Autonomous Software Systems Design

feat. Control Theory

Alessandro V. Papadopoulos
alessandro.papadopoulos@mdu.se
http://www.idt.mdh.se/~aps01/

Sep 1st, 2022
CASTOR Software days @ KTH
Interconnected World
Made of Computing Systems
Mobile Momentum Metrics

By 2023

- More Mobile Users: 5.7 Billion
- More Mobile Connections: 13.1 Billion
- Faster Mobile Speeds: 43.9 Mbps

Source: Ericsson
Whoops, something went wrong...

Netflix Streaming Error
We're having trouble playing this title right now. Please try again later or select a different title.
How Do We Tame Such Complexity?
Autonomous Software Systems Design
feat. Control Theory

What is control?
“The term *control* has many meanings and often varies between communities. In this book, we define control to be the use of algorithms and feedback in engineered systems.”

The Hidden Technology

😊 Widely used
😊 Very successful
+-+- Seldom talked about
😊 Except when disaster strikes
😊 Why?

Easier to talk about devices than ideas
Not enough attention to popularization
Control System: Conceptual Model

Very similar to the agent model in AI
Control System: Block Diagram

- Targets & Objectives
- Intelligence
- Actuators
- Plant
- Sensors
- Environment
Classical Application Areas

- Robotics
- Automotive
- Industrial and Production
- Medical and Biomedical
- Avionics and Aerospace
- Power plants
- Water management
- AND MANY MORE...
What is the common ground of these applications?

Physics provides **mathematical models** for describing the behaviour of physical systems:

\[
\dot{x}(t) = f(x(t), u(t), t) \\
y(t) = g(x(t), u(t), t)
\]
What Is the **Physics** of **Computing Systems**?
Research Context

Pros

- Control-based approaches are very powerful
  - Autonomous-by-design
  - Easy to implement
    - Lightweight
    - Energy-efficient
  - Mathematically grounded-approaches
  - Guarantees

Cons

- They require control experts to design
  - Can we automate this?
- They often require a model
Control Theory Can Be Employed in Many Different Ways
Control Theory Can Be Employed in Many Different Ways
Control Theory Can Be Employed in Many Different Ways

Explicit modeling

yes

many alternatives
Control Theory Can Be Employed in Many Different Ways

Explicit modeling
Control Theory Can Be Employed in Many Different Ways

Explicit modeling

- no
- yes
Control Theory Can Be Employed in Many Different Ways

Explicit modeling

- no
- yes

single input, single output

ICSE14
Control Theory Can Be Employed in Many Different Ways

**FSE15**
*Filieri, Hoffmann, Maggio*
Automated Multi-Objective Control for Self-Adaptive Software Design

**FSE16**
*Shevtsov, Weyns*
Keep It SIMPLEX: satisfying multiple goals with guarantees in control-based self-adaptive systems

**FSE17**
*Maggio, Papadopoulos, Filieri, Hoffmann*
Automated Control of Multiple Software Goals using Multiple Actuators

![Diagram](https://example.com/diagram.png)
Control Theory Can Be Employed in Many Different Ways

Explicit modeling

- no
- yes

Model based control
- single input, single output
- multiple input, multiple output

ICSE14
FSE17, 16, 15
Control Theory Can Be Employed in Many Different Ways

- Single input, single output
- Multiple input, multiple output

FSE17, 16, 15

ICSE14
Control Theory Can Be Employed in Many Different Ways

- Model free control
  - no
  - not explicitly
  - yes

- Model based control
  - single input, single output
  - multiple input, multiple output

ICSE14
FSE17, 16, 15
Task Scheduling
When a Model Does Exist

Papadopoulos et al., “Hard real-time guarantees in feedback-based resource reservations”, Real-Time Systems
Scheduling Problem

- **Ideal**
- **Actual**
- **Tentative Assignment**
- **Discrepancy**
- **Actual Execution**

(time)
Control-Based Scheduling

<table>
<thead>
<tr>
<th>Target Execution time</th>
<th>Scheduler</th>
<th>Time budget Assignment</th>
<th>Measure time</th>
<th>Discrepancies</th>
</tr>
</thead>
</table>
Model of the Task

\[ x(k) = u(k - 1) + d(k) \]
Controlling the Task Budget of Time

The allocated time budget can be controlled automatically based on an “Integral controller”

\[ u(k) = u(k - 1) + K \times (\text{targetBudget}(k) - \text{measuredBudget}(k)) \]

- \( K \) is the only parameter that needs to be tuned, and it can be chosen in the range (0,1] (stability reasons that we do not discuss).

