RTFs in fusion experiments

A plasma control experts' perspective

Future Improvements in Realtime Systems and Technologies (FIRST 2019)

Gianmaria De Tommasi
Email: detommas@unina.it

Cosylab - Ljubljana, Jan, 21st 2019

UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIETI.
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE

Disclaimer

- **I am the only and solely responsible** of the statements, opinions that you will see in this presentation

- **I am the only and solely responsible** of the statements, opinions that you will see in this presentation
- What is discussed in the next slides represents **my point of view** on the subject, which is based mainly on the experience that I did at JET and EAST, and working on International projects such as ITER and JT-60SA

Disclaimer

- **I am the only and solely responsible** of the statements, opinions that you will see in this presentation
- What is discussed in the next slides represents **my point of view** on the subject, which is based mainly on the experience that I did at JET and EAST, and working on International projects such as ITER and JT-60SA
- **If you think there should be someone to be blamed for what you will see, that person is me**

- **I am the only and solely responsible** of the statements, opinions that you will see in this presentation
- What is discussed in the next slides represents **my point of view** on the subject, which is based mainly on the experience that I did at JET and EAST, and working on International projects such as ITER and JT-60SA
- **If you think there should be someone to be blamed for what you will see, that person is me**
- **I need to thank all the colleagues together with whom I had the honor and pleasure to work, for everything you may find useful in this presentation**

Outline

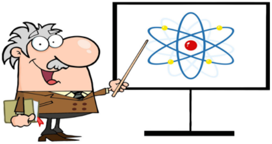
1. Development real-time systems in fusion experiments
   - The main characters
   - Automatic code generation
   - Model validation

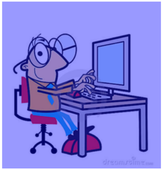
2. Standard control architecture for ITER-like machines

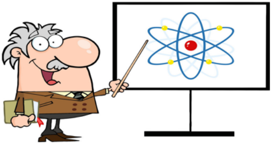
3. Conclusions & discussion points

The main characters

The CODAC expert (The Good), the Scientist (The Bad) and the Control Expert (The Ugly)

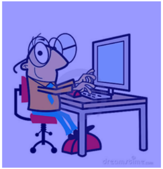


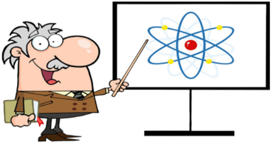




- **The CODAC expert**
  - knows very well the RTF (as developer or/and client)
  - has a good knowledge of the (real-time implementation of) control and diagnostic algorithms
-
-

The CODAC expert (The Good), the Scientist (The Bad) and the Control Expert (The Ugly)



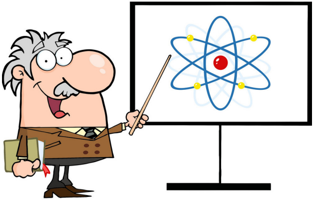


- **The CODAC expert**
  - knows very well the RTF (as developer or/and client)
  - has a good knowledge of the (real-time implementation of) control and diagnostic algorithms
- **The Scientist**
  - is (one of) the author(s) of the control or diagnostic algorithms
  - has some knowledge about RTF (sometimes a little bit more, sometimes little bit less)
-

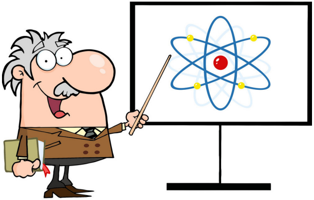
The CODAC expert (The Good), the Scientist (The Bad) and the Control Expert (The Ugly)

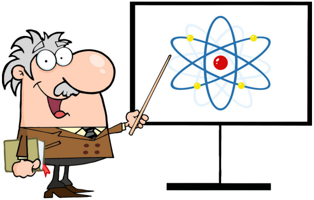
- **The CODAC expert**
  - knows very well the RTF (as developer or/and client)
  - has a good knowledge of the (real-time implementation of) control and diagnostic algorithms
- **The Scientist**
  - is (one of) the author(s) of the control or diagnostic algorithms
  - has some knowledge about RTF (sometimes a little bit more, sometimes little bit less)
- **The Control Expert**
  - works in a research institute or university
  - sometimes he/she does not have a strong knowledge of the domain
  - many times he/she knows nothing about RTF...
  - ...but he/she claims to be **THE** expert

- The three *roles* could overlap

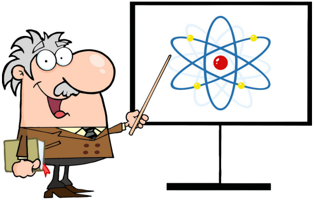


- The three *roles* could overlap
- Let us focus on **The Ugly**. . .

