Model Predictive Control on an FPGA: 
*Aerospace and Space Scenarios*

Edward Hartley (edward.hartley@eng.cam.ac.uk)

*Workshop on Embedded Optimisation EMBOPT 2014, IMT Lucca*
Monday 8th September 2014: 14:00–15:00
Introduction

1. What is MPC? *(You probably already know)*
   What is an FPGA?

2. Interior point QP-based MPC on an FPGA, for aircraft control

3. Interior point LP-based VH-LTV-MPC on an FPGA, for medium-range spacecraft rendezvous

4. First-order QP-based MPC on an FPGA, for terminal spacecraft rendezvous

5. Conclusions and lessons learnt
Introduction

Model Predictive Control

"Setup"

- Prediction model
- Objective function
- Constraints

"Online tasks"

- Optimal control problem
- Optimisation algorithm

Observer (KF)

"Real world"

- Disturbances
- Plant

"Real world"
Introduction

Model Predictive Control

"Online tasks"

- Prediction model
- Objective function
- Constraints

"Setup"

- Optimal control problem
- Optimisation algorithm

Observer (KF)

"Real world"

- Disturbances
- Plant

"Online tasks"
Field Programmable Gate Array

- Programmable hardware
- Contains many logic blocks:
  - Lookup tables, Flip-flops, RAM, Dedicated Multipliers
- User specifies how these should be connected together

- Implementing a circuit for an algorithm on an FPGA is not like programming a microprocessor
  - Multiple clocks
  - Parallelism
  - Timing
  - Custom numerical representations
Introduction

...and what’s this got to do with MPC?

Raw speed

- Exploit parallelism
- Make the controller latency ever lower
- Control fast processes

Latency vs. clock rate

- Achieve controller latency similar to running MPC on a desktop PC, but on embedded hardware at much lower clock rates

Embeddability

- System on a chip
- Power consumption advantages?
Introduction

... and what's this got to do with MPC?

Raw speed

- Exploit parallelism
- Make the controller latency ever lower
- Control fast processes

Latency vs. clock rate

- Achieve controller latency similar to running MPC on a desktop PC, but on embedded hardware at much lower clock rates

Embeddability

- System on a chip
- Power consumption advantages?
Aerospace Application

Scenario Description

Control Objective

- MPC: Control roll and pitch of B747
- Use all actuators and all state measurements
- Outer loop provides roll/pitch setpoints to track altitude/yaw trajectory
MIMO aircraft control

- **12 states:**
  - Roll rate, Pitch rate, Yaw Rate
  - Airspeed, AoA, Sideslip
  - Roll, Pitch
  - 4 engine dynamics (1st order lag)

- **17 inputs:**
  - 4 elevators, 1 trimmable horizontal stabiliser
  - 4 ailerons, 2 spoiler banks
  - 2 rudders
  - 4 engine setpoints

- **34 constraints:** Upper/lower bound on each input
Aerospace Application

MPC description

Why MPC
- Multivariable control
- Constraints
- Reconfiguration (not addressed here)

... on an FPGA?
- Embeddability
- Latency vs clock rate

MPC parameters
- Quadratic cost
- Tracking
- No state constraints
- Prediction horizon $N = 5$ or $N = 12$
- Uncondensed formulation (sparse with equality constraints)
Aerospace Application

\textit{MPC formulation} \\

\textbf{Finite horizon OCP}

\[ \min_{\theta} \| \delta x_N - \delta x_s \|_P^2 + \sum_{i=0}^{N-1} \left( \| \delta x_i - \delta x_s \|_Q^2 + \| \delta u_i - \delta u_s \|_R^2 \right) \] (1a)

subject to: \[ \delta x_0 = \delta \hat{x}(k) \] (1b)
\[ \delta x_{i+1} = A \delta x_i + B \delta u_i + B_d \hat{w}(k), \ i = 0, \ldots, N - 1 \] (1c)
\[ \delta u_{\text{min}} \leq \delta u_i \leq \delta u_{\text{max}}, \quad i = 0, \ldots, N - 1. \] (1d)

