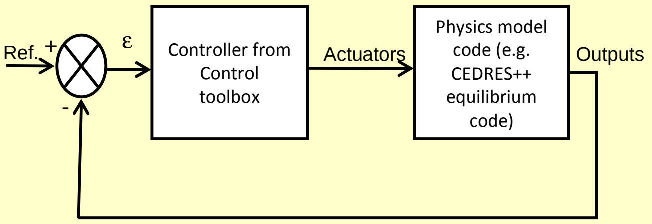
Feedback control simulation under the ITM platform

O. Barana[a], C. Boulbe[b], S. Bremond[a], S. Mannori[c], Ph. J. Moreau[a], N. Ravenel[a]
and the EFDA ITM Task Force contributors

[a]CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France.
[b]Université de Nice Sophia-Antipolis, 06108 Nice, France.
[c]INRIA Rocquencourt, 78153 Le Chesnay, France.

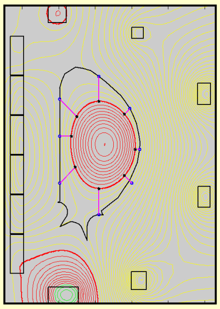
Purpose – Means - ITM Tasks

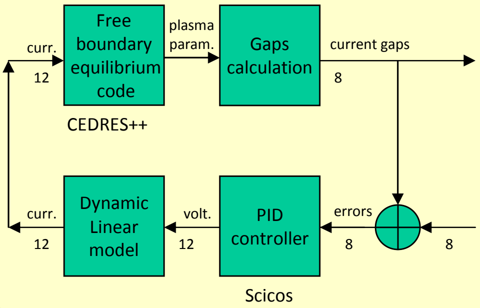
- Purpose: study of scenarios where feedback control is mandatory (i.e. scenario design with PF coils constraints included).
- Means: effective control simulation framework under ITM.
- ITM Tasks: cross-task project involving
  - ➢Task WP10-ITM-IMP12-ACT2: free-boundary equilibrium code;
  - ➢Task ISIP-T12: control toolbox.


Ref.
+
ε
-
Controller from Control toolbox
Actuators
Physics model code (e.g. CEDRES++ equilibrium code)
Outputs

Current status

- Development of a test case: ITER plasma shape control.
  - ➢Circular plasma to avoid vertical instability issues.


Test case reference MHD equilibrium
(flux contour plot from CEDRES++)
ITER limiter plasma

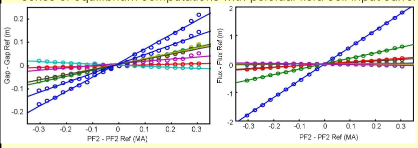

Free boundary equilibrium code
plasma param.
Gaps calculation
current gaps
8
curr.
12
CEDRES++
Dynamic Linear model
curr.
12
volt.
12
PID controller
errors
8
8
Scicos

Control scheme under Kepler.
Control of 8 gaps using 12 poloidal field coils as actuators.

Feedback control design

LINEAR PLASMA RESPONSE MODEL IDENTIFICATION WITH CEDRES++ :

- series of equilibrium computations with poloidal field coil input currents slightly varied from reference values.



1) $\delta Gaps = M_{GAP} * \delta I_{PF}$ 2) $\delta Flux = M_{FLUX} * \delta I_{PF}$

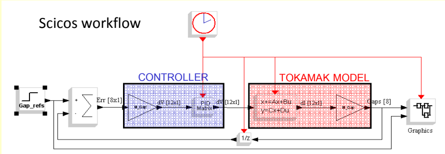
Circuit equations:
3) $\delta V = R * \delta I_{PF} + d/dt(\delta Flux)$
where R is a diagonal matrix of resistances.

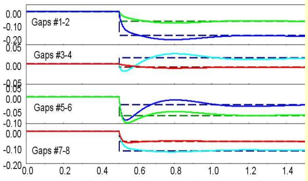
<u>Dynamic linear model</u>

$$d/dt(\delta I_{PF}) = -M_{FLUX}^{-1} * \delta I_{PF} + M_{FLUX} * \delta V$$

CONTROLLER DESIGN AND TUNING :

- Multivariable PID controller;
- Control toolbox: Scicoslab/Scicos (developed in METALAU project at INRIA);
- The current dynamic linear model is comprised in the state space equation system (ABCD block).



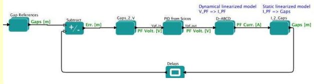


Closed loop response to steps on gap references

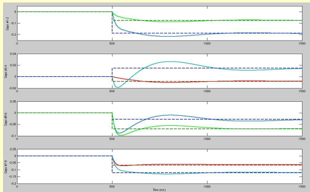
Kepler simulation 1/2

STEP 1 :
- Clone of Scicos workflow;
- Multivariable PID controller designed with Kepler actors.

STEP 2 :
- Substitution of the PID controller designed with Kepler actors with the PID controller generated using the Scicos control toolbox and fc2k.

