A Miniaturized Satellite Attitude Determination and Control System with Autonomous Calibration Capabilities

Sanny Omar
Dr. David Beale
Dr. JM Wersinger
Introduction

• ADACS designed for CubeSats

• CubeSats generally range in size from 1U to 3U (10x10x(10-30) cm)
  – Relatively cheap and easy to build
  – Highly capable due to miniaturized technology

• Require attitude control systems to properly orient science instruments and antennas
# Sensors and Actuators Trade Study

10 is most desirable, 1 is least desirable

<table>
<thead>
<tr>
<th>Name</th>
<th>Cost</th>
<th>Complexity</th>
<th>Legacy</th>
<th>Accuracy</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Magnetometer</td>
<td>8</td>
<td>8</td>
<td>10</td>
<td>6</td>
<td>32</td>
</tr>
<tr>
<td>Horizon Sensor</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>Sun Sensor</td>
<td>10</td>
<td>9</td>
<td>10</td>
<td>2</td>
<td>31</td>
</tr>
<tr>
<td>Star Tracker</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>10</td>
<td>19</td>
</tr>
<tr>
<td>Rate Gyros</td>
<td>10</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>Magnetorquers</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>5</td>
<td>31</td>
</tr>
<tr>
<td>3-axis Reaction Wheels</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>9</td>
<td>30</td>
</tr>
<tr>
<td>Gravity Boom</td>
<td>10</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>23</td>
</tr>
<tr>
<td>Aerodynamic Stabilization</td>
<td>10</td>
<td>10</td>
<td>2</td>
<td>5</td>
<td>27</td>
</tr>
</tbody>
</table>
Sensors and Actuators

- Reaction Wheels (3)
- Magnetorquers (3)
- Magnetometer (3-axis)
- Rate gyros (3-axis)
- Position Sensitive Detectors (PSDs)
Coordinate Frames

- Earth Fixed Inertial Frame – Origin at Earth center, x-axis toward vernal equinox, z-axis through North Pole
- Orbital Frame – Origin at satellite cm, x-axis aligned with velocity, z-axis toward Earth
- Satellite Body Frame – Origin at satellite cm and moves with the satellite
Coordinate Frame Transformation

- Direction Cosine Matrix

\[
\begin{bmatrix}
  r_{x2} \\
  r_{y2} \\
  r_{z2}
\end{bmatrix} =
\begin{bmatrix}
  a_{11} & a_{12} & a_{13} \\
  a_{21} & a_{22} & a_{23} \\
  a_{31} & a_{32} & a_{33}
\end{bmatrix}
\begin{bmatrix}
  r_{x1} \\
  r_{y1} \\
  r_{z1}
\end{bmatrix}
\]

- Quaternion
  - Rotation vector ($\vec{n}$) plus rotation about vector ($\theta$)

\[
\begin{bmatrix}
  q_0 \\
  q_1 \\
  q_2 \\
  q_3
\end{bmatrix} =
\begin{bmatrix}
  \cos(\theta/2) \\
  n_x \sin(\theta/2) \\
  n_y \sin(\theta/2) \\
  n_z \sin(\theta/2)
\end{bmatrix}
\]
Attitude Vectors

- Sun vector in body frame from PSDs
- Magnetic field vector in body frame from magnetometer
- Magnetic field vector in orbital frame from IGRF model
- Sun vector in orbital frame from heliocentric orbit propagator
- Angular velocity vector (in body frame) from rate gyros
Attitude Determination

• Defined as direction cosine matrix between body frame and orbital frame

• Triad algorithm gives DCM based on two vectors known in both frames

\[
\begin{bmatrix}
\overrightarrow{R_1} & \overrightarrow{R_2} & (\overrightarrow{R_1} \times \overrightarrow{R_2})
\end{bmatrix} = [A]
\begin{bmatrix}
\overrightarrow{r_1} & \overrightarrow{r_2} & (\overrightarrow{r_1} \times \overrightarrow{r_2})
\end{bmatrix}
\]

• DCM to quaternion conversion

\[
q_0^2 = \frac{\text{tr}(A) + 1}{4} = \frac{a_{11} + a_{22} + a_{33} + 1}{4}
\]

\[
q_1 = \frac{a_{32} - a_{23}}{4q_0}
\]

\[
q_2 = \frac{a_{13} - a_{31}}{4q_0}
\]

\[
q_3 = \frac{a_{21} - a_{12}}{4q_0}
\]
Error Quaternion

• Relates current orientation to desired orientation
  – Used in control algorithm
• Calculated based on measured and desired quaternions \((q_m \text{ and } q_d)\)

\[
L_d = \begin{bmatrix}
q_{0d} & q_{1d} & q_{2d} & q_{3d} \\
-q_{1d} & q_{0d} & q_{3d} & -q_{2d} \\
-q_{2d} & -q_{3d} & q_{0d} & q_{1d} \\
-q_{3d} & q_{2d} & -q_{1d} & q_{0d}
\end{bmatrix}
\]

\[
q_e = L_d q_m
\]
Control Algorithm

- B-Dot de-tumble algorithm
  - Minimize angular velocity by torqueing opposite to direction of angular velocity vector
    \[
    \hat{\mu} = k\hat{B}
    \]
    \[
    \hat{T} = \hat{\mu} \times \hat{B}
    \]
- Inverse Dynamics PD Controller for steady state pointing
  - Proportional (error) and derivative terms
  - Analytical gain estimation from block diagram

\[
T(s) = \frac{k_p}{s^2 - sk_d + \frac{k_p}{2}}
\]
**Inverse Dynamics PD Controller**

