Feedback Tracking Control Schemes For A Class of Underactuated Vehicles in $\text{SE}(3)$

Rakesh Warier$^1$, Amit Sanyal$^2$, Srikant Sukumar$^1$, & Sasi Prabhakaran$^2$

IIT Bombay$^1$
Syracuse University$^2$

Systems and Control Seminar,
IITB, India
July 26, 2017
Autonomous UAV: Underactuated:
- Six degrees of freedom
- Four degrees of actuation
  - Torque can be applied about any body-fixed coordinate axes
  - Thrust fixed along a single body-fixed axis
  - Only the magnitude of thrust is controlled

Figure: Example: Quadcopter / quadrotor
Coordinate frame definition

- **\( B \)** is the body fixed frame, fixed to the CoM of the vehicle
- **\( I \)** is fixed in space and takes the role of an inertial coordinate frame

Thus the pose of the vehicle in SE(3) given by,

\[
g = \begin{bmatrix} R & b \\ 0 & 1 \end{bmatrix} \in \text{SE}(3) \tag{1}
\]

- **\( R \in SO(3) \)** is rotation matrix from frame **\( B \)** to frame **\( I \)**
- **\( b \in \mathbb{R}^3 \)** is the position vector of the origin of frame **\( B \)** with respect to frame **\( I \)** represented in frame **\( I \)**
Kinematic Equations

Pose kinematics of the vehicle is given by

$$\dot{R} = R(\Omega)^\times,$$
$$\dot{b} = v.$$  

Here,

- $\Omega(t) \in \mathbb{R}^3$ is the angular velocity measured in the body frame
- $v(t) \in \mathbb{R}^3$ is the translational velocity measured in the inertial frame
- $(.)^\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric cross product operator
Dynamics Model

- Dynamics model for the underactuated vehicle is given by,

\[ m \ddot{v} = m g \, e_3 - f \, R \, e_3, \]
\[ J \dot{\Omega} = J \Omega \times \Omega + \tau. \]

Here,

- \( g \) is acceleration due to gravity
- \( e_3 = [0, 0, 1]^T \)
- \(-R \, e_3\) is the inertial direction of the thrust
- \( f \in \mathbb{R} \) is the thrust magnitude
- \( J \in \mathbb{R}^{3 \times 3} \) is the moment of inertia
- \( \tau \in \mathbb{R}^3 \) is the torque
Quadcopter Actuation Model

- VTOL quadcopter UAV model is considered here
  - Four identical actuators (propellers)
  - $D$ is the scalar distance from the propeller axis to the center of UAV
  - $u \in \mathbb{C} \subset \mathbb{R}^4$ is the control input which directly actuates the 3 rotational DoF and 1 translational DoF

- Actuation model of quadcopter is given by

\[
  u = \begin{bmatrix} -f & \tau \end{bmatrix}^T = \mathcal{K} \begin{bmatrix} \bar{\omega}_1 & \bar{\omega}_2 & \bar{\omega}_3 & \bar{\omega}_4 \end{bmatrix}^T,
\]

where

\[
  \mathcal{K} = \begin{bmatrix}
    -k_f & -k_f & -k_f & -k_f \\
    0 & -k_f D & 0 & k_f D \\
    k_f D & 0 & -k_f D & 0 \\
    -k_\tau & k_\tau & -k_\tau & k_\tau
  \end{bmatrix} \in \mathbb{R}^{4 \times 4}
\]

is a constant invertible matrix for $k_f \neq 0$ and $k_\tau \neq 0$
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- **On-line**
  - Compute control thrust $f$
    and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory
Integrated Trajectory Generation and Control

- Off-line
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- On-line
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $\mathbb{C}^2$

- **On-line**
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in \( \mathbb{R}^3 \) w.r.t \( \mathcal{I} \)
  - Generate a desired trajectory \( b^d(t) \in \mathbb{R}^3 \) is \( C^2 \)

- **On-line**
  - Compute control thrust \( f \) and the desired angular velocity \( \Omega^d \)
  - Compute the attitude control torque

- UAV tracking trajectory
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- **On-line**
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- **On-line**
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- **UAV tracking trajectory**
Integrated Trajectory Generation and Control

- Off-line
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- On-line
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory

[Diagram showing a UAV tracking a trajectory with vectors and frame definitions]
Integrated Trajectory Generation and Control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$
  - Generate a desired trajectory $b^d(t) \in \mathbb{R}^3$ is $C^2$

