IAC-18-B4.9.GTS.5

COMPARATIVE STUDY OF CLASSICAL AND FUZZY PID ATTITUDE CONTROL SYSTEM WITH EXTENDED KALMAN FILTER FEEDBACK FOR NANOSATELLITES

Prerna Baranwal[a]*, Karabee Batta[b], Tushar Kaushik[c]

[a] *Department of Electronics and Instrumentation, Birla Institute of Technology & Science-Pilani, India, 333031,* f2016568@pilani.bits-pilani.ac.in , prernabaranwal2190@gmail.com
[c] *Department of Electronics and Instrumentation, Birla Institute of Technology & Science-Pilani, India, 333031,* tktusharkaushik@gmail.com
[b] *Department of Computer Science and Information System, Birla Institute of Technology & Science-Pilani, India, 333031,* bkarabee@gmail.com

* Corresponding Author

Abstract

Active attitude control is a major part of any satellite mission. Fine pointing maneuvers like camera pointing for imaging, and antenna pointing for down-linking data are extremely important for mission success. Conventional PID Controllers in conjugation with Extended Kalman Filter as a sensor feedback have been actively used in the industry for fine pointing due to their robust and stable nature. Recently, fuzzy controllers have been observed to outperform classical controllers in certain non-linear systems. However, they are yet to be implemented on board a satellite. The paper covers a simulation study of both the classical and the fuzzy PID controller for a nano-satellite system being developed by our university. The system of interest is a 3U (30cm x 10cm x 10cm) nano-satellite with magnetorquers and reaction wheels as actuators. Optimal control coefficients for both fuzzy and classical PID are calculated. A comparative study based on various performance parameters such as pointing time, size of code etc. is performed and documented. The paper concludes with finding the suitable controller for the given environment and discussing the scope of optimisation in the controller to give better results.

Keywords: PID controller, Fuzzy PID controller, Extended Kalman Filter, nano-satellite

Nomenclature

$\mathbf{I}_{ii}$ = moment of Inertia at ii axis.
$\boldsymbol{\delta q}$ = difference between 2 quaternion values.
$\boldsymbol{\omega}$ = angular velocity of satellite.
$\boldsymbol{J}$ = moment of inertia.
$\dot{\boldsymbol{h}}$ = reaction wheel torque.
$\boldsymbol{m}$ = magnetic dipole moment.
$\boldsymbol{h}$ = angular momentum.
$\boldsymbol{B}$ = magnetic field.
$\boldsymbol{b}$ = unit vector of B.
$\boldsymbol{I}$ = 3x3 identity matrix.
$\boldsymbol{\tau^b}$ = torque on the satellite in body frame of reference.
$\acute{\omega}_{\boldsymbol{ib}}$ = derivative of angular velocity in body frame w.r.t inertial frame.
$\boldsymbol{\omega_{ib}}$ = angular velocity in body frame w.r.t inertial frame.
$\acute{\omega}_{\boldsymbol{i}}$ = angular acceleration on i axis.
$\boldsymbol{\omega_i}$ = angular velocity on i axis.
$\boldsymbol{\tau_y}$ = torque on i axis.
$\boldsymbol{\alpha}$ = angular acceleration.
$\boldsymbol{\omega_s}$ = angular velocity reading from sensor after EKF.
$\boldsymbol{\alpha_s}$ = angular acceleration reading from sensor after EKF.
$\boldsymbol{u(t)}$ = control system output at time t.
$\boldsymbol{K_p}$ = proportional constant.
$\boldsymbol{K_i}$ = integral constant.
$\boldsymbol{K_d}$ = derivative constant.

Acronyms/Abbreviations

Attitude Determination and Control System (ADCS), Birla Institute of Technology & Science (BITS), Extended Kalman Filter (EKF), Inertial Measurement Unit(IMU), Indian Space Research Organization (ISRO), Matlab Fuzzy Logic Toolbox (MFLTB), Proportional Integral Derivative (PID), Poly Picosatellite Orbital Deployer (P-POD), Set Point Variable (SPV), Takani-Sugeno-Kang (TSK).

