Decoupled Design of Controllers for Aerial Manipulation with Quadrotors

Pedro O. Pereira and Riccardo Zanella and Dimos V. Dimarogonas

Department of Automatic Control, KTH Royal Institute of Technology

October 13th
IROS 2016, Daejeon

Decoupled Design of Controllers for Aerial Manipulation with Quadrotors

Pedro O. Pereira and Riccardo Zanella and Dimos V. Dimarogonas

Department of Automatic Control, KTH Royal Institute of Technology

October 13th
IROS 2016, Daejeon

- IROS 2016, Thursday October 13th
- 14:50-15:05, Paper ThBT1.2
- (Special Session) Towards the Realization of the Aerial Robotic Workers
- Oral Presentation: 12 minutes presentation + 3 minutes discussion

Motivation

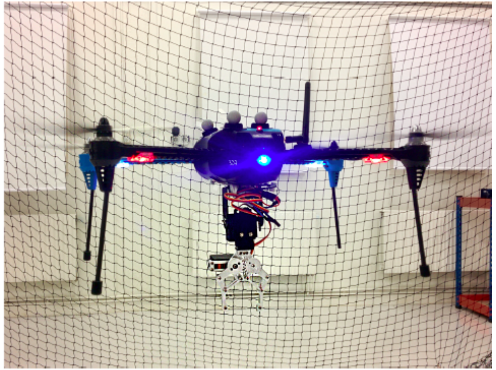
Automated fleet of aerial vehicles ARW's

- Reduce maintenance costs of infrastructure repairs
- Access to remote/hazardous locations

Pedro, IROS'16

- Deploy an automated fleet of vehicles: (collaborative) Aerial Robotic Workers (ARW's)
- Perform inspection and maintenance tasks of aging infrastructures
- AEROWORK also focuses on collaboration between ARW's, coverage tasks for inspection, high-level planning of complex tasks, design of manipulator appendices
- Aerial manipulation: one of the big tasks AEROWORKS wants to contribute to

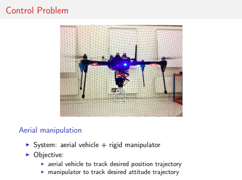
Control Problem



Aerial manipulation

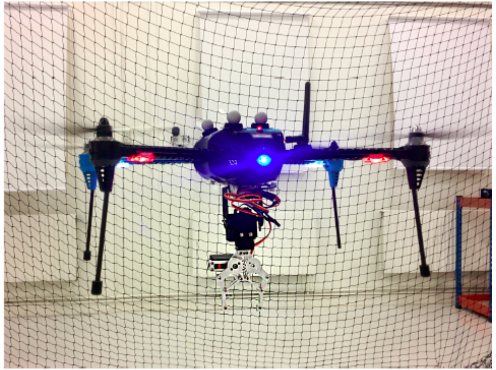
- System: aerial vehicle + rigid manipulator
- Objective:
  - aerial vehicle to track desired position trajectory
  - manipulator to track desired attitude trajectory

Pedro, IROS'16



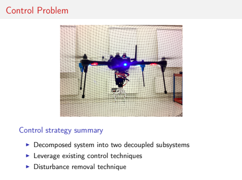
- Control for a system composed of a rigid manipulator and an aerial vehicle
- Disturbance removal: consider a constant thrust input disturbance (we design a disturbance estimator, and penalizes the position tracking error and the manipulation tracking error differently)
- Aerial vehicle: IRIS+; rigid manipulator :aluminum arm with a 1DOF revolute joint and a clamp gripper

Control Problem



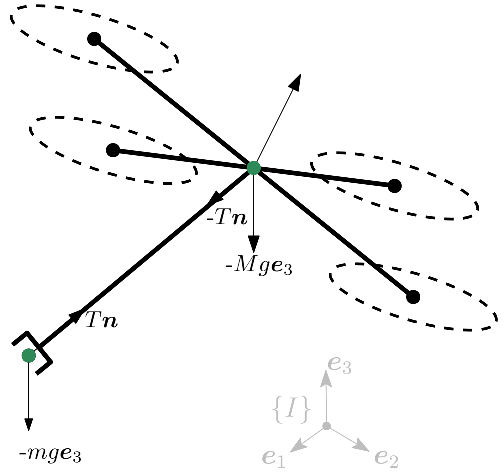
Control strategy summary

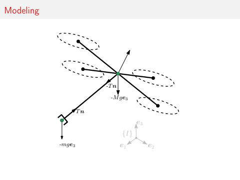
- Decomposed system into two decoupled subsystems
- Leverage existing control techniques
- Disturbance removal technique



- 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

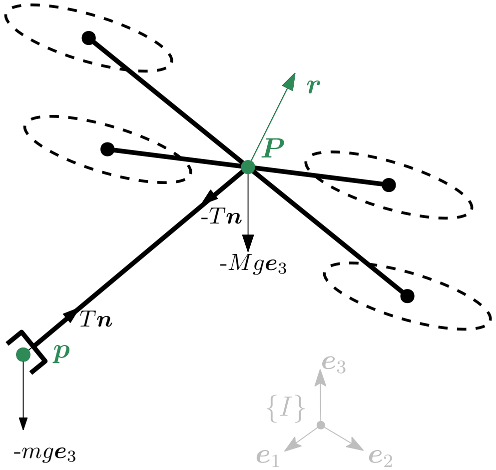
Modeling



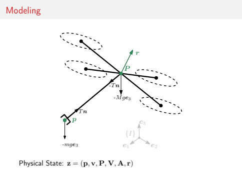


- UAV and rigid link connect by ball joint
- Load at the end effector

Modeling

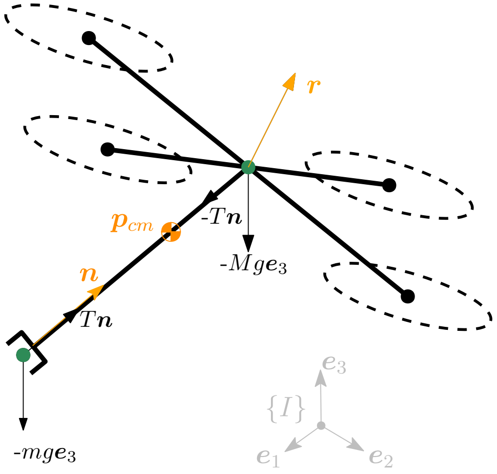


Physical State: $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{A}, \mathbf{r})$

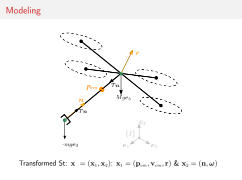


- Physical state: position+velocity of end-effector and UAV; body direction of UAV along which thrust can be provided