• (The \( \delta \) indicates deviation from trim point used for linearisation)
• \( \| \cdot \|_X^2 \triangleq \cdot ^T X \).
**OCP as a QP**

\[
\min \frac{1}{2} \theta^T H \theta + h^T \theta \quad \text{subject to} \quad G \theta \leq g, \ F \theta = f.
\]

**where...**

\[
H \triangleq 2 \left( I_N \otimes (Q \oplus R) \right) \oplus P
\]
\[
G \triangleq \begin{bmatrix}
I_N \otimes \begin{bmatrix} 0_{m \times n} & I_m \\ 0_{m \times n} & -I_m \end{bmatrix}, & 0_{2Nm \times n}
\end{bmatrix}
\]
\[
F \triangleq \begin{bmatrix}
-I_n \\ A & B & -I_n \\ \vdots & \vdots & \vdots \\ & & -I_n
\end{bmatrix}
\]
\[
h \triangleq \begin{bmatrix}
-I_N^T \otimes \left[ \delta x_T^Q \quad \delta u_T^R \right] - \delta x_T^P
\end{bmatrix}^T
\]
\[
g \triangleq 1_N \otimes \begin{bmatrix} \delta u_{\text{max}}^T \quad -\delta u_{\text{min}}^T \end{bmatrix}^T
\]

- \( \oplus \): Direct sum
- \( \otimes \): Kronecker product
Aerospace Application

Primal-Dual Interior Point Method

Initialisation

Compute residuals

Linearise relaxed KKT + block elimination

Solve linear system

Update iterate

$k \leq k_{\text{max}}$?

yes

no

Stop

Critical path
**Computational bottleneck**

\[
\begin{bmatrix}
H + \Phi_k & F^T \\
F & 0
\end{bmatrix}
\begin{bmatrix}
\Delta \theta_k \\
\Delta \nu_k
\end{bmatrix}
= b_k
\]

**Conventional approach**

- Factorise + substitute
- Many divisions
- Cannot terminate early
- Difficult to parallelise

**Alternative approach applied**

- Solve using iterative MINRES (Minimum Residual) algorithm
- Iterations vs accuracy
- Matrix-vector multiplication: very parallelisable
- Sparse structure (and most elements do not change)
- Sensitive to conditioning
- Sensitive to precision
- Inefficient without parallelisation
• Linear system conditioning important
• Diagonal online preconditioner
• **Offline model scaling:** Consider the substitution

\[
\begin{align*}
A &\leftarrow T_Q A T_Q^{-1} & B &\leftarrow T_Q B T_R^{-1} \\
Q &\leftarrow T_Q^{-1} Q T_Q^{-1} & R &\leftarrow T_R^{-1} R T_R^{-1} \\
P &\leftarrow T_Q^{-1} P T_Q^{-1}
\end{align*}
\]

for diagonal \(T_Q > 0\) and \(T_R > 0\), and corresponding scalings on the constraints. Choose \(T_Q, T_R\) to approximately normalise the 2-norms of the rows of

\[
\begin{bmatrix}
H + \text{diag} & F^T \\
F & 0
\end{bmatrix}
\]

(Heuristic, but effective)
Aerospace Application

Problem Scaling: importance

(a) No preconditioning
(b) Online preconditioning only

- Based on closed-loop simulation in software
- Timing estimate for 250 MHz FPGA based on analytical formula
- Without preconditioning, MINRES-based solver is disastrous
- With online preconditioning good results if enough MINRES iterations per PDIP iteration
Aerospace Application

Problem Scaling: importance

- With offline preconditioning, solution quality for low MINRES iteration counts improves
- Best results with both offline and online preconditioning
Aerospace Application

Implementation details

General Implementation

- Pure VHDL solver coded by Juan Jerez (Imperial, now ETH).
- Prediction model hard coded in ROM.
- Connect to MicroBlaze for HIL setup
- Software server on MicroBlaze enables communication over ethernet with plant model in Simulink, using UDP/IP
Aerospace Application

Hardware-in-the-loop Setup

- Nonlinear Plant
- Observer
- Reference Traj.

Simulink

Desktop/Laptop Computer

UDP/IP
100 Mbit Ethernet

ML605 Evaluation Board

Virtex 6 LX240T

Sequential stage

Parallel MINRES accelerator

QP Solver

Microblaze

Server

lwip

AXI Bus

Target calculator

Ether-ether PHY

Ether-net MAC
### Aerospace Application

**Computation time comparison**

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Relative numerical accuracy</th>
<th>Mean</th>
<th>Max Solution time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>cost</td>
<td>QP (ms)</td>
</tr>
<tr>
<td>QP Solver</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F /P-MINRES</td>
<td>32 12 51 9.67 × 10^{-4} 3.02 × 10^{-5}</td>
<td>5.2246</td>
<td>12</td>
</tr>
<tr>
<td>PC/RWR1998</td>
<td>64 12 – – – 5.89 × 10^{-3} 1.69 × 10^{-4}</td>
<td>5.2247</td>
<td>23</td>
</tr>
<tr>
<td>PC/FORCES</td>
<td>64 12 – – – 3.83 × 10^{-3} 7.31 × 10^{-5}</td>
<td>5.2249</td>
<td>1911</td>
</tr>
<tr>
<td>UB/FORCES</td>
<td>32 12 – – 9.10 × 10^{-4} 2.95 × 10^{-5}</td>
<td>5.2203</td>
<td>4</td>
</tr>
<tr>
<td>F /P-MINRES</td>
<td>32 5 30 – – – 1.04 × 10^{-3} 1.84 × 10^{-5}</td>
<td>5.2203</td>
<td>3</td>
</tr>
<tr>
<td>PC/RWR1998</td>
<td>64 5 – – – 5.00 × 10^{-3} 1.24 × 10^{-4}</td>
<td>5.2207</td>
<td>6</td>
</tr>
<tr>
<td>PC/FORCES</td>
<td>64 5 – – – 5.00 × 10^{-3} 1.24 × 10^{-4}</td>
<td>5.2207</td>
<td>6</td>
</tr>
<tr>
<td>UB/CVXGEN</td>
<td>32 5 – – – 5.00 × 10^{-3} 1.24 × 10^{-4}</td>
<td>5.2205</td>
<td>823</td>
</tr>
<tr>
<td>UB/FORCES</td>
<td>32 5 – – – 5.00 × 10^{-3} 1.24 × 10^{-4}</td>
<td>5.2205</td>
<td>823</td>
</tr>
</tbody>
</table>

(FPGA QP solver (F) running at 250 MHz, PC (PC) at 2.4 GHz and MicroBlaze (UB) at 100 MHz. (–) indicates a baseline. (??) indicates that meaningful data for control could not be obtained).

Predictive control of a Boeing 747 aircraft using an FPGA.
In *Proceedings of the IFAC conference on Nonlinear Model Predictive Control*,
August 23–27 2012.
doi: 10.3182/20120823-5-NL-3013.00016

Predictive control using an FPGA with application to aircraft control.
doi: 10.1109/TCST.2013.2271791
Rendezvous and Capture Scenario (Not to scale!)

- Active control of “chaser” to capture passive target
- Use fuel efficiently
- Cope with parametric and navigation uncertainty
- Work in circular or elliptical orbit
Spacecraft Application I

Why MPC on an FPGA?

Why MPC?

- Optimisation in the loop
- Natural constraint handling
- Potential for improved autonomy?

Why on an FPGA?

- MPC more computationally demanding (online) than PID, LQR, $H_\infty$.
- Power, solar radiation
- Clock frequency vs. parallelism vs. numerical precision vs. resource usage
- (Cycle-accurate timing)
Control objective

- Bring chaser spacecraft from 10 km to 100 m in a fuel efficient manner
- Stop at a sequence of "holding points" reducing separation from target
  - (These are periodic trajectories centred at prescribed separations.)
MPC Formulation to reach next holding point

- Elliptical orbit $\implies$ LTV prediction model (Yamanaka-Ankersen)
- Fuel minimisation proportional to thrust $\implies$ $1-$norm cost
- Completion-type problem $\implies$ variable horizon

\[
J^* = \min_{N, \theta} \sum_{i=0}^{N-1} (1 + \gamma \|u_i\|_1) \tag{2a}
\]

\[
\text{s.t.} \quad x_0 = x(k), \tag{2b}
\]
\[
x_{i+1} = A_ix_i + B_iu_i, \quad i \in \{0, \ldots, N - 1\}, \tag{2c}
\]
\[
x_N = x_T(k + N), \tag{2d}
\]
\[
0 \leq u_i \leq u_{\text{max}}, \quad i \in \{0, \ldots, N - 1\}, \tag{2e}
\]
\[
N \leq N_{\text{max}}, \tag{2f}
\]
Variable horizon MPC controller

Initialization:
1. \( J_{opt} = \infty, N_{opt} = 0, \theta_{opt} = 0; \)
for \( j = 1 \) to \( N_{max} \)
   2. Calculate \( A_j, B_j \) using Yamanaka-Ankersen (2002) equations;
   3. Solve (2) s.t. \( N = j \). If feasible, \( J_j = J^* \), else \( J_j = \infty \); if \( J_j < J_{opt} \)
      4. \( J_{opt} = J_j, N_{opt} = j, \theta_{opt} = \theta^*; \)
   end if
end for.
5. Return \( u_0 \) from \( \theta_{opt} \).
Spacecraft Application I

Custom QP Solver

- Initialisation
- Compute residuals
- Linearise relaxed KKT + block elimination
- Solve linear system
- Update iterate

$k \leq k_{\text{max}}$?

Custom circuit

- Time varying data
- Varying problem size
- Can’t use same design as before
- Still use MINRES
- Mixed software/hardware

Stop

yes

no
Software on MicroBlaze

- Iteration counting and control logic
- Prediction model computation
- Some mathematical operations
- Network communications
- Written in C using Xilinx EDK

Custom Circuit

- Communicate with MicroBlaze over AXI
- KKT linearisation and block elimination (RAM latency) and majority of MINRES method
  - Xilinx System Generator for DSP (Graphical), Floating point
- Lanczos algorithm (heavy computation)
  - Simulink HDL Coder (Graphical), Fixed point
Spacecraft Application I

Custom QP Solver Hardware Architecture

Outside world

Ethernet

Xilinx MicroBlaze

AXI-lite

FPGA

Linear system builder and preconditioner

Update QR, Givens Rot^n, Update Sol^n

Lanczos

MINRES solver
Spacecraft Application I
Custom QP Solver Hardware Architecture

Linear system builder and preconditioner

MINRES solver

Update QR
Update Soln
Lanczos

Givens Rot
Spacecraft Application I

Custom QP Solver Hardware Architecture

- FPGA
- Outside world
- Xilinx
- MicroBlaze
- Ethernet

MINRES solver

- Linear system builder and preconditioner
- Update QR
- Givens Rot
- Update Sol
- Lanczos
Spacecraft Application I

Custom QP Solver Hardware Architecture

Linear system builder and preconditioner

Update QR, Givens Rot^n, Update Sol^n

Lanczos

MINRES solver
Spacecraft Application

Custom QP Solver Hardware Architecture

FPGA

Outside world

Ethernet

MicroBlaze

MINRES solver

Update QR, Givens Rot\(^n\)
Update Sol\(^n\)

