Magnetic modeling and control of tokamaks, Part VI: Free boundary equilibrium (shape) control

Adriano Mele

Ecole Polytechnique Fédérale de Lausanne (EPFL),
Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland

EPFL Doctoral School Course PHYS-748, February 2025

EPFL

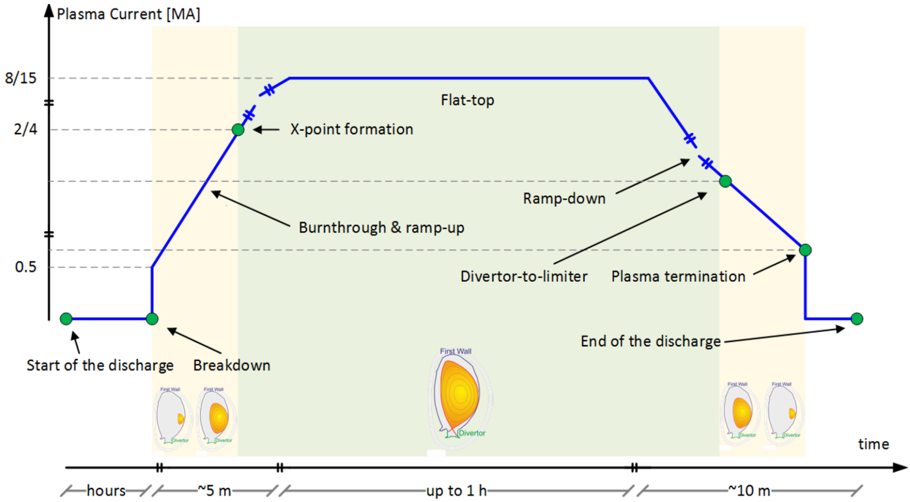
Outline I

1. Introduction
2. Classic shape control
   - Shape control at EAST
   - Plant decoupling through SVD
3. Shape control at TCV
4. Model Predictive Control for plasma shape

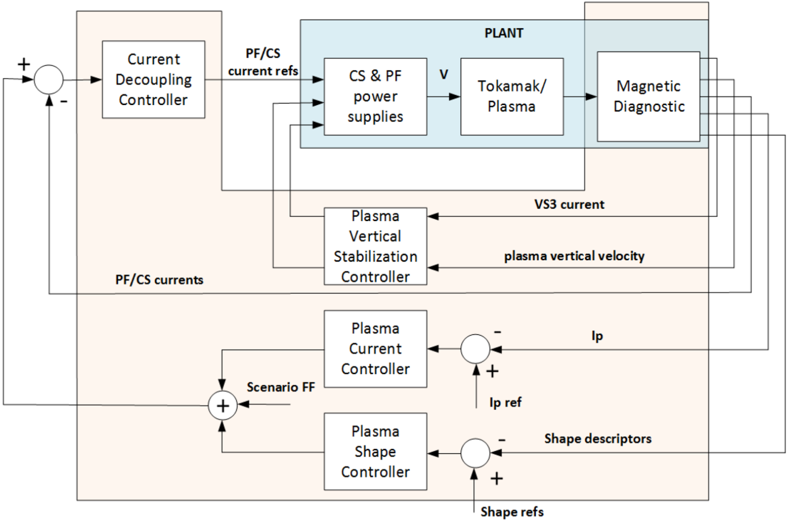
Section 1

Introduction

A typical discharge



Magnetic control: the big picture



Plasma shape control

- In modern tokamaks, accurate control of the plasma shape is desirable for several reasons
  - Keep a desired plasma-wall clearance
  - Optimize the vacuum chamber occupation
  - Limit the plasma growth rate
  - Achieve specific configurations to meet **scientific objectives** (e.g. negative triangularity) or for **technical reasons** (e.g. double-null plasmas or strike point sweeping for power exhaust handling)
  - etc.

M. Ariola, A. Pironti
Magnetic control of tokamak plasmas
Springer, 2008

R. Albanese et al.
Design, implementation and test of the XSC extreme shape controller in JET
Fus. Eng. Des., 2005

Plasma shape control

- Early approaches focused on control of global shape parameters
  - **Elongation**

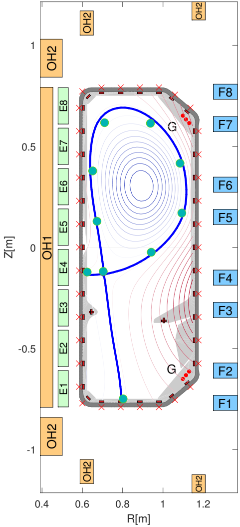
$$\kappa = \frac{Z_{max} - Z_{min}}{R_{max} - R_{min}}$$

  - **Triangularity**

$$\delta = \frac{R_{max} + R_{min} - (R_{up} + R_{low})}{R_{max} - R_{min}}$$

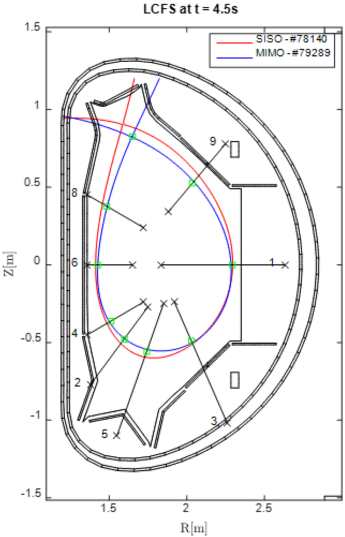
  - higher order moments (squareness, etc.)
- Modern controllers usually adopt two main approaches:
  - **Isoflux control**: the differences between poloidal flux values at various control points on the boundary are controlled to zero
  - **Gap control**: the distance between the plasma LCFS and the first wall is controlled to a desired value

Isoflux control



- Plasma boundary can be prescribed by requiring that all points on the boundary have the **same flux** ψ
- This forces an iso-ψ line to pass through the points (hence the name)
- $\times$**-point** locations can be prescribed by requiring that $|\nabla\psi| = 0$
- This is equivalent to requiring $|B| = 0$