- **On-line**
  - Compute control thrust $f$ and the desired angular velocity $\Omega^d$
  - Compute the attitude control torque

- UAV tracking trajectory
Translational Trajectory Tracking

- Position tracking error: $\tilde{b} := b - b^d$
- Translational velocity tracking error: $\tilde{v} := v - v^d = \dot{\tilde{b}}$
- Acceleration tracking error: $\tilde{a} := \tilde{v}$
- Translational error dynamics in inertial frame $\mathcal{I}$ are:

$$\dot{\tilde{b}} = \tilde{v} = v - v^d, \quad m \dot{\tilde{v}} = mge_3 - fr_3 - m\dot{v}^d,$$

(2)

- After differentiating (2) and reformulating (for $\alpha > 0$),

\[
\begin{cases}
   m \frac{d}{dt} (\tilde{a} + \alpha \tilde{v}) = -m (\ddot{v}^d + \alpha \dot{v}^d) + \alpha m g e_3 - (\dot{f} + \alpha f) R e_3 \\
   -f R (\Omega \times e_3)
\end{cases}
\]

- Fully actuated with controls $\dot{f}$ and $[\Omega_1, \Omega_2, 0]^T$
Translational Trajectory Tracking

- Position tracking error: \( \tilde{b} := b - b^d \)
- Translational velocity tracking error: \( \tilde{v} := v - v^d = \dot{\tilde{b}} \)
- Acceleration tracking error: \( \tilde{a} := \ddot{v} \)
- Translational error dynamics in inertial frame \( \mathcal{I} \) are:

\[
\dot{\tilde{b}} = \tilde{v} = v - v^d, \quad m\dot{\tilde{v}} = mge_3 - \dot{f}r_3 - m\dot{v}^d, \tag{2}
\]

- After differentiating (2) and reformulating (for \( \alpha > 0 \)),

\[
\begin{align*}
    m\frac{d}{dt} (\tilde{a} + \alpha \dot{v}) &= -m (\dot{v}^d + \alpha \dot{v}^d) + \alpha m g e_3 - (\dot{f} + \alpha \dot{f}) R e_3 \\
    -f R (\Omega \times e_3)
\end{align*}
\]

- Fully actuated with controls \( \dot{f} \) and \( [\Omega_{(1)}, \Omega_{(2)}, 0]^T \)
Control Framework

- **Position tracking control.** Design $\dot{\mathbf{f}}$ and $[\Omega^1, \Omega^2, 0]^T$ for tracking the desired position trajectory.

- **Angular velocity tracking control.** Synthesize control torque $\tau$ so that angular velocity tracks the angular velocity profile obtained from the position tracking control design.
Theorem

Let the feedback control law be given by:

\[
\dot{f} = -\alpha f + mR^T(\zeta(\ddot{a} + \alpha \tilde{v}) + \lambda(\tilde{v} + \alpha \tilde{b})) \cdot e_3 - mR^T(\ddot{v}^d + \alpha \dot{v}^d) \cdot e_3 + (\alpha mgR^T e_3) \cdot e_3
\]

\[
[\Omega_1, \Omega_2, 0]^T = \left(\frac{1}{f}\right)e_3 \times \left(mR^T(\zeta(\ddot{a} + \alpha \tilde{v}) + \lambda(\tilde{v} + \alpha \tilde{b}))\right) - \left(\frac{1}{f}\right)e_3 \times \left(mR^T(\ddot{v}^d - \alpha \dot{v}^d) + \alpha mgR^T e_3\right)
\]

where \(\zeta, \lambda \in \mathbb{R}^+\) satisfying \(\zeta^2 \neq 4\lambda\). Then tracking error dynamics is stabilized to \((\ddot{b}, \tilde{v}, \ddot{a}) = (0_{1 \times 3}, 0_{1 \times 3}, 0_{1 \times 3})\) exponentially.
Exponential stability of position tracking control shown by expressing the feedback dynamics for $\tilde{b}$ as:

$$
\left( \frac{d^3}{dt^3} + (\alpha + \zeta) \frac{d^2}{dt^2} + (\zeta \alpha + \lambda) \frac{d}{dt} + (\alpha \lambda) \right) \tilde{b} = \mathbf{0}_{1 \times 3}, \quad (3)
$$

which has non-repeating roots with negative real parts.
Angular Velocity Tracking Control

