Bei Kang and Dr. Chang-Hee Won

Hybrid Control for Autonomous System

ABSTRACT

Hybrid systems are used to model and analyze the systems which have both discrete and continuous dynamics. Many modern systems such as spacecrafts systems and ships have hybrid dynamics. In this paper we utilized hybrid control theory to maneuver operation of spacecrafts. Furthermore, we investigated the hybrid time trajectory, execution, reachability, nonblocking, and deterministic properties to show the existence and uniqueness of the hybrid system. In our model, the spacecraft orbit is controlled using Hohmann transfer; the spacecraft attitude is controlled using linear quadratic regulator control which is optimal control method used in continuous systems. We performed simulations of autonomous control for a single spacecraft control case as well as multiple spacecraft control case using MATLAB, Simulink and Stateflow softwares. Consequently, we showed that the hybrid control theory is an efficient method to autonomously control spacecrafts.

INTRODUCTION

Traditional satellite attitude and orbit control methods require constant monitoring and intervention from the ground stations to adjust satellite attitude and orbital parameters. Even though traditional satellite attitude and orbit control is sufficiently effective for of satellites, it would be desirable to autonomously control the satellite attitude and orbit. This is especially useful as the number of satellites grows. In order to autonomously control the orbit and the attitude, we need to model the continuous dynamics as well as the discrete dynamics. Recent developments in discrete event systems and the hybrid automata theory allow us to mathematically model and analyze the whole process of command, attitude control, and orbit maneuver of the satellites. So, we develop a model for autonomous satellite control using hybrid automata. Attitude and orbit are modeled using the traditional attitude and orbital equations. We incorporate telecommands as the discrete events and different satellite operation modes as the discrete states. Then we perform simulations to autonomously change the orbit and attitude of the satellites. The results show that hybrid automata theory can be effectively used to automatically control both attitude and orbit for a satellite.

In a discrete event system (DES), an event occurs spontaneously and causes a transition from one node to another. The study of DES is dedicated to developing models which describe the behavior of these systems and the mathematical basis for analysis, design and control. For more general description of the discrete event systems, readers may refer the book by Cassandras and Lafortune (2001) and the paper by Ramadge and Wonham (1987). Automata form a class of DES classes; it is an untimed modeling formalism. Automata theory is developed to describe the boundary between what a computer can do and it cannot do. The hybrid automata theory, which is an active research area, has been studied and applied in many different fields. References Henzinger, Ho, and Wong (1997), Lygeros et al. (2003), and Lygeros, Godbole, and Sastry (1998) give the examples of different systems where hybrid automata have been applied. In hybrid automata theory, a language is developed to model and analyze both continuous and discrete dynamics of a hybrid system. Hybrid automata in aerospace applications are a relatively new concept, however, it is well suited for autonomous control of spacecrafts. Stadter (2000) used discrete event system idea to control a formation flying spacecraft. Pennecot, Atkins and Sanner (2002) have used hybrid automata based model for spacecraft formation management and path planning. Here the general autonomous orbit and attitude maneuver hybrid automation ideas are presented, but it does not have a precise mathematical model. In 2003, Karsai, Abdelwahed and Biswas (2003) integrated
diagnosis and control for hybrid dynamic systems using hybrid bond graphs. More recently, the discrete event system concepts were utilized in autonomous mission management for unmanned aerial vehicles by Barbier and Chanthery (2004).

In the next section, a hybrid automata theory is briefly reviewed. The satellite attitude and orbital perturbation models are presented in Section III. In Section IV, a sample satellite configuration and parameters are described for the simulation purposes. Section V discusses the automata for the three satellites achieving the target orbit with linear quadratic regulator (LQR) attitude control and Hohman orbit transfer. Then the hybrid automata for general satellite orbit and attitude control are discussed and simulated using MATLAB® and Simulink® in Section VI. Finally, the paper ends with the conclusions.

Hybrid Automata Theory

Hybrid systems can be modeled by hybrid automata, which can be described using the graph theory as a state transition diagram. The transitions between each state in the diagram are the discrete transitions, while the continuous dynamics are included in each state. Certain conditions that can be viewed as discrete events constitute the transitions, causing the system to move from one state to another. In hybrid automata theory, there are several important definitions and properties which we will review in this paper. We will state the definitions and theorems as given by Lygeros et al. (2003), for the sake of completeness. For more details and proofs see references by Henzinger (1996), Henzinger and Rusu (1998), Lygeros et al. (2003), Lygeros et al. (1999), and Lygeros, Tomlin and Sastry (2002).

Hybrid Automaton

A hybrid automaton $H$ can be described mathematically as,

$$ H = (Q, X, f, \text{Init}, D, E, G, R) $$

where

- $Q$: Finite set of discrete variables representing the discrete dynamics of $H$;
- $X$: Finite set of continuous variables;
- $f : Q \times X \rightarrow TX$: Vector field, defining the continuous flow in each discrete node;
- $\text{Init} \subseteq Q \times X$: Set of initial states;
- $D : Q \rightarrow P(X)$: Domain;
- $E \subseteq Q \times Q$: Set of edges;
- $G : E \rightarrow P(X)$: Guard condition;
- $R : E \times X \rightarrow P(X)$: Reset map.

The set $Q$ is simply the set of discrete states where the system is allowed to exist. The set $X$ will represent all of the continuous variables that are possible in each of the states of $Q$. The vector field, $f$, usually consists of time dependent functions such as differential equations that describe how each of the variables in $X$ changes over time. $TX$ denotes the tangent bundle of $X$. The initial states are simply the initial values of the continuous variables and the domain, $D$, contains the range where each continuous variable is allowed to exist. The domain is also called the invariant set. $P(X)$ denotes the set of all subsets of $X$. The set of edges, $E$, describe what transitions are allowed to occur between states. The guard conditions, $G$, describe the events that must occur for a transition to take place. Finally, the reset map, $R$, is the set of conditions that cause the system to enter its initial state.

