Fast Model Predictive Control (MPC)

Manfred Morari

with thanks to
Colin Jones, Paul Goulart,
Alex Domahidi, Stefan Richter
and many other collaborators
Fast Model Predictive Control (MPC)

Manfred Morari

with thanks to
Colin Jones, Paul Goulart,
Alex Domahidi, Stefan Richter
and many other collaborators
Characterization of Structural Controllability

K. GLOVER, MEMBER, IEE, AND L. M. SILVERMAN, MEMBER, IEE

Abstract—A self-contained algebraic derivation of the necessary and sufficient conditions for a multiinput system with a fixed zero structure to be structurally controllable is given. In addition, a new recursive test for determining structural controllability which utilizes only Boolean operations is obtained.

IEEE TRANSACTIONS ON AUTOMATIC CONTROL, AUGUST 1976
Lino Guzzella appointed President of ETH Zurich

The Swiss Federal Council has appointed Lino Guzzella, ETH Rector and Professor of Thermotronics, as the future ETH President. He will be taking over from Ralph Eichler who is retiring at the end of 2014. By taking this decision, the Swiss Government has endorsed the unanimous proposal made by the ETH Board.
Fast Model Predictive Control (MPC)

Manfred Morari

with thanks to
Colin Jones, Paul Goulart,
Alex Domahidi, Stefan Richter
and many other collaborators
Synthesis of Optimal Control Laws

Infinite-Horizon Optimal Control

\[
J^*(x) = \min_{u_i \in U} \sum_{i=0}^{\infty} l(x_i, u_i)
\]
\[\text{s.t. } x_{i+1} = f(x_i, u_i) \quad x_i \in X\]

Dynamic Programming

\[
J^*(x) = \min_u l(x, u) + J^*(f(x, u))
\]
\[\text{s.t. } (f(x, u), u) \in X \times U\]

• Challenge is computation!
Synthesis of Optimal Control Laws

Infinite-Horizon Optimal Control

\[ J^*(x) = \min_{u_i \in U} \sum_{i=0}^{\infty} l(x_i, u_i) \]

s.t. \( x_{i+1} = f(x_i, u_i) \)
\( x_i \in X \)

Challenge is computation!
Closed-form solution for linear systems, no constraints only: LQR,…
Synthesis of Optimal Control Laws

Infinite-Horizon Optimal Control

\[ J^*(x) = \min_{u_i \in U} \sum_{i=0}^{\infty} l(x_i, u_i) \]
\[ \text{s.t. } x_{i+1} = f(x_i, u_i) \]
\[ x_i \in X \]

Dynamic Programming

\[ J^*(x) = \min_u l(x, u) + J^*(f(x, u)) \]
\[ \text{s.t. } (f(x, u), u) \in X \times U \]

Explicit calculation of control law \( u^*(x) \) offline

Model Predictive Control

\[ J^*(x_0) = \min_{u_i} \sum_{i=0}^{N} l(x_i, u_i) + V_f(x_N) \]
\[ \text{s.t. } (x_i, u_i) \in X \times U, \ x_N \in X_f \]
\[ x_{i+1} = f(x_i, u_i) \]

Online optimization problem defines control action \( u_0^*(x) \)
Model Predictive Control: Properties

Theory is well-established
Mayne, Rawlings, Rao, Scokaert (2000), *Automatica*
“MPC: Stability & Optimality (Survey Paper).”

- **Recursive feasibility**: Input and state constraints are satisfied
- **Stability** of the closed-loop system
  - $J^*(x)$ is a convex Lyapunov function

- **Assuming** the real-time optimization problem is solved to $\varepsilon$-optimality
Embedded Model Predictive Control

Traditional MPC

• Successful in process industries
• Sampling times of minutes
• Powerful computing platforms

Embedded MPC

• Small, high performance plants
• Sampling times of ms to ns
• Limited embedded platform
Embedded Systems by the Numbers

Worldwide unit shipments of embedded computing platforms:

![Graph showing unit shipments over years]


**ARM’s 32-bit embedded systems growing at 20% p.a.**

Source: Keith Clarke, ARM Vice President, *Keynote at CDNLive*, May 2013

New opportunity for automated decision making based on optimization
# Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit MPC</td>
<td>1\textsuperscript{st} Order–Fast Gradient</td>
</tr>
<tr>
<td>Approx. Explicit MPC</td>
<td>Interior Point Opt.</td>
</tr>
</tbody>
</table>
Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit MPC</td>
<td>1st Order–Fast Gradient</td>
</tr>
<tr>
<td>Approx. Explicit MPC</td>
<td>Interior Point Opt.</td>
</tr>
</tbody>
</table>
Explicit MPC: Online => Offline Processing