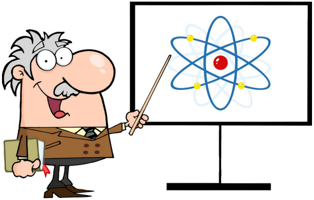




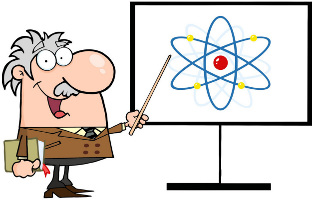
- The three *roles* could overlap
- Let us focus on **The Ugly**. . .
- . . . **The Control Expert**
  - works in Matlab/Simulink environment



- The three *roles* could overlap
- Let us focus on **The Ugly**. . .
- . . . **The Control Expert**
  - works in Matlab/Simulink environment
    - this is not necessarily true if we are dealing with a *Diagnostic Expert* (more on this later on)



- The three *roles* could overlap
- Let us focus on **The Ugly**. . .
- . . . **The Control Expert**
  - works in Matlab/Simulink environment
    - this is not necessarily true if we are dealing with a *Diagnostic Expert* (more on this later on)
  - thinks that (his/her) control algorithm is the most important thing in the world (kind of $\Pi\alpha\nu\acute{\alpha}\kappa\varepsilon\iota\alpha$)



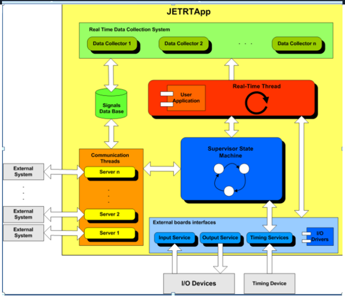
- The three *roles* could overlap
- Let us focus on **The Ugly**. . .
- . . . **The Control Expert**
  - works in Matlab/Simulink environment
    - this is not necessarily true if we are dealing with a *Diagnostic Expert* (more on this later on)
  - thinks that (his/her) control algorithm is the most important thing in the world (kind of Πανάκεια)
  - does not want to *loose time* on the details of the RTF infrastructure/architecture

- Since we are dealing with International projects, we cannot kill The Ugly

- Since we are dealing with International projects, we cannot kill The Ugly

- We need to learn how to leave with The Ugly, minimising the effort needed to get his amazing control algorithm

To confine (as much as possible) the Ugly



The (obvious) **plug-in** solution

- Was not so common in the fusion community (at the beginning of the new millennium)

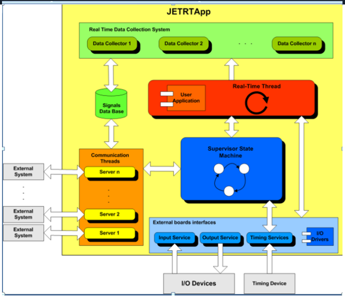
Some references (some very old)

G. De Tommasi et al.
A flexible software for real-time control in nuclear fusion experiments
Contr. Eng. Prac., 2006

A. Neto et al.,
MARTe: A Multiplatform Real-Time Framework
IEEE Trans. Nucl. Sci., 2010

W. Treutterer et al.,
ASDEX Upgrade Discharge Control System – A real-time plasma control framework
Fus. Eng. Des., 2014

To confine (as much as possible) the Ugly



Some references (some very old)

G. De Tommasi et al.
A flexible software for real-time control in nuclear fusion experiments
Contr. Eng. Prac., 2006

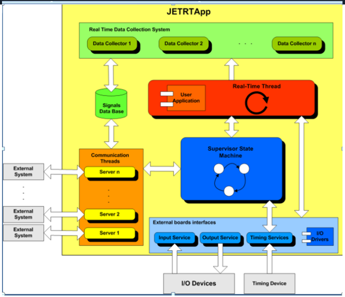
A. Neto et al.,
MARTe: A Multiplatform Real-Time Framework
IEEE Trans. Nucl. Sci., 2010

W. Treutterer et al.,
ASDEX Upgrade Discharge Control System – A real-time plasma control framework
Fus. Eng. Des., 2014

The (obvious) **plug-in** solution

- Was not so common in the fusion community (at the beginning of the new millennium)
- First version of the JET RTF in 2003. . . then *GAMs* in MARTe. . .

To confine (as much as possible) the Ugly



Some references (some very old)

G. De Tommasi et al.
A flexible software for real-time control in nuclear fusion experiments
Contr. Eng. Prac., 2006

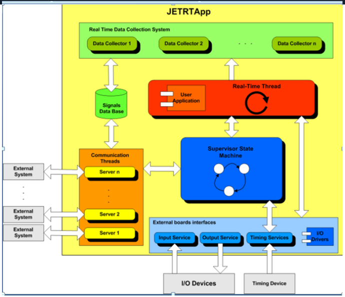
A. Neto et al.,
MARTe: A Multiplatform Real-Time Framework
IEEE Trans. Nucl. Sci., 2010

W. Treutterer et al.,
ASDEX Upgrade Discharge Control System – A real-time plasma control framework
Fus. Eng. Des., 2014

The (obvious) **plug-in** solution

- Was not so common in the fusion community (at the beginning of the new millennium)
- First version of the JET RTF in 2003. . . then *GAMs* in MARTe. . .
- . . . *ApplicationProcess* in ASDEX DCS. . .

