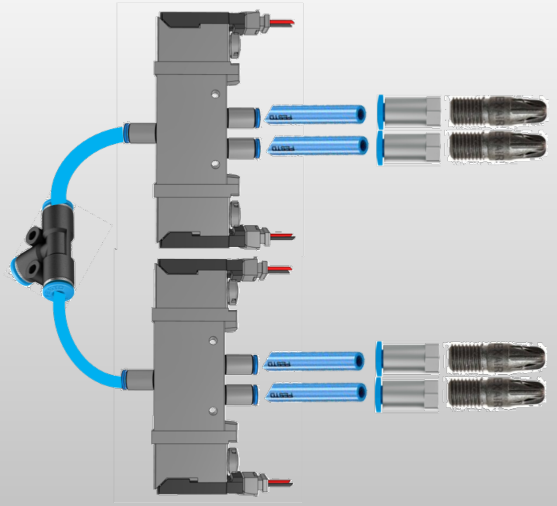
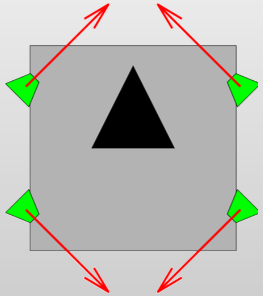
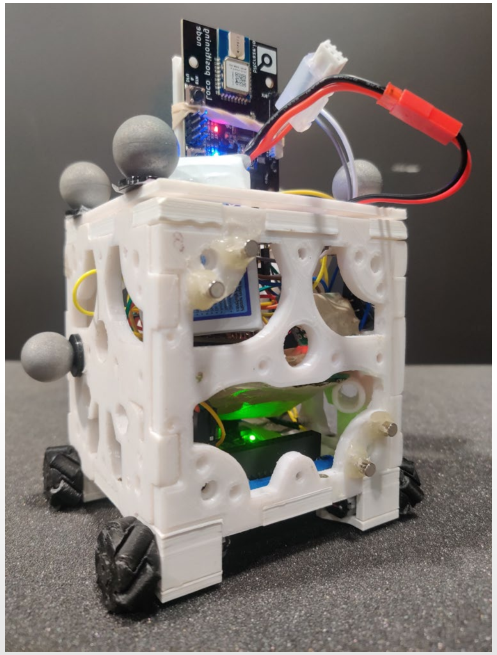
TPODS : Holonomic Motion Emulators

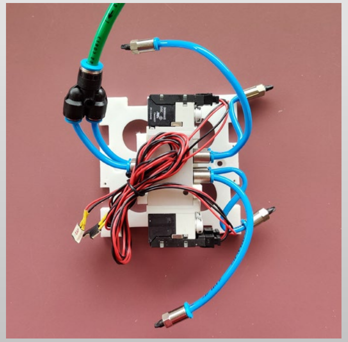
- Design Objective
  - 1U size
  - Holonomic planar motion
  - Portable and reproducible : does not need specific infrastructure
  - 3D printed and off-the-shelf parts

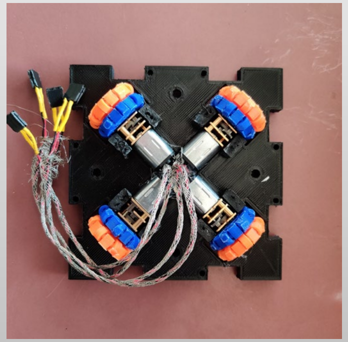


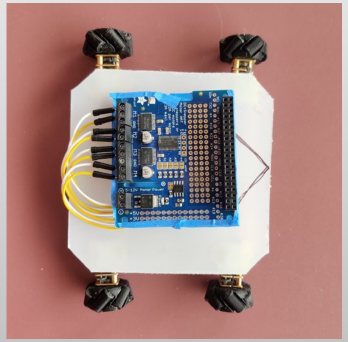


- Pneumatic Powertrain
  - Closely resembles holonomic motion
  - Limited on-board propellent
- Electric Powertrain
  - Longer runtime with holonomic capabilities









D. Parikh, A. H. Khowaja, N. Long, I. Down, J. McElreath, A. Bire, and M. Majji, “A Scalable Tabletop Satellite Automation Testbed: Design And Experiments,”

