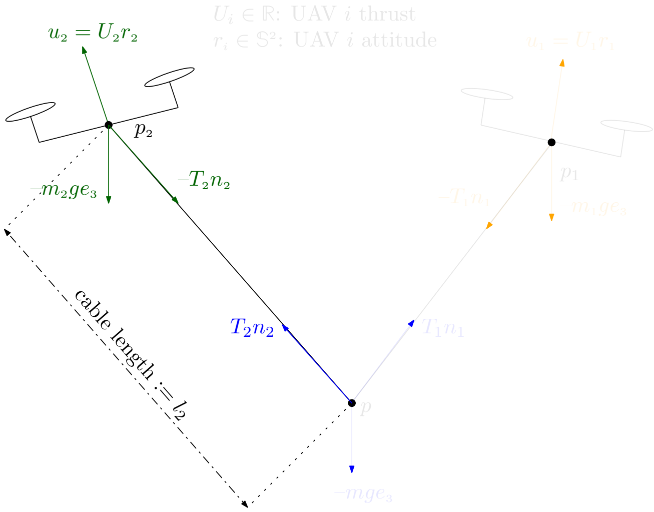
$$z \in \mathbb{Z} = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}$$

$\mathbb{R} \ni T_i$: Cable i tension
$\mathbb{S}^2 \ni n_i$: Cable i direction

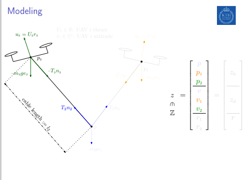


- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Modeling



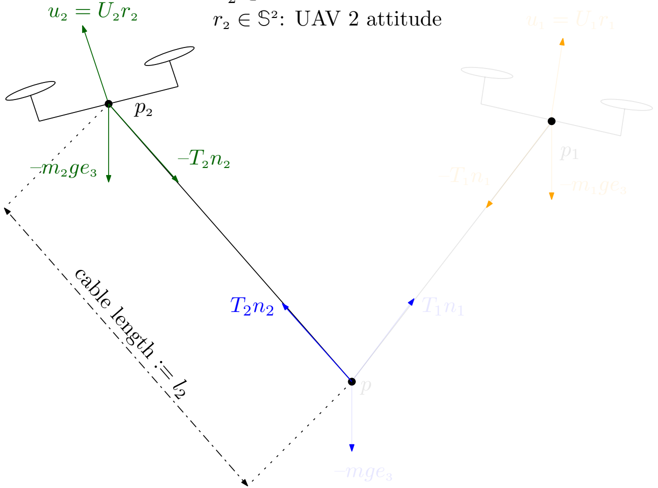
$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}, \quad z \in \mathbb{Z}$$



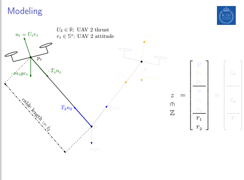
- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Modeling

$U_2 \in \mathbb{R}$: UAV 2 thrust
$r_2 \in \mathbb{S}^2$: UAV 2 attitude

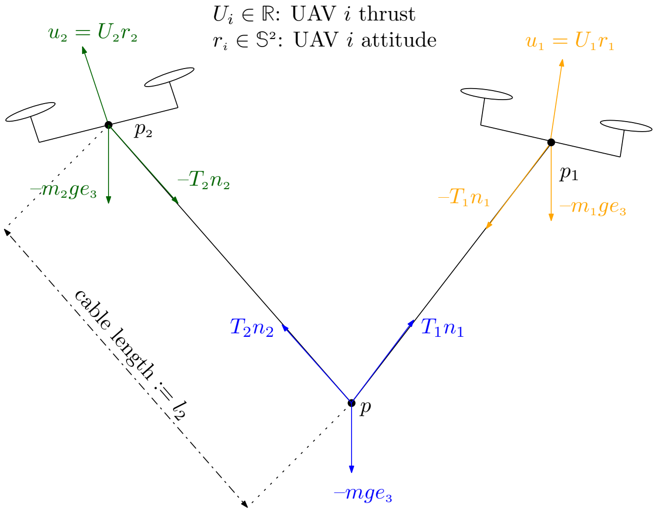


$$z \in \mathbb{Z} = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}$$



- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Modeling



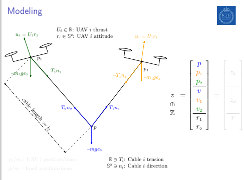
$$z \in \mathbb{Z} = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}$$

p_i/m_i: UAV i position/mass

p/m : Load position/mass

$\mathbb{R} \ni T_i$: Cable i tension

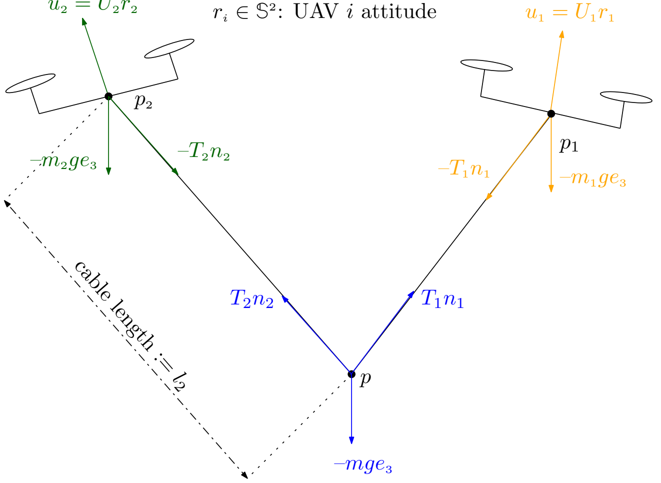
$\mathbb{S}^2 \ni n_i$: Cable i direction



- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Modeling

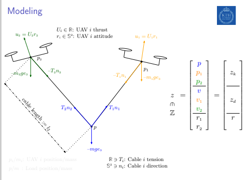
$U_i \in \mathbb{R}$: UAV i thrust
$r_i \in \mathbb{S}^2$: UAV i attitude



p_i/m_i: UAV i position/mass
p/m : Load position/mass

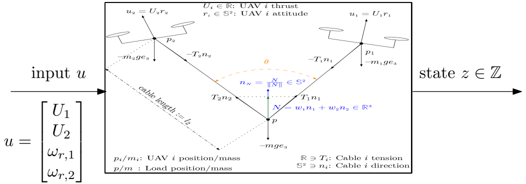
$\mathbb{R} \ni T_i$: Cable i tension
$\mathbb{S}^2 \ni n_i$: Cable i direction

$$z \in \mathbb{Z} = \begin{bmatrix} p \\ p_1 \\ p_2 \\ \hline v \\ v_1 \\ v_2 \\ \hline r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} z_k \\ \hline z_d \\ \hline r \end{bmatrix}$$

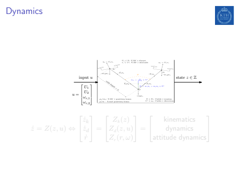


- State decomposition
- Linear and angular positions of bar + linear and angular velocities of bar
- Linear position of UAV + linear velocity of UAV (for both UAVs)
- Angular position of UAV, and we assume we control its angular velocity
- Integral states for the PID control laws

Dynamics

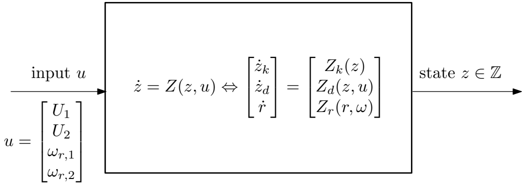


$$\dot{z} = Z(z,u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z,u) \\ Z_r(r,\omega) \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \\ \text{attitude dynamics} \end{bmatrix}$$

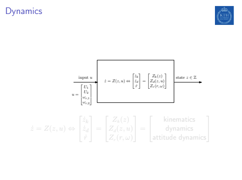


- Replace model with “open-loop vector field”
- Explain three components individually next: kinematics + dynamics + UAV attitude dynamics

Dynamics

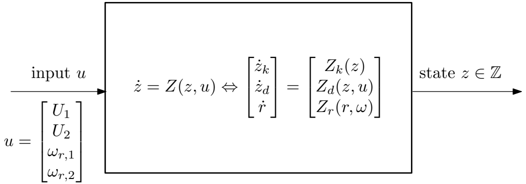


$$\dot{z} = Z(z,u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z,u) \\ Z_r(r,\omega) \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \\ \text{attitude dynamics} \end{bmatrix}$$

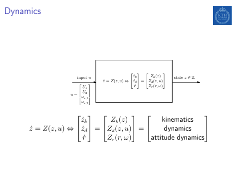


- Replace model with “open-loop vector field”
- Explain three components individually next: kinematics + dynamics + UAV attitude dynamics

Dynamics

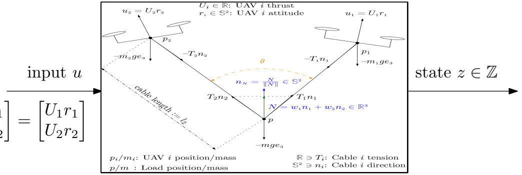


$$\dot{z} = Z(z,u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \\ \dot{r} \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z,u) \\ Z_r(r,\omega) \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \\ \text{attitude dynamics} \end{bmatrix}$$



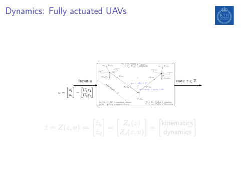
- Replace model with “open-loop vector field”
- Explain three components individually next: kinematics + dynamics + UAV attitude dynamics

