

## A Low-Cost Attitude Determination and Control System and Hardware-in-the-Loop Testbed for CubeSats

Benjamin Jensen, Zachary Manchester  
 Carnegie Mellon University  
 5000 Forbes Avenue, Pittsburgh, PA, 15213; 801-680-5635  
 bjensen@andrew.cmu.edu

### ABSTRACT

**[Note: This is the submitted abstract and should be updated]** The attitude determination and control system (ADCS) for a satellite is responsible for multiple key roles in a satellite’s mission, including detumbling the satellite after deployment, pointing payload sensors, and orienting antennas and solar panels for effective communication and power generation. Designing an effective ADCS is crucial to a mission’s success; however, current methods often rely on actuators and sensors that are bulky and expensive, such as reaction wheels and star trackers. While these systems can provide high accuracy, they often cannot be used on CubeSats due to volume, weight, and cost restrictions.

This work builds upon PyCubed, a radiation-tolerant avionics platform for CubeSats that is programmable entirely in Python, by adding a low-cost, open-source attitude determination and control system that is scalable to smaller spacecraft like 1U CubeSats. This system relies on simple consumer-grade magnetometers, gyroscopes, and sun sensors to estimate the orientation of the satellite, along with a set of magnetic torque coils for actuation. By combining these low-cost sensors and actuators with sophisticated calibration, estimation, motion planning, and control software, we are able to achieve full three-axis attitude determination and control. The system is also completely solid-state, with no moving parts or need for consumable propellant, greatly reducing the chance of hardware failure.

To further improve the development cycle and increase success rates for CubeSat missions, we have also developed an open-source hardware-in-the-loop simulator to enable rapid testing of ADCS algorithms and other flight software. The result is a robust, open-source development suite for CubeSats that is low cost, easy to program, and reliable.

### Introduction

The development of CubeSats—a particular class of nanosatellite created in 1999 by California Polytechnic State University and Stanford University—has led to an increased amount of focus in space research and engineering. Due to their relatively low cost and short developmental cycle, these satellites have become popular for use in technology demonstrations, scientific research, and training missions for students and hobbyists, providing unprecedented access to space. These satellites have been used in a variety of missions, from gathering atmospheric measurements to imaging Earth and tracking migration patterns [cite?]. Although these satellites have seen high levels of success, they also suffer from a high percentage of mission failures; according to one study, only around 15%-40% of CubeSat missions are fully successful.<sup>1</sup> The same study suggests that primary reasons for failing missions include software instability, environmental ware, and developmental processes that are too short and with

insufficient testing.

The attitude determination and control system (ADCS) of a satellite plays a key role in the success or failure of a mission. This system is responsible for orienting the satellite into a desired attitude, and is crucial to a satellite’s ability to communicate with a ground station, as well as to orient any payload or onboard sensor into the appropriate location to accomplish mission objectives. Due to the high importance of the ADCS, many different systems have been designed that are incredibly effective and robust, and these systems have been consistently effective on a variety of missions. Unfortunately, these methods rely on highly-optimized equipment, such as star-trackers and reaction wheels, which are often too large for use on nano and pico-class satellites. Commercial systems developed specifically for CubeSats exist,<sup>2,3</sup> but at a cost that precludes their use in projects developed by groups with tight cost constraints, such as university research groups or clubs. This has required universities interested in CubeSat developments to either invest a large amount of

money in each launched satellite, or develop their own systems, making it difficult to test and develop new projects.

Although CubeSats and other small satellites often tout a development process that is relatively cheap and fast when compared to other satellites, the process still requires a lot of time and resources. Failures in the hardware or the software of an ADCS can cause the entire mission to fail; because these satellites are not recovered after launch for reuse, these failures require a complete restart on the deployment process, during which time new errors and problems may develop.

To overcome these challenges, we propose a low-cost ADCS that utilizes commercial off-the-shelf sensors and is scalable to smaller spacecraft. This project builds on the work done for PyCubed, which is a radiation-resistant avionics platform for small satellites, and relies on simple magnetometers, gyroscopes, and sun sensors for attitude determination, as well as a set of magnetic torque coils for control.<sup>4</sup> Although a control system based exclusively on magnetic torquers faces challenges due to underactuation, recent work has demonstrated full three-axis attitude control in presence of a time-varying magnetic field.<sup>5</sup> The resulting system is a completely solid-state ADCS, without any moving parts or consumable fuel, greatly simplifying the dynamics model and reducing the chance of hardware failure. Additionally, we have developed an open-source hardware-in-the-loop (HITL) simulator that can be created and assembled quickly and at a low cost. This system allows on-board electronics to be used when testing various flight software, improving a developer's ability to detect both faulty software and faulty hardware before launch.