Gap control



- Define a set of **plasma-wall gaps** (points along wall + orientation)
- This has an easy physical interpretation, but provides less flexibility
- Gap values must be held close to a given reference
- *Note: EAST actually uses a mix of the two approaches*

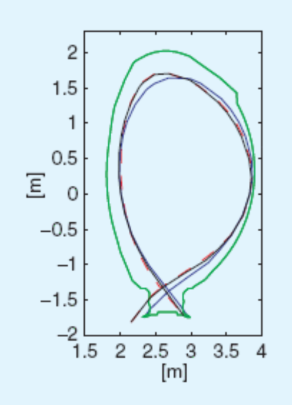
Section 2

Classic shape control

The early days of plasma shape control - JET

The **JET** tokamak has been a pioneer in the field of plasma shape control

- The **XSC** (eXtreme Shape Controller) was developed in the early 2000s
- It was a **gap** controller based on a **state-space model** of the plasma shape (from CREATE-L)

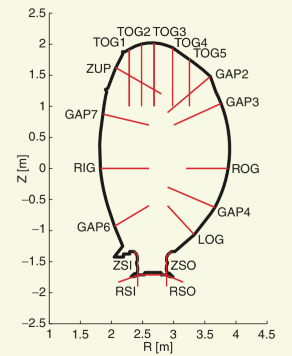


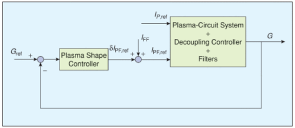
M. Ariola, A. Pironti
Plasma shape control for the JET tokamak: an optimal output regulation approach.
IEEE Control Systems Magazine, 2005

R. Albanese, F. Villone
The linearized CREATE-L plasma response model for the control of current, position and shape in tokamaks
Nuclear Fusion, 1998

Shape control at JET

- The idea is to take advantage the machine's **Vertical Stabilization** and **Coil Current control** systems
- The state-space model provides a (static) relation from **coil currents** to plasma-wall **gaps**
- The controller finds the optimal currents to keep the gaps at the desired values
- **MIMO control problem**:
  coil currents → gaps





MIMO shape control

- The **output equation** of the linearized model gives a *linear relation* between the coil currents ($\delta x = \delta I_{PF}$) and the outputs of interest (δy, in this case gap variations)

$$\delta y = C\delta x$$

- We are **neglecting the eddy currents and the internal profile variations** - we can consider this as a steady-state condition
- When the dimension of controlled outputs (m) is larger than the dimension of available currents (n), we are basically left with a **linear regression problem**

Shape control as a linear regression problem

- This can also be written as a **least-squares problem**

$$\min_{\delta x} ||C\delta x - \delta y^*||_2^2$$

... and you have already seen the solution! (in Part II)

- If C has **full rank**, we can solve the problem by choosing

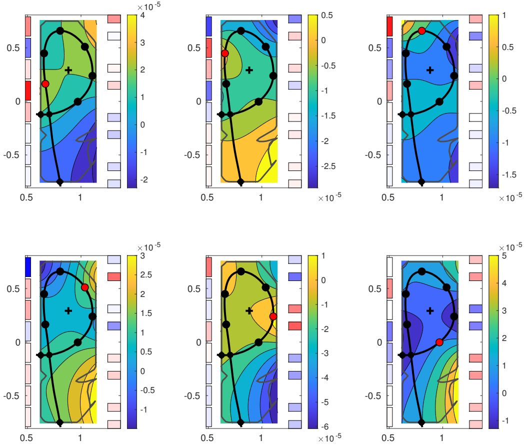
$$\nabla_{\delta x} J = 0 \implies \delta x = (C^T C)^{-1} C^T \delta y^* = C^\dagger \delta y^* ,$$

where $C^\dagger$ is the **Moore-Penrose (left) pseudoinverse** of C

- The name is due to the property $C^\dagger C = \mathbb{I}$

A (naive) example

- Directions for TCV shot #78071, linearized model from `fgess`, 6 contour points



An old debate: isoflux or gaps?

- Fluxes are "machine friendly" and **more flexible**...
- ...but gaps are "human friendly" and more **natural**
- Translate **isoflux errors** into **physical displacements** of LCFS control points.
- Converts **poloidal flux** and **magnetic field** variations into **shape modifications**.

A. Tenaglia, F. Pesamosca, F. Felici, D. Carnevale, S. Coda, A. Mele, A. Merle
An interpretable isoflux-based observer for plasma shape control errors in tokamaks
Fusion Engineering and Design, 2024

Plasma Shape Observer (courtesy of A. Tenaglia)

- Define the vector containing the **nominal values** computed at the control point locations and the same quantities **reconstructed** after the deformation of the starting equilibrium

$$y_0 = \begin{bmatrix} \psi_0 \\ B_{R,0} \\ B_{Z,0} \end{bmatrix}, \qquad y = \frac{I_{p,0}}{I_p} \begin{bmatrix} \psi \\ B_R \\ B_Z \end{bmatrix}.$$

- Design a **mask** to select the needed values

$$\xi = T_{\xi y}\,(y - y_0)\,,$$

- Using **linearized relations** to build a matrix such that

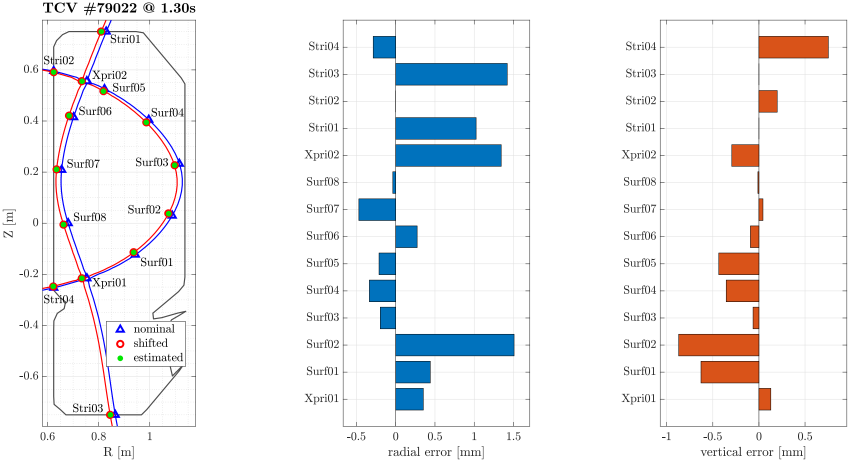
$$\xi = T_{\xi d}\, d\,,$$

- Combining (17) and (17), the **shape observer** is defined by

$$d = T_{\xi d}^{-1}\,\xi = T_{\xi d}^{-1}\, T_{\xi y}\,(y - y_0)\,,$$

appendix

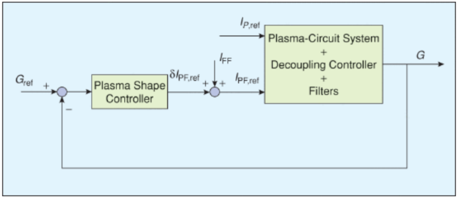
Numerical Results (courtesy of A. Tenaglia)