- Optimization problem is parameterized by state
- Control law piecewise affine for linear systems/constraints
- Pre-compute control law as function of state $x$ (parametric optimization)

Result: Online computation dramatically reduced

$$u^*(x_0) = \arg\min_{u_i} \sum_{i=0}^{N} l(x_i, u_i) + V_f(x_N)$$

s.t. $(x_i, u_i) \in X \times U$

$$x_{i+1} = f(x_i, u_i)$$

$x_N \in X_f$

[M.M. Seron, J.A. De Doná and G.C. Goodwin, 2000]
[T.A. Johansen, I. Peterson and O. Slupphaug, 2000]
[A. Bemporad, M. Morari, V. Dua and E.N. Pistokopolous, 2000]
Explicit MPC : Fast online evaluation

• Online evaluation reduced to:
  1. Point location
  2. Evaluation of affine function

• Online complexity is governed by point location
  - Function of number of regions in cell complex
  - Milli- to microseconds possible if small number of regions

\[ u^*(x) = x^T \begin{bmatrix} -0.47 \\ -1.37 \end{bmatrix} + 0.98 \]
Verifiable Control Synthesis

**Offline**
- Explicit MPC
  - Small problems
  - Simple
  - High speed
- Approx. Explicit MPC

**Online**
- 1st Order–Fast Gradient
- Interior Point Opt.
Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit MPC</td>
<td>1st Order–Fast Gradient</td>
</tr>
</tbody>
</table>
| • Small problems  
• Simple  
• High speed | |
| Approx. Explicit MPC | Interior Point Opt. |
| • Medium size  
• Specified complexity  
• High speed | |
Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explicit MPC</td>
<td>1&lt;sup&gt;st&lt;/sup&gt; Order–Fast Gradient</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>• Small problems</td>
<td></td>
</tr>
<tr>
<td>• Simple</td>
<td></td>
</tr>
<tr>
<td>• High speed</td>
<td></td>
</tr>
</tbody>
</table>

| Approx. Explicit MPC| Interior Point Opt.           |
|                    |                               |
| • Medium size      |                               |
| • Specified complexity |                             |
| • High speed      |                               |
Fast Gradient Method: Time bound to $\varepsilon$-optimality

- $\varepsilon$-solution in $i_{\min}$ steps

$$i_{\min} \geq \left[ \frac{\ln \frac{\delta}{\varepsilon}}{-\ln \left(1 - \sqrt{\frac{1}{\kappa}}\right)} \right]$$

- $\kappa$ condition number
- $\delta$ measure of initial error

**Cold start**

$$\delta = LR^2/2$$

- $R$: radius of feasible set
- Easy to compute

**Warm start**

$$\delta = 2 \max_{x \in X_0} J_N(U_w; x) - J_N^*(x)$$

- $U_w$: Warm start solution
- Worst distance measured in terms of initial cost
- Hard to compute

References:
- [Y. Nesterov, 1983]
- [S. Richter, C.N. Jones and M. Morari, CDC 2009]
## Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explicit MPC</strong></td>
<td><strong>1\textsuperscript{st} Order–Fast Gradient</strong></td>
</tr>
<tr>
<td>• Small problems</td>
<td>• Large problems</td>
</tr>
<tr>
<td>• Simple</td>
<td>• Less simple</td>
</tr>
<tr>
<td>• High speed</td>
<td>• Deterministic computation time</td>
</tr>
<tr>
<td><strong>Approx. Explicit MPC</strong></td>
<td><strong>Interior Point Opt.</strong></td>
</tr>
<tr>
<td>• Medium size</td>
<td></td>
</tr>
<tr>
<td>• Specified complexity</td>
<td></td>
</tr>
<tr>
<td>• High speed</td>
<td></td>
</tr>
</tbody>
</table>
## Verifiable Control Synthesis

<table>
<thead>
<tr>
<th>Offline</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Explicit MPC</strong></td>
<td><strong>1st Order–Fast Gradient</strong></td>
</tr>
<tr>
<td>• Small problems</td>
<td>• Large problems</td>
</tr>
<tr>
<td>• Simple</td>
<td>• Less simple</td>
</tr>
<tr>
<td>• High speed</td>
<td>• Deterministic computation time</td>
</tr>
<tr>
<td><strong>Approx. Explicit MPC</strong></td>
<td><strong>Interior Point Opt.</strong></td>
</tr>
<tr>
<td>• Medium size</td>
<td></td>
</tr>
<tr>
<td>• Specified complexity</td>
<td></td>
</tr>
<tr>
<td>• High speed</td>
<td></td>
</tr>
</tbody>
</table>
Convex Multistage Problem

minimize \[ \sum_{i=1}^{N} \frac{1}{2} v_i^T H_i v_i + f_i^T v_i \]
subject to \[ v_i \leq v_i \leq \bar{v}_i \]
\[ A_i v_i \leq b_i \]
\[ v_i^T Q_{i,j} v_i + l_{i,j}^T v_i \leq r_{i,j} \]
\[ C_i v_i + D_{i+1} v_{i+1} = c_i \]

where \( H_i, Q_{i,j} \geq 0 \) and \( A_i \) has full row rank

Captures MPC, MHE, portfolio optimization, spline optimization, etc.
### The FORCES Code Generator

#### Multistage QCQP

\[
\begin{align*}
\text{min} & \quad \sum_{i=1}^{N} \frac{1}{2} v_i^T H_i v_i + f_i^T v_i \\
\text{s.t.} & \quad z_i \leq v_i \leq \bar{v}_i \\
& \quad A_i v_i \leq b_i \\
& \quad v_i^T Q_{i,j} v_i + t_{i,j}^T v_i \leq r_{i,j} \\
& \quad C_i v_i + D_{i+1} v_{i+1} = c_i
\end{align*}
\]

#### Problem description

```matlab
stage = MultiStageProblem(N+1);
for i = 1:N+1
    % dimensions
    stages(i).dims.n = 10;
    stages(i).dims.r = 5;
    stages(i).dims.lb = 3;
    % cost
    stages(i).cost.H = Hi;
    stages(i).cost.f = fi;
    % inequalities
    stages(i).ineq.b.lbidx = 3:5;
    stages(i).ineq.b.lb = zeros(3,1);
    % equalities
    stages(i).eq.C = Ci;
    stages(i).eq.c = ci;
    stages(i).eq.D = Di;
end
generateCode(stages);
```

#### Embedded Hardware

- C code generation of primal-dual Mehrotra interior point solvers
- LPs, QPs, QCQPs
- Parametric problems
- Multi-core platforms
- Library-free
- Available: forces.ethz.ch

#### Generated Code

Solver (ANSI-C)

- solver.h
- solver.c
- solver.m
- solvermex.c
- makemex

MATLAB MEX interface for rapid prototyping
Speedups Compared to IBM’s CPLEX

- Standard MPC problem for oscillating chain of masses (on Intel i5 @3.1 GHz)
- CPLEX N/A on embedded systems

FORCES
Solve time: 90 μs
Code size: 52 KB

CPLEX
Solve time: 5470 μs
Code size: 11700 KB
Some Early Users of FORCES

Nonlinear MPC & MHE with ACADO
Milan Vukov, KU Leuven, 2012

MPC for Wind Turbines
Marc Guadayol, ALSTOM, 2012

Quadrotor Control
Marc Müller, IDSC, ETH Zurich, 2012

Adaptive MPC for Belt Drives
Kim Listmann, ABB Ladenburg, 2012
Micro-scale Race Cars

- 1:43 scale cars – 106mm
- Top speed: $5 \text{ m/s}$
  
  (774 km/h scale speed)
- Full differential steering
- Position-sensing: External vision
- 50 Hz sampling rate

Project goals:
1. Plan optimal path online in dynamic race environment
2. Demonstrate real-time control optimizing car performance
3. Beat all human opponents!

Challenges:
- Interaction with multiple unpredictable opponents
- Highly nonlinear dynamics
- High-speed planning and control
System Details

Camera System

- Infrared spotlight
- Reflectors on cars
- 3.36 mm accuracy
- 100 Hz update rate at 1024 x 1200 pixels

Embedded Board

- Custom built electronics
- Bluetooth communication
- IMUs & Gyro
- H-bridges for DC Motors
- ARM Cortex M4

Tracks

- Custom built high grip track
- Standard RCPtracks track

http://orcaracer.ethz.ch
Car Model Analysis

Analysis of zero accelerations (for fixed velocities) yield bifurcation:

Stationary velocities for $v_x = 1.5$

Stationary trajectories are circles, where the radius is related to the forward velocity and the yaw rate.
Path Planner and MPC

- Trajectory tracking MPC formulation:
  - re-linearizing the dynamics around trajectory
  - linear constraints to stay on track / avoid obstacles
  - 6 states, 2 inputs, horizon 16

\[
\begin{align*}
\min & \sum_{i=0}^{N-1} (x_i - x_i^{\text{ref}})^T Q (x_i - x_i^{\text{ref}}) + (u_i - u_i^{\text{ref}})^T R (u_i - u_i^{\text{ref}}) + (x_N - x_N^{\text{ref}})^T P (x_N - x_N^{\text{ref}}) \\
\text{s.t.} & \quad x_0 = x \\
& \quad x_{i+1} = A_i x_i + B_i u_i + g_i \\
& \quad G_i x_i \leq h_i \\
& \quad (x_i, u_i) \in X \times U
\end{align*}
\]

- Computation times on Intel Core i7 CPU @2.5 GHz:

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference trajectory gen.</td>
<td>4.13 ms</td>
</tr>
<tr>
<td>Problem data gen.</td>
<td>1.19 ms</td>
</tr>
<tr>
<td>Solving QP with CPLEX</td>
<td>6.00 ms</td>
</tr>
<tr>
<td>Solving QP with FORCES</td>
<td>0.83 ms</td>
</tr>
</tbody>
</table>

3.3 ms on ARM Cortex A9 @1.7 GHz
Autonomous RC Racing Using FORCES

- Reference tracking MPC solved in 3.3 ms on ARM Cortex A9 based chip
## Applications by the Automatic Control Lab

<table>
<thead>
<tr>
<th>Time</th>
<th>Application</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 ns</td>
<td>Multi-core thermal management (EPFL)</td>
<td>[Zanini et al 2010]</td>
</tr>
<tr>
<td>10 µs</td>
<td>Voltage source inverters</td>
<td>[Mariethoz et al 2008]</td>
</tr>
<tr>
<td>20 µs</td>
<td>DC / DC converters (STM)</td>
<td>[Mariethoz et al 2008]</td>
</tr>
<tr>
<td>25 µs</td>
<td>Direct torque control (ABB)</td>
<td>[Papafotiou 2007]</td>
</tr>
<tr>
<td>50 µs</td>
<td>AC / DC converters</td>
<td>[Richter et al 2010]</td>
</tr>
<tr>
<td>5 ms</td>
<td>Electronic throttle control (Ford)</td>
<td>[Vasak et al 2006]</td>
</tr>
<tr>
<td>20 ms</td>
<td>Traction control (Ford)</td>
<td>[Borrelli et al 2001]</td>
</tr>
<tr>
<td>40 ms</td>
<td>Micro-scale race cars</td>
<td></td>
</tr>
<tr>
<td>50 ms</td>
<td>Autonomous vehicle steering (Ford)</td>
<td>[Besselmann et al 2008]</td>
</tr>
<tr>
<td>500 ms</td>
<td>Energy efficient building control (Siemens)</td>
<td>[Oldewurtel et al 2010]</td>
</tr>
</tbody>
</table>
Online MPC at Megahertz Rates using FPGAs

Hardware architectures for fast-gradient and ADMM methods parameterized by:

- Degree of parallelism
- Fixed-point computing precision

Fixed-point arithmetic analysis:
- ensures reliable operation
- Solution accuracy specs → # bits

~250 variable optimization problem:

State-of-the-art: FPGAs:
- ~5 ms  → ~0.5 μs
- Desktop @ 3GHz  → Embedded @ 400MHz
- ~70 Watts  → ~5 Watts
Online MPC at Megahertz Rates using FPGAs

MPC of an Atomic Force Microscope, in collaboration with IBM

✓ Same tracking performance as double precision state-of-the-art solver
✓ 700 kilohertz on 1 Watt FPGA
✓ >1 megahertz on high-performance FPGA

Jerez, Goulart, Richter, Constantinides, Kerrigan, Morari
"Embedded Predictive Control on an FPGA using the Fast Gradient Method", in ECC 2013
Conclusions

• Computation technology is not limiting the application of MPC at any speed for any size problem
• When and where to employ MPC in industry is still a matter of judgment (modeling, maintenance, robustness)
Thank you, Keith!