## Satellite Hardware

The satellite hardware is built on PyCubed, an open-source complete avionics stack that was tested on KickSat-2.<sup>4</sup> This avionics stack was designed for reliability and ease of use, with the goal of increasing the mission success rate for first-time CubeSat developers. The onboard sensors include a BMX160 IMU, which integrates an accelerometer, a gyroscope, and a magnetometer, as well as six TSL2560 photodiodes. Additionally, we assume that the satellite has some way to estimate its position in space, either with a GPS or with information uploaded from a ground station. The magnetometer and photodiodes allow for estimation of the magnetic field and sun vectors in the body frame of the CubeSat, while the position allows for estimation of both vectors in

inertial frame, allowing for attitude estimation.

Additional onboard hardware includes solar panels for power generation and a radio for communication, as well as magnetic torquer coils embedded in each side panel for control. Because magnetic torque systems are inherently underactuated, they are traditionally used on satellites alongside other systems, like reaction wheels, which offer large amounts of torque but must be desaturated periodically. However, in the presence of a time-varying magnetic field, magnetorquer-only control is possible.<sup>5</sup> The resulting control system is less weight, has lower power requirements, and is more affordable than one relying on reaction wheels, and the lack of moving parts makes the system as a whole less prone to breaking.



Figure 1: **PLACE HOLDER!** Assembled CubeSat.

- FIG: Cost, weight, volume percentages of ADCS for existing systems; IF I can find enough info...

## Software Implementation

### Dynamics

[**Note:** I still need to find a copy of the book and verify all these equations] For a given satellite, the orbit dynamics can be modelled using the position of a satellite  $r$ , its velocity  $v$ , and acceleration  $a$ , all expressed in Earth-centered Earth-fixed (ECEF)

frame, which has its origin at the center of Earth's mass and rotates with the Earth.

There are a wide variety of factors that go into calculating the acceleration  $a$ . Although Earth's gravitational field is often approximated as uniform for simplicity, in reality the non-uniformities in the Earth contribute to a much more complicated gravitational field. This results in orbital trajectories that cannot be captured with a simplified model using uniform gravity, such as sun-synchronous orbits. In order to account for this behavior in simulation, the gravitational field is approximated using spherical harmonics, where each successive term accounts for higher-order factors contributing to  $a_g$ , the acceleration due to gravity.<sup>6</sup>

Additionally, because CubeSats and other nanosatellites are often flown in low Earth orbits (LEO), atmospheric drag can lead to significant trajectory differences, especially as it accumulates over multiple orbits. This drag can be estimated as

$$a_d = -\frac{C_d A \rho \|v\| v}{2m}, \quad (1)$$

where  $C_d$  is the coefficient of drag,  $m$  is the satellite mass,  $A$  is the cross-sectional area of the satellite, and  $\rho$  is the local atmospheric density, which is estimated using the Harris-Priester density model.<sup>6</sup>

Direct solar radiative pressure (SRP) also contributes to the acceleration of a spacecraft, and can be approximated as

$$a_{srp} = \frac{d C_r A r_{sun} * AU^2}{\|d\|^3}, \quad (2)$$

where  $d$  is the vector from the satellite to the sun,  $C_r$  is the coefficient of reflectivity,  $A$  is the cross-sectional area of the satellite facing the sun,  $r_{sun}$  is the position of the sun, and  $AU$  is the astronomical unit.<sup>6</sup>

Finally, the effect of the gravitational fields of both the moon and the sun are modeled as

$$a_M = -\frac{Gr_m}{\|r_m\|^3}, \quad (3)$$

$$a_S = -\frac{Gr_s}{\|r_s\|^3}, \quad (4)$$

where  $Gr_m$  and  $Gr_s$  are the gravitational coefficients of the moon and sun, respectively, and  $r_m$  and  $r_s$  are the distances between the satellite and the moon and sun.

The resulting acceleration  $a$  is then a sum of each

contributing term, so that

$$a = a_g + a_d + a_{srp} + a_M + a_S. \quad (5)$$

Note that in the simulation,  $a_{srp}$  and  $a_d$  both are attitude independent and assume a constant cross-sectional area  $A$  for simplicity.

The attitude dynamics of the satellite can be modeled using Euler's equation

$$\dot{\omega} = J^{-1}(\tau - \omega \times J\omega), \quad (6)$$

where  $\tau$  is the sum of the applied torques,  $J$  is the inertia matrix of the satellite, and  $\omega$  is the angular velocity. There are many ways to represent attitude, but we have selected unit quaternions for use in the simulator. This parameterization was chosen to avoid issues caused by singularities inherent to least-parameter representations (e.g., Euler angles, axis-angle, etc.) and more complicated constraints required by rotation matrices. However, this does require a mapping of the three-parameter angular velocity  $\omega$  to the four-parameter unit quaternion  $q$ . The relationship can be expressed as

$$\dot{q} = \frac{1}{2} L(q) H \omega, \quad (7)$$

where  $H$  converts  $\omega$  to a zero-scalar quaternion and

$$L(q) = \begin{bmatrix} q_s & -q_v^T \\ q_v & q_s I + [q_v]^\times \end{bmatrix} \quad (8)$$

with  $q_s$  and  $q_v$  representing the scalar and vector portions of the quaternion, respectively, and  $[v]^\times$  representing the skew-symmetric matrix formed by  $v$ .<sup>7</sup>

Finally, the bias  $\beta_\omega$  of the gyroscope is updated as a random walk for later use in updating sensor measurements, with its dynamics expressed as

$$\dot{\beta}_\omega \sim N(0, \sigma). \quad (9)$$

The resulting state vector for the simulator environment is then

$$x = [r \quad v \quad q \quad \omega \quad \beta_\omega]^T. \quad (10)$$

### Measurement Generation

In addition to simulating the orbit of a satellite, the simulator also generates data for each of the different sensor types onboard the CubeSat, which are based off of the environment state.

First, the magnetic field vectors in the body frame are generated. The position of the satellite is fed into the 13th generation International Geomag-

netic Reference Field (IGRF13)<sup>8</sup> model to generate the magnetic field vector in the inertial frame  $B^I$ . The attitude of the satellite is then used to convert the magnetic field into the body frame,  $B^B$ . Because sensors are imperfect, the measured magnetic field vector is not the same as the true vector. Magnetometer imperfections can be grouped into scale factors, nonorthogonality angles between each axes, and bias along each axis (see Section for more information). This must be accounted for when generating measurements, so that the measured magnetic field vector in the body frame  $\tilde{B}^B$  is

$$\tilde{B}^B = \eta_B T B^B + \beta_B, \quad (11)$$

where  $T$  is the matrix that converts a perfect magnetometer reading into an imperfect one in need of calibration,  $\beta_B$  is the magnetometer bias, and  $\eta_B$  represents **multiplicative/rotational?** Gaussian noise.

Second, the measured gyroscope vector  $\tilde{\omega}$  is generated from the true gyroscope reading  $\omega$  as

$$\tilde{\omega} = \omega + \beta_\omega + \eta_\omega, \quad (12)$$

with  $\eta_\omega$  as white Gaussian noise.

The current generated from each photodiode is a function of the incoming light, the surface normal of the photodiode, and the scale factor for the photodiode. In addition to the light coming directly from the sun, there is also light coming from Earth's albedo; this additional light source depends on the location of the sun and the satellite, and its effect can up to 30% – 4 – %<sup>9</sup> of the experienced solar irradiance. To account for this, we model the effect of Earth's albedo after the method proposed by Bhandari,<sup>10</sup> which involves dividing the surface of the Earth into a series of cells. The reflectivity of each of these cells is determined by averaging data gathered by the NASA TOMS mission over several years.<sup>11</sup> This reflectivity data is used for all cells that are in the field-of-view of both the sun and satellite to estimate the amount of light that would be reflected by the Earth's atmosphere onto the satellite.

