Minutes of the ITM Meeting on the Implementation of Controllers within the ITM Simulation Platform

March 23, 2010

Participants: F. Imbeaux (FI)
M. Mattei (MM)
S. Brémond (SB)
N. Ravenel (NR)
T. Bolzonella (TB)
C. Konz (CK)
G. Manduchi (GM)
R. Coelho (RC)
A. Soppelsa (AS)

Goal: Discuss options for handling of data and input parameters associated to the implementation of control time loops in the ITM simulation platform (Kepler). Select a single option and lay out a plan for its implementation.

Agenda:

1.) Welcome and Introduction (CK, 5')
2.) Short presentation of options for data and parameter handling (FI, 15') (file ControlToolboxSummary20100323.ppt)
3.) Short presentation on control scheme examples (SB, 15') (file ITM_control_scheme_example_2010_03_22.ppt)
4.) Short presentation of ITM standard for parameter handling (CK, 5') (file tutorial.pdf)
5.) Discussion and selection of single option (all, 45')
6.) Planning of implementation, next steps

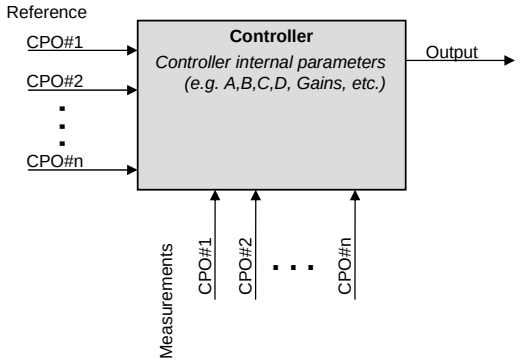
1 Introductory Material (from SB's 2009 report)

Examples for control loops (from SB's report):

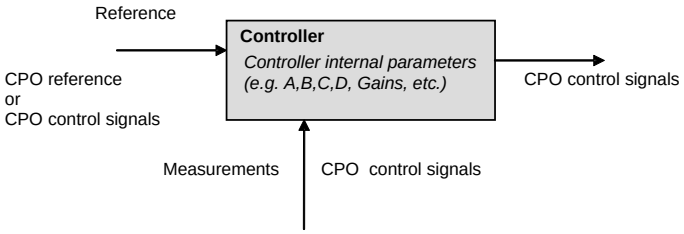
- plasma shape control loop
- plasma vertical position control loop
- plasma density control loop
- plasma burn control loop
- etc.

1.1 Handling of the controllers input / output data

1- The controller actor receives as input all the CPOs needed for its data processing



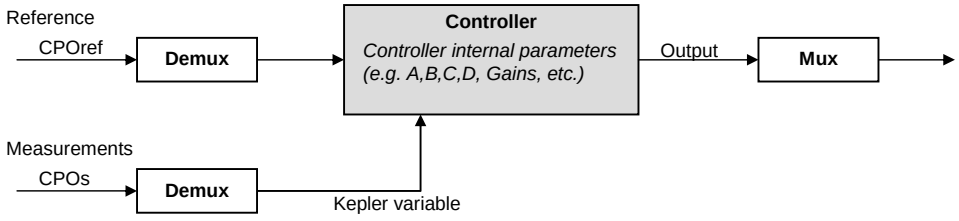
2- Implementation of a new CPO "control signals" which would contain the controllers inputs / outputs



Two sub-options

- The "control signals" CPO contains the entire set of data that may be of interest for all the controllers. The basic idea is here to match the set of data that are generally shared in real experiment real time network between sensors, actuators and controllers.
- Use multiple instances of a generic "control signals" CPO. In that case, the idea is to have a dedicated "control signals" CPO for each particular controller.

3- Implementation of a de-multiplexer/ Multiplexer actor /tool within Kepler to extract from the CPOs the control relevant data



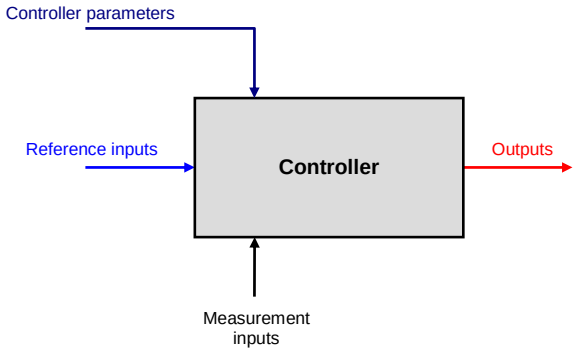
1.2 Handling / Setting of the controller internal parameters

1- Standard linear controllers defined either by a MultiInputMultiOuput Transfer function matrix (including the most widely used simple case of PIDs) or in the general state-space form by the standard ABCD matrix set (covering all the possible linear controllers). Therefore no code specific parameters. Construction directly inside Kepler.

2- More complex controllers (non linear, etc.) with code specific parameters in XML format

Handling of the controller internal parameters.

- to collect these parameters in a “controller” CPO (this holds in particular for standard linear controllers whose structure is well defined, this is actually already included in the present version of the generic “controller CPO”)
- to collect these parameters in a codeparam xml structure to be associated to each controller



2 Decisions

The decisions described here were mostly taken during the meeting on Febuary 24, 2010 (see Summary of Meeting 24022010.doc) but were discussed in detail and refined during this meeting. The results are summarized below.

2.1 Reference Inputs

The former ‘signal CPO’ is now remodelled into a ‘**reference CPO**’ which will contain (constants or time traces or time-dependent arrays for multiple point control), with some self-documentation. The CPO structure will be quite generic, not trying to capture a priori the definition of its detailed content.

