Microsecond timescale (N)MPC for Power-Electronics applications

Benjamin Stickan

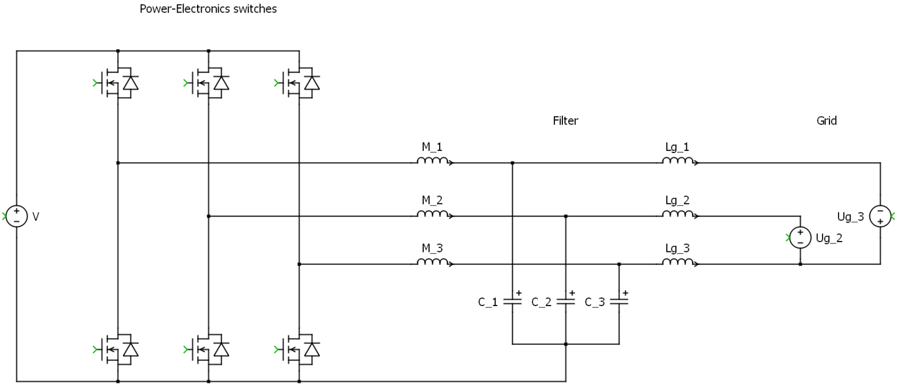
September 7, 2016

Outline

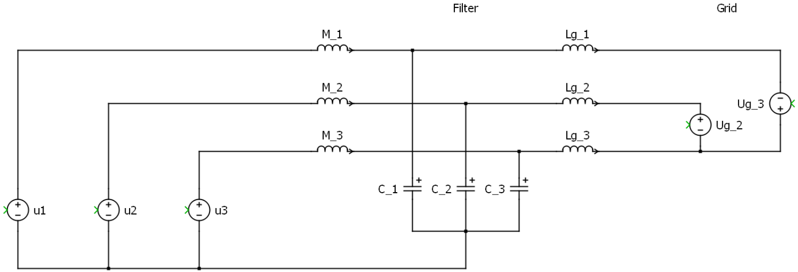
Project introduction

- Current project
  - EU project Netfficient
  - Local energy storage on the island of Borkum
  - Aim: support the grid
  - Reduce energy exchange between island and mainland
  - Storage System with 1 MW inverter (8x 125 kW)
- Recent work
  - Extention of inverter model to 3-phases
  - Development of EMPC-scheme for 125 kW inverter

Current System (1/3)

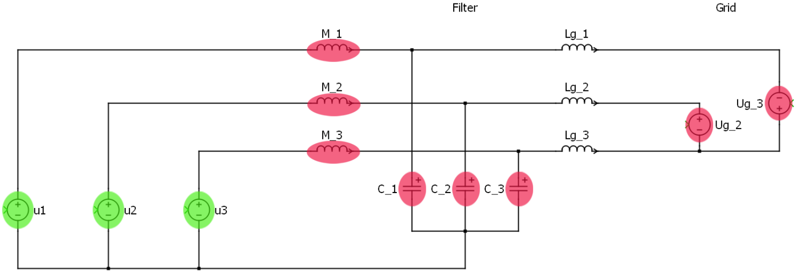


Current System (2/3)



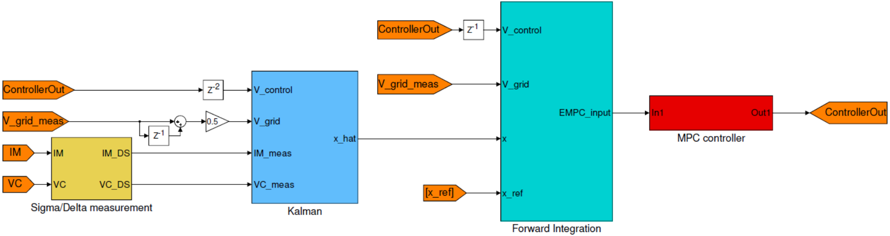
Current System(3/3)

Measurements: Main inductor current, Capacitor Voltage, Grid
Controls: Average input voltages u1-u3