Modeling

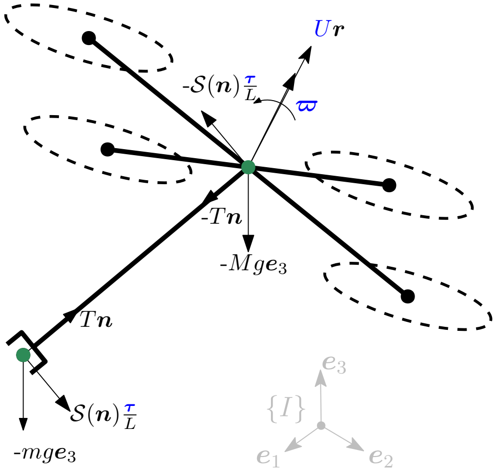


Transformed St: $\mathbf{x} \;= (\mathbf{x}_1, \mathbf{x}_2)$: $\mathbf{x}_1 = (\mathbf{p}_{cm}, \mathbf{v}_{cm}, \mathbf{r})$ & $\mathbf{x}_2 = (\mathbf{n}, \boldsymbol{\omega})$



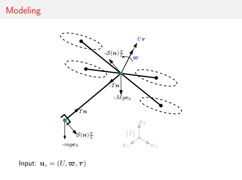
- Transformed state: position and velocity of center of mass and UAV body direction
- Transformed state: manipulator unit vector and its angular velocity

Modeling



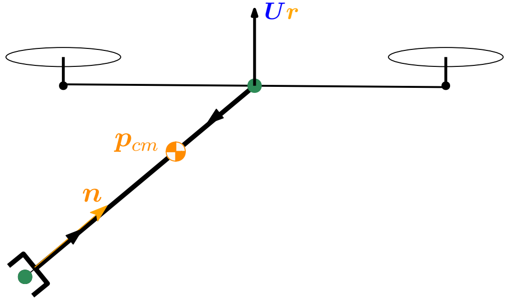
Input: $\mathbf{u}_z = (U, \boldsymbol{\varpi}, \boldsymbol{\tau})$

Pedro, IROS'16

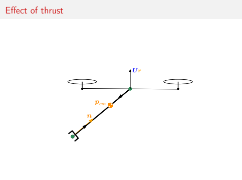


- Input is composed of thrust, angular velocity and torque on rigid link

Effect of thrust

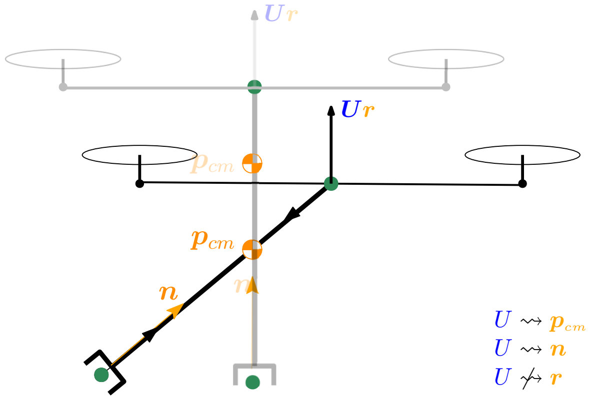


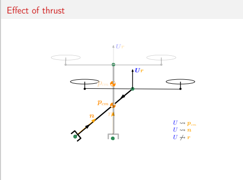
Pedro, IROS'16



- Effect of thrust: moves the whole systems upwards/downwards
- It also produces a torque on the rigid link, and makes it rotate
- The thrust does not affect the UAV body direction though

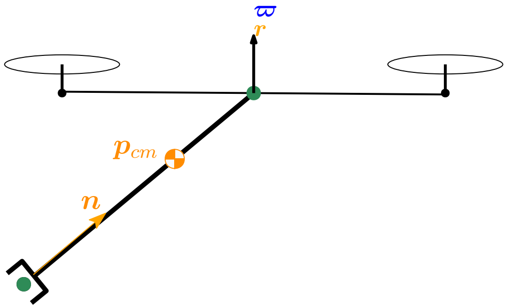
Effect of thrust



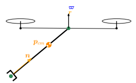


- Effect of thrust: moves the whole systems upwards/downwards
- It also produces a torque on the rigid link, and makes it rotate
- The thrust does not affect the UAV body direction though

Effect of angular velocity

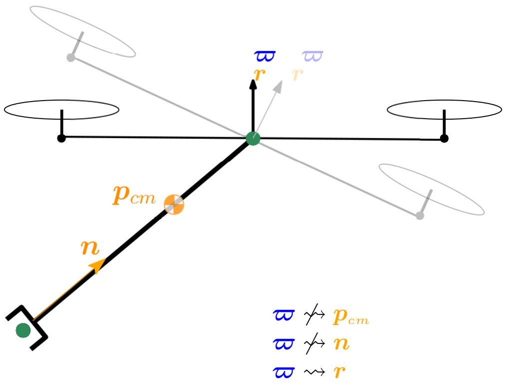


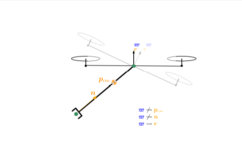
Effect of angular velocity



- Effect of angular velocity: only affects the UAV body direction

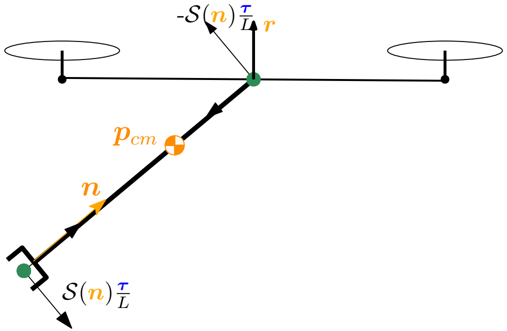
Effect of angular velocity



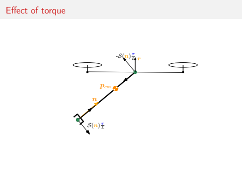


- Effect of angular velocity: only affects the UAV body direction

Effect of torque

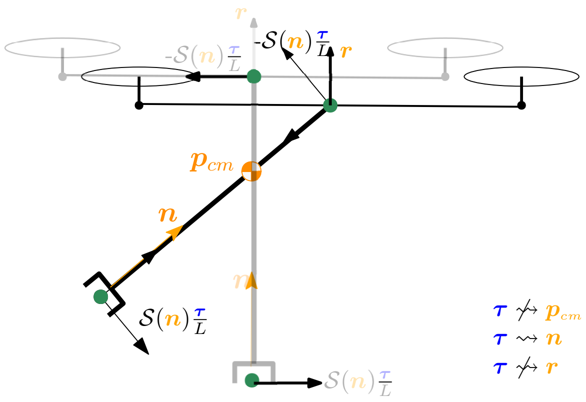


Pedro, IROS'16

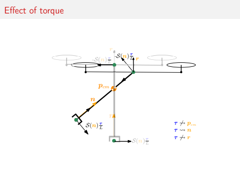


- Effect of torque: it only affects the rigid link, in particular it makes the rigid link rotate around the center of mass of the system
- The torque does not affect the UAV body direction, because the UAV and the rigid link are connected by a ball joint