Status: not yet implemented in data structures 4.08a
Action: on FI to finalize and implement the schema for the reference CPO

2.2 Measurement Inputs

Individual signals shall be extracted from the standard physics (diagnostics) CPOs in form of Kepler variables before sending them to the control actor. This requires the development of one or multiple corresponding demultiplexing actors (Demux actors).

Status: Demux actors not existent
Action: on Olivier Hoenen to develop required demux actor(s) in Java

2.3 Controller Outputs

Output signals shall be passed as individual Kepler variables explicitly in the workflow and then merged to a CPO by a multiplexer. This requires the development of one or multiple corresponding multiplexer actors (Mux actors).

Status: Mux actors not existent
Action: on Olivier Hoenen to develop required mux actor(s) in Java

2.4 Controller Actor

To benefit from the existing tools and to facilitate future developments, the decision was taken to keep the Kepler and Scicos/Simulink worlds as separate as possible. This means that existing Scicos/Simulink control workflows shall be wrapped as a whole using FC2K to generate an actor inside Kepler with I/O via Multiplexers/Demultiplexers as described above, rather than developing control workflows directly inside Kepler. The development of control algorithms inside Kepler, however, remains always possible.
Scicos and Simulink can generate executable C/C++ modules which can be converted into a Kepler actor using FC2K. No Simulink licenses will be required to run these actors.

Status:

- Scicos control scheme has been coupled to Kepler passing individual signals (test version) but not released
- Simulink control scheme yet to be coupled to Kepler

Action:

- On SB to generate a Kepler actor for a first full Scicos control scheme (e.g. Ip controller)
- On MM to generate a Kepler actor for a first full Simulink control scheme (e.g. Ip controller)

2.5 FC2K

The control actor is generated by FC2K from a source C/C++ module generated by Scicos/Simulink. A number of developments on FC2K are required:

- FC2K but be able to wrap in a no CPO I/O case (all I/O are Kepler variables)
- The FC2K interface shall allow naming the individual I/O signals (which will appear explicitly in the workflow edition screen)
- FC2K must allow sending a string to the control code information on the status of the actor: “init”, “fire”, “wrapup”.

Status: no progress
Action: on Philippe Huynh to modify FC2K according to the above mentioned requirements

2.6 Code Parameters

Controller parameters: we took the decision to parameterise the controller using the standard approach for code-specific parameters, i.e. use the « codeparam » xml structure. In case some parameters of the controller depend on the plasma state (e.g. exception handling), it is recommended to build within the control diagram a super-controller handling decision-making on the type of control to apply. One can even foresee to use Kepler variables to inform the actor on workflow status if needed (standard functionality of FC2K).
The codeparam structure contains a string array in XML format which requires parsing by an appropriate parser and an assignment function. Mathias Hoffmann (IMP4) has provided a C parser in xmllib under Gforge which is being implemented by the EQUINOX/CEDRES++ and TYR teams. The parser builds on an W3C XML schema for the code. Mathias Hoffmann will provide a tool to auto-generate the assignment function for C/C++.

Status: C/C++ parser for XML code parameters available
Action: on Mathias Hoffmann to provide the auto-generation tool for assignment functions
on SB and MM to contact CK for development of W3C XML schemas for control parameters

2.7 Time Integration

The current proposal foresees an explicit time step in the control scheme. Some controllers may require an implicit time step though such that the status of the control variables at the previous time step need to be saved. The memory of a Kepler actor stays alive during the entire workflow runtime such that using static or save keywords for the appropriate variables in the control module can solve this problem.

Status: not started
Action: on SB and MM to investigate the feasibility of implicit time steps using static variables inside Kepler

2.8 Tokamak Simulator

The measurement signals of a free boundary equilibrium calculation with feedback control are currently foreseen to result from the output of a free boundary equilibrium module inside Kepler (for the first phase). The free boundary equilibrium modules within IMP12 are CEDRES++ (SB) and CREATE-NL (MM). These modules need to be adapted to the ITM data structures and integrated in the Kepler workflow environment.

Status:

- CEDRES++ ready, but code parameters missing, no actor, not tested
- CREATE-NL on the gateway, code parameters missing (?), no actor, not tested

Action:

- On SB to provide a fully integrated CEDRES++ module for Kepler
- On MM to provide a fully integrated CREATE-NL module for Kepler

2.9 Test Cases

The decision was taken to start with a simple Ip control scheme to test the integrated Kepler workflow.

Status: not yet started
Action: on SB and MM to provide a Ip control scheme test case within Kepler

2.10 Documentation

Both the control schemes and actors as well as the tokamak simulators need to be documented. The documentation shall include a user documentation which explains how to run the workflow and each module as well as a developer documentation which describes the physics and numerical details of the modules and workflows.

Status: not yet started / unknown
Action:

- On SB to provide documentation on CEDRES++
- On MM to provide documentation on CREATE-NL
- On SB and MM to provide documentation on the control actors

2.11 Dissemination

The discussed requirements and needs may be incomplete with respect to the entire fusion community. To properly exploit the ITM tools, it is needed to disseminate the needs outside the ITM too as well as to propagate information and news within and outside the ITM.

Status: not yet started
Action: on RC to contact experts outside the ITM for input on control requirements and expertise
On TB to facilitate flow of information within and outside the ITM
On SB and MM to stay in contact for documentation of control efforts

2.12 Other Issues

Action: on FI to instruct GB on how to use FC2K
Action: on FI to remove controller CPO from data structure

3 Time Chart