Dynamics: Fully actuated UAVs



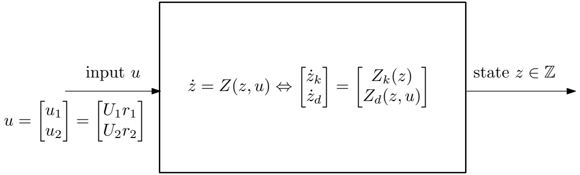
$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U_1 r_1 \\ U_2 r_2 \end{bmatrix}$$

$$\dot{z} = Z(z,u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z,u) \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \end{bmatrix}$$

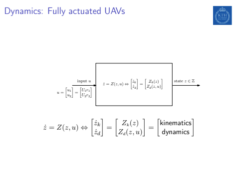


- Fully actuated UAVs: no attitude dynamics for UAVs
- We control the attitude of UAVs immediately
- Explain two components individually next: kinematics + dynamics

Dynamics: Fully actuated UAVs



$$\dot{z} = Z(z,u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z,u) \end{bmatrix} = \begin{bmatrix} \text{kinematics} \\ \text{dynamics} \end{bmatrix}$$

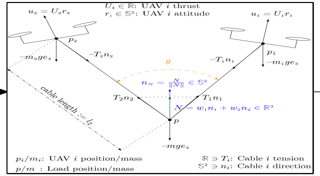


- Fully actuated UAVs: no attitude dynamics for UAVs
- We control the attitude of UAVs immediately
- Explain two components individually next: kinematics + dynamics

Kinematics

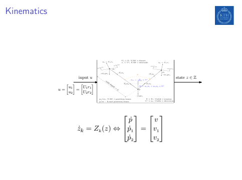
input u

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U_1 r_1 \\ U_2 r_2 \end{bmatrix}$$



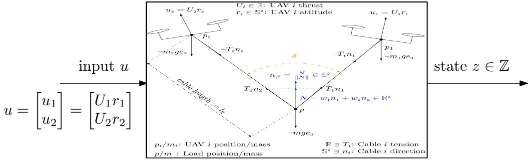
state $z \in \mathbb{Z}$

$$\dot{z}_k = Z_k(z) \Leftrightarrow \begin{bmatrix} \dot{p} \\ \dot{p}_1 \\ \dot{p}_2 \end{bmatrix} = \begin{bmatrix} v \\ v_1 \\ v_2 \end{bmatrix}$$

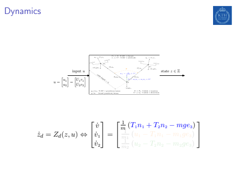


- kinematic equations

Dynamics

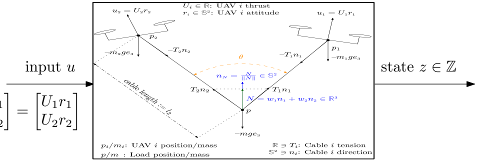


$$\dot{z}_d = Z_d(z, u) \Leftrightarrow \begin{bmatrix} \dot{v} \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(T_1 n_1 + T_2 n_2 - mge_3) \\ \frac{1}{m_1}(u_1 - T_1 n_1 - m_1 g e_3) \\ \frac{1}{m_2}(u_2 - T_2 n_2 - m_2 g e_3) \end{bmatrix}$$



- g stands for the acceleration due to gravity
- T_1, T_2 stand for the tensions on the cables
- tensions on the cables are functions of the state and the input
- Cable direction: $n_1 \equiv n_1(z) = \frac{p_1 - p}{l_1}$
- Input force: u_1
- Tension on cable: $T_1 = T_1(z, u_1, u_2) = \ldots$

Dynamics



$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U_1 r_1 \\ U_2 r_2 \end{bmatrix}$$

$$\dot{z}_d = Z_d(z, u) \Leftrightarrow \begin{bmatrix} \dot{v} \\ \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{m}(T_1 n_1 + T_2 n_2 - m g e_3) \\ \frac{1}{m_1}(u_1 - T_1 n_1 - m_1 g e_3) \\ \frac{1}{m_2}(u_2 - T_2 n_2 - m_2 g e_3) \end{bmatrix}$$

- g stands for the acceleration due to gravity
- T_1, T_2 stand for the tensions on the cables
- tensions on the cables are functions of the state and the input
- Cable direction: $n_1 \equiv n_1(z) = \frac{p_1 - p}{l_1}$
- Input force: u_1
- Tension on cable: $T_1 = T_1(z, u_1, u_2) = \dots$

State space

- Two constraints imposed by each cable
- Constraint 1: $C_1(z_k) = 0 \Leftrightarrow \|p_1 - (p + d_1 n)\| = l_1$
- Constraint 2: $C_2(z_k) = 0 \Leftrightarrow \|p_2 - (p + d_2 n)\| = l_2$

$$\mathbb{Z} := \left\{ z \in \mathbb{R}^{24} : \begin{cases} C_1(z_k) = 0 \text{ and } dC_1(z_k)\dot{z}_k = 0 \\ C_2(z_k) = 0 \text{ and } dC_2(z_k)\dot{z}_k = 0 \\ r_1, r_2 \in \mathbb{S}^2 \end{cases} \right\}$$

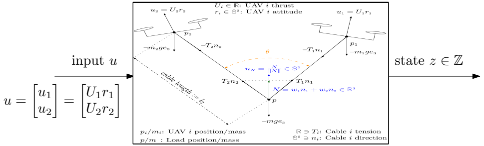
State space

- Two constraints imposed by each cable
- Constraint 1: $C_1(z_k) = 0 \Leftrightarrow \|p_1 - (p + d_1 n)\| = l_1$
- Constraint 2: $C_2(z_k) = 0 \Leftrightarrow \|p_2 - (p + d_2 n)\| = l_2$

$$\mathbb{Z} := \left\{ z \in \mathbb{R}^{24} : \begin{cases} C_1(z_k) = 0 \text{ and } dC_1(z_k)\dot{z}_k = 0 \\ C_2(z_k) = 0 \text{ and } dC_2(z_k)\dot{z}_k = 0 \\ r_1, r_2 \in \mathbb{S}^2 \end{cases} \right\}$$

- Cable imposes constraint: distance between end-points of the cable must be constant
- Cable also imposes constraints on the velocities
- Cable constraint is holonomic (it only depends on the positions)

Problem statement



Problem

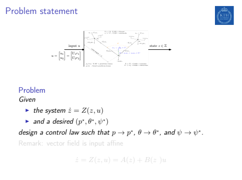
Given

- *the system* $\dot{z} = Z(z, u)$
- *and a desired* $(p^\star, \theta^\star, \psi^\star)$

design a control law such that $p \to p^\star$, $\theta \to \theta^\star$, *and* $\psi \to \psi^\star$.

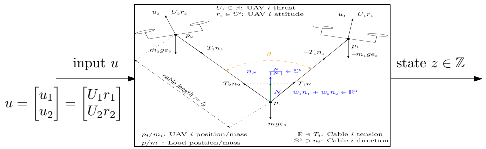
Remark: vector field is input affine

$$\dot{z} = Z(z, u) = A(z) + B(z)u$$



- System is underactuated
- Being input affine is helpfull

Problem statement



Problem

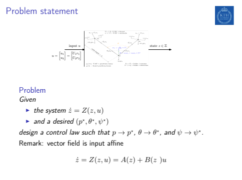
Given

- *the system* $\dot{z} = Z(z, u)$
- *and a desired* $(p^\star, \theta^\star, \psi^\star)$

design a control law such that $p \to p^\star$, $\theta \to \theta^\star$, *and* $\psi \to \psi^\star$.

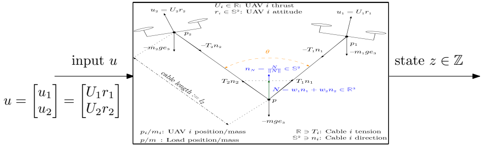
Remark: vector field is input affine

$$\dot{z} = Z(z, u) = A(z) + B(z\ )u$$



- System is underactuated
- Being input affine is helpfull

Problem statement



Problem

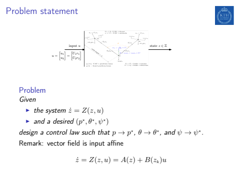
Given

- *the system* $\dot{z} = Z(z, u)$
- *and a desired* $(p^\star, \theta^\star, \psi^\star)$

design a control law such that $p \to p^\star$, $\theta \to \theta^\star$, *and* $\psi \to \psi^\star$.

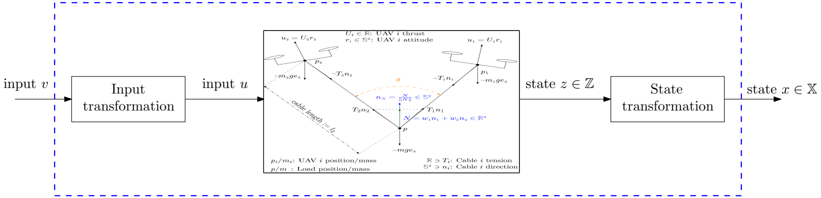
Remark: vector field is input affine

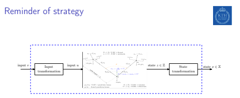
$$\dot{z} = Z(z, u) = A(z) + B(z_k)u$$



- System is underactuated
- Being input affine is helpfull

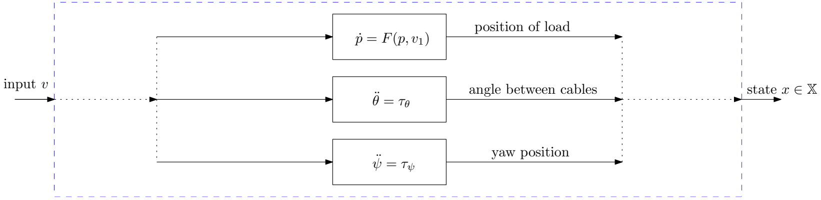
Reminder of strategy

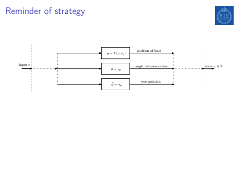




- vector field described before
- equilibrium described before
- equilibrium integral variables depend on model uncertainty
- (local) exponential stability is inferred from linearization