Effect of torque

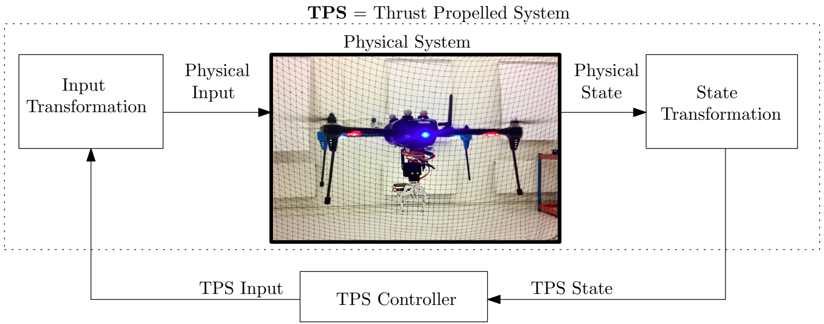


Pedro, IROS'16



- Effect of torque: it only affects the rigid link, in particular it makes the rigid link rotate around the center of mass of the system
- The torque does not affect the UAV body direction, because the UAV and the rigid link are connected by a ball joint

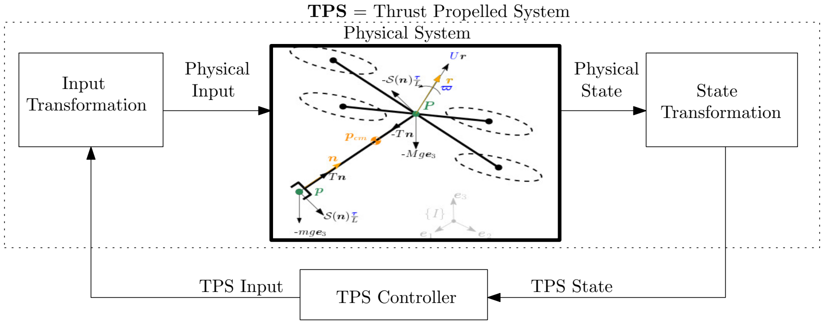
Control strategy



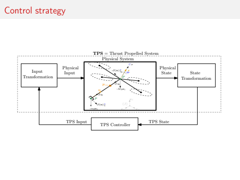
Pedro, IROS'16

- Control for 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
- Decompose the system into two decoupled subsystems
- The composition of the input and state transformation witht the system, leads to a vector for which we can find controllers in the literature

Control strategy



Pedro, IROS'16



- Simplified model needed for control design purposes
- The mappings (input and state transformation) are computed for the simplified model, described previously

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2, (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}$$

- $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

System dynamics
- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set
$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2, (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}$
- $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2, (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}$$

- $T_\mathbf{z}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\begin{aligned}\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 :\\ \mathbf{r}^T\mathbf{r} = 1,\\ (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2,\\ (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}\end{aligned}$$

- $T_\mathbf{z}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

System dynamics
- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set
$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2, (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}$
- $T_\mathbf{z}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \\ \mathbf{r}^T\mathbf{r} = 1, \\ (\mathbf{P} - \mathbf{p})^T(\mathbf{P} - \mathbf{p}) = L^2, \\ (\mathbf{V} - \mathbf{v})^T(\mathbf{P} - \mathbf{p}) = 0\}$$

- $T_\mathbf{z}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

System dynamics
- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set
$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P} - \mathbf{p})^T(\mathbf{P} - \mathbf{p}) = L^2, (\mathbf{V} - \mathbf{v})^T(\mathbf{P} - \mathbf{p}) = 0\}$
- $T_\mathbf{z}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\begin{aligned}\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 :& \\ \mathbf{r}^T\mathbf{r} &= 1, \\ (\mathbf{P} - \mathbf{p})^T(\mathbf{P} - \mathbf{p}) &= L^2, \\ (\mathbf{V} - \mathbf{v})^T(\mathbf{P} - \mathbf{p}) &= 0\}\end{aligned}$$

- $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

System dynamics
- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set
- $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System dynamics

- $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
- Ω_z: State set

$$\begin{aligned}\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 :& \\ \mathbf{r}^T\mathbf{r} = 1,& \\ (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2,& \\ (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}&\end{aligned}$$

- $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

Pedro, IROS'16

System dynamics
▸ $\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$: State
▸ Ω_z: State set
$\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1, (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L^2, (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}$
▸ $T_{\mathbf{z}}\Omega_z$: Tangent set to Ω_z at $\mathbf{z} \in \Omega_z$

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) constraints in velocities of UAV and end-effector: UAV and end-effector are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

Pedro, IROS'16

System Dynamics
(Open-loop) Vector field

- Three kinematic equations and two dynamic equations
- Acceleration of a rigid body is the net force applied on a rigid body, divided by the rigid body's mass
- End effect's acceleration; UAV's acceleration (its center of mass)
- There is an internal force when computing the accelerations, and this internal force is found by guaranteeing that the vector field belongs to the tangent set

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

System Dynamics
(Open-loop) Vector field

- Three kinematic equations and two dynamic equations
- Acceleration of a rigid body is the net force applied on a rigid body, divided by the rigid body's mass
- End effect's acceleration; UAV's acceleration (its center of mass)
- There is an internal force when computing the accelerations, and this internal force is found by guaranteeing that the vector field belongs to the tangent set

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\varpi)\mathbf{r} \end{bmatrix}$$

System Dynamics
(Open-loop) Vector field
$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\varpi)\mathbf{r} \end{bmatrix}$$

- Three kinematic equations and two dynamic equations
- Acceleration of a rigid body is the net force applied on a rigid body, divided by the rigid body's mass
- End effect's acceleration; UAV's acceleration (its center of mass)
- There is an internal force when computing the accelerations, and this internal force is found by guaranteeing that the vector field belongs to the tangent set

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

Vector field is tangent to set Ω_z, thus Ω_z is positively invariant

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \in T_{\mathbf{z}}\Omega_z$$

Pedro, IROS'16

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

Vector field is tangent to set Ω_z, thus Ω_z is positively invariant

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \in T_{\mathbf{z}}\Omega_z$$

- Three kinematic equations and two dynamic equations
- Acceleration of a rigid body is the net force applied on a rigid body, divided by the rigid body's mass
- End effect's acceleration; UAV's acceleration (its center of mass)
- There is an internal force when computing the accelerations, and this internal force is found by guaranteeing that the vector field belongs to the tangent set

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

Thrust input disturbance $\mathbf{e}_1 \in \mathbb{R}^7$

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$$

System Dynamics

(Open-loop) Vector field

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{v}} \\ \dot{\mathbf{P}} \\ \dot{\mathbf{V}} \\ \dot{\mathbf{r}} \end{bmatrix} = \mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := \begin{bmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{z}, \mathbf{u}_z) \\ \mathbf{V} \\ \mathbf{A}(\mathbf{z}, \mathbf{u}_z) \\ \mathcal{S}(\boldsymbol{\varpi})\mathbf{r} \end{bmatrix}$$

Thrust input disturbance $\mathbf{e}_1 \in \mathbb{R}^7$

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$$

- Three kinematic equations and two dynamic equations
- Acceleration of a rigid body is the net force applied on a rigid body, divided by the rigid body's mass
- End effect's acceleration; UAV's acceleration (its center of mass)
- There is an internal force when computing the accelerations, and this internal force is found by guaranteeing that the vector field belongs to the tangent set

Problem statement

Problem Statement

Given

- desired position trajectory $\mathbf{P}^\star : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^3$,
- desired orientation $\mathbf{n}^\star : \mathbb{R}_{\geq 0} \mapsto \mathbb{S}^2$,

design $\mathbf{u}_z = (U, \boldsymbol{\varpi}, \boldsymbol{\tau}) : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^7$, such that

- $\lim_{t\to\infty} (\mathbf{P}(t) - \mathbf{P}^\star(t)) = \mathbf{0}$
- $\lim_{t\to\infty} (\mathbf{n}(\mathbf{z}(t)) - \mathbf{n}^\star(t)) = \mathbf{0}$.

along the trajectory

- $\dot{\mathbf{z}}(t) = \mathbf{f}_z(\mathbf{z}(t), \mathbf{u}_z(t)), \mathbf{z}(0) \in \Omega_z$.

