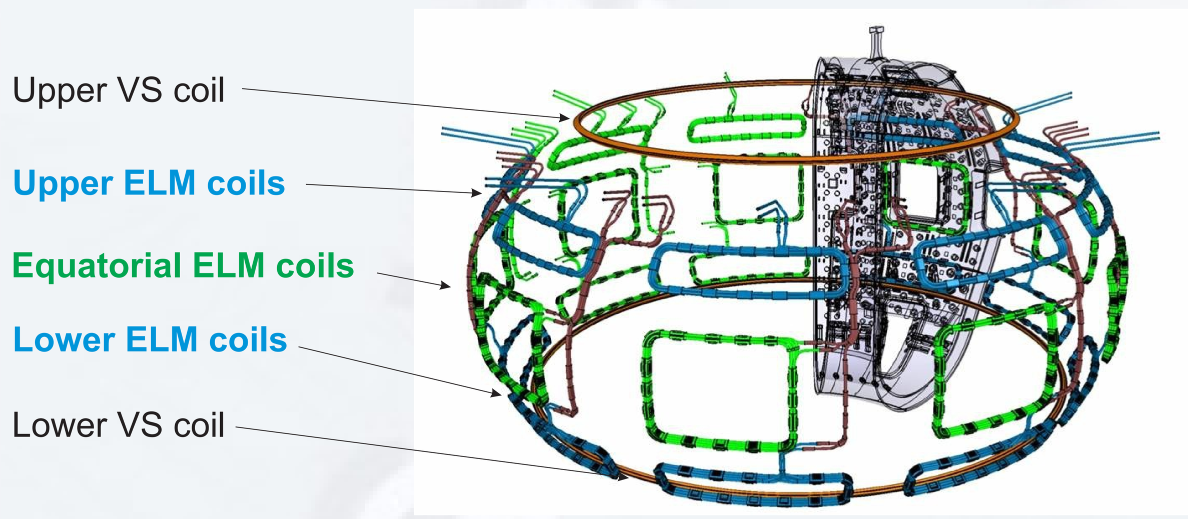
Stabilization of Resistive Wall Modes for ITER using Model Predictive Control

Samo Gerkšič[a], Boštjan Pregelj[a], Marco Ariola[b]

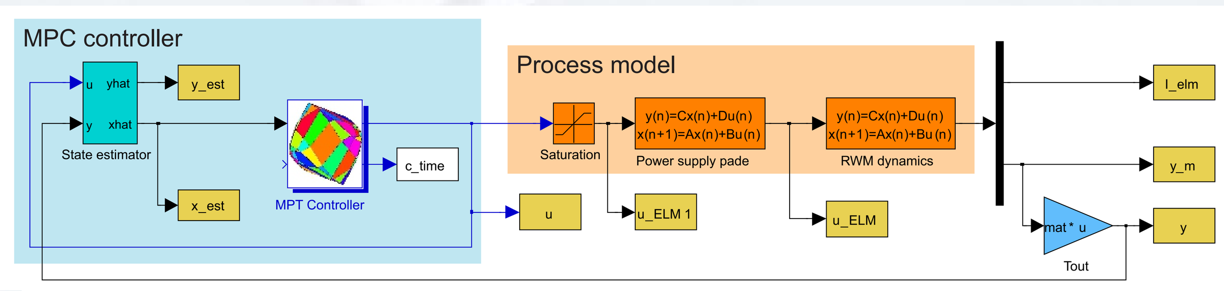
[a] Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
[b] Consorzio CREATE / Università degli Studi di Napoli Parthenope, Naples, Italy