To confine (as much as possible) the Ugly



Some references (some very old)

G. De Tommasi et al.
A flexible software for real-time control in nuclear fusion experiments
Contr. Eng. Prac., 2006

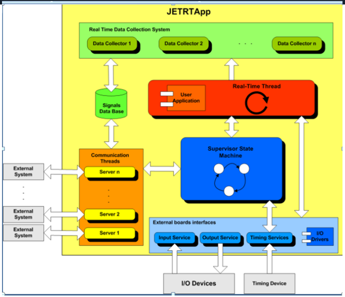
A. Neto et al.,
MARTe: A Multiplatform Real-Time Framework
IEEE Trans. Nucl. Sci., 2010

W. Treutterer et al.,
ASDEX Upgrade Discharge Control System – A real-time plasma control framework
Fus. Eng. Des., 2014

The (obvious) **plug-in** solution

- Was not so common in the fusion community (at the beginning of the new millennium)
- First version of the JET RTF in 2003. . . then *GAMs* in MARTe. . .
- . . . *ApplicationProcess* in ASDEX DCS. . .
- . . . there exists some *separation* also in the GA PCS (even if not documented in papers). . .

To confine (as much as possible) the Ugly



The (obvious) **plug-in** solution

- Was not so common in the fusion community (at the beginning of the new millennium)
- First version of the JET RTF in 2003. . . then *GAMs* in MARTe. . .
- . . . *ApplicationProcess* in ASDEX DCS. . .
- . . . there exists some *separation* also in the GA PCS (even if not documented in papers). . .
- . . . similar concept in the ITER RTF

Some references (some very old)

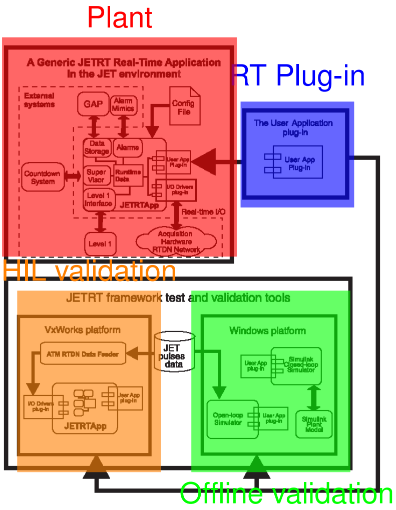
G. De Tommasi et al.
A flexible software for real-time control in nuclear fusion experiments
Contr. Eng. Prac., 2006

A. Neto et al.,
MARTe: A Multiplatform Real-Time Framework
IEEE Trans. Nucl. Sci., 2010

W. Treutterer et al.,
ASDEX Upgrade Discharge Control System – A real-time plasma control framework
Fus. Eng. Des., 2014

A **Real-time Application Plug-in** that can be used:

- to perform offline validation against a plat model
- to perform real-time validation with hardware-in-the-loop
- to run the real-time system on the plant



Automatic code generation

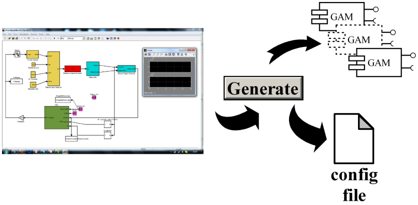


Figure: Old idea (never implemented, AFAIK).

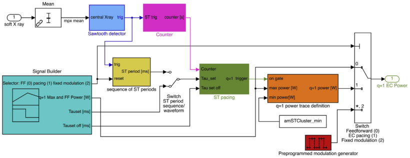


Figure: Taken from Felici et al., FED, 2014.

Ok, good, problem solved! The Ugly can be *isolated*

- **Automatic code generation**

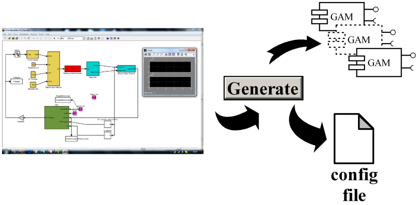


Figure: Old idea (never implemented, AFAIK).

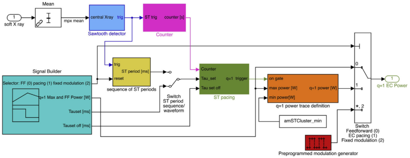


Figure: Taken from Felici et al., FED, 2014.

Ok, good, problem solved! The Ugly can be *isolated*

- **Automatic code generation**
- Commonly used in many fields: rapid prototyping (dSpace, NI, Mathworks,*ldots*), control education (Quanser), automotive, railway (ATO in the future?). . . .

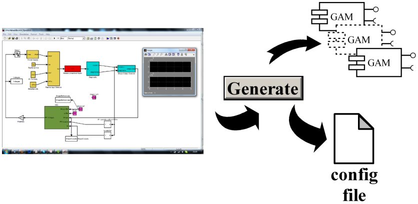


Figure: Old idea (never implemented, AFAIK).

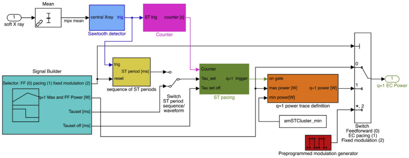


Figure: Taken from Felici et al., FED, 2014.