Problem statement
Problem Statement
Given
- desired position trajectory $\mathbf{P}^\star : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^3$,
- desired orientation $\mathbf{n}^\star : \mathbb{R}_{\geq 0} \mapsto \mathbb{S}^2$,
design $\mathbf{u}_z = (U, \boldsymbol{\varpi}, \boldsymbol{\tau}) : \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^7$, such that
- $\lim_{t\to\infty} (\mathbf{P}(t) - \mathbf{P}^\star(t)) = \mathbf{0}$
- $\lim_{t\to\infty} (\mathbf{n}(\mathbf{z}(t)) - \mathbf{n}^\star(t)) = \mathbf{0}$.
along the trajectory
- $\dot{\mathbf{z}}(t) = \mathbf{f}_z(\mathbf{z}(t), \mathbf{u}_z(t)), \mathbf{z}(0) \in \Omega_z$.

- We want the tracking to be accomplished in the presence of a thrust input disturbance as well.

Control Design

State Transformation

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{P} \\ \mathbf{V} \\ \mathbf{r} \end{bmatrix} \begin{matrix} g_z^x(t,\mathbf{z}) \\ \xrightarrow{} \\ \xleftarrow{} \\ g_x^z(t,\mathbf{x}) \end{matrix} \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{p}_{cm}(\mathbf{z}) - \mathbf{p}^\star_{cm}(t) \\ \mathbf{v}_{cm}(\mathbf{z}) - \dot{\mathbf{p}}^\star_{cm}(t) \\ \mathbf{r} \end{bmatrix}$$

$$\mathbf{x}_2 \approx \begin{bmatrix} \mathbf{n}(\mathbf{z}) - \mathbf{n}^\star(t) \\ \omega(\mathbf{z}) - \omega^\star(t) \end{bmatrix}$$

Pedro, IROS'16

Control Design

State Transformation

- We need to provide a state transformation
- The dynamics for the transformed state are in a form for which one can find controllers
- $\mathbf{x}_1$: position/velocity tracking error of center of mass
- $\mathbf{x}_2$: orientation tracking error of manipulator

Control Design

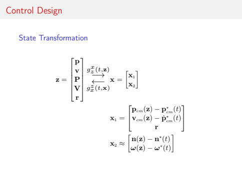
State Transformation

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \mathbf{v} \\ \mathbf{P} \\ \mathbf{V} \\ \mathbf{r} \end{bmatrix} \underset{g_x^z(t,\mathbf{x})}{\overset{g_z^x(t,\mathbf{z})}{\rightleftarrows}} \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}$$

$$\mathbf{x}_1 = \begin{bmatrix} \mathbf{p}_{cm}(\mathbf{z}) - \mathbf{p}^\star_{cm}(t) \\ \mathbf{v}_{cm}(\mathbf{z}) - \dot{\mathbf{p}}^\star_{cm}(t) \\ \mathbf{r} \end{bmatrix}$$

$$\mathbf{x}_2 \approx \begin{bmatrix} \mathbf{n}(\mathbf{z}) - \mathbf{n}^\star(t) \\ \boldsymbol{\omega}(\mathbf{z}) - \boldsymbol{\omega}^\star(t) \end{bmatrix}$$

Pedro, IROS'16



- We need to provide a state transformation
- The dynamics for the transformed state are in a form for which one can find controllers
- $\mathbf{x}_1$: position/velocity tracking error of center of mass
- $\mathbf{x}_2$: orientation tracking error of manipulator

Control Design

Input transformation

$$(\mathbf{u}_{x_1}, \mathbf{u}_{x_2}) \xrightarrow{\mathbf{u}_z^{cl}} (U, \boldsymbol{\varpi}, \boldsymbol{\tau})$$

$$((u, \boldsymbol{\varpi}), \boldsymbol{\alpha}) \xrightarrow{\mathbf{u}_z^{cl}} \left((M+m)u, \boldsymbol{\varpi}, L^2 \frac{Mm}{M+m} \mathcal{S}(\mathbf{n}) \left(\boldsymbol{\alpha} - \frac{(M+m)u}{LM} \mathbf{r}\right)\right)$$

Control Design

Input transformation

- We also need to provide an input transformation
- Recall that the thrust produces a torque on the rigid link, and, in the mapping, we are canceling this effect
- We are also "canceling" the effect of other constants (masses)

Input transformation

$$(\mathbf{u}_{x_1}, \mathbf{u}_{x_2}) \xrightarrow{\mathbf{u}_z^{cl}} (U, \boldsymbol{\varpi}, \boldsymbol{\tau})$$

$$((u, \boldsymbol{\varpi}), \boldsymbol{\alpha}) \xrightarrow{\mathbf{u}_z^{cl}} \left((M+m)u, \boldsymbol{\varpi}, L^2 \frac{Mm}{M+m} \mathcal{S}(\mathbf{n}) \left(\boldsymbol{\alpha} - \frac{(M+m)u}{LM} \mathbf{r}\right)\right)$$

Control Design
Input transformation
$$(\mathbf{u}_{x_1}, \mathbf{u}_{x_2}) \xrightarrow{\mathbf{u}_z^{cl}} (U, \boldsymbol{\varpi}, \boldsymbol{\tau})$$
$$((u, \boldsymbol{\varpi}), \boldsymbol{\alpha}) \xrightarrow{\mathbf{u}_z^{cl}} \left((M+m)u, \boldsymbol{\varpi}, L^2 \frac{Mm}{M+m} \mathcal{S}(\mathbf{n}) \left(\boldsymbol{\alpha} - \frac{(M+m)u}{LM} \mathbf{r}\right)\right)$$

- We also need to provide an input transformation
- Recall that the thrust produces a torque on the rigid link, and, in the mapping, we are canceling this effect
- We are also "canceling" the effect of other constants (masses)

Control Design

Vector field transformation

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \xrightarrow{g_z^x, \mathbf{u}_z^{cl}} \mathbf{f}_x(\mathbf{x}, \mathbf{u}_x) := \begin{bmatrix} \mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1}) \\ \mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2}) \end{bmatrix}$$

- Center-of-mass and manipulator dynamics are decoupled
- $\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_1) \leq 0$
- $\mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_2) \leq 0$

Control Design
Vector field transformation
$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \xrightarrow{g_z^x, \mathbf{u}_z^{cl}} \mathbf{f}_x(\mathbf{x}, \mathbf{u}_x) := \begin{bmatrix} \mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1}) \\ \mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2}) \end{bmatrix}$
Center-of-mass and manipulator dynamics are decoupled
$\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_1) \leq 0$
$\mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_2) \leq 0$

- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$
- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers and lyapunov functions
- They look like vector fields of thrust-propelled systems
- In the article, we propose specific ones, but other controllers would work
- They are only decoupled in the abcense of a disturbance

Control Design

Vector field transformation

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \xrightarrow{g_z^x, \mathbf{u}_z^{cl}} \mathbf{f}_x(\mathbf{x}, \mathbf{u}_x) := \begin{bmatrix} \mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1}) \\ \mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2}) \end{bmatrix}$$

- Center-of-mass and manipulator dynamics are decoupled
- $\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_1) \leq 0$
- $\mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_2) \leq 0$

Control Design

Vector field transformation

Center-of-mass and manipulator dynamics are decoupled

- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$
- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers and lyapunov functions
- They look like vector fields of thrust-propelled systems
- In the article, we propose specific ones, but other controllers would work
- They are only decoupled in the abcense of a disturbance

Control Design

Vector field transformation

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \xrightarrow{g_z^x, \mathbf{u}_z^{cl}} \mathbf{f}_x(\mathbf{x}, \mathbf{u}_x) := \begin{bmatrix} \mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1}) \\ \mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2}) \end{bmatrix}$$

- Center-of-mass and manipulator dynamics are decoupled
- $\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_1) \leq 0$
- $\mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_2) \leq 0$

Control Design
Vector field transformation
$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \xrightarrow{g_z^x, \mathbf{u}_z^{cl}} \mathbf{f}_x(\mathbf{x}, \mathbf{u}_x) := \begin{bmatrix} \mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1}) \\ \mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2}) \end{bmatrix}$
- Center-of-mass and manipulator dynamics are decoupled
- $\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_1) \leq 0$
- $\mathbf{f}_{x_2}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_2) \leq 0$

- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$
- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers and lyapunov functions
- They look like vector fields of thrust-propelled systems
- In the article, we propose specific ones, but other controllers would work
- They are only decoupled in the abcense of a disturbance

Disturbance removal

Remove effect of a constant thrust input disturbance $\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$

- Model battery drainage as thrust input disturbance
- Practice: works for low frequency disturbance

Disturbance estimate $\dot{\hat{b}} = f_{\hat{b}}(t, \tilde{\mathbf{x}})$

$$f_{\hat{b}}(t, \tilde{\mathbf{x}}) = \mathrm{Proj}\left(\Phi^T(\mathbf{x})\left(\underbrace{k_{\hat{b}_1}\partial_{\mathbf{x}_1} V_{x_1}(t, \mathbf{x}_1)}_{\text{penalty position tracking error}} + \underbrace{k_{\hat{b}_2}\partial_{\mathbf{x}_2} V_{x_2}(t, \mathbf{x}_2)}_{\text{penalty attitude tracking error}}\right), \hat{b}\right),$$

Cai, Queiroz, Dawson. A sufficiently smooth projection operator. TAC 2006

- Disturbance: in its presence, the errors are not steered to zero.
- We construct a disturbance estimate, and in particular, we construct how this disturbance must evolve (we design its vector field)
- Controller is not static (it has a dynamic component, the disturbance estimate)
- The two different errors penalize differently the disturbance estimate: one expects that if $k_{\hat{b}_1} \ggg k_{\hat{b}_2}$ we care more about the position tracking error; if $k_{\hat{b}_2} \ggg k_{\hat{b}_1}$ we care more about the attitude tracking error

Disturbance removal

Remove effect of a constant thrust input disturbance $\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$

- Model battery drainage as thrust input disturbance
- Practice: works for low frequency disturbance

Disturbance estimate $\dot{\hat{b}} = f_{\hat{b}}(t, \tilde{\mathbf{x}})$

$$f_{\hat{b}}(t, \tilde{\mathbf{x}}) = \mathrm{Proj}\left(\Phi^T(\mathbf{x})\left(\underbrace{k_{\hat{b}_1}\partial_{\mathbf{x}_1}V_{x_1}(t, \mathbf{x}_1)}_{\text{penalty position tracking error}} + \underbrace{k_{\hat{b}_2}\partial_{\mathbf{x}_2}V_{x_2}(t, \mathbf{x}_2)}_{\text{penalty attitude tracking error}}\right), \hat{b}\right),$$

Cai, Queiroz, Dawson. A sufficiently smooth projection operator. TAC 2006

Disturbance removal
Remove effect of a constant thrust input disturbance $\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$
Model battery drainage as thrust input disturbance
Practice: works for low frequency disturbance
Disturbance estimate $\dot{\hat{b}} = f_{\hat{b}}(t, \tilde{\mathbf{x}})$
Cai, Queiroz, Dawson. A sufficiently smooth projection operator. TAC 2006

- Disturbance: in its presence, the errors are not steered to zero.
- We construct a disturbance estimate, and in particular, we construct how this disturbance must evolve (we design its vector field)
- Controller is not static (it has a dynamic component, the disturbance estimate)
- The two different errors penalize differently the disturbance estimate: one expects that if $k_{\hat{b}_1} \ggg k_{\hat{b}_2}$ we care more about the position tracking error; if $k_{\hat{b}_2} \ggg k_{\hat{b}_1}$ we care more about the attitude tracking error

Disturbance removal

Remove effect of a constant thrust input disturbance $\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$

- Model battery drainage as thrust input disturbance
- Practice: works for low frequency disturbance

Disturbance estimate $\dot{\hat{b}} = f_{\hat{b}}(t, \tilde{\mathbf{x}})$

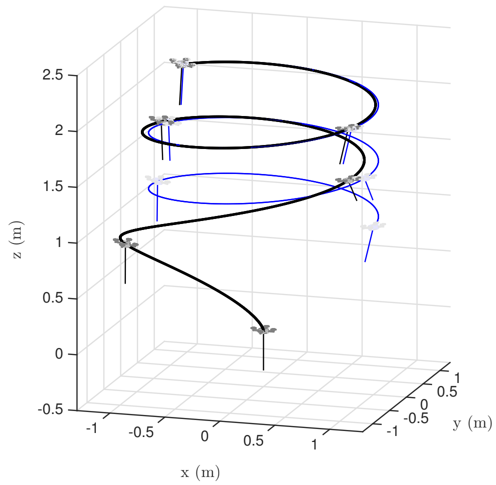
$$V(\mathbf{x}, \hat{b}) = k_{\hat{b}_1} V_{x_1}(t, \mathbf{x}_1) + k_{\hat{b}_2} V_{x_2}(t, \mathbf{x}_2) + \frac{(b - \hat{b})^2}{2}$$
$$\dot{V}(\mathbf{x}, \hat{b}) \leq 0$$

Pedro, IROS'16

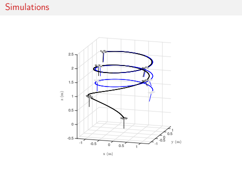
Disturbance removal
Remove effect of a constant thrust input disturbance $\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$
Model battery drainage as thrust input disturbance
Practice: works for low frequency disturbance
Disturbance estimate $\dot{\hat{b}} = f_{\hat{b}}(t, \tilde{\mathbf{x}})$
$V(\mathbf{x}, \hat{b}) = k_{\hat{b}_1} V_{x_1}(t, \mathbf{x}_1) + k_{\hat{b}_2} V_{x_2}(t, \mathbf{x}_2) + \frac{(b - \hat{b})^2}{2}$
$\dot{V}(\mathbf{x}, \hat{b}) \leq 0$