Lanczos

Linear system builder and preconditioner

AXI-lite

MINRES solver
Spacecraft Application I

Custom QP Solver Hardware Architecture

Outside world

FPGA

Ethernet

MINRES solver
Spacecraft Application

Custom QP Solver Hardware Architecture

FPGA

Outside world

Ethernet

MicroBlaze

MINRES solver

Update QR, Givens Rot^n, Update Sol^n

Lanczos

Linear system builder and preconditioner

AXI-lite

27/44
Advantages

• Visual documentation
• Simplified communication, maintenance and re-use
• MATLAB/SIMULINK familiar to control engineers
• Rapid prototyping and testing of fixed-point arithmetic
• Still a fair amount of control at the register level

Disadvantages

• Dependent on a longer tool-chain
• Still thinking at a register level
• Possible loss of flexibility (but can still include custom HDL or IP cores, or combine simpler operations)
• Division/square root sometimes needs hand-implementation in fixed point to avoid too many layers of logic
Spacecraft Application I

Closed-loop trajectories

FPGA-in-the-loop simulation

- Rendezvous trajectories for 100 simulations
- Similar HIL setup as for B747
  - MPC on FPGA, Nonlinear Plant in Simulink
- Plot shows 100 simulations with scattered parameters
## Computation time breakdown

<table>
<thead>
<tr>
<th>Description</th>
<th>Runs</th>
<th>Time (ms)</th>
<th>Alg.</th>
<th>Resource used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute/transfer LTV model</td>
<td>1</td>
<td>5.01</td>
<td>5.01</td>
<td>–</td>
</tr>
<tr>
<td>Initialise PDIP</td>
<td>1</td>
<td>0.04</td>
<td>0.04</td>
<td>1</td>
</tr>
<tr>
<td>Calc. $\mu_k$ and $w_k$</td>
<td>20</td>
<td>0.14</td>
<td>2.82</td>
<td>3–4</td>
</tr>
<tr>
<td>Calc. residual</td>
<td>20</td>
<td>0.15</td>
<td>3.05</td>
<td>5</td>
</tr>
<tr>
<td>Transfer to PCORE</td>
<td>20</td>
<td>0.11</td>
<td>2.15</td>
<td>5</td>
</tr>
<tr>
<td>Linear system &amp; MINRES</td>
<td>20</td>
<td>1.00</td>
<td>19.94</td>
<td>6–9</td>
</tr>
<tr>
<td>Transfer $\Delta\theta_k$</td>
<td>20</td>
<td>0.12</td>
<td>2.36</td>
<td>9</td>
</tr>
<tr>
<td>Wait for next result</td>
<td>20</td>
<td>0.00</td>
<td>0.02</td>
<td>10</td>
</tr>
<tr>
<td>Transfer $G(\theta_k + \Delta\theta_k) - g$</td>
<td>20</td>
<td>0.08</td>
<td>1.50</td>
<td>10</td>
</tr>
<tr>
<td>Sanity check for NaN</td>
<td>20</td>
<td>0.17</td>
<td>3.30</td>
<td>–</td>
</tr>
<tr>
<td>Calc. $\Delta s_k$, $\Delta z_k$</td>
<td>20</td>
<td>0.10</td>
<td>1.92</td>
<td>11–12</td>
</tr>
<tr>
<td>Line search</td>
<td>20</td>
<td>0.27</td>
<td>5.44</td>
<td>13</td>
</tr>
<tr>
<td>Update solution</td>
<td>20</td>
<td>0.16</td>
<td>3.18</td>
<td>14</td>
</tr>
<tr>
<td>Check iter. infeas.</td>
<td>20</td>
<td>0.01</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>Check final infeas.</td>
<td>1</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>50.94</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Compare with estimate for full-hardware solver

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>$c = 0$</th>
<th>$c = 20$</th>
<th>$c = 30$</th>
<th>$c = 40$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_c = 100$ MHz</td>
<td>55.43</td>
<td>62.87</td>
<td>66.59</td>
<td>70.31</td>
</tr>
<tr>
<td>$f_c = 200$ MHz</td>
<td>27.71</td>
<td>31.43</td>
<td>33.29</td>
<td>35.15</td>
</tr>
</tbody>
</table>

- Hybrid software/hardware design gives more flexibility in exchange for minor increase in total solution time
E. N. Hartley and J. M. Maciejowski.
Predictive control for spacecraft rendezvous in an elliptical orbit using an FPGA.
In *Proceedings of the European Control Conference*, pages 1359–1364, Zurich, Switzerland, July 17–19 2013a

E. N. Hartley and J. M. Maciejowski.
Field programmable gate array based predictive control system for spacecraft rendezvous in elliptical orbits.
doi: 10.1002/oca.2117

- Second paper also considers a terminal phase up to capture by re-purposing the same solver.
- Remaining slides discuss terminal phase but using an alternative approach.
Reference trajectory

- Relative local reference frame centred on target
- At 200 m, accelerate to 0.2 ms$^{-1}$
- Slow down to 0.1 ms$^{-1}$ at 100 m
- Open-loop drift from 3 m point.

**Figure:** Relative reference frame

**Figure:** Nominal reference trajectory (not to scale)
Spacecraft Application II
Problem Description

Constraints and oddities

- Input constraints (8 N)
- Minimum impulse bit

Uncertainty

- Sensor noise
- Thrust uncertainty
- Model parameter uncertainty

Timing

- Sampling time $T_s = 1 \text{ s}$

Figure: Differential thrust to counteract minimum impulse bit
States $x \in \mathbb{R}^5$ (out-of-plane, and nadir relative position), and in-track, out-of-plane and nadir relative velocities

Inputs $u \in \mathbb{R}^6$ (positive and negative thrusters in three dimensions)

Prediction matrices $(A, B)$ from Hill-Clohessy-Wiltshire Equations

$$
\min_{x_i, u_i} \left( (x_N - r)^T P (x_N - r) + \sum_{i=0}^{N-1} \left( (x_i - r)^T Q (x_i - r) + u_i^T R u_i + \|R \lambda u_i\|_1 \right) \right)
$$

Subject to:

$x_0 = x(k)$

$x_{i+1} = Ax_i + Bu_i$ \quad $i \in \{0, \ldots, N-1\}$

$u_i \geq 0$ \quad $i \in \{0, \ldots, N-1\}$

$u_i \leq u_{\text{max}}$ \quad $i \in \{0, \ldots, N-1\}$.

$Q \geq 0$, $R > 0$, $R \lambda \geq 0$ and diagonal, $P \geq 0$. $N = 20 \implies 120$ decision variables.

We can write this as a convex, bound constrained Quadratic Program (QP)
Spacecraft Application II

**Forming the optimal control problem**

- States \( x \in \mathbb{R}^5 \) (out-of-plane, and nadir relative position), and in-track, out-of-plane and nadir relative velocities
- Inputs \( u \in \mathbb{R}^6 \) (positive and negative thrusters in three dimensions)
- Prediction matrices \((A, B)\) from Hill-Clohessy-Wiltshire Equations

\[
\min_{x_i, u_i} (x_N - r)^T P (x_N - r) + \sum_{i=0}^{N-1} \left( (x_i - r)^T Q (x_i - r) + u_i^T Ru_i + \| R^\lambda u_i \|_1 \right)
\]

Subject to: 
\[
\begin{align*}
  x_0 &= x(k) \\
  x_{i+1} &= Ax_i + Bu_i & i \in \{0, \ldots, N - 1\} \\
  u_i &\geq 0 & i \in \{0, \ldots, N - 1\} \\
  u_i &\leq u_{\text{max}} & i \in \{0, \ldots, N - 1\}.
\end{align*}
\]

- \( Q \geq 0, R > 0, R^\lambda \geq 0 \) and diagonal, \( P \geq 0 \). \( N = 20 \implies 120 \) decision variables.
- We can write this as a convex, bound constrained Quadratic Program (QP)
Spacecraft Application II
Optimal control problem as a constrained QP

Quadratic Program

\[
\min_\theta \frac{1}{2} \theta^T H \theta + f^T \theta
\]

s.t.

\[
\theta_{\text{min}} \leq \theta \leq \theta_{\text{max}}
\]

The Good

- Convex
- Efficient solution algorithms (e.g. FGM)

The Not-So-Good

- More complex than simple feedback
- No analytical solution: iterative methods
- At odds with computational constraints
  \(\text{(Power consumption, radiation hardening)}\)

Design Objective

- Solve the QP \textbf{fast}, but keep \textbf{clock frequency low}
• Fixed point implementation
• Must consider timing. Simulink Delay block \rightarrow \text{register in FPGA}
• Control logic, counters etc. using MATLAB function blocks
• Includes interface for “on-line” loading of matrices
Spacecraft Application II

Synthesis and integration

- Use HDL Coder to generate VHDL
- Interface custom VHDL circuit to Xilinx MicroBlaze Soft Core as a memory mapped peripheral
  - Import HDL-Coder generated VHDL into Xilinx SysGen for DSP as “black box”, and implement shared memory interface
- Synthesise, place-and-route etc.
- Implement server application (in C) on MicroBlaze, to transfer data from payload of UDP packet to custom circuit and return result
- Test closed loop system (in simulation) connecting PC and Controller via Ethernet
Spacecraft Application II

Integration testing — Monte-Carlo Simulation

![Diagram of capture accuracy with tolerance zones: 7.5 cm, 10 cm, and 20 cm.](a) z − y position at capture

**Figure:** FPGA-in-the-loop Control performance over 2000 random simulations with parameter tuning for 7.5 cm accuracy

- 99.55% within 7.5 cm, 100% within 10 cm

![Histograms of completion times and ΔV usage.](b) Histograms
E. N. Hartley, M. Gallieri, and J. M. Maciejowski.  
Terminal spacecraft rendezvous and capture using LASSO MPC.  
doi: 10.1080/00207179.2013.789608

E. N. Hartley and J. M. Maciejowski.  
Graphical FPGA design for a predictive controller with application to spacecraft rendezvous.  
doi: 10.1109/CDC.2013.6760170
Conclusions (I)

Summary

MPC for Aircraft

- Interior point method
- Hardware-based MPC solver solution
- MINRES accelerated in hardware

Lessons Learnt

- Possible to implement MPC for quite large problems on FPGA
- Model scaling important for good algorithm performance
- (Having to redo place and route to accommodate model changes is a pain!)
Conclusions (II)

Summary

MPC for Elliptical Spacecraft Rendezvous

- Time-varying prediction model, Variable horizon
- Non-quadratic cost function (but LP = QP with zero Hessian)
- Interior point method
- MINRES accelerated in hardware. Software supervisory logic.
- Hardware design using graphical methods

Lessons Learnt

- Can implement quite complex custom circuits on FPGA for MPC using high level tools based on those familiar to control engineers
- Have to re-implement some basic things like division and square root in fixed point to get decent speed (might have improved?)
- Mixed hardware/software approach flexible with varying problem data — trade off with control latency
Conclusions (II)

Summary

MPC for Terminal Spacecraft Rendezvous

- LTI model, fixed horizon tracking/regulation problem
- LASSO cost function
- First order method
- FPGA circuit design using graphical methods

Lessons Learnt

- First order method quite simple to implement using HDL Coder!
- Fully fixed point implementation adequately accurate for the application
Thank you for listening!