Ok, good, problem solved! The Ugly can be *isolated*

- **Automatic code generation**
- Commonly used in many fields: rapid prototyping (dSpace, NI, Mathworks,*ldots*), control education (Quanser), automotive, railway (ATO in the future?). . . .
- IMHO, TCV is the *state of the art* in the community

Automatic code generation

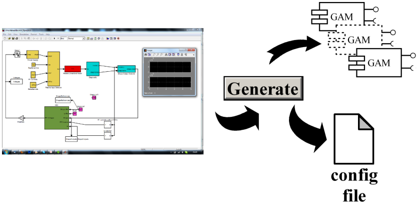


Figure: Old idea (never implemented, AFAIK).

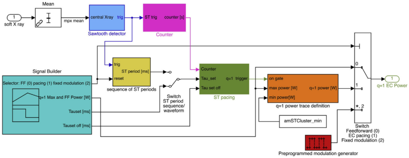


Figure: Taken from Felici et al., FED, 2014.

Ok, good, problem solved! The Ugly can be *isolated*

- **Automatic code generation**
- Commonly used in many fields: rapid prototyping (dSpace, NI, Mathworks,*ldots*), control education (Quanser), automotive, railway (ATO in the future?). . . .
- IMHO, TCV is the *state of the art* in the community **(but I am not sure that everything can be extended to ITER!)**

Automatic code generation at ITER

- Automatic code generation is one of the option included in the ITER PCS simulation platform (PCSSP)

Automatic code generation at ITER

- Automatic code generation is one of the option included in the ITER PCS simulation platform (PCSSP)
- So far, little effort has been put **on this aspect**
- The needed effort is not negligible

Automatic code generation at ITER

- Automatic code generation is one of the option included in the ITER PCS simulation platform (PCSSP)
- So far, little effort has been put **on this aspect**
- The needed effort is not negligible
- **Generally, the effort to develop a *robust* PCS simulation platform for ITER is not negligible**

Automatic code generation at ITER

- Automatic code generation is one of the option included in the ITER PCS simulation platform (PCSSP)
- So far, little effort has been put **on this aspect**
- The needed effort is not negligible
- **Generally, the effort to develop a *robust* PCS simulation platform for ITER is not negligible**
- The development of such a platform should not be left only to The Ugly(ies) **(remember the disclaimer!)**

More references

F. Felici et al.
Development of real-time plasma analysis and control algorithms for the TCV tokamak using Simulink
Fus. Eng. Des., 2014

M. L. Walker et al.,
A simulation environment for ITER PCS development
Fus. Eng. Des., 2014

More on automatic code generation. . .

- It works mainly for (simple) control algorithms
  - what about control algorithms that requires online solution of optimization problems?
  - what about *support functions* (ITER jargon for diagnostic functions *et similia*)?

MPC example

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

M. Perne et al.,
Soft inequality constraints in gradient method and fast gradient method for quadratic programming
Opt. Eng., 2018

. . . and more (on support functions)

- As far as *support functions* are concerned
  - legacy codes, usually written in Fortran, C, C++

. . . and more (on support functions)

- As far as *support functions* are concerned
  - legacy codes, usually written in Fortran, C, C++
  - plasma equilibrium codes: EFIT, PEFIT, CCS, LIOQE, . . .
  - plasma boundary reconstruction codes: XLOC, . . .

. . . and more (on support functions)

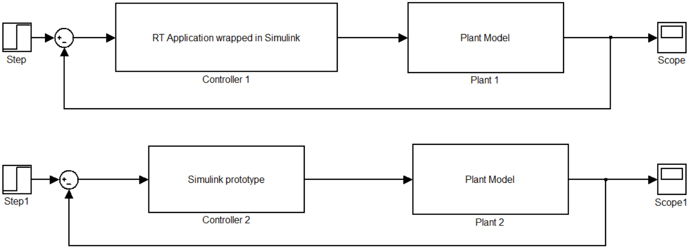
- As far as *support functions* are concerned
  - legacy codes, usually written in Fortran, C, C++
  - plasma equilibrium codes: EFIT, PEFIT, CCS, LIOQE, . . .
  - plasma boundary reconstruction codes: XLOC, . . .
- When dealing with The Ugly, for these *modules* (but even for existing control algorithms), it is important also to consider the **inverse process** (wrt automatic code generation)

- As far as *support functions* are concerned
  - legacy codes, usually written in Fortran, C, C++
  - plasma equilibrium codes: EFIT, PEFIT, CCS, LIOQE, . . .
  - plasma boundary reconstruction codes: XLOC, . . .
- When dealing with The Ugly, for these *modules* (but even for existing control algorithms), it is important also to consider the **inverse process** (wrt automatic code generation)
- The *inverse process* is useful (and should be included in the RTF, see also **Anze's presentation**)
  - to validate coding, when automatic code generation is not available

. . . and more (on support functions)