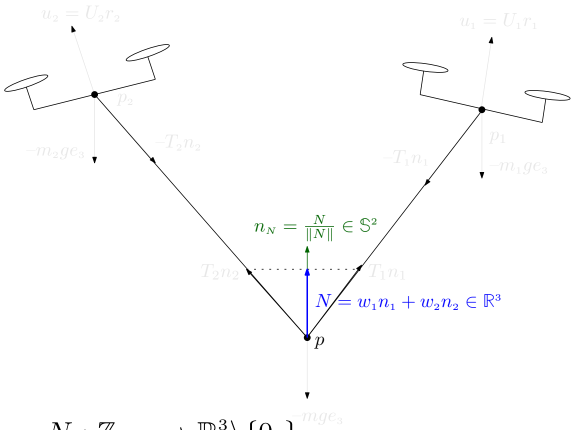
Reminder of strategy



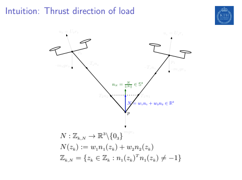


- vector field described before
- equilibrium described before
- equilibrium integral variables depend on model uncertainty
- (local) exponential stability is inferred from linearization

Intuition: Thrust direction of load

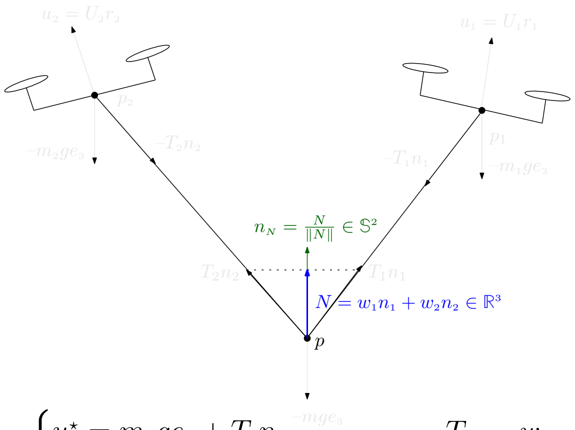


$$N : \mathbb{Z}_{k,N} \to \mathbb{R}^3 \backslash \{0_3\}$$
$$N(z_k) := w_1 n_1(z_k) + w_2 n_2(z_k)$$
$$\mathbb{Z}_{k,N} = \{z_k \in \mathbb{Z}_k : n_1(z_k)^T n_1(z_k) \neq -1\}$$



- Function N: direction of thrust by considering load as a thrust-propelled system
- Domain where N is defined is important
- Intuition for weights w_1 and w_2 by considering static case
- weights w_1 and w_2 determine weight distribution of load on each UAV

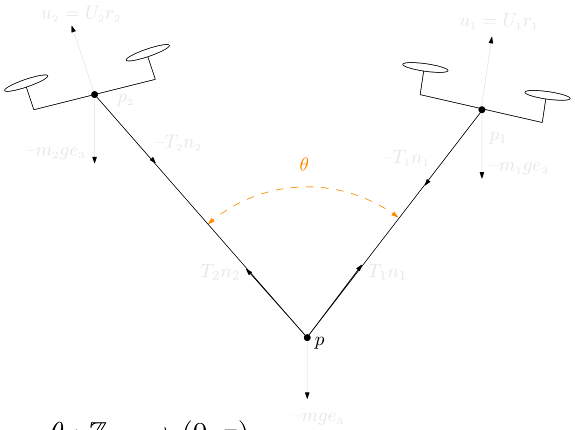
Intuition: Thrust direction of load



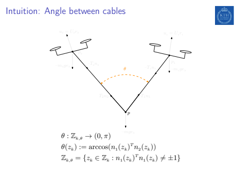
$$\begin{cases} u_1^\star = m_1 g e_3 + T_1 n_1 \\ u_2^\star = m_2 g e_3 + T_2 n_2 \end{cases} \quad \text{where} \quad \frac{T_1}{T_2} = \frac{w_1}{w_2}$$

- Function N: direction of thrust by considering load as a thrust-propelled system
- Domain where N is defined is important
- Intuition for weights w_1 and w_2 by considering static case
- weights w_1 and w_2 determine weight distribution of load on each UAV

Intuition: Angle between cables

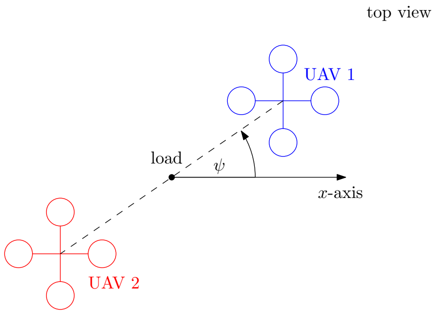


$$\theta : \mathbb{Z}_{k,\theta} \to (0, \pi)$$
$$\theta(z_k) := \arccos(n_1(z_k)^T n_2(z_k))$$
$$\mathbb{Z}_{k,\theta} = \{z_k \in \mathbb{Z}_k : n_1(z_k)^T n_1(z_k) \neq \pm 1\}$$



- Function θ, its domain and its definition

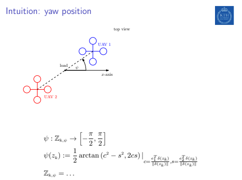
Intuition: yaw position



$$\psi : \mathbb{Z}_{k,\psi} \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

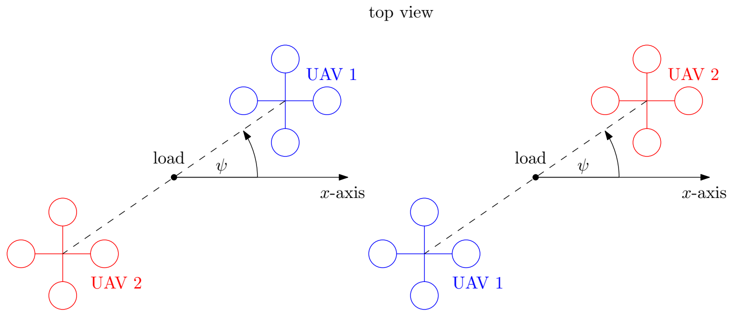
$$\psi(z_k) := \frac{1}{2}\arctan\left(c^2 - s^2, 2cs\right)\Big|_{c=\frac{e_1^T\delta(z_k)}{\|\delta(z_k)\|},\, s=\frac{e_2^T\delta(z_k)}{\|\delta(z_k)\|}}$$

$$\mathbb{Z}_{k,\psi} = \ldots$$



- Function ψ, its domain, and its definition
- $\delta(z_k) = \Pi(e_3)(n_1(z_k) - n_2(z_k))$

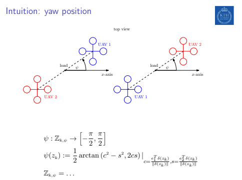
Intuition: yaw position



$$\psi : \mathbb{Z}_{k,\psi} \to \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$$

$$\psi(z_k) := \frac{1}{2} \arctan\left(c^2 - s^2, 2cs\right) \Big|_{c = \frac{e_1^T \delta(z_k)}{\|\delta(z_k)\|}, s = \frac{e_2^T \delta(z_k)}{\|\delta(z_k)\|}}$$

$$\mathbb{Z}_{k,\psi} = \ldots$$



- Function ψ, its domain, and its definition
- $\delta(z_k) = \Pi(e_3)(n_1(z_k) - n_2(z_k))$

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

- Change *physical inputs* to more *appropriate inputs*
- Computing tensions and angular accelerations is explained in the article
- Vector field is input affine: $\dot{z} = Z(z, u) = A(z) + B(z_k)u$
- Input transformation is a proper inverse of $v = A_R + B_R u$ provided that v lies in some space (which it will): angular acceleration of cable i needs to be orthogonal to cable i

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in\mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in\mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

- Change *physical inputs* to more *appropriate inputs*
- Computing tensions and angular accelerations is explained in the article
- Vector field is input affine: $\dot{z} = Z(z,u) = A(z) + B(z_k)u$
- Input transformation is a proper inverse of $v = A_R + B_R u$ provided that v lies in some space (which it will): angular acceleration of cable i needs to be orthogonal to cable i

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

Input transformation
What we wish to consider as inputs
$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$
Input transformation
$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

- Change *physical inputs* to more *appropriate inputs*
- Computing tensions and angular accelerations is explained in the article
- Vector field is input affine: $\dot{z} = Z(z, u) = A(z) + B(z_k)u$
- Input transformation is a proper inverse of $v = A_R + B_R u$ provided that v lies in some space (which it will): angular acceleration of cable i needs to be orthogonal to cable i

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = (B_R(z_k)^T B_R(z_k))^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

Input transformation

What we wish to consider as inputs

$$v = \begin{bmatrix} \text{Linear acceleration of load} \\ \text{Angular acceleration of } \frac{N}{\|N\|} \\ \text{Acceleration of } \theta \\ \text{Acceleration of } \psi \end{bmatrix} = \underbrace{A_R(z)}_{\in \mathbb{R}^8} + \underbrace{B_R(z_k)}_{\in \mathbb{R}^{8\times 6}} u$$

Input transformation

$$u = (B_R(z_k)^T B_R(z_k))^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

- Change *physical inputs* to more *appropriate inputs*
- Computing tensions and angular accelerations is explained in the article
- Vector field is input affine: $\dot{z} = Z(z,u) = A(z) + B(z_k)u$
- Input transformation is a proper inverse of $v = A_R + B_R u$ provided that v lies in some space (which it will): angular acceleration of cable i needs to be orthogonal to cable i

