Robust Adaptive Attitude Control of a Spacecraft

AER1503 Spacecraft Dynamics and Controls II
April 24, 2015
Agenda

Introduction

Model Formulation

Controller Designs

Simulation Results
Several Ways to control spacecraft attitude

Difficult because:

- MIMO nonlinear system
- Parametric uncertainties
- Non-parametric uncertainties

Will present a controller with the following components:

- PD feedback
- Feedforward – tracking response
- Adaptive – robustness to parametric
- Sliding Mode – robustness to non-parametric
Model Formulation

Motion Equations:
\[ I \ddot{\omega} + \omega \times I \omega = u \]
\[ \dot{\epsilon} = -\frac{1}{2} \omega \times \epsilon + \frac{1}{2} \eta \omega \]
\[ \dot{\eta} = -\frac{1}{2} \omega^T \epsilon \]

Euler Identities:
\[ \epsilon^T \epsilon + \eta^2 = 1 \]
\[ \epsilon = a \sin \frac{\phi}{2}, \quad \eta = \cos \frac{\phi}{2} \]

Quaternion Altitude Error:
\[ \epsilon_e = \eta \alpha \epsilon - \eta \epsilon_d - \epsilon_d \times \epsilon \]
\[ \eta_e = \eta \eta_d + \epsilon^T \eta_d \]

Spacecraft plant implemented in Simulink
Controller Design: Adaptive Control

- Using feedback and feedforward control, the equil. is GAS
  - Must know inertia matrix exactly

- With adaptive control, the inertia matrix can be estimated

- **Goal:** have the system behave as if the unknown parameters are known
  - **Adaptation law:** \( \hat{a} = -\Gamma Y^T s \)
  - **Control law:** \( u = Y \hat{a} - K_D s \)
    - \( \Gamma \) is the adaptation gain
    - \( a = [I_{11} \ I_{22} \ I_{33} \ I_{12} \ I_{13} \ I_{23}]^T \) is the unknown elements
    - \( \hat{a} \) is the estimated elements
    - \( s \) is the smoothed error
Controller Design: Adaptive Control

- **Y** is the regressor matrix corresponding to:
  \[ \dot{I}\omega_r + \omega^r \times \dot{I}\omega = Y\hat{a} \]

\[
Y = \begin{bmatrix}
\dot{\omega}_{r1} & -\omega_2\omega_{r3} & \omega_3\omega_{r2} & \dot{\omega}_{r2} - \omega_1\omega_{r3} & \dot{\omega}_{r3} + \omega_1\omega_{r2} & \omega_2\omega_{r2} - \omega_3\omega_{r3} \\
\omega_1\omega_{r3} & \dot{\omega}_{r2} & -\omega_3\omega_{r1} & \dot{\omega}_{r1} + \omega_2\omega_{r3} & -\omega_1\omega_{r1} + \omega_3\omega_{r3} & \dot{\omega}_{r3} - \omega_2\omega_{r1} \\
-\omega_1\omega_{r2} & \omega_2\omega_{r1} & \dot{\omega}_{r3} & \omega_1\omega_{r1} - \omega_2\omega_{r2} & \dot{\omega}_{r1} - \omega_3\omega_{r2} & \dot{\omega}_{r2} + \omega_3\omega_{r1}
\end{bmatrix}

- Parameter estimation error: \( \hat{a} = \hat{a} - a \)

- **Lyapunov Proof:**
  \[
  V(t) = \frac{1}{2} s^T Is + \frac{1}{2} \tilde{a}^T \Gamma^{-1} \tilde{a}
  \]

  \[
  \dot{V} = s^T I\dot{s} + \tilde{a}^T \Gamma^{-1} \tilde{a}
  = s^T I\dot{\omega} - s^T I\omega_r + \tilde{a}^T \Gamma^{-1} \tilde{a}
  = s^T(u - \omega^r \times I\omega) - s^T(Y\hat{a} - \omega^r \times I\omega) + \tilde{a}^T \Gamma^{-1}(\hat{a} - a)
  = s^T(Y\hat{a} - K DS - \omega^r \times I\omega) - s^T(Y a - \omega^r \times I\omega) + \tilde{a}^T \Gamma^{-1}(\hat{a} - a)
  = -s^TK DS + s^TY(\hat{a} - a) - s^T \omega^r \times I\omega + (-s^TY \Gamma) \Gamma^{-1}(\hat{a} - a)
  = -s^T K D S
  \]

- **Barbalat’s lemma:** \( \{ V(t) \text{ lower bounded}, \dot{V} \leq 0, \ddot{V} \text{ bounded} \} \), then \( \dot{V} \to 0 \). So \( s \) converges to zero and the system is GAS. (\( \varepsilon \to \varepsilon_d , \eta \to \eta_d \))
Controller Design: Adaptive Control

- Convergence is not exact in the estimated parameters
  - Controller will generate values that allow the tracking error to converge to zero
  - Trajectory must be sufficiently “rich” for convergence ($\hat{a} \to a$)

- Design parameters: $\lambda, K_D$ and $\Gamma$ are limited in magnitude
  - Due to high frequency unmodeled dynamics
    - actuator dynamics
    - structural resonant modes
    - sampling limitations
    - measurement noise

AER 1503 – Robust Adaptive Control
April 2015
Controller Design: Adaptive Control

Adaptive and feedforward block

Closed loop system with adaptation
Controller Design: Robust Adaptive

- Non-parametric uncertainties can reduce performance of controller when placed online
  - Drifting of estimated parameter terms in the adaptive controller
- Robustness in the adaptive controller can be achieved with sliding mode control
  - This creates a dead zone where the system does not adapt

- New smoothed error: $s_\Delta$
  - Such that: $|s| \leq \phi \iff s_\Delta = 0$
  - $s_\Delta = s - \phi \text{ sat} \left( \frac{s}{\phi} \right)$

Illustration of saturator function with dead zone
Controller Design: Robust Adaptive

- **Modified adaptation law:** \( \dot{\hat{a}} = -\Gamma Y^T s_{\Delta} \)
- **Motion equations with disturbance:** \( I\dot{\omega} + \omega \times I\omega = u + d \)
- **Lyapunov Proof:**

\[
V(t) = \frac{1}{2} I s_{\Delta}^2 + \frac{1}{2} \tilde{a}^T \Gamma^{-1} \tilde{a}
\]

\[
\dot{V} = s_{\Delta}^T I \dot{s} + \tilde{a}^T \Gamma^{-1} \tilde{a}
\]

\[
= s_{\Delta}^T (Y\dot{\hat{a}} - K_D s - \omega \times I\omega + d) - s_{\Delta}^T (Y\dot{a} - \omega \times I\omega) + \tilde{a}^T \Gamma^{-1} (\hat{a} - a)
\]

\[
= -s_{\Delta}^T K_D \left( s_{\Delta} + \emptyset \text{ sat} \left( \frac{s}{\emptyset} \right) \right) + s_{\Delta}^T Y (\dot{\hat{a}} - \dot{a}) - s_{\Delta}^T s \times I\omega + (-s_{\Delta}^T Y \Gamma) \Gamma^{-1} (\hat{a} - a)
\]

\[
= -s_{\Delta}^T K_D s_{\Delta} - K_D \emptyset |s_{\Delta}| + s_{\Delta} d
\]

\[
\leq -s_{\Delta}^T K_D s_{\Delta}
\]

- **Barbalat’s lemma:** \( \dot{V} \leq 0 \Rightarrow s_{\Delta} \to 0 \)
Controller Design: Robust Adaptive

Robust adaptive and feedforward block

Closed loop system with adaptation and noise
# Simulation Results

## Simulation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertia Matrix</td>
<td>$I$</td>
<td>$\begin{bmatrix} 15 &amp; 5 &amp; 5 \ 5 &amp; 10 &amp; 7 \ 5 &amp; 7 &amp; 20 \end{bmatrix}$ $kg \cdot m^2$</td>
</tr>
<tr>
<td>Spacecraft initial state</td>
<td>$x(0)$</td>
<td>$[0 \ 0 \ 0 \ 1]^T$ rad</td>
</tr>
<tr>
<td>Proportional gain in PD controller</td>
<td>$K_p$</td>
<td>200</td>
</tr>
<tr>
<td>Derivative gain in PD controller</td>
<td>$K_d$</td>
<td>20</td>
</tr>
<tr>
<td>Smoothed error $\varepsilon_e$ weight</td>
<td>$\lambda$</td>
<td>25</td>
</tr>
<tr>
<td>Initial estimate of inertia matrix params</td>
<td>$\hat{a}(0)$</td>
<td>$[15 \ 10 \ 20 \ 5 \ 5 \ 7]^T$ $kg \cdot m^2$</td>
</tr>
<tr>
<td>Adaptation law gain</td>
<td>$\Gamma$</td>
<td>15</td>
</tr>
<tr>
<td>Sliding mode dead zone</td>
<td>$\phi$</td>
<td>$\pm 0.15$</td>
</tr>
</tbody>
</table>
Simulation Results

**Smoothed error without sliding mode**

- At steady state, \( s \leq 0.1 \) so dead zone chosen to be \( \varnothing = 0.15 \) to prevent parameter drift

**Smoothed error with sliding mode**
Simulation Results

- Note how the tracking is almost perfect
Possible Controller Improvements

- Robust adaptive controller for s/c attitude tracking is not optimal

- Possible solutions:
  - Nonlinear Quadratic Regulator
  - Nonlinear Model Predictive Controller
    - MPCs considers the system actuation restrictions
    - Stability and robustness can be ensured through the proper choice of terminal constraints.
Conclusion

- Robust adaptive controller was designed for spacecraft attitude tracking
- Closed loop system was simulated entirely in software using Simulink
- Concepts from sliding mode control were used to add system robustness
Questions?
References


minimize \[ \sum_{t=0}^{T-1} l(x(t), u(t)) \]

subject to \[ c_{eq}(x) = 0 \]
\[ A_{eq}x = B_{eq} \]
\[ lb \leq x \leq ub \]
\[ x(0) = z, x(T) = 0 \]

The process of calculating the optimal input is as follows:

1) Measure/estimate the current system state \( z \)
2) Solve the optimization problem for the optimal control action plan (set \( U_t^* \)) based on the horizon defined by \( T \)
3) Execute the first optimal control action \( u_t^* \) in the action plan
4) Repeat
Controller Design: PD Feedback

Quaternion Altitude Error:

\[ \varepsilon_e = \eta_d \varepsilon - \eta \dot{\varepsilon}_d - \varepsilon_d \times \varepsilon \]
\[ \eta_e = \eta \eta_d + \varepsilon^T \eta_d \]

PD Control Law:

\[ u(t) = -K_d \omega(t) - k \varepsilon_e(t), \quad K_d = K_d^T, \quad k > 0 \]
Lyapunov Function Candidate:

\[ V(t) = \frac{1}{2} \omega^T I \omega + k [\varepsilon^T_e \varepsilon_e + (\eta_e - 1)^2] \]

\[ \dot{V} = \omega \times I \omega + 2k [\varepsilon^T_e \dot{\varepsilon}_e + (\eta_e - 1)\dot{\eta}_e] \]

\[ = \omega \times (u - \omega \times I \omega) \]
\[ = \omega \times (-K_d \omega - k \varepsilon_e - \omega \times I \omega) \]
\[ = -\omega \times K_d \omega \]

Figure 2: PD controller with plant
Controller Design: Feedforward

**Feedforward Torque:** \( u_d = I \dot{\omega}_r + \omega_r \times I \omega \)

**Controller Output:** \( u = u_d + \ddot{u} \)

**Smoothed Error:** \( s = \ddot{\omega} + \lambda \varepsilon_e = \omega - \omega_r \)

**Reference Angular Velocity:** \( \omega_r = \omega_d - \lambda \varepsilon_e \)

**Lyapunov Function Candidate:**

\[
V(t) = \frac{1}{2} \dddot{\omega}^T I \dddot{\omega} \\
\dot{V} = \dddot{\omega}^T I \dddot{\omega} \\
= \dddot{\omega}^T (\dddot{u} - \dddot{\omega} \times I \omega) \\
= \dddot{\omega}^T \dddot{u}
\]

Integrating both sides from \( t = 0 \) to \( T \):

\[
\int_0^T \dddot{\omega}^T \dddot{u} \, dt = V(T) - V(0) \\
= V(T) \\
\geq 0
\]
Controller Design: Feedforward

**Figure 3: Feedforward block**

**Figure 4: PD and feedforward controller**