Hybrid Time Trajectory

A hybrid time trajectory is a finite or infinite sequence of intervals $\tau = \{I_i\}_{i=0}^N$, such that

1) $I_i = [\tau_i, \tau_i']$, for all $i < N$;

2) if $N < \infty$, then either $I_N = [\tau_N, \tau_N']$ or $I_N = [\tau_N, \tau_N')$; and

3) $\tau_i \leq \tau_i' = \tau_{i+1}$ for all $i$.

We will use hybrid time trajectory to analyze time horizon of executions defined below.

Execution

An execution of a hybrid automata $H$ is a collection $\chi = (\tau, q, x)$, where $\tau$ is a hybrid time trajectory, $q : (\tau) \rightarrow Q$ is a map, and
\( x = \{ x' : i \in \langle \tau \rangle \} \) is a collection of differentiable maps \( x' : I_i \to X \), such that

1) \( (q^0, x^0) \in \text{Init}; \)

2) for all \( \tau \in [\tau, \tau'] \), \( x'(t) = f(q(i), x'(t)) \) and \( x'(t) \in D(q(i)) \);

3) for all \( i \in \{ \tau \} \setminus \{ N \} \),
\[ e = (q(i), q(i+1)) \in E, \]
\[ x'(\tau'_i) \in G(e), \quad \text{and} \quad x'^{(i)}(\tau_{i+1}) \in R(e, x'(\tau'_i)). \]

Here, \( \langle \tau \rangle \) is the set \{0, 1, 2, \ldots, N\} if \( N \) is finite and \{0, 1, 2, \ldots, \} if \( N = \infty \). In this paper, \( (q^0, x^0) \) denotes the initial state. We use \( \epsilon \) to denote the set of executions with the initial condition \( (q^0, x^0) \in \text{Init} \), \( E_H \) to denote the maximum executions, \( E_H' \) to denote all the finite executions and \( E_H'' \) to denote the set of all infinite executions.

- **Reachability**

Reachability is a concept of hybrid systems which specifies that the certain state \( q \in Q \) can be reached in the hybrid automaton \( H \) through a finite number of executions. The reachability of a hybrid automaton can be expressed in the following equation:

\[
\tag{1} \text{Reach}_{H} = \left\{ (q, \hat{x}) \in Q \times X : \exists \left( \{ [\tau, \tau'] \} \right)_{i=0}^{\infty} q, x \in E_H' \right\}.
\]

The set for which the continuous evaluation is impossible can be expressed as

\[
\text{Out}_{H} = \{ (q, x) \in Q \times X : \forall \epsilon > 0, \exists \tau \in [0, \epsilon), \psi(q, x, t) \notin D(q) \}.
\]

where \( \psi(q, x, t) \) is a solution of \( \dot{x} = f(q(i), x(t)) \) for \( x(0) = x^0 \).

- **Non-Blocking and Deterministic**

The non-blocking and deterministic properties are two important criteria for analyzing the existence and uniqueness of the executions of hybrid automata. The non-blocking property implies that executions exist for all initial states, while the deterministic property implies that the infinite executions are unique as showed by Lygeros, Tomlin and Sastry (2002).

A hybrid automaton is non-blocking if \( E_H''(q^0, x^0) \) is non-empty for all \( (q^0, x^0) \in \text{Init} \). The following lemma can be used to determine the non-blocking property of a hybrid automaton.

**Lemma I:** A hybrid automaton \( H \) is non-blocking if for all \( (q, x) \in \text{Reach}_H \cap \text{Out}_H \), there exists \( (q', q') \in E \) so that \( x \in G(q, q') \).

A hybrid automaton, \( H \), is called deterministic if \( E_H''(q^0, x^0) \) contains at most one element for all \( (q^0, x^0) \in \text{Init} \). Also, there is a lemma to examine the deterministic property of \( H \).

**Lemma II:** A hybrid automaton \( H \) is deterministic if and only if for all \( (q, x) \in \text{Reach}_H \),

1) if \( x \in G(q, q') \) for some \( (q, q') \in E \), then \( (q, x) \in \text{Out}_H \);
2) if \( (q, q') \in E \) and \( (q, q') \in E \) with \( q' \neq q' \), then \( x \notin G(q, q') \cap \text{Out}_H \);
3) if \( (q, q') \in E \) and \( x \in G(q, q') \), then \( R(q, q', x) \) contains at most one element.

- **Existence and uniqueness:**

Now, we present a theorem to determine the existence and uniqueness of the hybrid automaton, \( H \).

**Theorem I:** If a hybrid automaton, \( H \), satisfies Lemma I and Lemma II above, then \( H \) accepts a unique infinite execution for all \( (q^0, x^0) \in \text{Init}_H \).

In the following sections, we will utilize these definitions and theorems to characterize a hybrid system.

**Attitude and Orbital Perturbations Model**

The attitude and orbit model of a low-earth-orbit (LEO) satellite is presented in this section. As for the attitude model, we use the deterministic version of the model developed by Won (1999).

