Tiny Robots: Edge Computational Challenges and Opportunities

Brian Plancher
Barnard College, Columbia University
brianplancher.com

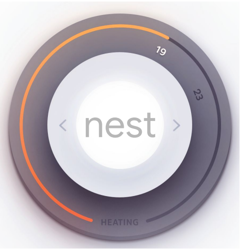
TinyML will soon be everywhere!

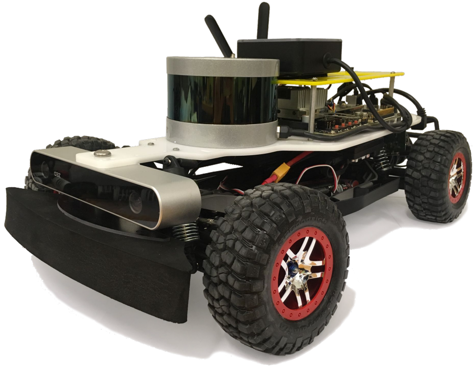
IoT 1.0:
Internet
of Things

↓

IoT 2.0:
Intelligence
on Things

Including on Robots!





Google Assistant

So what is Robotics?

Robotics is a **BIG** space



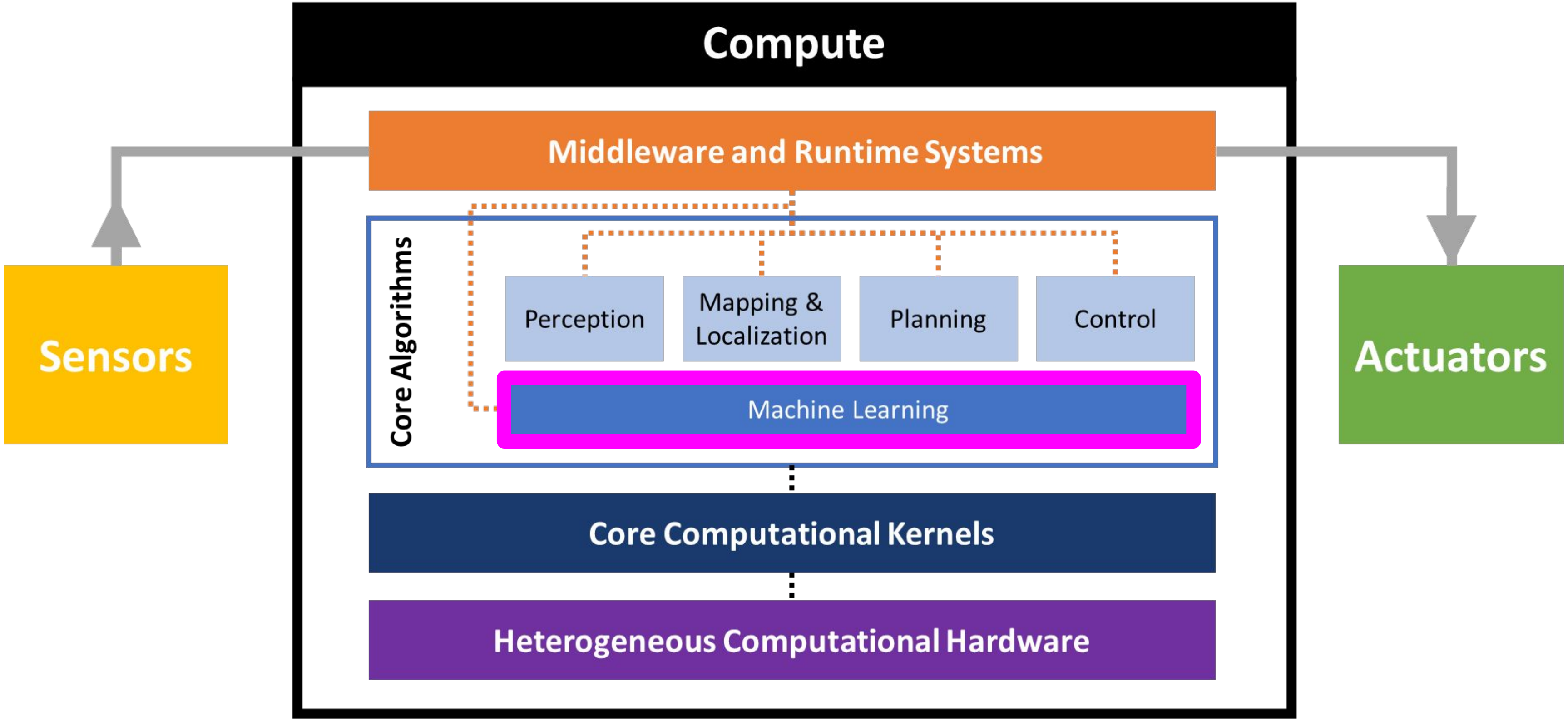




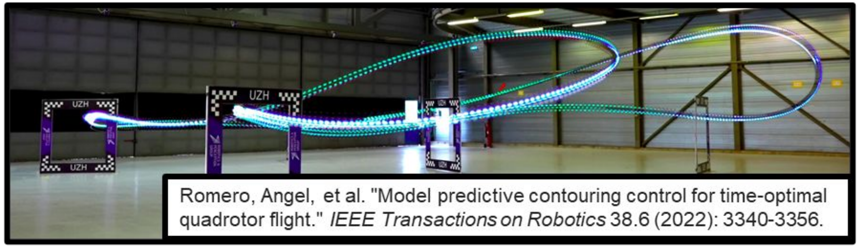
Autonomous Systems

~~Robotics~~ is a **BIG** space

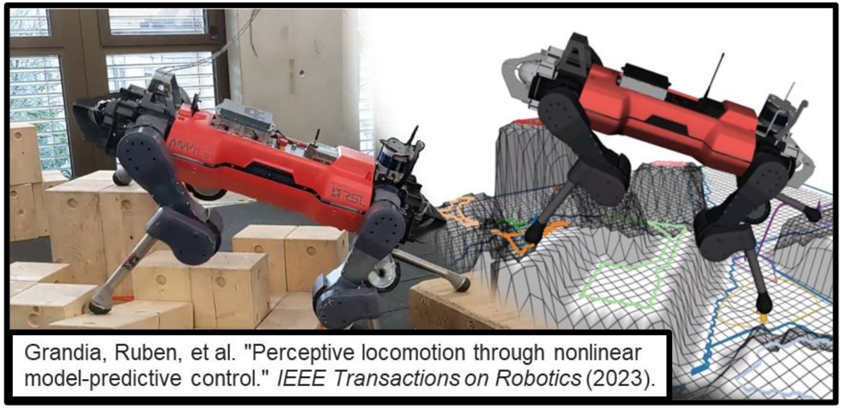
Robotics is a **BIG** space



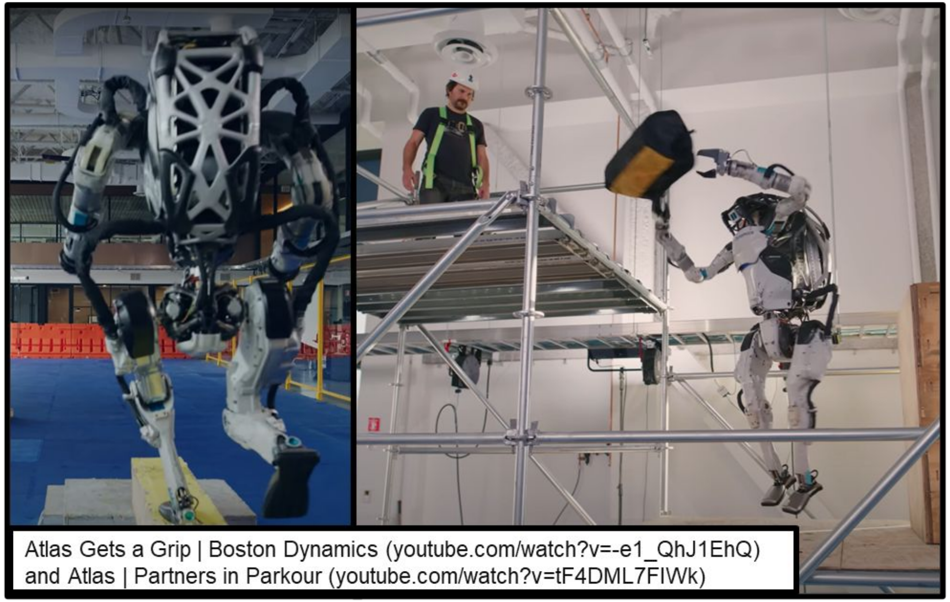
Robots can do amazing things…



Romero, Angel, et al. "Model predictive contouring control for time-optimal quadrotor flight." *IEEE Transactions on Robotics* 38.6 (2022): 3340-3356.



Grandia, Ruben, et al. "Perceptive locomotion through nonlinear model-predictive control." *IEEE Transactions on Robotics* (2023).

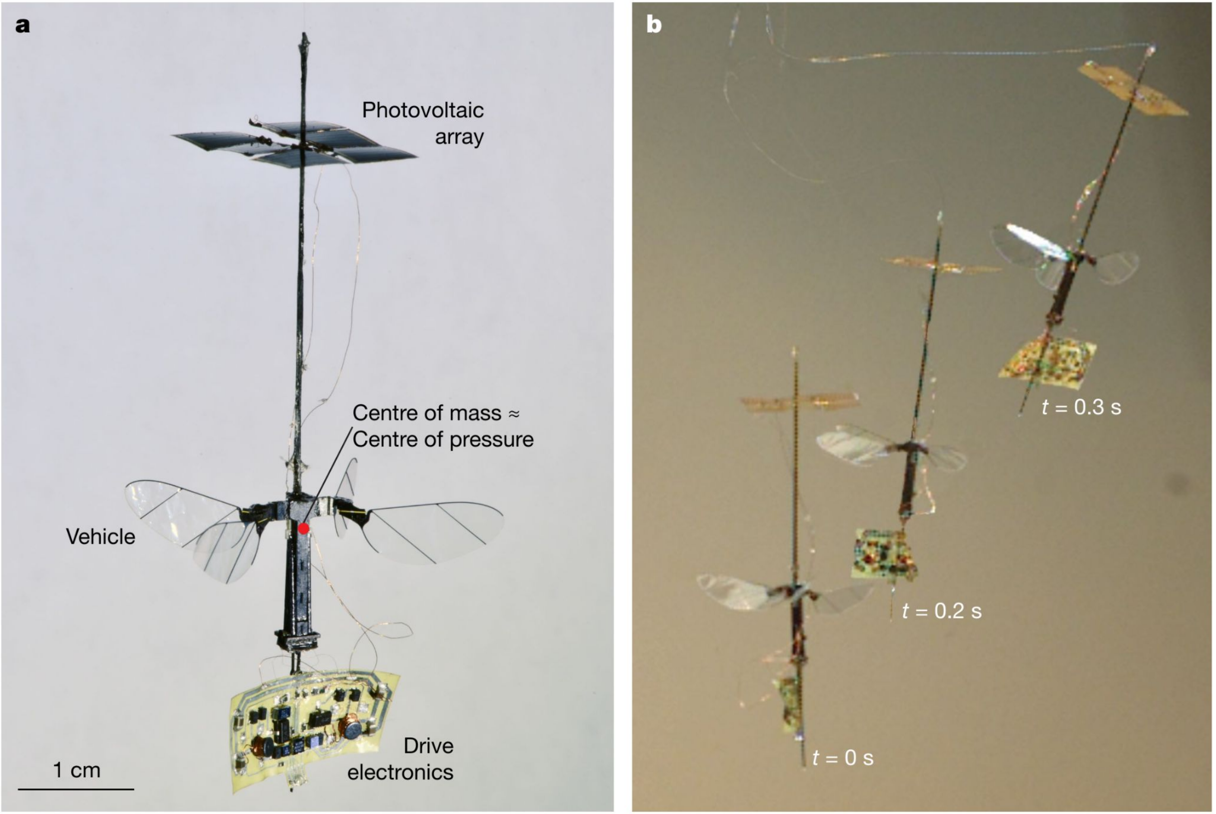


Atlas Gets a Grip | Boston Dynamics (youtube.com/watch?v=-e1_QhJ1EhQ) and Atlas | Partners in Parkour (youtube.com/watch?v=tF4DML7FIWk)

Robots can do amazing things…

… but they still have a long way to go!

Especially at small scales!



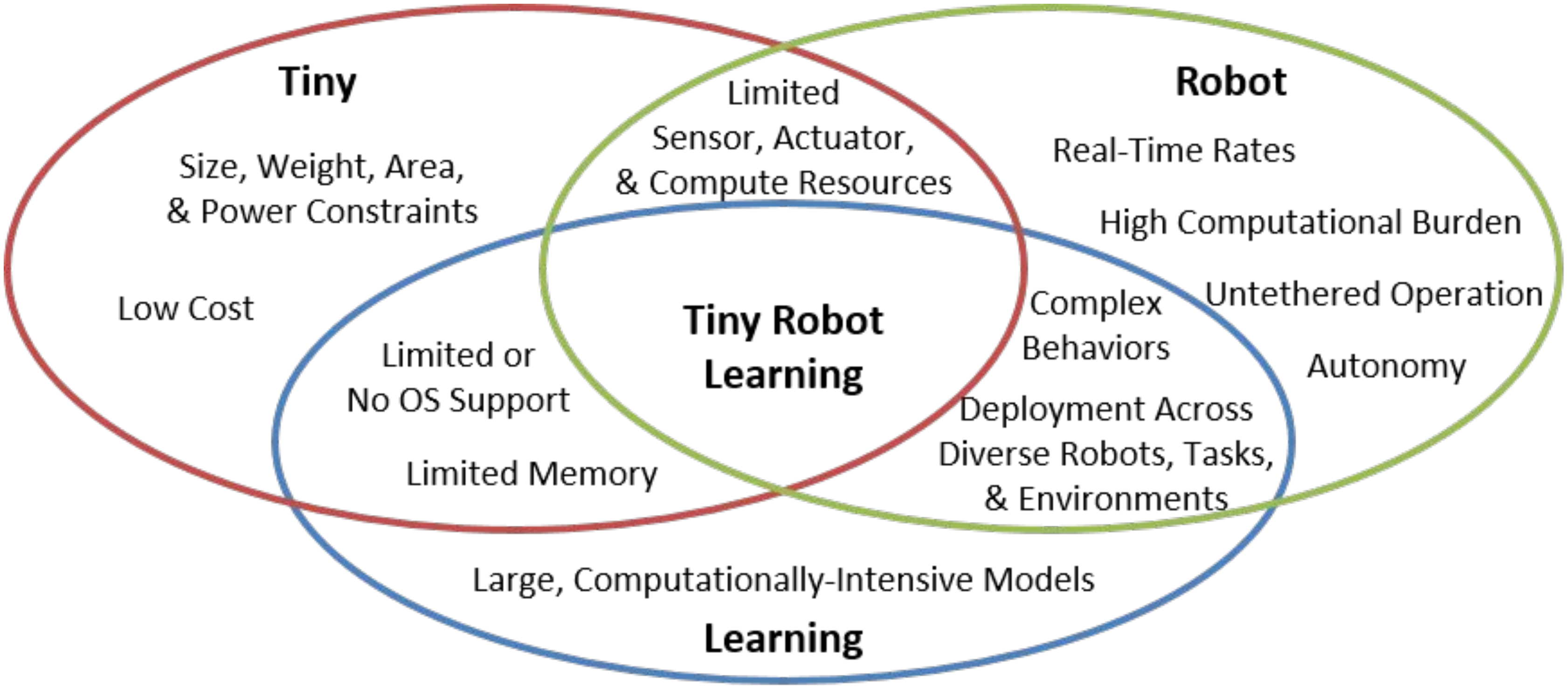
SWaP

Constrained

Size,
Weight,
and
Power

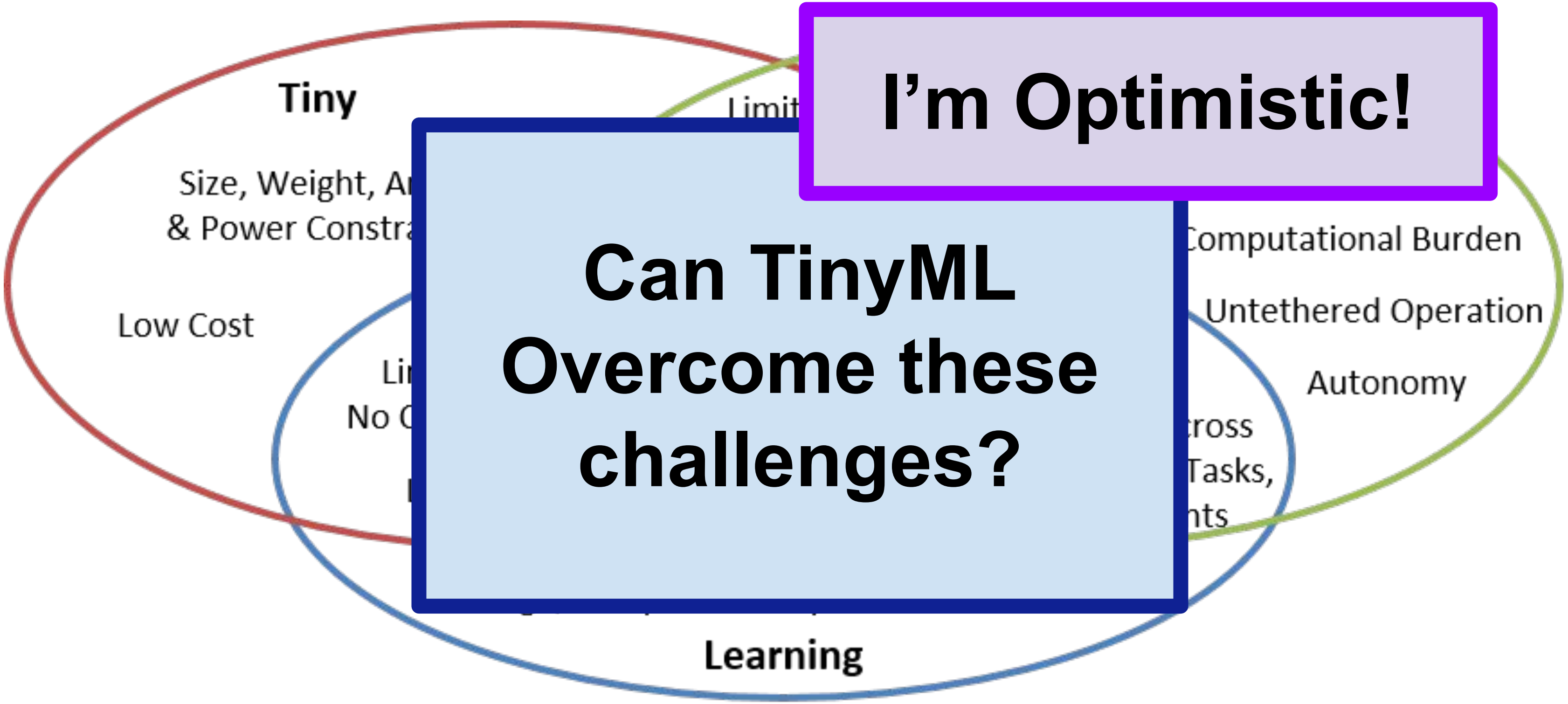
Jafferis, Noah T., et al. "Untethered flight of an insect-sized flapping-wing microscale aerial vehicle." Nature 570.7762 (2019): 491-495.

Especially at small scales!



Neuman, Sabrina M., et al. "Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots." 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS). IEEE, 2022.

Especially at small scales!



Neuman, Sabrina M., et al. "Tiny robot learning: Challenges and directions for machine learning in resource-constrained robots." 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS). IEEE, 2022.

Tiny Robots: Edge Computational Challenges and Opportunities

1. Microcontrollers can already compute more than you think! ← **TinyMPC**
2. TinyML is already being demonstrated for robotics! ← **tinyRL / Sniffy Bug**
3. On-Device Learning is coming to MCUs near you! ← **TyBox / TinyProp**

TinyMPC: Enabling state-of-the-art classical algorithms on Tiny Robots

Khai Nguyen*, Sam Schoedel*, Anoushka Alavilli, Elakhya Nedumaran, Brian Plancher, Zachary Manchester

The Barnard Accessible and Accelerated Robotics Lab

	Micro Platforms		Tiny Platforms				Full-Scale Platforms	
	Robobee 	HAMR-F 	Crazyflie2.1 	DeepPicar Micro 	PIXHAWK PX4 	Petoi Bittle 	Snapdragon Flight 	Unitree Go1edu 
Processor	ATtiny20 4-8 MHz 8-bit MCU	ATmega1284RF2 16MHz 8-bit MUC	STM32F405 168 MHz 32-bit M4 MCU	RP2040 133 MHz Dual-Core 32-bit M0+ MCU	STM32F765 216 MHz Dual-Core 32-bit M7 MCU	ESP32-WROOM-32D 240MHz Dual-Core 32-bit LX7 MCU	Qualcomm Snapdragon 801 2.15 GHz Quad-Core 32-bit CPU 450 MHz 32-pipeline GPU	Jetson Nano (x3) 1.43 GHz Quad-Core 64-bit CPU 921 MHz 128-core GPU
RAM	128 B	16 kB	196 kB	264 kB	512 kB	512 kB	2 GB	4 GB (x3)
Flash	2 kB	128 kB	1 MB	2 MB	2 MB	16 MB	32 GB	64-256 GB (via SD card x3)
Processor Power	0.015 W	0.045 W (with RF)	0.15 W	0.15 W	0.5 W	0.5-1 W	3-10 W	5-10 W (x3)

TinyMPC: Enabling state-of-the-art classical algorithms on Tiny Robots

Trade generality for speed and low-memory utilization

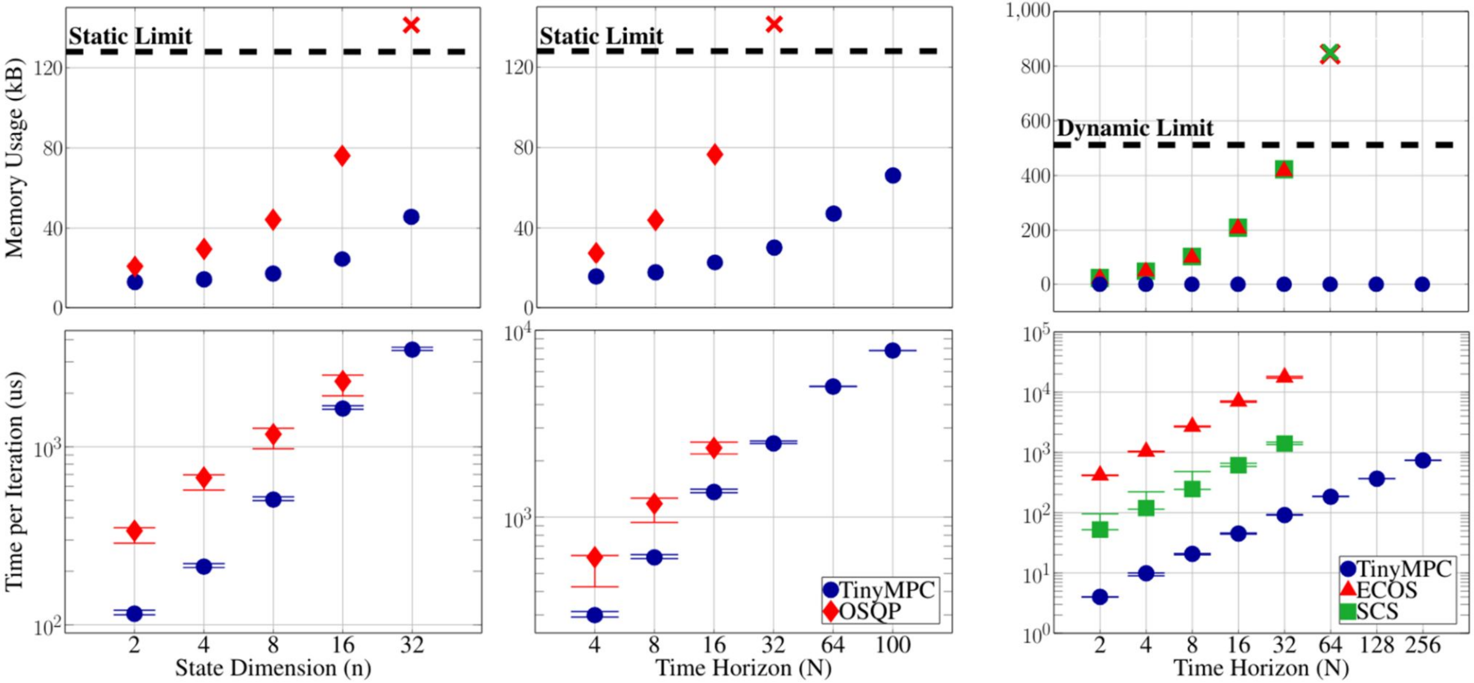
LQR

$$K_k = (R + B^\intercal P_{k+1} B)^{-1}(B^\intercal P_{k+1} A) \longrightarrow K_\infty$$
$$d_k = (R + B^\intercal P_{k+1} B)^{-1}(B^\intercal p_{k+1} + r_k)$$
$$P_k = Q + K_k^\intercal R K_k + (A - BK_k)^\intercal P_{k+1}(A - BK_k) \longrightarrow P_\infty$$
$$p_k = q_k + (A - BK_k)^\intercal (p_{k+1} - P_{k+1} B d_k) + K_k^\intercal (R d_k - r_k)$$

Offline vs. **Online**

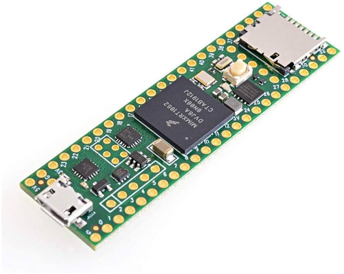
$$C_1 = \left(R + B^T P_\infty B\right)^{-1}$$
$$C_2 = (A - BK_\infty)^T$$
$$d_k = C_1(B^T p_{k+1} + r_k)$$
$$p_k = q_k + C_2 p_{k+1} - K_\infty^T r_k$$

TinyMPC: Enabling state-of-the-art classical algorithms on Tiny Robots

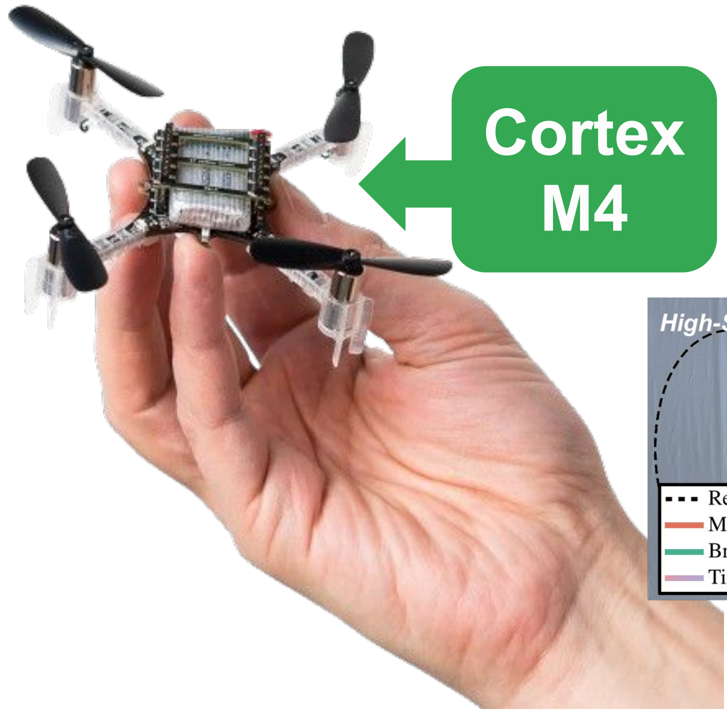


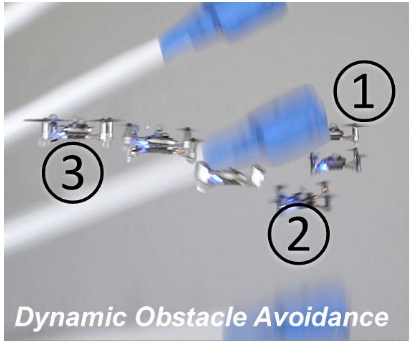
(a) Predictive safety filtering

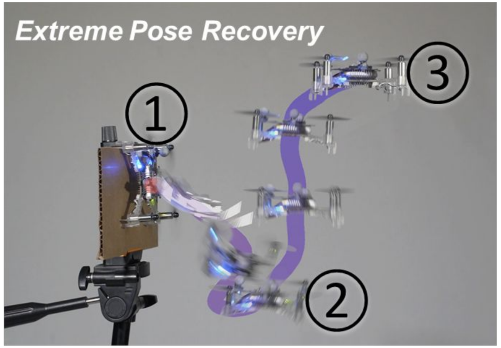
(b) Rocket soft-landing

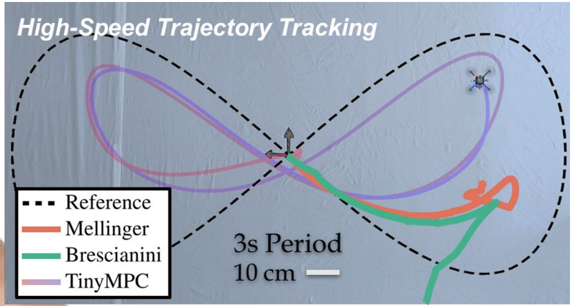


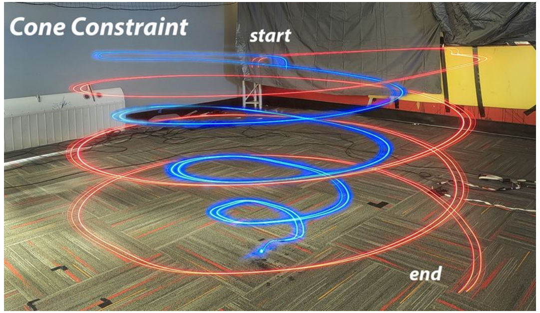
TinyMPC: Enabling state-of-the-art classical algorithms on Tiny Robots











TinyMPC: Enabling state-of-the-art classical algorithms on Tiny Robots

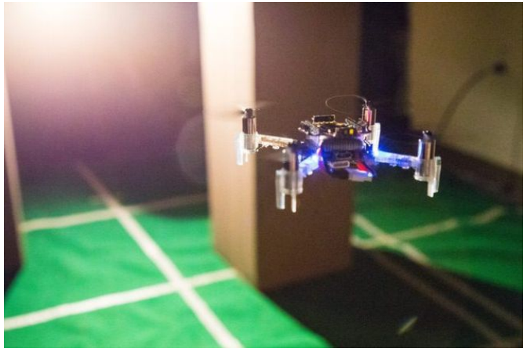
Tiny Robots: Edge Computational Challenges and Opportunities

1. Microcontrollers can already compute more than you think!
2. TinyML is already being demonstrated for robotics!
3. On-Device Learning is coming to MCUs near you!

- **TinyMPC** (→ 1)
- **tinyRL / Sniffy Bug** (→ 2)
- **TyBox / TinyProp** (→ 3)

Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments

Bardienus P. Duisterhof[1] Shushuai Li[1] Javier Burgués[2] Vijay Janapa Reddi[3] Guido C.H.E. de Croon[1]



Tiny Robot Learning (tinyRL) for Source Seeking on a Nano Quadcopter

Bardienus P. Duisterhof[1,3] Srivatsan Krishnan[1] Jonathan J. Cruz[1] Colby R. Banbury[1]
William Fu[1] Aleksandra Faust[2] Guido C. H. E. de Croon[3] Vijay Janapa Reddi[1]

TU Delft Delft University of Technology

Harvard John A. Paulson School of Engineering and Applied Sciences

Universitat de Barcelona

Sniffy Bug System design

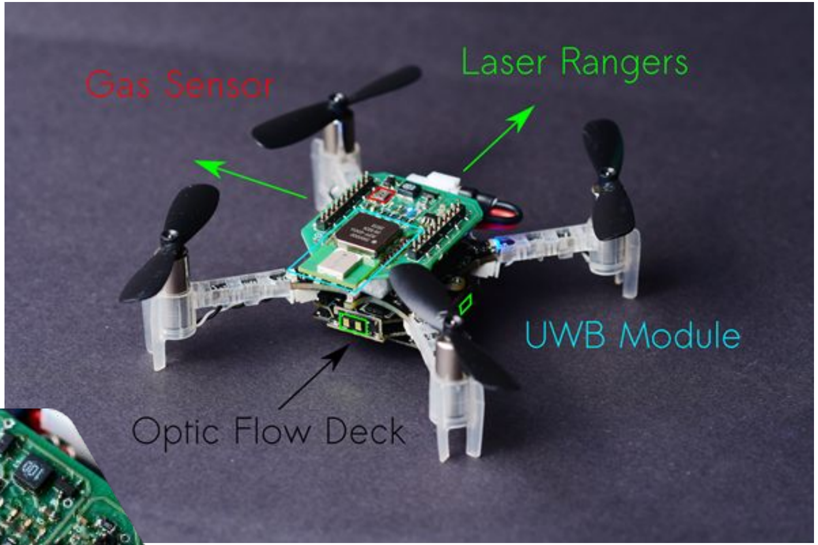
Requirements:

- Obstacle avoidance
- Odometry
- Gas sensing
- Relative ranging
- Communication

Payload:

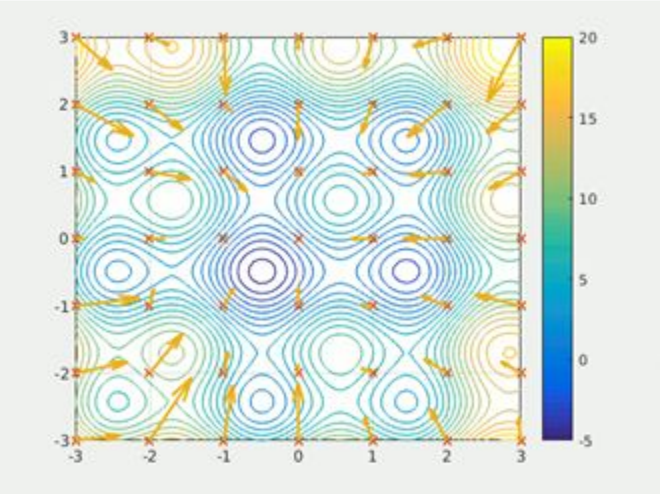
- Flow deck
- Multiranger deck
- Custom gas/UWB PCB

Weight: 37.5g



Sniffy Bug Algorithm and Results

Particle Swarm Optimization

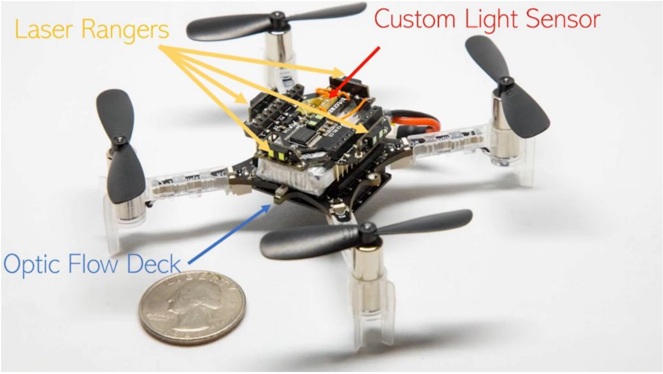


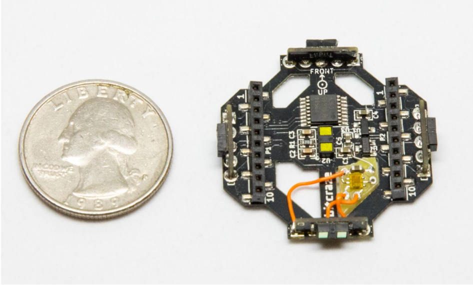
tinyRL System design

BitCraze CrazyFlie 2.1

- ARM Cortex-M4
- CPU: 1-core & 168 MHz
- RAM: 196 kB
- Storage: 1MB
- Available RAM: 33 kB
- Weight: 33 grams

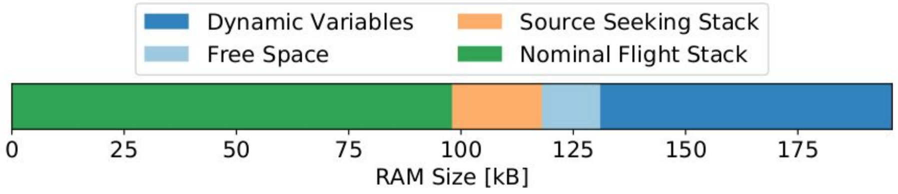
Training done in simulation.





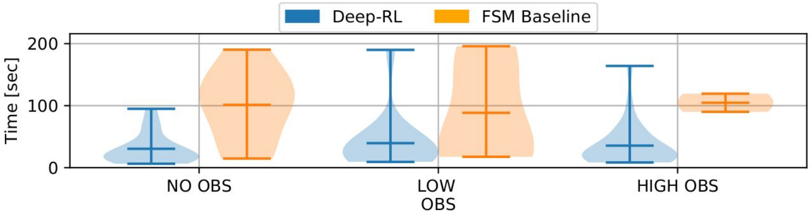
tinyRL Inference Implementation

- Obstacle avoidance requires low-latency inference.
- Libraries considered:
  - **TensorFlow Lite**, not fast enough.
  - **uTensor**, ran out of memory.
- Therefore, developed a custom lightweight C inference library!
- Result: capable of inference at up to 100Hz, higher than the sensor polling rate!

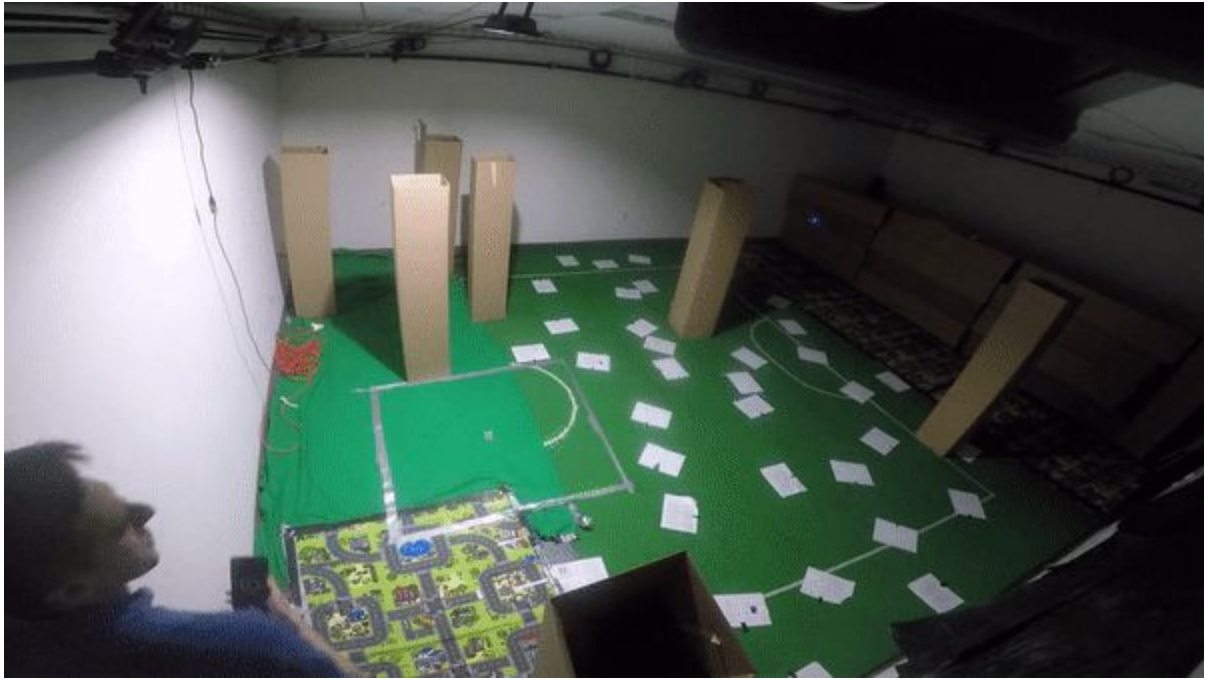


tinyRL Flight Test Results

- The deep-RL model reaches a **94%** success rate.
- The FSM Baseline reaches a **75%** success rate.
- Between obstacle densities, our policy found the source **55%-70% faster than the baseline**.
- The results show that our policy generalizes far beyond what was presented in simulation!



tinyRL Flight Test Results



Tiny Robots: Edge Computational Challenges and Opportunities

1. Microcontrollers can already compute more than you think! ← **TinyMPC**
2. TinyML is already being demonstrated for robotics! ← **tinyRL / Sniffy Bug**
3. On-Device Learning is coming to MCUs near you! ← **TyBox / TinyProp**

TyBox: An Automatic Design and Code Generation Toolbox for TinyML Incremental On-Device Learning

MASSIMO PAVAN and EUGENIU OSTROVAN, Politecnico di Milano, Italy
ARMANDO CALTABIANO, Truesense s.r.l., Italy
MANUEL ROVERI, Politecnico di Milano, Italy

On-Device Learning is Coming to MCUs near you!

TinyProp - Adaptive Sparse Backpropagation for Efficient TinyML On-device Learning

Marcus Rueb
Software Solutions / Artificial intelligence
Hahn-Schickard
Villingen-Schwenningen, Germany
Marcus.rueb@hahn-schickard.de

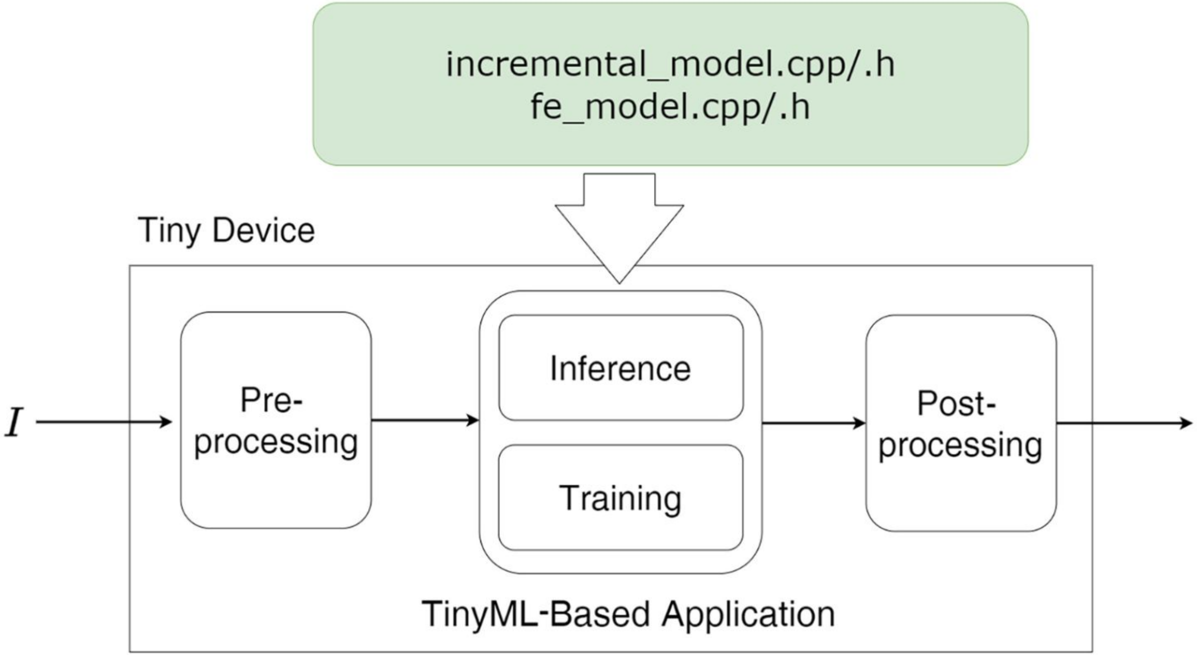
Daniel Maier
Software Solutions / Artificial intelligence
Hahn-Schickard
Villingen-Schwenningen, Germany
Daniel.Maier@hahn-schickard.de

Daniel Mueller-Gritschneder
Electronic Design Automation
TUM, Technical University Munich
Munich, Germany
daniel.mueller@tum.de

Axel Sikora
EMI
University of Applied Sciences Offenburg
Offenburg, Germany
axel.sikora@hs-offenburg.de

TyBox: An Automatic Design and Code Generation Toolbox for TinyML Incremental On-Device Learning

MASSIMO PAVAN and EUGENIU OSTROVAN, Politecnico di Milano, Italy
ARMANDO CALTABIANO, Truesense s.r.l., Italy
MANUEL ROVERI, Politecnico di Milano, Italy



TyBox: An Automatic Design and Code Generation Toolbox for TinyML Incremental On-Device Learning

MASSIMO PAVAN and EUGENIU OSTROVAN, Politecnico di Milano, Italy
ARMANDO CALTABIANO, Truesense s.r.l., Italy
MANUEL ROVERI, Politecnico di Milano, Italy

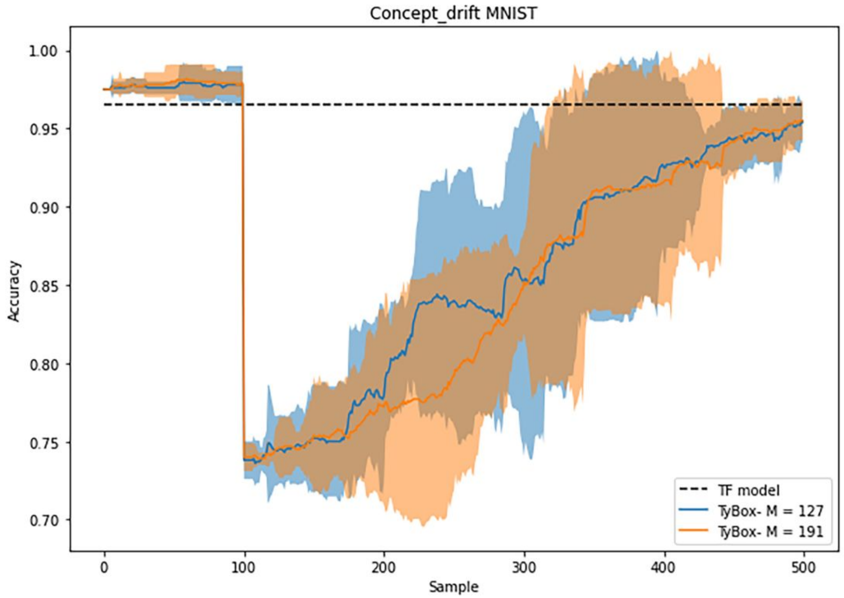
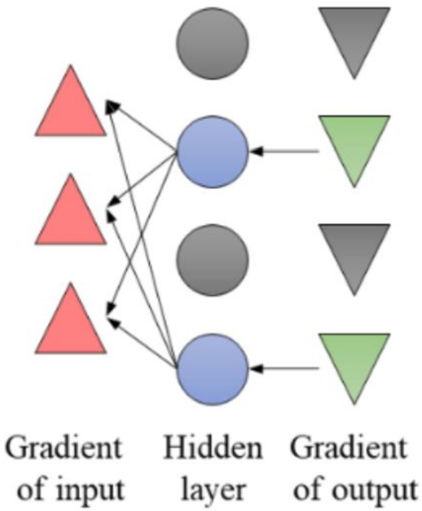


Fig. 5. The classification accuracy on the abrupt concept drift learning experiment for the image multi-class classification setting.

5. Sparse back Propagation

(Top k = 2)



TinyProp - Adaptive Sparse Backpropagation for Efficient TinyML On-device Learning

Marcus Rueb
Software Solutions / Artificial intelligence
Hahn-Schickard
Villingen-Schwenningen, Germany
Marcus.rueb@hahn-schickard.de

Daniel Maier
Software Solutions / Artificial intelligence
Hahn-Schickard
Villingen-Schwenningen, Germany
Daniel.Maier@hahn-schickard.de

Daniel Mueller-Gritschneder
Electronic Design Automation
TUM, Technical University Munich
Munich, Germany
daniel.mueller@tum.de

Axel Sikora
EMI
University of Applied Sciences Offenburg
Offenburg, Germany
axel.sikora@hs-offenburg.de

MNIST fine-tuning	Baseline	top-k 6500	top-k 12000	top-k 17000	top-k 30000	top-k 66000	TinyProp
Accuracy (%)	96.4	85.2	85.9	86.0	89.9	91.9	96.1
Back propagation Ratio	1	0.1	0.15	0.2	0.33	0.66	0.07
Runtime ESP32 per Epoch	150.15s	25.024s	30.03s	37.51s	50s	100.1s	18,1s
Acceleration	1x	6x	5x	4x	3x	1.5x	8,3x

Tiny Robots: Edge Computational Challenges and Opportunities

1. Microcontrollers can already compute more than you think!
2. TinyML is already being demonstrated for robotics!
3. On-Device Learning is coming to MCUs near you!

TinyMPC

tinyRL / Sniffy Bug

TyBox / TinyProp

Tiny Robots: Edge Computational Challenges and Opportunities

I'm **Optimistic** that **TinyML** can help **Overcome SWaP** Constraints for **Robotics**

Size, **W**eight, **a**nd **P**ower

Initial Results are Already Positive!

Brian Plancher
Barnard College, Columbia University
brianplancher.com