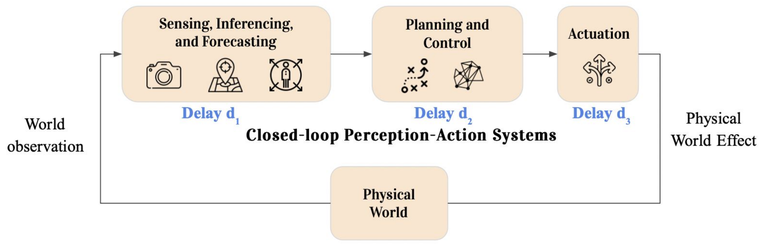
Adapting Control Computation for Optimizing Application Performance under Dynamic Delays in Distributed Settings

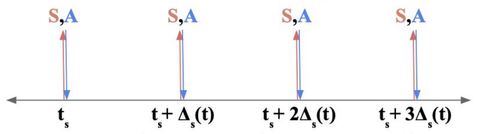
Pragya Sharma, Mani B. Srivastava (NESL UCLA)

Objectives

- **Research Innovation:** Investigate adaptive control placement with the objective of optimzing system performance in closed-loop Perception-Action systems under dynamic delays. Systems such as, anti-missile guidance and vehicle/human following, have a repeating PCA cycle (shown below) where the effect of the pipeline changes continuously based on world observation.
- **Army/Military Relevance:** While this work focuses on closed-loop PCA systems, the model can be extended to open-loop (one-shot missile interception, intruder capture) and situational awareness (adversary tracking using radar or cameras). This would help the soldier optimize battlefield knowledge and automate such PCA systems with minimal input.

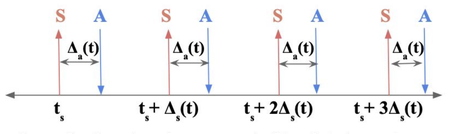


Delays in PCA



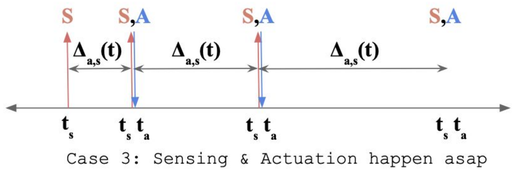
Case 1: Sensing & Actuation happen at fixed intervals

Fixed $\Delta_s(t) > d_1 + d_2 + d_3$



Case 2: Sensing happens at fixed intervals; Actuation happens asap

Fixed $\Delta_s(t) > d_1 + d_2 + d_3$; Variable $\Delta_a(t) = d_1 + d_2 + d_3$



Case 3: Sensing & Actuation happen asap

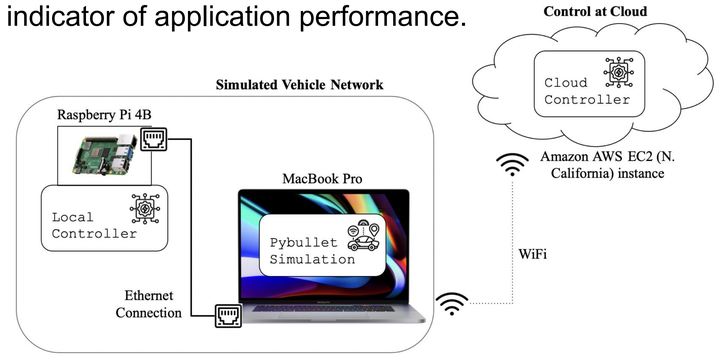
Variable $\Delta_{a,s}(t)$

$t_a = t_s = d_1 + d_2 + d_3$

(above) Types of delays in PCA loops based on system design

System Design

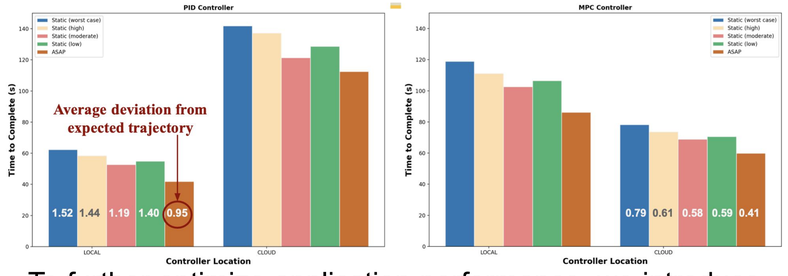
We use vehicle following as a representative example within the PyBullet simulator. We investigate PID & MPC controllers with two locations of control computation placement: local (on vehicle) and cloud (AWS EC2). We measure time to reach goal state from start state as an indicator of application performance.



(above) System Setup (below) breakdown of end-to-end latency as a function of controller type and placement

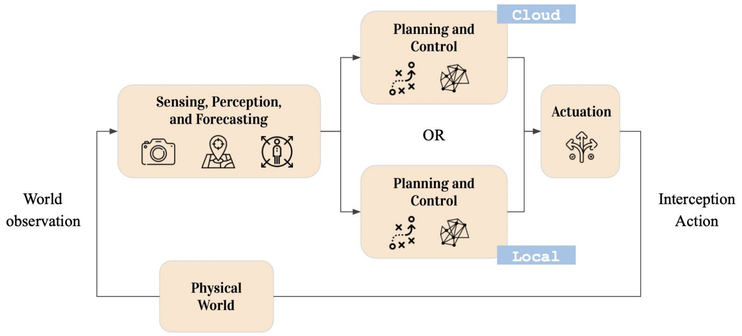
	Processor	Communication Latency (ms)	Computational Latency (ms)	End to End Latency (ms)
Local PID	Raspberry Pi 4B	(9.54, 44.9)	(56.8, 96.89)	(69.34, 111.24)
Local MPC	Raspberry Pi 4B	(7.92, 14.8)	(580.50, 773.21)	(580.62, 773.71)
Cloud PID	AWS EC2	(81.73, 263.9)	(0.25, 0.34)	(81.98, 264.18)
Cloud MPC	AWS EC2	(50.45, 117.53)	(157.66, 185.03)	(208.12, 269.85)

(below) Overall tracking latency under different kinds of delays. PID at cloud has extremely high latency while MPC at local is unfeasible due to large computation delays.

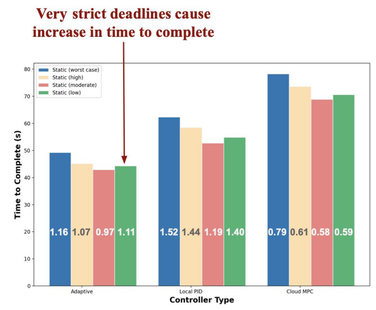


To further optimize application performance, we introduce adaptive control computation placement. We leverage the best of both controllers by running PID at local (on vehicle) and MPC at cloud as well as switching between the two based on network latency metrics

Adaptive Control Placement



(above) System architecture for adaptive control placement



- When network latency is high, we switch to local (less accurate) controller
- Static (worst-case) deadline scenario has the highest overall latency within adaptive control computation placement
- As the deadline decreases (worst-case → high → moderate), the average trajectory deviation and the time to complete also decrease
- In case of stricter deadlines (case: static (low)), the error is high because of increased invocations of the local PID controller

Application performance i.e. time to complete latency for all four cases along with average tracked trajectory deviation

Conclusions

- We explore control computation placement under different delay settings and investigate their impact on application performance.
- We show that adaptive control placement offers better overall application performance i.e. end-to-end latency in closed-loop perception-action systems

Path Forward

- Generalize the idea of dynamic deadlines for tracking scenarios for complex system with multimodal sensor inputs and extend the model to open-loop perception-action systems.
- Develop a robust deep reinforcement learning agent to predict control placement as a function of environmental context cues and network dynamics
- Extend system design to Sim2Real experiments

POINT OF CONTACT:

Pragya Sharma
pragyasharma@g.ucla.edu
Mani Srivastava
mbs@ucla.edu