- Angular velocity $\Omega$ is controlled to achieve stable tracking of desired translational motion
- Desired angular velocity $\Omega^d(t)$ chosen to be
  \[
  \Omega^d = \left(\frac{1}{f}\right) \mathbf{e}_3 \times \left( mR^T (\ddot{a} + \alpha \ddot{v}) + \lambda (\ddot{v} + \alpha \dddot{b}) \right) \\
  - \left(\frac{1}{f}\right) \mathbf{e}_3 \times \left( mR^T (\ddot{v}^d - \alpha \dot{v}^d) + \alpha mgR^T \mathbf{e}_3 \right)
  \]
- The third component of angular velocity chosen to be zero
- Angular velocity error defined as $e_\Omega = \Omega - \Omega^d$
- The error dynamics are
  \[
  \frac{d}{dt} (J e_\Omega) = J\Omega \times \Omega + \tau - J \frac{d}{dt} (\Omega^d)
  \]
The feedback control law $\tau$ for angular velocity control is given by

$$
\tau = - (J\Omega) \times \Omega + J \frac{d}{dt}(\Omega^d) - \frac{L(\Omega - \Omega^d)}{((\Omega - \Omega^d)^T L(\Omega - \Omega^d))^{1 - \frac{1}{p}}}
$$

where $L$ is a positive definite matrix and $p \in (1, 2)$. Then, the angular velocity error dynamics is stabilized to $e_\Omega = 0_{1 \times 3}$ in finite time.

Finite-time stability of the angular velocity tracking control scheme is shown by the Lyapunov function:

$$
\mathcal{V}_{rot} = \frac{1}{2}(\Omega - \Omega^d)^T J(\Omega - \Omega^d)
$$

(4)
Finite-Time Stable Angular Velocity Tracking Control

Theorem

The feedback control law $\tau$ for angular velocity control is given by

$$
\tau = -(J\Omega) \times \Omega + J \frac{d}{dt}(\Omega^d) - \frac{L(\Omega - \Omega^d)}{((\Omega - \Omega^d)^T L(\Omega - \Omega^d))^{1 - \frac{1}{p}}}
$$

where $L$ is a positive definite matrix and $p \in (1, 2)$. Then, the angular velocity error dynamics is stabilized to $e_\Omega = 0_{1 \times 3}$ in finite time.

Finite-time stability of the angular velocity tracking control scheme is shown by the Lyapunov function:

$$
V_{rot} = \frac{1}{2}(\Omega - \Omega^d)^T J(\Omega - \Omega^d)
$$

(4)
Stability of the Overall Feedback System

Theorem

The overall feedback system given by translational tracking error dynamics and the angular velocity error dynamics is exponentially stabilized to 
\((\tilde{b}, \tilde{\nu}, \tilde{\alpha}, e_{\Omega}) = (0_{1\times3}, 0_{1\times3}, 0_{1\times3}, 0_{1\times3})\)

Overall stability of system is shown by: (1) finite-time convergence and stability of angular velocity to desired angular velocity profile; and (2) exponential convergence of position tracking errors to zero thereafter.
Numerical Simulations

- Moment of inertia and mass of UAV are selected as
  $J = \text{diag}([0.0820, 0.0845, 0.1377]) \text{kg m}^2$, \hspace{1em} $m = 4.34 \text{kg}$.
- Initial configuration:
  \[
b(0) = [0, 0.35, 0]^T \text{ m}, \quad \nu(0) = [0, 0, 0]^T \text{ m/s}
  \]
  \[
  R(0) = I, \quad \Omega(0) = [0, 0, 1]^T \text{ rad/s}.
  \]
- At least two inertial sensors including three-axis rate gyro are assumed to be onboard the rigid body.
- The desired trajectory is chosen to be helical,
  $b^d(t) = [0.4t, 0.4 \cos(t), 0.6 \sin(t)]^T \text{ m}$
Desired and achieved position trajectories of the simulated vehicle
Tracking Errors
Control Inputs

Graph 1: Force $f[N]$ over time $[s]$

Graph 2: Torque $\tau[N-m]$ over time $[s]$, showing three curves $\tau_x$, $\tau_y$, and $\tau_z$. 
Conclusion

- Trajectory generation
- Exponentially stable position trajectory tracking
- Finite time stable angular velocity tracking
- Yaw rate stabilized to zero
- Validation of integrated guidance and control