Relative State Estimation

- Objective : Enable docking of multiple 1U CubeSats
- Sensors
    - Ultra Wide Band radar ranging for long range operations
    - IMU / Gyro
    - Monocular vision for approach and docking
- Sensor fusion : Multiplicative / Error state Extended Kalman Filter

IMU
VectorNav VN-100

Ranging
UWB radar

Monocular Vision
OenMV H7 R1

Central Compute
Teensy 4.1

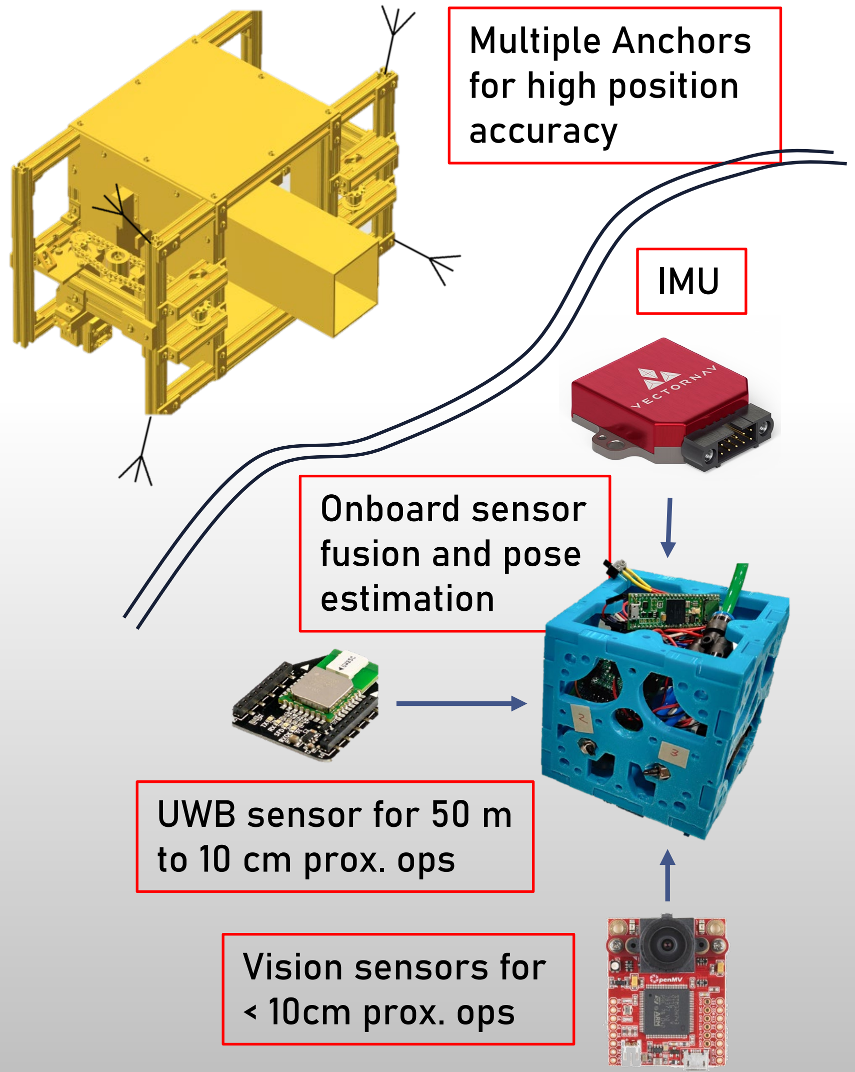
Current Amplifier

Solenoid
Valves

Electric
Motors

Telemetry
ESP 8266 WiFi

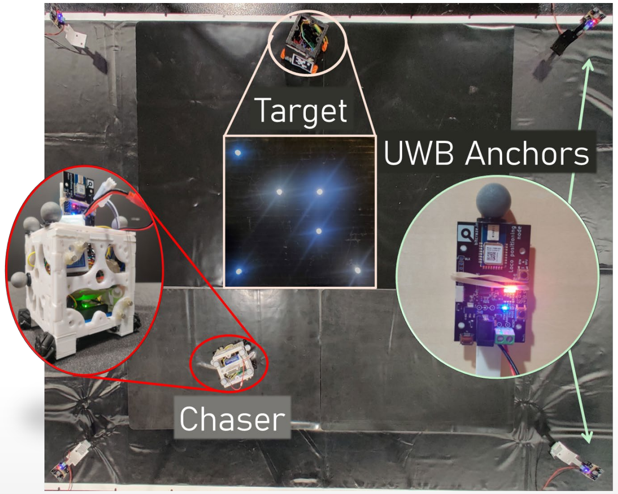
Ground Control

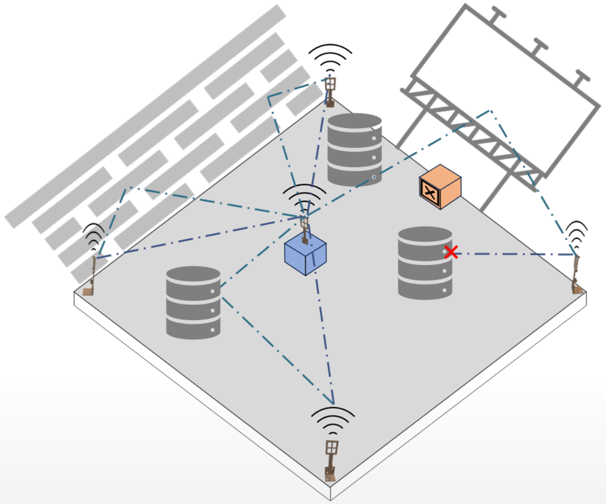


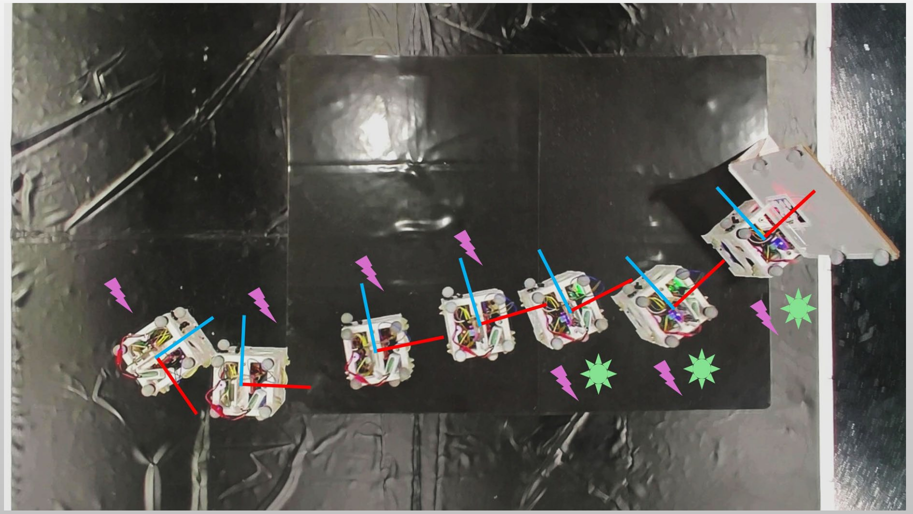
Pose estimation of CubeSats via sensor fusion and Error-State Extended Kalman Filter, Deep Parikh, Manoranjan Majji

Extensive Experimental Analysis

- On-board implementation
  - Joseph's form of covariance update for numerical stability
  - Eigen library for matrix algebra
  - RTOS for accurate timing and scheduling
- Practical Considerations
  - Mahalanobis distance-based outlier rejection to tackle multipath errors
  - Underweighting to produce bias free estimates
- Real-time MEKF based-pose estimator for the sensor fusion of range, angular rate and monocular vision measurements
  - Successful demonstration of docking with target having AprilTag







Computationally Efficient Model Predictive Controllers

$$\min_{\boldsymbol{u}\in\mathbb{R}^{p*n}} \ \boldsymbol{u}^T\boldsymbol{W}_u\boldsymbol{u} + \boldsymbol{u}^T\boldsymbol{S}_u^T\boldsymbol{W}_y\boldsymbol{S}_u\boldsymbol{u} + 2\left(\boldsymbol{x}_k\boldsymbol{S}_x^T\boldsymbol{W}_y\boldsymbol{S}_u\boldsymbol{u} - \boldsymbol{y}_r^T\boldsymbol{W}_y\boldsymbol{S}_u\boldsymbol{u}\right)$$