TCV shot #79022, (fictitious) negative radial rigid shift of 3 cm

Transient response

- The shape controller provides variations to the **PF coils current references**, that are fed to the PFC controller
- With this technique, we can **decouple** the shape control problem: each **column** of $C^{\dagger}$ represents a **current pattern** that is associated with one of the outputs
- If the PF current dynamics have been **equalized**, we can then tune a dynamic controller on the **nominal, SISO PF dynamics** to achieve the desired transient response
  - most common solution: **PID** controllers

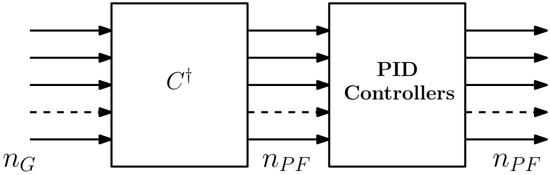


Additional remarks

- A **weighted** C matrix can also be used to promote accuracy on some of the shape descriptors or the usage of some of the actuators (see later)

$$\tilde{C} = WCQ$$

- The resulting controller block looks like this



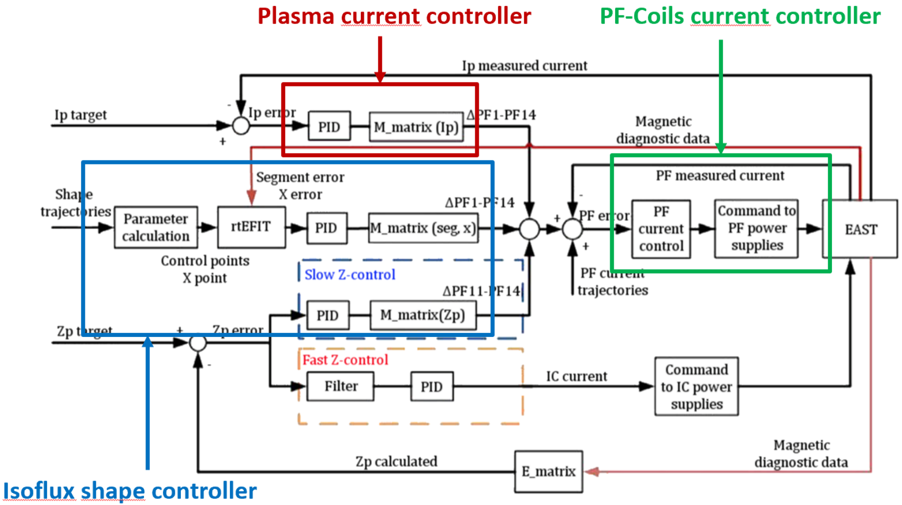
Subsection 1

Shape control at EAST

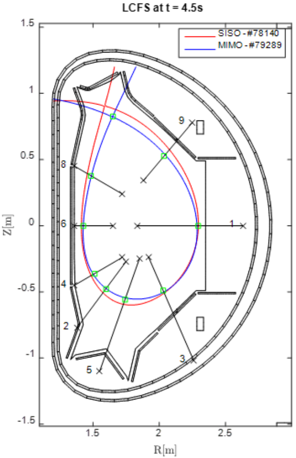
Additional remarks

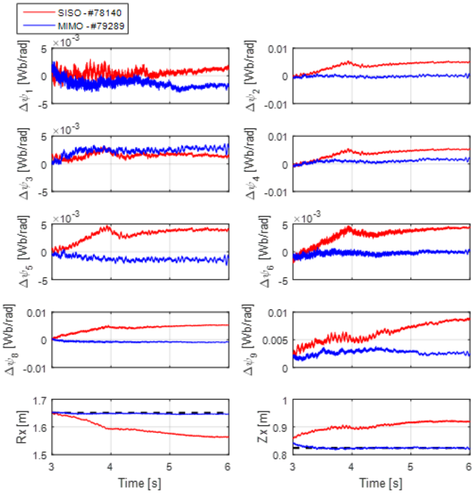
- We assumed that the plant is already **stabilized**
  - Ideally with dedicated (in-vessel) coils
  - Stabilization acts on fast time-scales, shape control usually has a **lower bandwidth**
- Usually shape controller is seen as a more sophisticated **alternative** to position control
  - ..but this is not always the case
- If all PIDs are the same in the previous scheme, they can be moved *upstream* of the $C^{\dagger}$ block: this was convenient for instance when designing a shape controller for EAST
  - PID + gain matrix is a common architecture, and was the one installed in the EAST PCS

Example: MIMO shape control at EAST

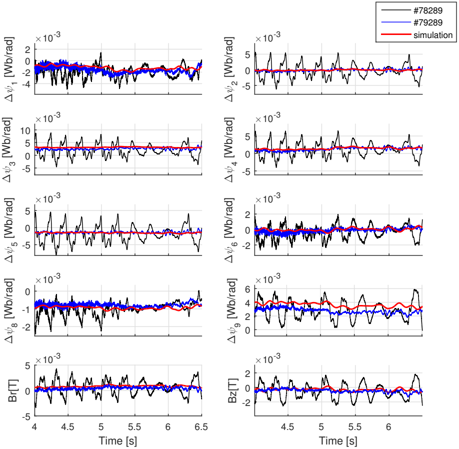


MIMO shape control at EAST





MIMO shape control at EAST



The controller in the previous slide was designed using data from a similar pulse

- black: EAST pulse #78289
  Controller taken from another shot with different I_p and plasma configuration
- red: simulation
  The controller is tuned to reduce oscillations
- blue: EAST pulse #79289
  (experimental results)

A. Mele et al.
MIMO shape control at the EAST tokamak: simulations and experiments
SOFT, 2018

Subsection 2

Plant decoupling through SVD

A second look at pseudoinversion

- The design we have seen relies on the assumption that $C^T C$ is invertible
- With $n = m$ we can control *exactly* n independent linear combinations of shape descriptors
- With $n < m$ the controller minimizes the **steady-state mean square error**
- . . .**is this the best choice? What if C does *not* have full rank?**

Singular Value Decomposition

- The **svd** of $C \in \mathbb{R}^{m\times n}$ is a factorization in the form

$$C = U\Sigma V^*$$

where

  - $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are *unitary*[1][2] matrices (a.k.a. *rotations*), whose columns are the **generalized (left/right) eigenvectors** of C
  - $\Sigma \in \mathbb{R}^{m\times n}$ is a *rectangular diagonal* matrix, whose non-negative entries σ_i are the **singular values** of C

- Let's look at this from a geometric perspective

[1] i.e. $A^{-1} = A^*$. This also implies $\det A = 1$.

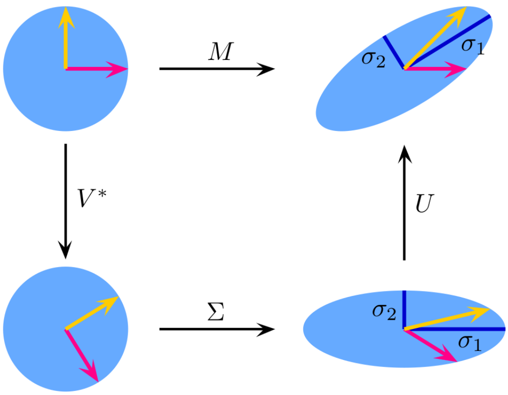
[2] The equivalent for real matrices is a **orthogonal** matrix, i.e. $A^{-1} = A^T$. For simplicity, we will always refer to the real case.

Low-rank approximation

- The cost function for our least-squares problem is

$$J = (\delta y^* - C\delta x)^T(\delta y^* - C\delta x) \sim (\delta x^* - \delta x)C^T C(\delta x^* - \delta x)$$

- The action of the positive semi-definite, symmetric matrix $C^T C$ on the error vector $\delta\tilde{x} = (\delta x^* - \delta x)$ can be visualized looking at how it transforms a **sphere** (i.e. every possible $\delta\tilde{x}$ with unit-norm)



$$M = U \cdot \Sigma \cdot V^*$$

Singular Value Decomposition

- The svd generalizes the **eigendecomposition** of a square normal[3] matrix with an orthonormal eigenbasis to any $m \times n$ matrix
- In our case, for instance

$$C^T C = (V\Sigma^T U^T)(U\Sigma V^T)$$

and since U, V are orthogonal, $U^{-1} = U^T$, $V^{-1} = V^T$ and

$$\underbrace{(C^T C)}_{\substack{\text{Hermitian}\\ \text{Positive}\\ \text{semi-definite}}} \quad V = V \quad \underbrace{(\Sigma^T \Sigma)}_{\substack{\text{Diagonal}\\ \text{Non-negative}\\ \text{entries}}}$$

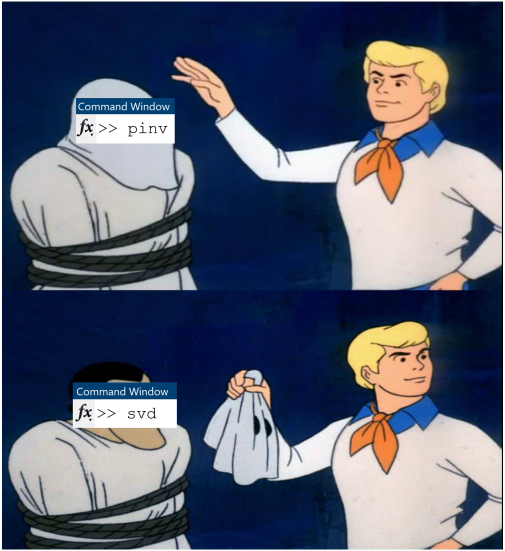
- similarly, $(CC^T)U = U(\Sigma\Sigma^T)$

[3] i.e. such that it commutes with its conjugate-transpose: $A^* A = AA^*$

Singular Value Decomposition

- To summarize:
  - the columns of V are the eigenvectors of $C^T C$
  - the columns of U are the eigenvectors of CC^T
  - the non-zero entries of Σ are the square roots of the eigenvalues of CC^T or $C^T C$
- **Why is this useful?**

Pseudoinversion through svd



Pseudoinversion through svd

- The pseudoinverse of a matrix can be computed via its **svd**

$$C = U\Sigma V^T \implies C^\dagger = V\Sigma^\dagger U^T$$

- $\Sigma^\dagger$ is obtained by replacing every **non-zero** diagonal entry by its reciprocal and transposing the result
- This procedure works also when C is **rank-deficient**

Low-rank approximation

- Another advantage of using the svd is that it can be used to compute a **truncated** version of $C^\dagger$ (so-called **low-rank** approximation)
- The cost function for the least-squares problem is

$$J = (\delta y^* - C\delta x)^T(\delta y^* - C\delta x) \sim (\delta x^* - \delta x)C^T C(\delta x^* - \delta x)$$

- The action of the positive semi-definite, symmetric matrix $C^T C$ on the error vector $\delta\tilde{x} = (\delta x^* - \delta x)$ can be visualized looking at how it transforms a **sphere** [4]

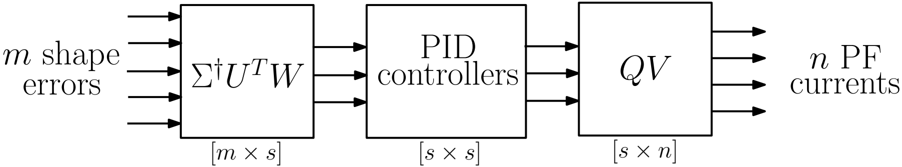
[4] i.e. every possible unit-norm $\delta\tilde{x}$.