- Disturbance: in its presence, the errors are not steered to zero.
- The proof: construct a Lyapunov function by combining the "decoupled" Lyapunov functions, and the extra term related to the disturbance estimation error

Simulations

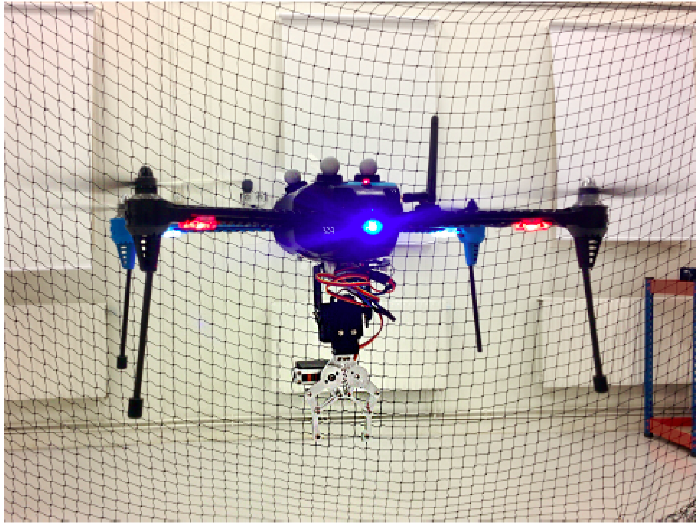


Pedro, IROS'16



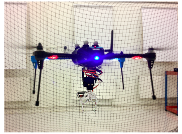
- Simulation video
- Desired motion: load describing an helix motion (0.1m/s up)
- Manipulator is supposed to make a circular-like motion

Experiments



Pedro, IROS'16

Experiments

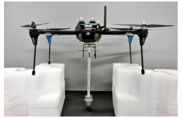


- Two experiments
- (1): Pick and place: aluminum arm with a 1DOF revolute joint and a clamp gripper. The weight is about 290g. Both the joint and the gripper are actuated by two servomotors MG995; length of the manipulator, from the joint to the extremity of the gripper, is 14cm. The PWM signals to control the two servomotors are provided by the output lines AUX OUT 1-2 of the autopilot Pixhawk. Using Mission Planner we can then link these outputs to the channels RC6-7 of the RC commands vector. Since Pixhawk does not provide power to the servos itself, these are powered by an external ESC, providing 5V, which can be connected to the servo outputs on the Pixhawk or to a servo directly.
- (2): Bar of aluminum rod of length 35cm, 350g most concentrated in extremity (shift of 7cm of the center of mass from uav's center of mass): compare controller ignoring its presence and controller that compensates for its presence (better performance when we consider its presence)

Experiments

Pedro, IROS'16

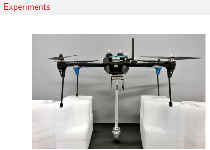
Experiments



- Two experiments
- (1): Pick and place: aluminum arm with a 1DOF revolute joint and a clamp gripper. The weight is about 290g. Both the joint and the gripper are actuated by two servomotors MG995; length of the manipulator, from the joint to the extremity of the gripper, is 14cm. The PWM signals to control the two servomotors are provided by the output lines AUX OUT 1-2 of the autopilot Pixhawk. Using Mission Planner we can then link these outputs to the channels RC6-7 of the RC commands vector. Since Pixhawk does not provide power to the servos itself, these are powered by an external ESC, providing 5V, which can be connected to the servo outputs on the Pixhawk or to a servo directly.
- (2): Bar of aluminum rod of length 35cm, 350g most concentrated in extremity (shift of 7cm of the center of mass from uav's center of mass): compare controller ignoring its presence and controller that compensates for its presence (better performance when we consider its presence)

Experiments

Pedro, IROS'16



- Two experiments
- (1): Pick and place: aluminum arm with a 1DOF revolute joint and a clamp gripper. The weight is about 290g. Both the joint and the gripper are actuated by two servomotors MG995; length of the manipulator, from the joint to the extremity of the gripper, is 14cm. The PWM signals to control the two servomotors are provided by the output lines AUX OUT 1-2 of the autopilot Pixhawk. Using Mission Planner we can then link these outputs to the channels RC6-7 of the RC commands vector. Since Pixhawk does not provide power to the servos itself, these are powered by an external ESC, providing 5V, which can be connected to the servo outputs on the Pixhawk or to a servo directly.
- (2): Bar of aluminum rod of length 35cm, 350g most concentrated in extremity (shift of 7cm of the center of mass from uav's center of mass): compare controller ignoring its presence and controller that compensates for its presence (better performance when we consider its presence)

Future work

Future work

- Extension to collaborative manipulation (multiple UAV's)
- Remove other types of disturbances
- Study robustness against model uncertainty

Future work
Future work
- Extension to collaborative manipulation (multiple UAV's)
- Remove other types of disturbances
- Study robustness against model uncertainty

- Proposed a contoller that decouples the system of manipulator+UAV into two subsystems
- We leveraged control strategies for thrust propelled systems
- Disturbance removal for constant thrust input disturbances

Thank you! Questions?

Thank you! Questions?

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

End effector's acceleration

$$\mathbf{a}_L(\mathbf{z}, \mathbf{u}_z) := \frac{T(\mathbf{z}, \mathbf{u}_z)}{m}\bar{\mathbf{n}}(\mathbf{z}) + \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\boldsymbol{\tau}}{mL} - g\mathbf{e}_3$$

System Dynamics
(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by
$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$
Manipulator's unit vector and angular velocity
$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$
End effector's acceleration
$$\mathbf{a}_L(\mathbf{z}, \mathbf{u}_z) := \frac{T(\mathbf{z}, \mathbf{u}_z)}{m}\bar{\mathbf{n}}(\mathbf{z}) + \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\boldsymbol{\tau}}{mL} - g\mathbf{e}_3$$

- Acceleration of a rigid is the net force applied on a rigid body, divided by the rigid body mass

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

UAV's acceleration

$$\mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z) := \frac{U}{M}\mathbf{r} - \frac{T(\mathbf{z}, \mathbf{u}_z)}{M}\bar{\mathbf{n}}(\mathbf{z}) - \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\boldsymbol{\tau}}{ML} - g\mathbf{e}_3$$

Pedro, IROS'16

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

UAV's acceleration

$$\mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z) := \frac{U}{M}\mathbf{r} - \frac{T(\mathbf{z}, \mathbf{u}_z)}{M}\bar{\mathbf{n}}(\mathbf{z}) - \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\boldsymbol{\tau}}{ML} - g\mathbf{e}_3$$

- Acceleration of a rigid is the net force applied on a rigid body, divided by the rigid body mass

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Tension/Compression on the manipulator

$$T(\mathbf{z}, \mathbf{u}_z) = \frac{m}{M + m}\left(U\mathbf{r}^T\bar{\mathbf{n}}(\mathbf{z}) + \frac{M}{L}\|\mathbf{V} - \mathbf{v}\|^2\right),$$