$$\begin{aligned}\boldsymbol{u} - V^u_{max}\epsilon &\le \boldsymbol{u}_{max}\\ -\boldsymbol{u} - V^u_{min}\epsilon &\le -\boldsymbol{u}_{min}\end{aligned}$$

$$\begin{aligned}\boldsymbol{S}_u\boldsymbol{u} - V^y_{max}\epsilon &\le \boldsymbol{y}_{max} - \boldsymbol{S}_x\boldsymbol{x}_k\\ -\boldsymbol{S}_u\boldsymbol{u} - V^y_{min}\epsilon &\le -\boldsymbol{y}_{min} + \boldsymbol{S}_x\boldsymbol{x}_k\end{aligned}$$

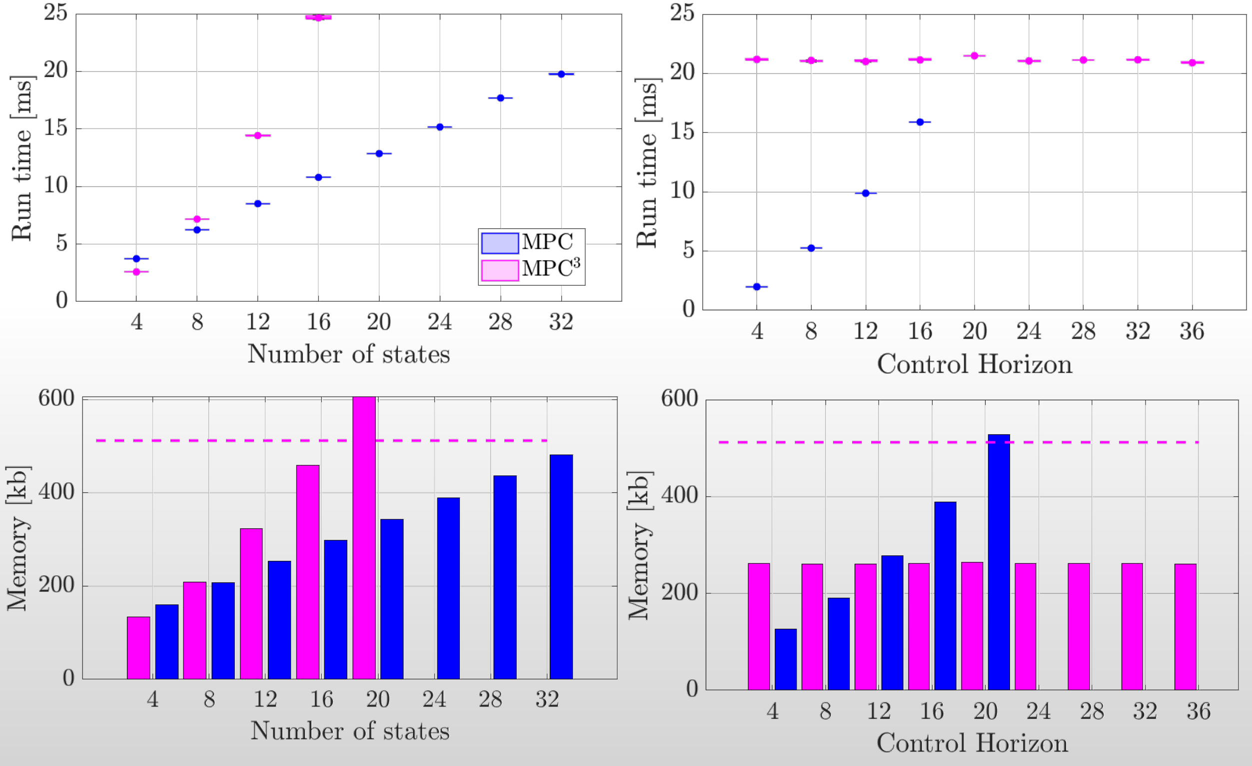
- SWaP constraints for aerospace applications