- Takes satellite dynamics into account
  - Ensures stability in all conditions
- PD law determines desired angular accelerations
- Equations of motion used to calculate required torques
- Gains independent of inertia properties

\[
\begin{bmatrix}
\alpha_x \\
\alpha_y \\
\alpha_z \\
\end{bmatrix} = k_p \begin{bmatrix} q_{e1} \\
q_{e2} \\
q_{e3} \\
\end{bmatrix} + k_d \begin{bmatrix} \omega_x \\
\omega_y \\
\omega_z \\
\end{bmatrix} \\
\sum M = \frac{dH}{dt} = 0
\]

\[
\begin{bmatrix}
T_x \\
T_y \\
T_z \\
\end{bmatrix}_m = I_w \begin{bmatrix} \alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\end{bmatrix}_w = \begin{bmatrix} I_{11} & 0 & 0 \\
0 & I_{22} & 0 \\
0 & 0 & I_{33} \\
\end{bmatrix}_{sc} \begin{bmatrix} \omega_x \\
\omega_y \\
\omega_z \\
\end{bmatrix}_{sc} + I_w \begin{bmatrix} \alpha_1 \\
\alpha_2 \\
\alpha_3 \\
\end{bmatrix}_w \times \begin{bmatrix} \omega_x \\
\omega_y \\
\omega_z \\
\end{bmatrix}_{sc} - \begin{bmatrix} I_{xx} & 0 & 0 \\
0 & I_{yy} & 0 \\
0 & 0 & I_{zz} \\
\end{bmatrix}_{sc} \begin{bmatrix} \alpha_x \\
\alpha_y \\
\alpha_z \\
\end{bmatrix}_{sc}
\]
Drawbacks of Direct Linear PD Controller

- Torques calculated directly from controller
  \[
  \begin{bmatrix}
  T_x \\
  T_y \\
  T_z
  \end{bmatrix}
  = \begin{bmatrix}
  k_{px} q_e1 \\
  k_{py} q_e2 \\
  k_{pz} q_e3
  \end{bmatrix}
  + \begin{bmatrix}
  k_{dx} \omega_x \\
  k_{dy} \omega_y \\
  k_{dz} \omega_z
  \end{bmatrix}
  \]
  - Controller does not take wheel angular momentum into account
  - Can result in instability
  - Controller gains dependent on satellite inertia properties
  - Separate gain value needed for each parameter
Instability of Direct PD Control when Angular Momentum is Large
**Inverse Dynamics PD Controller Performance**

- Desired angular accelerations achieved perfectly
- Performance independent of initial conditions and system inertia properties
- System exhibits a stable oscillatory equilibrium
  - Frequency and amplitude of oscillations decreases as system becomes more overdamped
  - Adding integral term does not mitigate oscillations
  - More work needed to determine exact cause for oscillations
Results
\[ \overrightarrow{\omega_0} = (0, 0, 0) \]

Critically Damped Case (\(K_p=1, k_d=-1.4\))

Underdamped Case (\(K_p=1, k_d=-0.5\))

Overdamped Case (\(K_p=1, k_d=-4\))
Simulations with $K_p = 1$ and $K_d = -4$

\[
\vec{\omega}_0 = (0, 0, 0)
\]

\[
\vec{\omega}_0 = (.5, .1, .7)
\]
Simulations with $\mathbf{\omega}_0 = (0.5, 0.1, 0.7)$

- $K_p = 1$
- $k_d = -8$

- $K_p = 1$
- $k_d = -20$
Minimizing Oscillations

- Damping ratio \( \zeta = \left| \frac{k_d}{2 \sqrt{k_p}} \right| \)

- Increase both \( k_d \) and \( k_p \) such that system is heavily overdamped
  - High damping minimizes oscillations
  - High \( k_p \) means system will get to desired orientation rapidly
Simulations with $K_p = 200$

- $K_d = -20$ and $\vec{\omega}_0 = (0, 0, 0)$

- $K_d = -500$ and $\vec{\omega}_0 = (0.5, 1, 0.7)$
Simulations with Different Gains

and \( \vec{\omega}_n = (0.5, 1, 0.7) \)
Active Momentum Dumping

• Magnetorquers constantly work to reduce wheel angular momentum ($H_w$)

\[ H_w = I_w \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \]

\[ \vec{\mu}_{id} = k^* (\vec{H}_w \times \vec{B}) \]

\[ \vec{T} = \vec{\mu}_{id} \times \vec{B} = k^* (\vec{H}_w \times \vec{B}) \times \vec{B} \]
Autonomous In-Flight Calibration

• Estimate moments of inertia and angular velocity based on kinematic response

\[
\sum M = \frac{d\bar{H}}{dt} + \bar{\omega}_{sc} \times \bar{H}
\]

– Calibrate to correct bias error in rate gyros.

\[
I_s \alpha_s = -I_w \alpha_w
\]

\[
\omega_x = \frac{I_w \omega_y \omega_1 - I_w \omega_3 - I_{zz} \omega_z}{(I_{yy} - I_{xx}) \omega_y + I_w \omega_3}
\]

\[
\omega_z = \frac{I_w \omega_x \omega_3 + I_w \omega_2 - I_{yy} \omega_y}{(I_{xx} - I_{zz}) \omega_x + I_w \omega_1}
\]

\[
\omega_y = \frac{I_w \omega_z \omega_2 - I_w \omega_1 - I_{xx} \omega_x}{(I_{zz} - I_{yy}) \omega_z + I_w \omega_3}
\]
Estimated Cost

- Four Positions Sensitive Detectors - $1000 total
- Three Magnetorquers - $100 total
- Three Reaction Wheels - $300 total
- Electronic Speed Controller - $50
- Magnetometer - $100
- Rate gyros - $100
- Microcontroller and PCB - $200
- Total - $1850
Questions?

• Contact:
  • Sanny Omar
    – AubieSat Team
• sanny.omar@gmail.com