Spiking Neural Network-based Flight Controller

Diego Chavez Arana[1]
Omar Garcia Alcantara[2]
Luis Rodolfo Garcia Carrillo[1]
Ignacio Rubio Scola[3]
Eduardo S. Espinoza[2]
Andrew T. Sornborger[4]

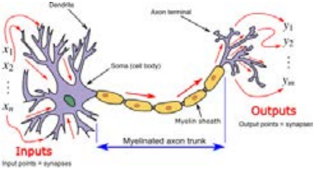
[1]New Mexico State University
[2]UMI-LAFMIA CONAHCYT-CINVESTAV
[3]INTI-CONICET-UNR
[4]Los Alamos National Laboratory

April 25, 2024

Motivation

Spiking Neural Networks (SNNs)

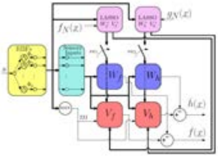
- SNNs inspired in the function of the mammalian brain.
- Energy and data-efficient alternative to Artificial Neural Networks (ANNs)
- Develop on neuromorphic computational architectures (Loihi, IBM's Truenorth, etc.)



Source: Wikipedia

Research of SNNs in Control Systems

- Stabilization of tracking error with a biologically plausible Limbic system inspired control (LISIC) (Rubio Scola, Garcia Carrillo, 2023)
- Spiking Neural Network-based Control Applied to a classical control system platform. (Chavez Arana, Garcia Carrillo, Sornborger 2022)

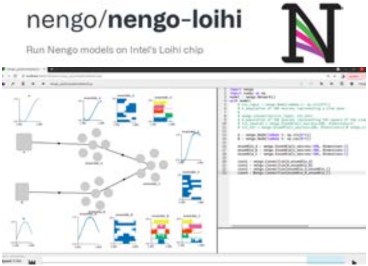


Source: Rubio Scola, Garcia Carrillo, and Hespanha, 2023

Objective

Develop an SNN-based controller to perform spatial stabilization and trajectory tracking of an Unmanned Aircraft System

- Used Proportional-Derivative (PD) control laws as its foundational framework
- Adoption of Neural Engineering Framework (NEF) through Nengo Python API
- Proposed controller effectiveness was evaluated using a flight simulation environment (X-Plane)



UAS dynamics: Mathematical model based on Newton-Euler formalism

$$\dot{\boldsymbol{\xi}} = \boldsymbol{V} \tag{1}$$

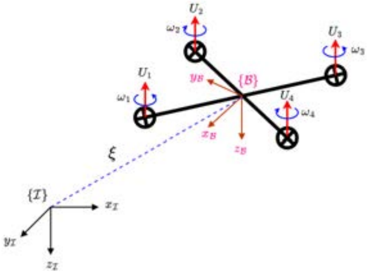
$$m\dot{\boldsymbol{V}} = \boldsymbol{R}\boldsymbol{F} \tag{2}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\hat{\boldsymbol{\Omega}} \tag{3}$$

$$\mathbb{I}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega} \times \mathbb{I}\boldsymbol{\Omega} + \boldsymbol{\Gamma} \tag{4}$$

- $\boldsymbol{F} \in \mathbb{R}^3$ - Total force
- $\boldsymbol{\Gamma} \in \mathbb{R}^3$ - Total torque on vehicle
- $\boldsymbol{V} = (\dot{x}, \dot{y}, \dot{z})^T$ - Translational velocity
- $\hat{\boldsymbol{\Omega}}$ - skew-symmetric matrix from $\hat{\boldsymbol{\Omega}}\boldsymbol{a} = \boldsymbol{\Omega} \times \boldsymbol{a}$

$\Omega \times \mathcal{I}\Omega$ is the direction and magnitude of the angular velocity and rotational

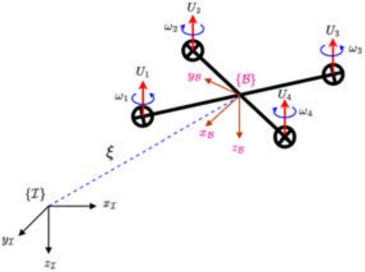


Forces acting on the UAS, inertial frame $\mathbb{I}$, and body frame $\mathcal{B}$