Future Work
- Experimental Results
Thank you!
Integrated Guidance and Nonlinear Feedback Control of Underactuated Unmanned Aerial Vehicles in SE(3)

Sasi Prabhakaran. V  
Postdoc  
Syracuse University

Amit K. Sanyal  
Associate Professor  
Syracuse University

Rakesh R. Warier  
PhD Student  
IIT Bombay

Systems and Control Seminar, IIT Bombay

Mumbai, India
Unmanned Vehicle Systems

Autonomous Multirotor UAV
Unmanned Vehicle Systems

Autonomous Multirotor UAV

Underactuated:

- Four independent control inputs
- Six Degrees of Freedom (DoF); 3 DoF attitude is controlled
- Only the magnitude of thrust (1 DoF in translation) is controlled
Coordinate frame definition

- $\mathcal{B}$ is a body fixed frame, fixed to the mass center of the vehicle
- $\mathcal{I}$ is fixed in space and takes the role of an inertial coordinate frame

The pose of the vehicle is represented in matrix form as follows:

$$g = \begin{bmatrix} R & b \\ 0 & 1 \end{bmatrix} \in \text{SE}(3)$$  \hspace{1cm} (1)$$

- $R \in \text{SO}(3)$ is rotation matrix from frame $\mathcal{B}$ to frame $\mathcal{I}$
- $b \in \mathbb{R}^3$ denotes position vector of origin of frame $\mathcal{B}$ with respect to frame $\mathcal{I}$ represented in frame $\mathcal{I}$
Pose kinematics of the vehicle is given by,

\[
\dot{g}(t) = g(t)\xi(t)^\vee,
\]

(2)

with \( \xi^\vee = \begin{bmatrix} \Omega^\times & \nu \\ 0 & 0 \end{bmatrix} \in \mathfrak{se}(3) \subset \mathbb{R}^{4\times4} \) for \( \xi = \begin{bmatrix} \Omega \\ \nu \end{bmatrix} \in \mathbb{R}^{6} \).

Here,

- \( \Omega(t) \in \mathbb{R}^{3} \) is the angular velocity
- \( \nu(t) \in \mathbb{R}^{3} \) is the translational velocity
- \((\cdot)^\times : \mathbb{R}^{3} \rightarrow \mathfrak{so}(3) \subset \mathbb{R}^{3\times3}\) is the skew-symmetric cross-product operator that gives the vector space isomorphism between \( \mathbb{R}^{3} \) and \( \mathfrak{so}(3) \).
“Nominal” model of the dynamics for the underactuated vehicle is given by

\[
\dot{\mathbb{I}}\dot{\xi} = \text{ad}^*_\xi \mathbb{I} \dot{\xi} + \varphi(g, \xi) + B u, \quad u \in \mathcal{C} \subset \mathbb{R}^4,
\]

where,

\[
\mathbb{I} = \begin{bmatrix} J & 0_{3 \times 3} \\ 0_{3 \times 3} & ml_3 \end{bmatrix} \in \mathbb{R}^{6 \times 6}, \quad B = \begin{bmatrix} l_3 & 0_{3 \times 1} \\ 0_{3 \times 3} & e_3 \end{bmatrix} \in \mathbb{R}^{6 \times 4},
\]

\( \mathbb{I} \) denotes the mass \((m)\) and inertia \((J)\) properties of the vehicle

\( \varphi(g, \xi) \in \mathbb{R}^6 \) is the vector of known (modeled) moments and forces

\( e_3 = [0 \ 0 \ 1]^T \in \mathbb{R}^3 \)
Quadcopter Actuation Model

VTOL quadcopter UAV model is considered here:
- 4 identical actuators (propellers)
- $D$ is the scalar distance from the propeller axis to the center of UAV
- $u \in \mathbb{R}^4$ is the control input which directly actuates the 3 rotational DoF and 1 translational DoF

Actuation model of quadcopter is given by

$$u = \begin{bmatrix} \tau^T & f \end{bmatrix}^T = \mathcal{K} \begin{bmatrix} \tilde{\omega}_1^2 & \tilde{\omega}_2^2 & \tilde{\omega}_3^2 & \tilde{\omega}_4^2 \end{bmatrix}^T,$$

where

$$\tau \in \mathbb{R}^3, \quad \mathcal{K} = \begin{bmatrix} -k_f & -k_f & -k_f & -k_f \\ 0 & -k_f D & 0 & k_f D \\ k_f D & 0 & -k_f D & 0 \\ -k_\tau & k_\tau & -k_\tau & k_\tau \end{bmatrix} \in \mathbb{R}^{4 \times 4}$$

is a constant invertible matrix for $k_f \neq 0$ and $k_\tau \neq 0$. 
Integrated trajectory generation and control