The total amount of current  $I$  produced by the  $j$ th diode can then be computed as

$$I_j = \hat{n}_j^T \hat{s}^B + \frac{E_{a,j}}{E_{AM0}}, \quad (13)$$

where  $\hat{n}_j$  is the unit surface normal of the  $j$ th diode,  $\hat{s}^B$  is the unit vector in the direction of the sun expressed in the body frame,  $E_{AM0}$  is the irradiance of sunlight at 1AU and with no loss due to atmosphere, and the Earth's albedo  $E_{a,j}$  is a function of

both satellite position and attitude.<sup>9</sup> To generate the measured current  $\tilde{I}_j$ , the true value is scaled by a scale factor  $C_j$  and noise is added, so that

$$\tilde{I}_j = C_j I_j + \eta_I \quad (14)$$

Finally, the position of the satellite is perturbed to add in some noise, so that

$$\tilde{r} = r + \eta_r. \quad (15)$$

### MEKF?

[**Note:** Not sure if I should explain the MEKF or not]

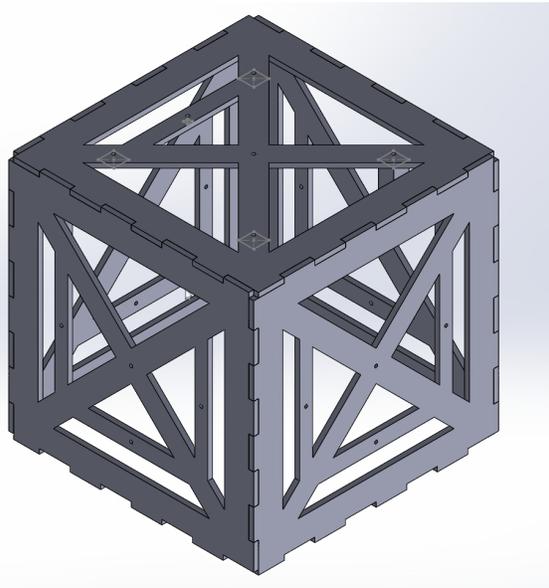
### Hardware-in-the-Loop Testbed

Simulators help with conceptual and algorithmic development, but to truly validate the satellite hardware and detect errors before launch, a hardware-in-the-loop testbed is necessary. While there currently exist hardware-in-the-loop testbeds for CubeSats, these rely on air-bearing tables designed to create a low-torque environment,<sup>12</sup> sometimes with the addition of a fixed light source to simulate the sun;<sup>13</sup> our testbed takes a different approach. We propose a HITL test box that the CubeSat (or other smallsat) can be placed into for testing. Each panel contains an LED with an adjustable level of light that can be used to simulate the sun vector. Rather than relying on the air-bearing table to allow the satellite to rotate, we generate the body-frame sun vector and use the LEDs to illuminate the appropriate sides of the satellite at the appropriate proportions. Additionally, each panel has a mounted magnetometer to measure the magnetic field generated by pulsing the magnetic torquer coils. This box is placed inside of a Helmholtz coil to cancel out the geomagnetic field and replace it with a desired magnetic field.

This system allows for the testing of various sensors in isolation (e.g., running the LEDs to validate that the sun-vector estimation system works), or for the integration of the satellite hardware into a simulator (e.g., feeding the measured magnetic fields created by the magnetic torquer coils back into the simulator dynamics), allowing for the detection of faulty hardware or algorithms before launch. Note that the current setup does not allow for testing of the satellite gyroscope.

The box itself is made out of laser-cut wood or acrylic and the mounts are all made inside of a 3D printer. Additionally, the system is controlled with an Arduino and uses only off-the-shelf LEDs and magnetometers. The result is a cheap-to-make and

easy-to-assemble test station that can be built in-house and requires only common materials that are easy to acquire.



**Figure 2: PLACE HOLDER! Assembled hardware-in-the-loop test box.**

## Sensor Calibration

Due to manufacturing and installation errors, sensors often need to be calibrated in order to maximize their use; this is particularly true for more affordable versions of off-the-shelf sensors. For our system, the onboard sensors include a magnetometer, the sun sensors, and the gyroscope. Although the onboard gyroscope experiences a bias that must be accounted for, because it is time-varying it is estimated as part of the state using a multiplicative extended Kalman filter (MEKF). The calibration process for the remaining sensors are described in their respective sections below.