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Tension/Compression on the manipulator

$$T(\mathbf{z}, \mathbf{u}_z) = \frac{m}{M + m}\left(U\mathbf{r}^T\bar{\mathbf{n}}(\mathbf{z}) + \frac{M}{L}\|\mathbf{V} - \mathbf{v}\|^2\right),$$

- Acceleration of a rigid is the net force applied on a rigid body, divided by the rigid body mass

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Vector fields is tangent to set Ω_z, thus Ω_z is positively invariant

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \in T_{\mathbf{z}}\Omega_z$$

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Vector fields is tangent to set Ω_z, thus Ω_z is positively invariant

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) \in T_{\mathbf{z}}\Omega_z$$

- Acceleration of a rigid is the net force applied on a rigid body, divided by the rigid body mass

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Thrust input disturbance $\mathbf{e}_1 \in \mathbb{R}^7$

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$$

System Dynamics

(Open-loop) Vector field $\mathbf{f}_z : \Omega_z \times \mathbb{R}^7 \mapsto \mathbb{R}^{15}$ is given by

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z) := (\mathbf{v}, \mathbf{a}_L(\mathbf{z}, \mathbf{u}_z), \mathbf{V}, \mathbf{a}_Q(\mathbf{z}, \mathbf{u}_z), \mathcal{S}(\boldsymbol{\varpi})\mathbf{r})$$

Manipulator's unit vector and angular velocity

$$\bar{\mathbf{n}}(\mathbf{z}) = \frac{\mathbf{P} - \mathbf{p}}{\|\mathbf{P} - \mathbf{p}\|}, \bar{\boldsymbol{\omega}}(\mathbf{z}) := \mathcal{S}(\bar{\mathbf{n}}(\mathbf{z}))\frac{\mathbf{V} - \mathbf{v}}{L}$$

Thrust input disturbance $\mathbf{e}_1 \in \mathbb{R}^7$

$$\mathbf{f}_z(\mathbf{z}, \mathbf{u}_z + b\mathbf{e}_1)$$

- Acceleration of a rigid is the net force applied on a rigid body, divided by the rigid body mass

Maps

$$\mathbf{g}_z^x(\mathbf{z}) := (\mathbf{g}_z^{x1}(\mathbf{z}), \mathbf{g}_z^{x2}(\mathbf{z}))$$
$$\mathbf{g}_z^{x1}(\mathbf{z}) := (\mathbf{p}_{cm}(\mathbf{z}), \mathbf{v}_{cm}(\mathbf{z}), \mathbf{r})$$
$$\mathbf{g}_z^{x2}(\mathbf{z}) := (\bar{\mathbf{n}}(\mathbf{z}), \bar{\boldsymbol{\omega}}(\mathbf{z}))$$

Control law

$$\mathbf{u}_z^{cl}(\mathbf{x}, \mathbf{u}_x) = \left((M+m)u, \boldsymbol{\varpi}, L^2 \frac{Mm}{M+m} \mathcal{S}(\mathbf{n}) \left(\boldsymbol{\alpha} - \frac{(M+m)u}{LM} \mathbf{r} \right) \right)$$

System with state $\mathbf{x}_1 = (\mathbf{p}_{cm}, \mathbf{v}_{cm}, \mathbf{r})$ and input $\mathbf{u}_{x_1} = (u, \boldsymbol{\varpi})$

$$\mathbf{f}_{x_1}(\mathbf{x}_1, \mathbf{u}_{x_1}) = (\mathbf{v}, T\mathbf{r} - (g\mathbf{e}_3 + \mathbf{p}_{cm}^\star(t)), \mathbf{r}),$$

System with state $\mathbf{x}_2 = (\mathbf{n}, \boldsymbol{\omega})$ and input $\mathbf{u}_{x_2} = \boldsymbol{\alpha}$

$$\mathbf{f}_{x_1}(\mathbf{x}_2, \mathbf{u}_{x_2}) = (\mathcal{S}(\boldsymbol{\omega})\mathbf{n}, \Pi(\mathbf{n})\boldsymbol{\alpha}).$$

- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers
- They look like vector fields of thrust-propelled systems

Control Design Idea

Find

- map $\mathbf{g}_z^x(t,\cdot):\Omega_z \mapsto \Omega_x$
- control law $\mathbf{u}_z^{cl}:\Omega_x \times \mathbb{R}^7 \mapsto \mathbb{R}^7$

such that

$$\mathbf{f}_x(t,\mathbf{x}) = \left(\partial_t \mathbf{g}_z^x(t,\mathbf{z}) + \partial_\mathbf{z} \mathbf{g}_z^x(t,\mathbf{z})\mathbf{f}_z(\mathbf{z},\mathbf{u}_z^{cl}(\mathbf{x},\mathbf{u}_x))\right)\big|_{\mathbf{z}=\mathbf{g}_x^z(t,\mathbf{x})}$$
$$\mathbf{f}_x(t,\mathbf{x}) = (\mathbf{f}_{x_1}(t,\mathbf{x}_1,\mathbf{u}_{x_1}),\mathbf{f}_{x_2}(t,\mathbf{x}_1,\mathbf{u}_{x_1}))$$

- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers and lyapunov functions
- They look like vector fields of thrust-propelled systems
- In the article, we propose specific ones, but other controllers would work
- They are only decoupled in the abcense of a disturbance

Control Design Idea

Find

- map $\mathbf{g}_z^x(t,\cdot) : \Omega_z \mapsto \Omega_x$
- control law $\mathbf{u}_z^{cl} : \Omega_x \times \mathbb{R}^7 \mapsto \mathbb{R}^7$

$$\mathbf{g}_z^x(\mathbf{z}) := (\mathbf{g}_z^{x1}(\mathbf{z}), \mathbf{g}_z^{x2}(\mathbf{z}))$$
$$\mathbf{g}_z^{x1}(\mathbf{z}) := (\mathbf{p}_{cm}(\mathbf{z}), \mathbf{v}_{cm}(\mathbf{z}), \mathbf{r})$$
$$\mathbf{g}_z^{x2}(\mathbf{z}) := (\bar{\mathbf{n}}(\mathbf{z}), \bar{\boldsymbol{\omega}}(\mathbf{z}))$$

$\mathbf{f}_{x_1}(t, \mathbf{x}_1, \mathbf{u}_{x_1})$: we find $\mathbf{u}_{x_1}^{cl}, V_{x_1}$ such that $\dot{V}_{x_1}(t, \mathbf{x}_{x_1}) \leq 0$

$\mathbf{f}_{x_1}(t, \mathbf{x}_2, \mathbf{u}_{x_2})$: we find $\mathbf{u}_{x_2}^{cl}, V_{x_2}$ such that $\dot{V}_{x_2}(t, \mathbf{x}_{x_2}) \leq 0$

Control Design Idea
Find
map $\mathbf{g}_z^x(t,\cdot) : \Omega_z \mapsto \Omega_x$
control law $\mathbf{u}_z^{cl} : \Omega_x \times \mathbb{R}^7 \mapsto \mathbb{R}^7$