Overview

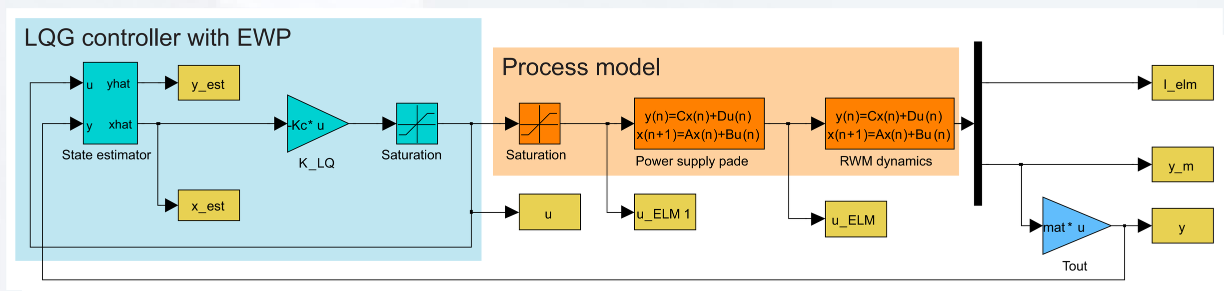
- Resistive Wall Mode (RWM) control for ITER
- Control of the dominant $n = 1$ non-axisymmetric kink mode instability expected in advanced tokamak scenarios
- Model Predictive Control (MPC) using a primal Fast Gradient Method (FGM) quadratic programming (QP) solver suitable for sub-ms sampling time
- Compared to Linear Quadratic Gaussian (LQG) optimal controller with estimator wind-up protection (EWP)



ITER in-vessel coils



MPC control scheme



Reference control scheme - LQG with EWP

Model Predictive Control (MPC)

MPC is an advanced model-based process control technique, based on on-line optimization of predicted future courses of the process signals. Typically, a simplified control model in the discrete-time linear state space form is used, and the MPC controller is used in conjunction with a Kalman state estimator. It is closely related to LQ (linear quadratic) optimal control, and is convenient for control of multivariable processes such as the RWM system.

It is efficient in **handling of constraints** - in the case of ITER RWM, the power-supply voltage (actuator) constraints are considered.

Solving QP problems for MPC using FGM

In MPC, a QP optimisation problem must be solved in each time step of the algorithm, which is difficult with large-scale multivariable systems with fast dynamics.

Using complexity reduction techniques for MPC and C code optimisation, with the primal FGM algorithm the required accuracy was achieved in 20 iterations with peak computation times 0.1 ms using a standard Intel x86 CPU using a single thread.

This is considered sufficiently fast for ITER, but not for experimantation on smaller tokamaks with faster dynamics.

The execution of the FGM algorithm may be accelerated by parallelisation of matrix-vector numerical operations within iterations, but this is not well-suited for the standard CPU architecture with the thread scheduler timescale 10 ms, thread synchronisation takes more time than computation.

Process model

Simulation model:

A linearized state-space dynamic model generated using the CarMa code that combines plasma MHD equations (MARS) with equations describing the currend induced in the conductive structures (CARRIDI)

Inputs: 27 in-vessel coil voltages

Outputs: 6 vertical field sensors; 27 in-vessel coil currents

4000+ dynamic states

Control model:

A reduced-order discrete-time state-space model

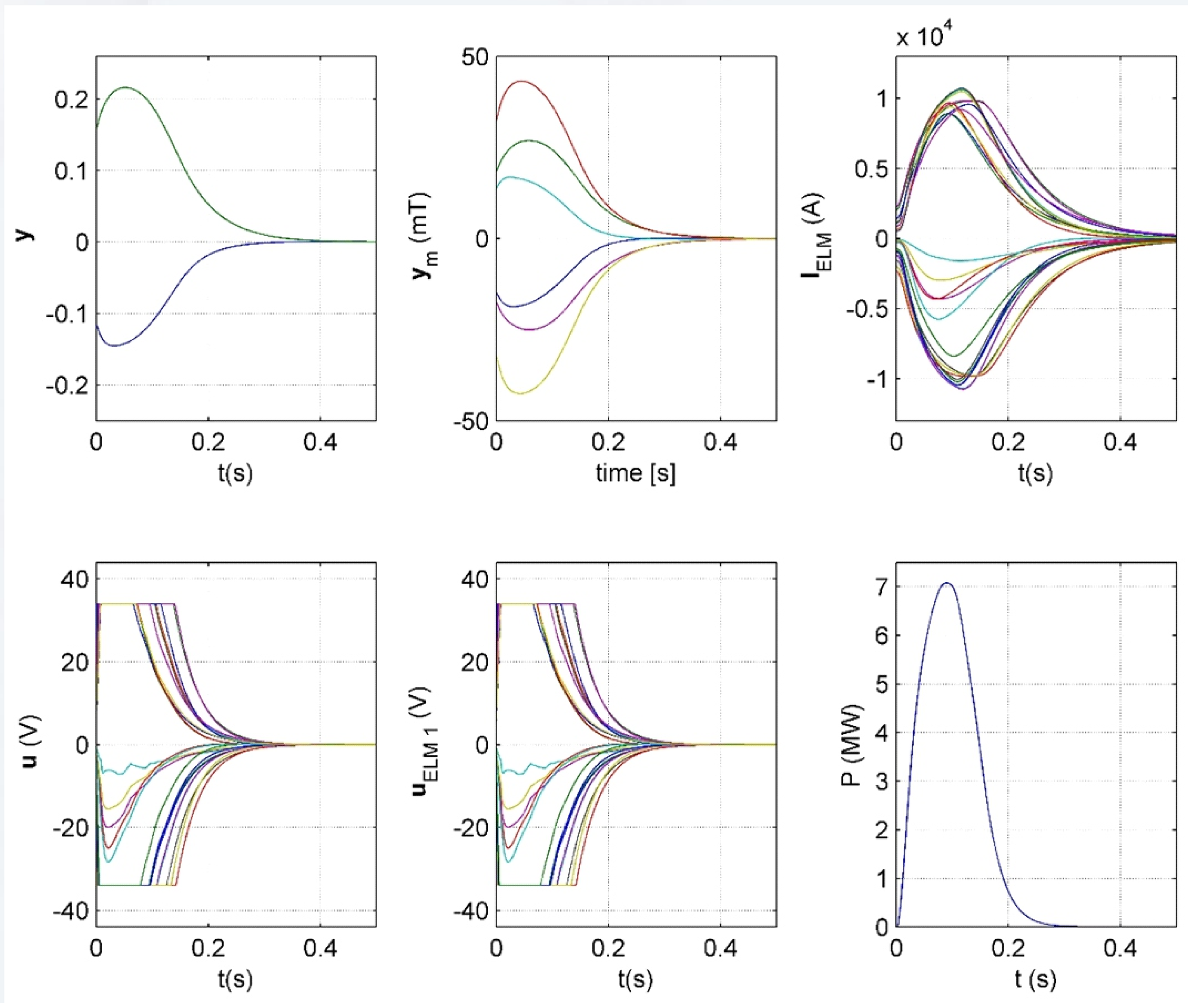
Also includes power-supply dynamics (FOPTD)

