A Spiking Neural Network as a Quadcopter Flight Controller - Part I

Sushrut Thorat
*Department of Physics, IIT Bombay
Email: isush4u@iitb.ac.in

***Abstract*—The dynamics of a quadcopter are unstable and non-linear. As a result, the quadcopter's flight relies heavily on the Flight controller. Here we present a robust Control Scheme which can act as the flight controller for the quadcopter, even in the presence of noisy winds, IMU noise, and delayed signals. We also present an introduction to Spiking Neural Networks. In Part II, we would describe a SNN-based Flight Controller, and analyse its performance.**

I. Introduction

Quadcopter design is popular in Unmanned Aerial Vehicle (UAV) research. The need for aircraft with greater maneuverability and hovering ability has led to the current rise in quadcopter research. They have been recently used in acquiring aerial imagery, assisting in disaster assessment, surveillance by the military and delivering parcels [1]. It is a cheap platform and is easily available to enthusiasts who would want to experiment with aerial flight. It has also been used to study and demonstrate the capability of various control schemes. We would ultimately like to use the quadcopter to demonstrate the abilities of Spiking Neural Networks.

A quadcopter is a four-rotor rotorcraft. Instead of a variable pitch main rotor as in a helicopter, the craft is maneuvered by adjusting the relative speed between the individual motors [17]. The difference between the speeds of the motors causes a torque which leads to a rotation around the corresponding axis. This also tells us that slight variations in motor speeds could lead to instability, if unchecked. But, as we can use smaller, fixed-pitch motors, atleast the construction becomes easier as compared to a helicopter.

A quadcopter requires a robust controller. As aerial vehicles are susceptible to disturbances and unpredictable dynamics, making an accurate model of the system is very difficult. Thus, we need to simplify the model in use. We do that in designing our Control Scheme.

In this report, we analyse the merits of a Control Scheme in controlling a quadcopter. We need a control scheme such that the quadcopter must be able to:

- Maintain stable hover.
- Follow velocity waypoints (velocities required by the Flight Control module).
- Recover from disturbances.

in an environment with,

- Noisy winds.
- Noisy, and delayed IMU data.

We will then present an overview of the field of Spiking Neural Networks as a starting point towards building a SNN based on the Control Scheme.

Fig. 1. The DJI Phantom FC40 RC Quadcopter

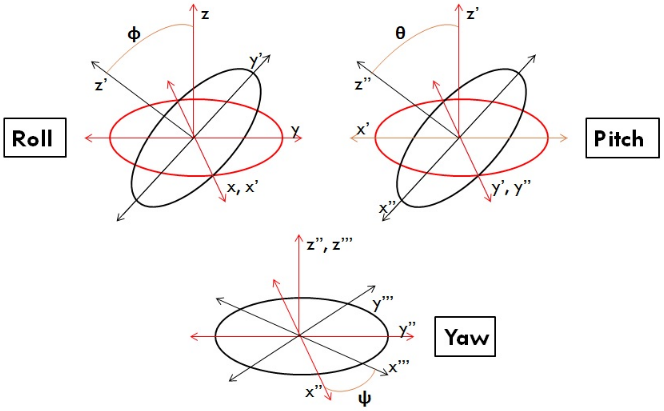


Fig. 2. The Euler angles θ, ϕ, and ψ are shown. The three types of rotations shown, commute, and hence the angles can be used to describe the state in the vehicle frame. The n-primed coordinates are the new body frame coordinates acquired after the appropriate rotations from the $(n-1)$-primed coordinates.

II. Dynamical Model

A quadcopter has 4 rotor blades which rotate and produce thrust. The frame of the quadcopter holds the Inertial Measurement Unit(IMU), the processor, the sensors, and other payload. A DJI Phantom Quadcopter is shown for reference in fig.1.

The *state* of a quadcopter can be described by 6 coordinates - the three Cartesian coordinates (X, Y, and Z) of the Center of Mass of the quadcopter, in the ground frame, and the three Euler angles (θ, ϕ, and ψ) in the vehicle frame (a cartesian frame which moves with the quadcopter, without any rotations w.r.t the ground frame) as shown in fig.2.

The external forces that act on the quadcopter are: 1. Gravitational force, 2. Air pressure and drag. The difference in air pressure creates a thrust at each rotor (Similar to how airplanes generate lift). The air also exerts a drag force on the frame of the quadcopter.