1. Introduction

Development of nanosatellites is an upcoming research area in engineering and space science. They form a major part in small scale missions for technology demonstration and are a cheaper alternative to perform experiments by University students [1]. Team Anant, the student satellite team of BITS Pilani, is one such mission which is buiding a 3-U cubesat that has the payload as hyper spectral camera which will be used to test the pollution levels in the Indian Ocean

Like in any mission, ADCS also forms a major part in performing attitude control and orbit manoeuvring operations. Camera pointing for imaging, antenna pointing for down-linking data are all examples of fine pointing that is used in active attitude control.

Conventionally PID controllers in conjugation with EKF have been used to perform fine pointing in nanosatellites. Fuzzy controllers have been in general observed to outperform classical controllers in non linear systems. Though being widely implemented in on ground controllers, they still have to be implemented onboard a nano satellite. This paper discusses in detail the formation of optimal PID with an EKF as a sensor feedback, and Fuzzy PID controller for a nanosatellite. A comparative study between the converging time of both these systems is done and the results obtained are documented in this paper.

2. Satellite Specifications

The system on which the comparative study has been performed has its technical specifications as follows:

- 3U Cubesat (10cm x 10cm x 30cm).
- Orbit Assumption: Nearly circular, polar sun-synchronous orbit (Table 1).
- 3 Reaction Wheels: Hyperion: RW210 (Table 2), one on each axis.
- 3 magnetorquers – 1 coil, 2 rods, coil for 10cm x 10cm face of cubesat (Tables 3,4).
- Disturbance torques. (Table 5).
- Angular Velocity obtained after detumbling in the satellite frame of reference: 0.00199 rad/s; -0.00002 rad/s; 0.00035 rad/s at x, y, and z axes respectively.
- Moment of Inertia (Table 6)
- Pointing Accuracy Necessary: 0.004 degrees.

Table 1. Orbit Assumption Data

Field	Value	Unit
Semi Major Axis	6978	Km
Inclination	97	Degrees
Eccentricity	0.0000715	-
Argument of Perigee	150	Degrees
Right Ascension Node	0	Degrees
True Anamoly	10	Degrees

Table 2. Hyperion RW210 Reaction Wheel Specifications [2].

Field	Value	Unit
Total momentum	+/- 3.0	mN m s
Maximum torque	+/- 0.1	mN m
Maximum rotation rate	15000	rpm
Control Accuracy	+/- 0.5	rpm
Outer Dimensions	25x25x15	mm
Mass	32	g

Table 3. Magnetorquer Coil Parameters

Property	Value	Unit
Number of turns	440	-
Wire radius	0.127	Mm
Total mass	70.763	G
Maximum Moment	0.3353	A m^2
Resistance	51.96	Ohm
Inductance	0.0796	H
Maximum Power	481	mW

Table 4. Magnetorquer Rod Parameters

Property	Value	Unit
Core Length	9	cm
Core Radius	2	mm
Demagnetisation factor	0.0056	-
Total Mass	16.913	G
Maximum Moment	0.3318	A m^2
Resistance	344.3234	Ohm
Inductance	0.9544	H
Maximum Power	72.5	mW

Table 5. Disturbance Torques

Source of Disturbance	Maximum Induced Torque (N m)
Atmosphere	4.29×10^{-9}
Solar Radiation	3.8×10^{-9}
Gravity Gradient	2.82×10^{-8}
Residual Magnetic Moment	4.5×10^{-8}

Table 6. Moment of Inertia of Cubesat

Inertial Axis	Value (kg m^2)
I_{xx}	0.0060237
I_{xy}	0.0000029

I_{xz}	0.0000042
I_{yx}	0.0000029
I_{yy}	0.0013045
I_{yz}	0.0000131
I_{zx}	0.0000029
I_{zy}	0.0000131
I_{zz}	0.0060135

3. Theory

3.1 Necessity of Control Algorithms

The separation of the satellite from a launch vehicle, subjects it to fairly significant torques that lead to unusual spinning of the satellite. The uncontrolled body rates must be dissipated before the hyper spectral camera can be used to capture the images. Control algorithms are hence necessary to reduce the velocity of the satellite and achieve it to the pointing accuracy necessary for clear capture of images.

3.2 Classification of Control Algorithms

The control algorithms used on board the satellite can be broadly classified into 2 types.

- Active Control: This control is used using the actuators present on the satellite. It provides fine pointing accuracy and is mainly used in modes that require precise pointing.
- Passive Control: This control is used without the utilisation of actuators. It is used for modes that require coarse pointing like the detumbling mode.