Input transformation

$$u = \left(B_R(z_k)^T B_R(z_k)\right)^{-1} B_R(z_k)^T \left(v - A_R(z_k)\right)$$

is equivalent to

$$\bar{u}(z,\nu) := \bar{u}_3(z,(T,\tau,\tau_\theta,\bar{\tau}_\psi))\Big|_{\bar{\tau}_\psi = \frac{\delta(z_k)^T\delta(z_k)\sqrt{1-(n_1(z_k)^T n_2(z_k))^2}}{\left(\frac{1}{w_1}+\frac{1}{w_2}\right)(e_1^T n_1(z_k+e_1^T n_2(z_k)(n_1(z_k)^T n_2(z_k)-1)}\left(\tau_\psi - \left(A_\psi(z)+B_\psi(z_k)\bar{u}_3(T,\tau,\tau_\theta,0)\right)\right)},\ \text{where}$$

$$\bar{u}_3(z,\nu) := \bar{u}_2(z,(T,\tau,\bar{\tau}_\theta,\tau_\psi))\Big|_{\bar{\tau}_\theta = \frac{w_1 w_2 (w_1+w_2 n_1(z_k)^T n_2(z_k))(w_2+w_1 n_1(z_k)^T n_2(z_k))}{\sqrt{N(z_k)^T N(z_k)}^3}(\tau_\theta - (A_\theta(z)+B_\theta(z_k)\bar{u}_2(T,\tau,0,0)))},\ \text{where}$$

$$\bar{u}_2(z,\nu) := \bar{u}_1(z,T) + \left[\begin{matrix} \frac{l_1 m_1}{w_1}\left\{\begin{matrix}\left(\frac{\tau_\theta}{n_1(z_k)^T n_N(z_k)} + \frac{N(z_k)^T N(z_k)}{(w_1+w_2)(1+n_1(z_k)^T n_2(z_k))}\bar{\tau}^T \frac{\mathcal{S}(n_N(z_k))n_1(z_k)}{\|\mathcal{S}(n_N(z_k))n_1(z_k)\|}\right)\frac{\Pi(n_1(z_k))n_N(z_k)}{\|\Pi(n_1(z_k))n_N(z_k)\|}+\cdots \\ \cdots+\left(\tau_\psi + \frac{\sqrt{N(z_k)^T N(z_k)}}{2}\bar{\tau}^T \frac{\Pi(n_N(z_k))n_1(z_k)}{\|\Pi(n_N(z_k))n_1(z_k)\|}\right)\frac{\mathcal{S}(n_1(z_k))n_N(z_k)}{\|\mathcal{S}(n_1(z_k))n_N(z_k)\|}\end{matrix}\right\} \\ \frac{l_2 m_2}{w_2}\left\{\begin{matrix}\left(\frac{\tau_\theta}{n_2(z_k)^T n_N(z_k)} + \frac{N(z_k)^T N(z_k)}{(w_1+w_2)(1+n_1(z_k)^T n_2(z_k))}\bar{\tau}^T \frac{\mathcal{S}(n_N(z_k))n_2(z_k)}{\|\mathcal{S}(n_N(z_k))n_2(z_k)\|}\right)\frac{\Pi(n_2(z_k))n_N(z_k)}{\|\Pi(n_2(z_k))n_N(z_k)\|}+\cdots \\ \cdots+\left(\tau_\psi + \frac{\sqrt{N(z_k)^T N(z_k)}}{2}\bar{\tau}^T \frac{\Pi(n_N(z_k))n_2(z_k)}{\|\Pi(n_N(z_k))n_2(z_k)\|}\right)\frac{\mathcal{S}(n_2(z_k))n_N(z_k)}{\|\mathcal{S}(n_2(z_k))n_N(z_k)\|}\end{matrix}\right\}\end{matrix}\right]\Bigg|_{\bar{\tau}=\tau-\left(A_{n_N}(z)+B_{n_N}(z_k)\bar{u}_2(T,\tau,0,0)\right)}$$

$$\bar{u}_1(z,T) := \begin{bmatrix}\bar{u}_1 n_1(z_k) \\ \bar{u}_2 n_2(z_k)\end{bmatrix}\Bigg|_{\begin{bmatrix}\bar{u}_1 \\ \bar{u}_2\end{bmatrix} = \begin{bmatrix}\frac{m_1}{m} & 0 \\ 0 & \frac{m_1}{m}\end{bmatrix} M_T(z_k)\left(\frac{mT}{\sqrt{N(z_k)^T N(z_k)}}\begin{bmatrix}w_1 \\ w_2\end{bmatrix} - T(z,0_6)\right)}.$$

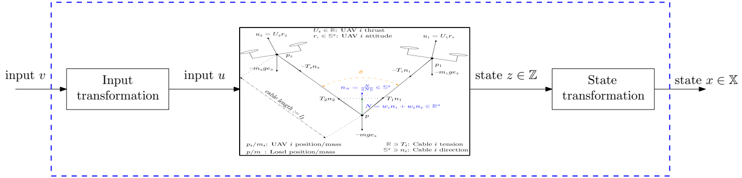
which is well defined in

$$Z_{k,\mathsf{inv}} = \left\{ z_k \in \mathbb{Z}_k : \begin{cases} e_3^T n_1(z_k) > 0 \\ e_3^T n_2(z_k) > 0 \\ n_1(z_k)^T n_2(z_k) > 0 \end{cases} \right\}$$

- Input transformation is well defined
- There is a cascade structure that can be used to compute the input transformation explicitly
- Actually

$$Z_{k,\mathsf{inv}} = \left\{ z_k \in \mathbb{Z}_k : \begin{cases} z_k \in Z_{k,N} \cap Z_{k,\theta} \cap Z_{k,\psi} \\ e_3^T n_1(z_k) > 0 \\ e_3^T n_2(z_k) > 0 \\ n_1(z_k)^T n_2(z_k) > 0 \end{cases} \right\}$$

Input Transformation

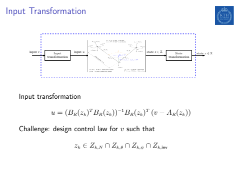


Input transformation

$$u = (B_R(z_k)^T B_R(z_k))^{-1} B_R(z_k)^T (v - A_R(z_k))$$

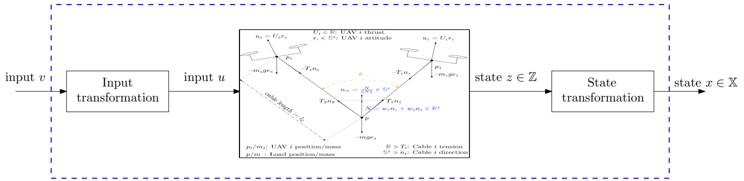
Challenge: design control law for v such that

$$z_k \in Z_{k,N} \cap Z_{k,\theta} \cap Z_{k,\psi} \cap Z_{k,\mathsf{inv}}$$

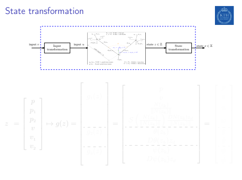


- move on to state transformation

State transformation

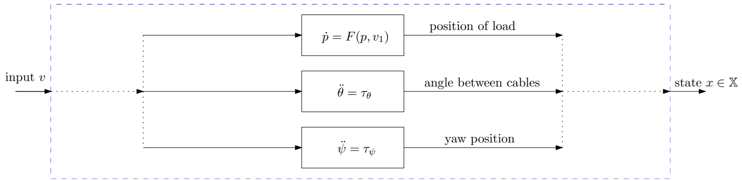


$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hline g_2(z) \\ \hline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ S\left(\frac{N(z_k)}{\|N(z_k)\|}\right)\frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hline \theta(z_k) \\ D\theta(z_k)z_d \\ \hline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} = \begin{bmatrix} \\ \hline \\ \hline \\ \end{bmatrix}$$

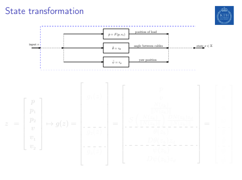


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

State transformation

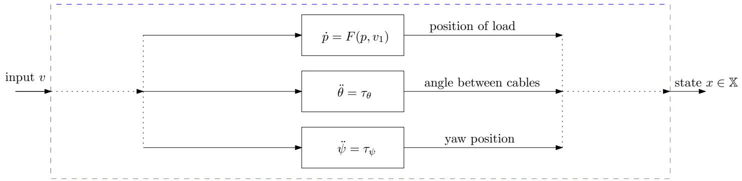


$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hline g_2(z) \\ \hline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ S\left(\frac{N(z_k)}{\|N(z_k)\|}\right)\frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hline \theta(z_k) \\ D\theta(z_k)z_d \\ \hline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} =$$

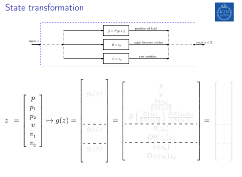


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

State transformation

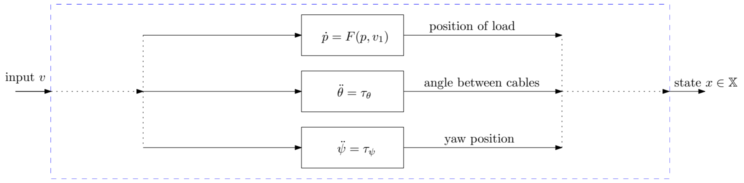


$$
z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hline g_2(z) \\ \hline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ \mathcal{S}\left(\frac{N(z_k)}{\|N(z_k)\|}\right) \frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hline \theta(z_k) \\ D\theta(z_k)z_d \\ \hline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} = \begin{bmatrix} p \\ v \\ n \\ \omega \\ \hline \theta \\ \dot{\theta} \\ \hline \psi \\ \dot{\psi} \end{bmatrix}
$$