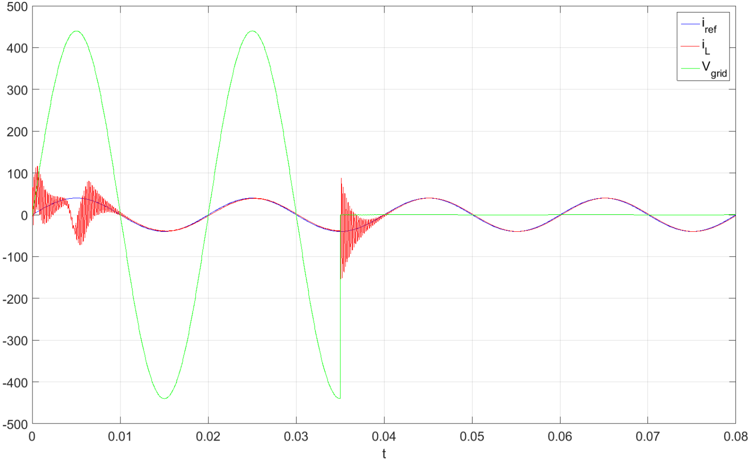


Control Scheme

Control loop frequency: 40 kHz (25 μs)
System states: 8
Estimator states: 20
Control variables: 3

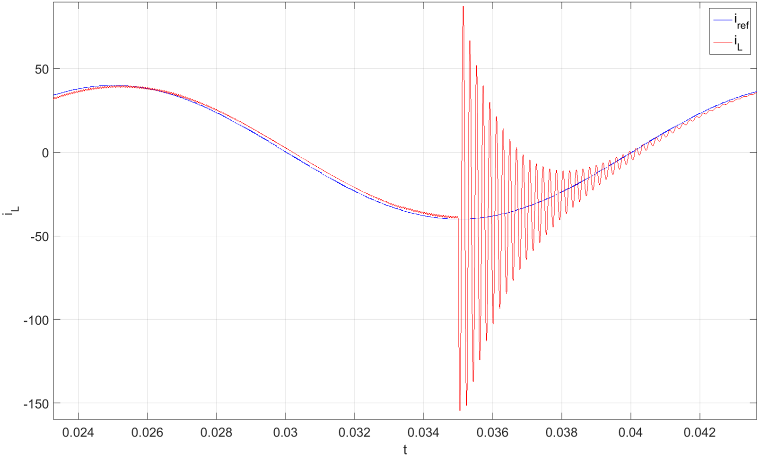


- Controller: EMPC, H=3, move blocking over whole horizon
- Decision variables: 3 (only one phase shown here)



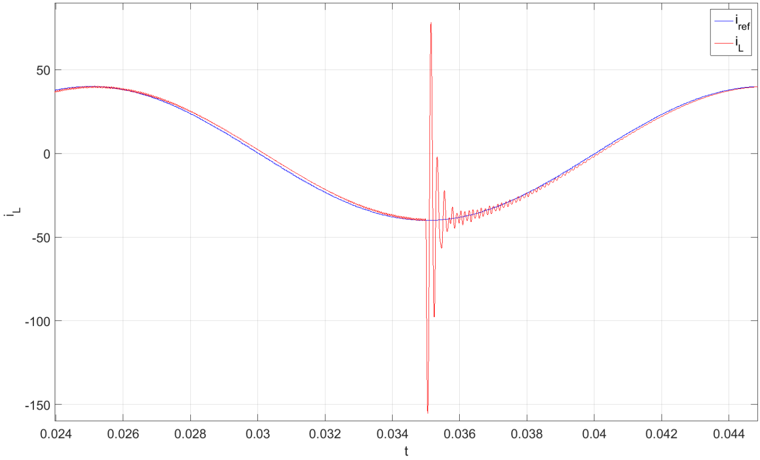
Simulation results (2/3)

- Controller: EMPC, H=3, move blocking over whole horizon
- Decision variables: 3



Simulation results (3/3)

- Controller: qpOASES, H=5
- Decision variables: 15



Questions

- How to solve problems fast and **reliable**?
- Are FPGAs an option?
  - Which methods would be suited?
  - How large is the effort?

C/GMRES

- Developed by Toshiyuki Ohtsuka
- 'Indirect' approach
- Based on Pontryagin's maximum (minimum) principle

OCP formulation

Consider general nonlinear system equations:

$$\dot{x} = f(x(t), u(t), p(t)),$$

$x(t) \in \mathbb{R}^n$	:	state vector
$u(t) \in \mathbb{R}^{m_u}$	:	input vector
$p(t) \in \mathbb{R}^{m_p}$	:	vector of given time dependent variables.