The satellite attitude model for the Euler angle, angular velocity, and reaction wheel speed vector is given as,
\[ \begin{align*}
\dot{\Theta} &= \frac{1}{\cos \psi} \begin{bmatrix}
\cos \psi & -\cos \phi \sin \psi & \sin \phi \sin \psi \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \cos \psi & \cos \phi \cos \psi
\end{bmatrix} \begin{bmatrix}
0 \\
\phi' \\
0
\end{bmatrix} + \omega \times (l_t \omega + L' \omega \Omega) - L' \omega_w + T_{\text{thrust}} + 3n^2 \xi \Omega \times l_t \Omega \\
\dot{\Omega} &= l' \omega_w - l \omega
\end{align*} \]

where the superscript, \( T \), denotes transpose, \( I_t \) is the total moment of inertia for the satellite body, \( I_w \) is the moment of inertia matrix for the wheels, \( I_g = I_t - L' I_w L \) is the total moment of inertia minus the moment of inertia of the wheels, \( L \) is the wheel orientation matrix, \( \Theta = [\phi, \theta, \psi] \) is the roll, pitch, yaw Euler angles, \( \omega \) is the angular velocity vector in body fixed coordinate system, \( \Omega \) is the wheel speed vector, \( \omega_w \) is the absolute torque due to the reaction wheels, \( T_{\text{thrust}} \) is the torque due to the thrusters, \( n \) is the orbital rate, \( C^2 \) represents the cross product matrix of a vector \( C \).

\[
[x \; y \; z]^T_{BFC} = \begin{bmatrix}
\cos \psi \cos \theta & \sin \psi & -\cos \psi \sin \theta \\
-\cos \phi \sin \psi \cos \theta + \sin \phi \sin \theta & \cos \phi \cos \psi & \cos \phi \sin \theta + \sin \phi \cos \theta \\
\sin \psi \cos \theta & -\sin \phi \sin \psi \sin \theta + \cos \phi \cos \theta & \cos \phi \cos \psi
\end{bmatrix} \begin{bmatrix}
x_L \\
y_L \\
z_L
\end{bmatrix}_{LVLH} = \begin{bmatrix}
x_1 \\
y_2 \\
z_3
\end{bmatrix} \begin{bmatrix}
x_L \\
y_L \\
z_L
\end{bmatrix}_{LVLH}
\]

where \( BFC \) stands for body-fixed-coordinate and \( LVLH \) stands for local-vertical-local-horizontal coordinate system. In LVLH model, \( x \) is the direction of the velocity vector, \( z \) is pointing towards the Earth, and \( y \) completes the triad. The above model is linearized and utilized as the attitude dynamics model as showed by Won (1999). We note that we are ignoring the magnetic and the aerodynamic torques.

As for the orbital perturbation model, we incorporate atmospheric drag and Earth oblateness (J2) effects only. We assume other effects are negligible. The following orbit decay rate due to the atmospheric drag is used in the book by Wertz (1999). This drag force will affect the velocity in \(-x\) direction only.

\[ F_{\text{drag}} = -2\pi(C_d A/m) \rho \frac{r^2}{p} \]

where \( \rho \) is the atmospheric density at a particular point in the solar cycle, \( C_d \) is the satellite ballistic coefficient, \( A \) is the satellite cross-sectional area, \( m \) is the satellite mass, \( r \) is the Earth radius plus altitude, and \( P \) is the orbital period.

Earth oblateness will be the other perturbing effect that will be included in the orbit model. This perturbing force due to J2 on orbit altitude can be represented as described by Chobotov (1996),

\[
\begin{align*}
F_x &= -\frac{3}{2} \frac{\mu J_2 R^2 E}{r^4} \left[ \sin^2 i \sin(2u) \right] \\
F_y &= -\frac{3}{2} \frac{\mu J_2 R^2}{r^4} \left[ \sin i \cos i \sin u \right] \\
F_z &= -\frac{3}{2} \frac{\mu J_2 R^2}{r^4} \left[ 1 - \frac{3}{2} \sin^2 i \cos 2u \right]
\end{align*}
\]

where \( u \) is the argument of latitude (the sum of true anomaly \( u \) and argument of perigee \( \nu \)), \( J_2 \) is taken as 0.00108263, \( i \) is the inclination of the orbit, \( \mu \) is the gravitational parameter of the Earth, \( R_E \) is the equatorial radius of the Earth, and \( r \) is the Earth radius plus the altitude, \( h \), of the satellite. The total altitude decay rate, \( F_0 \), for the satellite is the sum of (4) and (5). To simplify the orbital maneuver, the Hohmann transfer orbital maneuver will be used to control the orbit.

**Satellite Configurations for Simulations**

Orbit and attitude control mechanisms of spacecrafts using hybrid automata are addressed in this section. Matlab, Simulink, and Stateflow are the simulation tools that were used to investigate the satellite orbit and attitude control problem. Stateflow is an application that allows the user to create the discrete transitions of the discrete event systems while Simulink is used to model the continuous dynamics. Matlab allows
the interfacing of each of these applications. This model demonstrates the use of hybrid automata to control satellite attitude and orbit. We assume that satellite altitude is controlled with coplanar orbit changes preceded and followed by changes in the attitude from LVLH coordinates.

In all the simulations, we will assume a small satellite with thrusters on one side of the spacecraft. In order to change the altitude, the satellite has to change its attitude to fire the thrusters. Figure 1 shows the Earth Pointing Mode at the right with the camera pointing towards the earth and the Thruster Firing Mode with the thrusters aligned with the direction of the satellite velocity. Figure 1 also exhibits the difference between BFC and LVLH coordinates.

The parameters of the Korea Multipurpose Satellite (Kompsat), which is a LEO remote sensing satellite, are utilized for the simulation purposes. The attitude parameters, for example, \( n = 0.0010636 \), associated with this satellite are given by Won (1999). We develop a hybrid automata model using the LQR attitude control method, and the Hohmann transfer orbit control. The idea here is to have the control system perform a re-boost when the altitude decays to the value outside of an altitude requirement. This re-boost would require the satellite to change attitude to align the re-boost thrusters with the direction of the velocity vector. The schematic diagram of the autonomous orbit and attitude control is given in Figure 2. For autonomous control of a group of satellites, we need to incorporate the command, attitude control, and orbit maneuver. And hybrid automata theory provides the mathematical framework for modeling systems with both discrete and continuous dynamics. In the following sections we will develop the approximate hybrid automata models.