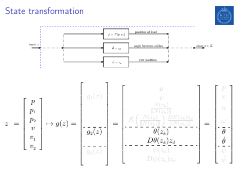


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

State transformation

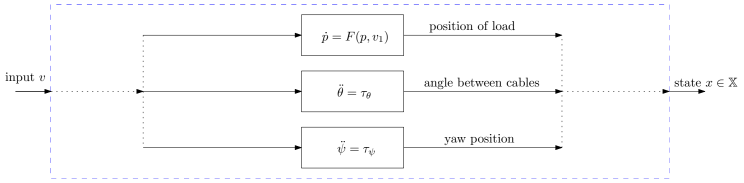


$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hdashline g_2(z) \\ \hdashline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ S\left(\frac{N(z_k)}{\|N(z_k)\|}\right)\frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hdashline \theta(z_k) \\ D\theta(z_k)z_d \\ \hdashline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} = \begin{bmatrix} p \\ v \\ n \\ \omega \\ \hdashline \theta \\ \dot{\theta} \\ \hdashline \psi \\ \dot{\psi} \end{bmatrix}$$

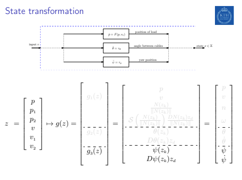


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

State transformation

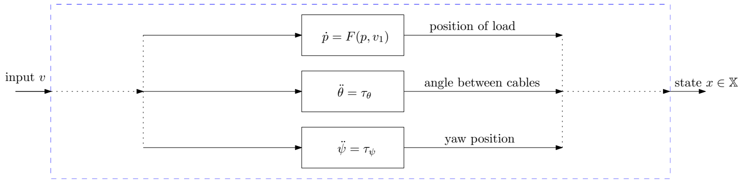


$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hdashline g_2(z) \\ \hdashline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ S\left(\frac{N(z_k)}{\|N(z_k)\|}\right)\frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hdashline \theta(z_k) \\ D\theta(z_k)z_d \\ \hdashline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} = \begin{bmatrix} p \\ v \\ n \\ \omega \\ \hdashline \theta \\ \dot\theta \\ \hdashline \psi \\ \dot\psi \end{bmatrix}$$

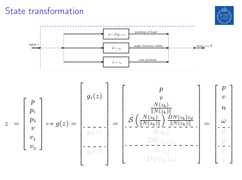


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

State transformation

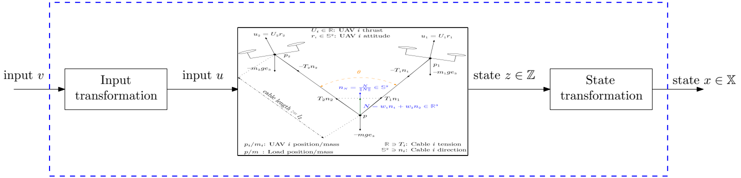


$$z = \begin{bmatrix} p \\ p_1 \\ p_2 \\ v \\ v_1 \\ v_2 \end{bmatrix} \mapsto g(z) = \begin{bmatrix} g_1(z) \\ \hdashline g_2(z) \\ \hdashline g_3(z) \end{bmatrix} = \begin{bmatrix} p \\ v \\ \frac{N(z_k)}{\|N(z_k)\|} \\ \mathcal{S}\left(\frac{N(z_k)}{\|N(z_k)\|}\right)\frac{DN(z_k)z_d}{\|N(z_k)\|} \\ \hdashline \theta(z_k) \\ D\theta(z_k)z_d \\ \hdashline \psi(z_k) \\ D\psi(z_k)z_d \end{bmatrix} = \begin{bmatrix} p \\ v \\ n \\ \omega \\ \hdashline \theta \\ \dot{\theta} \\ \hdashline \psi \\ \dot{\psi} \end{bmatrix}$$

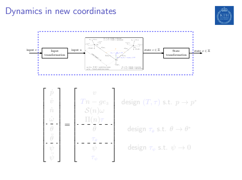


- g_1 isolates the position of the bar
- g_2 maps to the angle between the cables
- g_3 maps to the yaw position
- The inverse mapping g_3^{-1} would also be necessary

Dynamics in new coordinates

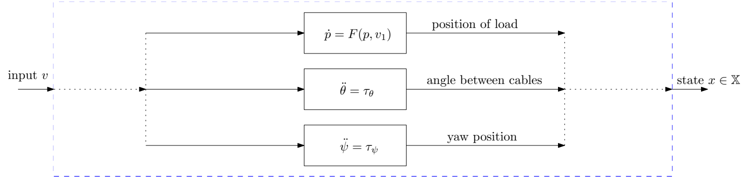


$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \\ \hline \dot{\theta} \\ \ddot{\theta} \\ \hline \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v \\ Tn - ge_3 \\ \mathcal{S}(n)\omega \\ \Pi(n)\tau \\ \hline \dot{\theta} \\ \tau_\theta \\ \hline \dot{\psi} \\ \tau_\psi \end{bmatrix} \quad \begin{matrix} \text{design } (T,\tau) \text{ s.t. } p \to p^\star \\ \text{design } \tau_\theta \text{ s.t. } \theta \to \theta^\star \\ \text{design } \tau_\psi \text{ s.t. } \psi \to 0 \end{matrix}$$

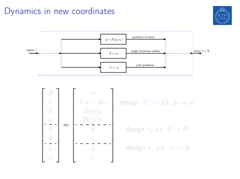


- transformed input $v = (T, \tau, \tau_\theta, \tau_\psi)$.
- The choice of the input and state transformations is now clear: it induces three decoupled vector fields.
- Two double integrators + the vector field of a thrust propelled system.
- Vector fields are decoupled, we can design control laws for them separately.
- Design T and τ such that $p \to p^\star$
- Design τ_θ such that $\theta \to \theta^\star$
- Design τ_ψ such that $\psi \to \psi^\star$

Dynamics in new coordinates

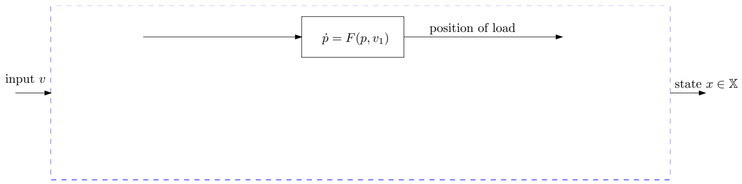


$$
\begin{bmatrix} \dot p \\ \dot v \\ \dot n \\ \dot\omega \\ \hdashline \dot\theta \\ \ddot\theta \\ \hdashline \dot\psi \\ \ddot\psi \end{bmatrix}
=
\begin{bmatrix} v \\ Tn - ge_3 \\ \mathcal{S}(n)\omega \\ \Pi(n)\tau \\ \hdashline \dot\theta \\ \tau_\theta \\ \hdashline \dot\psi \\ \tau_\psi \end{bmatrix}
\quad
\begin{array}{l}
\text{design } (T, \tau) \text{ s.t. } p \to p^\star \\[2em]
\text{design } \tau_\theta \text{ s.t. } \theta \to \theta^\star \\[1em]
\text{design } \tau_\psi \text{ s.t. } \psi \to 0
\end{array}
$$



- transformed input $v = (T, \tau, \tau_\theta, \tau_\psi)$.
- The choice of the input and state transformations is now clear: it induces three decoupled vector fields.
- Two double integrators + the vector field of a thrust propelled system.
- Vector fields are decoupled, we can design control laws for them separately.
- Design T and τ such that $p \to p^\star$
- Design τ_θ such that $\theta \to \theta^\star$
- Design τ_ψ such that $\psi \to \psi^\star$

Dynamics in new coordinates

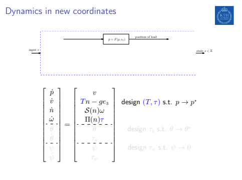


$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \\ \hline \dot{\theta} \\ \ddot{\theta} \\ \hline \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v \\ Tn - ge_3 \\ \mathcal{S}(n)\omega \\ \Pi(n)\tau \\ \hline \dot{\theta} \\ \tau_\theta \\ \hline \dot{\psi} \\ \tau_\psi \end{bmatrix}$$

design (T, τ) s.t. $p \to p^\star$

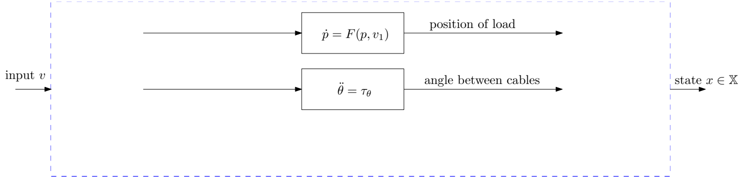
design τ_θ s.t. $\theta \to \theta^\star$

design τ_ψ s.t. $\psi \to 0$



- transformed input $v = (T, \tau, \tau_\theta, \tau_\psi)$.
- The choice of the input and state transformations is now clear: it induces three decoupled vector fields.
- Two double integrators + the vector field of a thrust propelled system.
- Vector fields are decoupled, we can design control laws for them separately.
- Design T and τ such that $p \to p^\star$
- Design τ_θ such that $\theta \to \theta^\star$
- Design τ_ψ such that $\psi \to \psi^\star$