Inertial frame, $\mathcal{I} := \{X, Y, Z\}$

Body-fixed frame, $\mathcal{B} := \{a_1, a_2, a_3\}$

Inspect the environment

Initialize the UAV

Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$

Generate a desired trajectory $b_d(t) = \mathbb{C}(\mathbb{R}^3)$

On-line

Compute control thrust

Generate the desired attitude trajectory, $\mathbb{R}_d(t)$

Compute the attitude control torque

UAV tracking trajectory

Prabhakaran, Sanyal, Warier
Guidance Scheme

Overview

Integrated trajectory generation and control

- Off-line
  - Inspect the environment

- On-line
Guidance Scheme

Overview

Integrated trajectory generation and control

**Off-line**
- Inspect the environment
- Initialize the UAV

**On-line**

---

**Guidance Scheme**

- Off-line
  - Inspect the environment
  - Initialize the UAV

- On-line

---

Inertial frame, \( \mathcal{I} := \{X, Y, Z\} \)

Body-fixed frame, \( B := \{a_1, a_2, a_3\} \)
Integrated trajectory generation and control

- **Off-line**
  - Inspect the environment
  - Initialize the UAV
  - Select desired waypoints in $\mathbb{R}^3$ w.r.t $\mathcal{I}$

- **On-line**
Integrated trajectory generation and control

**Off-line**
- Inspect the environment
- Initialize the UAV
- Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
- Generate a desired trajectory $b_d(t) = C^2(\mathbb{R}^3)$

**On-line**
Integrated trajectory generation and control

Off-line
- Inspect the environment
- Initialize the UAV
- Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
- Generate a desired trajectory $b_d(t) = C^2(\mathbb{R}^3)$

On-line
- Compute control thrust $f$
- Generate the desired attitude trajectory, $R_d(t)$
- Compute the attitude control torque

Inertial frame, $\mathcal{I} := \{X, Y, Z\}$
Body-fixed frame, $\mathcal{B} := \{a_1, a_2, a_3\}$
Integrated trajectory generation and control

**Guidance Scheme**

**Overview**

**Integrated trajectory generation and control**

---

**Off-line**
- Inspect the environment
- Initialize the UAV
- Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
- Generate a desired trajectory $b_d(t) = C^2(\mathbb{R}^3)$

**On-line**
- Compute control thrust $f$
- Generate the desired attitude trajectory, $R_d(t)$
- Compute the attitude control torque

**UAV tracking trajectory**

Inertial frame, $I := \{X, Y, Z\}$
Integrated trajectory generation and control

Off-line
- Inspect the environment
- Initialize the UAV
- Select desired waypoints in $\mathbb{R}^3$ w.r.t $I$
- Generate a desired trajectory $b_d(t) = C^2(\mathbb{R}^3)$

On-line
- Compute control thrust $f$
- Generate the desired attitude trajectory, $R_d(t)$
- Compute the attitude control torque

UAV tracking trajectory
Tracking errors definition

- Position tracking error: $\tilde{b} := b - b_d$
- Translational velocity tracking error: $\tilde{v} := v - v_d = \dot{b}$
- Attitude tracking error: $Q = R_d^T R$
- Angular velocity tracking error: $\omega = \Omega - Q^T \Omega_d$

Translational error dynamics in inertial frame $\mathcal{I}$ is:

$$\begin{align*}
\dot{\tilde{b}} &= \tilde{v} = v - v_d, \\
m\dot{\tilde{v}} &= mge_3 - fr_3 - mv_d,
\end{align*}$$

(True attitude of the body is represented by $r_3 = Re_3$
- Control force vector, $fr_3 = \bar{\varphi}_c$ is expressed in $\mathcal{I}$)
Theorem