Inputs: 27 PS voltage input signals for the IV coils

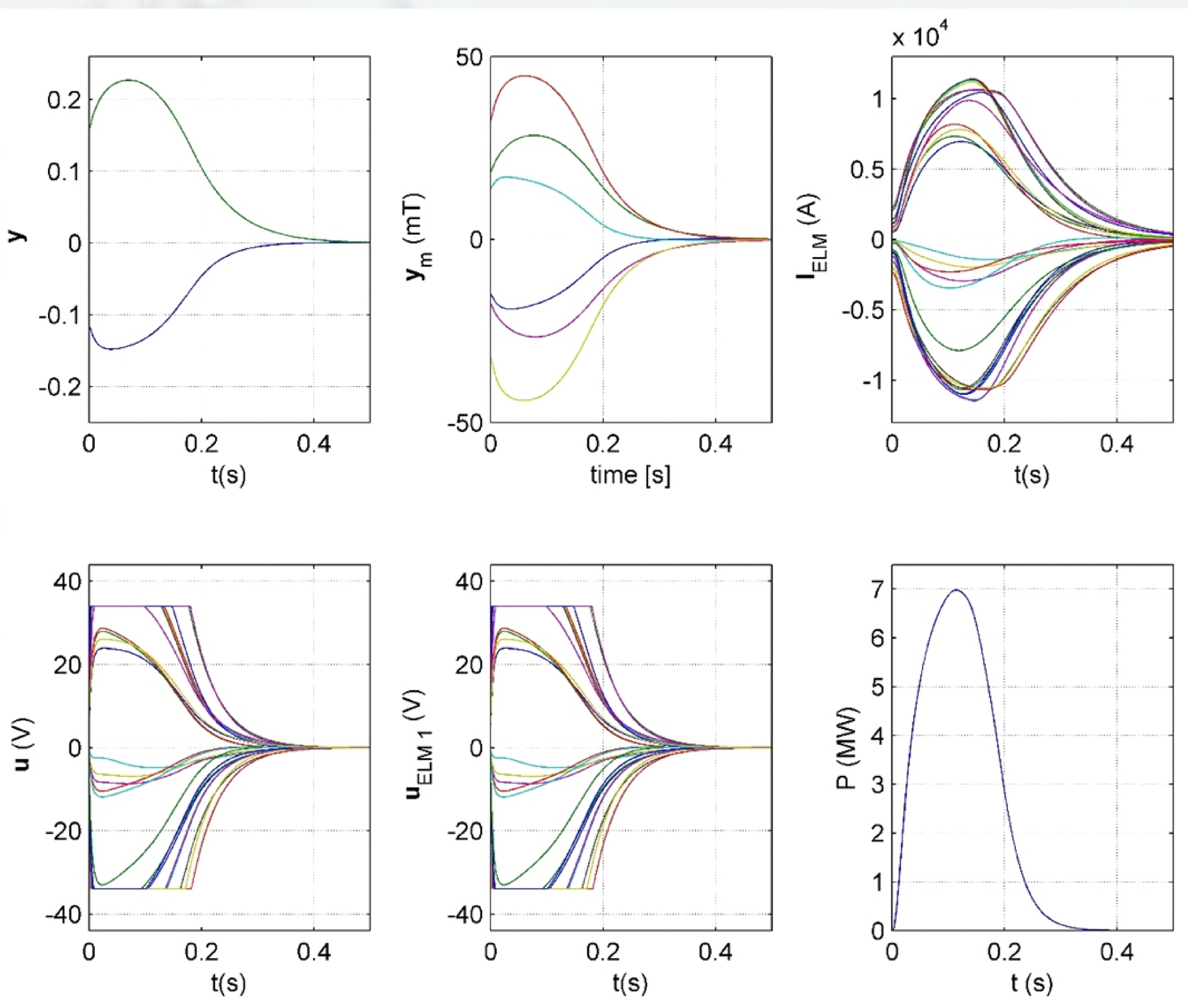
Outputs: reduced-dimensional (2) measurement vector **y**

50 dynamic states

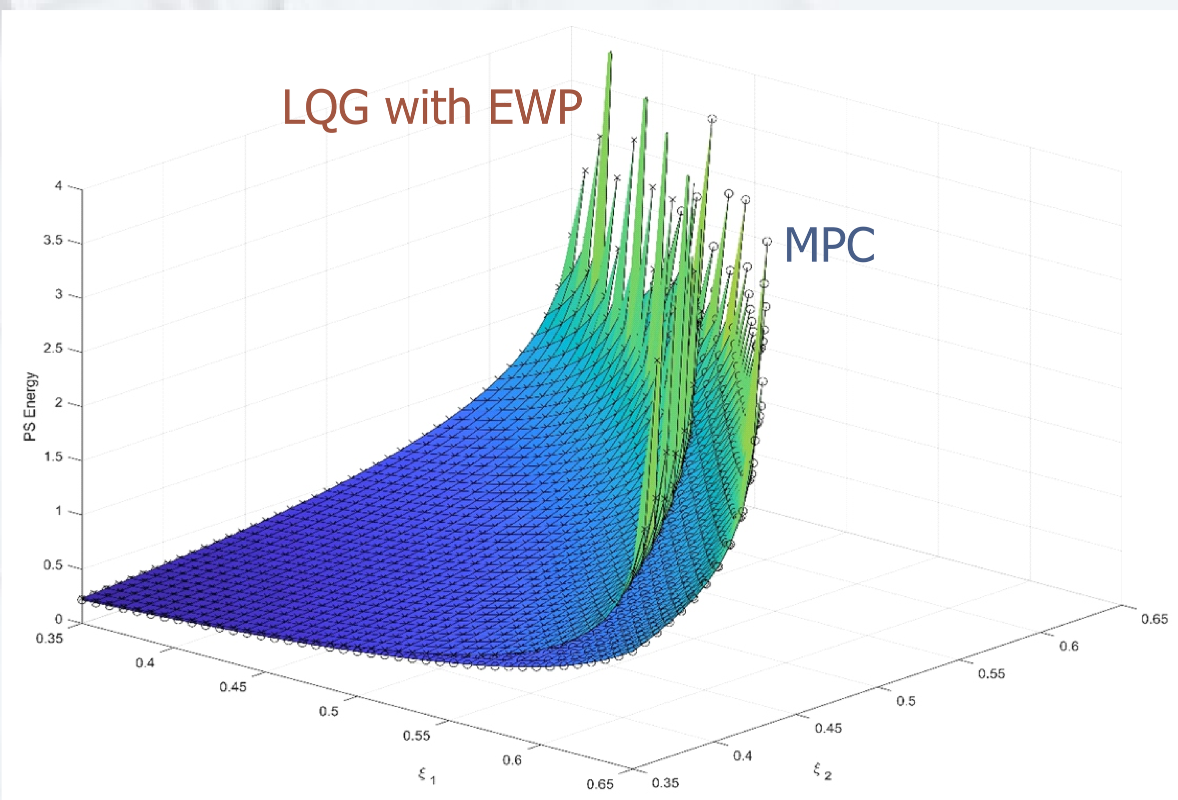
$$\mathbf{y} = \begin{bmatrix} y_A \\ y_B \end{bmatrix} = \mathbf{T}_{\mathrm{out}} \mathbf{y}_m, \qquad \mathbf{T}_{\mathrm{out}} = \begin{bmatrix} \cos\varphi_1 & \sin\varphi_1 \\ \cos\varphi_2 & \sin\varphi_2 \\ \vdots & \vdots \\ \cos\varphi_6 & \sin\varphi_6 \end{bmatrix}^{\dagger}$$