FIGURE 1.  Left: Thruster Firing Mode, Right: Earth Pointing Mode.
Hybrid Automata for Satellite Orbit and Attitude Control

We now apply the hybrid automata theory to a satellite orbit and attitude control scenario. In this scenario, we assume that a satellite is in an initial orbit with the initial altitude, and with an initial attitude. Satellite will perform the orbit maneuver and attitude adjustment according to the input conditions which include Target Attitude (roll, pitch and yaw angles), Target Altitude. The automata mechanism enables the satellite to implement the transfer from initial insertion orbit and initial orbit plane to target orbit and target plane with target attitude automatically. Here the Hohmann transfer is utilized for the orbit change of the satellite. As for the attitude control method, we use linear quadratic regulator (LQR) optimal control method.

In this model, the attitude adjustment mechanism can be divided into two states: Attitude Calculation Node and Attitude Control Node. The Attitude Control Node can control the attitude of satellite to any angles of roll, pitch and yaw, which are calculated within the Attitude Calculation Node, according to the different input conditions. Also, we construct the general orbit control state as Orbit Maneuver Node in which the satellite can perform the Hohmann orbit transfer to change its orbit. In this state, the satellite will first check the target altitude to decide whether it is necessary to perform orbit transfer with different initial altitudes and target altitudes. If it is necessary to change the orbit of the satellite, then the satellite will fire the thrusters to perform the Hohmann orbit transfer to reach the target orbit. Here we assume that all nodes know the target orbit and target attitude information so that the we do not have to input target information to the system during the operation. The state transition diagram for the general satellite attitude and orbit control is given in Figure 3.

For the satellite hybrid automation, the set of discrete nodes can be defined into set $Q$ as $Q = \{q_1, q_2, q_3, q_4\}$, which represents the following satellite working nodes:

- $q_1$: Initial Node. In this node, the satellite is at the initial orbit with its initial altitude, initial attitude. The satellite will check the status of Orbit-Attitude Command to see if there is a requirement to do the orbit. If Orbit-Attitude Command is true, the satellite will jump to the $q_2$ and determine whether an orbit maneuver is necessary or not.
- $q_2$: Orbit Maneuver Node. In this node the satellite changes its altitude to the target orbit determined by the input condition. It will first check the current altitude with the target altitude, when there is a difference between the current and the target, then it will check the current attitude to see whether the satellite is in the thruster firing mode, if not, it will jump into $q_3$ to change its attitude to the thruster firing mode to boost the satellite to the target altitude. If the satellite is already in the thruster firing mode, satellite will fire the thrusters to change the orbit of the satellite. If the altitude match the target orbit requirement, the satellite will go to $q_3$ for the target attitude calculation directly. The node $q_2$ is triggered by Orbit-Attitude Command from $q_1$ or from $q_4$, the Attitude Adjustment Node.
- $q_3$: Attitude Calculation Node. In this node, the satellite calculates and outputs the different Euler angles, which are needed by $q_4$ to implement the attitude change, according to the input conditions. For example, if the satellite goes from $q_2$ to $q_3$ to implement an target attitude adjustment, $q_3$ node will calculate the roll, pitch and yaw angles used in $q_4$ node to adjust the attitude to the target; if the satellite carries out orbit change due to the Orbit-Attitude Command, $q_3$ node will calculate the corresponding roll, pitch and yaw angles for the thruster firing.
• \( q_4 \): Attitude Adjustment Node. In this node, the satellite adjusts its attitude to the input Euler angles calculated by Attitude Calculation Node for the satellite orbit and attitude change. When the current attitude reaches the calculated attitude from \( q_3 \), it will check the current altitude with the target altitude, if the current altitude does not match the target, the satellite will jump to \( q_2 \) node again to implement the orbit maneuver otherwise it will jump into \( q_5 \) satellite operation node.

• \( q_5 \): Satellite Operation Node. In this node the satellite operates at the final orbit with a target attitude of roll, pitch, and yaw angles. The satellite will check the current attitude and the current altitude, if the attitude of satellite drops below the target attitude or the current attitude is different from the target attitude, it will jump to the \( q_2 \) node and the orbit maintenance procedure will take in effect to maintain the orbit and the attitude of the satellite.

\[
f(q_1, x) = \left[ \begin{array}{c} \gamma \\ \omega_x \\ \delta \omega_y \\ \omega_z \\ \phi \\ \theta \\ \psi \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \\ \end{array} \right]^T
\]

\[
f(q_2, x) = \left[ \begin{array}{c} a(1 - e^2) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \end{array} \right]^T
\]

\[
f(q_3, x) = \left[ \begin{array}{c} h - F \times \Delta t \\ \left( \phi \times 9.2334e-7 + \omega_y \times 2.8937e-4 + \omega_y \times 2.1753e-8 \times \Omega_2 \times \right) \\ 2.1753e-8 \times \Omega_2 \times 2.1753e-8 \times \Omega_4 \times 2.1753e-8 \\ \left( \theta \times 2.2557e-6 \right) \\ \left( - \omega_x \times 8.3747e-4 \times \Omega_1 \times 3.0565e-8 \times \Omega_2 \times \right) \\ 3.0565e-8 \times \Omega_2 \times 3.0565e-8 \times \Omega_4 \times 3.0565e-8 \\ \left( \psi \times 1.0636e-3 + \omega_y \right) \\ \left( \delta \omega_y \right) \\ \left( - \phi \times 1.0636e-3 + \omega_x \right) \\ \int \left( - \phi \times 5.3309e-7 + \theta \times 1.285e-6 + \omega_x \times 4.8352e-4 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 5.0877e-9 - \Omega_2 \times 3.0205e-8 \times \Omega_3 \times \right) \\ \int \left( - \omega_x \times 8.3747e-4 \times \Omega_1 \times 3.0565e-8 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 3.0205e-8 \times \Omega_4 \times 5.0877e-9 \\ \phi \times 5.3309e-7 + \theta \times 1.285e-6 + \omega_x \times 4.8352e-4 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 3.0205e-8 \times \Omega_2 \times 5.0877e-9 - \Omega_3 \times \right) \\ \int \left( - \omega_x \times 8.3747e-4 \times \Omega_1 \times 3.0565e-8 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 3.0205e-8 \times \Omega_4 \times 5.0877e-9 \\ \phi \times 5.3309e-7 + \theta \times 1.285e-6 + \omega_x \times 4.8352e-4 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 3.0205e-8 \times \Omega_2 \times 5.0877e-9 - \Omega_3 \times \right) \\ \int \left( - \omega_x \times 8.3747e-4 \times \Omega_1 \times 3.0565e-8 \times \Omega_2 \times \right) \\ 1.6707e-4 + \Omega_1 \times 3.0205e-8 \times \Omega_4 \times 5.0877e-9 \end{array} \right]
\]