Dynamics in new coordinates

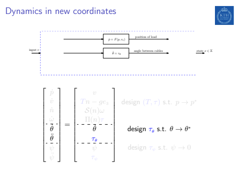


$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \\ \hline \dot{\theta} \\ \ddot{\theta} \\ \hline \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} v \\ Tn - ge_3 \\ \mathcal{S}(n)\omega \\ \Pi(n)\tau \\ \hline \dot{\theta} \\ \tau_\theta \\ \hline \dot{\psi} \\ \tau_\psi \end{bmatrix}$$

design (T, τ) s.t. $p \to p^\star$

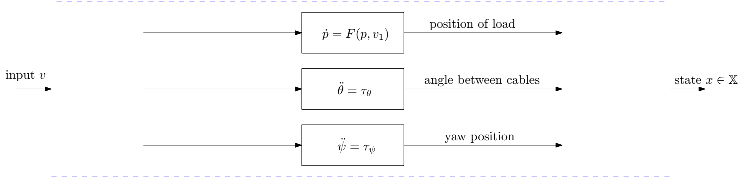
design τ_θ s.t. $\theta \to \theta^\star$

design τ_ψ s.t. $\psi \to 0$



- transformed input $v = (T, \tau, \tau_\theta, \tau_\psi)$.
- The choice of the input and state transformations is now clear: it induces three decoupled vector fields.
- Two double integrators + the vector field of a thrust propelled system.
- Vector fields are decoupled, we can design control laws for them separately.
- Design T and τ such that $p \to p^\star$
- Design τ_θ such that $\theta \to \theta^\star$
- Design τ_ψ such that $\psi \to \psi^\star$

Dynamics in new coordinates

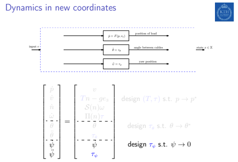


$$\begin{bmatrix} \dot p \\ \dot v \\ \dot n \\ \dot\omega \\ \hdashline \dot\theta \\ \ddot\theta \\ \hdashline \dot\psi \\ \ddot\psi \end{bmatrix} = \begin{bmatrix} v \\ Tn - ge_3 \\ S(n)\omega \\ \Pi(n)\tau \\ \hdashline \dot\theta \\ \tau_\theta \\ \hdashline \dot\psi \\ \tau_\psi \end{bmatrix}$$

design (T, τ) s.t. $p \to p^\star$

design τ_θ s.t. $\theta \to \theta^\star$

design τ_ψ s.t. $\psi \to 0$



- transformed input $v = (T, \tau, \tau_\theta, \tau_\psi)$.
- The choice of the input and state transformations is now clear: it induces three decoupled vector fields.
- Two double integrators + the vector field of a thrust propelled system.
- Vector fields are decoupled, we can design control laws for them separately.
- Design T and τ such that $p \to p^\star$
- Design τ_θ such that $\theta \to \theta^\star$
- Design τ_ψ such that $\psi \to \psi^\star$

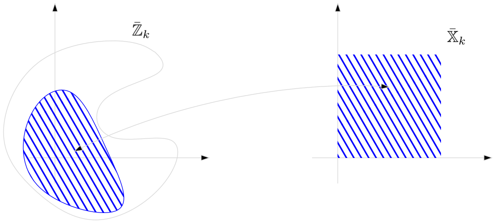
Control challenge

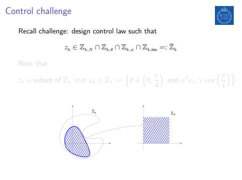
Recall challenge: design control law such that

$$z_k \in \mathbb{Z}_{k,N} \cap \mathbb{Z}_{k,\theta} \cap \mathbb{Z}_{k,\psi} \cap \mathbb{Z}_{k,\text{inv}} =: \bar{\mathbb{Z}}_k$$

Note that

$$z_k \in \text{subset of } \bar{\mathbb{Z}}_k \iff x_k \in \bar{\mathbb{X}}_k := \left\{\theta \in \left(0, \frac{\pi}{4}\right) \text{ and } n^T e_3 > \cos\left(\frac{\pi}{4}\right)\right\}$$





- We are going to design controllers in the new coordinates
- If we design controllers that guarantee that solution in new coordinates stays in $\bar{\mathbb{X}}_k$, then solution in original coordinates stays in $\bar{\mathbb{Z}}_k$.

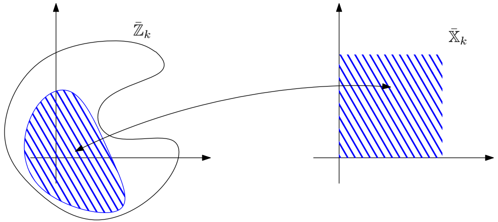
Control challenge

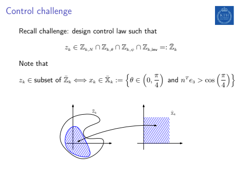
Recall challenge: design control law such that

$$z_k \in \mathbb{Z}_{k,N} \cap \mathbb{Z}_{k,\theta} \cap \mathbb{Z}_{k,\psi} \cap \mathbb{Z}_{k,\mathsf{inv}} =: \bar{\mathbb{Z}}_k$$

Note that

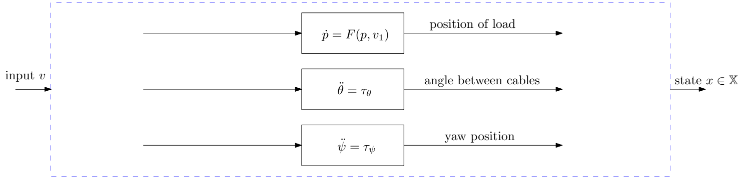
$$z_k \in \text{subset of } \bar{\mathbb{Z}}_k \iff x_k \in \bar{\mathbb{X}}_k := \left\{\theta \in \left(0, \frac{\pi}{4}\right) \text{ and } n^T e_3 > \cos\left(\frac{\pi}{4}\right)\right\}$$





- We are going to design controllers in the new coordinates
- If we design controllers that guarantee that solution in new coordinates stays in $\bar{\mathbb{X}}_k$, then solution in original coordinates stays in $\bar{\mathbb{Z}}_k$.

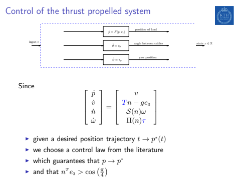
Control of the thrust propelled system



Since

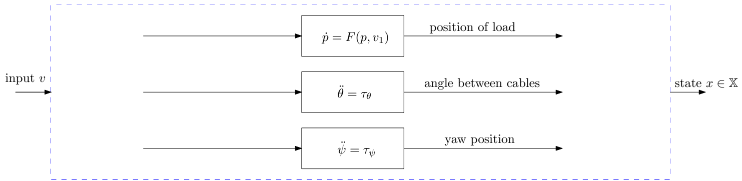
$$\begin{bmatrix} \dot p \\ \dot v \\ \dot n \\ \dot \omega \end{bmatrix} = \begin{bmatrix} v \\ {\color{blue}T} n - g e_3 \\ \mathcal{S}(n)\omega \\ \Pi(n){\color{blue}\tau} \end{bmatrix}$$

- given a desired position trajectory $t \to p^\star(t)$
- we choose a control law from the literature
- which guarantees that $p \to p^\star$
- and that $n^T e_3 > \cos\left(\frac{\pi}{4}\right)$



- thrust-propelled system: quadrotors literature

Control of angle between the cables



Since we want $\theta \in \left(0, \frac{\pi}{4}\right)$, we choose the control law

$$\tau_\theta^{cl} = -k_p \frac{\theta - \theta^\star}{\theta\left(\theta - \frac{\pi}{4}\right)} - k_d \dot{\theta}$$

- $\theta^\star \in \left(0, \frac{\pi}{4}\right)$ is some necessary angle between the cables
- $\theta > 0$: cables do not overlap
- $\theta < \frac{\pi}{4}$: cables not orthogonal to each other

Control of angle between the cables
Since we want $\theta \in \left(0, \frac{\pi}{4}\right)$, we choose the control law
$\theta^\star \in \left(0, \frac{\pi}{4}\right)$ is some necessary angle between the cables
$\theta > 0$: cables do not overlap
$\theta < \frac{\pi}{4}$: cables not orthogonal to each other

- $\theta > 0$: makes sure cables do not overlap
- $\theta < \frac{\pi}{4}$: makes cables cannot be orthogonal to each other (combined with $n^T e_3 > \cos\left(\frac{\pi}{4}\right)$)

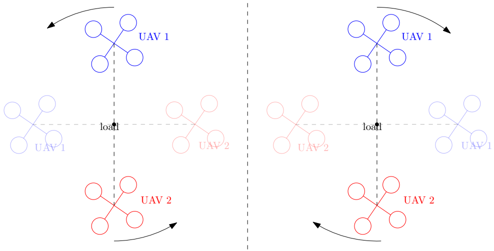
Control of yaw position

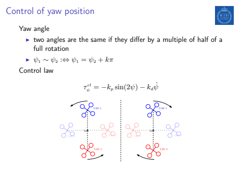
Yaw angle

- two angles are the same if they differ by a multiple of half of a full rotation
- $\psi_1 \sim \psi_2 :\Leftrightarrow \psi_1 = \psi_2 + k\pi$

Control law

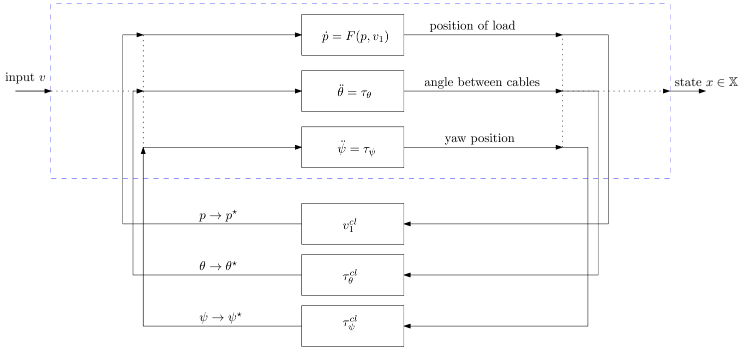
$$\tau_\psi^{cl} = -k_p \sin(2\psi) - k_d \dot{\psi}$$





- Factor of 2 in $\sin(2\psi)$ because two angles are the same if they differ by a multiple of half of a full rotation