Low-rank approximation

- Another advantage of using the svd is that it can be used to compute a **truncated** version of $C^\dagger$ (so-called **low-rank** approximation)
- The columns of V associated to the **largest singular values** represent the **most controllable current directions**
  - A small variations along such directions results in a **significant effect** on the shape control errors
- On the other hand, to affect the directions in U associated to **small** σ_i, we need **large current variations**
  - These could stress the actuators, so it is common practice to **discard the smallest singular values** (usually below some tolerance, e.g. 5% of σ_1)
  - In practice, we just set them to zero in $\Sigma^\dagger$

Control in the reduced space

- Finally, we only need to control the amplitude of the **generalized eigenmodes**
- The final scheme looks like this (including weight matrices)



Section 3

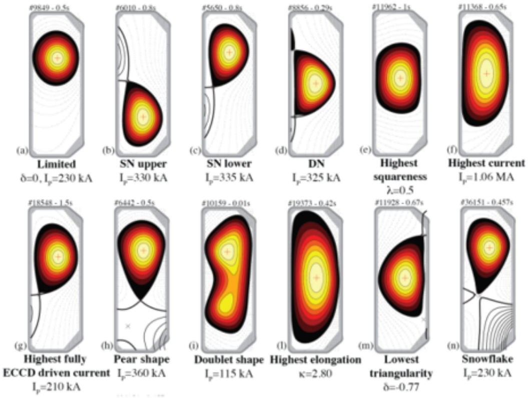
Shape control at TCV

Shape control at TCV

TCV is known for its **flexibility** in terms of plasma shaping

Early days approach:

- Rely on a simple core magnetic controller (*hybrid*)
    - robust & fast **RZIp** control
    - Shape obtained through careful **feedforward** design (`fbt`)



Shape control at TCV

Today:

- Reliable **modeling tools** (`fge`), allowing for reliable design and (offline) testing of feedback controllers
- **Digital control system** (with hybrid emulator - see C. Galperti's lecture), allowing for easy development and real-time testing of new controllers

Small history of shape control at TCV

- **Ariola et al., FT 1999**: pioneering attempt, controlled plasma shape parameters such as elongation and triangularity
- **Anand et al., NF 2017**: first isoflux controller, did not re-use the existing magnetic control architecture
- **Degrave, Felici et al., Nature 2022**: first deep reinforcement learning controller for shape control, very flexible but not suitable for day-to-day use (yet)

M. Ariola et al.
A modern plasma controller tested on the TCV tokamak
Fusion Technology, 1999

H. Anand et al.,
A novel plasma position and shape controller for advanced configuration development on the TCV tokamak
Nuclear Fusion, 2017

J. Degrave, F. Felici et al.
Magnetic control of tokamak plasmas through deep reinforcement learning
Nature, 2022

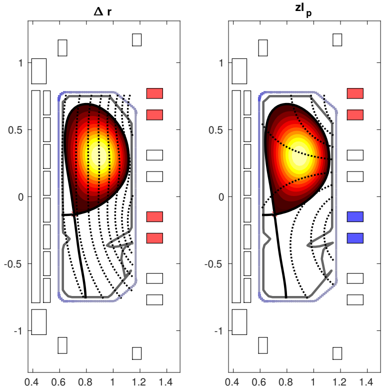
Shape control at TCV

Current design:

- Keep the *hybrid* controller
  - very reliable (30 years/82k shots of experience)
  - good fallback
- Build the **shape controller** on top of it
  - **Pro**: no need to worries about the rest of the magnetic controller, especially the VS
  - **Con**: slightly slower than theoretically achievable
- **Isoflux** approach

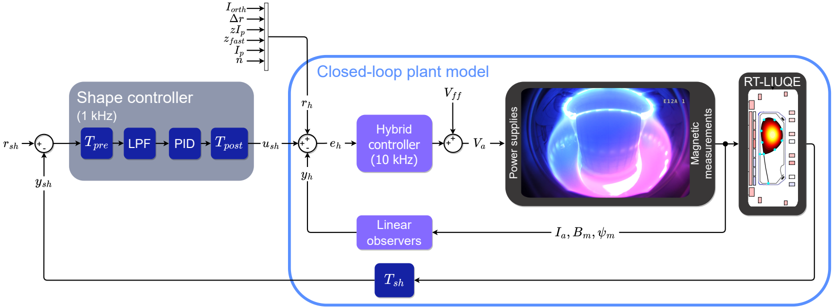
Shape control at TCV

- As we have seen, classic shape control relies on a static relation between **PF currents** and **shape descriptors**
- In *hybrid* however, some current directions are "directly plugged" into the position controller
- PF currents are controlled in a lower-dimensional subspace



Shape control at TCV

- *Hybrid* is a fully linear controller: it admits a **state-space representation**
- This state-space can be connected to a linearized `fgess` model
- The **static gain** of the closed-loop model can be used to obtain the T_{pre}/T_{post} decoupling matrices through low frequency, svd-based decoupling (instead of C)