- $\mathbf{f}_{x_1}$ and $\mathbf{f}_{x_2}$ in form for which we can already find controllers and lyapunov functions
- They look like vector fields of thrust-propelled systems
- In the article, we propose specific ones, but other controllers would work
- They are only decoupled in the abcense of a disturbance

System dynamics

System's State and Input

$$\mathbf{z} = (\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in \Omega_z$$
$$\mathbf{u}_z = (T, \boldsymbol{\varpi}, \boldsymbol{\tau}) \in \mathbb{R} \times \mathbb{R}^3 \times \mathbb{R}^3$$

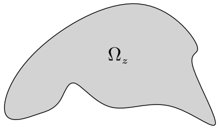
System's State Set

$$\begin{aligned}\Omega_z := \{(\mathbf{p}, \mathbf{v}, \mathbf{P}, \mathbf{V}, \mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\mathbf{r} = 1,\\ (\mathbf{P}-\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = L, \}\\ (\mathbf{V}-\mathbf{v})^T(\mathbf{P}-\mathbf{p}) = 0\}\\ T_{\mathbf{z}}\Omega_z := \{(\delta\mathbf{p}, \delta\mathbf{v}, \delta\mathbf{P}, \delta\mathbf{V}, \delta\mathbf{r}) \in (\mathbb{R}^3)^5 : \mathbf{r}^T\delta\mathbf{r} = 0,\\ (\delta\mathbf{P}-\delta\mathbf{p})^T(\mathbf{P}-\mathbf{p}) = 0,\\ (\delta\mathbf{V}-\delta\mathbf{v})^T(\mathbf{P}-\mathbf{p}) + (\mathbf{V}-\mathbf{v})^T(\delta\mathbf{P}-\delta\mathbf{p}) = 0\}\end{aligned}$$

Pedro, IROS'16

- System state: end-effector position and velocity; UAV's position, velocity and attitude
- Input to system: UAV thrust and angular velocity, and torque on manipulator
- State set:
  - (1) Unit vector for attitude;
  - (2) distance between end-effector and uav is the same as the manipulator's length L
  - (3) end-effector and uav are not moving away (in distance) from each other
- Tangent space to State set at a point $\mathbf{z} \in \Omega_z$: compute it with differentials

State Transformation



$\mathbf{z} \in \Omega_z$

Figure: State transformation

- System dynamics w.r.t state $\mathbf{z} \in \Omega_z$
- Coordinate transformation to state $\mathbf{x} \in \Omega_x$
- Decoupled dynamics (undisturbed case)
  - $\mathbf{f}_{x_1}$ independent of $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ independent of $\mathbf{x}_1$

State Transformation

Ω_z

$\mathbf{z} \in \Omega_z$

Figure: State transformation

- System dynamics w.r.t state $\mathbf{z} \in \Omega_z$
- Coordinate transformation to state $\mathbf{x} \in \Omega_x$
- Decoupled dynamics (undisturbed case)
  - $\mathbf{f}_{x_1}$ independent of $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ independent of $\mathbf{x}_1$

- We write the dynamics for a state in Ω_z
- For the controller design, we consider a different state, and state set Ω_x
- We provide a mapping from $\mathbf{z}$ to $\mathbf{x}$ and vice-versa (this is a mapping for each time instant)
- The dynamics for the transformed state are in a form for which one can find controllers
- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$

State Transformation

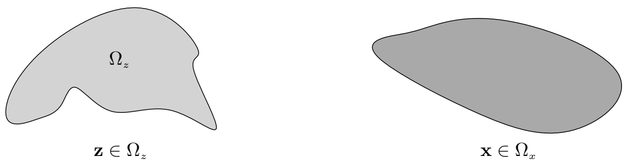
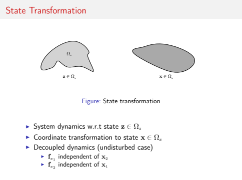


Figure: State transformation

- System dynamics w.r.t state $\mathbf{z} \in \Omega_z$
- Coordinate transformation to state $\mathbf{x} \in \Omega_x$
- Decoupled dynamics (undisturbed case)
  - $\mathbf{f}_{x_1}$ independent of $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ independent of $\mathbf{x}_1$



- We write the dynamics for a state in Ω_z
- For the controller design, we consider a different state, and state set Ω_x
- We provide a mapping from $\mathbf{z}$ to $\mathbf{x}$ and vice-versa (this is a mapping for each time instant)
- The dynamics for the transformed state are in a form for which one can find controllers
- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$

State Transformation

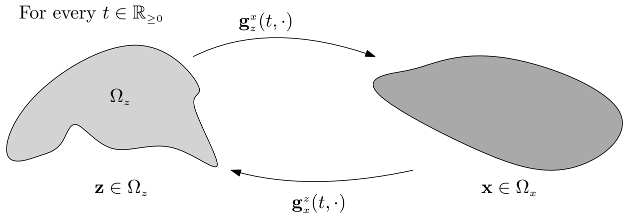
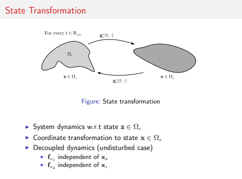


Figure: State transformation

- System dynamics w.r.t state $\mathbf{z} \in \Omega_z$
- Coordinate transformation to state $\mathbf{x} \in \Omega_x$
- Decoupled dynamics (undisturbed case)
  - $\mathbf{f}_{x_1}$ independent of $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ independent of $\mathbf{x}_1$

Pedro, IROS'16



- We write the dynamics for a state in Ω_z
- For the controller design, we consider a different state, and state set Ω_x
- We provide a mapping from $\mathbf{z}$ to $\mathbf{x}$ and vice-versa (this is a mapping for each time instant)
- The dynamics for the transformed state are in a form for which one can find controllers
- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$

State Transformation

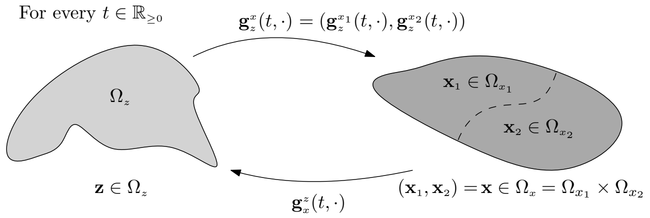
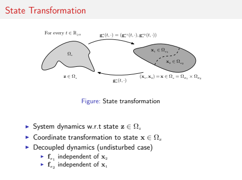


Figure: State transformation

- System dynamics w.r.t state $\mathbf{z} \in \Omega_z$
- Coordinate transformation to state $\mathbf{x} \in \Omega_x$
- Decoupled dynamics (undisturbed case)
  - $\mathbf{f}_{x_1}$ independent of $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ independent of $\mathbf{x}_1$



- We write the dynamics for a state in Ω_z
- For the controller design, we consider a different state, and state set Ω_x
- We provide a mapping from $\mathbf{z}$ to $\mathbf{x}$ and vice-versa (this is a mapping for each time instant)
- The dynamics for the transformed state are in a form for which one can find controllers
- Decoupled: (without disturbance) the dynamics of the transformed state are decoupled, i.e.
  - $\mathbf{f}_{x_1}$ does not depend on $\mathbf{x}_2$
  - $\mathbf{f}_{x_2}$ does not depend on $\mathbf{x}_1$