The Next Twenty Years of CPS

Toward Cyber-Physical AI

Sanghoon Lee
Postdoctoral Research Fellow
DGIST, Republic of Korea
leesh2913@dgist.ac.kr



Twenty Years Ago



Me at seven, at the wheel

2006

The NSF CPS journey began.
I was seven years old.

2026

We look to the next twenty years.
I am here to listen and to share one view.

A Little About Me

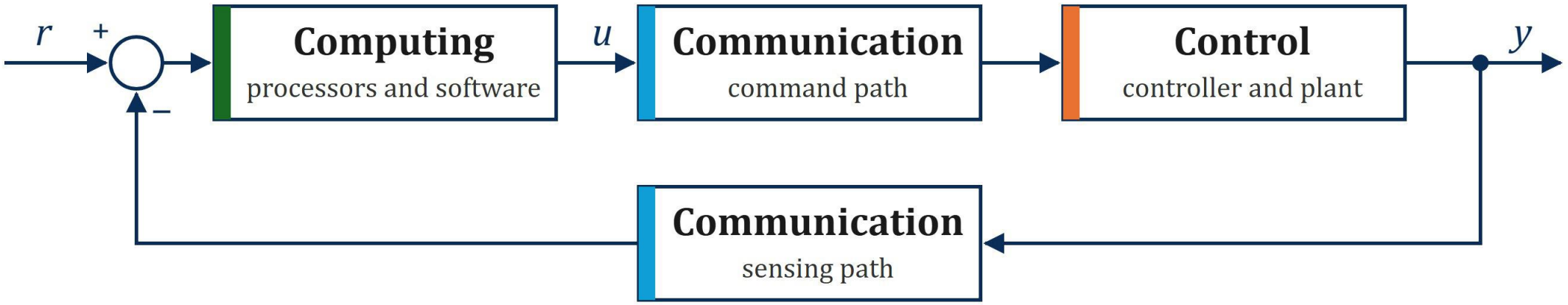
- Sanghoon Lee, Postdoctoral Research Fellow at DGIST, Korea
- I work one layer below the robot
  - Robot middleware for Physical AI
  - Where control, communication, and computing meet
- Goal: dependable foundations for AI that acts in the physical world

Wants for 20 Years from Now

- **AI will act** in the physical world, almost everywhere
  - Learned components will steer vehicles, move robots, schedule processors, and manage networks
- **We will still want** what CPS has always promised
  - Behavior we can bound, explain, and trust, even when part of the system learns

My wish for 2046. AI that is as dependable as the CPS this community built.

What CPS Gave Us



Bounded determinism

Physical behavior stays within bounds we can compute in advance, despite uncertainty and shared resources.

[1] E. A. Lee 2006 · [2] E. A. Lee 2008 · [3] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic 2010

Built Piece by Piece by This Community

1973 — Real-time scheduling
Deadlines that can be proven

[4] Liu and Layland 1973
[5] Vestal 2007

1998 — Runtime assurance
A simple, verified fallback

[6] Seto et al. 1998 · [7] Sha 2001

1999 — Networked control
Stability over imperfect networks

[8] Walsh, Ye, and Bushnell 1999

2008 — Time in models
Timing as part of what a program means

[2] Lee 2008
[9] Lee and Seshia 2017

2012 — Event-triggered control
Act only when it matters

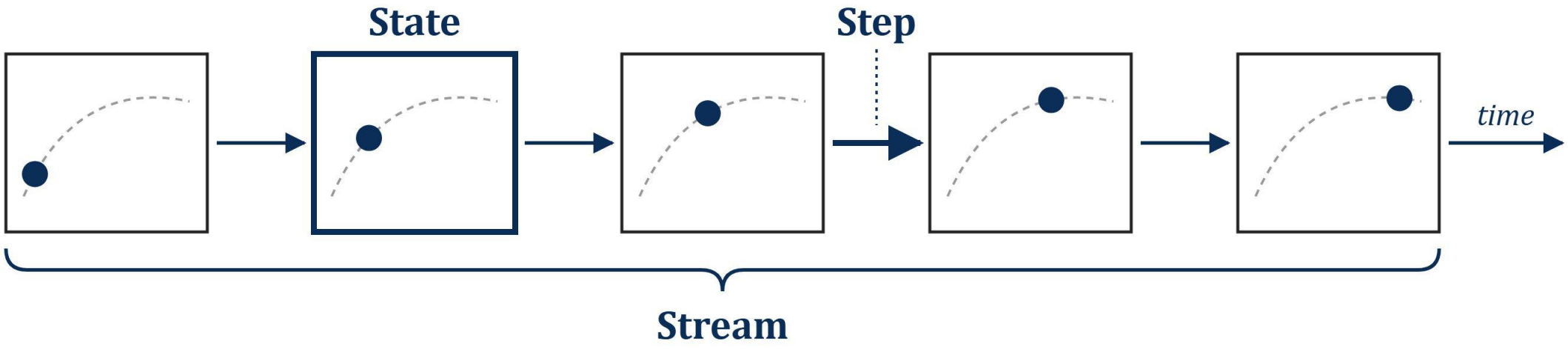
[10] Heemels, Johansson, and Tabuada 2012

2018 — Contracts
Assumptions and guarantees at interfaces

[11] Benveniste et al. 2018

Each made one part predictable. Together they made the loop trustworthy.

State, Step, and Stream



State	**Step**	**Stream**
the system at one instant	how it moves to the next	where it goes over time

Builds on transition systems and hybrid automata · [12] Henzinger 1996 · [9] Lee and Seshia 2017

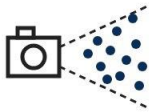
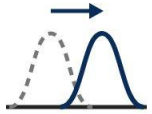
Control, computing, and communication all share these three levels.

One Frame for Three Traditions

	State the system at one instant	**Step** how it moves to the next	**Stream** where it goes over time
Control	Position, velocity, attitude	Physics and the **controller**	Physical trajectory
Computing	Memory, registers, queues	Execution and the **scheduler**	Execution trace
Communication	Queues, channel, protocol state	Channel and the **protocol**	Message history

Bounded determinism must hold at every level and across every interface.

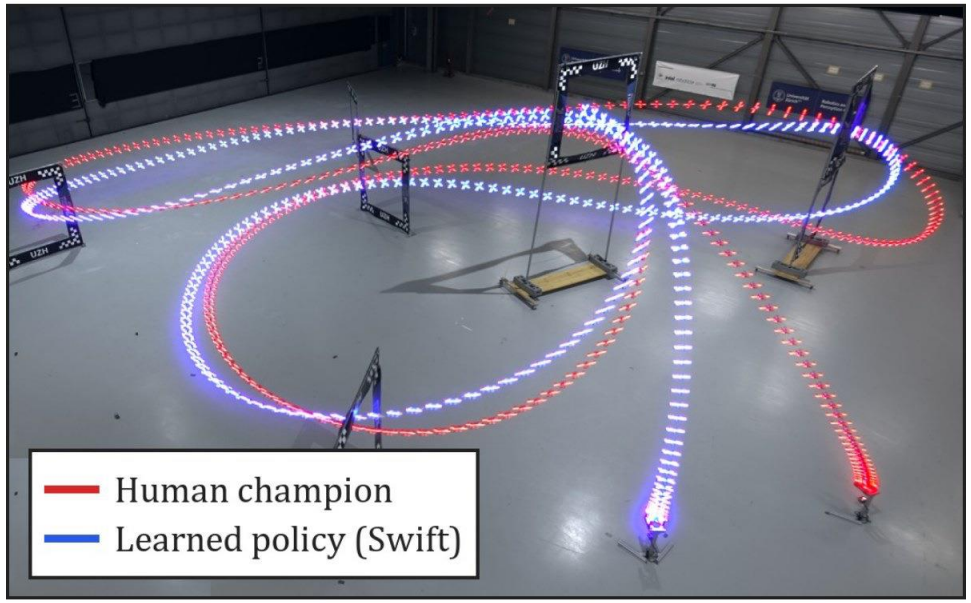
The World Around CPS Has Changed

State one instant		Cameras and LiDAR create states we cannot list at design time So the state itself must be learned from data
Step the next move		Floors, workloads, and network traffic keep shifting So the rule for the next move must change while the system runs
Stream over time		Over months in the open world, new kinds of faults appear So guarantees fixed at design time cannot cover them all

[13] Firoozi et al. 2023 · [14] Hendrycks et al. 2021

Learning-based AI grew up with exactly the capabilities these gaps call for.

AI Has Entered the Loop



[16] Photo R. Sablotny, CC BY 4.0, adapted

Control [15] Levine et al. 2016 · [16] Kaufmann et al. 2023
Learned policies fly drones and steer robots

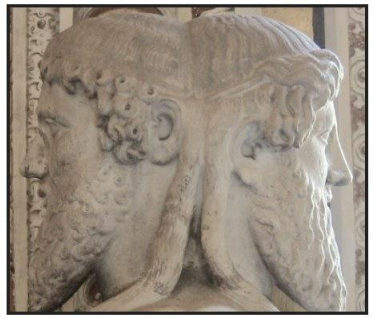
Computing [17] Mao, Alizadeh, Menache, and Kandula 2016
Learned schedulers allocate shared resources

Communication [18] Mao, Netravali, and Alizadeh 2017
Learned policies choose how video is streamed

Physical AI, as we define it
AI whose decisions shape physical behavior, within limits it cannot redefine.

Two Faces of the Same Strength

What it brings	A learned component	What it puts at risk
Sees rich data such as camera images	State **Learns its own representation**	Its safe region cannot be specified in advance
Can adapt while the system runs	Step **Decides by learned statistics**	Its worst case cannot be analyzed
Notices drift before failure	Stream **Depends on familiar data**	Its guarantees hold only while data stays familiar

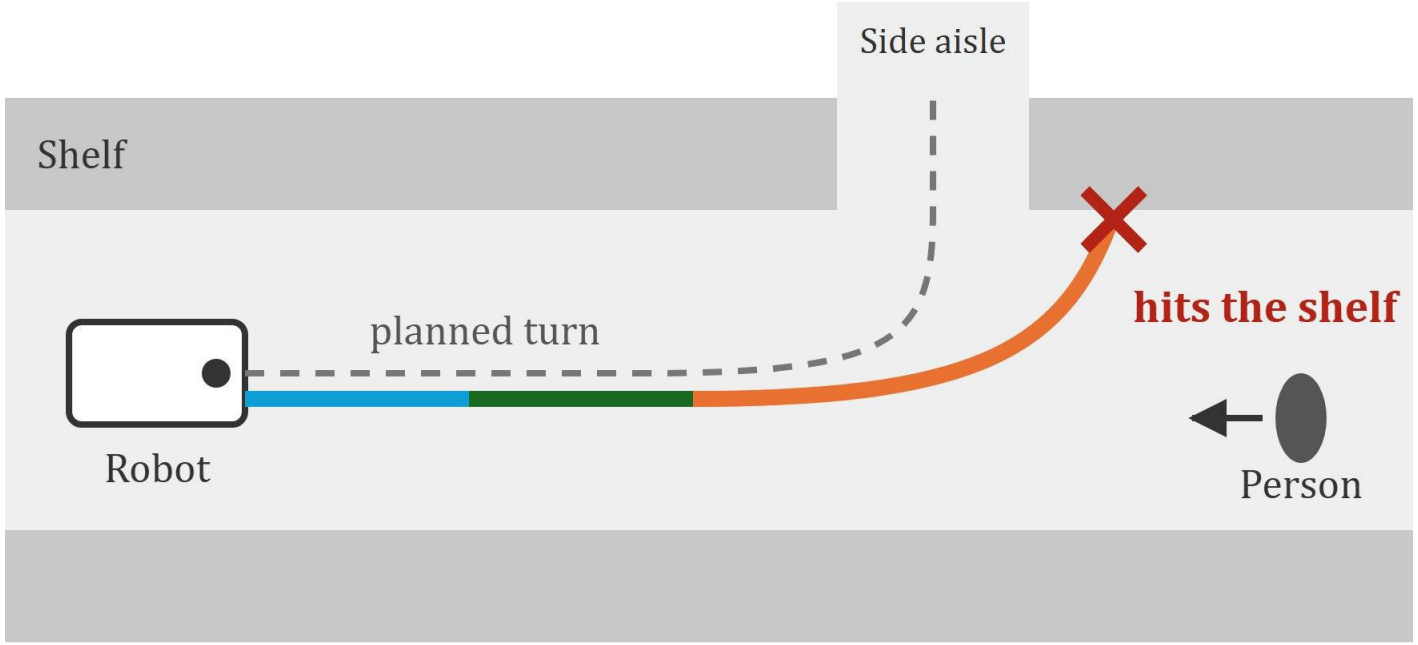


The same property expands capability and erodes guarantees.

Janus, Vatican Museums. Photo from Wikimedia Commons, public domain

Small Delays That AI Makes Common

- Multimodal sensor data overloads the network
- GPU and memory contention cause inference jitter
- The policy's output has no worst-case bound



Each part meets its own specification. Together they produce a collision.

Cyber-Physical AI

The promise of CPS, carried into systems that learn.

Bounded determinism for the Physical AI era.

A paradigm that unifies the learned component
and the cyber-physical system that receives it.

Where assurance moves

Proof inside the component → **Enforcement at its boundary**

Guarantees fixed at design time → **Assurance at run time**

Checking each part → **Reasoning about the whole loop**

Where Can AI Stand in the Loop?

Augmentation

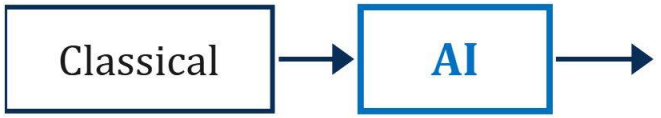


AI prepares the input.
The classical part decides.

Example. Learned state estimation for a classical controller

Without AI. Classical control still runs.

Mediation



The classical part proposes.
AI refines the final value.

Example. Learned residual on model predictive control

Without AI. Classical output remains.

Automation



AI decides alone.

Example. End-to-end learned policy for drone racing

Without AI. Nothing is left to act.

The more authority AI holds, the more carefully we must guard its boundary.

Guarding the Boundary with What We Know

	What it does	What it builds on
P	**Projection** Keeps each output inside a safe set	Safety filters and control barrier functions [19] Ames et al. 2019
I	**Isolation** Keeps time and resources within budget	Real-time scheduling and mixed criticality [4] Liu and Layland 1973 · [5] Vestal 2007
T	**Transfer** Hands control to a verified baseline	The Simplex architecture [6] Seto et al. 1998 · [7] Sha 2001

PIT composes this community's tools where AI meets the system.

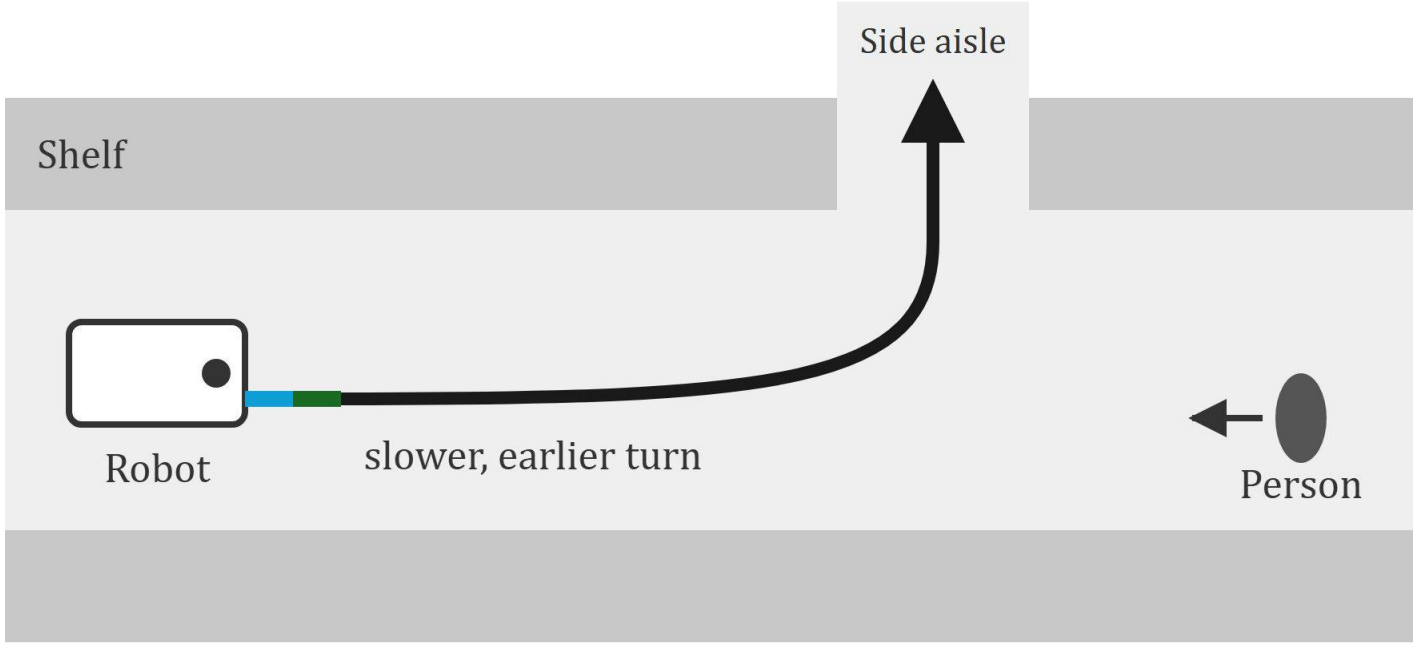
What AI Should Declare About Itself

The AI declares	For our robot	PIT uses it
Where may its outputs go? Envelope, like an operating range	Paths the drive can follow from where the robot really is	**P** **Projection** keeps each output inside
What budget does it need? Cost, like a power budget	Inference time and GPU budget for one replanning	**I** **Isolation** reserves that budget
Can we trust it right now? Validity, like operating conditions	A calibrated sign that camera frames look familiar	**T** **Transfer** hands over to a safe baseline

A datasheet for a learned component, checked while the system runs.

Same Robot, Same Disturbance, with PIT

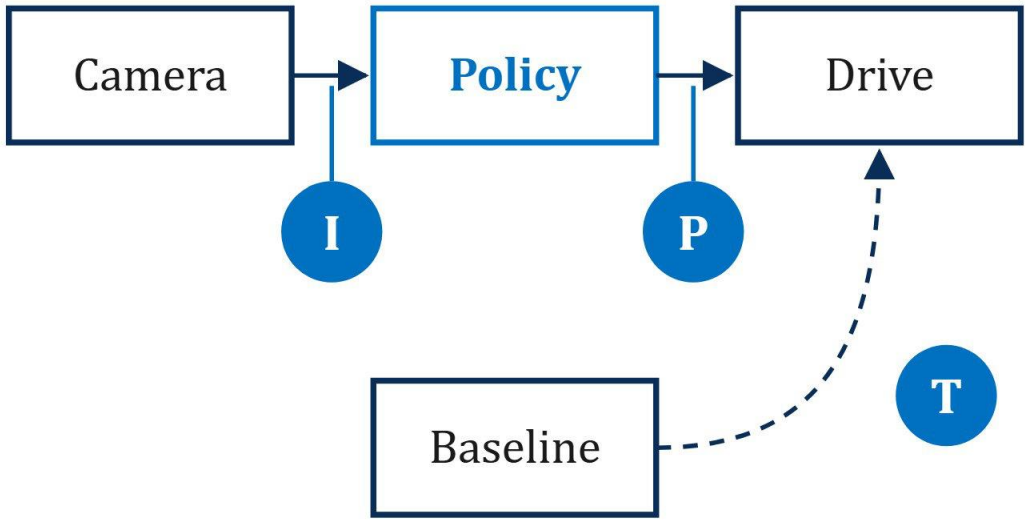
- **I** Frames and the plan arrive on time
- **P** The turn is limited to what the drive can follow
- **T** A braking baseline stays ready



The drive's controller keeps its stability guarantee, because its design assumptions are kept.

Building It Together on ROS 2

- ROS 2 is the de facto standard for building robots
  - From research labs to factories and a NASA lunar rover
    [20] Open Robotics 2022 · [21] Macenski et al. 2022
- CPAI can be built there today
  - Robot middleware is where PIT runs
  - A first prototype already runs all three



First prototype on a ROS 2 navigation stack
[22] Lee, Chae, and Park, Middleware 2026

If your group runs robots on ROS 2, I would love to build this together.

Questions for the Next Twenty Years

Optimizing across the three Cs

Pairs have been co-designed for decades.
Can we tune all three together
when each part learns?

1

AI that knows its own system

Delay-aware learning has begun.
Can a model adapt to its deadlines, compute,
and network, with guarantees?

2

Evidence of safety

Is the absence of trouble evidence of safety?
Can counting successful grasps measure safety?

3

The last boundary

If AI runs all three end to end,
where does its authority end?

4

Twenty Years from Now



2006

CPS asked how computation meets physics.



2026

We ask how learning meets physics, while keeping what we built.

Thank you. I came here to learn from you.

Acknowledgements

- Kyung-Joon Park, DGIST
- Karl H. Johansson, KTH Royal Institute of Technology
- Xue Liu, McGill University and MBZUAI
- Guoliang Xing, The Chinese University of Hong Kong

References

[1] E. A. Lee, “Cyber-physical systems: Are computing foundations adequate?” Position paper, NSF Workshop on Cyber-Physical Systems, Austin, TX, Oct. 2006.

[2] E. A. Lee, “Cyber physical systems: Design challenges,” in Proc. IEEE ISORC, 2008.

[3] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, “Cyber-physical systems: The next computing revolution,” in Proc. ACM/IEEE DAC, 2010.

[4] C. L. Liu and J. W. Layland, “Scheduling algorithms for multiprogramming in a hard-real-time environment,” J. ACM, vol. 20, no. 1, 1973.

[5] S. Vestal, “Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance,” in Proc. IEEE RTSS, 2007.

[6] D. Seto, B. Krogh, L. Sha, and A. Chutinan, “The Simplex architecture for safe online control system upgrades,” in Proc. ACC, 1998.

[7] L. Sha, “Using simplicity to control complexity,” IEEE Software, vol. 18, no. 4, 2001.

[8] G. C. Walsh, H. Ye, and L. Bushnell, “Stability analysis of networked control systems,” in Proc. ACC, 1999.

[9] E. A. Lee and S. A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd ed., MIT Press, 2017.

[10] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in Proc. IEEE CDC, 2012.

[11] A. Benveniste et al., “Contracts for system design,” Found. Trends Electron. Des. Autom., vol. 12, no. 2–3, 2018.

References

[12] T. A. Henzinger, “The theory of hybrid automata,” in Proc. IEEE LICS, 1996.

[13] R. Firoozi et al., “Foundation models in robotics: Applications, challenges, and the future,” arXiv:2312.07843, 2023.

[14] D. Hendrycks, N. Carlini, J. Schulman, and J. Steinhardt, “Unsolved problems in ML safety,” arXiv:2109.13916, 2021.

[15] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” JMLR, vol. 17, no. 39, 2016.

[16] E. Kaufmann et al., “Champion-level drone racing using deep reinforcement learning,” Nature, vol. 620, 2023.

[17] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, “Resource management with deep reinforcement learning,” in Proc. ACM HotNets, 2016.

[18] H. Mao, R. Netravali, and M. Alizadeh, “Neural adaptive video streaming with Pensieve,” in Proc. ACM SIGCOMM, 2017.

[19] A. D. Ames et al., “Control barrier functions: Theory and applications,” in Proc. ECC, 2019.

[20] Open Robotics, “ROS 2 in Science Robotics,” Open Robotics blog, May 2022.

[21] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot Operating System 2: Design, architecture, and uses in the wild,” Science Robotics, vol. 7, no. 66, 2022.

[22] S. Lee, J. Chae, and K.-J. Park, “Harness engineering for physical AI: Robot middleware is the harness layer,” in Proc. ACM/IFIP Middleware, 2026.