The finite set of real valued continuous variables for the satellite control system is defined as, \( X = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11} \} \), where \( x_1 \) represents the altitude of the satellite, \( x_2 \) represents angular velocity \( \omega_x \), \( x_3 \) represents the roll Euler angle \( \phi \), \( x_4 \) is the pitch Euler angle \( \theta \), \( x_5 \) is the yaw Euler angle \( \psi \), and \( x_8 \) to \( x_{11} \) represent the angular velocities of the four reaction wheels.

The continuous dynamics are specified as:

\[ f(q_3, x) = \left[ \begin{array}{c} \gamma \\ \omega_x \\ \delta \omega_y \\ \omega_z \\ \phi \\ \theta \\ \psi \\ \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \\ \end{array} \right]^T \]

There is no continuous dynamics in state \( q_3 \) because that \( q_3 \) is the static node, in this node the Euler angles are calculated and outputted directly according to the input conditions.

The domains—where the continuous dynamics are valid—are given as

\[ D(q_1) = \left\{ x \in \mathbb{R}^{11} : \text{Orbit_Attitude_Command} = 0 \ \text{and} \ (x_5 = \text{TargetRoll}, x_6 = \text{TargetPitch}, x_7 = \text{TargetYaw}) \right\}, \]

\[ D(q_2) = \left\{ x \in \mathbb{R}^{11} : (x_1 \leq \text{TargetAltitude}) \ \text{or} \ (\text{Orbit_Attitude_Command} = 1) \right\}, \]
Because that $q_3$ has no continuous dynamics so that the domain of $q_3$, can be presented as:

$$D(q_3) = \{ x \in \mathbb{R}^{11} : \text{EulerOutput} = 0 \} ,$$

$$D(q_4) = \{ x \in \mathbb{R}^{11} : (x_5 \neq \text{CalculatedRoll}, \ x_6 \neq \text{CalculatedPitch}, x_7 \neq \text{CalculatedYaw}) \} ,$$

$$D(q_5) = \{ x \in \mathbb{R}^{11} : (x_1 \geq \text{TargetAltitude}) \} .$$

Initial conditions enforced on the system are:

$$Init = \{ q_1 \} \times \left\{ x \in \mathbb{R}^{11} : \begin{cases} x_1 = \text{Initial Altitude} \\ (x_3 = \text{Initial Roll}, \ x_4 = \text{Initial Pitch}, \ x_5 = \text{Initial Yaw}) \end{cases} \right\} ,$$

$$\text{and} \ (x_2 = 0^\circ / s, \ x_3 = 0^\circ / s, \ x_4 = 0^\circ / s) ,$$

$$\text{and} \ (x_8 = 0^\circ / s, x_9 = 0^\circ / s, x_{10} = 0^\circ / s, x_{11} = 0^\circ / s) .$$

The discrete transitions in the system are:

$$E = \left\{ \begin{array}{l} (q_1, q_2) \rightarrow \{(\text{Orbit}\_\text{Attitude}\_\text{Command} = 1) \} \\ (q_2, q_3) \rightarrow \{ \begin{cases} x \in \mathbb{R}^{11} : (x_1 \geq \text{Target Altitude}) & \text{or} \\ x_1 \neq \text{Target Yaw} \end{cases} \} \\ (q_3, q_4) \rightarrow \{ \text{EulerOutput} = 1 \} \\ (q_4, q_5) \rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Calculated Roll} \\ x_6 = \text{Calculated Pitch} \\ x_7 = \text{Calculated Yaw} \end{cases} \} \\ \text{and} \ (x_1 < \text{Target Altitude}) \} \\ (q_4, q_5) \rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Calculated Roll} \\ x_6 = \text{Calculated Pitch} \\ x_7 = \text{Calculated Yaw} \end{cases} \} \\ \text{and} \ (x_1 \geq \text{Target Altitude}) \} \\ (q_5, q_5) \rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Target Roll} \\ x_6 = \text{Target Pitch} \\ x_7 = \text{Target Yaw} \end{cases} \} \\ \text{and} \ (x_1 \leq \text{Target Altitude}) \} \right\} .$$

The guard conditions are defined as

$$G_1 \ G_2 \ G_3 \ G_4 \ G_5 \ G_6$$

$$\begin{align*}
(q_1, q_2) & \Rightarrow \{(\text{Orbit}\_\text{Attitude}\_\text{Command} = 1) \} \\
(q_2, q_3) & \Rightarrow \{ \ x \in \mathbb{R}^{11} : (x_1 \geq \text{Target Altitude}) \text{or} \\
& \quad \text{and} \ (x_5 \neq \text{Calculated Roll}, \ x_6 \neq \text{Calculated Pitch}, \ x_7 \neq \text{Calculated Yaw}) \} \\
(q_3, q_4) & \Rightarrow \{ \text{EulerOutput} = 1 \} \\
(q_4, q_5) & \Rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Calculated Roll} \\ x_6 = \text{Calculated Pitch} \\ x_7 = \text{Calculated Yaw} \end{cases} \} \\
& \quad \text{and} \ (x_1 < \text{Target Altitude}) \} \\
(q_4, q_5) & \Rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Calculated Roll} \\ x_6 = \text{Calculated Pitch} \\ x_7 = \text{Calculated Yaw} \end{cases} \} \\
& \quad \text{and} \ (x_1 \geq \text{Target Altitude}) \} \\
(q_5, q_5) & \Rightarrow \{ x \in \mathbb{R}^{11} : \begin{cases} x_5 = \text{Target Roll} \\ x_6 = \text{Target Pitch} \\ x_7 = \text{Target Yaw} \end{cases} \} \\
& \quad \text{and} \ (x_1 \leq \text{Target Altitude}) \} \end{align*}$$

Reach$_H$

$$= \{ q_1 \} \times \{ x \in \mathbb{R}^{11} \}$$

$$\bigcup \{ q_2 \} \times \{ x \in \mathbb{R}^{11} : (\text{Orbit}\_\text{Attitude}\_\text{Command} = 1) \}$$

$$\bigcup \{ q_3 \} \times \{ x \in \mathbb{R}^{11} : (x_1 \leq \text{Target Altitude}) \}$$

$$\bigcup \{ q_4 \} \times \{ x \in \mathbb{R}^{11} : \begin{cases} (\text{Orbit}\_\text{Attitude}\_\text{Command} = 1) \\ (x_5 \neq \text{Calculated Roll}, \ x_6 \neq \text{Calculated Pitch}, \ x_7 \neq \text{Calculated Yaw}) \end{cases} \}$$

$$\bigcup \{ q_5 \} \times \{ x \in \mathbb{R}^{11} : (x_1 \geq \text{Target Altitude}) \}$$

$$\bigcup \{ q_5 \} \times \{ x \in \mathbb{R}^{11} : (x_1 \leq \text{Target Altitude}) \} \} .$$

The equations show that $q_1, q_2, q_3, q_4$ and $q_5$ can all be reached from the initial condition within a finite execution $(\tau, q) \in E^*_H$ with $\langle \tau \rangle = N < \infty$ and $q(\tau^n) = q_1$.

For Non-Blocking property, we examine

$$(q, x) \in \text{Reach}_H \bigcap \text{Out}_H ,$$

where
\[ \text{Reach}_H \cap \text{Out}_H = \{q_1 \times \{x \in \mathbb{R}^1 \mid \text{Orbit\_Attitude\_Command} = 1\} \]
\[ \cup \{q_2 \times \{x \in \mathbb{R}^1 \mid x_i = \text{TargetAltitude}\} \]
\[ \cup \{q_3 \times \{x \in \mathbb{R}^1 \mid \text{EulerOutput} = 1\} \]
\[ \cup \{q_4 \times \{x \in \mathbb{R}^1 \mid (\text{Orbit\_Attitude\_Command} = 1) \} \]
\[ \cup \{q_5 \times \{x \in \mathbb{R}^1 \mid (x_i = \text{TargetAltitude}) \} \]
\[ \text{or} \{x \in \mathbb{R}^1 \mid x_i \leq \text{TargetAltitude}\} \]
\[ \cup \{x \in \mathbb{R}^1 \mid (x_i = \text{TargetAltitude}) \}. \]

From the above result, we note that for all \((q, x) \in \text{Reach}_H \cap \text{Out}_H\), there exists \((q, q') \in E\) such that \(x \in G(q, q')\). This shows that the hybrid automaton is non-blocking.

For the deterministic property, we study the expression of \(\text{Reach}_H\) again, where we can find that from the initial condition, \((q_1, x)\), the only transition that can occur is \((q_1, q_2)\) when \((q, x) \in \text{Out}_H\); for \((q_2, x)\), the transitions that can occur are \((q_2, q_4)\) when \(x_i \geq \text{TargetAltitude}\), when \((q, x) \in \text{Out}_H\); and for \((q_3, x)\), the transition \((q_3, q_4)\) will occur when EulerOutput equals to 1; for \((q_4, x)\), the possible transitions are \((q_4, q_5)\) and \((q_4, q_6)\), however, \(G(q_4, q_5) \cap G(q_4, q_6) = \emptyset\), thus for all \((q, x) \in \text{Reach}_H\), \(x \in G(q_4, q_5) \cap G(q_4, q_6)\); when the node is in \((q, x)\), the only transition that can occur is \((q_4, q_5)\), when \((q, x) \in \text{Out}_H\).

Therefore, the conditions of Lemma II are satisfied. The result verifies that the hybrid automata of general control case are deterministic. According to Theorem I, we conclude that the hybrid automaton accepts a unique infinite execution.

To test the general automaton for the satellite control, we construct a satellite operation scenario and perform the simulation. Here, the satellite is inserted into the 300 km orbit with the initial attitude of 0° roll, 0° pitch, and 0° yaw angles and the inclination of 26 degrees. The input conditions are Target Attitude of 20° roll, 90° pitch, and 0° yaw angles; Target Altitude of 40,000 km; and Target Inclination of 0 degree. In this scenario, the inclination change can be implemented by firing the thrusters at the specific attitude. The algorithm used for calculating the proper attitude for inclination change is mentioned in the book by Wertz (1999). The satellite is deployed into the initial orbit with the initial attitude, then an orbit transfer command is triggered at the simulating time of 200 seconds to transfer the satellite to the target orbit. The attitude calculation node calculates the needed attitude changes for the inclination change and outputs the Euler angles of 0° roll, 0° pitch, and 50.18° yaw angles. Then the orbit maneuver node will ensure the satellite move to the new altitude. When the satellite reaches the new orbit, it will restore the target attitude and enter the operation mode. Figure 4 shows the orbit and attitude change of a satellite.