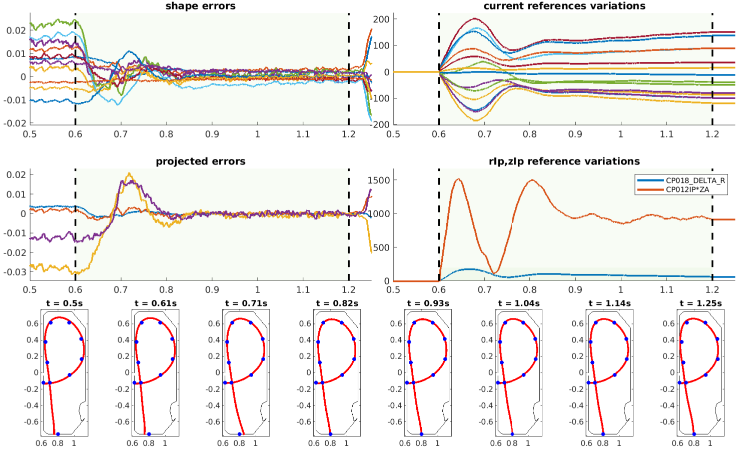


Shape control at TCV

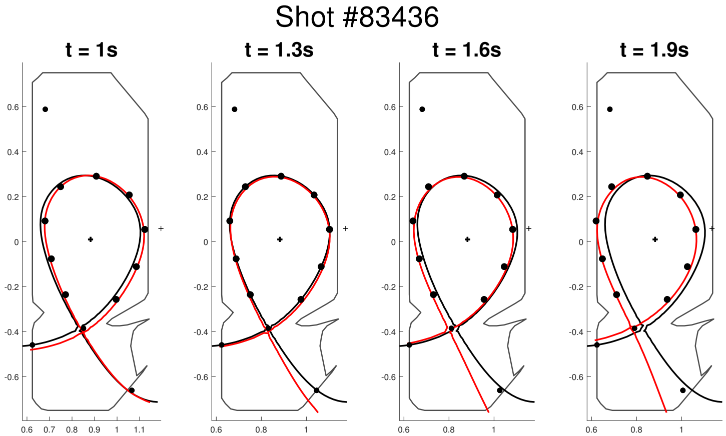
- Add PIDs on each svd mode to regulate dynamic performance
- Automatic tuning:
  - inversion formulae (fix crossing frequency + phase margin for each channel assuming perfect decoupling)
  - matlab automatic PID tuning
- The availability of a model of *hybrid* also allows **offline validation**!

Some results

Shot #79742

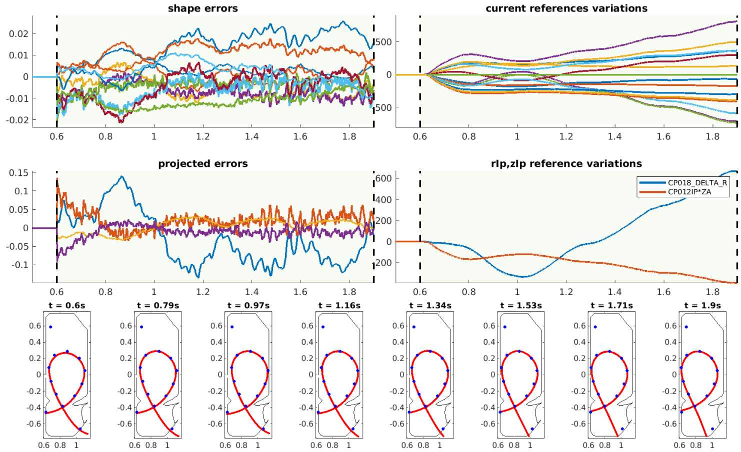


Some results

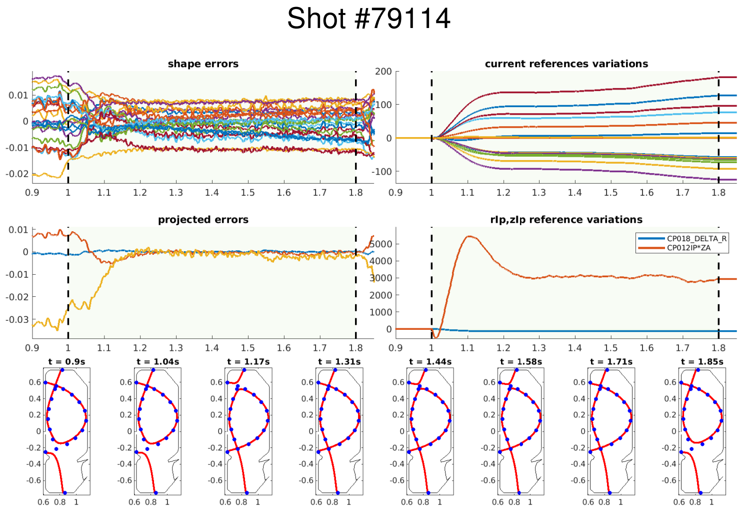


Some results

Shot #83436

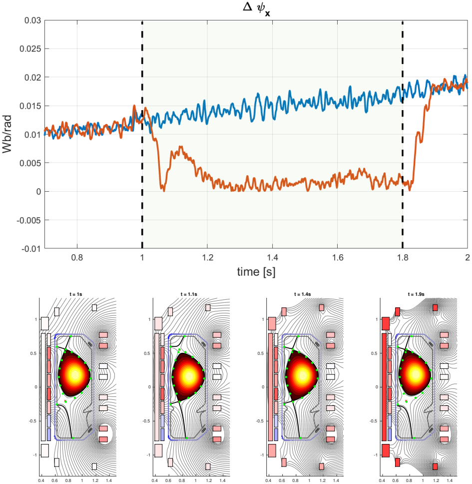


Some results



Some results

Shot #79102 vs #79114: improved null-points balancing

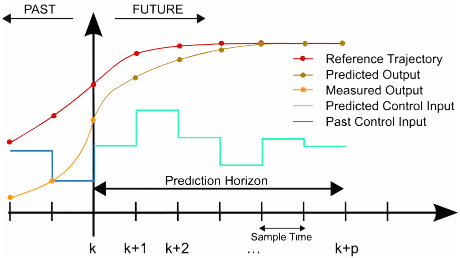


Section 4

Model Predictive Control for plasma shape

Model Predictive Control

- 90%+ of all controllers installed in the world are **PIDs**
- ...but recently, **MPC** is emerging as an alternative paradigm
- Core ideas:
  - Use a model to **forecast** the future trajectory
  - Find input sequence that provides the **optimal** forecast
  - Apply the **first** input from the sequence
  - Shift everything **forward** by one step and repeat



Model Predictive Control

- MPC-based shape control has been proposed for **future devices** (ITER, DEMO)
- These are expected to have slower dynamics $\rightarrow$ **more time** to solve the optimization problem
- Computational issues traditionally prevented the use of MPC for smaller, faster tokamaks... until now :)

M. Mattei et al.
A constrained control strategy for the shape control in thermonuclear fusion tokamaks
Automatica, 2013

S. Gerksic et al.
Model predictive control of ITER plasma current and shape using singular-value decomposition
Fus. Eng. Des., 2018

