Plasma Current, Position and Shape Control in Tokamaks
Hands-on Session

16-18 January 2012 - Jožef Stefan Institute

Gianmaria De Tommasi[1]
[1]CREATE, Università di Napoli Federico II
detommas@unina.it

Outline

Hands-on Session

Plasma Shape Control Design for the JET tokamak
- PF Currents Controller
- Plasma Current Controller
- Plasma Shape Control

Plasma Vertical Stabilization for the ITER tokamak
- SISO loop
- Additional loop on VS1

The XSC Tools
- Open Issues

Hands-on Session

- This hand-on session focuses on:
  1. The design of a **plasma current and shape control** for the **JET tokamak**:
     1.1 PF Current Control
     1.2 Plasma Current Control
     1.3 Plasma Shape Control (in an *XSC-flavor*)
     - In this case we will assume the plasma is vertically stabilized on a faster timescale
  2. The design of a **vertical stabilization system** for the **ITER tokamak**:
     2.1 SISO controller which makes use of the in-vessel coils for vertical stabilization
     2.2 Additional loop to reduce the current in the in-vessel coils when a Vertical Displacement Event (VDE) is considered
  3. Overview of the **XSC Tools**
- **We will work in the Matlab/Simulink environment**

Plasma linearized model of JET tokamak

The linearized plasma model used for plasma current and shape control is

$$\begin{aligned}\delta\dot{x} &= A\delta x + B\delta u \\ \delta y &= C\delta x\end{aligned}$$

where the state and input vectors are given by

$$\delta x = \begin{pmatrix} \delta I_{PF} \\ \delta I_p \end{pmatrix} \quad \text{and} \quad \delta u = \begin{pmatrix} \delta V_{PF} \\ \delta V_p \end{pmatrix}$$

- δI_{PF}, δV_{PF} are the PF current and voltage variations
- δI_p, δV_p are the plasma current and loop-voltage variations

Model Outputs

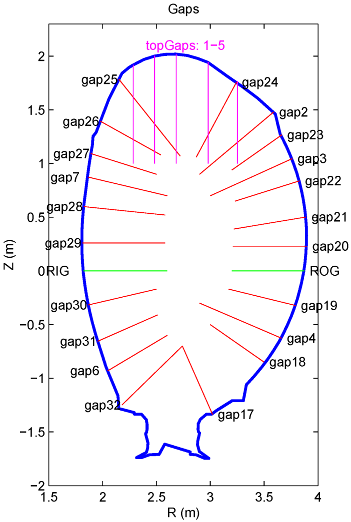
The output vector is equal to

$$\delta y = \begin{pmatrix} \delta I_{PF} \\ \delta I_p \\ \delta g \end{pmatrix}$$

where δg holds the plasma shape descriptors, i.e.

- gaps
- strike-points
- x-points

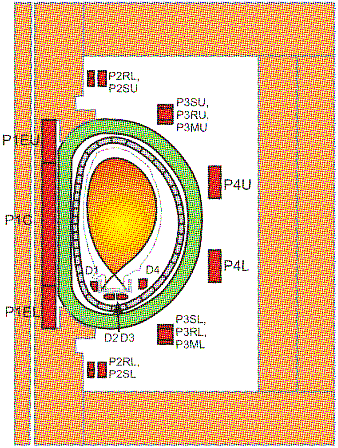
Plasma shape descriptors at JET



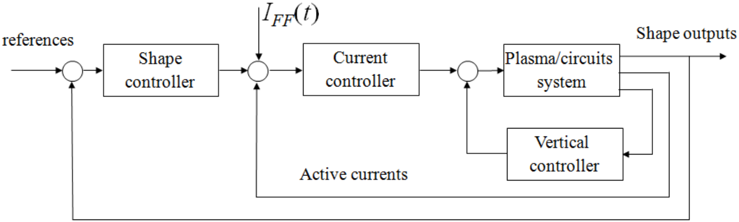
The JET PF system

The 9 currents in the PF coils are

- I_{P1} - current in the $P1$ circuit
- I_{P4T} - current in the $P4$ circuit
- I_{IMB} - imbalance current in the $P4$ circuit
- I_{PFX} - current in the FX circuit
- I_{SHP} - current in the shaping circuit
- I_{D1} , I_{D2} , I_{D3} , I_{D4} - currents in the divertor coils



Reference Control Scheme



- SC generates current references
- A PF currents controller must be designed

JET Current Controller

Plasmaless model

$$\mathbf{V}_{PF} = \begin{bmatrix} L_1 & M_{12} & \dots & M_{1N} \\ M_{12} & L_2 & \dots & M_{2N} \\ \dots & \dots & \dots & \dots \\ M_{1N} & M_{2N} & \dots & L_N \end{bmatrix} \frac{\mathrm{d}\mathbf{I}_{PF}}{\mathrm{dt}} + \begin{bmatrix} R_1 & 0 & \dots & 0 \\ 0 & R_2 & 0 & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & R_N \end{bmatrix} \mathbf{I}_{PF}$$

Resistive compensation

$$\mathbf{V}_{PF_{ref}} = \hat{\mathbf{R}}\mathbf{I}_{PF} + \mathbf{K}(\mathbf{Y}_{ref} - \mathbf{Y})$$

Static relationship between PF coils current and controlled variables

$$\mathbf{Y} = \mathbf{T}\mathbf{I}_{PF}$$

Control Matrix

$$\mathbf{K} = \hat{\mathbf{M}}\mathbf{T}^{-1}\Lambda^{-1} \text{ with } \Lambda \text{ diagonal matrix}$$

JET Shape Controller

Closed-loop system

$$\mathbf{M}\mathbf{T}^{-1}\dot{\mathbf{Y}} + \mathbf{R}\mathbf{I}_{PF} = \mathbf{M}\mathbf{T}^{-1}\Lambda^{-1}(\mathbf{Y}_{ref} - \mathbf{Y}) + \mathbf{R}\mathbf{I}_{PF} \Rightarrow$$
$$\Rightarrow \ \dot{\mathbf{Y}} = \Lambda^{-1}(\mathbf{Y}_{ref} - \mathbf{Y})$$

By a proper choice of the **T** matrix it is possible to achieve:

- current control mode
- plasma current control mode
- gap control mode

Plasma current model

A simplified model of the plasma current circuit is considered

- plasma resistance is neglected
- only the mutual inductance with the *P1* circuit is retained

The following broadly valid linear model can be derived

$$\dot{I}_p(t) = -c\dot{I}_{P1}(t)\,, \quad \text{with } c > 0\,.$$

eXtreme Shape Controller (XSC)

- The *eXtreme Shape Controller* (XSC) controls the whole plasma shape, specified as a set of 32 geometrical descriptors, calculating the PF coil current references.
- Let $\mathbf{I}_{PF_N}(t)$ be the PF currents normalized to the equilibrium plasma current, it is

  $$\delta\mathbf{g}(t) = \mathbf{C}\ \delta\mathbf{I}_{PF_N}(t).$$

  It follows that the plasma boundary descriptors have the same dynamic response of the PF currents.
- The XSC design has been based on the **C** matrix. Since the number of independent control variables is less than the number of outputs to regulate, it is not possible to track a generic set of references with zero steady-state error.

  $$\delta\mathbf{I}_{PF_{N_{req}}} = \mathbf{C}^{\dagger}\delta\mathbf{g}_{error}$$

eXtreme Shape Controller (XSC)

- The XSC has then been implemented introducing weight matrices both for the geometrical descriptors and for the PF coil currents.
- The determination of the controller gains is based on the Singular Value Decomposition (SVD) of the following weighted output matrix:

  $$\widetilde{\mathbf{C}} = \widetilde{\mathbf{Q}}\,\mathbf{C}\,\widetilde{\mathbf{R}}^{-1} = \widetilde{\mathbf{U}}\,\widetilde{\mathbf{S}}\,\widetilde{\mathbf{V}}^{T},$$

  where $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{R}}$ are two diagonal matrices.
- The XSC minimizes the cost function

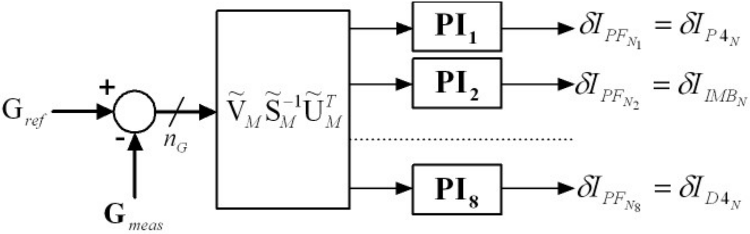
  $$\widetilde{J}_1 = \lim_{t\to+\infty}(\delta\mathbf{g}_{ref} - \delta\mathbf{g}(t))^T\widetilde{\mathbf{Q}}^T\widetilde{\mathbf{Q}}(\delta\mathbf{g}_{ref} - \delta\mathbf{g}(t)),$$

  using $\bar{n} < 8$ degrees of freedom, while the remaining $8 - \bar{n}$ degrees of freedom are exploited to minimize

  $$\widetilde{J}_2 = \lim_{t\to+\infty}\delta\mathbf{I}_{PF_N}(t)^T\widetilde{\mathbf{R}}^T\widetilde{\mathbf{R}}\delta\mathbf{I}_{PF_N}(t).$$

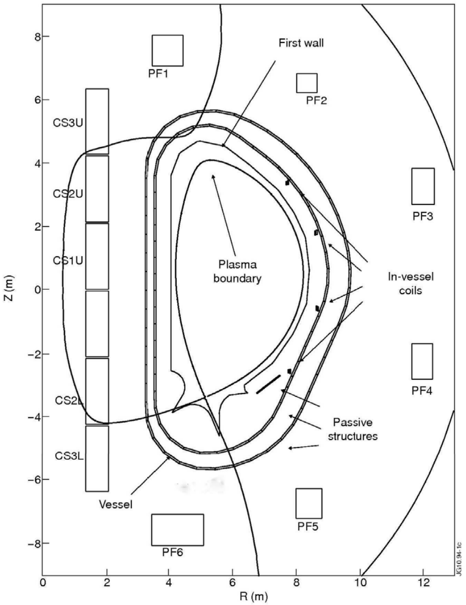
  (it contributes to avoid PF current saturations)

XSC - Gap controller



The ITER PF system

- The CS and PF coils are superconductive
- The CS and PF coils are used to control plasma current and shape
- PF2-3-4-5 can be used to generate a up-down symmetric radial filed for vertical stabilization (the VS1 circuit)
- The in-vessel coils are used for vertical stabilization
- **The in-vessel coils are connected in anti-series, i.e. the current in the upper coils flows in opposite way wrt the current in the lower coils ⇒ a single voltage and current are considered for the in-vessel circuit.**



Plasma linearized model of ITER tokamak

The linearized plasma model used for the design of the ITER vertical stabilization is

$$\begin{aligned} \delta\dot{x} &= A\delta x + B\delta u \\ \delta y &= C\delta x \end{aligned}$$

where the output vector is given by

$$\delta y = \begin{pmatrix} \delta I_{CS\,,PF} \\ \delta I_{iv} \\ \delta I_p \\ z_c \\ \dot{z}_c \end{pmatrix},$$

where z_c and $\dot{z}_c$ are the plasma vertical position and velocity, respectively.

A simple vertical stabilization for the ITER tokamak

In principle, we would like to use a proportional-derivative (PD) regulator

$$U_{iv}(s) = k_1 \cdot Z_c(s) + k_2 \cdot sZ_c(s) = (k_1 + k_2 s)\, Z_c(s)\,.$$

Since the current $I_{iv}(t)$ in the in-vessel coils is proportional to the plasma vertical posizion $z_c(t)$, it is

$$U_{iv}(s) = k_1 \cdot I_{iv}(s) + k_2 \cdot sZ_c(s)\,.$$

VS for the ITER tokamak in a “JET style”

- We will use a controller structure similar to the one adopted at the JET tokamak

$$U_{iv}(s) = K_1(s) \cdot \left(sZ_c(s) + k_2 \cdot I_{iv}(s)\right) ,$$

  where the transfer function $K_1(s)$ and the gain k_2 are adjusted (adapted) during the different phases of the discharge, in order to optimize the performance
- The unstable mode (*growth rate*) varies significantly and sometimes unpredictably during a discharge

M. Lennholm et al.
Plasma control at JET
Fusion Engineering and Design, vol. 48, no. 1–2, pp. 37–45, Aug. 2000

F. Sartori, G. De Tommasi and F. Piccolo
The Joint European Torus - Plasma position and shape control in the world’s largest tokamak
IEEE Control Systems Magazine, vol. 26, no. 2, pp. 64–78, Apr. 2006

Design of the stabilization loop

1. choose a value for k_2 and $K_1(s)$
2. check the closed-loop stability via the Nyquist criterion
3. **if** the closed-loop system is stable **then**
    3.1 check the stability margins (e.g. by using the Nichols plot)
    3.2 **if** the stability margins are satisfactory **then** **go to** step **5**
    3.3 **else go to** step **1**
4. **else go to** step **1**
5. **GOAL!**

Reminiscences of control fundamentals...

Nyquist Criterion

Consider a loop frequency response $L(j\omega)$ and let

- P be the number of poles of $L(s)$ with strictly positive real part
- Z be the number of zeros of $L(s)$ with strictly positive real part

The Nyquist plot of $L(j\omega)$ makes a number of encirclements N (**clockwise**) about the point $(-1\,, j0)$ equal to

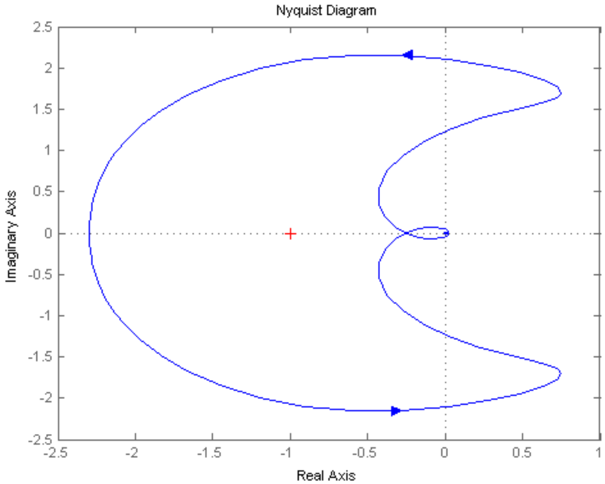
$$N = Z - P\,.$$

It turns out that the closed-loop system is asymptotically stable **if and only if** the Nyquist plot of $L(j\omega)$ encircle (**counter clockwise**) the point $(-1\,, j0)$ a number of times equal to P.

The criterion is valid if the Nyquist plot of $L(j\omega)$ do not intersect the point $(-1\,, j0)$.

Example of Nyquist plot for the ITER plant

Example of Nyquist plot obtained for the ITER plant after having selected k_2 and considering $sZ_c(s) + k_2 \cdot I_{iv}(s)$ as input.



Vertical Displacement Event (VDE)

- The VDE is an uncontrolled growth of the plasma unstable vertical mode
- These uncontrolled growths can occur for different reasons, such as:
  - fast disturbances acting on a time scale which is outside the control system bandwidth
  - delays in the control loop
  - wrong control action due to measurement noise, when plasma velocity is almost zero
- From the vertical stabilization point of view a VDE is equivalent to a sudden and almost instantaneous change in plasma position
- VDE can be modeled as instantaneous change of the state vector
- The response of the plant to a VDE can be studied considering a given initial state $\mathbf{x}(0) = \mathbf{x}_{VDE}$.

Constraint on the in-vessel coils

- The in-vessel coils are **copper coils**, hence a constraint on the current must be considered
- Due to joule losses, there is a thermal constraint which limits the rms value of $I_{iv}(t)$ in presence of a VDE
- In particular

$$\sqrt{\frac{1}{T}\int_0^T I_{iv}^2(t)dt} < L\,.$$

For example, $T = 1\ s$ and $L = 30\ kA$ when a VDE of $20\ cm$ is considered.

Design of the additional loop on VS1

- In order to reduce the joule losses it is possible to use the external superconductive circuit $VS1$
- We use a simple *proportional controller*

$$u_{VS1}(t) = k_3 \cdot I_{iv}(t)\,.$$

- It should be noticed that, with this *simple* additional loop the system is not able to control to zero the vertical position ⇒ usually the shape controller will take care of it!
- **Let's try to design a simple vertical position control loop!**

Robustness issues

- It is possible to design a robust set of parameters $K_1(s)$, k_2 and k_3, in order to vertically control the plasma during the whole ITER discharge (at least in simulation!)

G. Ambrosino et al.
Robust vertical control of ITER plasmas via static output feedback
IEEE Multi-Conference on Systems and Control (MSC'11), Denver, Colorado, Sep. 2011, pp. 276–281.

- In general, such a robust controller cannot meet the requirements when *extreme* scenarios are considered
- In order to meet the requirement the controller parameters need to be adapted
- The proposed controller has a simple structure which permits to envisage effective adaptive algorithms

Material

All the material (slides + source code) is also available from

http://wpage.unina.it/detommas/ijs.html

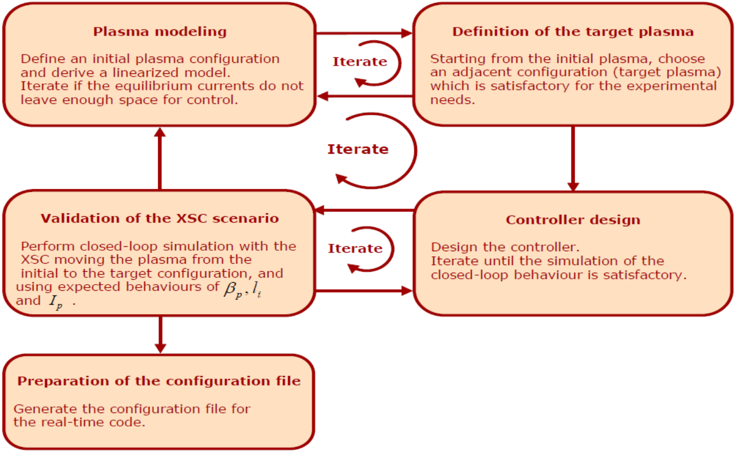
The eXtreme Shape Controller (XSC)

- The XSC has been developed to control the whole plasma boundary, and it can be used to obtain high performance plasmas
- The XSC is currently used during experimental campaigns to control ITER-like plasmas
- The use of the XSC is not an easy thing, since it consists in a number of not automated steps, which required the involvement of several experts for both the plasma modeling and controller design.
- Furthermore we would like to use the output of the controller design phase, usually carried out using Matlab/Simulink, straightforwardly as input of the C++ code, which implements the XSC control algorithm on the plant system.

The XSC Tools

- The XSC Tools have been developed to automate both the controller design and validation phases.
- An additional effort has been made during the definition of the XSC Tools in order to make them machine independent

XSC design procedure



Plasma modeling

Define an initial plasma configuration and derive a linearized model. Iterate if the equilibrium currents do not leave enough space for control.

Definition of the target plasma

Starting from the initial plasma, choose an adjacent configuration (target plasma) which is satisfactory for the experimental needs.

Controller design

Design the controller. Iterate until the simulation of the closed-loop behaviour is satisfactory.

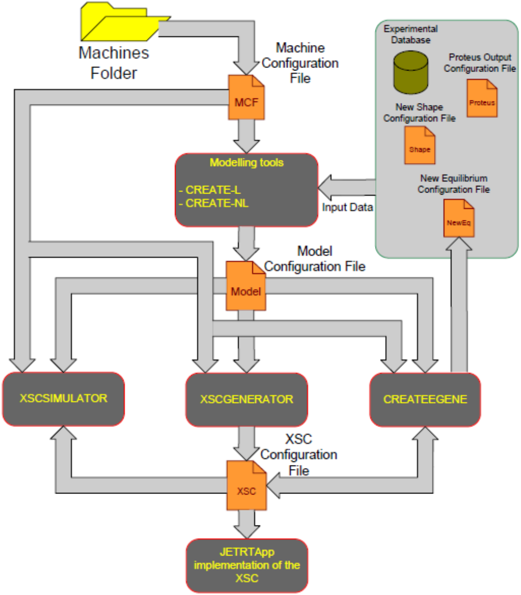
Validation of the XSC scenario

Perform closed-loop simulation with the XSC moving the plasma from the initial to the target configuration, and using expected behaviours of β_p, l_i and I_p .

Preparation of the configuration file

Generate the configuration file for the real-time code.

Overview of the XSC Tools



What is missing ?

Wishes

- Avoid implementation *by-hands* → automatic real-time code generation
- Allow to perform closed–loop validation with the real-time code

Limitations

- **JETRT** didn't provide a **real** separation between the algorithmic part of a real-time application from the plant-interface software
- **JETRT** didn't allow the user to plug in a plant model in order to perform closed–loop validation of the real-time system

Outline
Hands-on Session
Plasma Shape Control Design
PF Current Controller
Plasma Current Controller
Shape Controller
Plasma Vertical Stabilization Design
SISO loop
Additional loop on VS1
XSC Tools
Open Issues

The MARTe framework

- The development of the JETRT framework stopped since we moved to MARTe, which provides a real separation between the algorithmic part of a real-time application from the plant-interface software
- The design and validation phases were still carried out in Matlab/Simulink environment
- The control algorithm was still coded “by-hands” as a Generic Application Module (GAM) in MARTe
- Validation of the real-time code were performed by closed–loop simulations *GAMifying* the CREATE plasma model. Already done for:
  - The new VS system (PCU project)
  - The Current Limit Avoidance System

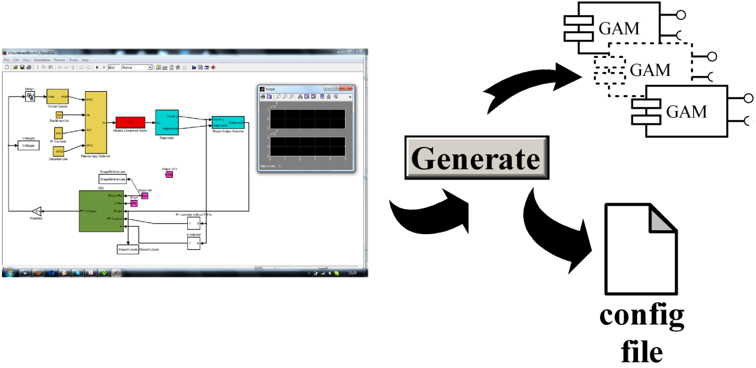
T. Bellizio et al.
A MARTe based simulator for the JET Vertical Stabilization system
Fusion Engineering and Design, vol. 86, no. 6–8, pp. 1026–1029, Oct. 2011.

Something is still pending!

- Within MARTe we still need automatic code generation tools!
- We would like to exploits the Mathworks Real-Time Workshop to automatically generate GAMs starting from Simulink schemes

We would like to make it easy!

We would like to just press a button and generate the code for real-time



From Simulink to GAM

Matlab/Simulink environment is a standard *de facto* for the design of control algorithms

Possible activity

Automatic GAM generation starting from Simulink can be done:

- Exploiting the Real-Time Workshop Target Language Compiler (TLC) to generate a new *target* for the MARTe GAMs, which is HW independent

  C. Centioli et al.
  Using Real Time Workshop for rapid and reliable control implementation in the Frascati Tokamak Upgrade Feedback Control System running under RTAI-GNU/Linux
  Fusion Engineering and Design, vol. 74, no. 1–4, pp. 593–597, Nov. 2005.

- Generating a *general purpose* C/C++ code with the Real-Time Workshop and then wrap it into a GAM (a similar solution is currently adopted by RFX)