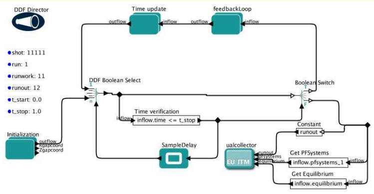


Feedback control loop within the main Kepler loop



Closed loop response to steps on gap references

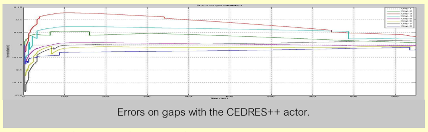
STEP 3 :
- Substitution of the dynamical and static linearized model actors with the CEDRES++ actor.
- The CEDRES++ code was modified in order to compute the gaps too.
- Introduction of the actors needed to exchange data with the Universal Access Layer (UAL).



Standard ITM loop for simulations under Kepler

- The *ualinit* actor is hidden in the Initialization actor;
- The *ualmux* and *ualdemux* actors are used, inside the *feedbackLoop* actor, to manage the PF current update (see also <u>Data Handling</u> in the <u>Major Issues</u> section).

Kepler simulation 2/2



Errors on gaps with the CEDRES++ actor.

CEDRES++ PROBLEMS :
- The workflow with the CEDRES++ actor does not have the same behaviour as the workflow using the linearized models;
- In particular:
  - the dynamic is much slower (unchanged controller);
  - jumps can be noticed in the error curves; these jumps might be the results of the mesh used in CEDRES++ and might explain why the dynamic has changed.

ADDITIONAL PROBLEMS :
- The current CEDRES++ version takes about 10 s to compute an equilibrium on the ITM Gateway;
- A feedback control loop simulation typically spans over several hundreds of time steps, with an actual simulation time of several hours (e.g. in our case a time step is equal to 1 ms and the theoretical simulation time is 1.5 s, but with 10 s for each time step the actual simulation time is greater than 4 hours);
- On the ITM Gateway it is very difficult to run simulations which last for hours and to tune parameters: the CPU time is limited, often the communication and, most of all, a workflow execution is very demanding in terms of resources (causing the process to be killed); currently a batch execution is not possible.

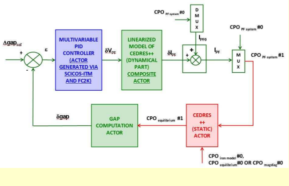
Major issues

DATA HANDLING :

- CPOs (Consistent Physical Objects) are the standard ITM structures for data exchange;

- Real experimental controllers are unaware of CPOs and do not support them;

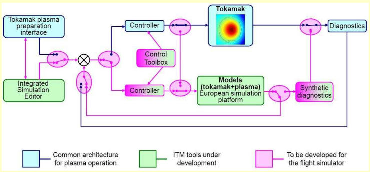
- Multiplexer/Demultiplexer (Mux/Demux) Kepler actors were developed in order to extract control relevant signals from CPOs and to reproduce, as far as possible, the real controller conditions.



Example of use of Mux/Demux actors within the Kepler workflow for ITER gap control.

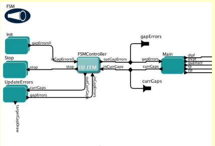
UNDER EXAMINATION :
- Initialization phase of actors within *fc2k*;
- Scicos: management of vectors and advanced blocks (e.g. state-space equation block);
- Universal Access Layer (UAL) access needs to be tested in extreme conditions (problems experienced in the past when using several times the Mux/Demux actors);
- Effective way to debug actors generated by *fc2k*.

FUTURE DEVELOPMENTS :
- Link to the Integrated Simulation Editor (ISE);
- Human Machine Interface (HMI) for experiment set-up;
- Possibility to deployment the controller code in the control hardware (hardware in the loop configuration in the Flight Simulator, see also figure below).


Tokamak plasma preparation interface
Integrated Simulation Editor
Controller
Control Toolbox
Controller
Tokamak
Models (tokamak+plasma) European simulation platform
Synthetic diagnostics
Diagnostics
Common architecture for plasma operation
ITM tools under development
To be developed for the flight simulator

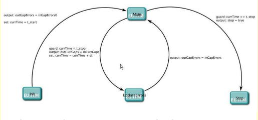
Current status of the plasma discharge Flight Simulator for feedback control simulation and design being developed at CEA.

TO DISCUSS :
- Finite State Machines (FSMs) and Modal Models (FSM generalizations) are Kepler features inherited from Ptolemy;
- They look appealing for feedback control loop design: versatility and easy control design;
- The time may be handled using the *setActions* feature of the transitions.



Example of Modal Model

- The FSM director and the FSMController actor are the relevant blocks;
- The FSMController is a Modal Model actor;
- The composite actors around the FSMController are *refinements* which describe the behaviour of the different states.



Content of the Modal-Model FSMController actor

- The Modal Model actor contains the actual architecture of the FSM (states and transitions)

- FSMs and Modal Models are better supported in Kepler 2.0.0 (e.g. the refinements should not explicitly appear in the workflow, but should be hidden);
- A FSM director can be handled by a PN director.