As the lifts generated by the motors can be considered as proportional to the motor speeds squared [2],

$$U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2), \tag{1a}$$

$$U_2 = lb(-\Omega_2^2 + \Omega_4^2), \tag{1b}$$

$$U_3 = lb(-\Omega_1^2 + \Omega_3^2), \tag{1c}$$

$$U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2), \tag{1d}$$

$$\Omega = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{1e}$$

where, Ω_n is the speed of motor n, U_1 is the overall thrust produced, U_2 is the roll torque, U_3 is the pitch torque, U_4 is the yaw torque, b is the Thrust factor, d is the Drag factor, and l is the length of an arm of the quadcopter. The function of Ω will become clear in a moment.

If we consider that the wind velocity is $\mathbf{v}_w$, and the torque due to the gradient in the wind speed over the face of the quadcopter is $\mathbf{T}$, then the coordinate accelerations are given by [3],

$$\ddot{X} = (sin\psi.sin\phi + cos\psi.sin\theta.cos\phi)\frac{U_1 + A(\dot{X} - v_{w;x})}{m} \tag{2a}$$

$$\ddot{Y} = (-cos\psi.sin\phi + sin\psi.sin\theta.cos\phi)\frac{U_1 + A(\dot{Y} - v_{w;y})}{m} \tag{2b}$$

$$\ddot{Z} = -g + (cos\theta.cos\phi)\frac{U_1 + A(\dot{Z} - v_{w;z})}{m} \tag{2c}$$

$$\ddot{\phi} = \frac{I_{YY} - I_{ZZ}}{I_{XX}}\dot{\theta}\dot{\psi} - \frac{J_{TP}}{I_{XX}}\dot{\theta}\Omega + \frac{U_2 + T_\phi}{I_{XX}} \tag{2d}$$

$$\ddot{\theta} = \frac{I_{ZZ} - I_{XX}}{I_{YY}}\dot{\phi}\dot{\psi} - \frac{J_{TP}}{I_{XX}}\dot{\phi}\Omega + \frac{U_3 + T_\theta}{I_{XX}} \tag{2e}$$

$$\ddot{\psi} = \frac{I_{XX} - I_{YY}}{I_{ZZ}}\dot{\phi}\dot{\theta} + \frac{U_4 + T_\psi}{I_{ZZ}} \tag{2f}$$

where, A is the cross-section area of the quadcopter as viewed along the Z-direction in the hovering state, and Ω results due to Gyroscopic torque.

The motor speed response to the input motor voltage (v_n) is modelled by the equation,

$$\dot{\Omega}_n = -\frac{K_E K_M}{RJ_{TP}}\Omega_n - \frac{d}{J_{TP}}\Omega_n^2 + \frac{K_M}{RJ_{TP}}v_n \tag{3}$$

The values of all the constants are provided in the Appendix.

A. Simulation conditions

We simulate the dynamics of the state in MATLAB, using Euler integration. The integration step size is 0.1 ms. We chose this step-size as it is required while modelling Spiking Neural Networks, and we would like our system to be compatible with our future goals. We observe convergent results at this step size.

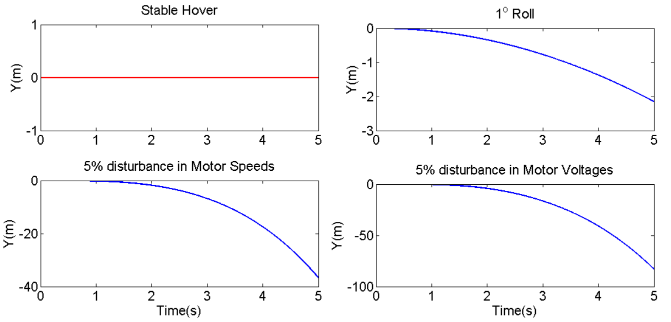


Fig. 3. Effect of small disturbances in the hovering state. Disturbances were provided in the initial roll angle, the initial motor speeds, and the input voltages, as stated in the figure. Even small disturbances have large effects, in this case a drift in the Y-direction.

Wind drag force is modelled by using a Gaussian white noise, filtered by using the 5-point Moving Average scheme [4]. The corresponding torque is modelled in the same way.

III. Necessity of Control

A quadcopter should be able to hover, i.e. stay afloat at a fixed position. It might seem trivial as we could just compute the motor speeds required for hovering and provide appropriate input voltages, but a look at fig.3 shows that the state of hovering can be easily disrupted by minute disturbances. In the figure, the disturbances provided are small, but the quadcopter deviates from its setpoint by atleast 5 metres.

What is the cause of such large a deviation? The acceleration due to gravity is around 9.8 m/s. To counter the force of gravity and keep its vertical position, the quadcopter has to generate a huge amount of thrust. So, the slightest deviation in the Euler angles will result in a big unopposed component of force along that particular direction. Similarly, the slightest deviation from the set motor speeds would result in an unopposed torque which will make the quadcopter spin and crash. There is no restoring force or torque that is inherent in the system.