- As far as *support functions* are concerned
  - legacy codes, usually written in Fortran, C, C++
  - plasma equilibrium codes: EFIT, PEFIT, CCS, LIOQE, . . .
  - plasma boundary reconstruction codes: XLOC, . . .
- When dealing with The Ugly, for these *modules* (but even for existing control algorithms), it is important also to consider the **inverse process** (wrt automatic code generation)
- The *inverse process* is useful (and should be included in the RTF, see also **Anze's presentation**)
  - to validate coding, when automatic code generation is not available
  - to trust automatic code generation, when available

The *inverse process*



Embedding RTF code into Simulink

- In most of the cases to check the real-time version of the controller it is convenient to run it against the plant model in the Simulink environment.
- The real-time version and the Simulink version of the system can be run in *parallel* in order to perform the validation

Example: commissioning of the ITER-like VS at EAST

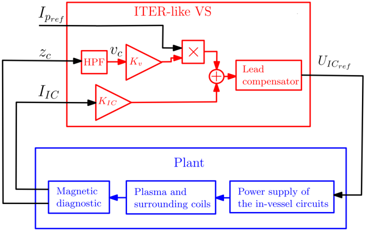


Figure: ITER-like vertical stabilization controller.

$$U_{IC_{ref}}(s) = \frac{1+s\tau_1}{1+s\tau_2} \cdot \left(K_v \cdot \bar{I}_{p_{ref}} \cdot \frac{s}{1+s\tau_z} \cdot Z_c(s) + K_{IC} \cdot I_{IC}(s) \right)$$

CREATE

3. *Commissioning procedure*

3.1 Commissioning of the voltage-driven VS system

This section describes the commissioning procedure for the voltage-driven VS system.

3.1.1 Pre-requisites for the voltage-driven VS commissioning

The following systems/services are required and must be successfully commissioned before the VS commissioning:

1. PFC current control system.
2. Plasma current control system.
3. Bang-bang voltage-driven VS controller.

Furthermore:

4. The **reference pulse 52444** should be reproduced under the current machine configuration.
5. The code that implements the control algorithm (1) should be validated against its Simulink version.

Item 5 will be performed once the CREATE team will be onsite.

Figure: Excerpts of the ITER-like VS commissioning procedure.

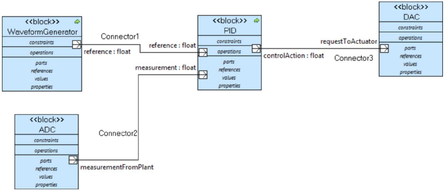
Model-driven design

- This issue is not really related to The Ugly, but it is somehow linked to automatic code generation

Model-driven design

- This issue is not really related to The Ugly, but it is somehow linked to automatic code generation
- Once a library of reusable blocks is available within the RTF, it may be useful to have a model-driven design tool

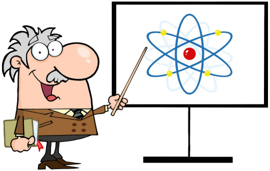
- This issue is not really related to The Ugly, but it is somehow linked to automatic code generation
- Once a library of reusable blocks is available within the RTF, it may be useful to have a model-driven design tool
  - Example: model your system in SysML and automatically generate the RTF configuration

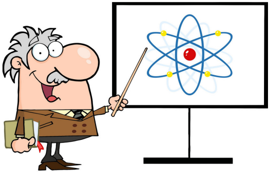


An incomplete example for FTU

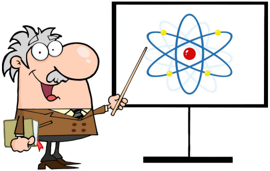
G. De Tommasi et al.,
Modeling of MARTe-Based Real-Time Applications With SysML
IEEE Trans. Ind. Inf., 2013

- **The Control Expert** designs the control algorithms using a model-based approach

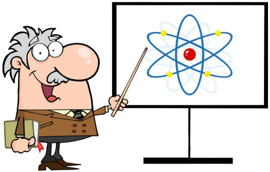




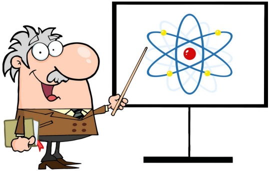
- **The Control Expert** designs the control algorithms using a model-based approach
- **Control-oriented models** are (simple) linear and time-invariant; the overall plant model is obtained exploiting simplifying assumptions
  - These models are indeed used for the design of the control algorithm



- **The Control Expert** designs the control algorithms using a model-based approach
- **Control-oriented models** are (simple) linear and time-invariant; the overall plant model is obtained exploiting simplifying assumptions
  - These models are indeed used for the design of the control algorithm
  - In order to assess (robust) performance a more complex model is desirable, which takes into account the *peculiarities* (nonlinearities) of the diagnostic and/or actuators



- **The Control Expert** designs the control algorithms using a model-based approach
- **Control-oriented models** are (simple) linear and time-invariant; the overall plant model is obtained exploiting simplifying assumptions
  - These models are indeed used for the design of the control algorithm
  - In order to assess (robust) performance a more complex model is desirable, which takes into account the *peculiarities* (nonlinearities) of the diagnostic and/or actuators
  - The ***inverse process*** may help also in this case