Consider the translational dynamics given in (4); define

\[ z(t) = \frac{\tilde{b}}{\left( \tilde{b}^T \tilde{b} \right)^{1/\rho}}, \]  

where \( \rho \in (1, 2) \). The feedback control \( u \in \mathbb{R}^3 \) given by,

\[ u = ge_3 - \tilde{v}_d + k \dot{z} + k \tilde{b} + \frac{kP(\tilde{v} + kz)}{\left[ (\tilde{v} + kz)^T P(\tilde{v} + kz) \right]^{1/\rho}}, \]

where control gain matrix \( P \succ 0 \), stabilizes the translational error dynamics

\[ \dot{\tilde{v}} = ge_3 - \tilde{v}_d - u; \text{ where } mu(t) = f(t)r_3, \]

in finite time.
Generating desired attitude trajectory

- Attitude is controlled to achieve stable tracking of the desired translational motion
- Desired attitude $R_d = [r_{2d} \times r_{3d} \quad r_{2d} \quad r_{3d}] \in SO(3)$ is generated as follows:
  \[
  r_{3d} = R_d e_3 = \frac{u}{\|u\|}. 
  \] (7)
- Now compute $r_{2d} = \frac{r_{3d} \times s_d}{\|r_{3d} \times s_d\|} = R_d e_2$, by selecting an appropriate $s_d(t) \in C^2(\mathbb{R}^3)$ such that it is transverse to $r_{3d}$

**Theorem**

Let $r_{3d} = \left[ a_1 \quad a_2 \quad a_3 \right]^T \in S^2 \subset \mathbb{R}^3$ be a known unit vector as given in (7). The vector
\[
  s_d = \left[ a_2 + a_3 \quad a_3 - a_1 \quad -a_1 - a_2 \right]^T 
  \] (8)
is orthogonal to $r_{3d}$. 
Finite-time Stable Attitude Tracking Control on TSO(3)

Theorem

The feedback control law $\tau_c$ for attitude control is given by

$$
\tau_c = J \left( Q^T \dot{\Omega}_d - \frac{\kappa H(s_K(Q))}{(s_K(Q)^T s_K(Q))^{1-1/p}} \omega(Q, \omega) \right) \\
+ (Q^T \Omega_d)^T J(Q^T \Omega_d - \kappa z_K(Q)) + \kappa J(z_K(Q) \times Q^T \Omega_d) \\
+ \kappa J(\omega + Q^T \Omega_d) \times z_K(Q) - k_p s_K(Q) \\
- \frac{L \Psi(Q, \omega)}{(\Psi(Q, \omega)^T L \Psi(Q, \omega))^{1-1/p}},
$$

(9)

where

$$
\Psi(Q, \omega) = \omega + \kappa z_K(Q), \text{ and } H(x) = I - \frac{2(1 - 1/p)}{x^T x} xx^T.
$$

(10)

Then the feedback attitude tracking error dynamics is stabilized to $(Q, \omega) = (I, 0)$ in finite time.
Stability of the Overall Feedback System on $\text{TSE}(3)$

**Theorem**

The overall feedback control system given by the tracking error kinematics and dynamics is finite-time stable for the generated state trajectory $(b_d(t), R_d(t), v_d(t), \Omega_d(t)) \subset \text{TSE}(3)$. Moreover, the domain of convergence is almost global over the state space.

Lyapunov functions used (for translational and rotational motions):

$$V_{tr}(\tilde{b}, \tilde{v}) = \frac{1}{2} \left( k \tilde{b}^T \tilde{b} + (\tilde{v} + kz)^T (\tilde{v} + kz) \right), \quad (11)$$

$$V_{rot}(Q, \omega) = k_p \langle K, I - Q \rangle + \Psi(Q, \omega)^T J \Psi(Q, \omega). \quad (12)$$
Numerical Results

$\cdots\cdots \quad b_d \quad \text{desired trajectory}$

$\quad b \quad \text{achieved trajectory}$
Numerical Results

(a) Norm of position tracking error
(b) Norm of velocity tracking error
(c) Attitude tracking error
(d) Magnitude of total thrust
Position trajectory tracking on $\mathbb{R}^3$ through given waypoints

- Thrust force control for finite-time stable position trajectory tracking
- Attitude trajectory generation on $SO(3)$ based on desired thrust direction
- Finite time stable attitude control to track required attitude
- Validation of integrated guidance and control
Future/Ongoing Work

- Optimal $C^2$ position trajectory generation with known bounds on velocities and accelerations
- Optimal $C^2$ position and pointing trajectory generation through given waypoints (on $\mathbb{R}^3 \times S^2$)
- Experiments to be carried out in Autonomous Unmanned Systems Laboratory (AUSL), Syracuse Center of Excellence
THANKS

Questions and Comments