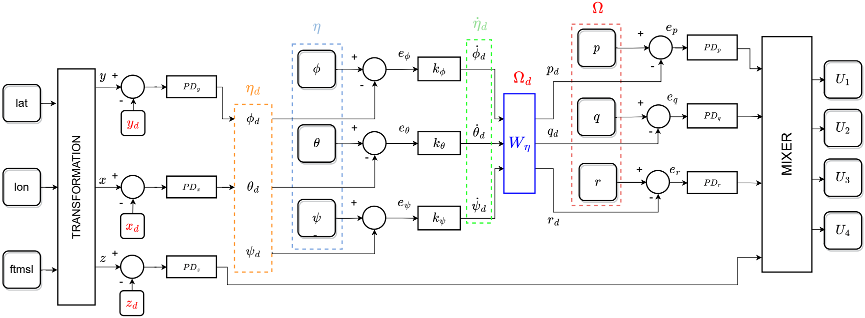
UAS dynamics

- Reduced model does not consider all the effects acting on the vehicle

$$
\begin{aligned}
\ddot{x} &= -u\left(\cos\phi\cos\psi\sin\theta + \sin\phi\sin\psi\right)/m \\
\ddot{y} &= -u\left(\cos\phi\sin\theta\sin\psi - \cos\psi\sin\phi\right)/m \\
\ddot{z} &= -u\left(\cos\theta\cos\phi\right)/m + g \\
\ddot{\phi} &= M_\phi/I_{xx} + \phi\tan\theta M_\theta/I_{xx} + \tan\theta M_\psi/I_{zz} \\
\ddot{\theta} &= \cos\phi M_\theta/I_{yy} - \sin\phi M_\psi/I_{zz} \\
\ddot{\psi} &= \phi M_\theta/(I_{yy}\cos\theta) + M_\psi/(I_{zz}\cos\theta)
\end{aligned}
\tag{5}
$$



Classic PD control: inner and outer loops



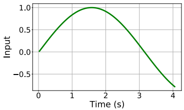
Neural Engineering Framework (NEF)

Our Spiking Neural Network Based Controller (SNNBC) was constructed using NEF. NEF proposes a way to transform a physical magnitude or signal into a spiking neuron firing rate.

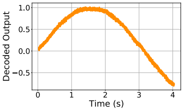
The **encoding** process of a vector representation in a neural population

$$a_i(\mathbf{x}(t)) = G_i \left[\alpha_i \left\langle \tilde{\zeta}_i \mathbf{x}(t) \right\rangle_l + J_i^{\text{bias}} \right]$$

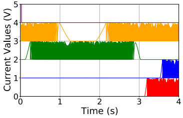
The neural populations represent a dynamic state over time through nonlinear encoding and linear decoding



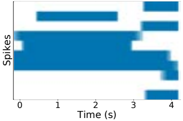
Input current



Output current



Filtered Spike Trains



Spike Trains

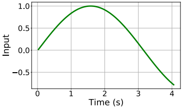
Neural Engineering Framework (NEF)

Our Spiking Neural Network Based Controller (SNNBC) was constructed using NEF. NEF proposes a way to transform a physical magnitude or signal into a spiking neuron firing rate.

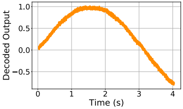
The **decoding** process translates the neural response into the desired output

$$\hat{\mathbf{x}}(t) = \sum_{i,n} \zeta_i \left(t - t_{in}\right)$$

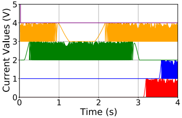
NEF uses a mix of decoding matrix weights (convolution operation with synaptic filter)



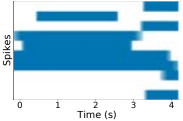
Input current



Output current



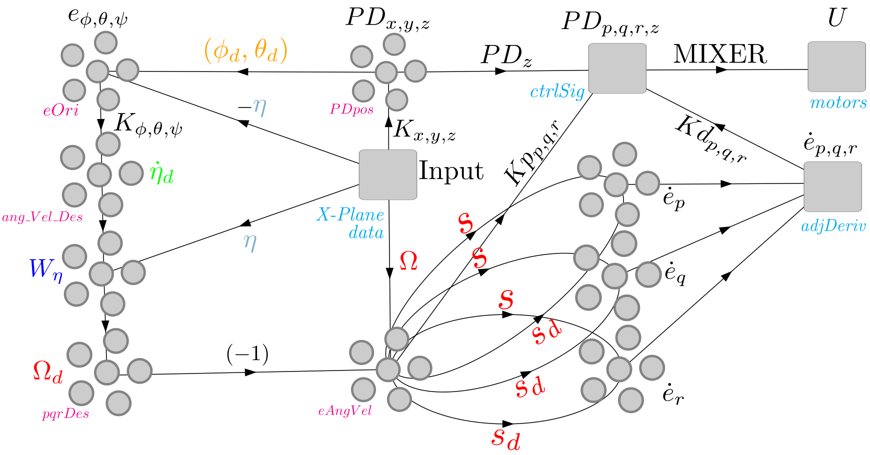
Filtered Spike Trains



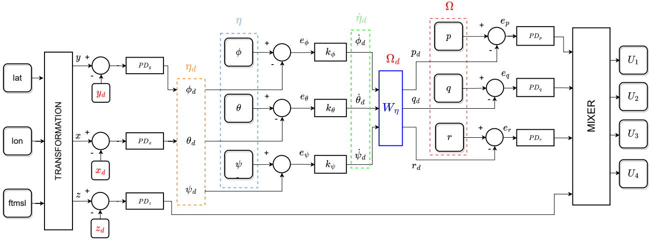
Spike Trains

Spiking Neural Network Based Controller (SNNBC)

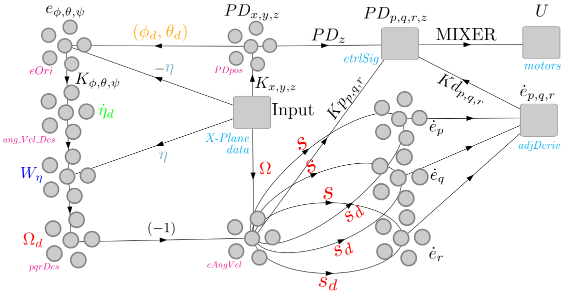
SNN PID control: inner and outer loops



Classic PD control: inner and outer loops



SNN PD control: inner and outer loops



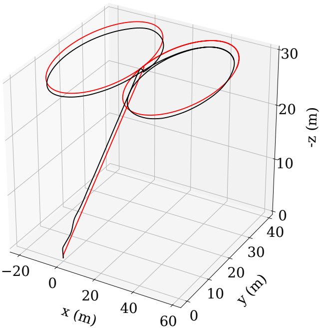
Experiments

The process integrates the UAS mathematical model of the quad rotorcraft in the simulation environment and implements the SNNBC within the simulator.

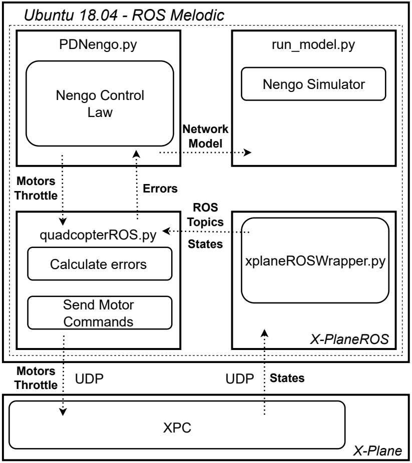
Mission profile:

- Take-off
- Climbing
- Tracking of a figure-8 reference

Parameter	kp	kd	k
x	0.09	0.208	—
y	-0.0936	-0.192	—
z	-0.096	-0.72	—
ϕ	—	—	3.9
θ	—	—	3.9
ψ	—	—	3.4
p	0.12	0.005	—
q	0.108	0.01	—
r	0.25	0.25	—