### Magnetometer Calibration

**[Note: This section is kinda just a rehash of the Springmann/Cutler paper]** Magnetometers are popular sensors in attitude determination because they are lightweight, have low power requirements, and involve no moving parts. However, these sensors can be very noisy, particularly for lower-cost models, and this noise can result in poor-quality attitude estimation for the satellite. There are several factors that contribute to this error, including installation error and corruption from soft-iron and hard-iron metals. The effects of these factors are often

grouped into three error categories: scale factors, nonorthogonality angles, and bias. Because these errors are time-invariant, they can be determined in advance and their effects can be corrected out of a magnetometer’s measurement; this allows for more accurate measurements and reduces the uncertainty of off-the-shelf magnetometers.

The measured magnetic field ( $\tilde{B}_x^B, \tilde{B}_y^B, \tilde{B}_z^B$ ) can be modeled as:

$$\tilde{B}_x^B = aB_x + x_0 + \eta_x \quad (16)$$

$$\tilde{B}_y^B = b(B_x \sin \rho + B_y \cos \rho) + y_0 + \eta_y \quad (17)$$

$$\begin{aligned} \tilde{B}_z^B = c(B_x \cos \lambda + B_y \cos \lambda \sin \phi + B_z \cos \lambda \cos \phi) \\ + z_0 + \eta_z, \end{aligned} \quad (18)$$

where  $a$ ,  $b$ , and  $c$  are the scale factors along the  $x$ ,  $y$ , and  $z$  axes;  $x_0$ ,  $y_0$ , and  $z_0$  are the bias terms along each axis;  $\rho$  is the nonorthogonality angle between a  $y$ -axis orthogonal to  $x$  and the measured  $\tilde{y}$ ;  $\lambda$  is the nonorthogonality angle between a  $z$ -axis orthogonal to  $x$  and the measured  $\tilde{z}$ ; and  $\phi$  is the nonorthogonality angle between a  $z$ -axis orthogonal to  $y$  and the measured  $\tilde{z}$ .<sup>14</sup> Additional measurement noise along each axis is included as  $\eta_x$ ,  $\eta_y$ , and  $\eta_z$

Because the magnetometers are used in attitude determination, it is helpful to have an attitude-independent calibration method. This is possible because the magnetometer errors affect the magnitude of a measurement, but a rotation between two frames does not; as such, techniques that rely only on the magnitude of a magnetometer measurement are effective without any need for information about attitude, as shown by Foster<sup>14</sup> and Springmann.<sup>15</sup> This method relies on batch estimation and nonlinear least squares minimization is performed using Gauss-Newton. This minimizes a cost function

$$J = \frac{1}{2} \left[ B_E^2 - f(\tilde{B}^B, x) \right]^T \left[ B_E^2 - f(\tilde{B}^B, x) \right], \quad (19)$$

where  $B_E$  is a vector of the magnitudes of the expected magnetic field vectors at each time step,  $\tilde{B}^B$  is a vector of the measured magnetic field vectors in body frame,  $x$  is the current guess for the calibration parameters, and  $f(\tilde{B}^B, x)$  provides the magnitude of the measured vectors after correction by the calibration parameters in  $x$ . Note that, while attitude-independent, this method does require a time-varying magnetic field.

## Diode Calibration

[Note: This section is also kinda just a rehash of the Springmann/Cutler paper]

Photodiodes generate a current when illuminated that is proportional to the angle between the photodiode surface normal and the vector in the direction of the sun. This angle can be used to provide a component of the sun vector in the body frame of the satellite. When multiple photodiodes are used, the full sun vector can be estimated. However, effective estimation requires accurate surface normals for each photodiode, as well as the scale factor for each photodiode so as to normalize the measured currents. There are many methods for calibrating photodiodes, including attitude-independent methods such as; however using a recursive, attitude-dependent method such as that proposed by Springmann<sup>9</sup> allows for arbitrary numbers and configurations of photodiodes. Additionally, the attitude-dependent method allows for the inclusion of Earth’s albedo described above. This calibration is done on-orbit to account for changes that may occur during launch, and allows for periodic re-estimation, as scale factors may degrade over time due to radiation.