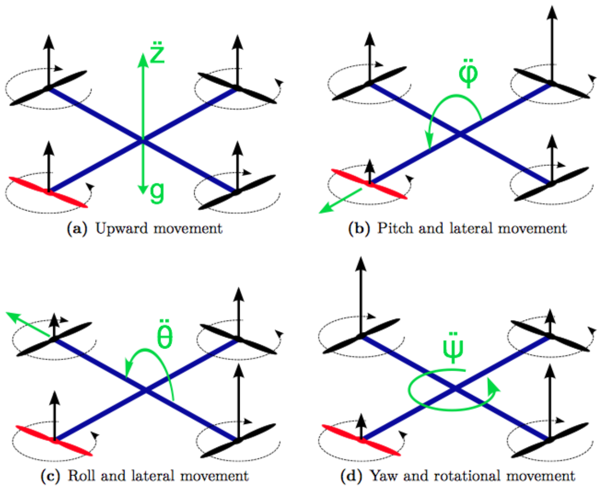
This is where a robust control scheme is important, which should make any desired setpoint a point of global stable equilibrium, or atleast a point of local stable equilibrium in a substantial domain.

IV. The Control Scheme

Before designing a control scheme, we need to understand what steps are to be taken to make a quadcopter move.

A. The Canon of Quadcopter Control

We change the Euler angles to effect motion. If we can get the quadcopter to move on the coordinate axes in the ground frame, we can couple those movements to move in any direction. The basic movements are shown in fig.4. Opposite blades are made to rotate in the directions shown, to ensure that the Gyroscopic torque remains zero. By changing the rotation speeds, we can navigate in space.



(a) Upward movement (b) Pitch and lateral movement

(c) Roll and lateral movement (d) Yaw and rotational movement

Fig. 4. Movement of the Quadcopter. The clockwise/anticlockwise arrows around the blades show the direction of rotation of the blades. The arrows perpendicular to the blade rotation plane denote the thrusts produced. (Source: http://uav-society.blogspot.in/2014/06/quadcopter-mechanics.html)

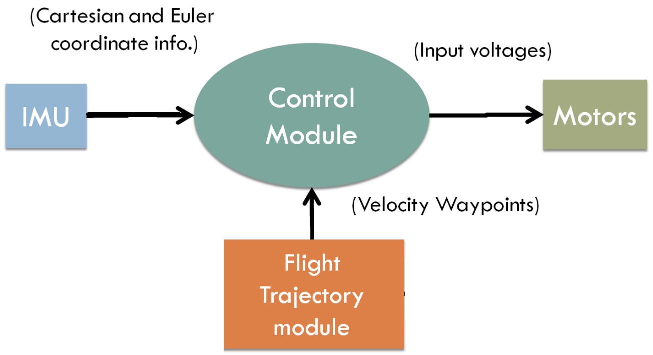


Fig. 5. The flow of information in the quadcopter. We get the instantaneous values of the cartesian coordinate accelerations and rotational velocities from the sensors in the IMU, and process them to estimate the Euler Angles. We get the instantaneous value of the expected velocity profile from the Trajectory Module. This information is processed to generate input voltages for the motors that make the quadcopter follow the velocity waypoints.

B. The Scheme

We have stated the equations governing the dynamics. The onboard IMU will provide us with instantaneous flight data. We assume that a Flight Trajectory Module will provide us with Velocity Waypoints [6]. The Control Module will then process the given data and output the Input Voltages to the motors for the succesful execution of the task. The information flow is depicted in fig.5.

To control the cartesian velocities, we need to control the state of the quadcopter.

To control the state, we need information about the instantaneous cartesian accelerations and velocities, and the instantaneous euler angles and angular velocities.

The on-board Inertial Measurement Unit (IMU) provides us with the cartesian accelerations, and the angular velocities. We integrate the cartesian accelerations to get the velocities, and use a Kalman filter to estimate the Euler Angles from the IMU output [5].

