Tutorial: Algorithms & Hardware for Embedded Optimization

14:00 What Is Different about Embedded Optimization? 
  Eric Kerrigan
14:40 Survey of Industrial Applications of Embedded MPC 
  Alexander Domahidi
15:00 Efficient QP Frameworks for Industrial Embedded MPC 
  Giorgio Kufoalor
15:20 Implicit vs Explicit MPC 
  Martin Klauco (on behalf of Michal Kvasnica)
15:40 Robustness of Explicit MPC 
  Pedro Ayerbe
What can I expect from this session?
What is Different About Embedded Optimization?

Eric Kerrigan, Bulat Khusainov, George Constantinides
On time & on budget in an uncertain world
Applications of Embedded Optimization
A Fundamental Problem

\[ \frac{dt}{dt} = 1 \]

Correctness should be a function of time
Process + Embedded Optimizer = Uncertain Cyber-Physical System

\[ u^*(y) := \arg \min_u f(u, y) \]
\[ \text{s.t. } g(u, y) = 0 \]
\[ h(u, y) \leq 0 \]

\[ (p^*, c^*) := \arg \min_{p, c} \phi(p, c) \]
\[ \text{s.t. } \alpha(p, c) = 0 \]
\[ \beta(p, c) \leq 0 \]

Time + Uncertainty \Rightarrow Approximate today might be better than accurate tomorrow
Discretized Optimal Control/Estimation Problems

\[
\min_{q \in Q, s \in S} \sum_{i=0}^{N-1} \ell(q_i, s_i, s_{i+1}, i, d, t)
\]

\[
f(q_i, s_i, s_{i+1}, i, d, t) = 0, \ i = 0, \ldots, N - 1
\]

\[
g(q_i, s_i, s_{i+1}, i, d, t) \leq 0, \ i = 0, \ldots, N - 1
\]

Ordering is important:

\[
x := (s_0, q_0, s_1, q_1, \ldots, s_{N-1}, q_{N-1}, s_N)
\]

Sparse & structured matrices
⇒ small (and full) not always better
Exploit Structure: Interior Point Method

**Time**

![Graph showing computation time vs. time horizon](image)

The graph illustrates the computation time as a function of the time horizon for both the proposed method and the Rao–Wright–Rawlings approach.

**Space**

![Graph showing computation time vs. number of subsystems](image)

Similarly, the graph shows the computation time as a function of the number of subsystems for the same methods.

Proposed = [Cantoni, Farokhi, Kerrigan, Shames, AuCC16]
Control / Automation

Unknown Inputs

Known Inputs

SYSTEM

Estimated of Known Outputs

AlGORITHM

Unknown Outputs

Corrections

Known Outputs

Estimates of Unknown Outputs

Controller
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

![Diagram]

- Output
- Input
Real-time Optimal/Predictive Control (e.g. Receding Horizon)
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
4. Go to step 1
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
4. Go to step 1
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
4. Go to step 1
Real-time Optimal/Predictive Control (e.g. Receding Horizon)

1. Take measurement
2. Solve optimal control problem
3. Implement first part
4. Go to step 1

Feedback Algorithm
Signal Processing / Estimation / Learning

System Model Algorithm

Unknown Inputs → Known Inputs

Corrections

Unknown Outputs → Known Outputs

Estimates of Unknown Outputs

Estimates of Known Outputs

Estimator
Process + Embedded Optimizer = Uncertain Cyber-Physical System

\[ u(t) \in \text{Proj} \arg \min_{x} \{ J(x, d, t) \mid x \in X(d, t), \ d = D(y, v, t) \} \]

Optimal closed-loop system?
When is Sub-optimal Optimal?
Real-time Dynamic Optimization

- **Objective function value**
  - **Non-monotonic & sub-optimal**
  - **Monotonic & sub-optimal**
  - **Optimal**

- **Latency/computational delay**
When is Sub-optimal Optimal?
Real-time Dynamic Optimization

$$(u^*(\cdot, \delta), y^*(\cdot, \delta)) := \arg \min_{(u, y)} V(u), \quad V(u) := \int_{0}^{1} u(t)^2 dt$$

s.t. \quad y(0) = 0, \quad y(1) \geq 1

\quad u(t) = 0, \quad \forall t \in [0, \delta)

\quad \dot{y}(t) = u(t), \quad \forall t \in [0, 1)

$$u^*(t, \delta) = \begin{cases} 0 & \forall t \in [0, \delta) \\ 1/(1 - \delta) & \forall t \in [\delta, 1) \end{cases}$$

$$V(u^*(\cdot, 0.2)) = 1.25 < V(1.1u^*(\cdot, 0.2)) < V(u^*(\cdot, 0.4)) \approx 1.7$$
Precision, Accuracy and Latency: Fast Gradient Method

 Bounds for $i \to \infty$ [Jerez et al., ECC 2013, IEEE TAC 2014]
Quantization Errors in Communication: Distributed First-Order Optimization

[Pu, Zeilinger, Jones, arXiv 2015]
Hardware or Algorithm?