Calibration is done by augmenting the state of a traditional MEKF—which estimates attitude and gyroscope bias—with additional states to track diode calibration value and surface normal. Because each surface normal is constrained to the surface of a unit sphere, they can be parameterized in terms of an elevation angle  $\epsilon$  and azimuth angle  $\alpha$ , so that the state to be estimated at each time step in the augmented MEKF is

$$x = [q \quad \beta \quad \mathbf{C} \quad \boldsymbol{\alpha} \quad \boldsymbol{\epsilon}]^T, \quad (20)$$

where  $q$  represents the attitude (as a unit quaternion, in our case),  $\beta$  is the gyroscope bias,  $\mathbf{C}$  is the vector of the scale factor for each diode, and  $\boldsymbol{\alpha}, \boldsymbol{\epsilon}$  are the vectors of each diode’s surface normal, represented as azimuth and elevation angles. At each step of the MEKF, the expected currents (predicted using Eq. 13) are compared to the measured currents (generated using Eq. 14). This information is used to iteratively update the estimate for each calibration value.

## Hardware Experiments

Demonstrate diode calibration:

- Simulator generates sun (w, w/o albedo), sun lights up, sat estimates. Track angular offset in estimate from true value

- Run diode calibration with a simplified ground truth, using ground truth for attitude and bias
- Run sun vector estimation again and show that diode calibration helped (hopefully)

**Table 1: Error in sun vector estimation before and after sun sensor calibration.**

	Error ( $\mu$ )	Error ( $\sigma$ )
Before		
After		

(Can I finish the Helmholtz coil in time?)

## Conclusion

- Conclusion stuff
- Next steps (implement the controller, build the Helmholtz coils, port it all to python, generate performance results?)

## References

- [1] Michael Swartwout. Cubesat mission success: Are we getting better? In *CubeSat Developers’ Workshop*, 2019.
- [2] BlueCanyon Technologies. Components.
- [3] CubeSat Shop. Cube adcs.
- [4] Max Holliday, Andrea Ramirez, Connor Settle, Tane Tatum, Debbie Senesky, and Zac Manchester. Pycubed: An open-source, radiation-tested cubesat platform programmable entirely in python. In *AIAA/USU Conference on Small Satellites (SmallSat)*, 2019.
- [5] Andrew Gatherer and Zac Manchester. Magnetorquer-only attitude control of small satellites using trajectory optimization. In *Proceedings of AAS/AIAA Astrodynamics Specialist Conference*, August 2019.
- [6] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer, 2012.
- [7] Brian E. Jackson, Kevin Tracy, and Zachary Manchester. Planning with attitude. *IEEE Robotics and Automation Letters*, 6(3):5658–5664, 2021.
- [8] P. Alken, E. Thébaud, and C.D. et al. Beggan. International geomagnetic reference field: the thirteenth generation. *Earth Planets Space*, 74, 2021.

- [9] John C. Springmann and James W. Cutler. On-orbit calibration of photodiodes for attitude determination. *Journal of Guidance Control and Dynamics*, 37:1808–1823, 2014.
- [10] Dan D. V. Bhandari. *Spacecraft Attitude Determination with Earth Albedo Corrected SunSensor Measurements*. PhD thesis, Aalborg University, Fredrik Bajers Vej 7, DK-9220 Aalborg Ø, Denmark, 2005.
- [11] R. McPeters, P. K. Bhartia, A. J. Krueger, , and J. R. Herman. Earth probe total ozone mapping spectrometer (toms) data products user’s guide. *Technical Report No. 19987-206895, National Aeronautics and Space Administration*, 1998.
- [12] J. Prado, G. Bisiacchi, L. Reyes, E. Vicente, F. Contreras, M. Mesinas, , and A. Juárez. Three-axis air-bearing based platform for small satellite attitude determination and control simulation. *Journal of Applied Research and Technology*, 3(3):222–237, 2005.
- [13] F.Reichel, P.Bangert, S.Busch, K.Ravandoor, and K.Schilling. The attitude determination and control system of the picosatellite uwe-3. *IFAC Symposium on Automatic Control in Aerospace*, 46(19):271–276, 2013.
- [14] C. C. Foster and G. H. Elkaim. Extension of a two-step calibration methodology to include nonorthogonal sensor axes. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1070–1078, 2008.
- [15] John C. Springmann and James W. Cutler. Attitude-independent magnetometer calibration with time-varying bias. *Journal of Guidance Control and Dynamics*, 2012.