- **The Control Expert** designs the control algorithms using a model-based approach
- **Control-oriented models** are (simple) linear and time-invariant; the overall plant model is obtained exploiting simplifying assumptions
  - These models are indeed used for the design of the control algorithm
  - In order to assess (robust) performance a more complex model is desirable, which takes into account the *peculiarities* (nonlinearities) of the diagnostic and/or actuators
  - The ***inverse process*** may help also in this case
- **The *inverse process* is useful also to validate the control-oriented models**

Embedding support functions in the Ugly's simulation suite

- In order to take into account the nonlinear effect of diagnostics and/or actuators it is useful to provide the **Control Expert** with a Simulink version of specific functions that run on the real system

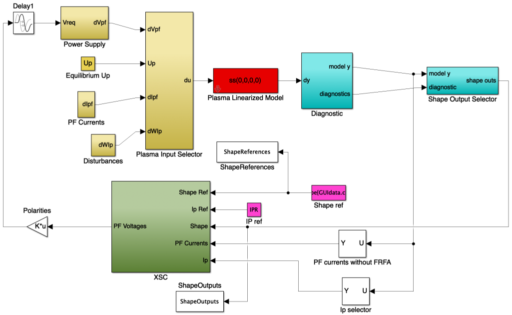
Embedding support functions in the Ugly's simulation suite

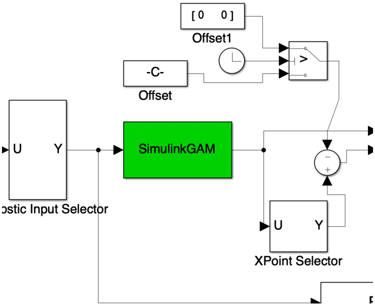
- In order to take into account the nonlinear effect of diagnostics and/or actuators it is useful to provide the **Control Expert** with a Simulink version of specific functions that run on the real system
    - **Usually requires *ad hoc* coding**

Embedding support functions in the Ugly's simulation suite

- In order to take into account the nonlinear effect of diagnostics and/or actuators it is useful to provide the **Control Expert** with a Simulink version of specific functions that run on the real system
  - **Usually requires *ad hoc* coding**
  - **It would be nice to have the inverse option in the RTF**

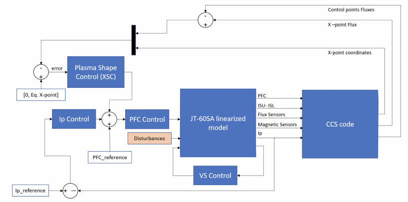
- Plasma shape controller designed on a linear approximation of the *gaps* behavior
- Validation in closed-loop using the real-time code for plasma boundary reconstruction *wrapped* into a S-function
- **Some work needed, but ok (modularity and separation of MARTe were exploited)**





The JT-60SA CCS case

- Objective: preliminary assessment of the effect of the measurement noise on plasma boundary reconstruction
- QST colleagues provided the Cauchy Condition Surface (CCS) code
- Fortran compiled as an executable that runs only on Linux and I/O with text files
- **Closed-loop simulations take ages! KO**



A2. How to run CCS

Binary of CCS: ccs_sa
Sample shell script for running binary of CCS: run.csh

[If you implement CCS with CREATE]

1. CREATE make “fort.70” in reference to inner plasma equilibrium and “fort.21” provided by QST magnetic controller
2. Run “ccs_sa”
3. CREATE read controlled variables and so on from “fort.71” which is output of binary of CCS

- Although their are simple, the linear models need to be validated and *tuned* on the experiment
- This experimental validation is an important step to be carried out before starting the design of *advanced model-based controller*

Validation of the Ugly's modelling tools

- Although their are simple, the linear models need to be validated and *tuned* on the experiment
- This experimental validation is an important step to be carried out before starting the design of *advanced model-based controller*
  - Advanced controller = whatever *more* than a PI experimentally tuned

Validation of the Ugly's modelling tools

- Although their are simple, the linear models need to be validated and *tuned* on the experiment
- This experimental validation is an important step to be carried out before starting the design of *advanced model-based controller*
    - Advanced controller = whatever *more* than a PI experimentally tuned
- Both open-loop and closed-loop validations are needed

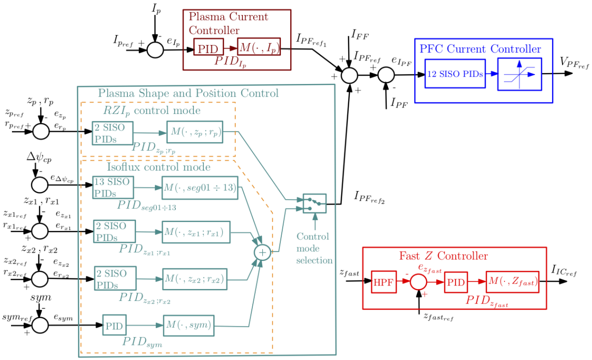
- Although their are simple, the linear models need to be validated and *tuned* on the experiment
- This experimental validation is an important step to be carried out before starting the design of *advanced model-based controller*
  - Advanced controller = whatever *more* than a PI experimentally tuned
- Both open-loop and closed-loop validations are needed
- **Closed-loop validation is important when dealing with unstable plants**