![Figure 4. Satellite Orbit and Inclination Change.](image)

**Automata for Three Satellites Cooperative Control**

We extend the single satellite control automata to a multiple satellite cooperative control case. To demonstrate this feasibility, we construct a scenario where there are three satellites with different altitude, and they are transferred to the same altitude (680 km) according to the Orbit Command. We note that each each satellite has identical automata, and we use the same LQR control method for attitude adjustment and Hohmann transfer for orbit change as in the previous section. The graphical representation of the automaton for each satellite is given in Figure 5.
The attitude scenario for this simulation is that the system maintains an attitude of 0° in the roll axis, −90° in the pitch axis and 0° in the yaw axis of LVLH as depicted in the bottom right section of Figure 1. This attitude is the one in which a camera is mounted for earth pointing at all times for target tracking and imaging, and it is called the Earth pointing node. When the system detects that an orbital change is necessary, the attitude of the spacecraft will be adjusted to 0° roll, 0° pitch, and 0° yaw angles, which is named the thruster firing node. It is necessary to align the thrusters of the spacecraft that provide the required change in velocity for the maneuver, which is also depicted in the left section of Figure 1.

In the three satellite control case, the discrete nodes are defined as \( Q = \{ q_1, q_2, q_3 \} \), where \( q_1 \) is Orbit Monitoring Node, \( q_2 \) is Attitude Adjustment Node, \( q_3 \) is Orbit Adjustment (altitude) Node. Note that in \( q_2 \), the satellite adjusts to either thruster firing mode or earth pointing mode according to the input command.

The finite set of real valued continuous variables for the LQR case is defined as, \( X = \{ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11} \} \), the same as the variable set in the previous section. The continuous dynamics are specified as

\[
f(q_1, x) = \begin{bmatrix} \gamma & \omega_x & \delta \omega_x & \omega_z & \phi & \theta & \psi & \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^T,
\]

and

\[
f(q_3, x) = \begin{bmatrix} h - F \times \Delta \theta \\ \phi \times 9.2334e-7 + \alpha_y \\ \theta \times 8.2353e-8 + \Omega_1 \times 2.1753e-8 + \Omega_2 \times 2.1753e-8 \\ \theta \times 2.2257e-8 \\ -\alpha_x \times 8.3747e-4 + \Omega_1 \times 3.0565e-8 + \Omega_2 \times 3.0565e-8 \\ \Omega_3 \times 3.0565e-8 + \Omega_4 \times 3.0565e-8 \\ \Omega_1 \times 9.2334e-7 + \alpha_y \\ \Omega_2 \times 2.8937e-4 + \Omega_3 \times 2.1753e-8 + \Omega_4 \times 2.1753e-8 \\ \Omega_1 \times 2.1753e-8 + \Omega_2 \times 2.1753e-8 + \Omega_3 \times 2.1753e-8 + \Omega_4 \times 2.1753e-8 \\ \Omega_1 \times 2.1753e-8 + \Omega_2 \times 2.1753e-8 + \Omega_3 \times 2.1753e-8 + \Omega_4 \times 2.1753e-8 \\ \Omega_1 \times 2.1753e-8 + \Omega_2 \times 2.1753e-8 + \Omega_3 \times 2.1753e-8 + \Omega_4 \times 2.1753e-8 \end{bmatrix} dt.
\]

The domains are

\[
D(q_1) = \begin{cases} x \in \mathbb{R}^{11} : (x_4 = 0, x_6 = 90, x_7 = 0) \\ \text{and ( Orbit_CMD = 0) } \end{cases},
\]

\[
D(q_2) = \begin{cases} x \in \mathbb{R}^{11} : (x_5 \neq 0, x_6 \neq 0, x_7 \neq 0) \\ \text{and ( Orbit_CMD = 1) } \end{cases},
\]

\[
D(q_3) = \begin{cases} x \in \mathbb{R}^{11} : (x_1 < 680km) \text{ or ( Orbit_CMD = 1) } \end{cases}.
\]

Initial conditions enforced on the system are:
\[
\text{Init} = \{\{q_1\}\} \times \left\{ x \in \mathbb{R}^5 : (x_1 = 680 \text{ km}) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (x_2 = 0^\circ / s, x_3 = 0^\circ / s, x_4 = 0^\circ / s) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (x_5 = 0^\circ, x_6 = 0^\circ, x_7 = 0^\circ) \\
\left. \quad \quad \quad \quad \quad \quad \quad \text{and } (x_8 = 0^\circ, x_9 = 0^\circ, x_{10} = 0^\circ, x_{11} = 0^\circ) \right\}.
\]

The discrete transitions in the system are
\[
E = \left\{ \{q_1, q_2\}, \{q_2, q_3\}, \{q_3, q_2\}, \{q_2, q_1\} \right\}.
\]

The guard conditions are as below.
\[
G = \left[ G_1, G_2, G_3, G_4 \right]^T
\]

\[
(q_1, q_2) \implies \left\{ x \in \mathbb{R}^7 : (x_1 = 0^\circ, x_6 = -90^\circ, x_7 = 0^\circ) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (\text{Orbit Command} = 1) \}
\]

\[
(q_2, q_3) \implies \left\{ x \in \mathbb{R}^7 : (x_1 = 0^\circ, x_6 = 0^\circ, x_7 = 0^\circ) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (\text{Orbit Command} = 1) \}
\]

\[
(q_3, q_2) \implies \left\{ x \in \mathbb{R}^7 : (x_1 \geq 680 \text{ km}) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (x_6 = 0^\circ, x_7 = 0^\circ) \}
\]

\[
(q_2, q_1) \implies \left\{ x \in \mathbb{R}^7 : (x_1 \geq 680 \text{ km}) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (x_6 = 0^\circ, x_7 = 0^\circ) \}
\]