Nominal simulation, MPC



Nominal simulation, LQG with EWP



Stabilizable region of unstable modes

MPC control problem: quadratic program

Hard (actuator) constraints only, "condensed form"

$$\min_{\tilde{\mathbf{u}}} \frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{H}_c \tilde{\mathbf{u}} + \mathbf{f}_c^T \tilde{\mathbf{u}} + c_c$$

$$\text{subject to } \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \tilde{\mathbf{u}} \le \begin{bmatrix} \tilde{\mathbf{u}}_{\max} \\ \tilde{\mathbf{u}}_{\min} \end{bmatrix}$$

Primal FGM algorithm

Initialize: $\mathbf{v}^1 = \tilde{\mathbf{u}}^0 \in \mathbb{R}^{27N_u}$, sequence β^i, vector $\mathbf{f}_c$

for $i = 1$ to $i_{\max}$

$\mathbf{x}^i = \mathbf{v}^i - \mathbf{L}^{-1}(\mathbf{H}_c \mathbf{v}^i + \mathbf{f}_c)$ (gradient step)

$\tilde{\mathbf{u}}^i = \mathrm{prox}_{\psi}^{\mathbf{L}}(\mathbf{x}^i)$ (projection onto feasible set)

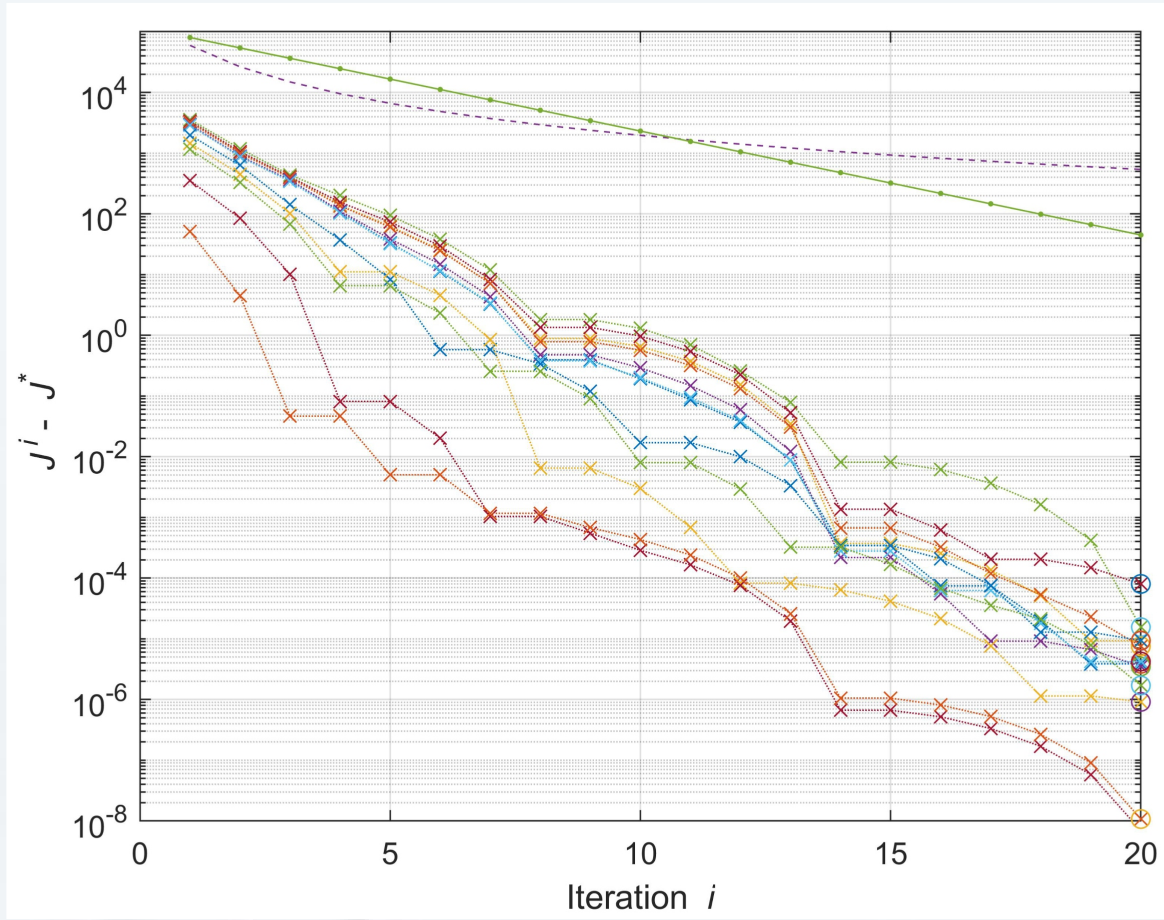
$\mathbf{v}^{i+1} = \tilde{\mathbf{u}}^i + \beta^i(\tilde{\mathbf{u}}^i - \tilde{\mathbf{u}}^{i-1})$ (acceleration)

if $(\mathbf{v}^i - \tilde{\mathbf{u}}^i)^T(\tilde{\mathbf{u}}^i - \tilde{\mathbf{u}}^{i-1}) > 0$ (adaptive restart)

$\mathbf{v}^{i+1} = \tilde{\mathbf{u}}^{i-1}, \quad \tilde{\mathbf{u}}^i = \tilde{\mathbf{u}}^{i-1}$

endif

endfor



Convergence of the cost function J: theoretical limits and selected simulation samples

Real-time implementability

C code generated by modified QPgen

Low-latency Linux environment of Ubuntu Linux 14.04, 4.4.9-rt17 SMP PREEMPT RT kernel on a standard desktop computer based on the Intel Core i7-2600K CPU

Double precision: on average, 26 µs for the prologue and epilogue, and 2.6 µs for each iteration of the main loop.

Single precision: about 20% less

Maximum computation time: ca 20% higher than average

Desired worst-case MSE below 10^{-4} reached at maximum computation time 0.08 ms.

$$MSE(\tilde{\mathbf{u}}^i, \tilde{\mathbf{u}}^*) = \sqrt{\frac{1}{27N_u} \sum_{k=1}^{27N_u} \left(\frac{\tilde{u}_k^i - \tilde{u}_k^*}{\tilde{u}_{\max,k} - \tilde{u}_{\min,k}} \right)^2}$$

Conclusions

Infinite-horizon MPC for the RWM control problem considering actuator constraints using a solver such as the primal FGM is viable, with an acceptable computation time 0.1 ms.

The MPC controller increases the stabilizable region of the initial values of the unstable modes and use less power than the LQG controller.

Simulations show robustness to a wide range of (lower) instability growth rates and frequencies and to noise

Further acceleration is presumably possible by using fixed-point arithmetics and FPGA implementation.

Department of Systems and Control, Jožef Stefan Institute · CREATE PPCC team · COSYLAB · Slovenian Research Agency

P2-0001