G. Tartaglione et al.
Plasma magnetic control for DEMO tokamak using MPC
IEEE Conference on Control Technology and Applications, 2022

Ingredients of MPC

- **System model**: ✔
  Mathematical description of **system dynamics**
- **Prediction**:
  Use model to predict future behavior **over a time horizon**
- **Optimal Control Sequence**:
  Calculate optimal control minimizing a **cost function**, considering **constraints**
- **Implementation**: **[mostly]**
  Feed real-time data to the controller
  Code everything into the control system

Real-time optimization

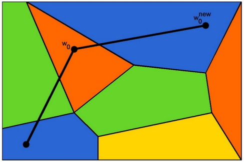
- Commercial and open-source solvers available
- They can process QP problems in standard form

$$\min_x x^T H x + f^T x$$
$$\text{s.t.} \quad Gx \leq h$$

$\rightarrow$ we will try to put our problem in this form...

OSQP
UNIVERSITY OF OXFORD

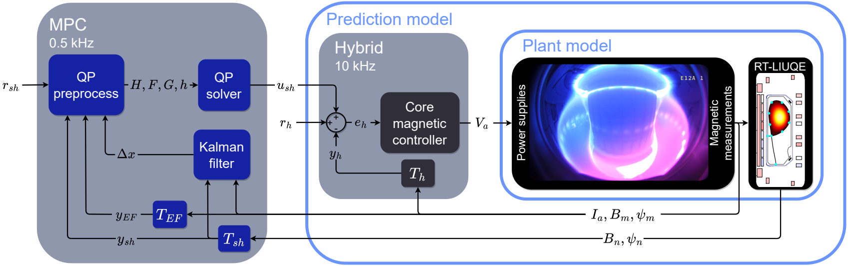
qpOASES



ODYS
Advanced Controls & Optimization

MPC control scheme

...to get a controller that will looks this:



Before we start: a fundamental remark

"You can only control as precisely as you can model!
If you want a highly tuned controller, you need a very accurate model."

J.A. Rossiter,
Model-Based Predictive Control - A practical approach, 2005

Prediction model

We need to recast our state-space(s) into a useful prediction model

- Plant model:

$$\begin{aligned}\delta\dot{x}_p(t) &= A_p\delta x_p(t) + B_p V_a(t) + E_p\delta\dot{w}(t)\\ \delta y_p(t) &= C_p\delta x_p(t) + F_p\delta w(t)\,,\end{aligned}$$

- *Hybrid* model:

$$\begin{aligned}\dot{x}_h(t) &= A_h x_h(t) + B_h e_h(t)\\ V_a(t) &= C_h x_h(t) + D_h e_h(t)\,,\end{aligned}$$

Prediction model

- All together (neglect δw for now, use *hybrid* reference as input)

$$
\begin{aligned}
\delta\dot{x}_p(t) &= A_p\delta x_p(t) + B_p(C_h x_h(t) + D_h(r_h - T_h C_p \delta x_p(t))) \\
\dot{x}_h(t) &= A_h x_h(t) + B_h(r_h - T_h C_p \delta x_p(t)) \\
V_a(t) &= C_h x_h(t) + D_h(r_h - T_h C_p \delta x_p(t)) \quad \text{(not used here, but useful!)} \\
\delta y_p(t) &= C_p \delta x_p(t)\,,
\end{aligned}
$$

- State-space matrices

$$
A = \begin{bmatrix} A_p - D_h T_h C_p & B_p C_h \\ -B_h T_h C_p & A_h \end{bmatrix}, \qquad B = \begin{bmatrix} D_h \\ B_h \end{bmatrix},
$$

$$
C = \begin{bmatrix} T_{EF} C_p & \mathbf{0}_{[l_{EF}\times n]} \\ T_{sh} C_p & \mathbf{0}_{[l_{sh}\times n]} \end{bmatrix}, \qquad D = \mathbf{0}_{[l\times n]}\,,
$$

Prediction model

- Often, prediction model is put in **velocity form**

$$x_k \to \Delta x_k := x_k - x_{k-1}$$
$$u_k \to \Delta u_k := u_k - u_{k-1}$$

  $\to$ Helps achieve **offset-free** tracking
- We keep the **"standard" expression** for the shape-related outputs y_k
  $\to$ Simplifies computing cost and imposing constraints
  $\to$ Update feedback from previous sample
- Sometimes the model is **augmented** to account for external disturbances
  $\to$ Neglected for now

Prediction model

- Velocity form state-space:

$$\begin{aligned} \Delta x_{k+1} &= A\Delta x_k + B\Delta u_k \\ y_k &= C\Delta x_k + D\Delta u_k + y_{k-1} \end{aligned}$$

- We need to predict the future trajectory, so define

$$y_{[k,k+N]} = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+N} \end{bmatrix}, \quad \Delta u_{[k,k+N]} = \begin{bmatrix} \Delta u_k \\ \Delta u_{k+1} \\ \vdots \\ \Delta u_{k+N} \end{bmatrix}$$

Prediction model

$$y_{[k,k+N]} = \underbrace{\Psi \Delta x_k + \Omega_I y_{k-1}}_{\substack{\hat{y}_{[k,k+N]}, \\ \text{prediction without} \\ \text{control}}} + \underbrace{\Phi \Delta u_{[k,k+N]}}_{\text{control effect}} \, .$$

where

$$\Psi = \begin{bmatrix} C \\ C + CA \\ \vdots \\ \sum_{i=0}^{N} CA^i \end{bmatrix}, \quad \Omega_I = \begin{bmatrix} \mathbf{I}_I \\ \mathbf{I}_I \\ \vdots \\ \mathbf{I}_I \end{bmatrix}, \quad \Phi = \underbrace{\begin{bmatrix} D & \mathbf{0} & \dots & \mathbf{0} \\ D + CB & D & \dots & \mathbf{0} \\ D + CB + CAB & D + CB & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ D + \sum_{i=0}^{N-1} CA^i B & D + \sum_{i=0}^{N-2} CA^i B & \dots & D \end{bmatrix}}_{\text{(Toeplitz matrix)}}$$