OCP formulation

Consider general nonlinear system equations:

$$\dot{x} = f(x(t), u(t), p(t)),$$

$x(t) \in \mathbb{R}^n$	:	state vector
$u(t) \in \mathbb{R}^{m_u}$	:	input vector
$p(t) \in \mathbb{R}^{m_p}$	:	vector of given time dependent variables.

$$\underset{x,u}{\text{minimize}}: \quad J = \varphi(x^*(T,\ t), p(t+T)) + \int_0^T L(x^*(\tau,\ t),\ u^*(\tau,\ t),\ p(t+\tau))\mathrm{d}\tau,$$

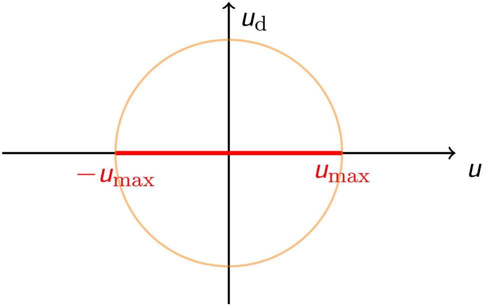
$$\text{Subject to} \quad \begin{cases} x^*_\tau(\tau,\ t) = f(x^*(\tau,\ t),\ u^*(\tau,\ t),\ p(t+\tau)), \\ x^*(0,\ t) = x(t), \\ C(x^*(\tau,\ t),\ u^*(\tau,\ t),\ p(t+\tau)) = 0 \end{cases}$$

Inequality Constraints

How deal with inequality constraints $|\, u \,| \leqslant u_{\max}$?

Employ dummy input u_d:

$$C := (u^2 + u_{\mathrm{d}}^2 - u_{\max}^2) = 0$$



First Order Optimality Conditions

$$
\begin{aligned}
&x_\tau^* = f(x^*,\ u^*,\ p),\\
&x^*(0,\ t) = x(t),\\
&\lambda_\tau^* = -H_x^{\mathrm{T}}(x^*,\ \lambda^*,\ u^*,\ \mu^*, p),\\
&\lambda^*(T,\ t) = \varphi_x^{\mathrm{T}}(x^*(T,\ t), p(t+T)),\\
&H_u(x^*,\ \lambda^*,\ u^*,\ \mu^*, p) = 0,\\
&C(x^*,\ u^*,\ p) = 0,
\end{aligned}
$$

with Hamiltonian

$$H(x^*,\ \lambda^*,\ u^*,\ \mu^*, p) := L(x^*, u^*, p) + \lambda^{\mathrm{T}} f(x^*,\ u^*, p) + \mu^{\mathrm{T}} C(x^*,\ u^*, p),$$

$\lambda^* \in \mathbb{R}^n$	:	costate vector
$\mu^* \in \mathbb{R}^{m_c}$	:	Lagrange multiplier associated with the equality constraints

Discretized Formulation (1/3)

$$
\begin{aligned}
&x^*_{i+1}(t) = x^*_i(t) + f(x^*_i(t),\ u^*_i(t),\ p^*_i(t))\Delta\tau(t),\\
&x^*_0(t) = x(t),\\
&\lambda^*_i(t) = \lambda^*_{i+1}(t) + H_x^{\mathrm{T}}(x^*_i(t),\ \lambda^*_{i+1}(t),\ u^*_i(t),\ \mu^*_i(t),\ p^*_i(t))\Delta\tau(t),\\
&\lambda^*_N(t) = \varphi_x^{\mathrm{T}}(x^*_N(t),\ p^*_i(t)),\\
&H_u(x^*_i(t),\ \lambda^*_{i+1}(t),\ u^*_i(t),\ \mu^*_i(t),\ p^*_i(t)) = 0,\\
&C(x^*_i(t),\ u^*_i(t),\ p^*_i(t)) = 0,
\end{aligned}
$$

$$
\begin{aligned}
\Delta\tau(t) &:= T(t)/N\\
x_i(t)^* &:= x^*(i\Delta\tau, t)\\
p^*_i(t) &:= p(t + i\Delta\tau)
\end{aligned}
$$