Well I guess that settles that old argument.
Computing Systems and Resources

- Processing
- Communication
- Storage
- Time
- Space
- Energy

Diagram showing relationships between computing resources and systems.
## Possible Design Parameters

<table>
<thead>
<tr>
<th>Computer Hardware</th>
<th>Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost, space, energy, power</td>
<td>accuracy, termination tolerances</td>
</tr>
<tr>
<td># processors/cores/arithmetic units</td>
<td># iterations in each loop</td>
</tr>
<tr>
<td>pipeline depth</td>
<td>step length parameters</td>
</tr>
<tr>
<td>clock frequency and supply voltage</td>
<td>amount of data/results to store</td>
</tr>
<tr>
<td>memory architecture, latency, size</td>
<td>time horizon</td>
</tr>
<tr>
<td>communication architecture, bandwidth</td>
<td>complexity of physical model</td>
</tr>
<tr>
<td>number representation, word length</td>
<td>coarseness of discretization</td>
</tr>
<tr>
<td>actuation and sampling schedule/rate</td>
<td>scheduling/communication strategy</td>
</tr>
</tbody>
</table>
Co-Design as Multi-objective Optimization

\[
\min_{c} \{ F(H(c), c) \mid (H(c), c) \in G \}
\]

[Khusainov, Kerrigan, Constantinides, ECC2016]
Explicit Constrained LQR in Fixed Point

\[
\min_c \{ F(H(c), c) \mid (H(c), c) \in G \}
\]

\[ c := \#\text{bits} \]

[Suardi et al., ECC 2013]
Size is Very Important in Microprocessor Design

Die area = 1
Working = 64

Die area = 4
Working = 4

Cost per die = f(area^x), x\in[2,4]
FPGA Resources for an Optimal Controller

Interior-point [Rao et al., 1998] on Xilinx Virtex 6

[Longo, Kerrigan, Constantinides, Automatica 2014]
Optimal Control in Low Precision Floating-Point

[Longo, Kerrigan, Constantinides, Automatica 2014]
## Computational Resources for an Adder

### Xilinx Virtex-7 XT 1140 FPGA:

<table>
<thead>
<tr>
<th>Number representation</th>
<th>Registers/Flip-Flops (FFs)</th>
<th>Lookup-Tables (LUTs)</th>
<th>Latency/delay (clock cycles)</th>
</tr>
</thead>
<tbody>
<tr>
<td>double floating-point</td>
<td>1035</td>
<td>852</td>
<td>12</td>
</tr>
<tr>
<td>52-bit mantissa</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single floating-point</td>
<td>542</td>
<td>445</td>
<td>12</td>
</tr>
<tr>
<td>23-bit mantissa</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixed-point</td>
<td>53</td>
<td>53</td>
<td>1</td>
</tr>
<tr>
<td>53 bits</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixed-point</td>
<td>24</td>
<td>24</td>
<td>1</td>
</tr>
<tr>
<td>24 bits</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Cheap and low power processors often only have fixed-point.*
Floating-Point Arithmetic

1100000010011110001

0101010000100111111

001001001100110000

round-off error

round-off error

round-off error

time
Fixed-Point Arithmetic
Computational Resources for an Adder

Given a fixed amount of silicon (£/$/€):

200x more fixed point additions

- per second
- per Joule

than in floating point
Fast Gradient Method in Fixed Point Arithmetic

Atomic force microscope
Actuation rate > 1 MHz

\[
\begin{align*}
\min_{u} & \quad u' H u + u' D(y, v, t) \\
\text{s.t.} & \quad \underline{u} \leq u \leq \bar{u}
\end{align*}
\]

QP solver on FPGA:
latency < 1 \(\mu\)s
power < 1 W
within 0.1% of optimal

[Jerez et al., ECC 2013 & IEEE TAC 2014]
Latency, Precision and Silicon: MINRES

Xilinx Virtex-7 XT 1140 FPGA

[Jerez, Constantinides, Kerrigan, IEEE TC 2015]
Fixed point (FPGA) vs floating point (GPU): MINRES

- NVIDIA C2050
  - 1.03 TFLOP/s
  - 1.15GHz, 100W
  - >180 GOP/s/W

- Xilinx Virtex-7 XT 1140
  - 400MHz, 22W
  - >180 GOP/s/W

Error tolerance for >90% of problems ($\eta$)

- k = 17
  - P = 21

- k = 23
  - P = 11

- k = 58
  - P = 2

- k = 41
  - P = 4

[Jerez, Constantinides, Kerrigan, IEEE TC 2015]
What is Different About Embedded Optimization?

+ Optimal?

Feedback

Structure

Hardware ⇔ Algorithm