- Extremely simple model
- Hard Real-Time guarantees can be provided
- Easy to implement the scheduling strategy
Cloud Computing Applications

When the Model Is Hard To Find

Lakew et al., “KPI-agnostic Control for Fine-Grained Vertical Elasticity”, CCGrid 2017
Cloud computing promises an infinite capacity but...

There is no cloud
it’s just someone else’s computer
Cloud control

- Several different problems
  - Load balancing
  - Autoscaling
  - Fault tolerance
  - Performance
  - Real-time guarantees
  - Resiliency
Workload Characterisation

Easy and predictable

Completely unpredictable

Flash crowd
• 82% of end-users give up on a lost payment transaction

• 25% of end-users leave if load time > 4s

• 1% reduced sale per 100ms load time

• 20% reduced income if 0.5s longer load time
Regular operation
Regular operation

Brownout
Brownout in Cloud Systems

Klein et al., ICSE 2014
Brownout in Cloud Systems

- Higher Resiliency
- Better User Experience
- Increased Revenue
Multiple Replicas Load Balancing

- Useful for
  - Scale beyond a physical machine
  - Resilience
    - Hide auto-scaling mishaps
    - Hide infrastructure failures

GOAL: Maximize Optional Content
Multiple Failures Scenario with SQF

- Replica 4 fails
- Replica 3 fails
- Replica 1 fails
- Replica 2 restored
- Replica 3 restored
- Replica 4 restored

Only replica 0 during this interval
Can One Do Better?
Control-Based Approach

- **Idea:** Modify the SQF policy to maximize the optional content served $\Theta$, and minimize the queue-length $q$

- **We measure:** queue-lengths $q_i$, and dimmers $\Theta_i$

- **We control:** queue-offsets $u_i$
  
  - SQF picks the replica with smallest value of $q_i - u_i$
  
  - Queue-offsets computed with a “PI-Based Heuristic” (PIBH) policy:
    
    $$ u_i(k + 1) = (1 - \gamma)[u_i(k) + \gamma_p \Delta \Theta_i(k) + \gamma_l \Theta_i(k)] + \gamma q_i(k) $$

  Maximize the optional content
  Minimize the queue length
Optional Content vs Response Time

Pushing for more optional content

95th percentile of the response time [ms]
Resiliency

Only replica 0 during this interval

Optional Content Ratio [%]

Square Footprint (SQF)

Timeouts [req/s]

PIBH

Timeouts [req/s]

Optional Content Ratio [%]

Replica 4 fails
Replica 2 fails
Replica 1 restored
Replica 3 restored
Replica 4 restored
Replica 1 fails
Replica 2 fails
Replica 3 falls
Replica 4 restored

Time [s]
Control Has Proven Useful in the Design of Autonomous Software Components
Autonomous Software Design With Control

- Feedback Scheduling
- Wireless Sensor Networks
- Hard-RT Guarantees
- Mixed-critical systems

- Application heartbeat
- Control of application
- Memory management

- Real-Time and Embedded Systems
- Operating Systems
- Performance Engineering
- Cloud and Fog Computing

- Self-Adaptive Software (SAS)
- Requirement-based control design
- Automated control synthesis for SAS
- Proactive Control

- Autoscaling
- Load balancing
- Resiliency
- Fault-tolerance
- Data streaming

- Control-based benchmarking
- Probabilistic Performance evaluation
- Methodological principles

© Alessandro V. Papadopoulos
Current Challenges

- **Automate** the decision-making algorithm design
- Include and model **Humans-In-The-Loop**
- Cross-fertilisation with other fields
  - Artificial Intelligence
  - Machine learning
  - Formal Methods
  - Real-Time Systems
- **General formulation of **Physics Theory of Computing Systems**
Comments, Feedback, and Questions Are Welcome

This work has been supported by:

[Logos of European Commission, Vetenskapsrådet, and KK-stiftelsen]