	MPC
State Equation	$\boldsymbol{x}_{k+1} = \boldsymbol{A}_d\boldsymbol{x}_k + \boldsymbol{B}_d\boldsymbol{u}$, $\boldsymbol{u}\in\mathbb{R}^m$, $\boldsymbol{x}\in\mathbb{R}^q$
Recursion	$\boldsymbol{y} = \boldsymbol{S}_x\boldsymbol{x}_k + \boldsymbol{S}_y\boldsymbol{u}$
Objective function	$\sum_{i=0}^{p-1}\left(\boldsymbol{u}_{k+i}^T\boldsymbol{W}_u\boldsymbol{u}_{k+i} + (\boldsymbol{y}_{k+i}-\boldsymbol{y}_r)^T\boldsymbol{W}_y(\boldsymbol{y}_{k+i}-\boldsymbol{y}_r)\right)$
Decision Variables	$\boldsymbol{u_k}\in\mathbb{R}^{p\times m}$
# of Equality Constraints	0
# of Inequality Constraints	$2p\times q$

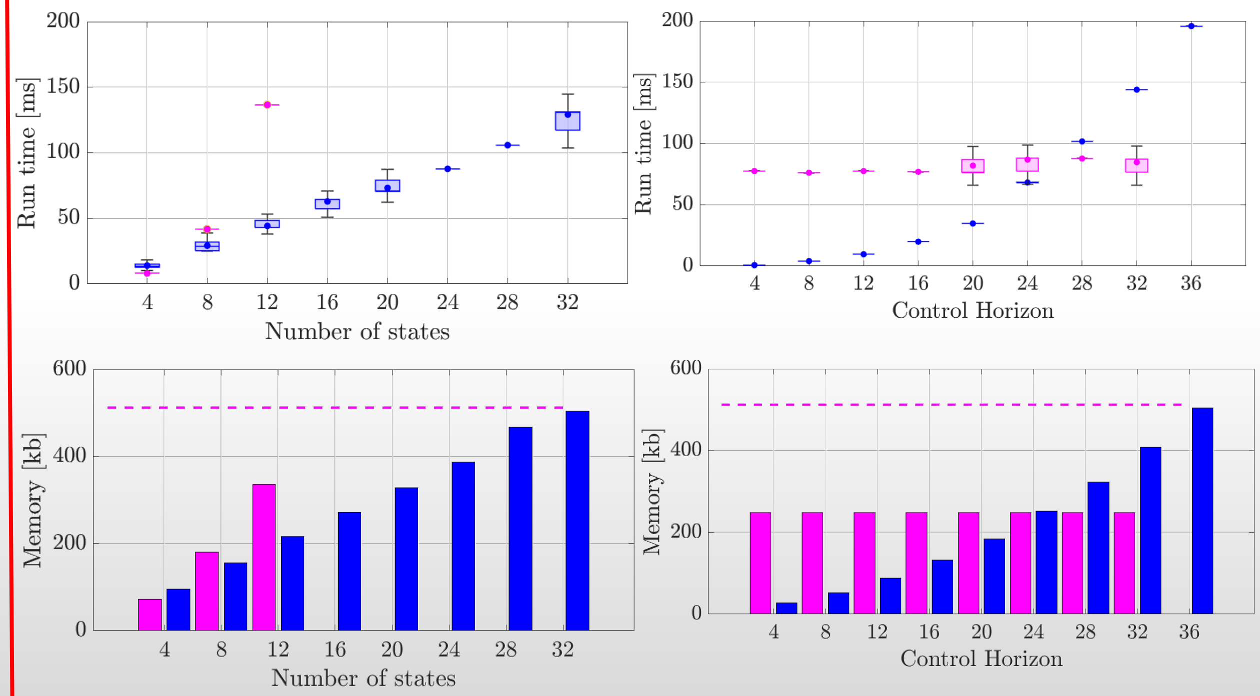
	Ground Vehicles			Aerial Vehicles			Space Vehicles		
	Yard Force / OpenMower	Serve Gen 3	Openpilot / Mazda	CrazyFlie 2.1+	Zipline P2	Skydio X10	NASA Astobee	NASA ACS3	NASA + APL DARTS
Size	22×15×11 in	27x22X22 in	180x73x66 in	3.6x3.6x1.1 in	50x40x10 in	31x26x6 in	13x13x13 in	390x390x12 in	71x72x102 in
Speed	2 m/s	5 m/s	45 m/s	1 m/s	30 m/s	16 m/s	0.5 m/s	Orbital	6000 m/s (rel.)
Prediction Horizon	6 m / 3 s	10 m / 2 s	90 m/ 2 s	0.03 m / 0.03 s	150 m / 5 s	48 m / 3 s	5 m / 10 s	Open Loop	6000 m / 1s
Compute Board	Raspberry Pi 4	NVIDIA Jetson Orin	Snapdragon 845	STM32F405	Dual NVIDIA Jetson Orin NX	NVIDIA Jetson Orin	NXP i.MX6 Duallite + Inforce IFC6501	NanoAvionics SatBus 3C2	Cobham LEON3FT + RTG4 FPGA
CPU RAM	4 GB	16 GB	8 GB	192 KB	2 x 16 GB	16 GB	1 GB + 2 GB	1 MB	16 MB+ 16 MB
Power	15 W	40 W	60 W	0.3 W	2 x 40W	40 W	3 W + 10 W	0.5W	2 W

MPC³: Model Predictive Controller using Chebyshev Collocation

OSQP (Running on Teensy 4.1)



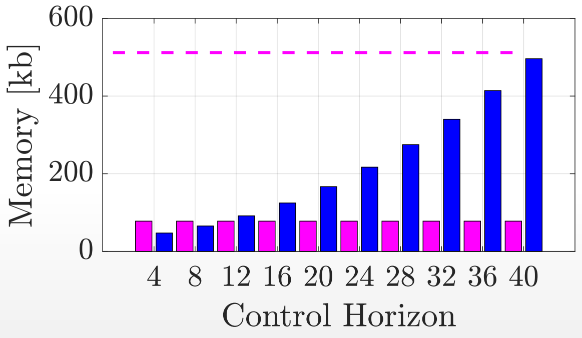
qpSwift (Running on Teensy 4.1)

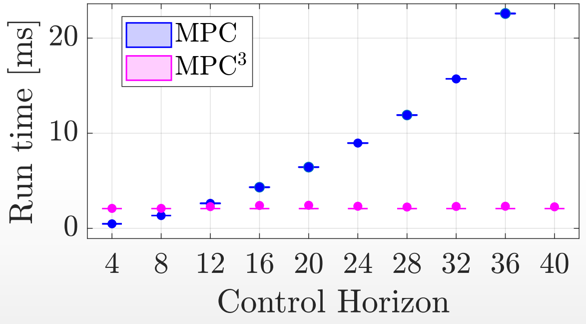


- MPC is superior for lower control horizons, generic / coupled dynamical systems
- MPC³ is better for higher control horizons, and for higher order dynamical systems

MPC^3 Applications and Experiments

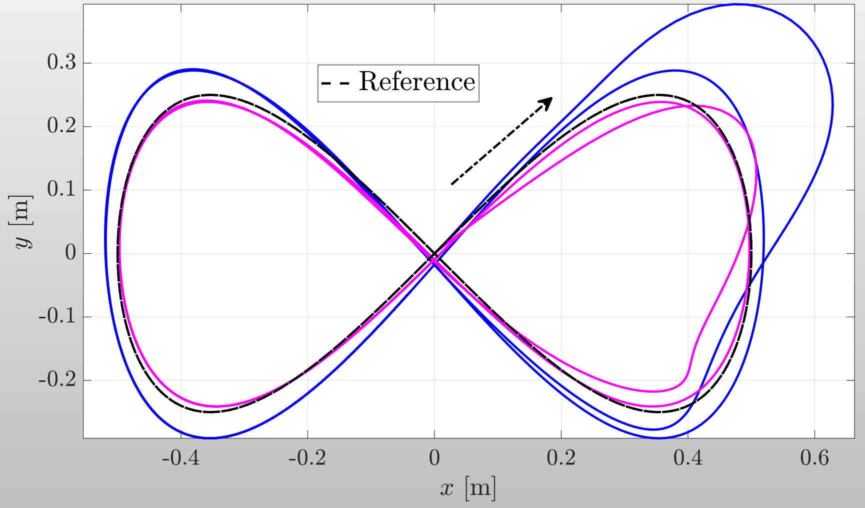
- Reference Trajectory Tracking for Quadrotors





MPC: $50hz, CH = 15, \delta t = 0.3s$

MPC^3: $50Hz, n = 5, \delta t = 0.4s$



- Collision Avoidance of TPODS