Discretized Formulation (2/3)

$$
\begin{aligned}
&x^*_{i+1}(t) = x^*_i(t) + f(x^*_i(t),\ u^*_i(t),\ p^*_i(t))\Delta\tau(t),\\
&x^*_0(t) = x(t),\\
&\lambda^*_i(t) = \lambda^*_{i+1}(t) + H_x^{\mathrm{T}}(x^*_i(t),\ \lambda^*_{i+1}(t),\ u^*_i(t),\ \mu^*_i(t),\ p^*_i(t))\Delta\tau(t),\\
&\lambda^*_N(t) = \varphi_x^{\mathrm{T}}(x^*_N(t),\ p^*_i(t)),\\
&H_u(x^*_i(t),\ \lambda^*_{i+1}(t),\ u^*_i(t),\ \mu^*_i(t),\ p^*_i(t)) = 0,\\
&C(x^*_i(t),\ u^*_i(t),\ p^*_i(t)) = 0,
\end{aligned}
$$

Stack optimization variables in single vector:

$$U(t) := [u_0^{*\mathrm{T}}(t), \mu_0^{*\mathrm{T}}(t), \ \cdots, \ u_{N-1}^{*\mathrm{T}}(t), \mu_{N-1}^{*\mathrm{T}}(t)]^{\mathrm{T}} \in \mathbb{R}^m,$$
$$m := m_u + m_c,$$

Solve:

$$F(U(t), \ x(t), \ t) := \begin{bmatrix} H_u^{\mathrm{T}}(x_0^*, \lambda_1^*, u_0^*, \mu_0^*, p(t)) \\ C(x_0^*, u_0^*, p(t)) \\ \vdots \\ H_u^{\mathrm{T}}(x_{N-1}^*, \lambda_N^*, u_{N-1}^*, \mu_{N-1}^*, p(t+(N-1)\Delta\tau)) \\ C(x_{N-1}^*, u_{N-1}^*, p(t+(N-1)\Delta\tau)) \end{bmatrix} = 0.$$

Continuation Method

$$F(U(t), x(t), t) = 0$$

must be fulfilled at any time t.

$$\Rightarrow \dot{F}(U(t), x(t), t) = 0$$
$$\Leftrightarrow \dot{U} = F_U^{-1}(-F_x \dot{x} - F_t)$$

Find initial solution for $U(0)$ and then integrate over $\dot{U}(t)$.
Drawback: inverse and jacobians needed.

Forward Difference (1/3)

Approximate approach: employ forward difference approximation for products of Jacobians and vectors.

$$D_h F(U, x, t : W, w, \omega) := \frac{F(U + hW, x + hw, t + h\omega) - F(U, x, t)}{h}$$

Forward Difference (2/3)

Approximate

$$\dot{F} = F_U \dot{U} - F_x \dot{x} - F_t$$

by

$$D_h F(U, x, t : \dot{U}, \dot{x}, 1) = A_s F(U, x, t)$$

(A_s is a stable Matrix that stabilizes $F = 0$ to suppress numerical errors that may accumulate through numerical integration)

Forward Difference (3/3)

Result: linear equation of the form $Ax = b$:

$$D_h F(U, x + h\dot{x}, t + h : \dot{U}, 0, 0) = b(U, x, \dot{x}, t),$$

with $b(U, x, \dot{x}, t) := A_s F(U, x, t) - D_h F(U, x, t : 0, \dot{x}, 1)$.

The solution $\dot{U}$ can then be integrated to obtain the next optimal control vector U

GMRES

- How to solve linear equations fast?
  - $\rightarrow$ Ohtsuka uses Krylov subspace based iterative GMRES method
  - Theoretically direct solver (for infinite precision)
  - Matrices and vectors grow with every iteration
  - Can be restarted

Questions

- Similarities between Ohtsuka's discretization method and direct methods?
- How 'exact' is the continuation method? Applicable to Power-Electronics?
- Can the GMRES algorithm generally be implemented on an FPGA?
  - Alternative: other iterative solvers?

Microsecond timescale (N)MPC for Power-Electronics applications

Benjamin Stickan

September 7, 2016