Optimization problem

We can now build the optimization problem

- Cost function

$$J(\underbrace{r_{[k,k+N]}}_{\substack{\text{depends on}\\ \text{the problem}}}, \underbrace{\Delta u_{[k,k+N]}}_{\substack{\text{optimization}\\ \text{variables}}}) =$$

$$(r_{[k,k+N]} - y_{[k,k+N]})^T W (r_{[k,k+N]} - y_{[k,k+N]}) + \Delta u_{[k,k+N]}^T Q \Delta u_{[k,k+N]} =$$

$$(\hat{e}_{[k,k+N]} - \Phi \Delta u_{[k,k+N]})^T W (\hat{e}_{[k,k+N]} - \Phi \Delta u_{[k,k+N]}) + \Delta u_{[k,k+N]}^T Q \Delta u_{[k,k+N]} \,,$$

where

$$\hat{e}_{[k,k+N]} = r_{[k,k+N]} - \hat{y}_{[k,k+N]}$$

Optimization problem

Define Hessian matrix and linear coefficient as

$$H = \Phi^T W \Phi + R \,, \quad f = -W \hat{e}_{[k,k+N]} \,.$$

to get to the standard QP form

$$\begin{aligned} \min_{\Delta u_{[k,k+N]}} \quad & \Delta u_{[k,k+N]}^T H \Delta u_{[k,k+N]} + f^T \Delta u_{[k,k+N]} \\ \text{s.t.} \quad & G \Delta u_{[k,k+N]} \leq h \,. \end{aligned}$$

Optimization problem

Any linear constraint can be imposed on the outputs:
for instance, **current saturations**

$$\underline{I}_{EF} \leq y_{EF_k} \leq \overline{I}_{EF}$$

$$\gamma_{EF} = \begin{bmatrix} \mathbf{I}_{n_{EF}} & \mathbf{0}_{[n_{EF},\, l-n_{EF}]} \end{bmatrix},$$

$$\Gamma = \begin{bmatrix} \gamma_{EF} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \gamma_{EF} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \gamma_{EF} \end{bmatrix},$$

$$\underbrace{\begin{bmatrix} +\Gamma\Phi \\ -\Gamma\Phi \end{bmatrix}}_{G} \Delta u_{[k,k+N]} \leq \underbrace{\begin{bmatrix} -\Gamma\hat{y}_{[k,k+N]} + \Omega_{n_{EF}}\overline{I}_{EF} \\ +\Gamma\hat{y}_{[k,k+N]} - \Omega_{n_{EF}}\underline{I}_{EF} \end{bmatrix}}_{h}$$

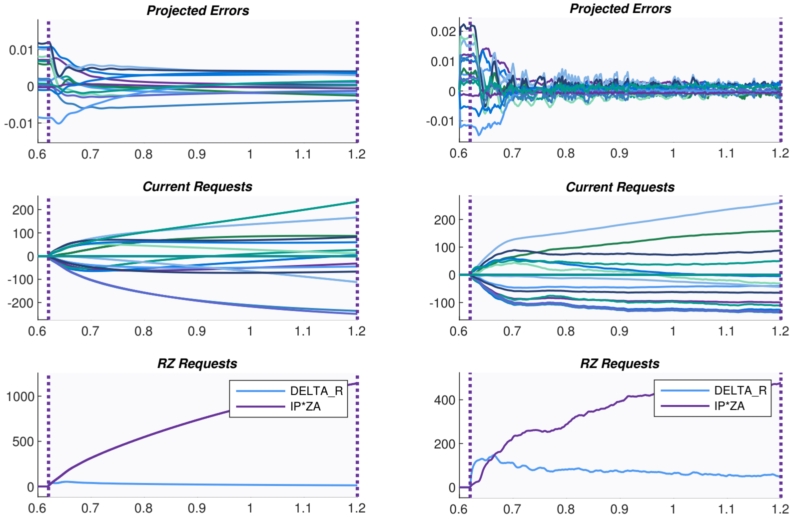
MPC streetfighting

Computational efficiency is key for MPC (remember, we only have few ms!)

- Pre-compute matrices where possible
- Reduce optimization problem
    - Use reduced **control horizon** $N_c < N$
    - Use SVD decomposition
    - Apply constraints every other step
- Choose solver wisely, use **warm-start** if possible

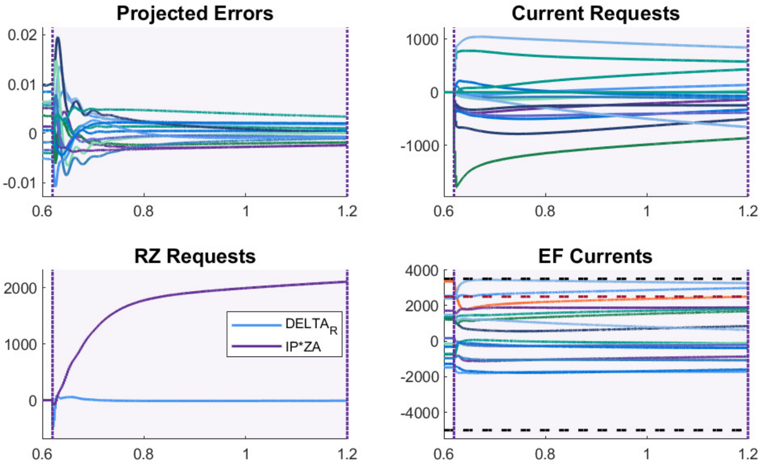
Some (preliminary) results

Simulated vs experimental results for a standard TCV discharge



Some (preliminary) results

Simulation with an artificially saturated coil



What next?

This approach is quite flexible and can be extended in several directions:

- **Solvers**: many solutions out there, waiting to be tested
- **Constraints**: add voltage or shape constraints
- **Disturbance rejection**: add a disturbance model to the prediction
- **Improve state estimation**: EKF, model identification, machine learning...
- **Adaptive MPC**: update the model online
- **Robust MPC**: account for model uncertainties
- **Learning MPC**: use machine learning to improve/correct/update the model

...a lot of work still to be done! Join the effort :)

Questions?