The separation of the cubesat from a launch interface like the P-POD will subject the cubesat to high disturbance torques, leading it to start tumbling rapidly. To stop the satellite from spinning rapidly and to point it accurately to the desired location, both passive and active controls are employed. Immediately after the launch, B-Dot algorithm is employed [2]. This results in the cubesat being detumbled, and having angular velocity of the order 10^{-4} rad/s, and pointing accuracy as 0.2 degrees. However since our payload is extremely position specific hence the satellite needs to have angular velocity 0 and the pointing accuracy as 0.002 rad. For this active control is used which involves the use of magnetorquers and reaction wheels. The PID controller with EKF as sensor feedback and the fuzzy PID controller are implemented after the implementation of B-Dot algorithm. The construction of both the controllers is described in the next 2 sections.

3.1 PID Controller

The PID controller is the a control loop feedback mechanism that involves continuously modulated controller. It continuously calculates the error as the difference between the desired set point and the measured process variable. The correction applied to it is based on proportional, integral and derivative terms (Fig. 1). The goal is to derive the actual quaternion q to a desired quaternion q_c, and the angular velocity of the system to approach 0.

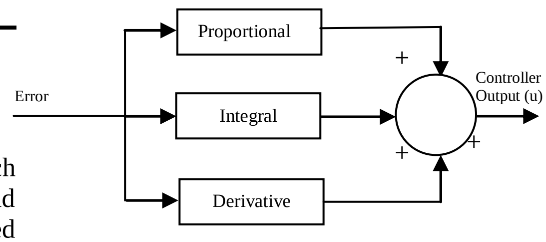

Fig. 1. Block Diagram Representing a PID Controller

The state to be controlled is represented by $X = \begin{bmatrix} \delta q \\ \alpha \\ \omega \end{bmatrix}$, where $\delta q = \begin{bmatrix} \delta q_{1:3} \\ \delta q_4 \end{bmatrix}$. $\delta q_{1:3}$ represents the vector part of the controller and δq_4 is the scalar part, such that the magnitude of the vector is 1. The desired quaternion q_c for our δq is taken as $[0\ 0\ 0\ 1]^T$.

3.1.1 Construction of sensor model

The dynamics of the satellite and the reaction wheel are represented by

$$J_{\dot\omega} = -[\omega\ \mathrm{x}\ ]J\omega + L$$
$$\dot h = -[\omega\ \mathrm{x}\ ]h + L$$

The total angular momentum represented by $J_{\dot\omega} + h$ is conserved. The effective wheel torque input is represented by L. The $[\omega\ \mathrm{x}\ ]$ represents the conversion of the angular velocity into the frame of reference in which the equations are being applied.

The total magnetic torque being generated by the magnetotorquer can be calculated as follows [2]:

$$m = \frac{k}{\|B\|}\ h \times b$$

Hence the torque generated due to the magnetic field is represented by:

$$L = -k(I_3 - bb^T)h$$

Considering the satellite as a rigid body, the angular momentum can be represented as:

$$h = I\omega,$$

Hence the total torque acting on the body can be represented by

$$\tau = \frac{d}{dt}(I\omega)$$

When this torque is represented in body frame, relative to the Earth Centred Inertial Frame, the following equation upon differentiation and frame conversion is observed.

$$\tau^b = I\acute{\omega}_{ib} + \omega_{ib}$$
,
Rearrangement of the equation gives:

$$\acute{\omega}_{ib} = I^{-1}(-\omega_{ib} \times (I\omega_{ib}) + \tau^b)$$

Upon writing the equation into component form results into the following equations:

$$\acute{\omega}_x = \frac{1}{I_{xx}}((I_{yy} - I_{zz})\omega_y\omega_z + \tau_x)$$

$$\acute{\omega}_y = \frac{1}{I_{yy}}((I_{zz} - I_{xx})\omega_x\omega_z + \tau_y)$$

$$\acute{\omega}_z = \frac{1}{I_{zz}}((I_{xx} - I_{yy})\omega_y\omega_x + \tau_z)$$

The torques in the above equations includes the torques generated due to both the reaction wheels and the magnetorquer.

The value of δq can be calculated using the following formula.

$$\delta q = \frac{1}{2}\Omega(\omega)q(t)$$

3.1.2 Control Law

The error that goes as input in the PID controller is represented by:

$$e(t) = \delta q$$
$$e(t) = \omega - \omega_s$$
$$e(t) = \alpha - \alpha_s$$

Hence the linear control law that is used in the PID controller is represented by:

$$u(t) = K_p \times e + K_d \times \frac{\Delta e}{\Delta t} + K_i \int_{t=0}^{t=t} e.dt$$ [3].

Since our state variable is δq, ω and α respectively hence all the 3 are controlled, with the desired quaternion to be [0 0 0 1] , and the desired angular velocity and acceleration to be 0.

3.2 EKF as feedback for sensors

The state estimate using EKF finds the best estimate of the state along with the measurements that are corrupted with noise. EKF here is used because of its ability of linearising the non-linear model of the cubesat. Along with the PID controller and the sensor model, this forms a closed loop controller.

The sensor model simulates the accelerometers and gyroscopes. Hence the state space vector of the EKF consists of quaternion, gyro drift rate bias, angular velocity and angular acceleration. The flowchart in the Fig.3 mentions the step by step formation of EKF. The parameters used in the implementation of the EKF model are as follows.

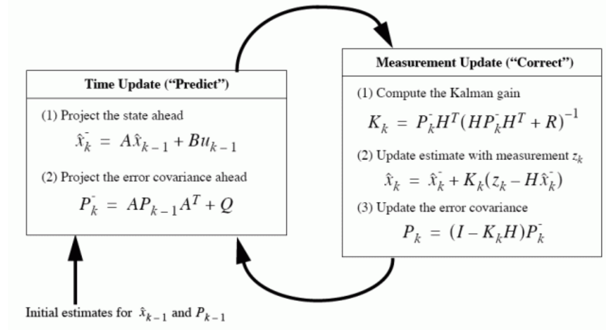

Fig 3: Block diagram of EKF [4]

The state fed to the matrix is represented by *X*. Gyro drift bias has been included to remove the error generated due to a drift in the gyroscope. K is the Kalman gain of the filter, and H is the measurement sensitivity matrix for the EKF, and is obtained after simplifying the Jacobian matrix. For simplicity its assumed to be identity. The variable k represents the time stamp. The matrix Q is used to account for the instrumentation error. The matrix R represents the state covariance. The matrix A and B are obtained by arranging the kinematics equations in a matrix form. All the matrices mentioned above have the size as 10x10.

3.3 Fuzzy PID Controller

To represent the real world scenario, the mathematical model that is proposed here is a fuzzy PID controller. Unlike the PID controller that linearises the non-linear system of the cubesat, this controller is a more accurate representation of the conditions of the cubesat.

There are 2 main kinds of architecture that are available here. Mamdani and TSK, which are described briefly below [3].

.

3.3.1 Mamdani architecture:

The rule set of Mamdani architecture looks as follows:

$$If\ A_{i1}(x_1), A_{i2}(x_2), \dots \dots, A_{im}(x_m), then\ Y\ is\ B_i$$

where: x_i 's are the input variables, $A_{ij}(x_j)$is a fuzzy set described on x_j, Y is an output variable, and B_i is a fuzzy set described on Y.

3.3.2 TSK Architecture:

The TSK architecture is used to represent the concepts of weights and membership functions. A rule of this form is given by:

$$If\ A_{i1}(x_1), A_{i2}(x_2), \dots, A_{im}(x_m),$$
$$Then\ Y = f(x_1, x_{2,\dots\dots,}x_n)$$

where: x_i 's are the input variables, $A_{ij}(x_j)$is a fuzzy set described on x_j , Y is an output variable, and $f(x_1, x_{2,.....,} x_n)$ is a fuzzy set described on Y.

In both Mamdani and TSK architecture, $A(x)$ is a function that takes x as its argument and returns the 'meaning' of this symbol in terms of the fuzzy set or the membership function of the fuzzy set. The antecedent part in this case is processed in exactly the same way as the Mamdani method, and then an obtained degree of applicability is assigned to the value of Y calculated as the function of real inputs. Table 7 shows a comparison between the 2 architectures.

Table7: Mamdani v/s. TSK

Similarity	
TSK	**Mamdani**
The antecedent parts of the rules are the same.	
Difference	
The consequent parts are fuzzy sets.	The consequent parts are singletons (single spikes) or mathematical functions of them.
Advantages	
TSK is easily understandable by a human expert.	More effective computationally.
Simpler to formulate rules.	More convenient in mathematical analysis and in system analysis.
Proposed earlier and commonly used.	Guarantees continuity of the output surfaces.