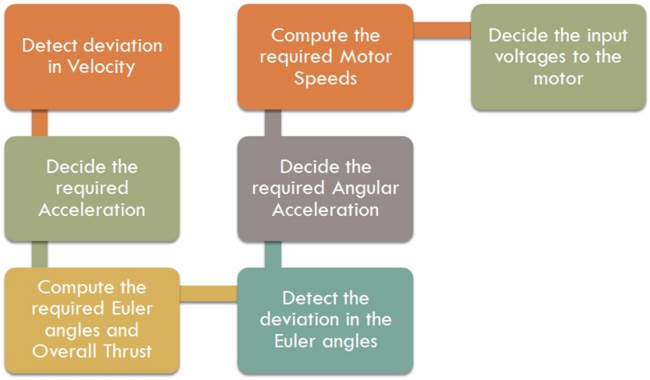

Detect deviation in Velocity
Decide the required Acceleration
Compute the required Euler angles and Overall Thrust
Detect the deviation in the Euler angles
Decide the required Angular Acceleration
Compute the required Motor Speeds
Decide the input voltages to the motor

Fig. 6. The Control Scheme. We quantify these decisions in Eqs.(5)&(6)

The most accurate way of designing a control scheme would be to take the entire dynamics we discussed so far, into account. But such a bottom-up approach is computationally intensive. So, we simplify the model we would subscribe to, in building the control scheme.

We consider Near-Hover flight, where the deviation in the Euler Angles is taken to be minute. The angular speeds are small. We neglect the drag force and torque for now (They cannot be measured by the IMU anyway). We will then see if this simplification can work even in the cases where the Euler Angles and angular speeds are large, and in the presence of wind. As we are designing a mechanism to overcome disturbances, we expect that it would work in some extreme cases too.

We can navigate through space by changing two Euler Angles, and keeping the third one constant. Here, we keep the yaw angle constant. This is because it isn't affected by the Gyroscopic torque, and so we can relax the control requirements on it as compared to the other two Euler angles. Also, the roll and pitch motion have similar governing dynamics, and it is easier to design a control scheme using them. So, we would maintain $\psi = 0$, and control the roll and the pitch.

So, we simplify the equations (2) to,

$$\ddot{X} = \theta\frac{U_1}{m} \qquad \ddot{Y} = -\phi\frac{U_1}{m} \qquad \ddot{Z} = -g + \frac{U_1}{m}, \tag{4a}$$

$$\ddot{\phi} = \frac{U_2}{I_{XX}} \qquad \ddot{\theta} = \frac{U_3}{I_{XX}} \qquad \ddot{\psi} = \frac{U_4}{I_{ZZ}} \tag{4b}$$

The equations (1) and (3) cannot be simplified for Near-Hover flight.

The overall control scheme is shown in fig.6.

The equations involved in the control scheme are:

$$\Delta\vec{v} = \vec{v}_{inst} - \vec{v}_{set} \qquad \vec{a}_{req} = -k_a \Delta\vec{v}, \tag{5a}$$

$$\theta_{req} = \frac{\ddot{X}_{req}}{\ddot{Z}_{req} + g} \qquad \phi_{req} = -\frac{\ddot{Y}_{req}}{\ddot{Z}_{req} + g}, \tag{5b}$$

$$U_{1req} = (\ddot{Z}_{req} + g)m \qquad \vec{A}_{req} = [\theta_{req} \quad \phi_{req}]^T, \tag{5c}$$

$$\Delta\vec{A} = \vec{A}_{inst} - \vec{A}_{req} \qquad \vec{\omega}_{req} = -k_o \Delta\vec{A}, \tag{5d}$$

$$\omega_{\psi;req} = -k_{o\psi}\psi_{inst} \qquad \Delta\vec{\omega} = \vec{\omega}_{inst} - \vec{\omega}_{req}, \tag{5e}$$

$$\Delta\omega_{\psi} = \omega_{\psi;inst} - \omega_{\psi;req} \qquad \vec{\alpha} = -k_{ac}\Delta\vec{\omega}, \tag{5f}$$

$$\alpha_{\psi} = -k_{ac\psi}\Delta\omega_{\psi} \qquad \vec{V} = f.\vec{\Omega}\circ\vec{\Omega} + g1.\vec{\Omega} \tag{5g}$$

where,

$$\vec{\Omega} = [\Omega_1 \quad \Omega_2 \quad \Omega_3 \quad \Omega_4]^T, \tag{6a}$$

$$\vec{\Omega}\circ\vec{\Omega} = abs\left(\begin{bmatrix} b & b & b & b \\ -lb & 0 & lb & 0 \\ 0 & -lb & 0 & lb \\ -d & d & -d & d \end{bmatrix}^{-1} \begin{bmatrix} U_{1req} \\ I_{XX}\vec{\alpha} \\ I_{ZZ}\alpha_{\psi} \end{bmatrix}\right) \tag{6b}$$

$$\vec{V} = \begin{bmatrix} V_1 & V_2 & V_3 & V_4 \end{bmatrix}^T \tag{6c}$$

where, $abs(\vec{v})$ maps the elements of the vector $\vec{v}$ to their absolute values. The k's are tuning parameters which were optimised manually. The values of the constants k's, f, and $g1$ are given in the Appendix.

We consider ψ separately because the moment of inertia along the Z'-axis is twice of that along the X' and Y' axes in the body frame.

Let us analyse the overall behavior of the Control Scheme.

C. Simulation conditions

The Control module is executed with the model of the quadcopter dynamics. The update rate of the control module is 100 Hz [7].

The noise in the IMU output is modelled as a Gaussian white noise, filtered with a 5-point moving average. We model the delay in the IMU output by executing the control module at a random time before the normal set time of execution, but outputting the voltage at the normal set time.

Simulations are run for hovering, velocity waypoint navigation, and recovery from angular disturbances. In all the cases, the environment has a noisy wind with mean speed of 3.5 m/s (normal wind speed), and standard deviation of 0.5 m/s. The noisy drag torque has a mean of 0, and standard deviation of 1.35×10^{-3} Nm. The delay in the IMU data is around 0.47 ms [8]. The noise in IMU data is 15% of the signal. The results are explained in the next section.

V. Results

One objective was to convert any given Velocity Setpoint into a stable fixed point of the system. As seen in fig.7, when **v**=0 is the desired Velocity Setpoint, trajectories starting with different initial conditions converge to the setpoint.

Now let us see how the Control Scheme fares in the tasks of hovering, velocity waypoint navigation, and recovery from angular disturbances.

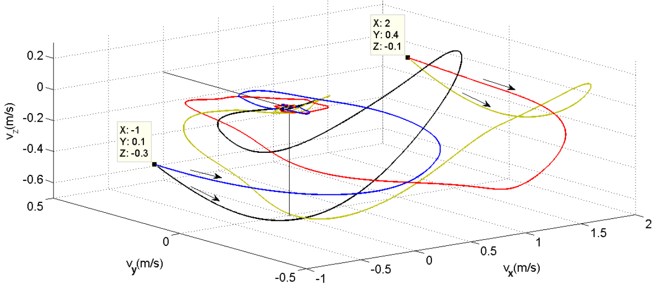


Fig. 7. Creation of a stable fixed point at **v**=0. The simulations for different initial conditions were run for 5 s each. Both the initial cartesian velocities, and euler angles were varied (in the absence of wind).

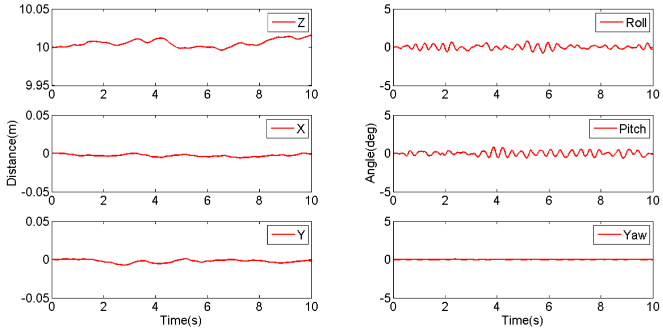


Fig. 8. Hovering. The displacement in the cartesian coordinates is unchecked as we are looking to control the velocities. Here, the velocity setpoints are zero throughout. The controller stabilizes the system to those setpoints. The Trajectory Module which will provide the velocity waypoints will ensure that we get appropriate velocity setpoints to avoid the loss of altitude.

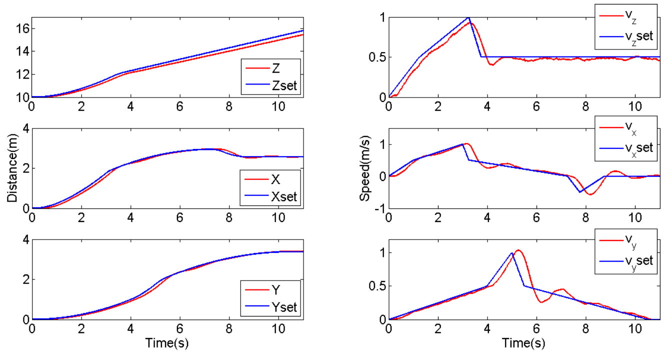


Fig. 9. Velocity Waypoint control. We can robustely follow accelerations upto 0.5 m/s^2. The displecement profile shows that the quadcopter can follow velocity profiles without much error in displacement.

Fig.8 shows that the quadcopter can now hover in noisy conditions and wind.

In fig.9, the Trajectory Module provides Velocity Waypoints to the Control Module, and they are implemented perfectly if the accelerations involved do not exceed 0.5 m/s^2 for a long duration.