- In order to validate/tune the CREATE plasma/tokamak linear model a campaign of simulations was carried out

Validation of EAST magnetic model

- In order to validate/tune the CREATE plasma/tokamak linear model a campaign of simulations was carried out
- Since EAST colleagues **did not** provide a Simulink version of the existing controllers. . .
  1. Back engineering of the existing magnetic control system
     A LOT OF TIME
  2. Tuning of the model
  3. Design of new controllers
  4. Something not working because changes have been made in the EAST PCS (new control modes, new bumpless transfer, . . .)? $\rightarrow$ **go back to 1**

Back engineering the EAST magnetic control system



Q. P. Yuan et al.
Plasma current, position and shape feedback control on EAST
Nucl. Fus., 2013

A. Castaldo et al.
Simulation suite for plasma magnetic control at EAST tokamak
Fus. Eng. Des., 2018

Open-loop model validation

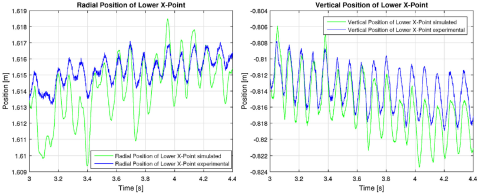


Fig. 6. Comparison between simulated (green solid line) and experimental (blue solid line) plasma Lower X-point radial (left figure) and vertical (right figure) position for pulse #69449. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

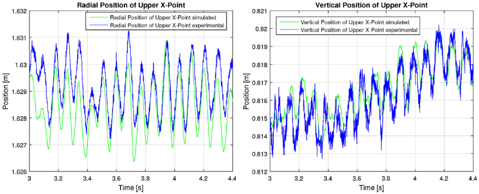


Fig. 7. Comparison between simulated (green solid line) and experimental (blue solid line) plasma Upper X-point radial (left figure) and vertical (right figure) position for pulse #69449. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

A. Castaldo et al.
Simulation suite for plasma magnetic control at EAST tokamak
Fus. Eng. Des., 2018

Open-loop EAST controller validation

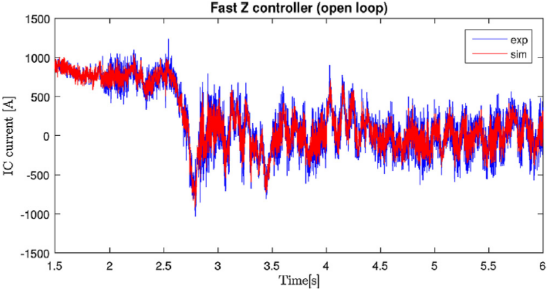


Fig. 12. Fast Z controller (including the high pass filter) output for pulse #74104. The small discrepancies are due to a subsampling of the experimental feedback signal.

A. Castaldo et al.
Simulation suite for plasma magnetic control at EAST tokamak
Fus. Eng. Des., 2018

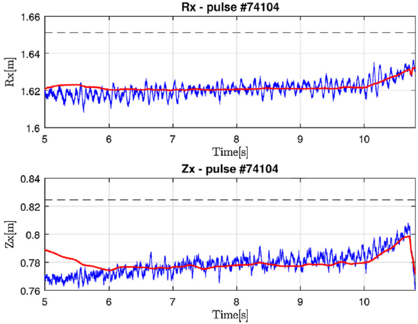
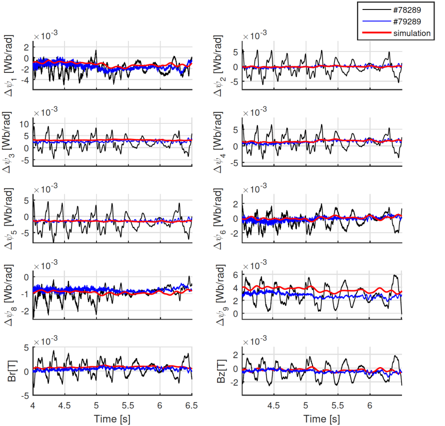


Fig. 14. Simulated and experimental X-point position for pulse #74104, obtained with the existing EAST controller. The experimental signal is shown in blue, while the simulated one is in red. The dashed black line shows the reference signal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

A. Castaldo et al.
Simulation suite for plasma magnetic control at EAST tokamak
Fus. Eng. Des., 2018

Model-based controller tuning



A. Mele et al.

MIMO shape control at EAST tokamak: simulations and experiments

30th Symposium on Fusion Technology (SOFT), 2018

Filling RTFs with control algorithms

- RTFs are one part of the whole story
- There are also control algorithms and support functions