Since the factor of ease of computation is extremely important, hence the TSK architecture will be used. The two inputs are error and change in error.The control variable is α (angular acceleration) or ω (angular velocity).

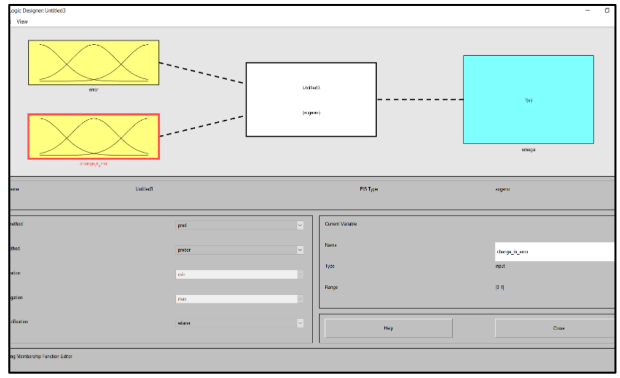

Fig 2: Structure of a TSK controller in MFLTB

A linear controller can be converted into a fuzzy controller by the following steps. The control equation with optimal values of Kp, Ki, Kd are converted into a rule base.Hence control variable is

$$u(t) = \omega \text{ (or } \alpha)$$

The error at time t is

$$e(t) = \omega - \tau \text{ (or } \alpha - \tau \text{)}$$

where τ is the SPV.

The change in error is

$$\Delta e(t) = e_2 - e_1 = \omega_2 - \omega_1 \text{ (or } \alpha_2 - \alpha_1)$$

3.3.3: Fuzzifying the standard PID Controller

The equation for a standard PID Controller is:

$$u(t) = K_p \times e + K_d \times \frac{\Delta e}{\Delta t} + K_i \int_{t=0}^{t=t} e.dt \text{ [3]}$$

The controller also needs a variable to store the sum of errors for the integral part of the controller, denoted by:

$$\sigma e(t) = \sum_{i=1}^{i=t} e(i)$$

Table8: Rule Set

e(t) \ Δe(t)	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PS
ZO	NB	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PB	PB
PM	NS	ZO	PS	PM	PM	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

The Table 8 represents the rule set that is used to describe the fuzzy set. The rule description is described in Table 9.

Table 9: Rule Description (in SI units)

	NB (Negative Big)	NM (Negative Medium)	NS (Negative Small)	Zero	PS (Positive Small)	PM (Positive Medium)
Error	[-3e-3 2e-3 -1e-3]	[-2e-3 -5e-4 -1e-4]	[-5e-4 -5e-5 -1e-15]	[-9e-6 -1e-15 9e-6]	[-1e-15 9e-6 1e-4]	[1e-4 5e-4 2e-3]
Change in error	[-3e-3 -2e-3 -1e-3]	[-2e-3 -5e-4 -1e-4]	[-5e-4 -5e-5 -1e-15]	[-9e-6 -1e-15 9e-6]	[-1e-15 9e-6 1e-4]	[1e-4 5e-4 2e-3]
Omega	-2e-3	-5e-4	-5e-5	-1e-15	9e-6	5e-4

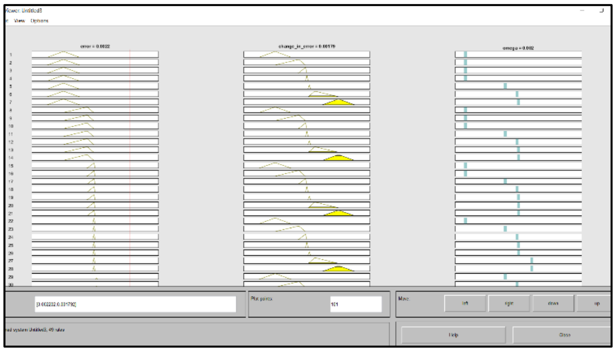

Fig 4: Rule Set Representation in MFLTB

3.3.4: Defuzzification

The defuzzification goal in Mamdani-type fuzzy controllers is to produce a crisp output taking the fuzzy output obtained after rules processing, clipping or scaling. No defuzzification is necessary in TSK fuzzy controller. Because the input for the defuzzification procedure is the clipped (or scaled) membership function, most of the methods propose the procedure of calculating the integrated evaluation of this input.

There are various defuzzification procedures which include:

- Centroid defuzzification
- Centre of gravity defuzzification
- Mean of maxima
- First of maxima defuzzification
- Middle of maxima defuzzification
- Height defuzzification

No procedure amongst the one written above has shown significant advantage over the other, hence it doesn’t effect significantly if either of the above is chosen. Here we use centroid defuzzification:

$$u^* = \frac{\sum_{i=1}^{i=k} u_i * \mu(u_i)}{\sum_{i=1}^{i=k} \mu(u_i)}$$

where µ is the membership function.

For the continuous case, the summation can be replaced by the integral on the specified interval. It takes the mean of all values between the combined membership function.

3.4 PID controller with EKF

The PID controller that is formed using the above conditions is the open loop. On combination of the controller with the EKF along with the sensor readings forms a closed loop. The necessity for the adding an EKF along with a PID controller is to optimise the control action of the controller. The EKF filters out the external noise, and linearizes the non linear equations of the PID controller. This leads to formation of optimal PID controller. The figure 5 shows the block diagram used to simulate the same. A sensor model along with the addition of white Gaussian noise, simulates the Inertial Measurement Sensors present on our satellite.

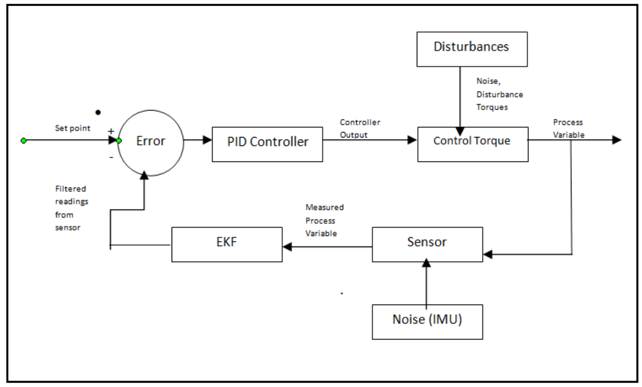

Fig 5: Complete Block diagram of PID controller with EKF

3.5 Fuzzy PID controller with EKF

Since the fuzzy PID controller simulates the real world due to the fuzzy sets present, and is implemented on non linear systems, addition of EKF will result in increased complexity and run time of the code. Hence EKF along with a sensor input was not implemented with Fuzzy PID controller.

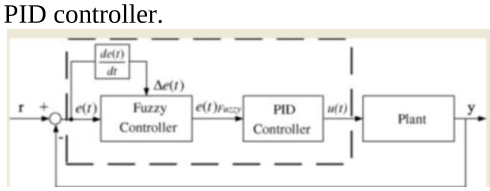

Fig 7: Fuzzy PID block diagram

4. Results

The output of fuzzy PID controller is as follows: ‘

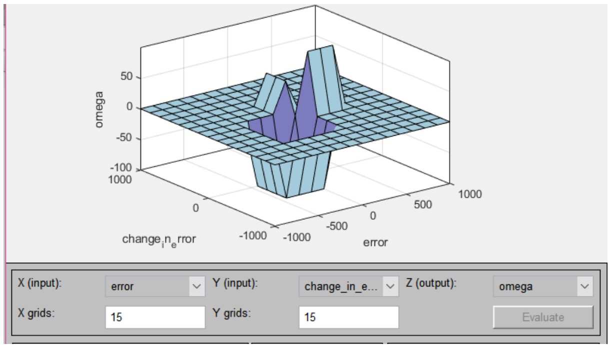

Fig 8: Graph showing convergence of PID

This controller is able to get the required pointing accuracy of 10e-3 rad, with the angular velocity and acceleration being of the order 10e-6, which is sufficient for our satellite.

The output achieved by the normal PID in addition to EKF is as follows:

The optimal EKF PID again converges to 0.002 degrees, with angular velocity and acceleration of the order 10e-6. Though the computation time required was more as compared to Fuzzy PID. The proportional, integral and derivative values obtained are 0.01012, 0.0000503and 0.00101 respectively.

5. Conclusions

The pointing accuracy provided by both the controllers are well within the requirements of our cubesat. However, the implementation of EKF along with fuzzy PID, though converging within the required time limit occupies a lot of computation power Also with the EKF matrices having size 10x10, it becomes quite complicated. The run time of simulating a fuzzy PID is observed to be lesser than that of normal PID. On the other hand, a fuzzy controller mimics the real life situations which takes into account the presence of attenuation/noise as well. Hence for majority of cases, where unattenuated signals and attenuated signals will fall in the same membership class. This reduces the need to add extra filtering modules and computing memory, hence it is more efficient computationally when dealing with small amounts of noises. A fuzzy PID controller works well enough, if there is a system with low disturbance noise, like the one in ours. In case the system consists of large noise signals, there is a chance that the noise signal and the control input may lie in different membership functions.

6. Discussion on Optimal Fuzzy PID

In recent years, two objective methods for the design of fuzzy control have been developed. The first of these is a method for the primary goal, the best possible imitation of the static and/or dynamic control behaviour of a human expert. The second method is the model based design of optimal, relatively robust, fuzzy controllers. For the design of fuzzy systems to such a "best possible" level of control, it is necessary to modify the criteria of non-linear control theory, design a parametric fuzzy concept, and select efficient search strategies. The first category is called fuzzy mode control and the second is called fuzzy control. Either way, parameters have to be optimised. The problem of parameterization of the fuzzy concept demands that the large number of degrees of freedom of the common fuzzy logic concept be reduced. This will be achieved if:

- the rule set is treated as invariant
- non-linear, parametric membership functions for the fuzzy inputs are defined.

There are multiple ways to approach this problem. Here we will consider genetic algorithms and bee colony approach briefly

.

6.1 Genetic algorithms: [5]

This relatively new approach to fuzzy systems is an example of a probabilistic nature-based heuristic. It produces children from parents using two genetic operations: crossover and mutation. The children are the different values that can be taken by the control variable. Multiple possible values of the parameter are encoded as bit strings and pairwise strings are taken, their substrings interchanged and the resulting substring is checked to see if the control conditions are satisfied in a better way. Mutation occurs when we deliberately change one or some of the bits of the bit string to achieve a new evolved species. by doing so we increase the probability of getting the desired values. The advantage of genetic algorithms over conventional optimisation algorithms is that they allow parallel inheritance, that is they allow multiple parameters to evolve simultaneously.

6.2Bee Colony optimization: [6]

It is another technique that belongs to the class of nature based meta heuristic methods. This method (BCO) uses an analogy based on the way the bees do their foraging in nature, and the way in which they apply their search optimization methods to find optimal routes between the hive and the source food.

The model defines three main components as shown below:

1) Source of food: the value of a food source depends on many factors, including its proximity to the hive, wealth or food concentration and ease of food extraction. Here the source will be the set value.

2) Working bees: they are associated with a current or exploited food source. They carry with them information about that particular source, its distance, location and return to share, with a certain probability. Here a suitable iterator data structure can be defined.

3) Scout bees: they are in constant search of a food source.

The nodes will be the list of possible values that can be taken by the control variable.

A bee is expected to work out a way, the set of paths used by 1 bee will be used by others for foraging. During foraging, a bee goes from a node to another node until the destination is reached. In the model bee a heuristic rule is used to help transition the bee in its decision making about what node to visit next.

After a set of bees lands at one node, it checks for the nearest m nodes. The fittest bees of the current existing lot are selected for the next transition until the termination condition is reached. Fuzzy logic is used to dynamically update the parameter conditions at run time.

Acknowledgements

Team Anant, Student Satellite Team BITS Pilani, BITS Pilani Administration

References

[1] Junquan Li, Mark Post, Thomas Wright, and Regina Lee, "Design of Attitude Control Systems for CubeSat-Class Nanosatellite," Journal of Control Science and Engineering, vol. 2013, Article ID 657182, 15 pages, 2013.
[2] F. Landis Markley, John L. Crassidis, Fundamentals of Spacecraft Attitude Determination and Control, New York, 2014.
[3] Leonid Reznik, Fuzzy Controllers, Melbourne Australia, 1997.
[4]http://www.ilanman.io/blog/2016/4/29/kalman, accessed 28th September, 2018.
[5] Camilo Caraveo, Fevrier Valdez, Oscar Castillo∗, Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation, Tijuana Institute of Technology, Tijuana Mexico.
[6]]: Oscar Castillo∗, Héctor Neyoy, José Soria, Patricia Melin, Fevrier Valdez, A new approach for dynamic fuzzy logic parameter tuning in AntColony Optimization and its application in fuzzy control of a mobilerobot, Tijuana Institute of Technology, Tijuana, Mexico