In fig.10, we create a periodic disturbance with an increasing magnitude in the Roll angle. The quadcopter now recovers from disturbances upto 110^o. The problem in recovering from

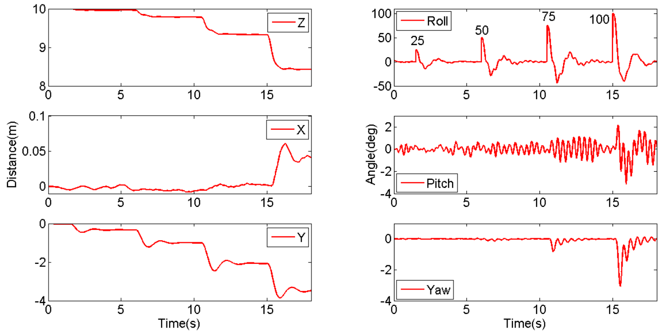


Fig. 10. Recovery from Angular Disturbances. The space needed to recover increases non-linearly with angular disturbance. But that is okay, as our primary goal is to follow velocity waypoints robustely.

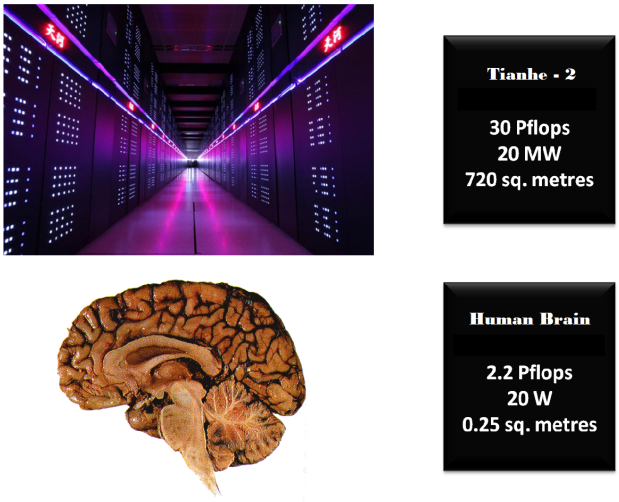


Fig. 11. A comparison of speed, power consumption, and the area occupied, between Tianhe-2, the world's fastest supercomputer, and the Human Brain.

greater angles is the model we have chosen - Near-Hover Flight. The simplified set of equations (4) cannot account for the response required to recover from very large angles. We also observe large overshoots in displacement. In our model, the system tends to approach the velocity setpoint with minimal overshoot in velocity. But that leads to longer duration of the cartesian acceleration being in one direction, and therefore the large overshoot in displacement. This overshoot can be corrected by creating newer setpoints via the Trajectory Module.

Thus we have looked at a control scheme which follows velocity waypoints in realistic environmental conditions. Now we turn to an introduction to Spiking Neural Networks.

VI. Spiking Neural Networks

Spiking Neural Networks(SNNs) are the third generation of Neural Networks, which are realistic models of biological neurons, and are considered computationally more powerful than the first two generations [9].

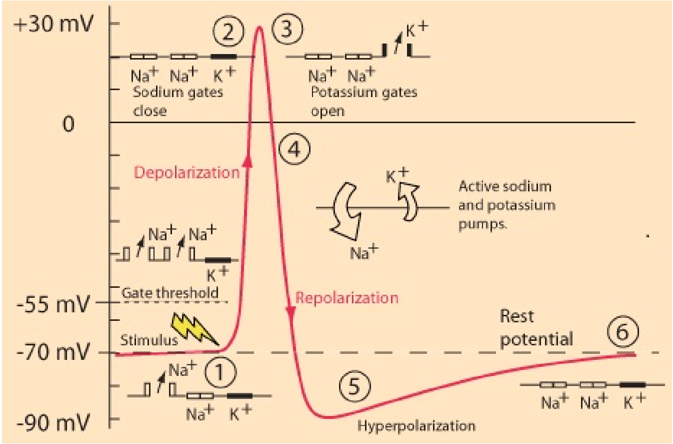


Fig. 12. Profile of an Action Potential (Spike). When a neurotransmitter binds with a dendrite, a reaction follows which opens up the Na channels, and Na$^+$ flows into the neuron. But the Na channels have a property which makes them expand as the membrane potential rises. This leads to a sharp rise in the potential, until the Na channels finally snap close, and the cell polarises again as the K channels open up. The K channels remain open even when the potential reaches the resting state, and that leads to hyperpolarisation. The leak channels then stabilise the potential towards the resting state.[12]

A. Why are Neural Networks important?

In fig.11, a comparison is shown between the world's fastest supercomputer and the human brain. For starters, look at the huge difference in power consumption. Add to that the fact that even with such a high processing power, the Tianhe-2 cannot do half the things that we can - Semantic representation, Learning, etc. This clearly indicates that the human brain is a very advanced system, and to take a giant leap in the computing domain, we need to study the brain.