Filling RTFs with control algorithms

- RTFs are one part of the whole story
- There are also control algorithms and support functions
- **Why CODAC Experts (The Good) should leave the fun only to the Control Experts (The Ugly)?**

Filling RTFs with control algorithms

- RTFs are one part of the whole story
- There are also control algorithms and support functions
- **Why CODAC Experts (The Good) should leave the fun only to the Control Experts (The Ugly)?**
- **Standard control algorithms (and support functions) should come together with the RTF (GA PCS approach)**

Filling RTFs with control algorithms

- RTFs are one part of the whole story
- There are also control algorithms and support functions
- **Why CODAC Experts (The Good) should leave the fun only to the Control Experts (The Ugly)?**
- **Standard control algorithms (and support functions) should come together with the RTF (GA PCS approach)**
- **The Control Expert can still play with advanced control algorithms (The Ugly becomes The Man With The Rifle)**

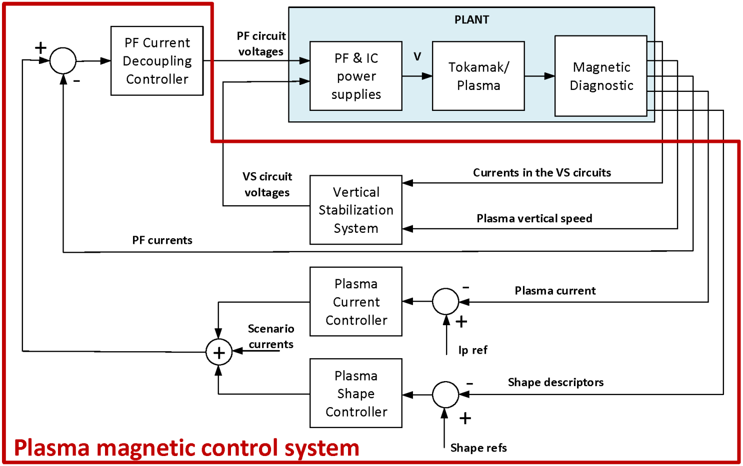
- A magnetic control architecture able to operate the plasma for an entire duration of the discharge, from the initiation to plasma ramp-down

- A magnetic control architecture able to operate the plasma for an entire duration of the discharge, from the initiation to plasma ramp-down
- *Machine-agnostic* architecture (aka *machine independent* solution)
- Model-based control algorithms
  - $\rightarrow$ the design procedures relies on (validated) control-oriented models for the response of the plasma and of the surrounding conductive structures
- The proposal is based on the JET experience

- A magnetic control architecture able to operate the plasma for an entire duration of the discharge, from the initiation to plasma ramp-down
- *Machine-agnostic* architecture (aka *machine independent* solution)
- Model-based control algorithms
  - → the design procedures relies on (validated) control-oriented models for the response of the plasma and of the surrounding conductive structures
- The proposal is based on the JET experience
- The architecture has been proposed for ITER & JT-60SA (& DEMO) and has been partially deployed at EAST (ongoing activity)

G. De Tommasi
Plasma Magnetic Control in Tokamak Devices
J. Fus. Energy, 2018

A possible architecture - 1/2



A possible architecture - 2/2

- **Include *standard* algorithms for each controllers of the architecture**
  - Current decoupling controller
  - Vertical stabilization controller
  - Plasma current controller
  - Plasma shape controller

Conclusions - Discussion points

- The Control Expert is happy if he/she can easily plug his/her own stuff into RTF without caring too much about the details
- The Control Expert is even more happy if he/she get easily get pieces of an existing system and run them into his/her simulation environment

Conclusions - Discussion points

- The Control Expert is happy if he/she can easily plug his/her own stuff into RTF without caring too much about the details
- The Control Expert is even more happy if he/she get easily get pieces of an existing system and run them into his/her simulation environment
- **These concepts, although they might seem obvious, they are not!**

Conclusions - Discussion points

- The Control Expert is happy if he/she can easily plug his/her own stuff into RTF without caring too much about the details
- The Control Expert is even more happy if he/she get easily get pieces of an existing system and run them into his/her simulation environment
- **These concepts, although they might seem obvious, they are not!**
  - Example: **in my view** EU contribution to JT-60SA may be affected by the lack of these (and other) features

Conclusions - Discussion points

- The Control Expert is happy if he/she can easily plug his/her own stuff into RTF without caring too much about the details
- The Control Expert is even more happy if he/she get easily get pieces of an existing system and run them into his/her simulation environment
- **These concepts, although they might seem obvious, they are not!**
  - Example: **in my view** EU contribution to JT-60SA may be affected by the lack of these (and other) features
- It may be the case to invest not only on RTFs, but also on *standard* control algorithms (as GA PCS)

. . .I hope that The Ugly will appear a little less ugly to The Good

RTFs in fusion experiments

A plasma control experts' perspective

Future Improvements in Realtime Systems and Technologies (FIRST 2019)

Gianmaria De Tommasi
Email: detommas@unina.it

Cosylab - Ljubljana, Jan, 21st 2019

Thank you!

DIETI. UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE