A SIMULATION-BASED ARCHITECTURE FOR SMART CYBER-PHYSICAL SYSTEMS

THOMAS GABOR, LENZ BELZNER, MARIE KIERMEIER, MICHAEL TILL BECK, ALEXANDER NEITZ

LMU MUNICH
INSTITUTE FOR INFORMATICS
E.g. autonomous factories, smart cities, ...

**Partially uncontrolled** environment

**Incomplete** design-time knowledge

**Complex decisions** at runtime

**Space of behavioral solutions** available

**Task:**

Choose adequate solution in time
CHALLENGES

• **Challenge 1:** Changing environment
  • Non-stationary optimal behavior

• **Challenge 2:** Aleatoric uncertainty
  • Optimal behavior is probabilistic
  • Irreducible

• **Challenge 3:** Epistemic uncertainty
  • “Small data”
  • Reducible

• **Challenge 4:** Behavioral uncertainty
  • Due to resource constraints
A SIMULATION-BASED ARCHITECTURE FOR SMART CYBER-PHYSICAL SYSTEMS
MOTIVATION

Planning

● Given an MDP
● State space $S$, action space $A$
● Model of domain dynamics $P(S|S \times A)$
● Reward function $R : S \times A \times S \rightarrow \mathbb{R}$ (encoding system goals)
● Find strategy $P(A|S)$ that maximizes expectation of gathered reward

Challenge

● Partially uncontrolled domains
● Environment changes without system taking action
● Necessitates efficient decision making
**ONLINE PLANNING**

**Simulation-Based Online Planning**
- Interleaving of planning and execution
- Concentrates planning effort on current situation
- Fast planning by sampling the search space

**State of the Art (excerpt)**
- Discrete Bandit-Based Planners: MCTS, UCT, SHOT
- Rolling Horizon Evolutionary Algorithms
- Continuous Bandit-Based Planners: HOOT, HOLOP
- Cross Entropy Open Loop Planning

---

Cazenave, Tristan. "Sequential halving applied to trees." Computational Intelligence and AI in Games. 2015.
THE DIGITAL TWIN ARCHITECTURE

[Diagram showing the architecture with components like Controller, Aggregator, Intuitive Reactor, Planner, and Digital Twin, with arrows indicating interactions such as `+publishTo(Event sensorData)`, `+sendAction`, `+publishTo(Event observations)`, `+init/+destroy`, etc.]
THE DIGITAL TWIN ARCHITECTURE
CONTROL TIERS

- Four tiers of control

- **Physical necessity**
  - Laws of nature, domain inherent
  - In particular important for high-fidelity simulations

- **Machine-environment interface**
  - Circuit control, sensors, actuators, computational hardware
  - Is a system design variable (in contrast to physical necessity)
  - Fixed in hardware, not easily changed at runtime

- **Immediate reaction**
  - Watchdogs, fixed behavioral rule sets, expert systems
  - Software, changeable at runtime

- **Planned reaction**
  - Utility functions, online planning, use of the digital twin
  - Software, changeable at runtime
CHALLENGES, APPROACHES & CASE STUDIES
CHALLENGES & CASE STUDIES

Challenges for digital twin control

- Scalability to complex models $\rightarrow$ adaptive abstraction
- Safe operation $\rightarrow$ risk awareness

Case Studies

- Smart factory agent control
- Smart grid energy commitment
TIME ADAPTIVE MOTION PLANNING

- Consider a smart factory
- Materials and machines move probabilistically
- Model based motion planning

**Problem:**
- Environment changes continuously
- Optimality of behavior changes continuously
- Simulation is expensive

**Goal:** Collect targets

**Constraint:** Avoid static and dynamic obstacles

**How to actuate in order to collect targets and avoid obstacles?**

- **We answer this question using a simulation @ runtime**
  - Evaluate potential actuations with the digital twin
  - Use of *time adaptation* for situative abstraction
TIME ADAPTIVE MOTION PLANNING

- Fixed temporal resolution is not always optimal
  - Typical hyperparameter problem: Where to fix?

- Assumption: A simulation is available as a function of time $P(S|S \times A \times T)$
- Idea: Incrementally identify 'important' moments
  - Optimize action duration
  - Yields adaptation of simulation effort to current situation
- A similar argument can be made for simulation depth

RESULTS

With/without temporal abstraction for empirical worst/best case parameters

→ Temporal abstraction reduces time needed to collect the items
RESULTS

Temporal abstraction reduces simulation effort

→ Enables effective use of simulation
Consider a smart grid
Highly volatile/variable energy consumption
Continuous reconfiguration of energy production

Problem:
- Overproduction induces cost
- Power supply has to be guaranteed
- Only a limited quantity can be sold to other markets

Goal: Minimize production – consumption
Constraint: \(0 < \text{production} - \text{consumption} < \text{maximum oversupply}\)

How much power to produce on next commitment?

We answer this question using a simulation @ runtime
- Evaluate potential production allotments with the digital twin
- We compared risk-neutral and risk-aware strategies
- Both use simulations, but differ in their evaluation
**Safe adaptive abstraction**
- Adaptive abstraction may provide scalability
- Potentially “abstracting away” risks?
- How to provide (statistical) guarantees?

**Learning/adapting models @runtime**
- Machine learning enables model adaptation @runtime
- However, this introduces model uncertainty
- How to treat safety issues under model uncertainty?
- E.g. behavior optimization vs. model confidence
- Possible approach → QoS-awareness
  - Optimize “up to” QoS requirement
  - Maximize confidence
- Consequences of adaptive abstraction and model uncertainty?
SUMMARY & NEXT

Summary
- Smart simulation-based CPS
- Digital twin architecture
- Control tiers
- Application examples
- Challenges for simulation-based control

Up next
- QoS-awareness, V&V @runtime
- Learning models @runtime
- Combining abstraction & safety
- Distributed simulation-based CPS