The brain is composed of two distinct cells - Neurons and Glia. Glial cells are usually thought of as the maintenance and regulatory systems of the brain [10]. The Neurons interact with each other through a huge amount of inter-connections (synapses), and render the brain capable of doing a variety of activities that go beyond just survival. The brain has around 100 billion neurons and 100 trillion synapses [12]. Scientists believe that it is this dense networking that gives rise to the complex behavior we exhibit.

The study of Neural Networks (NNs) is gaining momentum.

B. Neurons

A neuron is a biological cell which has three distinct parts - the Dendrites, the Soma, and the Axon. The Dendrites are the inputs to the cell, the Axon the output, and the Soma is where the processing (change in membrane potential) occurs [11]. When neurotransmitters reach a dendrite, they initiate a reaction (which involves the balance of Na and K ions in the cell) as shown in fig.12. The neurotransmitter release is in fact governed by these 'spikes'. Thus, neurons can be considered as analog-to-spike and spike-to-analog converters, as shown in fig.13. The token of information in a Neural Network is, therefore, a Spike.

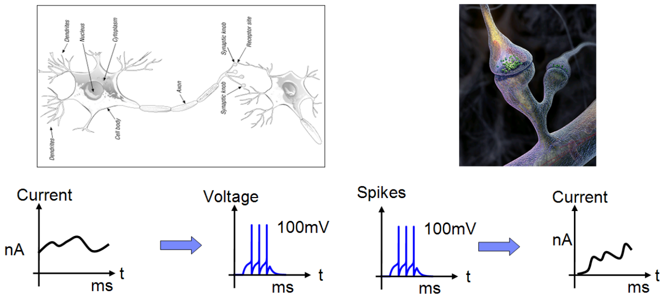


Fig. 13. The Analog-to-Spike conversion happens in the Neuron. The Spike-to-Analog conversion happens at the Synapse. (Courtesy: Prof. Bipin Rajendran)

There are two types of synapses - Excitatory, and Inhibitory. As the names suggest, they either lead to an increase the potential of the neuron, or inhibit the rise in potential, in response to a spike from the pre-synaptic neuron.

C. Neuron Models

To analyse the behavior of neurons, we need to identify the mathematical equations that model the dynamics. One of the most accurate models of a neuron, which takes into consideration the activity of the ion channels, is the Hodgkin-Huxley model [13]. But we can construct simpler models which get just the spike-time information right. This is because, for the stereotypical neurons we use to build SNNs, what matters during the neurotransmitter release at the synapse is the time of the action potential and not its shape. The Hodgkin-Huxley model (4 coupled differential equations) is computationally more intensive than the Leaky-Integrate-and-Fire (LIF) model (1 differential equation) [14], or the Izhikevich model (2 coupled differential equations) [15] that we usually use.

D. Applications of SNNs

The first and second generations of Neural Networks have found their way into many applications like Character Recognition, Image Compression, and Stock Market Prediction (Visit: http://stanford.io/1Eh7rWK, for further information).

People are still trying to figure out the regions where SNNs could speed-up computation, and there are only a handful of applications out there. The prominent application is to understand how the brain computes stuff by first trying to understand how simpler neural systems like the nervous system of the worm C. Elegans functions [16]. Progress in this field is steady, but a lot is yet to be learned.

VII. Looking Forward

We have explained the Control Scheme and its efficiency in controlling the quadcopter flight. We also introduced the field of Spiking Neural Networks. This was the work done in the Phase I of the BTech project.

In the second phase of this project, we will assess the computational superiority of the SNNs by building a SNN to take over as the Control Scheme of our quadcopter. During that time, we hope to learn about the areas where the SNN performs better than the classical algorithms, and therefore push the boundaries of the field of SNNs.

VIII. Acknowledgement

This report is in partial fulfillment of the requirements of the BTech Project I (EP493).

I thank Prof. Bipin Rajendran for his guidance on the project. This project wouldn't have been this structured without his insights. I also thank Sukanya Patil and Shruti Kulkarni who also participated in the discussions towards building the Control Scheme that is presented in the report.