The reachability of each node in the LQR case is represented as:
\[
\text{Reach}_{\text{LQR}} = \{q_1\} \times \{ x \in \mathbb{R}^5 : (x_1 = 0^\circ, x_5 = 0^\circ, x_7 = 0^\circ) \}
\]

\[
\cup \{q_2\} \times \left\{ x \in \mathbb{R}^5 : (x_1 \geq 680 \text{ km}) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (\text{Orbit Command} = 1) \}
\]

\[
\cup \{q_3\} \times \left\{ x \in \mathbb{R}^5 : (x_1 \leq 680 \text{ km}) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (x_5 = 0^\circ, x_7 = 0^\circ) \}
\]

The \text{Out}_{\text{LQR}} is:
\[
\text{Out}_{\text{LQR}} = \{q_1\} \times \{ x \in \mathbb{R}^5 : \text{Orbit Command} = 1 \}
\]

\[
\cup \{q_2\} \times \{ x \in \mathbb{R}^5 : (x_1 \geq 680 \text{ km}) \}
\]

\[
\cup \{q_3\} \times \{ x \in \mathbb{R}^5 : (x_1 = 680 \text{ km}) \}
\]

The equation shows that \( q_1, q_2 \) and \( q_3 \) can all be reached from the initial condition within a finite execution \((\tau, q) \in E_{\tau}^t \) with \( \langle \tau \rangle = N < \infty \) and \( q(\tau_N) = q_1 \).

We now use Lemma I and Lemma II to examine the existence and uniqueness properties of the hybrid automata, we first consider the non-blocking and deterministic properties of the system. For Non-Blocking property, we examine \((q, x) \in \text{Reach}_{\text{LQR}} \cap \text{Out}_{\text{LQR}} \) where
\[
\text{Reach}_{\text{LQR}} \cap \text{Out}_{\text{LQR}}
\]

\[
= \{q_1\} \times \left\{ x \in \mathbb{R}^5 : (x_1 = 0^\circ, x_6 = -90^\circ, x_7 = 0^\circ) \right. \\
\quad \quad \quad \quad \quad \quad \quad \text{and } (\text{Orbit Command} = 1) \}
\]

The result verifies that the hybrid automaton is non-blocking.

Next we examine the deterministic property for the control model. We determine the following using the expression of \( \text{Reach}_{\text{LQR}} \). From \((q_2, x)\), the only transition that can occur is \((q_2, q_2)\) when \( x \in \text{Out}_{\text{LQR}} \); for \((q_2, x)\), the transitions that can occur are \((q_2, q_3)\) and \((q_2, q_1)\), however,
\[
\text{G}(q_2, q_1) \cap \text{G}(q_2, q_1)
\]

and \( x \not\in \text{G}(q_2, q_1) \cap \text{G}(q_2, q_1) \) ensures the second condition of Lemma II to be held; for \((q_3, x)\) the only transition that can occur is \((q_3, q_3)\) when \( x \in \text{Out}_{\text{LQR}} \). The result verifies that the hybrid automaton of LQR case are deterministic in this scenario.

Therefore we conclude that the hybrid automaton used in the satellite scenario accepts a unique infinite execution for all \((q_0, x_0) \in \text{Init}_{\text{LQR}} \).

Here, we perform a simulation of controlling three satellites with hybrid automata theory. The initial altitudes of each satellite were set at 680 km, 670 km, and 660 km. A command was sent at a simulation time of 300 seconds which directed each satellite to adjust their orbits to the nominal altitude of 680 km. The results of this simulation are given in Figure 6. Where \( \text{Alt}_{1,3} \) denotes the
altitude of each satellite, $\theta_{-3}$ denotes the pitch angle of each satellite. $\phi$ and $\psi$ denote the roll and yaw angles respectively. Each satellite executes the attitude maneuver when the command is sent followed by the correct orbital maneuver that brings each satellite into the same orbit.


Conclusions

The hybrid automata are developed for the autonomous satellite control. We model command, operation modes, attitude control, and orbit maneuver of a satellite using hybrid automata. We utilize the definitions of reachability, non-blocking, and deterministic system for hybrid automaton to verify the existence and uniqueness of the developed satellite control hybrid automaton. The simulation results illustrate that the spacecraft maneuvers can be executed automatically through the use of hybrid automata. In the first simulation, we changed the attitude using LQR control for the attitude and Hohmann transfer for the orbit maneuver, which showed the feasibility of applying hybrid automata on a group of satellites for the autonomous attitude and orbit maneuver. Then we generalize the model to implement any attitude and altitude change. Through these simulations, we showed that the hybrid automata theory can be effectively used to autonomously control the orbits and attitudes for multiple satellites.

References


**Acknowledgment**

This research was supported by NSF Grant ECS–0554748.

**Bei Kang** received the B.S. and M.S. degree in electrical engineering from Northwestern Polytechnical University, Xian, China, in 2000 and 2003. He is now pursuing his Ph.D. degree in electrical engineering at Temple University. His research interests include the control of sensor networks, and hybrid systems.

**Dr. Chang-Hee Won** received his M.S. and Ph.D. from the University of Notre Dame in 1993 and 1995 respectively. From 1995 to 2000 he was a Senior Research Engineer at the Electronics and Telecommunications Research Institute, South Korea. From 2000 to 2005 Dr. Won was an Assistant Professor in the Department of Electrical Engineering at the University of North Dakota. He is now an assistant professor in the Department of Electrical Engineering at Temple University. His research interests include stochastic optimal control theory, spacecraft systems engineering, cooperative control, hybrid automata, and navigation. Dr. Won is a Member of IEEE, Eta Kappa Nu, and the American Institute of Aeronautics and Astronautics (AIAA).