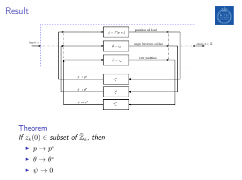
Result



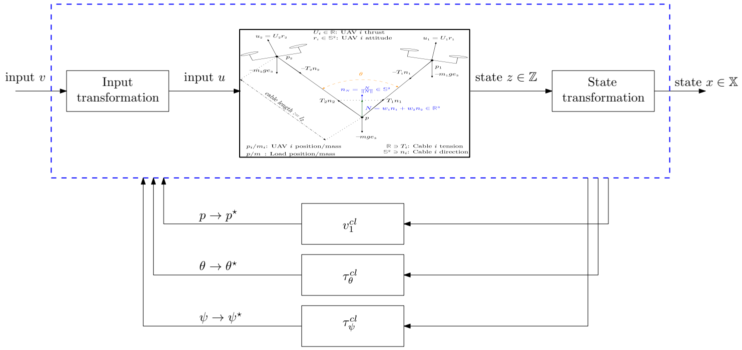
Theorem

If $z_k(0) \in$ subset of $\bar{\mathbb{Z}}_k$, then

- $p \to p^\star$
- $\theta \to \theta^\star$
- $\psi \to 0$



Result



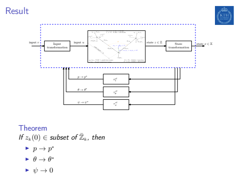
Theorem

If $z_k(0) \in$ *subset of* $\bar{\mathbb{Z}}_k$*, then*

- $p \to p^\star$
- $\theta \to \theta^\star$
- $\psi \to 0$

2017-12-11

Slung Load Transportation
└Main Result
└Result



Non-fully actuated UAVs

input u $\longrightarrow$ $\dot{z} = Z(z, u) \Leftrightarrow \begin{bmatrix} \dot{z}_k \\ \dot{z}_d \end{bmatrix} = \begin{bmatrix} Z_k(z) \\ Z_d(z, u) \end{bmatrix}$ $\longrightarrow$ state $z \in \mathbb{Z}$

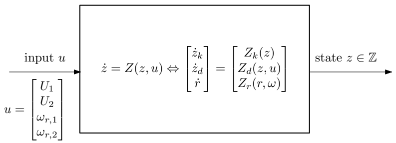
$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} U_1 r_1 \\ U_2 r_2 \end{bmatrix}$

$$\begin{bmatrix} \dot{z} \\ \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} = \begin{bmatrix} Z\left(z, \begin{bmatrix} u_1 - u_1 + U_1 r_1 \\ u_2 - u_2 + U_2 r_2 \end{bmatrix}\right) \\ \mathcal{S}(\omega_{r_1}) r_1 \\ \mathcal{S}(\omega_{r_2}) r_2 \end{bmatrix}$$

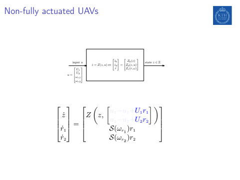
Non-fully actuated UAVs

- Recall that Z is input affine

Non-fully actuated UAVs

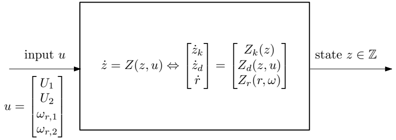


$$\begin{bmatrix} \dot{z} \\ \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} = \begin{bmatrix} Z\left(z, \begin{bmatrix} u_1 - u_1 + U_1 r_1 \\ u_2 - u_2 + U_2 r_2 \end{bmatrix}\right) \\ \mathcal{S}(\omega_{r_1}) r_1 \\ \mathcal{S}(\omega_{r_2}) r_2 \end{bmatrix}$$

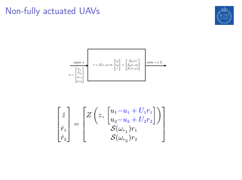


- Recall that Z is input affine

Non-fully actuated UAVs



$$\begin{bmatrix} \dot{z} \\ \dot{r}_1 \\ \dot{r}_2 \end{bmatrix} = \begin{bmatrix} Z\left(z, \begin{bmatrix} u_1 - u_1 + U_1 r_1 \\ u_2 - u_2 + U_2 r_2 \end{bmatrix}\right) \\ \mathcal{S}(\omega_{r_1}) r_1 \\ \mathcal{S}(\omega_{r_2}) r_2 \end{bmatrix}$$



- Recall that Z is input affine

Non-fully actuated UAVs

$$
\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{r}_1 \\ \dot{r}_2 \end{bmatrix}
=
\underbrace{\begin{bmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \dot{\theta} \\ \tau_\theta \\ \dot{\psi} \\ \tau_\psi \\ 0 \\ 0 \end{bmatrix}}_{\text{same as before}}
+
\begin{bmatrix} 0 \\ \sum B_{T,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\tau,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\theta,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\psi,i}(U_i r_i - u_i) \\ S(\omega_{r_1}) r_1 \\ S(\omega_{r_2}) r_2 \end{bmatrix}
$$

Option 1:

- Minimize the error $\|U_i r_i - u_i\| \Rightarrow U_i = r_i^T u_i$
- Not a good option because it breaks cascaded structure of thrust propelled system

- Recall that Z is input affine
- acceleration components come corrupted
- For option 1, backstepping cannot be use because cascaded structure does not exist
- We cannot use literature based on backstepping for control of thrust propelled systems

Non-fully actuated UAVs

$$
\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{n} \\ \dot{\omega} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \\ \dot{r}_1 \\ \dot{r}_2 \end{bmatrix}
=
\underbrace{\begin{bmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \dot{\theta} \\ \tau_\theta \\ \dot{\psi} \\ \tau_\psi \\ 0 \\ 0 \end{bmatrix}}_{\text{same as before}}
+
\begin{bmatrix} 0 \\ \sum B_{T,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\tau,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\theta,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\psi,i}(U_i r_i - u_i) \\ \mathcal{S}(\omega_{r_1}) r_1 \\ \mathcal{S}(\omega_{r_2}) r_2 \end{bmatrix}
$$

Option 1:

- Minimize the error $\|U_i r_i - u_i\| \Rightarrow U_i = r_i^T u_i$
- Not a good option because it breaks cascaded structure of thrust propelled system

- Recall that Z is input affine
- acceleration components come corrupted
- For option 1, backstepping cannot be use because cascaded structure does not exist
- We cannot use literature based on backstepping for control of thrust propelled systems

Non-fully actuated UAVs

$$
\begin{bmatrix} \dot p \\ \dot v \\ \dot n \\ \dot\omega \\ \dot\theta \\ \ddot\theta \\ \dot\psi \\ \ddot\psi \\ \dot r_1 \\ \dot r_2 \end{bmatrix}
=
\underbrace{\begin{bmatrix} \cdots \\ \cdots \\ \cdots \\ \cdots \\ \dot\theta \\ \tau_\theta \\ \dot\psi \\ \tau_\psi \\ 0 \\ 0 \end{bmatrix}}_{\text{same as before}}
+
\begin{bmatrix} 0 \\ \sum B_{T,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\tau,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\theta,i}(U_i r_i - u_i) \\ 0 \\ \sum B_{\psi,i}(U_i r_i - u_i) \\ \mathcal{S}(\omega_{r_1}) r_1 \\ \mathcal{S}(\omega_{r_2}) r_2 \end{bmatrix}
$$

Option 1:

- Minimize the error $\|U_i r_i - u_i\| \Rightarrow U_i = r_i^T u_i$
- Not a good option because it breaks cascaded structure of thrust propelled system

Non-fully actuated UAVs
same as before
Option 1:
Minimize the error $\|U_i r_i - u_i\| \Rightarrow U_i = r_i^T u_i$
Not a good option because it breaks cascaded structure of thrust propelled system

- Recall that Z is input affine
- acceleration components come corrupted
- For option 1, backstepping cannot be use because cascaded structure does not exist
- We cannot use literature based on backstepping for control of thrust propelled systems

Non-fully actuated UAVs

$$\begin{bmatrix}\dot p\\ \dot v\\ \dot n\\ \dot\omega\\ \dot n_1\\ \dot n_2\\ \dot\omega_1\\ \dot\omega_2\\ \dot r_1\\ \dot r_2\end{bmatrix} = \underbrace{\begin{bmatrix}\cdots\\ \cdots\\ \cdots\\ \cdots\\ \dot\theta\\ \tau_\theta\\ \dot\psi\\ \tau_\psi\\ 0\\ 0\end{bmatrix}}_{\text{same as before}} + \begin{bmatrix}0\\ \cancel{\sum B_{T,i}(U_i r_i - u_i)}\\ 0\\ \sum B_{\tau,i}(U_i r_i - u_i)\\ 0\\ \sum B_{\theta,i}(U_i r_i - u_i)\\ 0\\ \sum B_{\psi,i}(U_i r_i - u_i)\\ \mathcal{S}(\omega_{r_1})r_1\\ \mathcal{S}(\omega_{r_2})r_2\end{bmatrix}$$

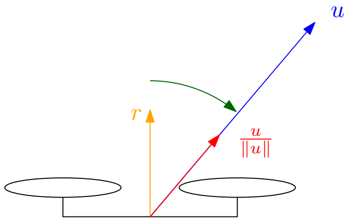
Option 2:

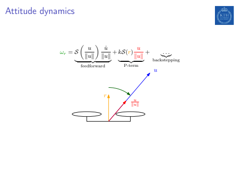
- Preserve cascaded structure: $U_i = \frac{n_i^T u_i}{n_i^T r_i}$
- Design UAVs angular velocities via backstepping step
- $r_i \to \frac{u_i}{\|u_i\|}$

- Recall that Z is input affine
- $\omega_{r_1}, \omega_{r_1}$: UAVs angular velocities
- $U_i = \frac{n_i^T u_i}{n_i^T r_i}$ implies that tensions in cables are the same as before

Attitude dynamics

$$\omega_r = \underbrace{\mathcal{S}\left(\frac{u}{\|u\|}\right)\frac{\dot{u}}{\|u\|}}_{\text{feedforward}} + \underbrace{k\mathcal{S}(r)\frac{u}{\|u\|}}_{\text{P-term}} + \underbrace{\cdots}_{\text{backstepping}}$$

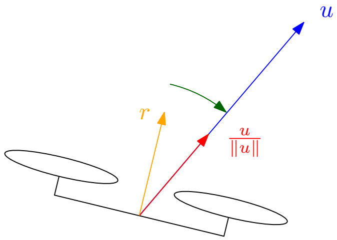


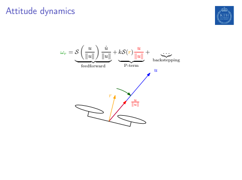


- Attitude inner loop: makes the UAV track the desired attitude
- Desired attitude is given by the $\frac{u}{\|u\|}$

Attitude dynamics

$$\omega_r = \underbrace{\mathcal{S}\left(\frac{u}{\|u\|}\right)\frac{\dot{u}}{\|u\|}}_{\text{feedforward}} + \underbrace{k\mathcal{S}(r)\frac{u}{\|u\|}}_{\text{P-term}} + \underbrace{\cdots}_{\text{backstepping}}$$

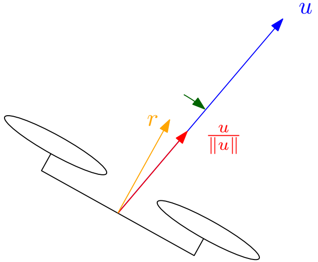




- Attitude inner loop: makes the UAV track the desired attitude
- Desired attitude is given by the $\frac{u}{\|u\|}$

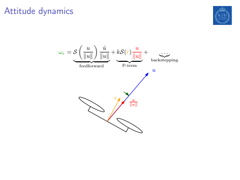
Attitude dynamics

$$\omega_r = \underbrace{\mathcal{S}\left(\frac{u}{\|u\|}\right)\frac{\dot{u}}{\|u\|}}_{\text{feedforward}} + \underbrace{k\mathcal{S}(r)\frac{u}{\|u\|}}_{\text{P-term}} + \underbrace{\cdots}_{\text{backstepping}}$$



2017-12-11

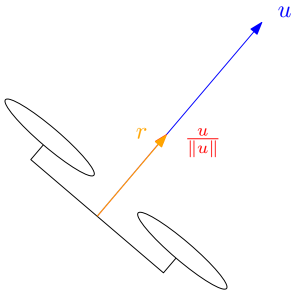
Slung Load Transportation
└─Non-fully actuated UAVs
└─Attitude dynamics

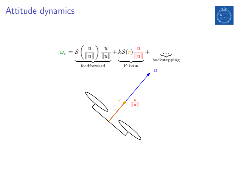


- Attitude inner loop: makes the UAV track the desired attitude
- Desired attitude is given by the $\frac{u}{\|u\|}$

Attitude dynamics

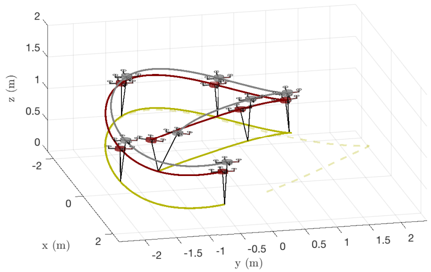
$$\omega_r = \underbrace{\mathcal{S}\left(\frac{u}{\|u\|}\right)\frac{\dot{u}}{\|u\|}}_{\text{feedforward}} + \underbrace{k\mathcal{S}(r)\frac{u}{\|u\|}}_{\text{P-term}} + \underbrace{\cdots}_{\text{backstepping}}$$

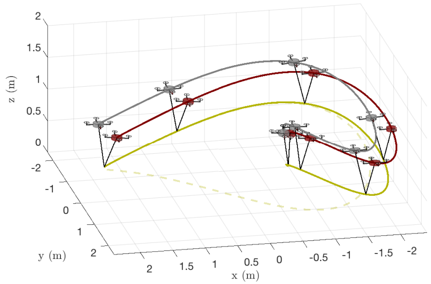


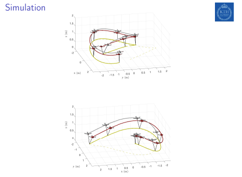


- Attitude inner loop: makes the UAV track the desired attitude
- Desired attitude is given by the $\frac{u}{\|u\|}$

Simulation

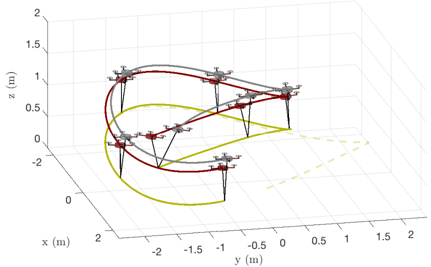


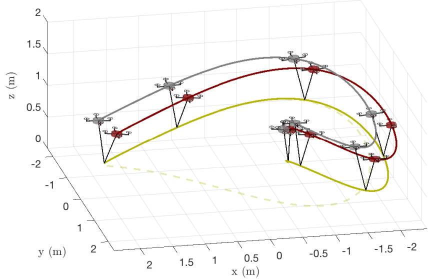


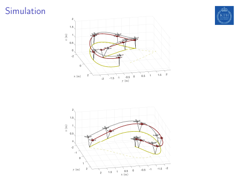


- Desired pose trajectory: circle motion in horizontal plane + oscillatory vertical motion
- Two perspectives of the same simulation
- Physical parameters listed in the article
- $\theta^\star = 30°$
- $\frac{w_1}{w_2} = \frac{2}{3}$

Simulation

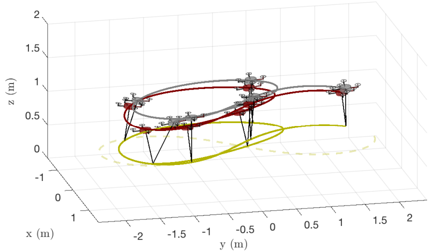


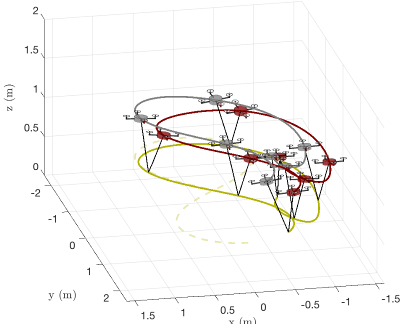


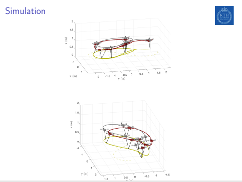


- Desired pose trajectory: circle motion in horizontal plane + oscillatory vertical motion
- Two perspectives of the same simulation
- Physical parameters listed in the article
- $\theta^\star = 30°$
- Transparent colors: “desired system”. Full colors: real system

Simulation

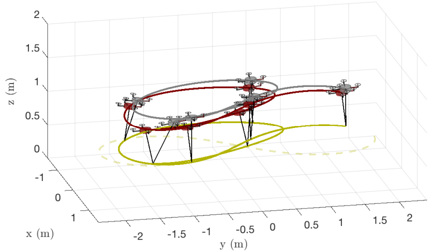


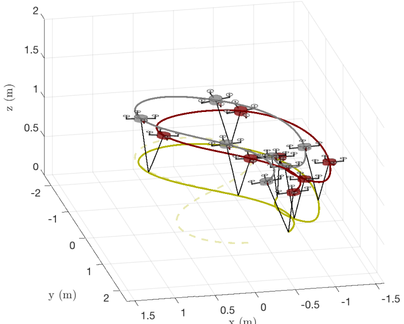


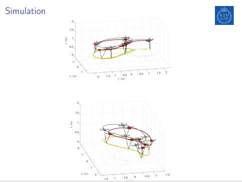


- Desired pose trajectory: eight-like motion in horizontal plane, at height of 0.5m
- Two perspectives of the same simulation
- Physical parameters listed in the article
- $\theta^\star = 30°$
- $\frac{w_1}{w_2} = \frac{2}{3}$

Simulation

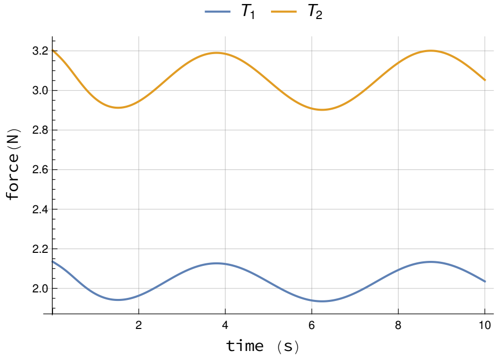




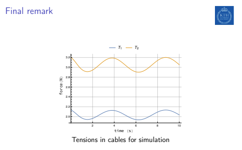


- Desired pose trajectory: eight-like motion in horizontal plane, at height of 0.5m
- Two perspectives of the same simulation
- Physical parameters listed in the article
- $\theta^\star = 30^\circ$
- $\frac{w_1}{w_2} = \frac{2}{3}$
- Transparent colors: "desired system". Full colors: real system

Final remark



Tensions in cables for simulation



- Tensions on cables are positive: cables need to remain under tension (taut), otherwise model is not valid
- $\frac{w_1}{w_2} = \frac{2}{3}$, in static equilibrium $\frac{T_1}{T_2} = \frac{w_1}{w_2}$
- Vehicle 2 is carrying more of the weight of the load, following the choice of weights w_1, w_2 we have made.

Thank you! Questions?

Thank you! Questions?