Appendix

Constant	**Label**	**Value**
Radius of the frame	l	$0.24\,\mathrm{m}$
Thrust factor	b	$54.2 \times 10^{-6}\,\mathrm{Ns}^2$
Drag factor	d	$1.1 \times 10^{-6}\,\mathrm{Nms}^2$
Surface area of upper face	A	$0.25\,\mathrm{m}^2$
Total Mass	m	$1kg$
Acceleration due to gravity	g	$9.81\,\mathrm{m/s}^2$
Moment of Inertia along X	I_{XX}	$8.1 \times 10^{-3}\,\mathrm{Nms}^2$
Moment of Inertia along Y	I_{YY}	$8.1 \times 10^{-3}\,\mathrm{Nms}^2$
Moment of Inertia along Z	I_{ZZ}	$14.2 \times 10^{-3}\,\mathrm{Nms}^2$
Total rotational moment of inertia around the propeller axis	J_{TP}	$104 \times 10^{-6}\,\mathrm{Nms}^2$
Electric Motor constant	K_E	$6.3 \times 10^{-3}\,\mathrm{Vs.rad}^{-1}$
Mechanic Motor constant	K_M	$6.3 \times 10^{-3}\,\mathrm{NmA}^{-1}$
Motor resistance	R	0.6Ω
Drag factor	d	$1.1 \times 10^{-6}\,\mathrm{Nms}^{-2}$
Tuning Parameter 1	k_a	$6\,\mathrm{s}^{-1}$
Tuning Parameter 2	k_o	$2.3\,\mathrm{s}^{-1}$
Tuning Parameter 3	$k_{o\psi}$	$2\,\mathrm{s}^{-1}$
Tuning Parameter 4	k_{ac}	$61.73\,\mathrm{s}^{-1}$
Tuning Parameter 5	$k_{ac\psi}$	$35.21\,\mathrm{s}^{-1}$
Derived Parameter 1	f	$1.05 \times 10^{-4}\,\mathrm{Vs}^2\mathrm{.rad}^{-2}$
Derived Parameter 2	$g1$	$6.3 \times 10^{-2}\,\mathrm{Vs.rad}^{-1}$

References

[1] Wikipedia Contributers, *Quadcopter*. Wikipedia, The Free Encyclopedia, 5 Nov, 2014.

[2] G. Hoffmann, H. Huang, S. Waslander and C. Tomlin, *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. AIAA Guidance, Navigation and Control Conference and Exhibit, South Carolina, 2007.

[3] T. Bresciani, *Modelling, Identification and Control of a Quadrotor Helicopter*. Department of Automatic Control, Lund University, 2008.

[4] S. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 2nd edition, 1999.

[5] S. Fux, *Development of a planar low cost Inertial Measurement Unit for UAVs and MAVs*. Autonomous Systems Lab, ETH Zurich, 2008.

[6] We wanted our quadcopter to execute a foraging task. That task does not require exact coordinate information. All we have to control is the velocity. So, the Flight Trajectory Module provides the Control Module with instructions about the instantaneous velocity required to forage and overcome obstacles. The task of the Control Module is to robustely implement the instructions with a short response time.

[7] We would like to implement the control scheme as a Spiking Neural Network in the future. And SNNs have spike update rates of about 100Hz. A 100Hz update rate is just to check that the quadcopter control is possible at low update rates.

[8] M. Looney, *Analyzing Frequency Response of Inertial MEMS in Stabilization Systems*. Analog Dialogue 46-07, 2012.

[9] W. Maass, *Networks of Spiking Neurons: The Third Generation of Neural Network Models*. Neural Networks, Vol.10, No.9, Elsevier Science Ltd., Great Britain, 1997.

[10] M. Temburni, *New functions for Glia in the Brain*. PNAS 2001 98 (7) 3631-3632, 2001.

[11] This is a simplistic picture. And the neurons we are considering here are 'stereotypical'. There is a huge variety among neurons, but scientists agree that Spikes are the predominant token of information in the Human Nervous System.

[12] E. Kandel, J. Schwartz and T. Jessell, *Principles of Neural Science*. McGraw-Hill Medical, 4th edition, 2000.

[13] A. Hodgkin and A. Huxley, *A quantitative description of membrane current and its application to conduction and excitation in nerve*. J Physiol. Aug 28, 1952; 117(4): 500544, 1952.

[14] L. Lapicque, *Recherches quantitatives sur l'excitation e'lectrique des nerfs traitee comme une polarization*. J. Physiol. Pathol. Gen. 9:620, 1907.

[15] E. Izhikevich, *Simple Model of Spiking Neurons*. IEEE Transactions on Neural Networks, Vol. 14, No. 6, 2003.

[16] A. Bora, A. Rao and B. Rajendran, *Mimicking the worm An adaptive spiking neural circuit for contour tracking inspired by C. Elegans thermotaxis*. IJCNN 2014, pgs.2079 - 2086, 2014.

[17] J. Shepherd and K. Tumer, *Robust Neuro-Control for a Micro Quadrotor*. GECCO'10, Oregon, USA, 2010.