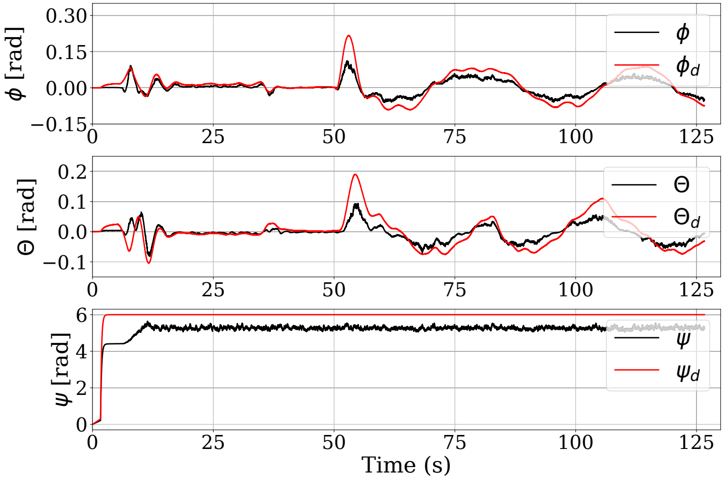
Communication structure



X-plane

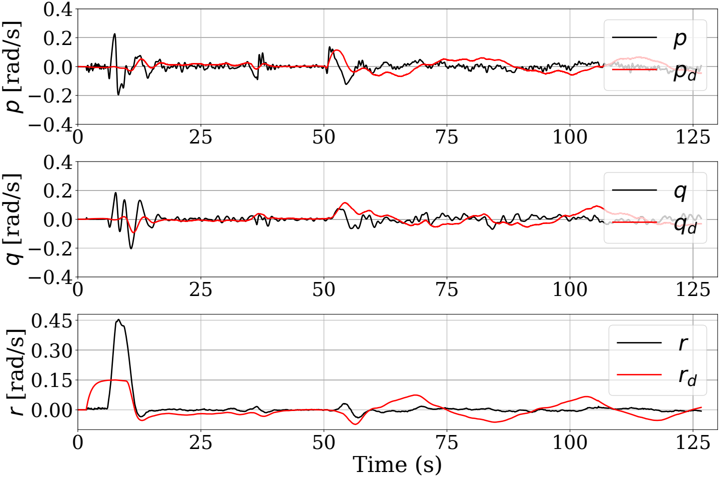
- Physics-based flight simulator

Euler angles: desired and measured states



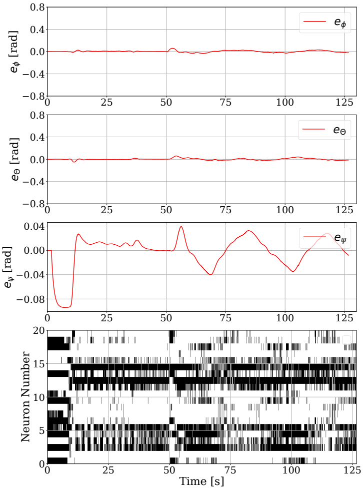
Euler angles vs. desired euler angles.

Angular velocities: desired and measured states

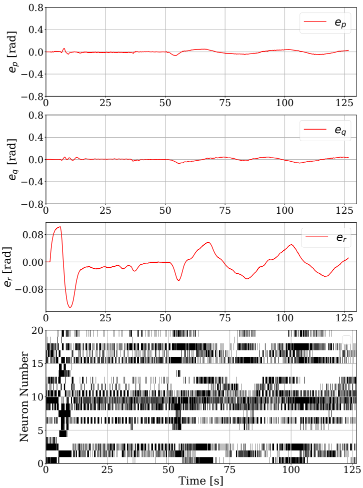


Angular velocities vs. desired angular velocities.

Results (Cont.)



Orientation errors and spiking activity.



Angular velocity errors and spiking activity.

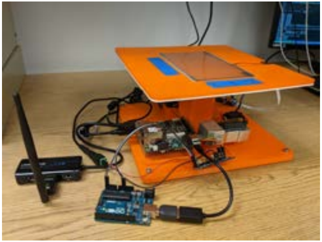
Conclusions

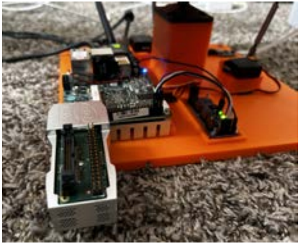
Outcomes:

- Demonstrated adaptability and performance of SNNBC throughout the autonomous flight mission of a subactuated UAS

Future Directions:

- Development of a SNNBC for UAS implemented on neuromorphic hardware and incorporating neuromorphic sensors (event-based camera)
- Writing low-level code using NxSDK to develop SNN on Loihi





Acknowledgments

Luis Rodolfo Garcia Carrillo

Andrew Sornborger

Ignacio Rubio Scola

Eduardo Steed Espinoza

Work supported by: