Control integration

Federico Felici

Swiss Plasma Center

EPFL, Lausanne

Switzerland

11th ITER International School

San Diego, CA, USA 25-29 July 2022

SWISS PLASMA CENTER

EPFL

Control integration

Federico Felici

Swiss Plasma Center

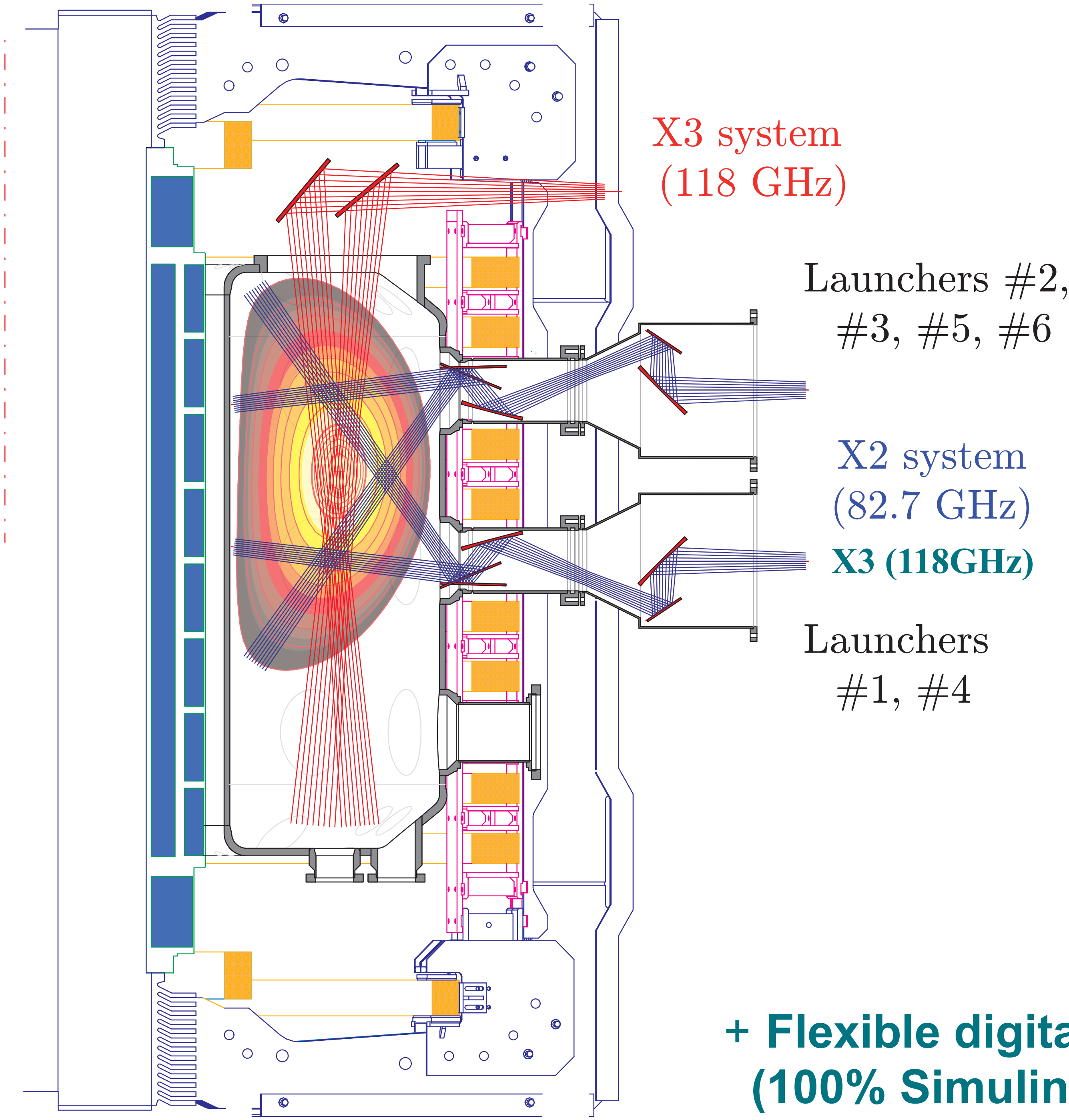
EPFL, Lausanne

Switzerland

11th ITER International School
San Diego, CA, USA 25-29 July 2022

SWISS
PLASMA
CENTER

TCV: Tokamak à Configuration Variable

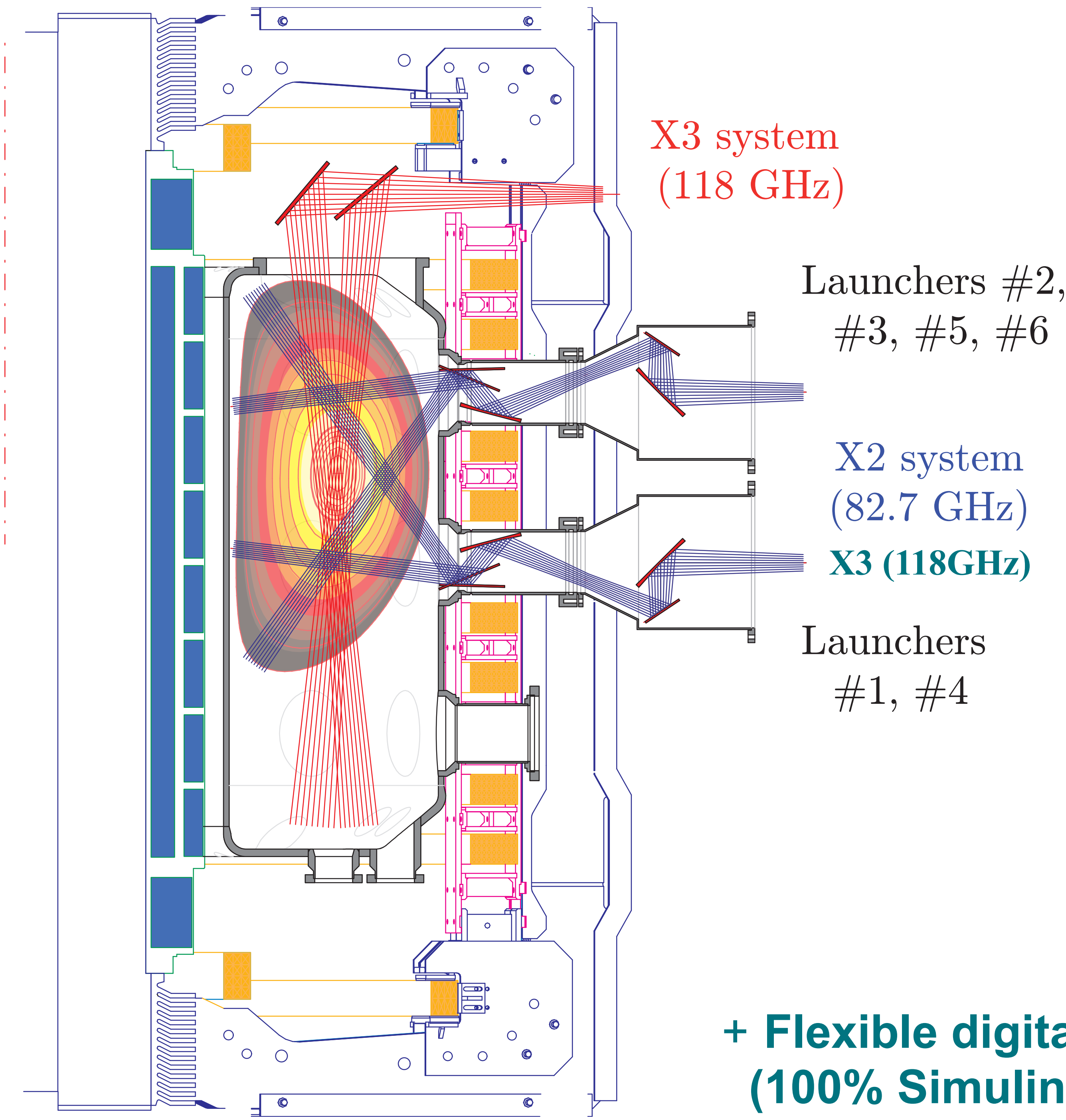


+ Flexible digital control system (100% Simulink-programmable)

B_t = 1.5T, I_p<1MA, a=0.25m R=0.88m κ<2.8

19 separately controllable PF coils

TCV: Tokamak à Configuration Variable



- **Shape effects on plasma**
  - **Core, edge studies**
  - **Negative triangularity, snowflakes**
- **NBI and ECRH heating**
- **Plasma control research**
  - **Formation of advanced plasma shapes & vertical control (1990s-…)**
  - **MHD control: NTMs, Sawteeth (2000-…)**
  - **State estimation & Profile control (2010-…)**
  - **Integrated control (2015-…)**

+ Flexible digital control system (100% Simulink-programmable)

B_t = 1.5T, I_p<1MA, a=0.25m R=0.88m κ<2.8
19 separately controllable PF coils

Outline

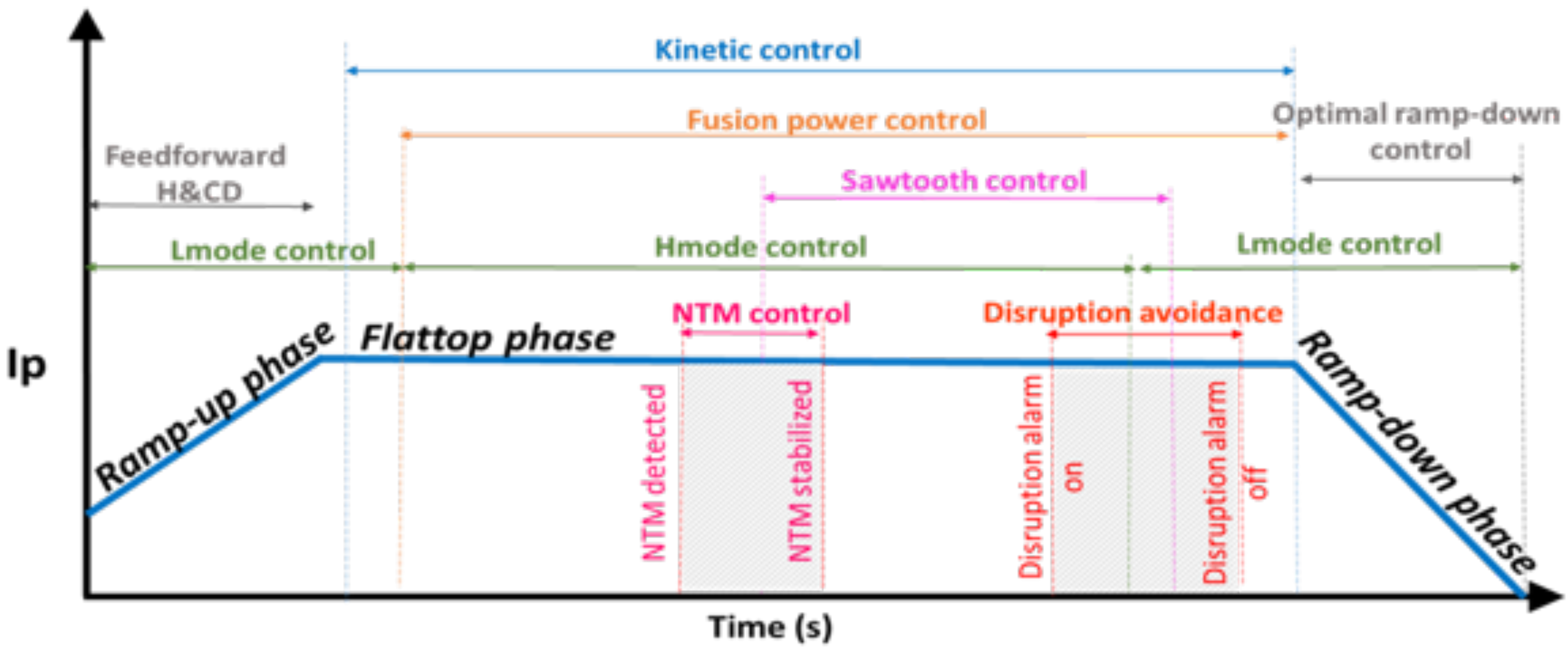
Part 1: Integrated control: architectures and some examples of solutions

Part 2: Software engineering aspects of plasma control integration

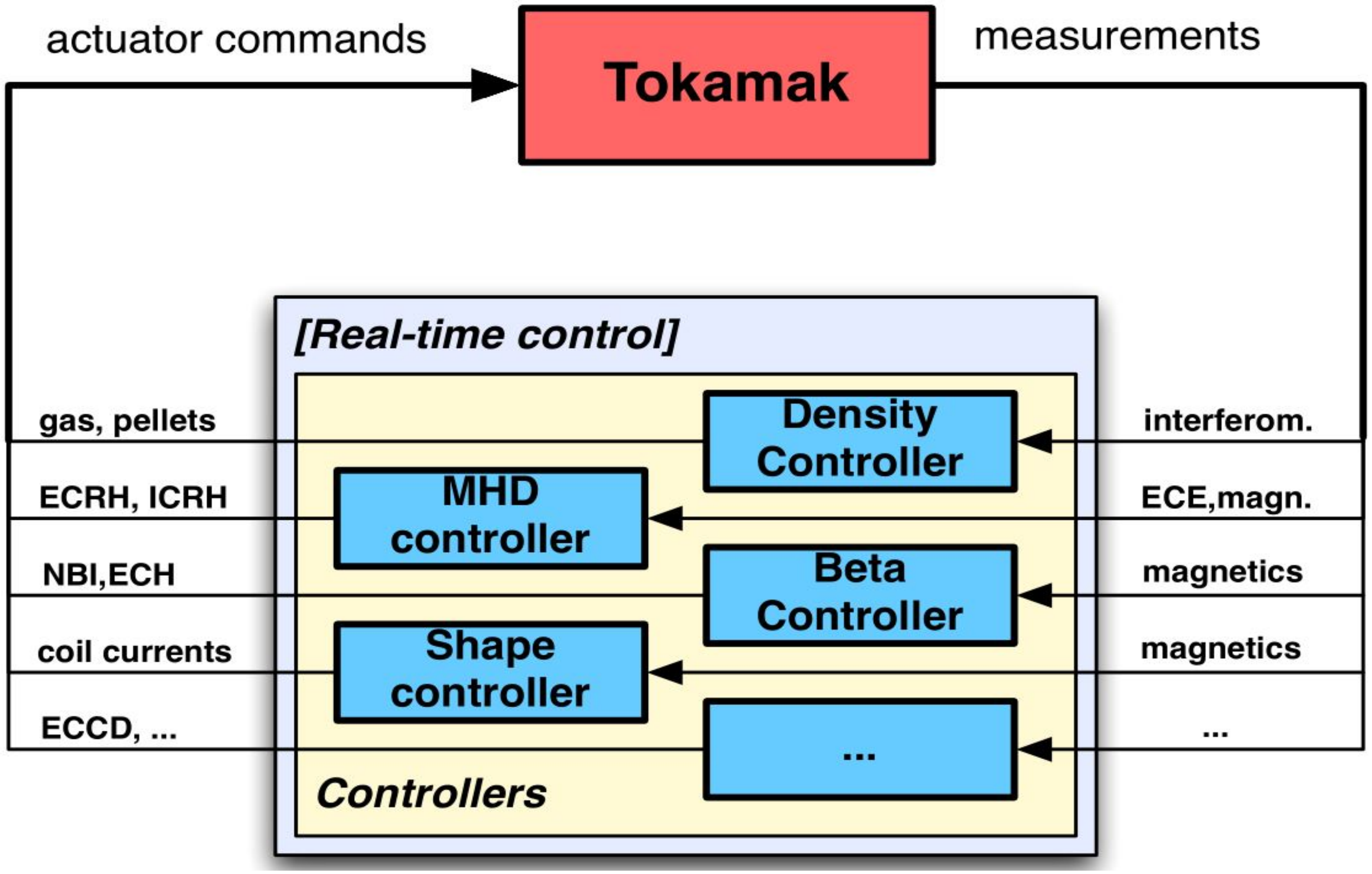
Integrated control: key issues and some examples of solutions

Motivation: future tokamak reactors will need to fulfil multiple control tasks with a limited set of actuators

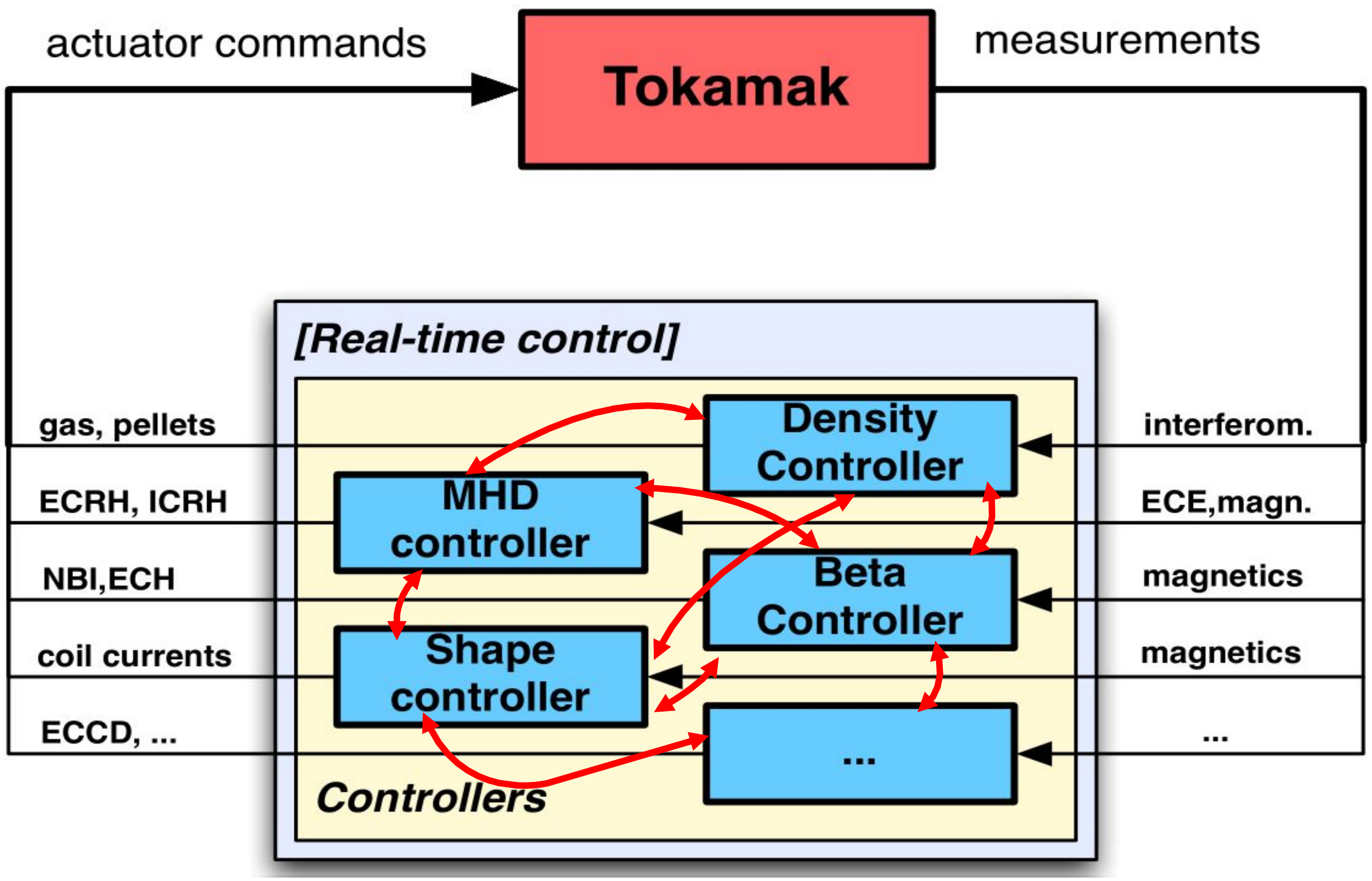
- **New control challenges:**
  - **Simultaneous execution of several (complex) control tasks with scarce actuators.**
  - **Real-time prioritisation of these tasks based on evolving plasma state/events.**
  - **Real-time automated assignment of scarce actuators to fulfil various tasks.**



Traditional control architectures with separate controllers are not sufficient for next-generation tokamaks



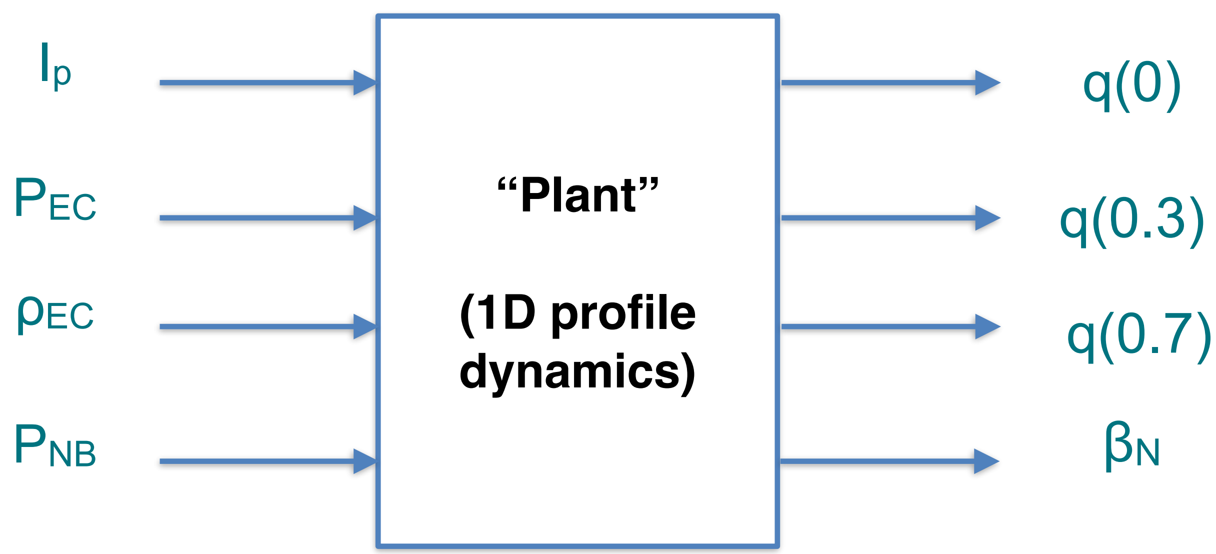
Traditional control architectures with separate controllers are not sufficient for next-generation tokamaks



- **Issues for integrated control:**
  - **Interaction/competition between controllers**
  - **Time-varying priorities for control**
  - **Time-varying actuator availability**
  - **Response to off-normal events**

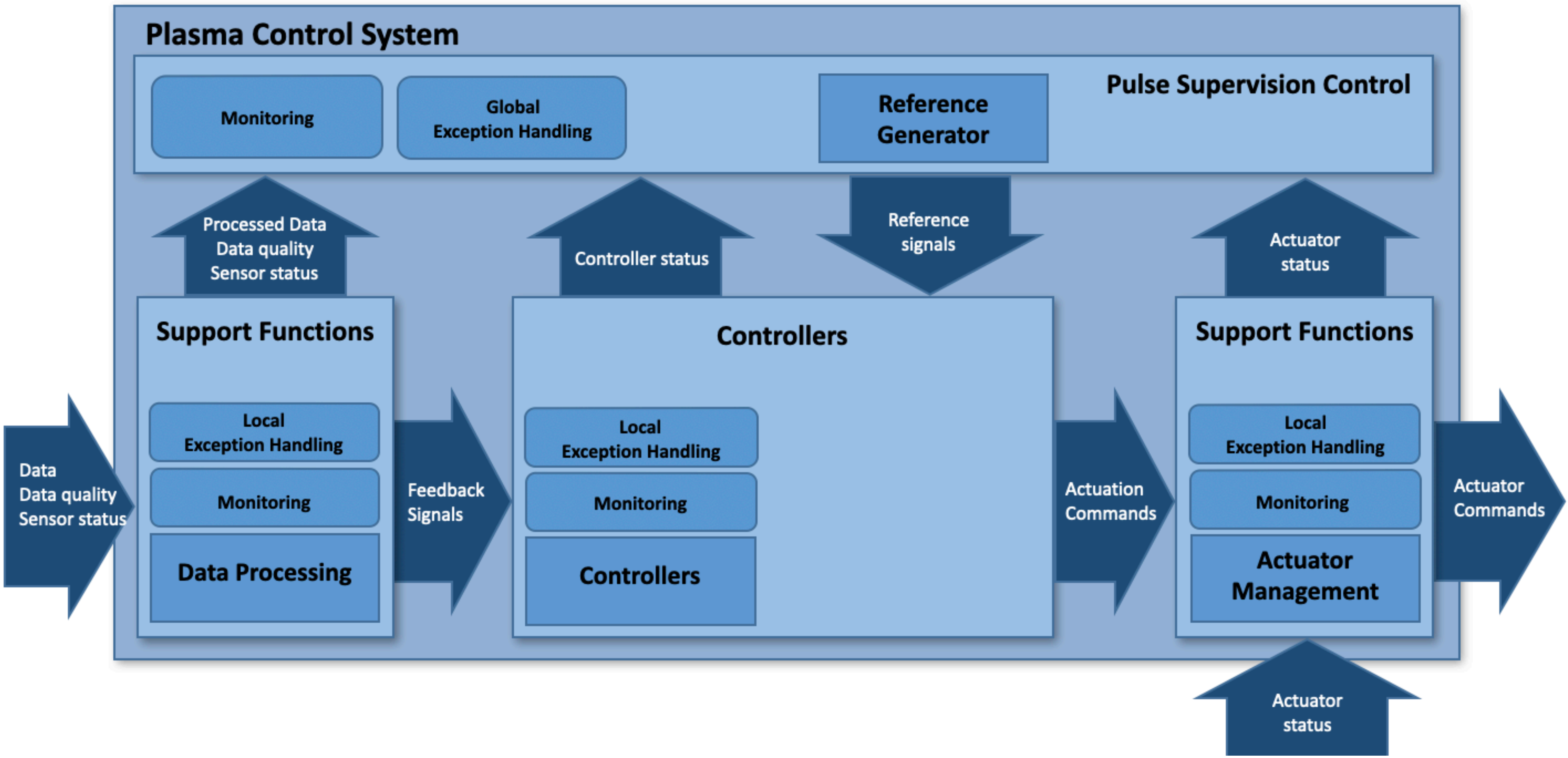
Control integration via Multivariable Controller Design

- **Multivariable (MIMO) controller design**
  - **Design one controller that takes interactions into account explicitly.**
  - **Necessary when problems are strongly coupled dynamically.**
  - **Quickly becomes intractable as size of system increases.**
  - **Examples:**
    - **Shape control (many coils -> many shape control parameters) [DeTommasi lecture, Tue]**
    - **q profile (+betaN) control (many control points -> several actuators) [Schuster lecture, Wed]**



But: we can not (yet) make one single controller for everything - we will have several separate controllers

ITER PCS architecture design: Supervision layer, controllers, support functions



Supervisory control architectures under study in existing tokamaks

- **DIII-D / KSTAR / EAST:**
  - **Finite state Off Normal Fault Response (ONFR) [1]**
- **ASDEX-Upgrade / ITER:**
  - **Local/Global exception handling [2],**
- **TCV:**
  - **Supervision Actuator Management and Off-Normal Event handling (SAMONE) [3]**
    - **Control ‘task’ based approach, described in more details next**

[1] N. W. Eidietis, et al, Nucl. Fusion, vol. 58, no. 5, p. 056023, May (2018).
[2] W. Treutterer et al, Fus. Eng Des. 117, (2017)
[3] Vu IEEE TNS (2021) and references therein

Introduction to the task-based approach

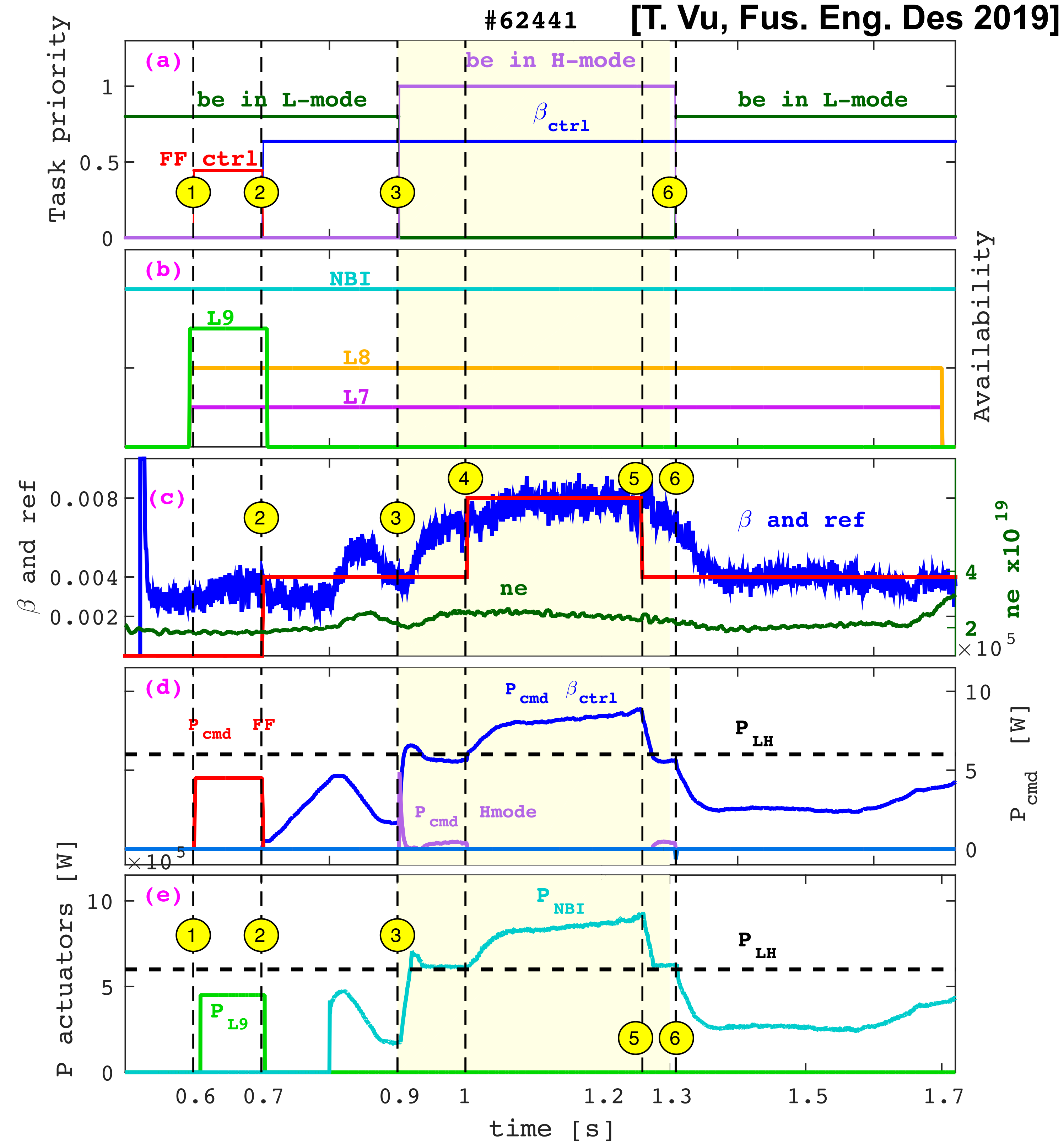
- **Control tasks:**
  - Tokamak independent, general formulation for any tokamak
  - Represents ‘something’ that needs to be done by the control system
- **Separate responsibilities for task handling:**
  - A supervisor decides control task priorities based on plasma state.
  - A set of controllers execute one or more control tasks: receiving plasma state information and compute actuator requests
  - An actuator manager decides allocation of resources for prioritized control tasks

Examples of control tasks:

- 3/1 NTM preemption
- 2/1 NTM stabilization
- track q profile reference
- track β reference
- track I_p reference
- track V_{loop} reference
- go to H mode
- stay in H mode

Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **"Be in L mode"**
  - **"Be in H mode"**



Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

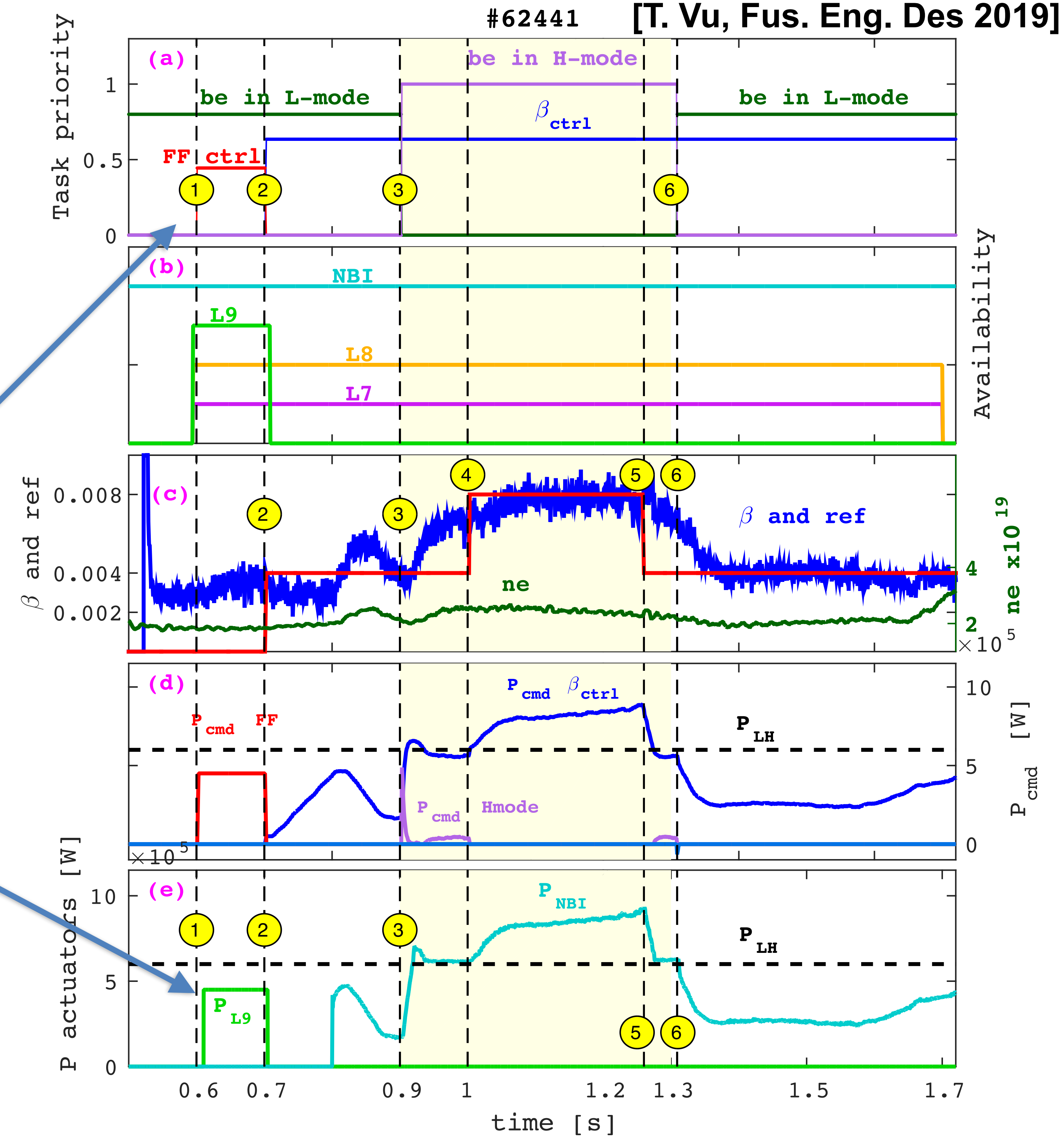
Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **“Be in L mode”**
  - **“Be in H mode”**

1 **Feedforward control task:**
Apply central heating
Actuator priority: L7,8,9, NBI

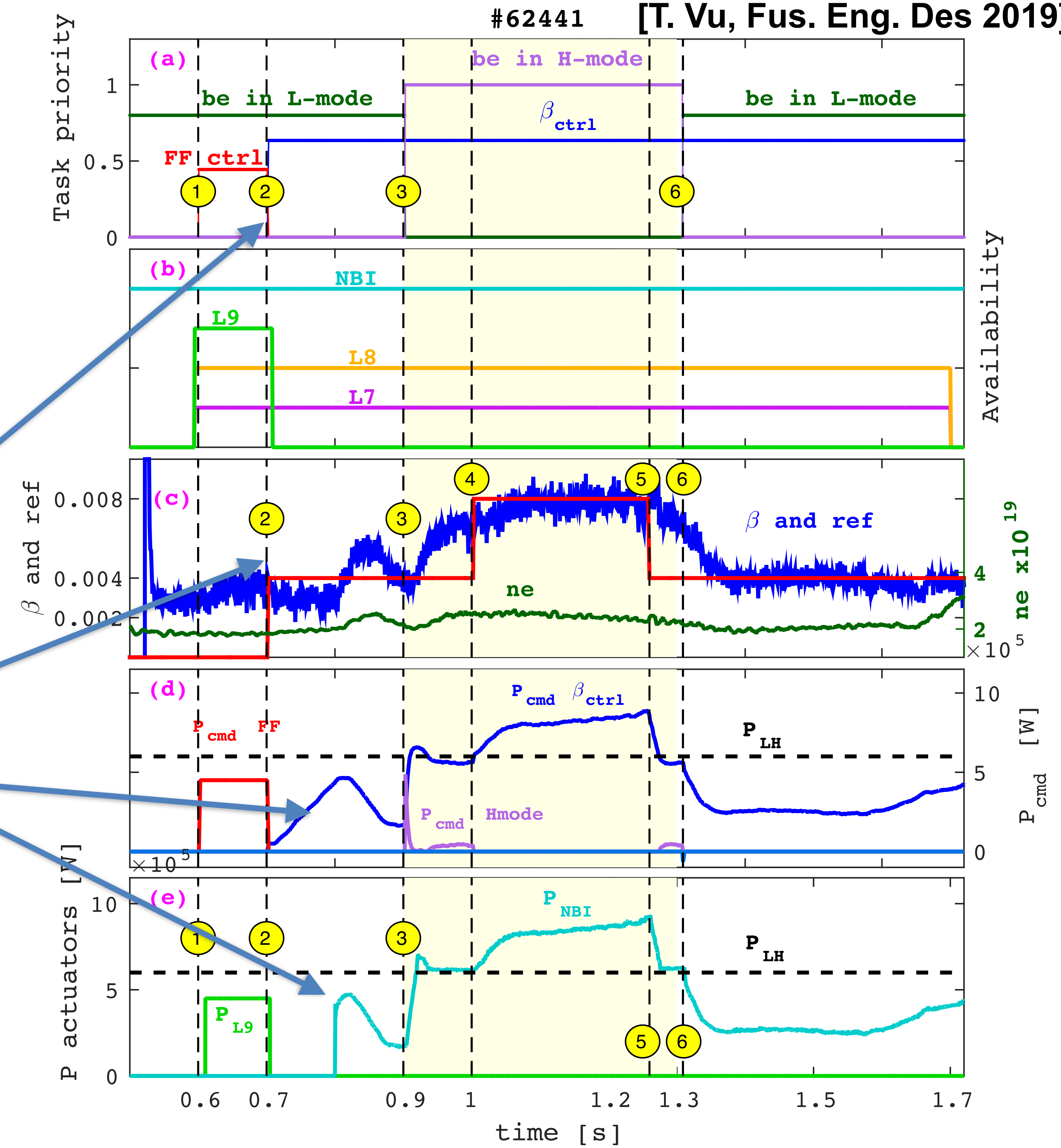
Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

#62441 [T. Vu, Fus. Eng. Des 2019]



Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **“Be in L mode”**
  - **“Be in H mode”**



2

β control task:
Track reference β
Actuator priority: NBI
NBI power increase requested
(NBI switches on late due to technical issues)

Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

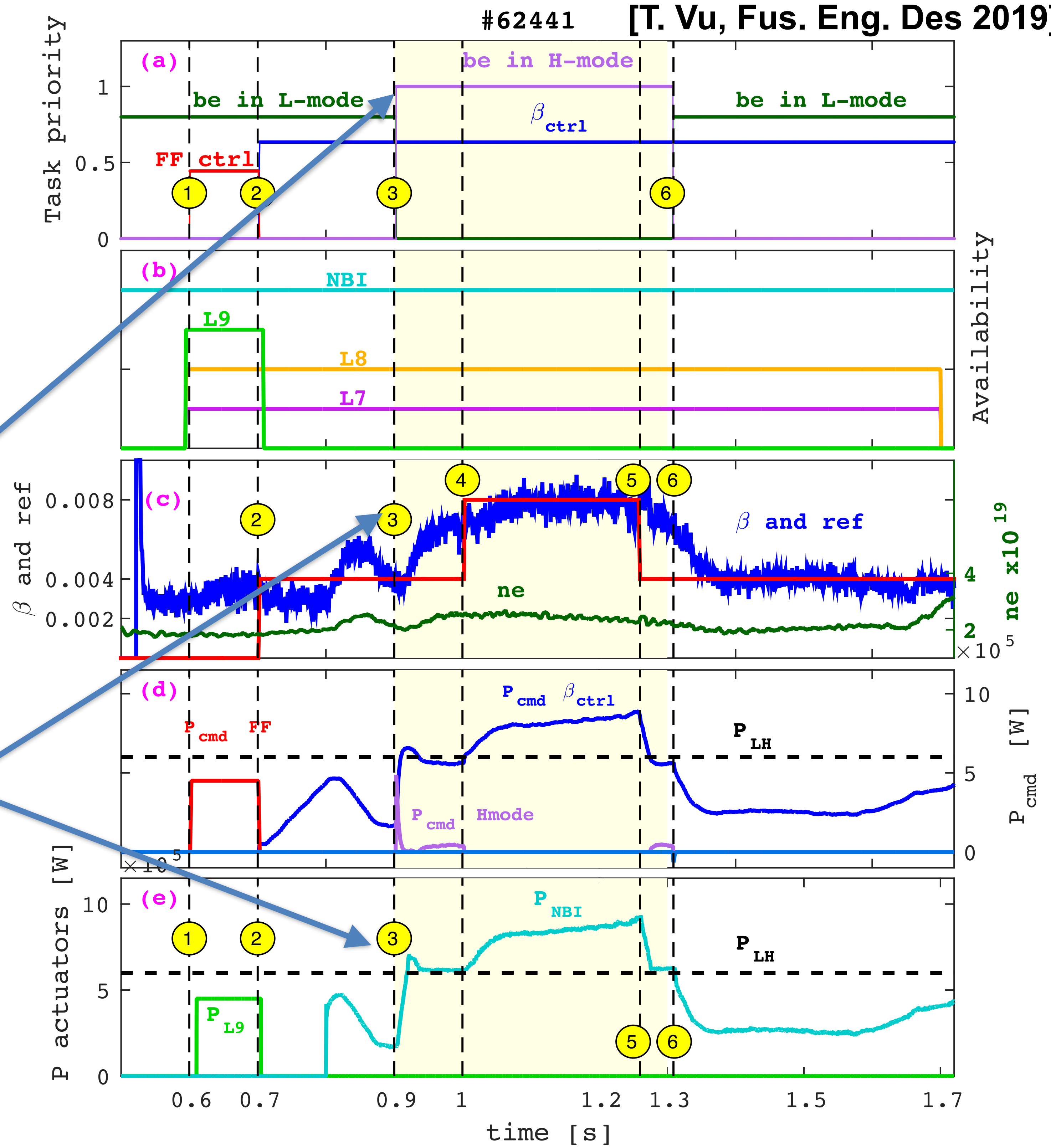
Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **“Be in L mode”**
  - **“Be in H mode”**

3

Be in H mode
Highest priority task
Overrides power request to P>P_{LH}
β reference no longer tracked

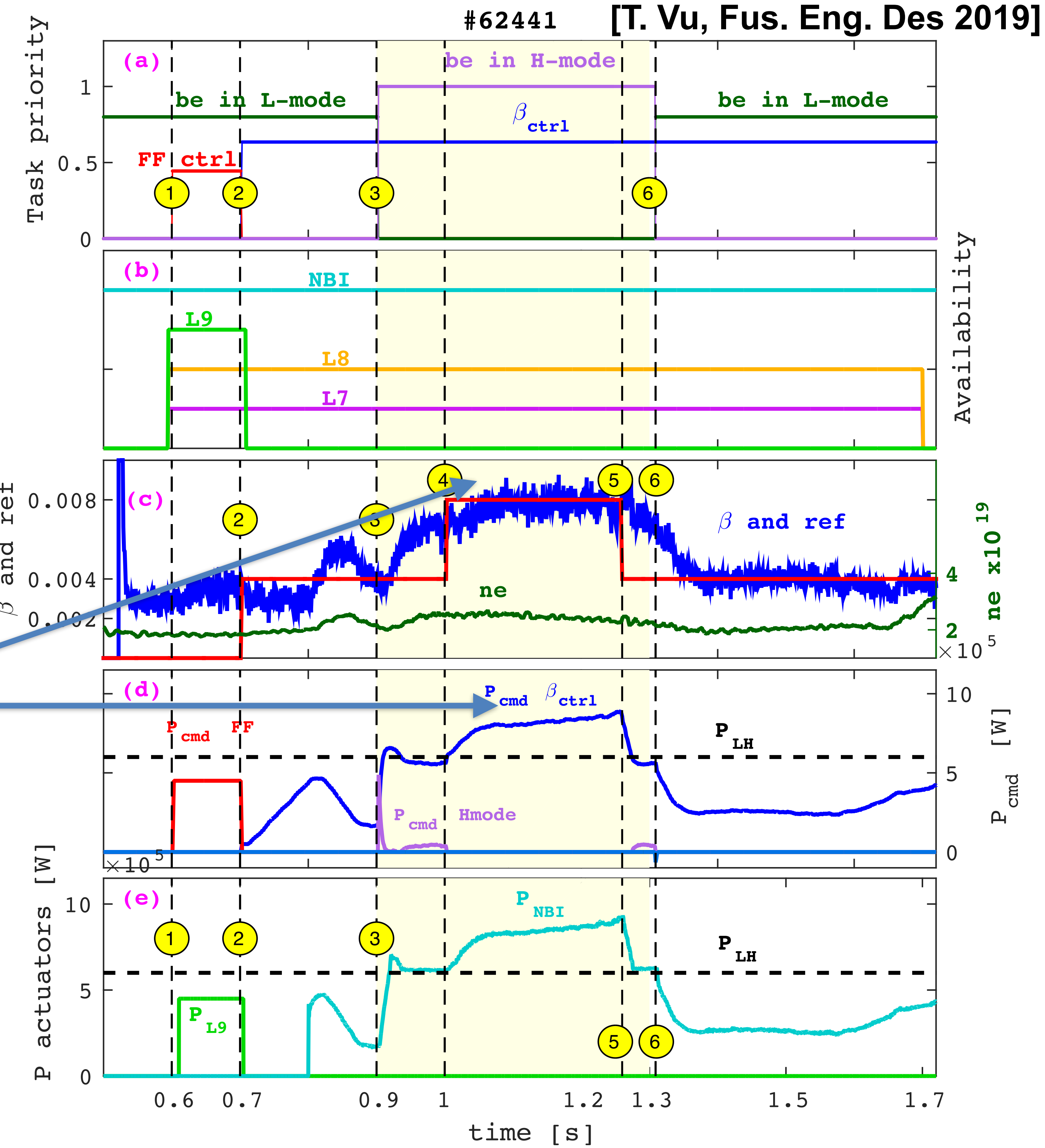
Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task



Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **"Be in L mode"**
  - **"Be in H mode"**

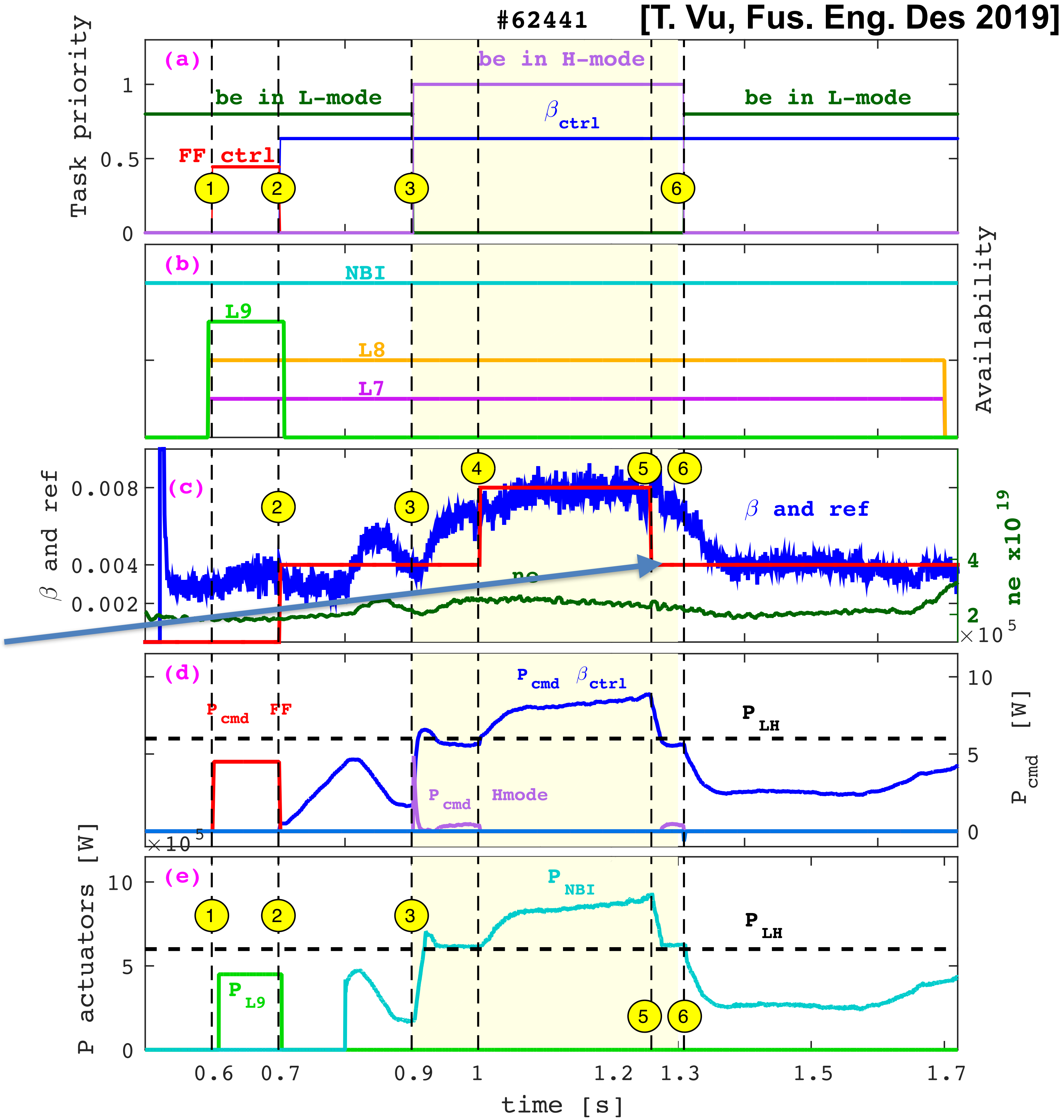
4 **β control reference increase**
β reference can be tracked
since $P_{request}$>P_{LH}



Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **“Be in L mode”**
  - **“Be in H mode”**



β control reference lowered

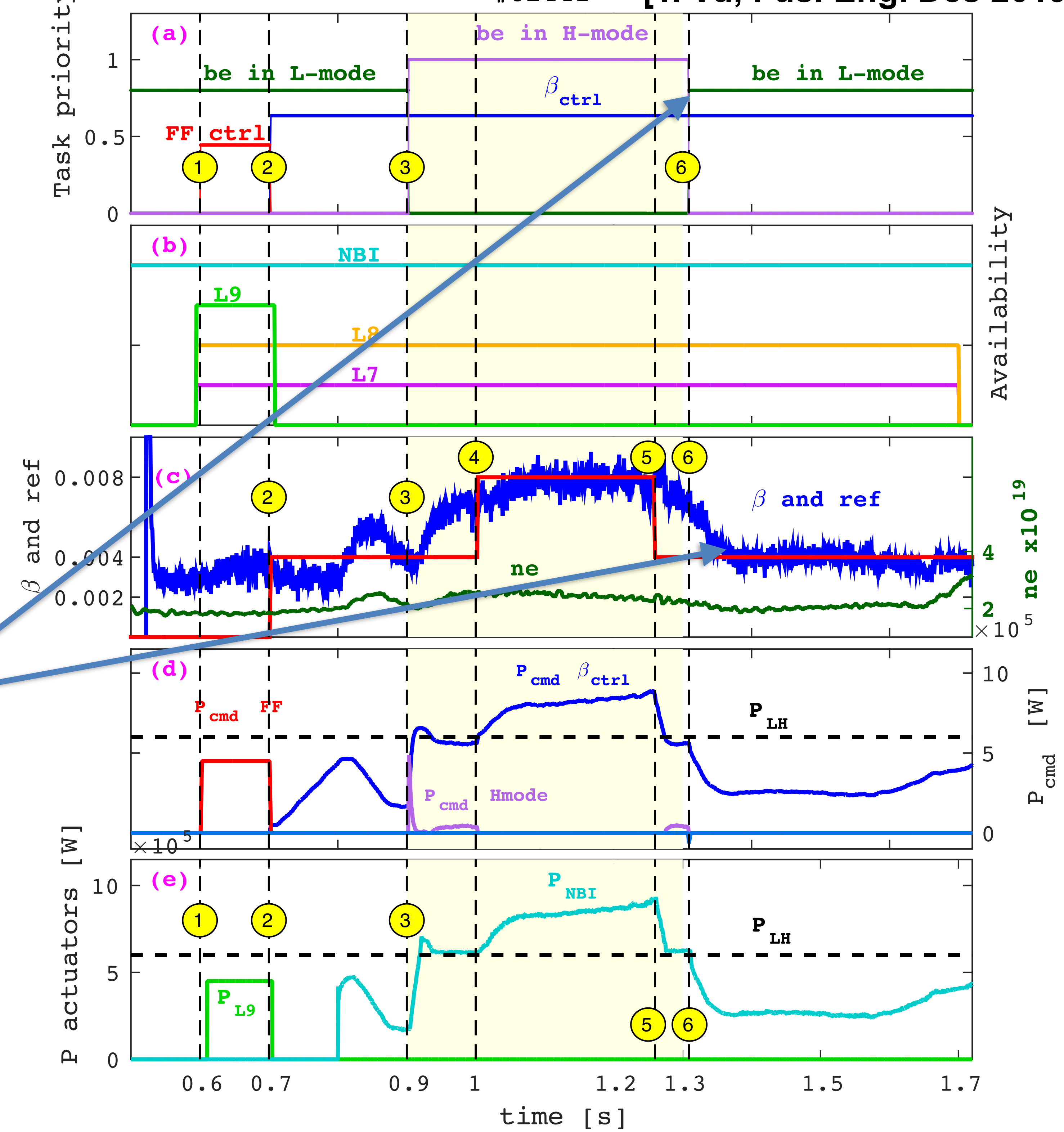
5 Not tracked due to H-mode request

Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

Example of task-based control on TCV: Simultaneous H-mode and β control

- **RT allocation of 4 actuators**
  - **1x P_{NBI}, 3x P_{ECRH} (L7,8,9)**
- **4 prioritized tasks:**
  - **Feedforward power**
  - **β control**
  - **“Be in L mode”**
  - **“Be in H mode”**

#62441 [T. Vu, Fus. Eng. Des 2019]

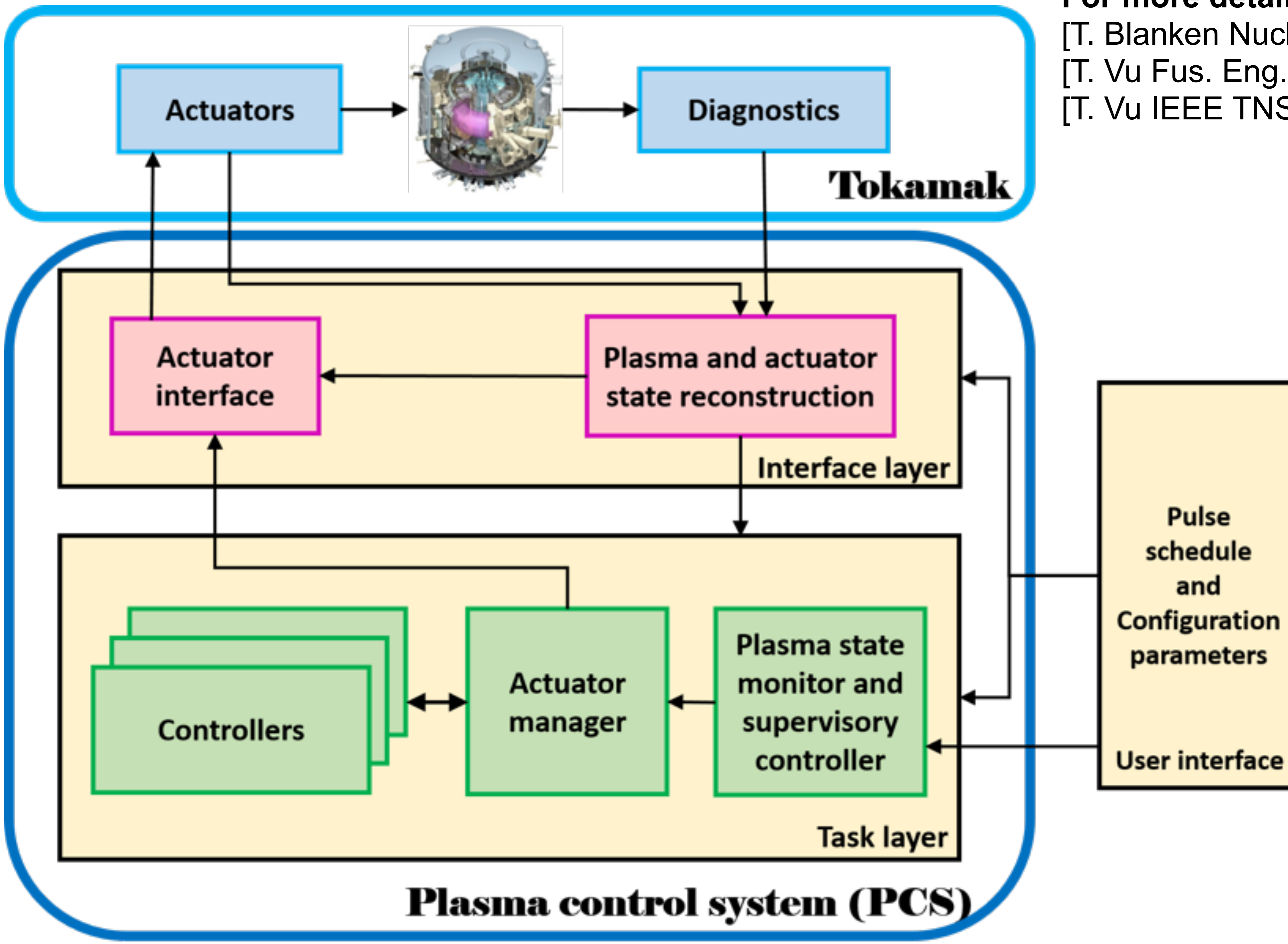


6

Switch to “Be in L mode” task
Low β reference
successfully tracked

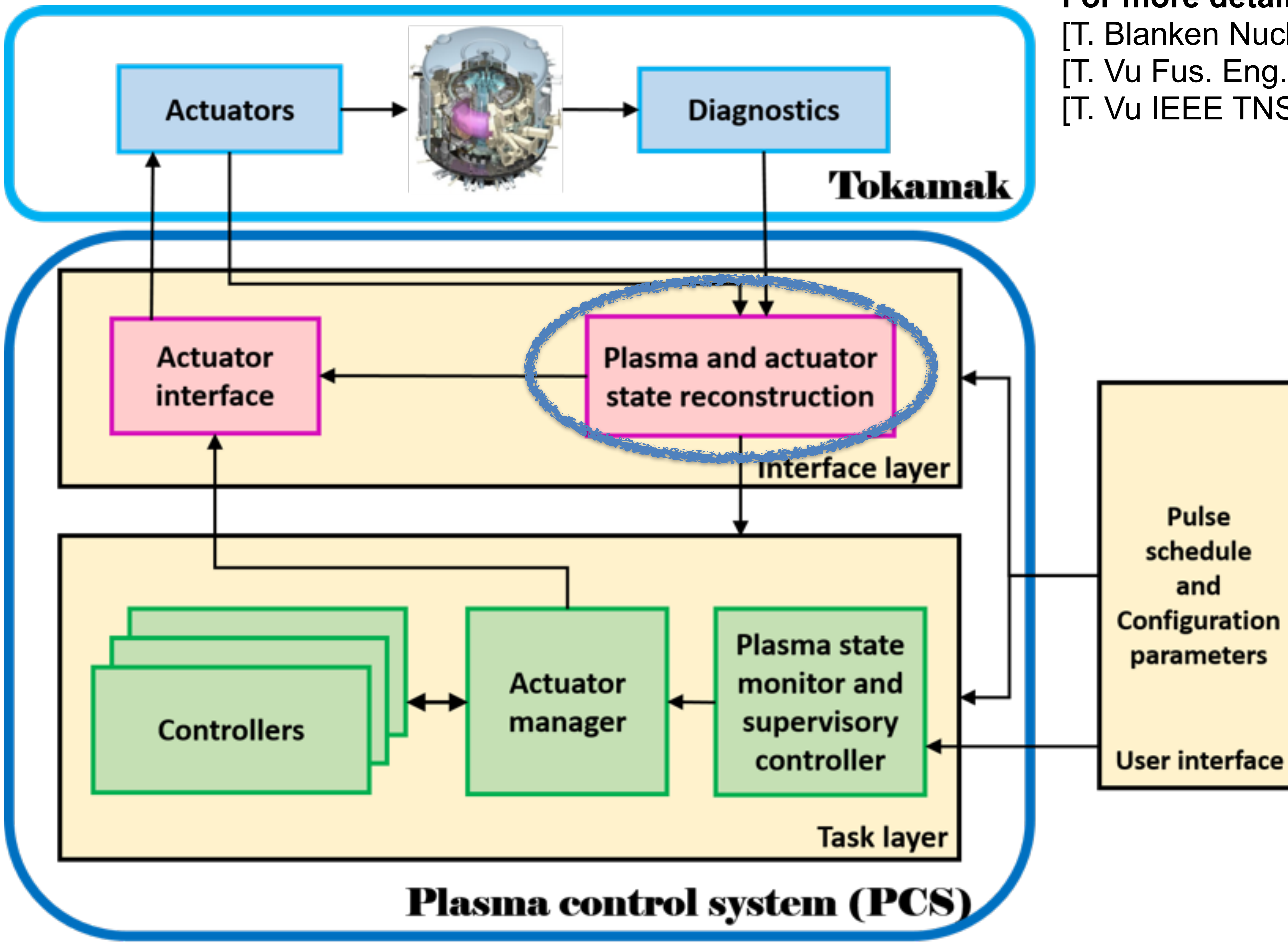
Pre-programmed:
Control task priorities
Preferred actuators per task
Control references, gains per task

Architecture of task-based PCS: separation between specific interface layer and generic task layer



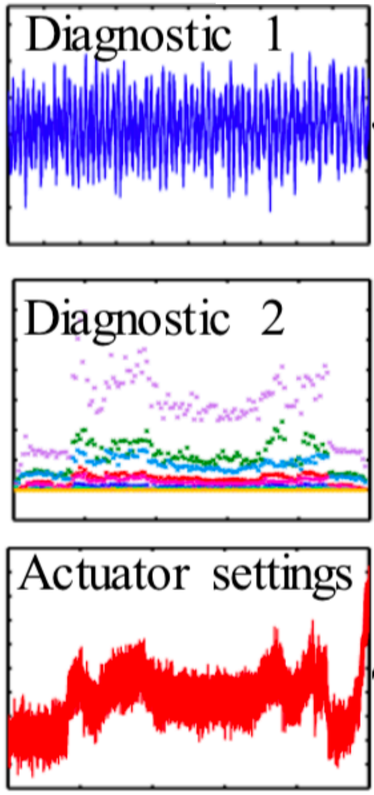
For more details:
[T. Blanken Nucl. Fus 2019]
[T. Vu Fus. Eng. Des 2019]
[T. Vu IEEE TNS 2021]

Architecture of task-based PCS: separation between specific interface layer and generic task layer



For more details:
[T. Blanken Nucl. Fus 2019]
[T. Vu Fus. Eng. Des 2019]
[T. Vu IEEE TNS 2021]

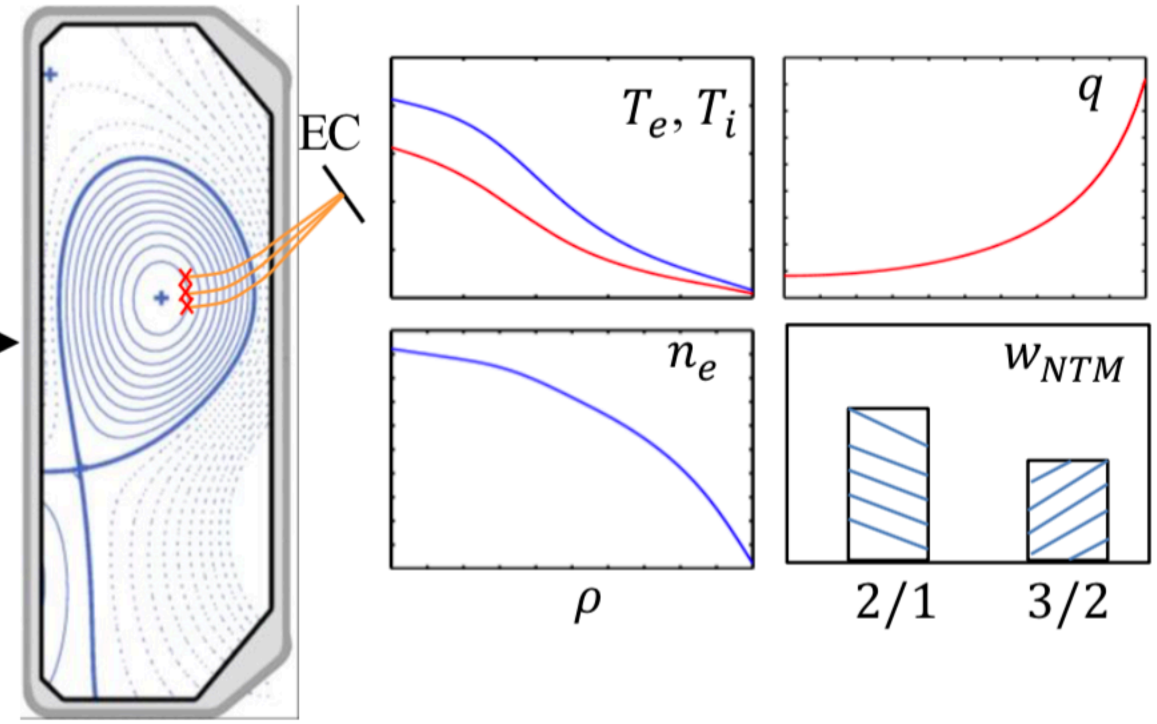
Plasma state reconstruction: combine specific diagnostic signals into to generic tokamak state descriptions



Suite of interconnected codes:

RT SVD (MHD analysis) [1]
RAPTOR (Te, q profile estimation + prediction) [2]
RAPDENS (density profile) [7]
LIUQE (equil. reconstr) [3]
TORBEAM (ECRH deposition) [4,5]
RABBIT (NBI deposition) [6]

+ Various event detectors (ELMs, Sawteeth, LH transition..)



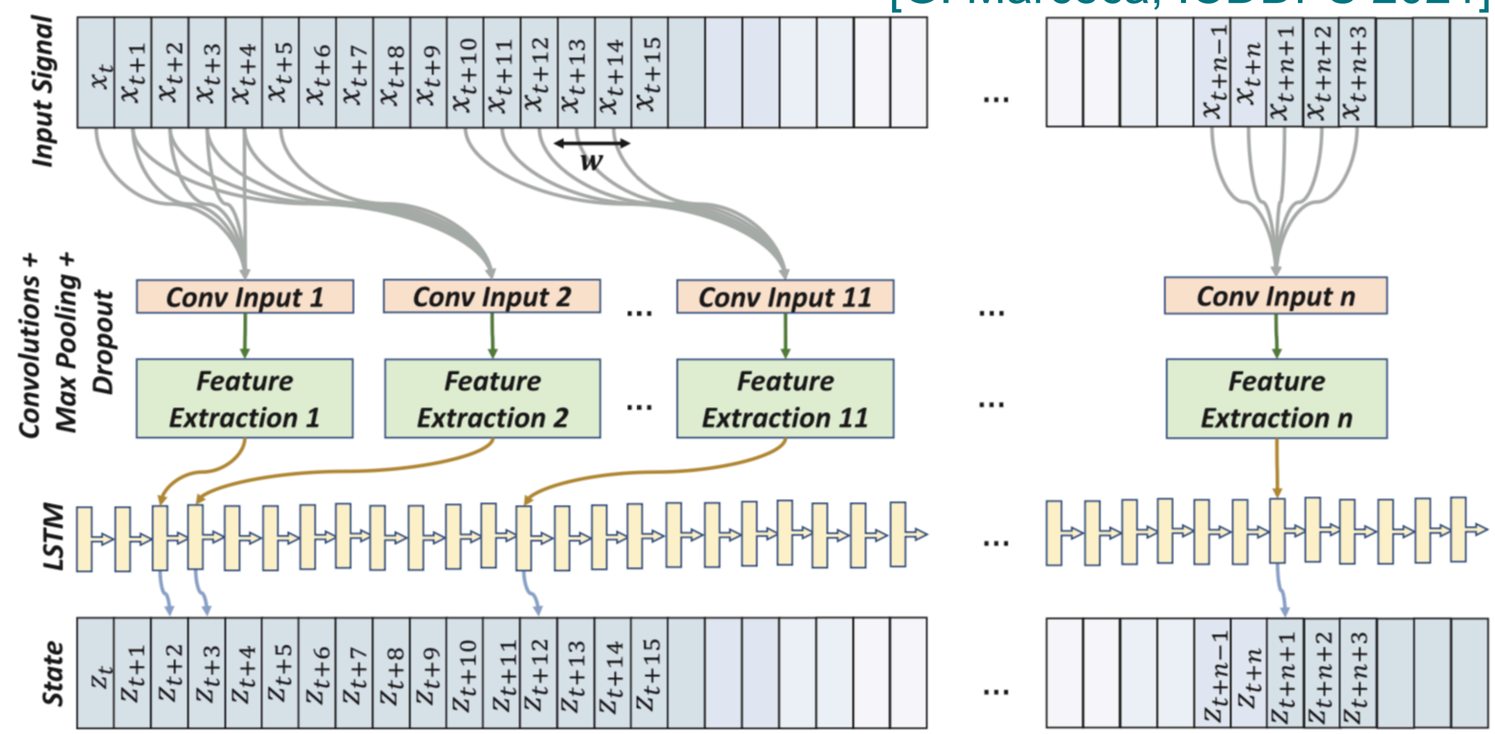
RT diagnostics & actuator info. → **Generic plasma and actuator states**

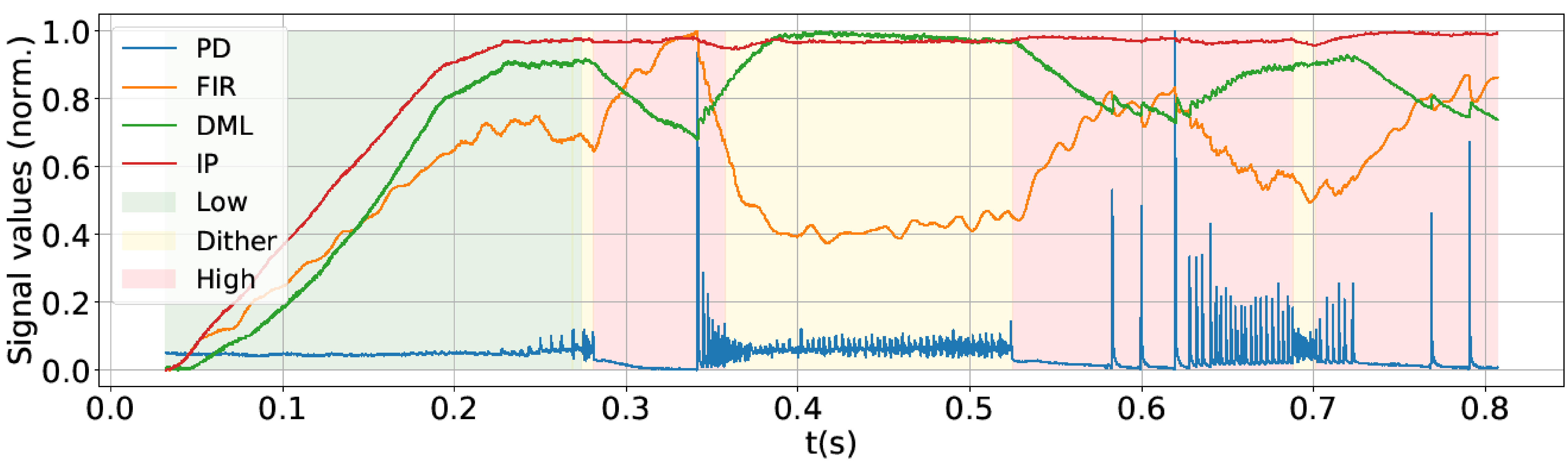
[1] C. Galperti et al., IEEE Trans. Nucl. Science 64 (2017) 1446-1454
[2] F. Felici et al., 26th IAEA FEC, 2016 [3] J-M. Moret et al, FED 2015
[4] E. Poli et al., CPC 225 (2018) 36-46 [5] M. Reich et al., FED 100 (2015) 73-80
[6] M. Weiland et al., 27th IAEA FEC (TH/6-3), 2018
[7] T. Blanken al, FED 2019

Event detection example: Real-time plasma confinement state detector using Deep Learning

- **Combines convolutional layers (CNN) + LSTM**
- **Based on [Matos, NF 2020]**

[G. Marceca, ICDDPS 2021]

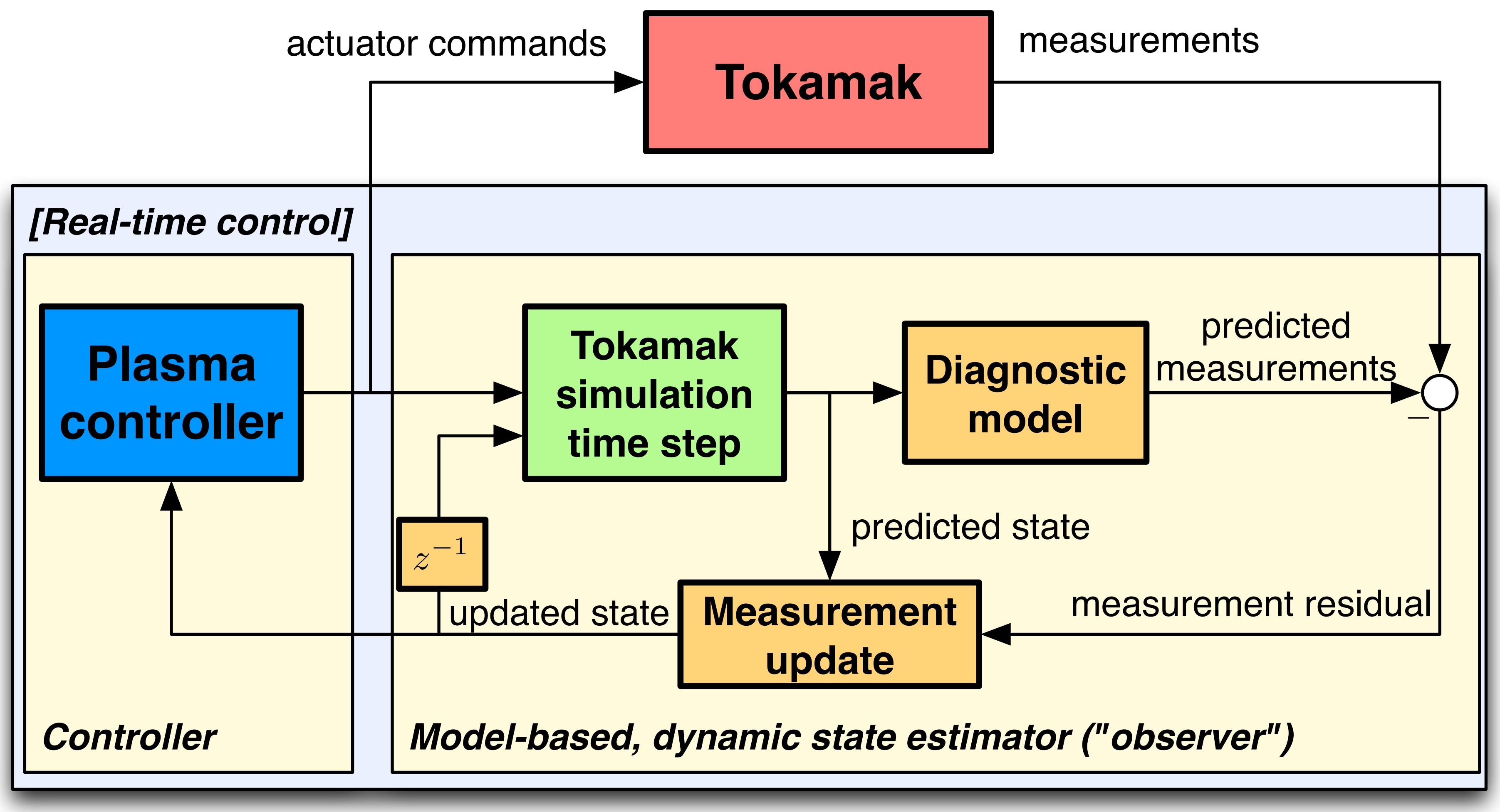




Model-based, dynamic state observer: merge model prediction and diagnostic measurements

- **Amounts to performing a *real-time simulation* of the plasma time evolution, with *corrections from measurements***
  - **Known in control literature as *dynamic state observer*, or *Kalman filter.***
  - **Widely used in robotics, image processing, broad literature exists**

e.g. [Kailath, *Linear Estimation*, Prentice Hall (2000)]



Nonlinear observers: need linearization around nonlinear trajectory

- **Nonlinear model**

$$x_{k+1} = f(x_k, u_k)$$
$$y_k = h(x_k)$$

- **A well-known observer for nonlinear systems is the Extended Kalman Filter (EKF).**
  - **Evolve state error covariance matrix S_k together with state x_k.**
  - **Matrices R_k (sensor noise covariance) and Q_k (process noise cov.) to be tuned**

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + L_k[y_k - h(\hat{x}_{k|k-1})] \quad \text{state meas. update}$$
$$\hat{x}_{k+1|k} = f_k(\hat{x}_{k|k}, u_k) \quad \text{Predicted state}$$
$$L_k = S_{k|k-1}H_k^T\Omega_k^{-1}, \Omega_k = H_kS_{k|k-1}H_k^T + R_k \quad \text{Kalman gain}$$
$$S_{k|k} = (I - S_{k|k-1}H_k^T\Omega_k^{-1}H_k)S_{k|k-1} \quad \text{covariance meas. update}$$
$$S_{k+1|k} = F_kS_{k|k}F_k^T + G_kQ_kG_k^T \quad \text{covariance time update}$$

Need Jacobians

$$F_k = \frac{\partial f}{\partial x_k}$$
$$G_k = \frac{\partial f}{\partial u_k}$$
$$H_k = \frac{\partial h}{\partial x_k}$$

- **Other nonlinear filtering methods exist e.g. Unscented KF …**

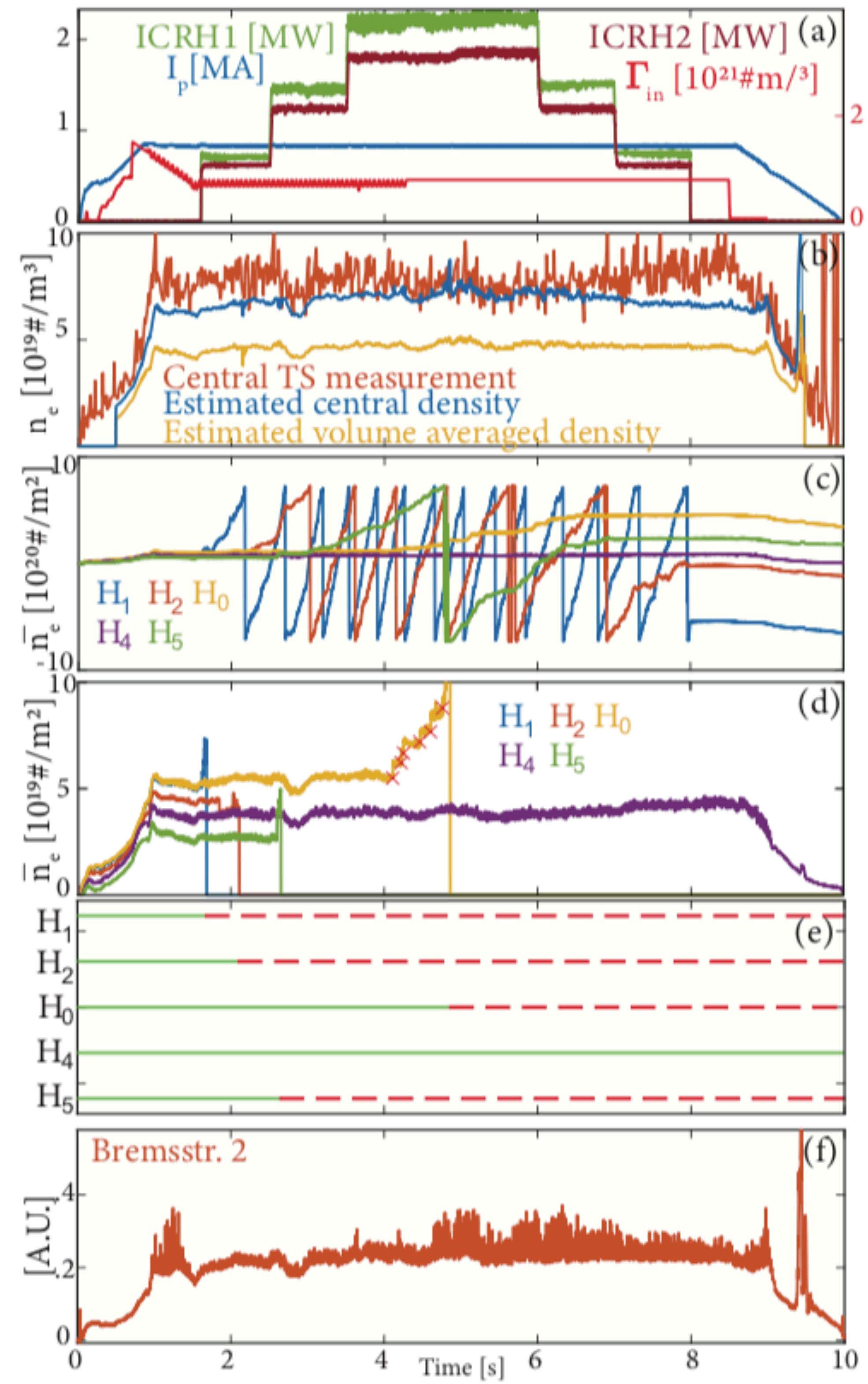
‘Kalman gain’ reflects confidence in models vs measurements

- If the system is linear, the Kalman Filter is the *optimal filter*
  - Gives the smallest state error covariance w.r.t. any other filter
- Though all systems are nonlinear, we can still use KF
- Useful features:
  - Model-based filter to remove diagnostic noise and obtain estimated *states*
    - (e.g. velocity from position measurements)
  - Can treat uncertainties in model parameters
    - Can represented as fictitious process noise entering the state equation
    - Can define *augmented state = [plasma state; model parameters] to be estimated*
  - Diagnostic fault detection
    - Single diagnostic residual increases → detect fault → eliminate from update law.
  - Anomaly detection
    - Sudden deviation of a set of diagnostic measurements from model → trigger disruption mitigation.

Example: real-time density profile reconstruction on ASDEX-Upgrade using state observer

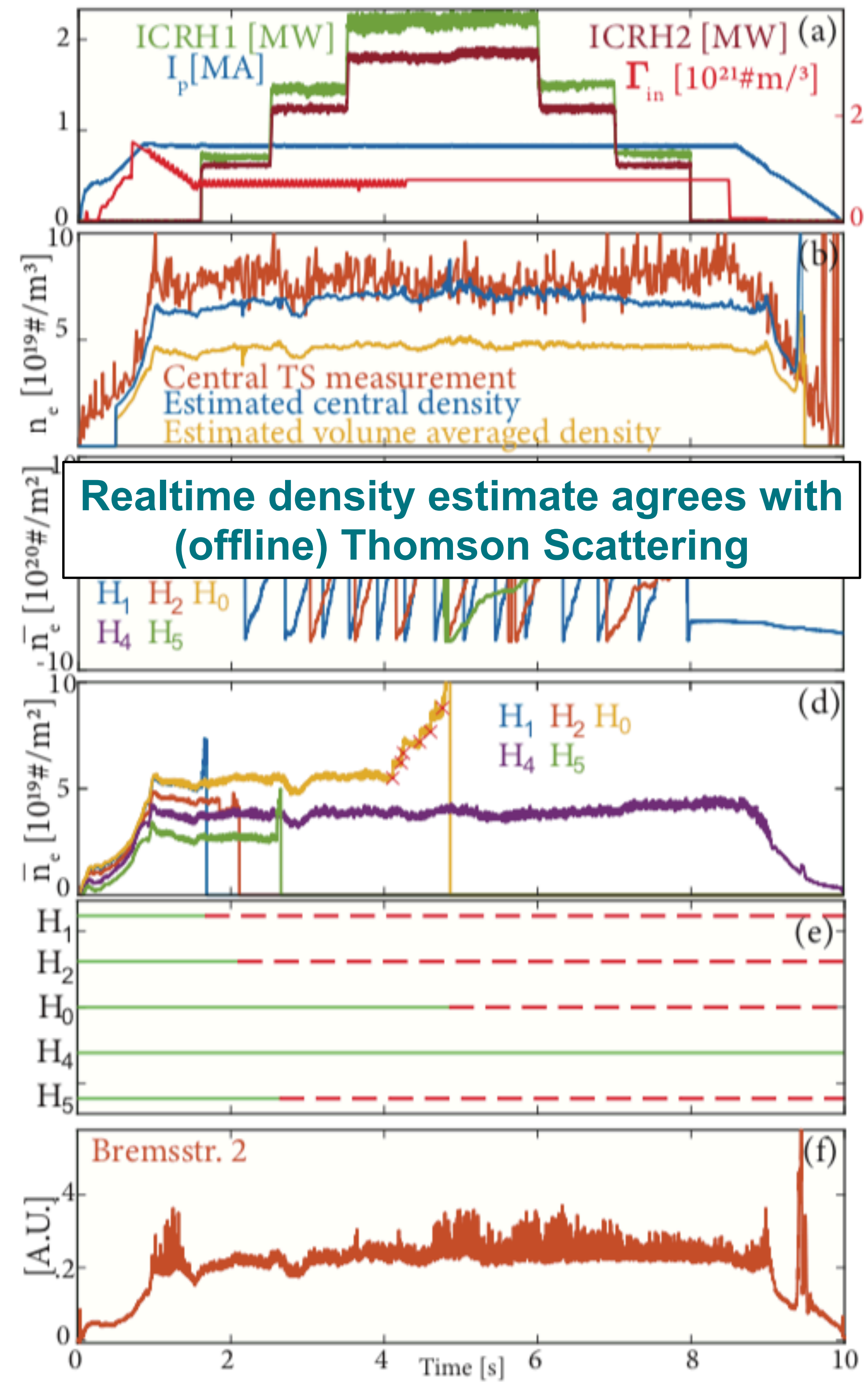
- **RAPDENS model (related to RAPTOR)**
  - **Model combining 1D profile evolution and particle inventory model.**
  - **Update to these predictions using interferometer & bremsstrahlung measurements**
  - **Detection and rejection of diagnostic faults and model inaccuracies.**
- **Needs**
  - **Real-time capable simulator for 1D profiles**
  - **Ad-hoc models for transport coefficients + sources**
  - **Real-time diagnostics**
- **Future improvements of models, or diagnostics, feed into same state observer, no need to change controller.**

[T. Bosman, Fus. Eng. Des, 2021]
TCV implementation [F. Pastore, Poster Tuesday]



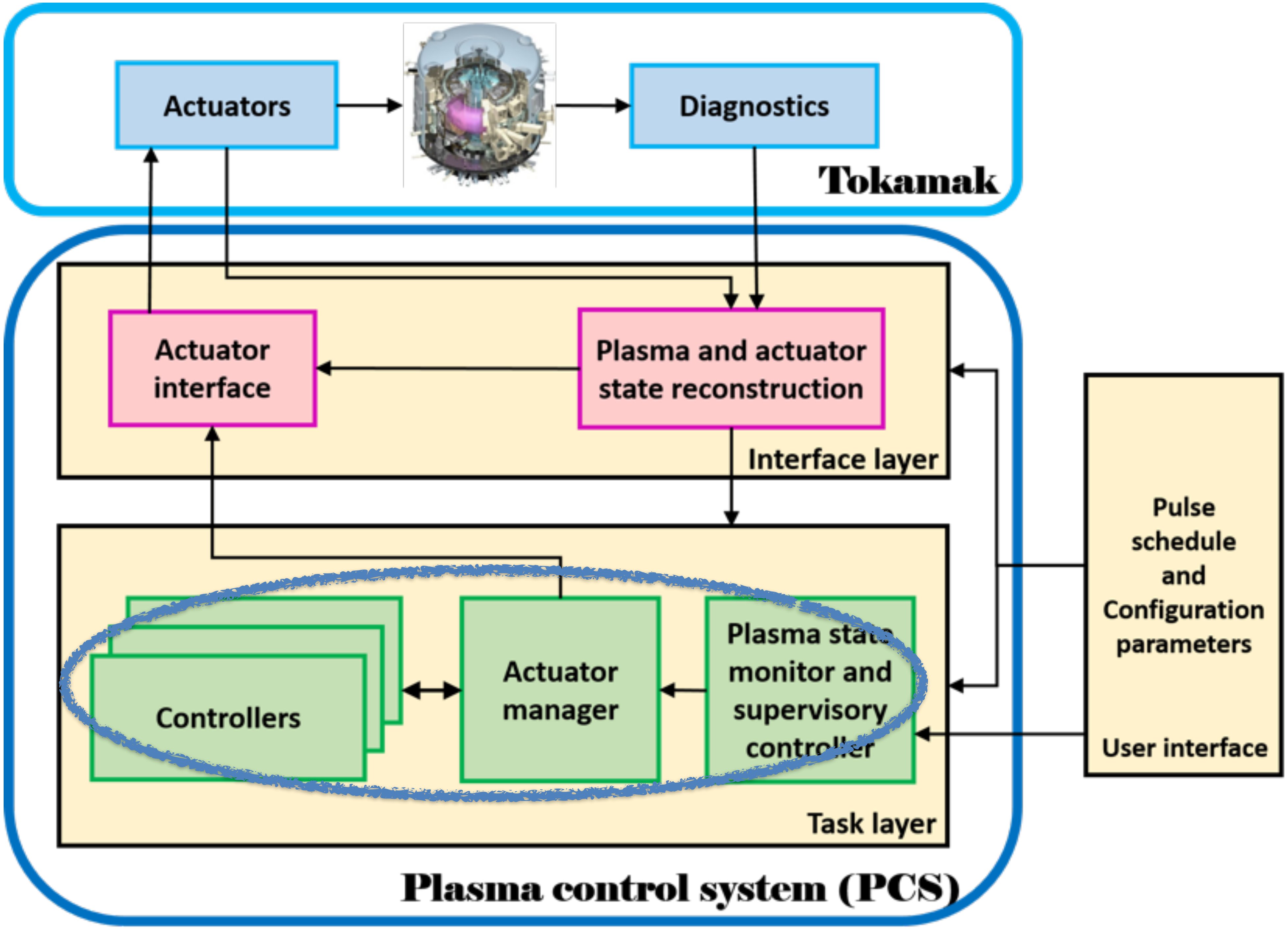
Example: real-time density profile reconstruction on ASDEX-Upgrade using state observer

- **RAPDENS model (related to RAPTOR)**
  - Model combining 1D profile evolution and particle inventory model.
  - Update to these predictions using interferometer & bremsstrahlung measurements
  - Detection and rejection of diagnostic faults and model inaccuracies.
- **Needs**
  - Real-time capable simulator for 1D profiles
  - Ad-hoc models for transport coefficients + sources
  - Real-time diagnostics
- **Future improvements of models, or diagnostics, feed into same state observer, no need to change controller.**

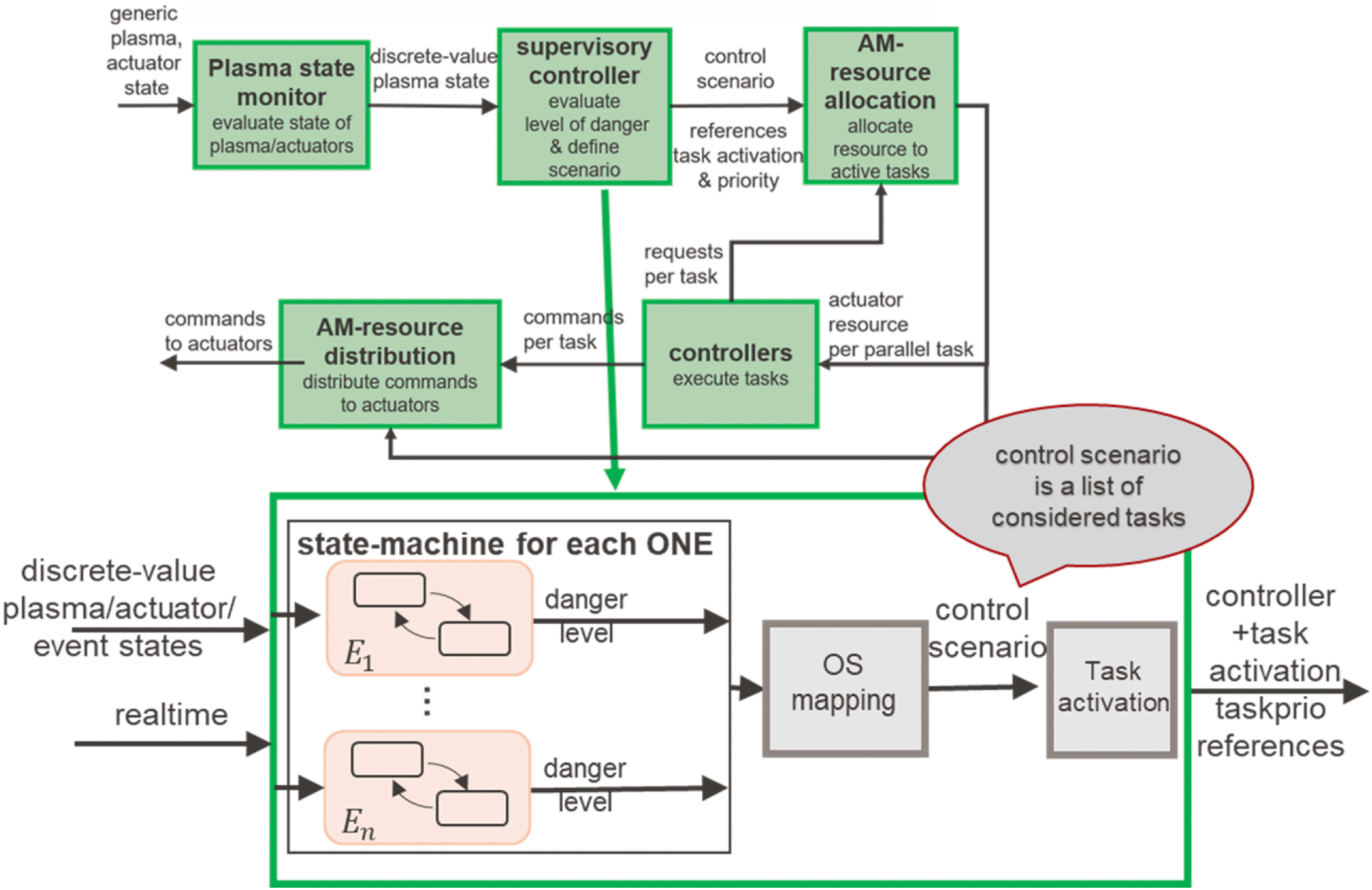


Realtime density estimate agrees with (offline) Thomson Scattering

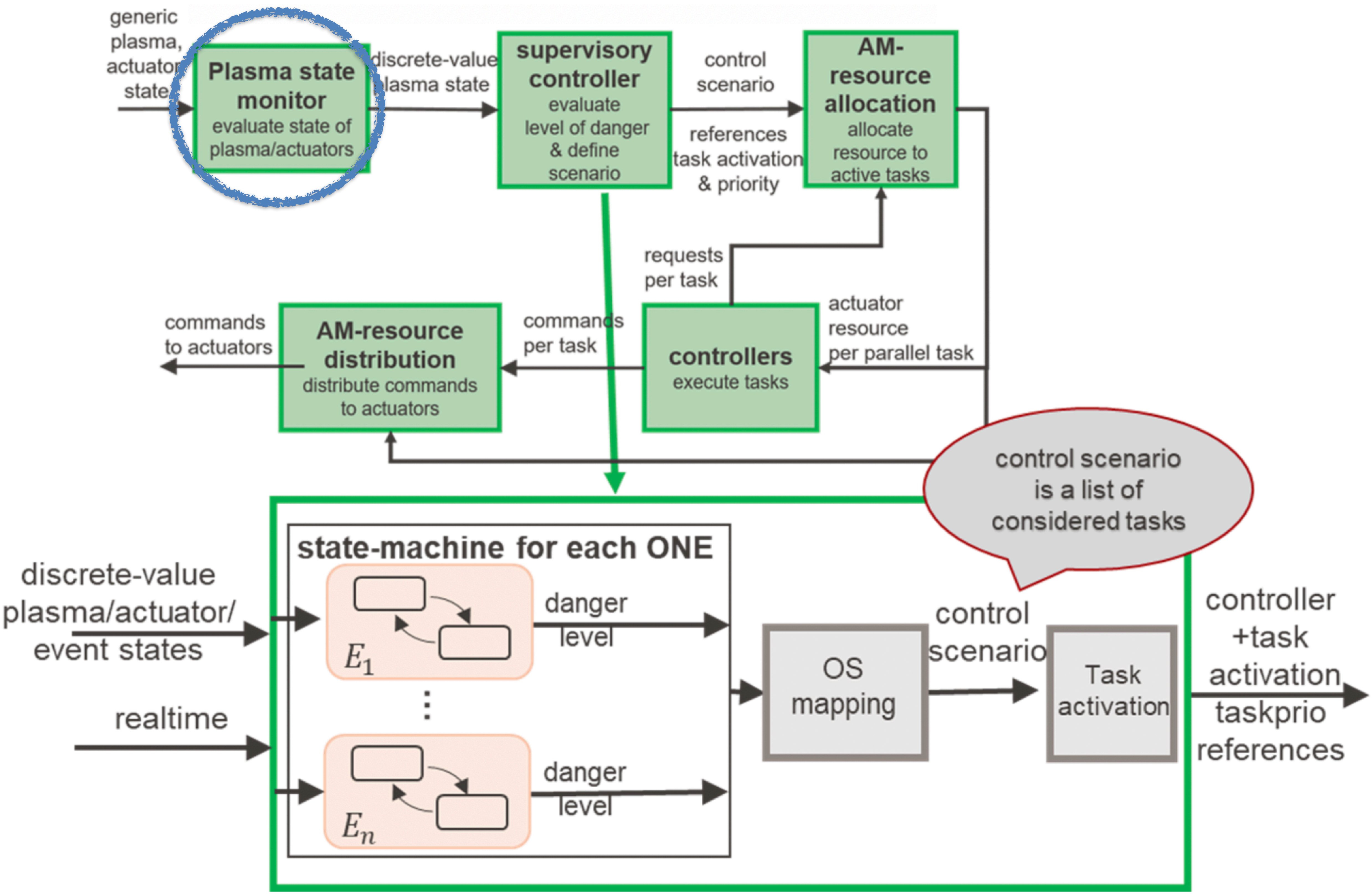
[T. Bosman, Fus. Eng. Des, 2021]
TCV implementation [F. Pastore, Poster Tuesday]


Actuators
Diagnostics
Tokamak
Actuator interface
Plasma and actuator state reconstruction
Interface layer
Controllers
Actuator manager
Plasma state monitor and supervisory controller
Task layer
Pulse schedule and Configuration parameters
User interface
Plasma control system (PCS)

Details of ‘Task’-based control layer

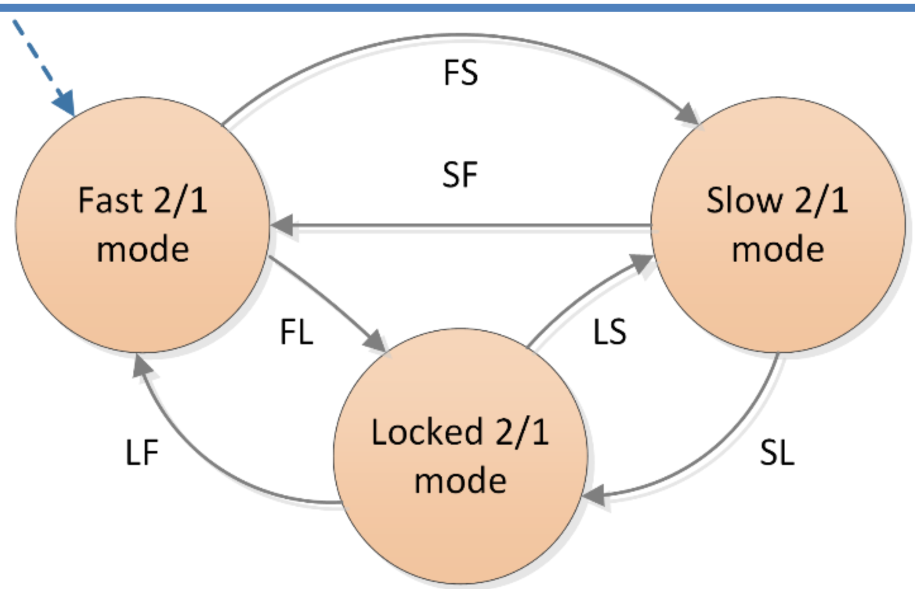


Details of ‘Task’-based control layer



Plasma state monitor translates continuous-valued plasma state estimate into discrete states

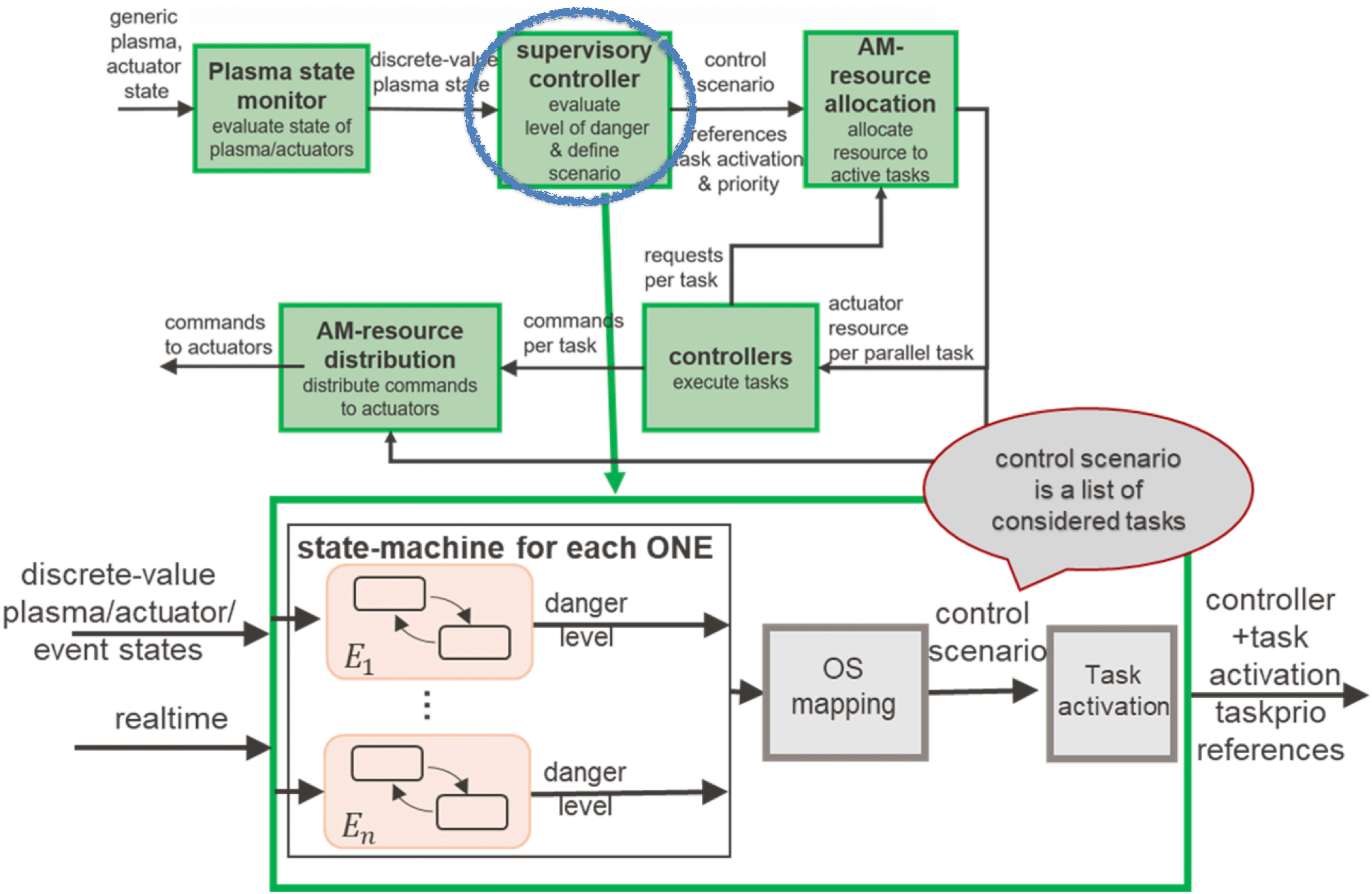
[T. Blanken NF 2019]



Transition	Conditional test
FS	$f_{2/1\mathrm{mode}} < f_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}}$ AND $A_{2/1\mathrm{mode}} > A_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}} + \epsilon_{\mathrm{A}}$
SF	$f_{2/1\mathrm{mode}} > f_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}} + \epsilon_{\mathrm{f}}$ OR $A_{2/1\mathrm{mode}} < A_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}}$
FL, SL	$f_{2/1\mathrm{mode}} < f_{2/1\mathrm{mode}}^{\mathrm{thresh,SL}}$ OR $A_{n=1\mathrm{LM}} > A_{n=1\mathrm{LM}}^{\mathrm{thresh}} + \epsilon_{\mathrm{A}}$
LF	$f_{2/1\mathrm{mode}} > f_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}} + \epsilon_{\mathrm{f}}$ AND $A_{\mathrm{n}=1\mathrm{LM}} < A_{\mathrm{n}=1\mathrm{LM}}^{\mathrm{thresh}}$
LS	$f_{2/1\mathrm{mode}} > f_{2/1\mathrm{mode}}^{\mathrm{thresh,SL}} + \epsilon_{\mathrm{f}}$ AND $f_{2/1\mathrm{mode}} < f_{2/1\mathrm{mode}}^{\mathrm{thresh,FS}} + \epsilon_{\mathrm{f}}$ AND $A_{n=1\mathrm{LM}} < A_{n=1\mathrm{LM}}^{\mathrm{thresh}}$

- **Discrete representation of plasma state (including events)**
  - **Receives continuous-valued information from state reconstruction.**
- **User-configurable thresholds**
  - **Different thresholds for each tokamak.**

Details of ‘Task’-based control layer

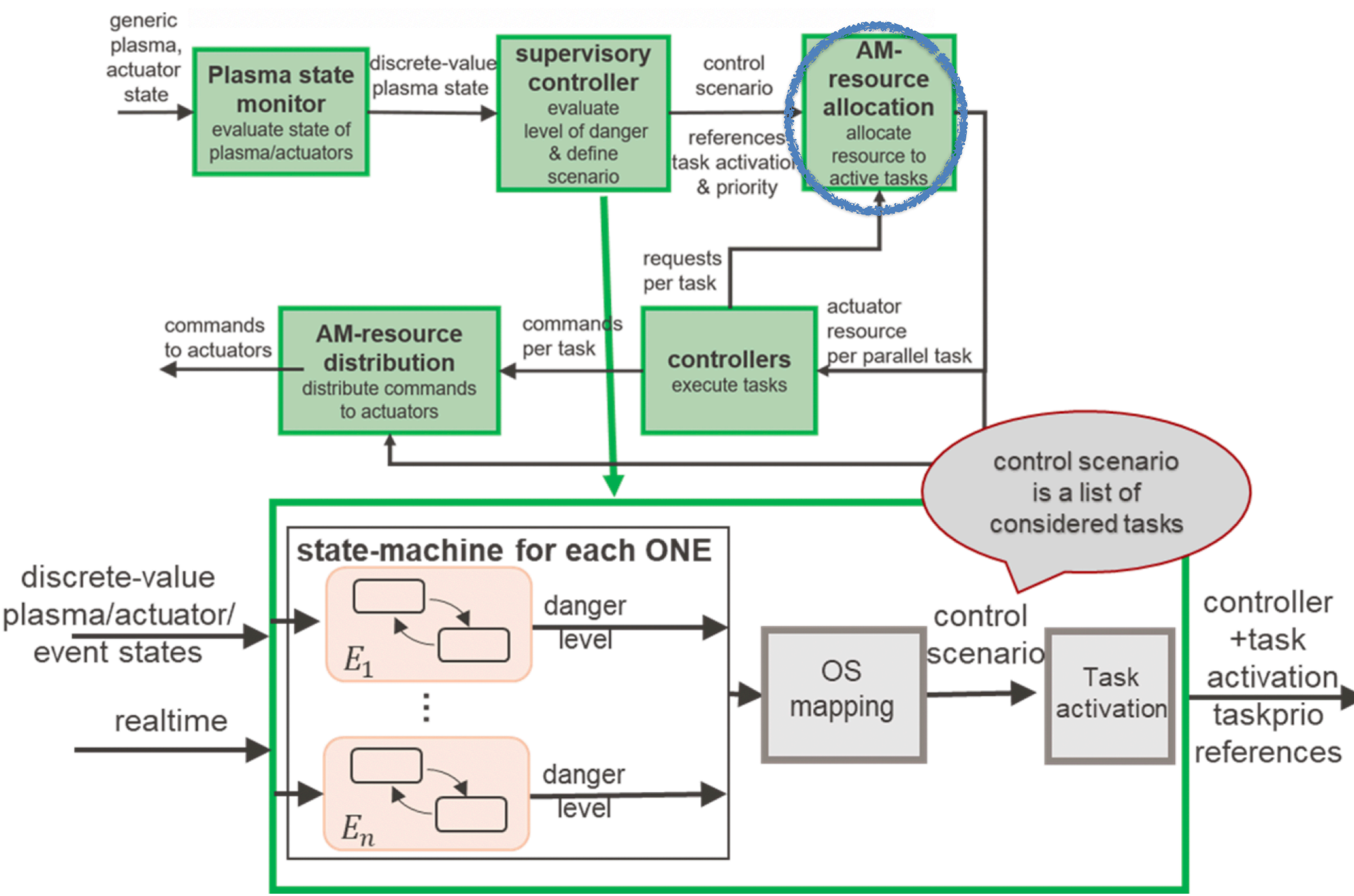


Supervisor: map discrete-valued plasma state description into prioritized tasks

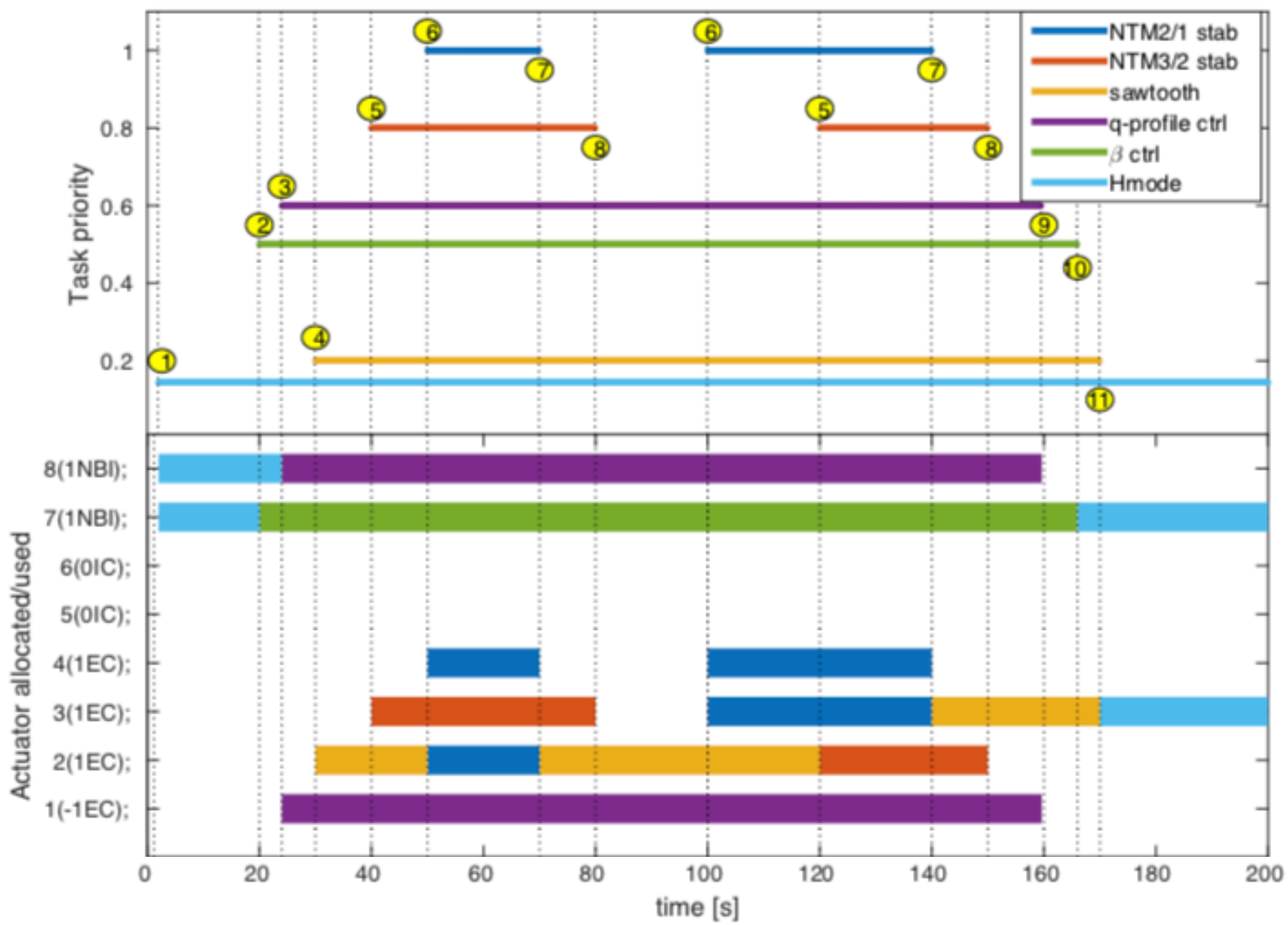
- **Rule-based mapping. Example:**

	Plasma parameters are within defined 'normal' bounds	A 2/1 NTM is present (size = SMALL or MEDIUM)	A 2/1 NTM is present (size == LARGE)
Tasks (prioritized)	• 2/1 NTM preemption • β control • q profile control	• 2/1 NTM stabilization • β control with lower reference	• Perform soft-stop (ramp-down)
Control task parameters	• High β reference. • 2 MW EC on q=2.	• Lower β reference. • Increase EC power on q=2 until NTM is stabilized.	• Appropriate soft-stop trajectory given present state. • (OR trigger disruption mitigation etc)

Details of ‘Task’-based control layer



Actuator manager decides in real-time which actuator resources are assigned to which control tasks



Example of RT actuator allocation for ITER control tasks
see [T. Vu et al, Fus. Eng Des 2019]

- **Constrained optimization problem with both integer and continuous variables.**
  - **Heuristic approach works for case with few actuators / tasks.**

Mixed-integer quadratic programming formulation of actuator allocation problems

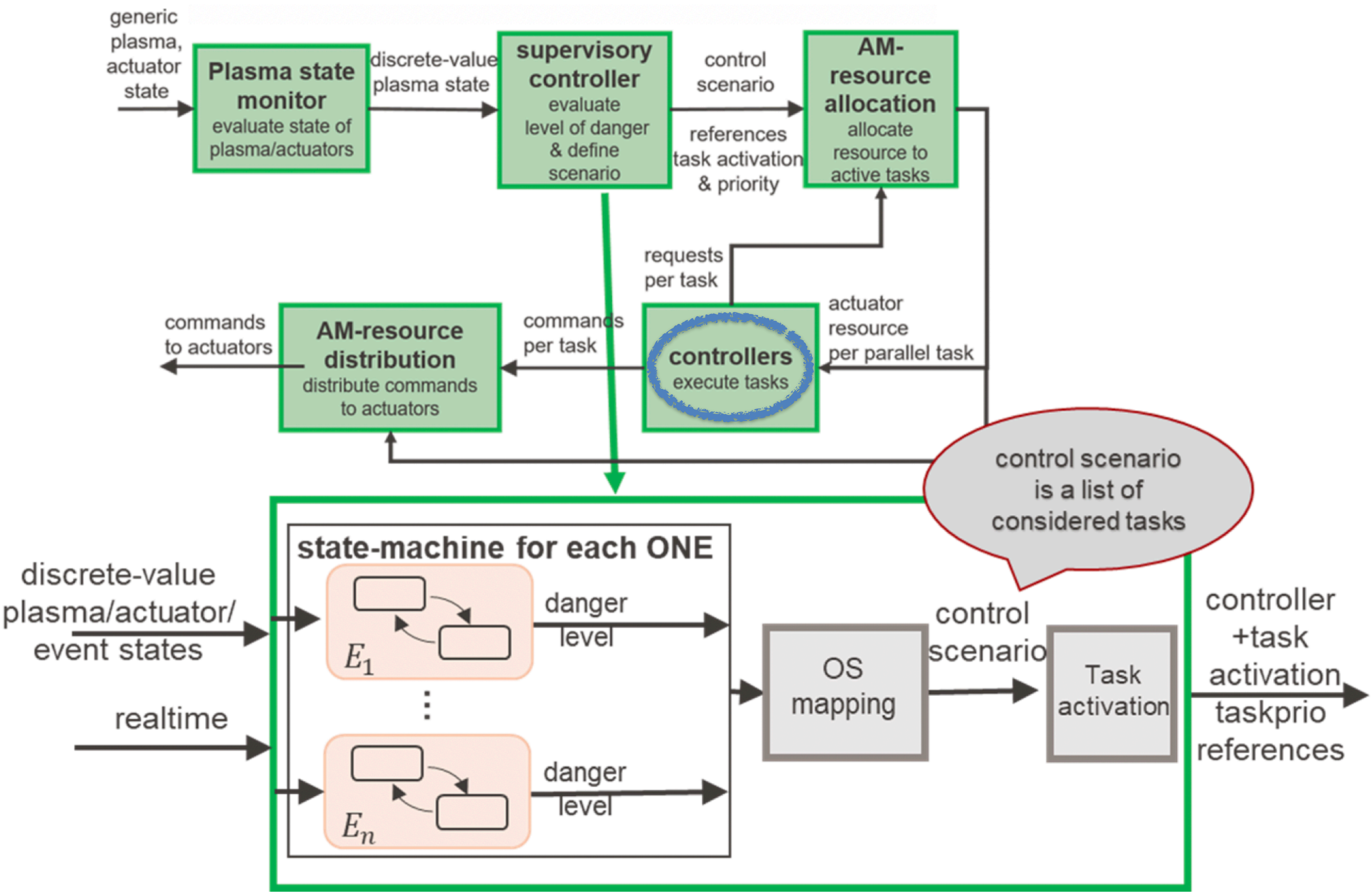
[E. Maljaars & F. Felici, Fus. Eng Des 2017]

- Resource allocation problems have often been formulated in a flexible format as Mixed Integer (Quadratic) Programming problems
  - Optimization problem involves integer (and continuous) variables

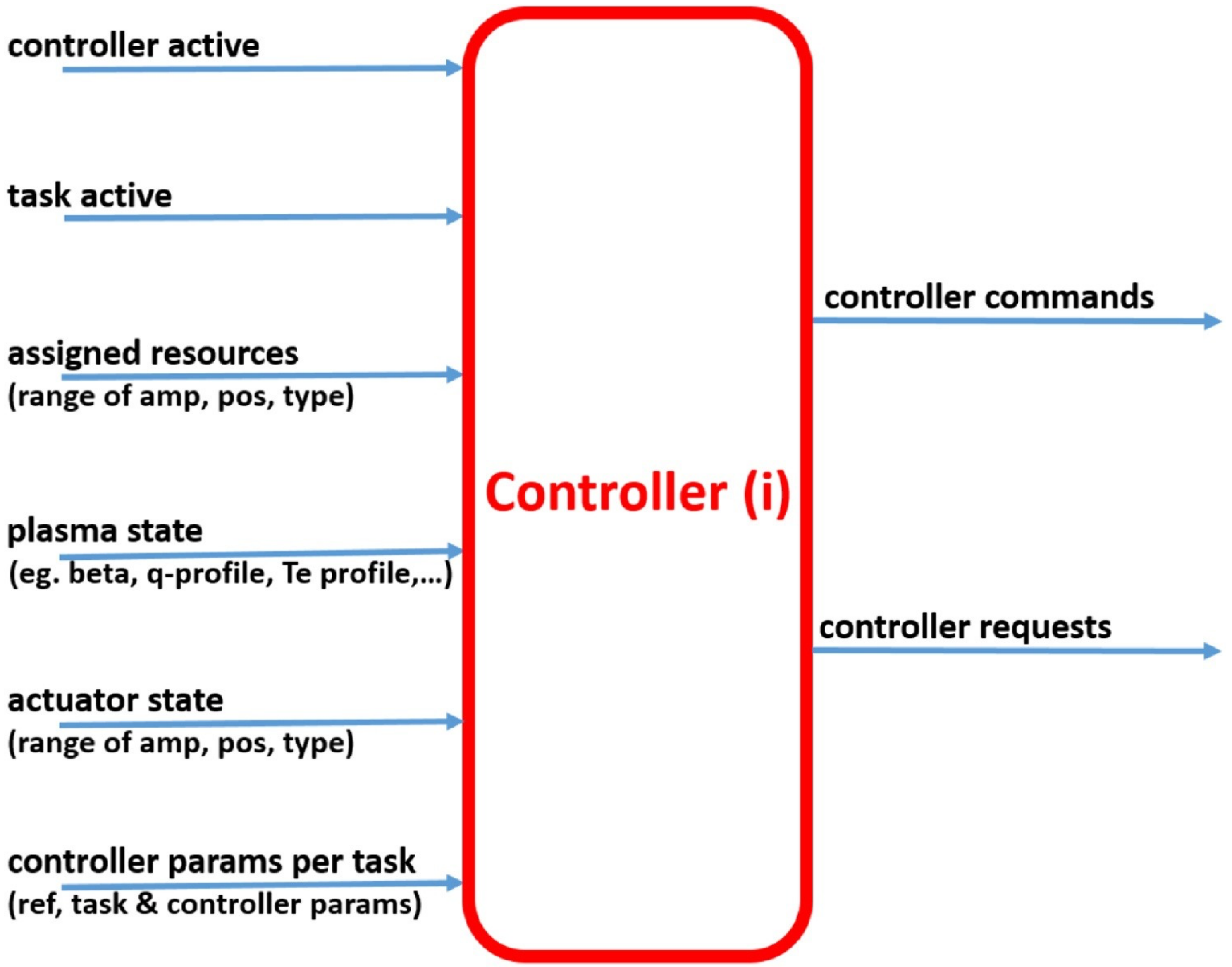
$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & J(x) = x^\top H x + f^\top x \\
\text{subject to} & A_{ineq} x \leq b_{ineq} \\
 & x_{min} \leq x \leq x_{max} \\
 & x_i \in \mathbb{N}
\end{array}
$$

- Cost function: things that are desired (easy to add/remove terms)
  - Actuator allocation: promote good / penalize bad allocations
- Constraints: things that must be satisfied (easy to add/remove terms)
  - For actuator allocation: actuator availability and allowed allocations

Details of ‘Task’-based control layer

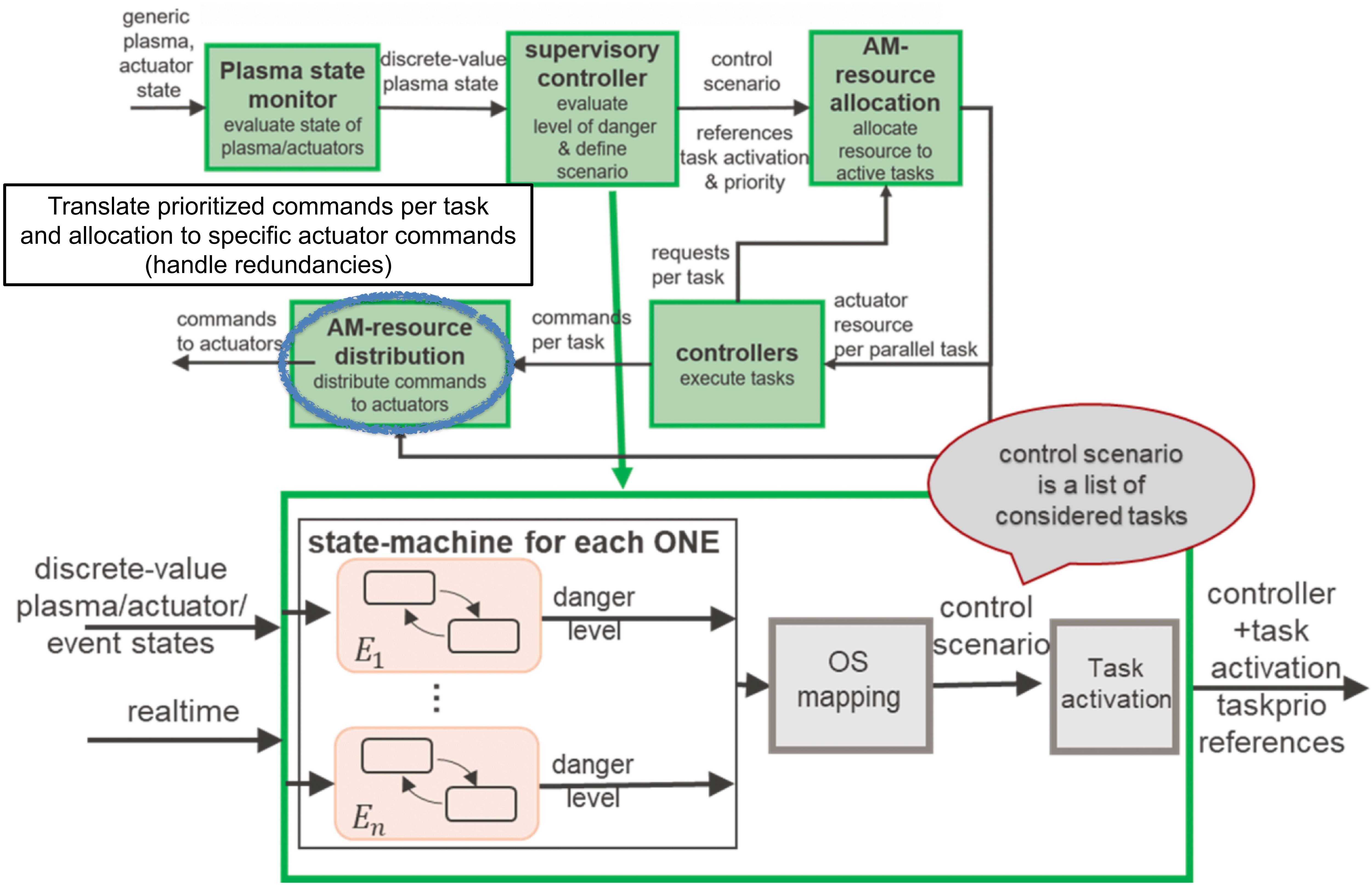


Controllers execute (one or several) control tasks, receive resource allocations and send resource requests

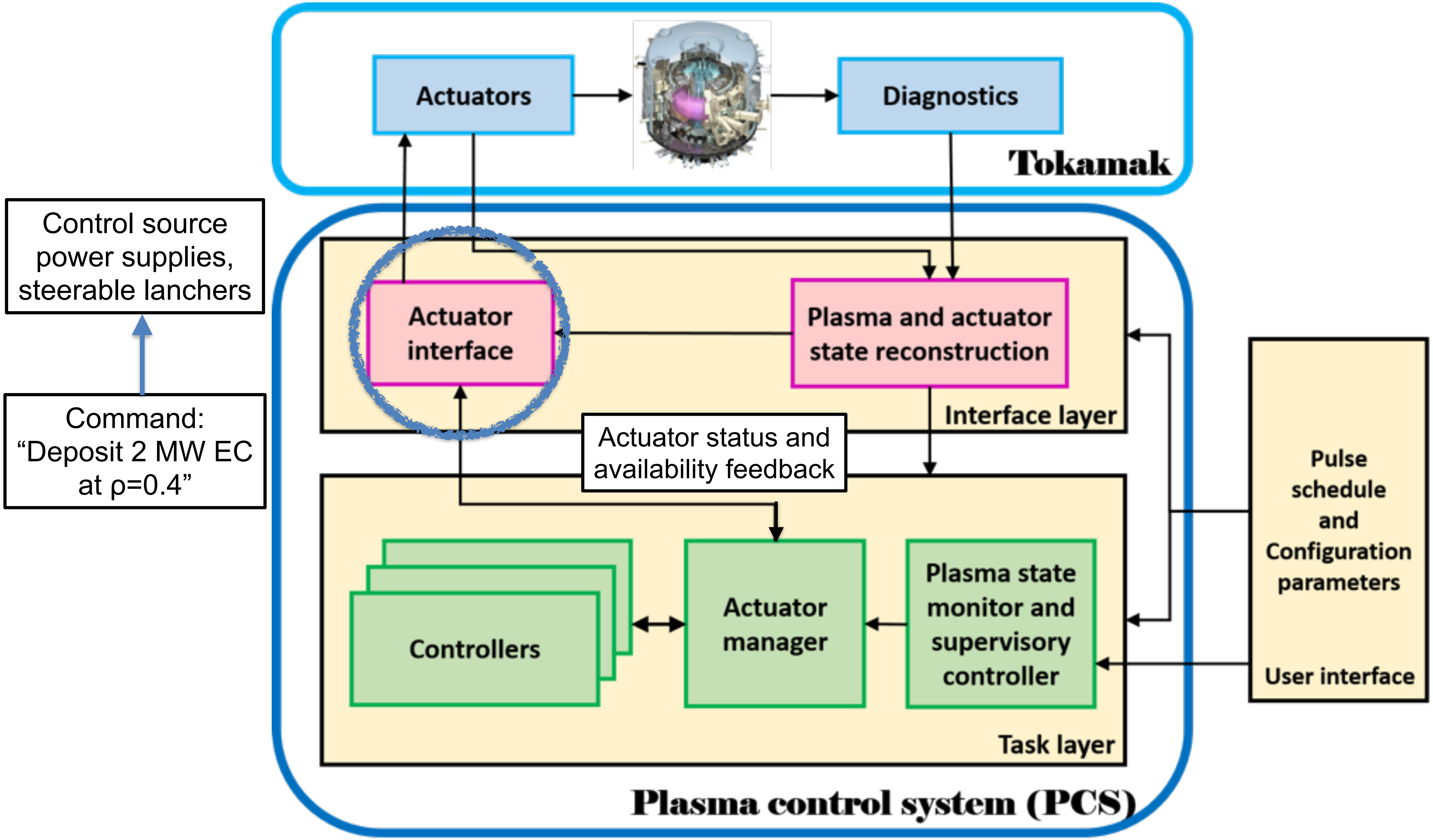


- **Generic interfaces for all controllers**
- **Enables use of resource-aware controllers (e.g. Model Predictive Control)**
  - **More details & examples in [Schuster, lecture Wednesday)**

Details of ‘Task’-based control layer



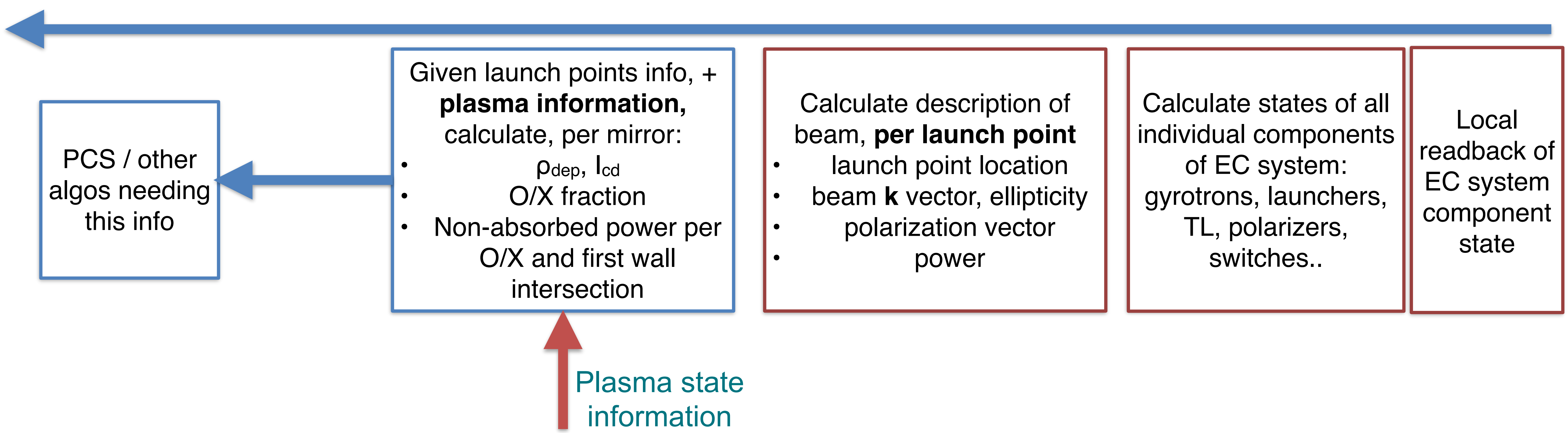
Actuator interface translates generic actuator commands into (hardware-)specific commands for a given tokamak



Example of ITER EC actuator interface proposal

See [G. Carannante proceedings EC-21 conference (2022)]

- **Function 1: Knowing where EC power is being deposited now**



- **NB Plasma information comes from plasma state reconstruction support functions**

Example of ITER EC actuator interface proposal

- **Function 2: Describe potential availability, now and in the future**

PCS algos needing this info

Actuator interface

Actuator availability in terms of P_{dep}, rho_{dep}, I_{ECCD}…

Translate availability in terms of k vector, power per mirror… into rho, I_current drive..

Per launch point, calculate: Availability of power, location, k & polarization vectors, (present and future)

Calculate set of potential states of EC system components (present and future)

Local readback of EC system component state + potentially settable states

- **Needs representation of EC availability in terms of power/ polarization/angles of last mirror.**
  - **Representation to be determined, likely a set of inequality constraints, or a tree**
  - **Include mutual exclusion conditions etc**

Example of ITER EC actuator interface proposal

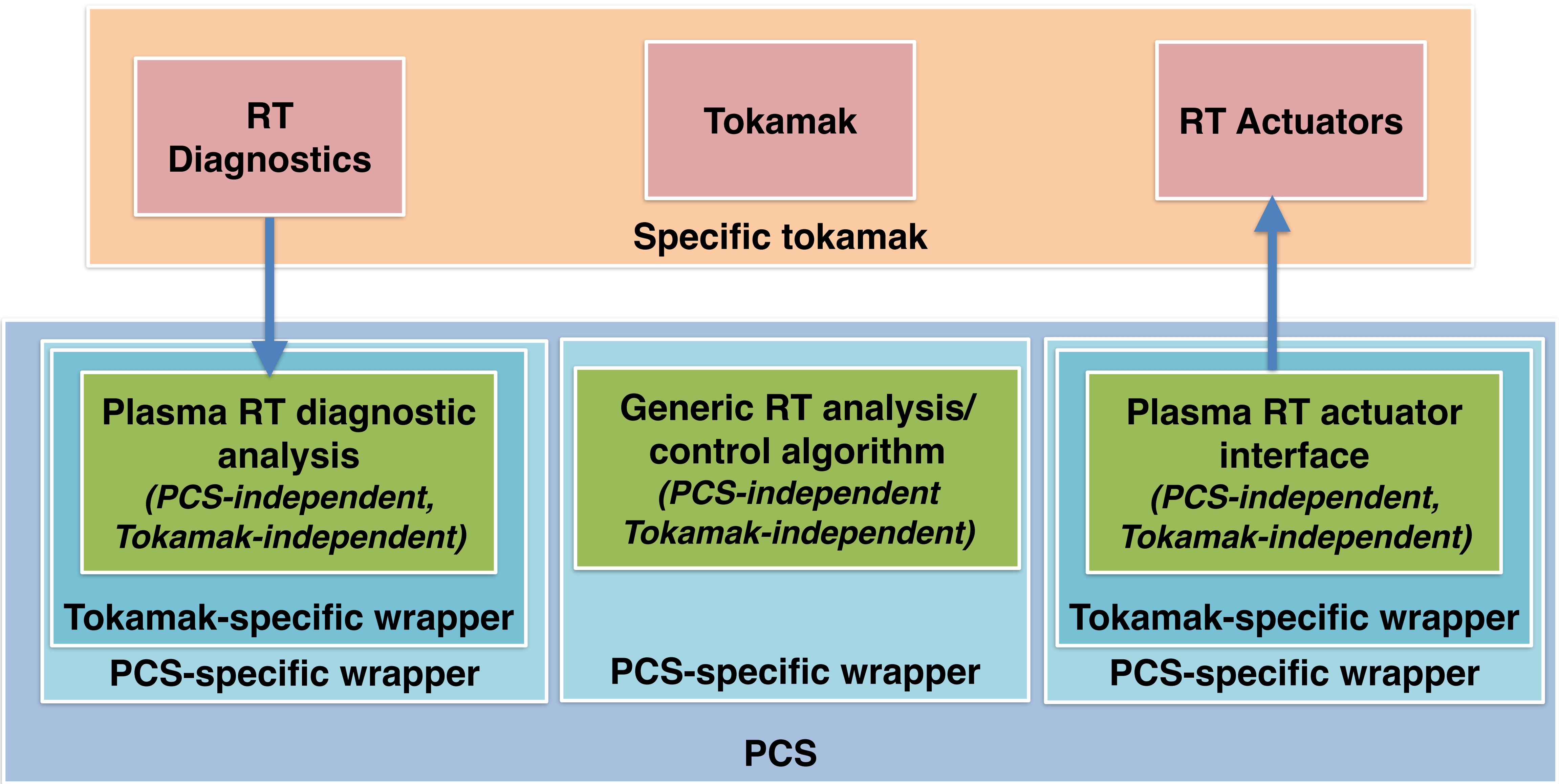
- **Function 3: 'Command' to inject EC at desired location**

PCS control task prioritization	PCS command: "Deposit to given rho with given power and I_{cd}	Decide X or O mode. Determine launch point, **k** vector, etc to achieve desired deposition. Find polarization vector **at mirror** for desired O/X mode. (inv. ray tracing + optimization if multiple solutions)	Decide how to set launchers/gyrotrons/ switches/tl/polarizers to actuate command.	Positioning of mirror angles, polarizers, switches	Local control of EC system components

- **Separation of concerns:**
  - **Actuator management on PCS side does optimization based only on effect of EC on plasma (+wall) in terms of rho, I_{ECCD}, $P_{absorbed}$, and decides desired EC system state at launch points.**
  - **EC system decides how to actuate EC system components to obtain desired EC power at launch points.**

Implementation aspects to promote algorithm portability

- **Try to strictly separate parts of PCS software:**
  - **Tokamak-dependent / Tokamak-independent**
  - **PCS-dependent / PCS-independent**



Outlook for supervisory control

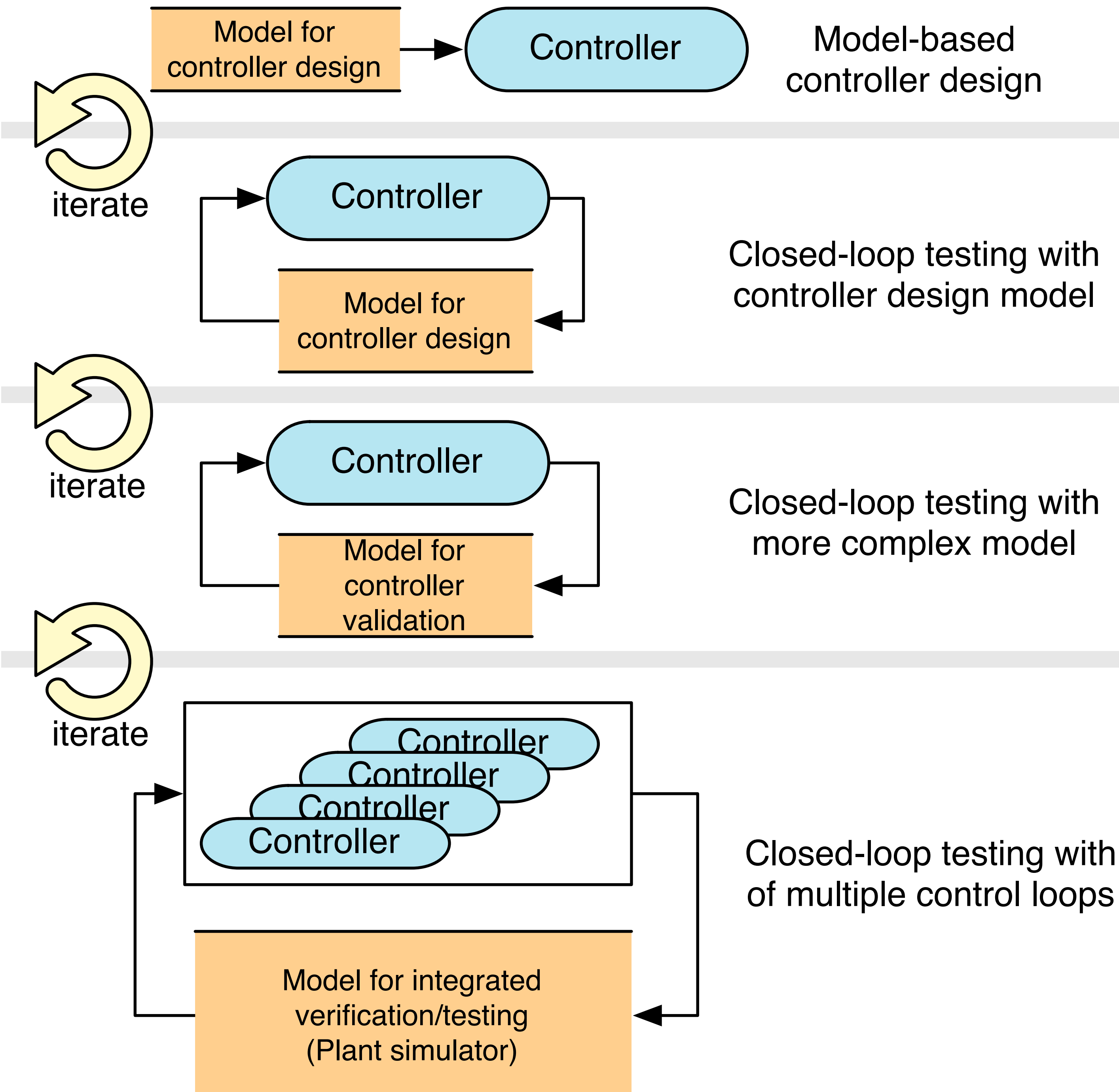
- **Architectures are being tested successfully on various tokamaks**
  - **Also enable new experiments studying physics in better-controlled ways**
- **Solid, *extensible* architecture designed for ITER**
- **Tricks are in the details: implementing and validating:**
  - **State observers giving us all the physics quantities we need to know in real-time**
  - **Event detectors for all the N events we care about**
  - **Controllers for everything we want to control**
    - **Incl. resource-aware controllers, predictive controllers, …**
  - **Program it all, validate and test it all**

- **From the control point of view, present research-oriented tokamaks are a dream**
  - **Many diagnostics, many flexible actuators -> 'pay' in control complexity**
- **What about a fusion reactor?**
  - **Run one scenario but fewer diagnostics and actuators**

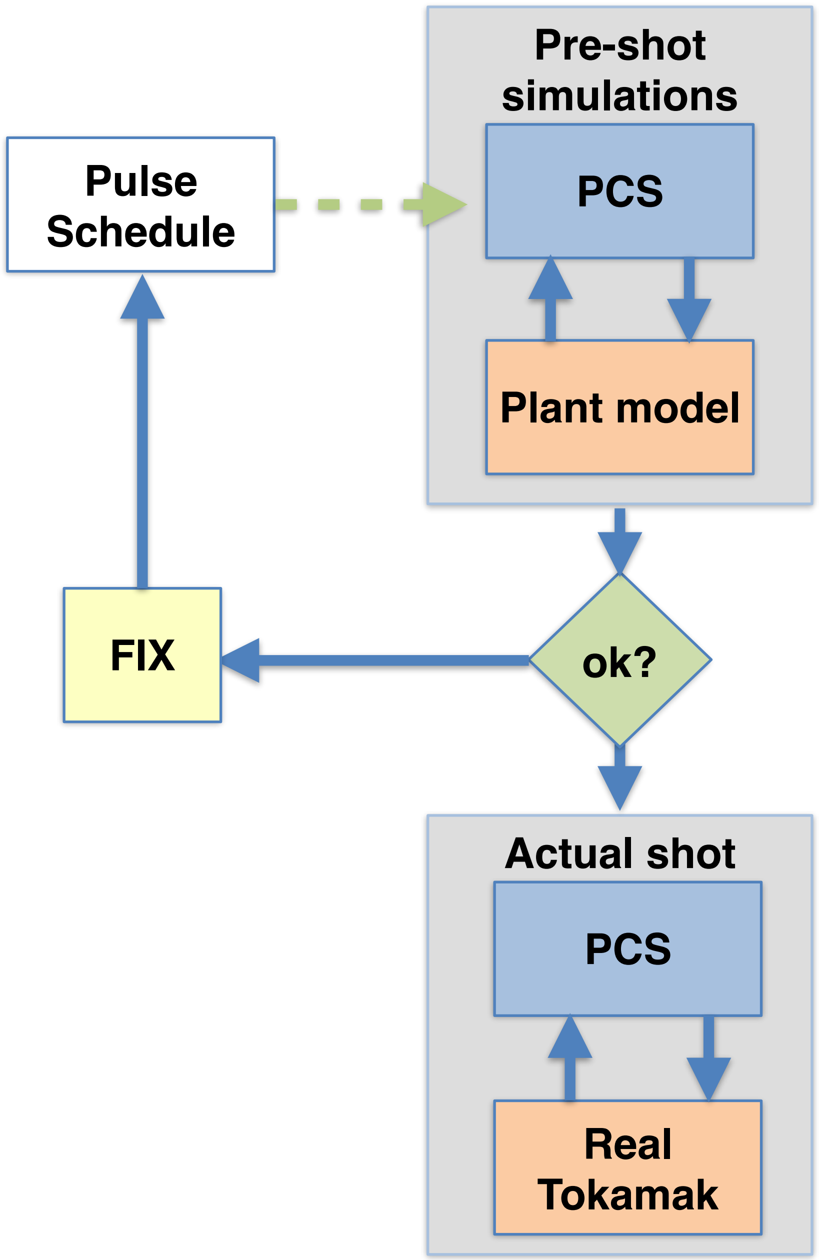
Implementation challenges and software aspects

A hierarchy of models is needed for different phases of controller design/validation/verification

- **To design and test controllers, a model of the system is essential**
  - **Models of varying complexity are used at various stages of design/testing**
  - **Design/choice of correct model for task is an integral part of the control engineer's task.**
- **Typical examples**
  - **Controller design models:**
    - **CREATE-L, RAPTOR, RZIP**
  - **More complex 'integrated' simulators**
    - **ASTRA, RAPTOR, COTSIM, CREATE-NL**
  - **Full tokamak 'plant' simulators:**
    - **Control-oriented 'Flight Simulator' (faster, empirical parameters)**
    - **High Fidelity Plasma Simulators (slower, more physics-based)**

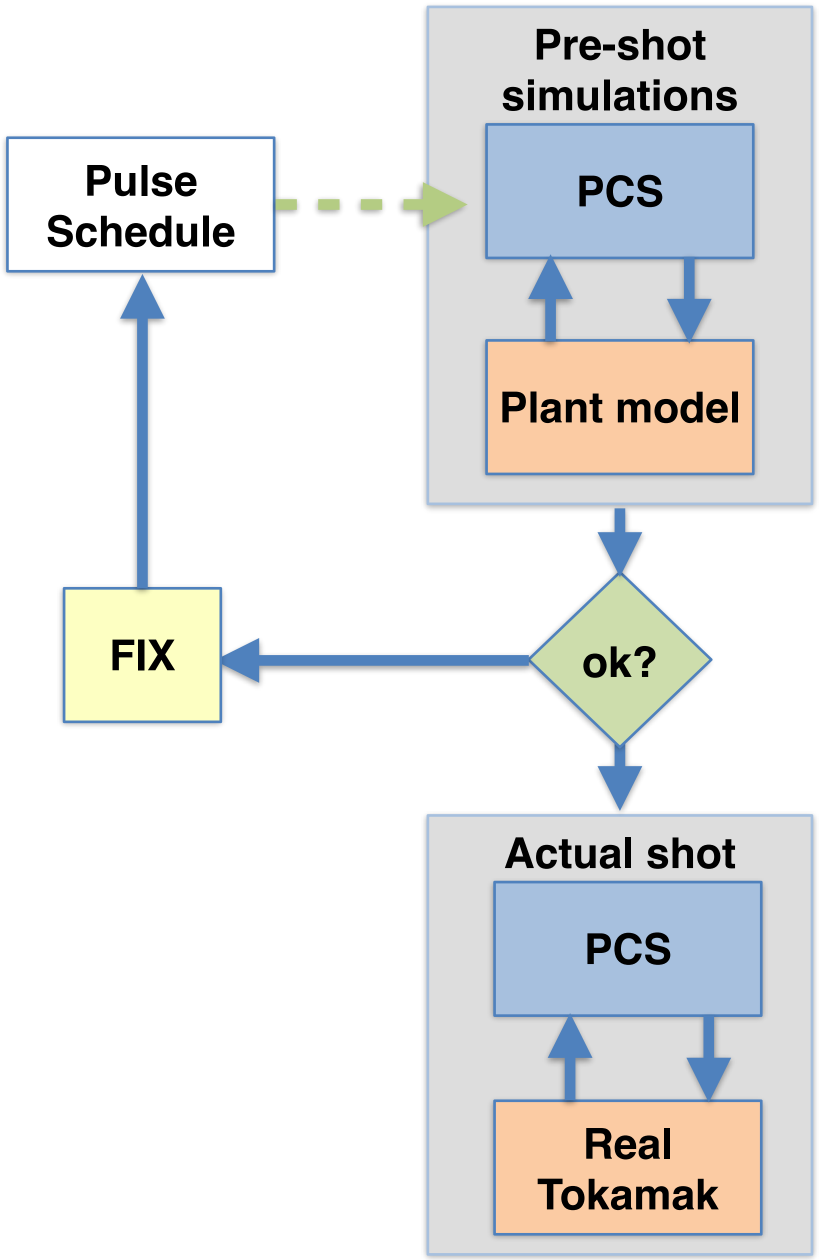


Pre-shot model-based validation of discharge program …
… & feedback of experimental data into model



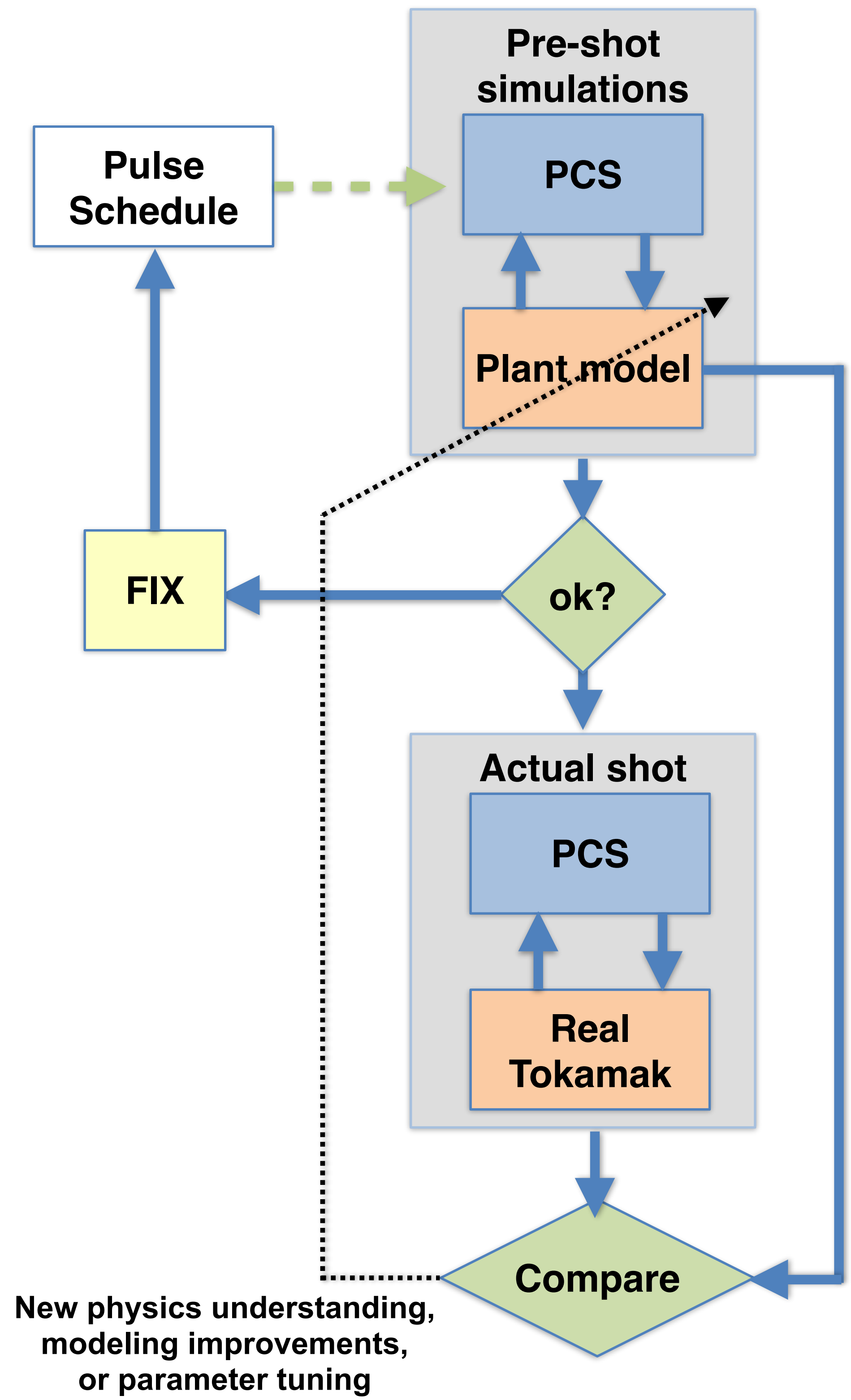
Pre-shot model-based validation of discharge program …
… & feedback of experimental data into model

- **Operational limit checking:**
  - Check that discharge program does not exceed operational boundaries (though we have real-time protection systems)
- **Use best available “Flight Simulators” in closed-loop with a PCS (simulated or real)**



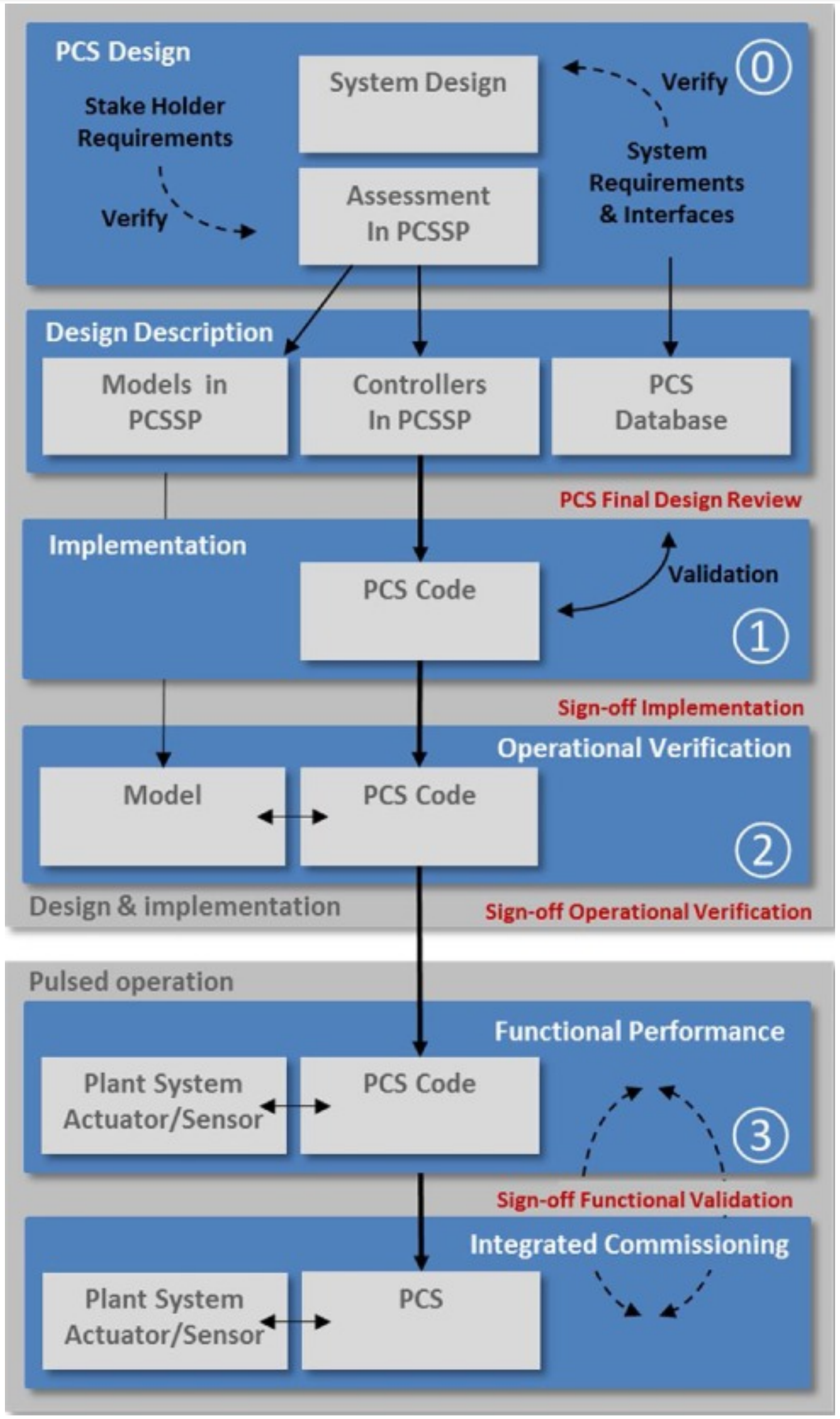
Pre-shot model-based validation of discharge program …
… & feedback of experimental data into model

- **Operational limit checking:**
  - Check that discharge program does not exceed operational boundaries (though we have real-time protection systems)
- **Use best available "Flight Simulators" in closed-loop with a PCS (simulated or real)**
- **Deviations between pre-shot validation simulation and post-shot data contains valuable information**
  - Improvement of models by changing device-specific parameters.
  - The physics we are trying to learn
  - Feed improved understanding into better models used for future control validations
  - Validated models (the code itself) are one of the key products of operating a tokamak



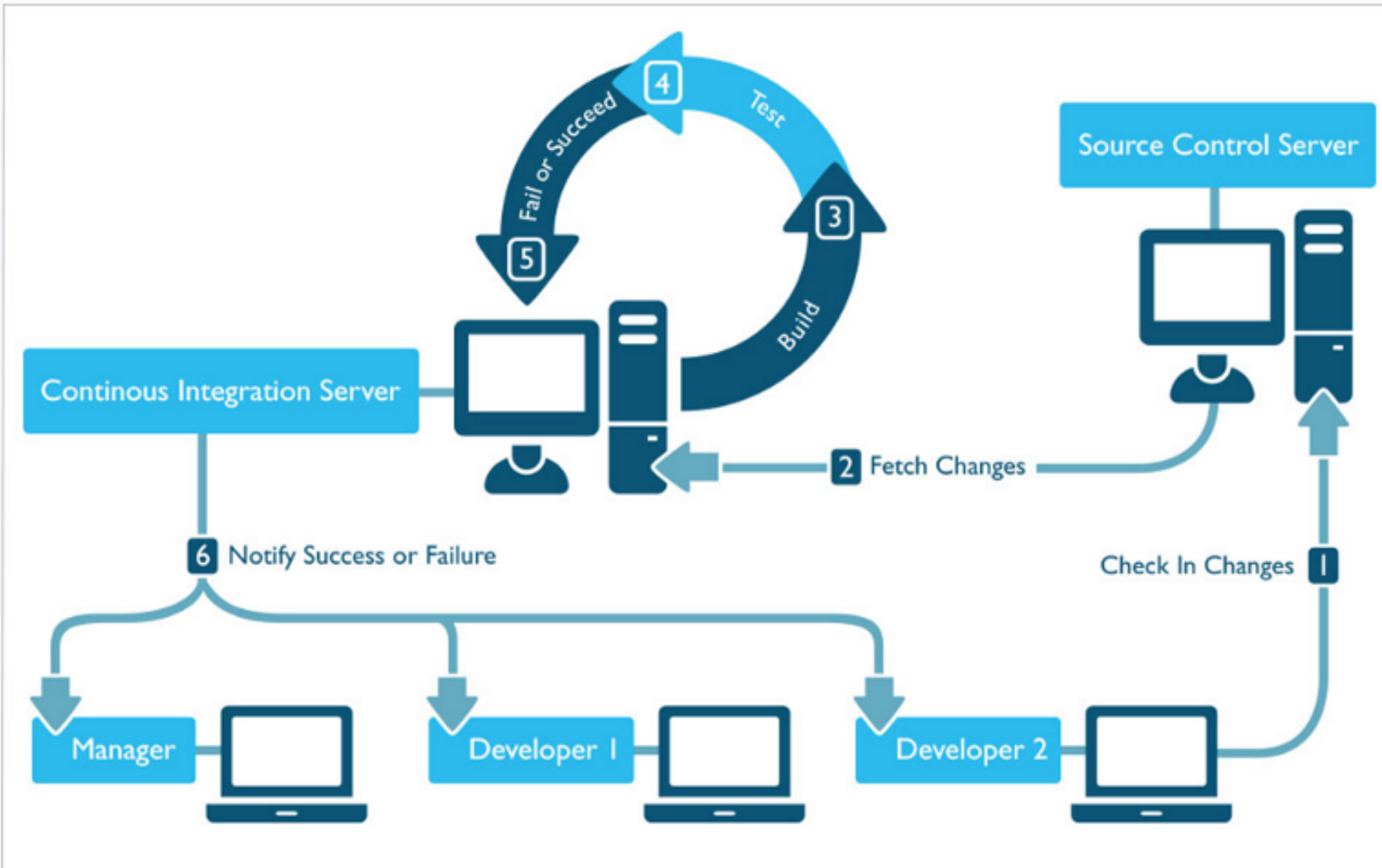
Managing workflows of different stages of software validation is challenging but essential for future devices

- Validation of PCS software via closed-loop simulations with plant models
- Verification & validation tests on:
  - Control software
  - Model software used to test the controls
- Need to do this:
  - Over ITER lifetime (several decades)
  - On several parallel versions of PCS software for various stages
  - While dozens++ of contributors propose changes and upgrades
- This is a “Large Software Project”
  - Need concepts from software engineering: continuous integration / deployment / DevOps



From [P. de Vries et al. Fus. Eng. Des 2018]

Continuous Integration (CI)

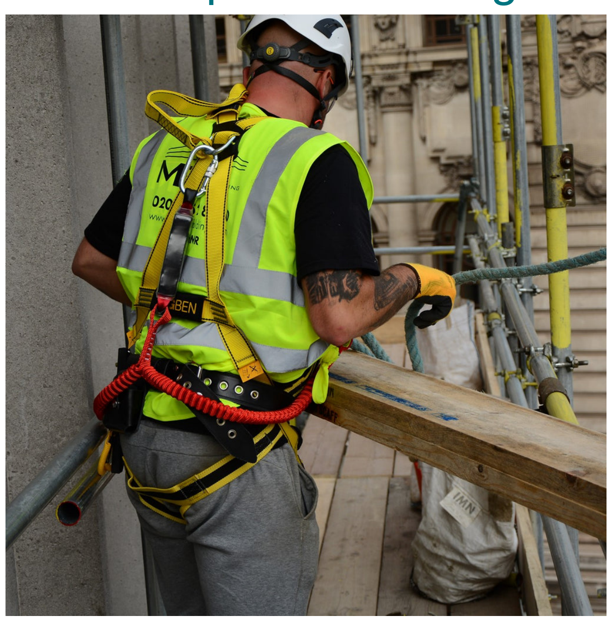


- **Automated, fast & frequent feedback of effects of code changes!**
  - **Requires codes with TESTS**

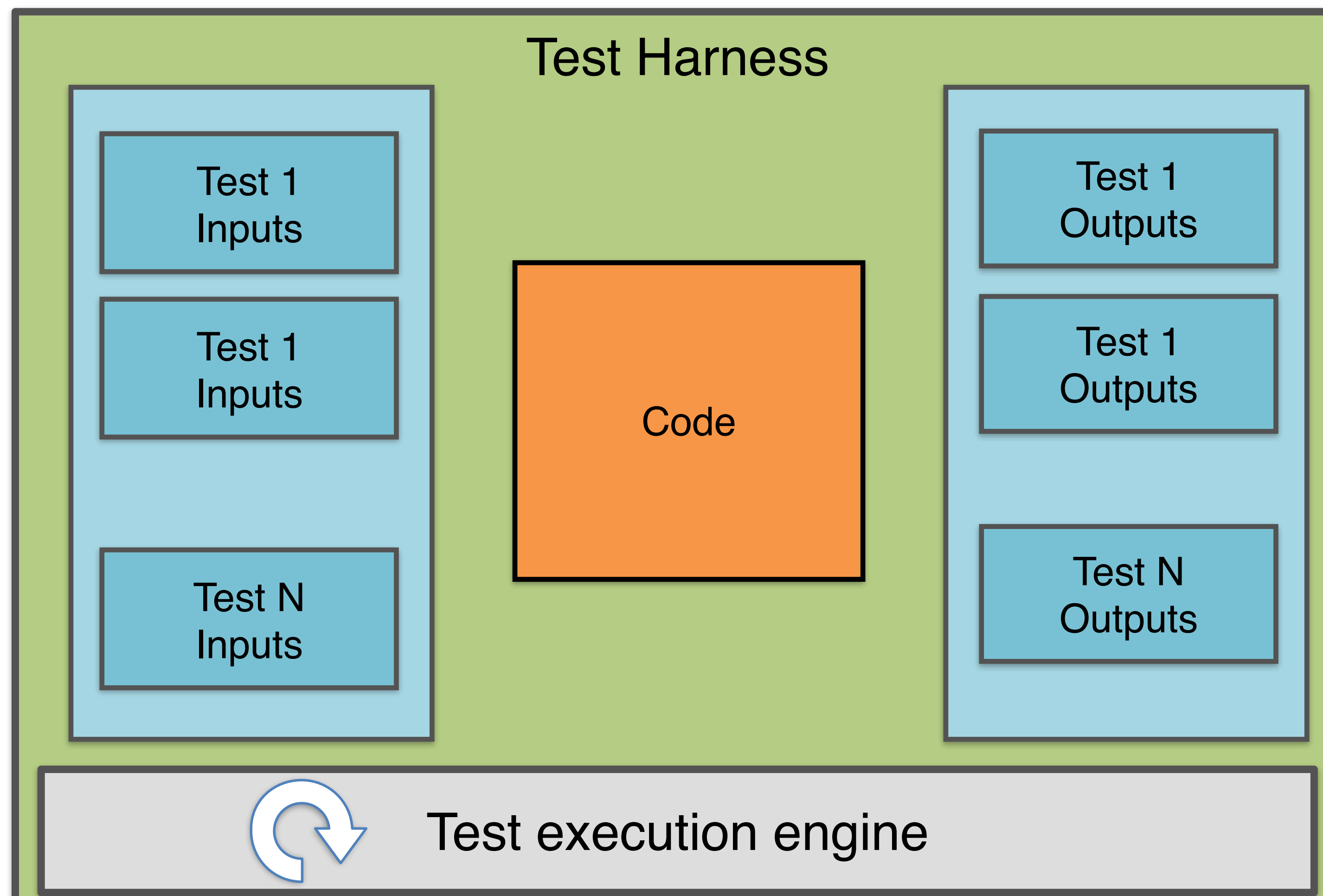
The importance of testing in software engineering

- **Write tests <u>together</u> with code**
  - **For given input, expect a given output**
  - **As functionality expands, expand test suite**
- **Establish a 'contract', fixing expected code behaviour**
- **Run tests automatically and regularly**

Harness allows this man to mis-step without danger



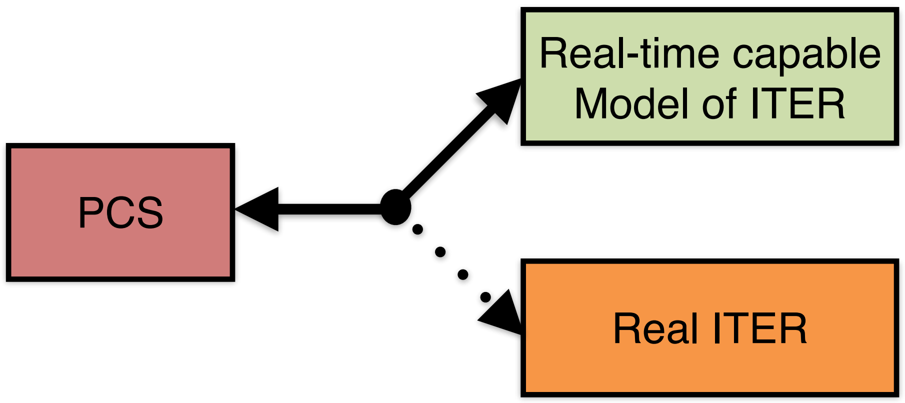
Test harness allows developer to change code without fear of introducing bugs



Types of tests

- **Various levels of testing:**
  - **Unit testing: test small functional units of code - e.g. test an ODE solver**
  - **Integration tests: Tests of useful combinations of units**
  - **End-to-end tests: Test the whole thing**

- **Various aspects of Plasma Control software to be tested:**
  - **Functional tests of individual controllers (ITER: PCSSP)**
  - **Functional tests of combinations of controllers (ITER: PCSSP)**
  - **Tests that control code in simulation code same behaviour as code in production**
    - **PCSSP version vs RTF version (could be the same)**
  - **Hardware-in-the-loop tests**
    - **Tests of production PCS on real-time capable model of the whole system**



The DevOps confusion

From: https://www.devops.ch/2017/05/10/devops-explained/

David is a DEVeloper !

David wants to maximize **change**

Control algorithm developer

The DevOps confusion

From: https://www.devops.ch/2017/05/10/devops-explained/

Wall of Confusion

David is a DEVeloper !

David wants to maximize **change**

Control algorithm developer

Peter is an OPerator !

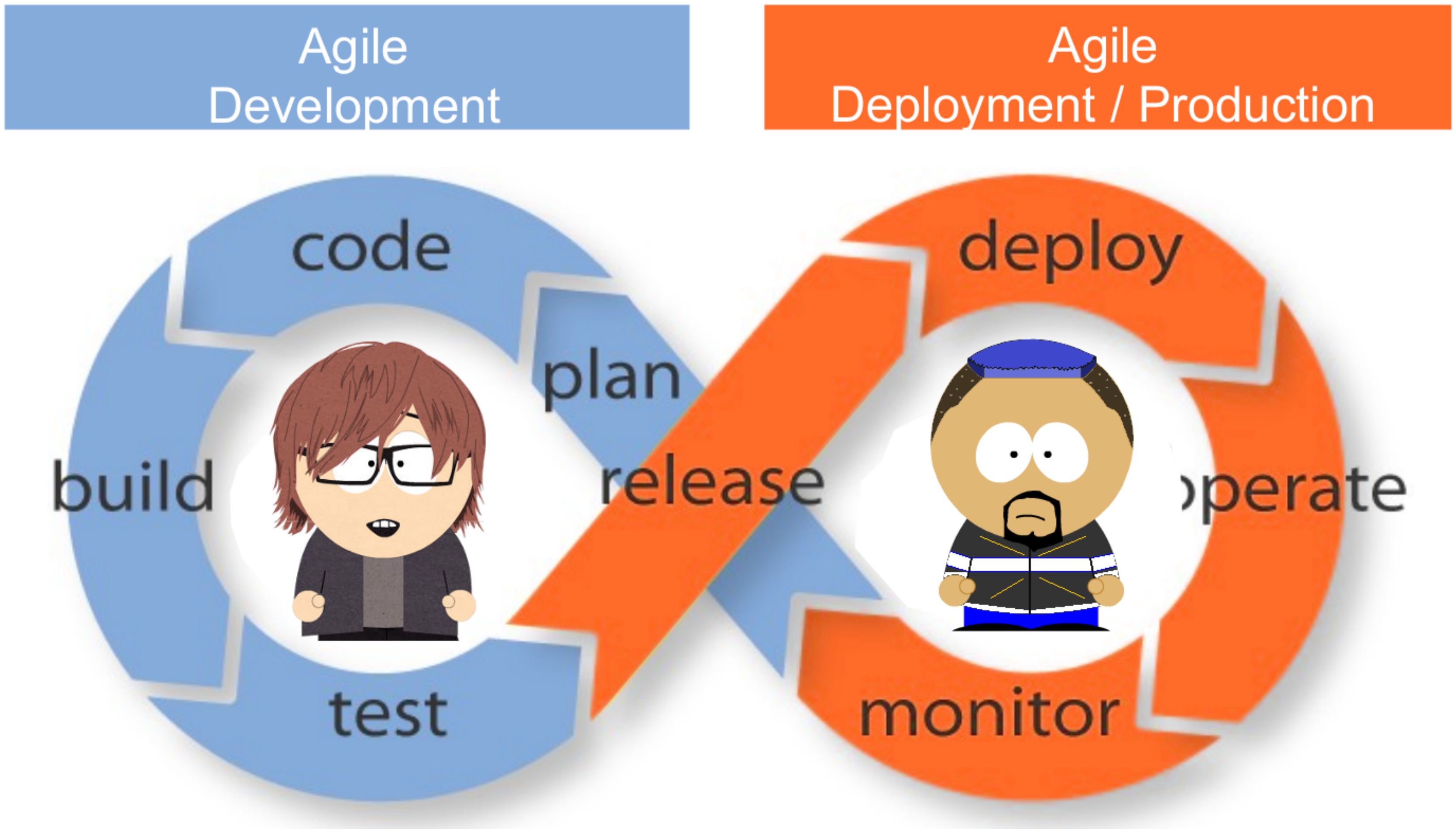
Peter wants to optimize **stability**

Tokamak operator


WORKED FINE IN DEV
OPS PROBLEM NOW
memegenerator.net

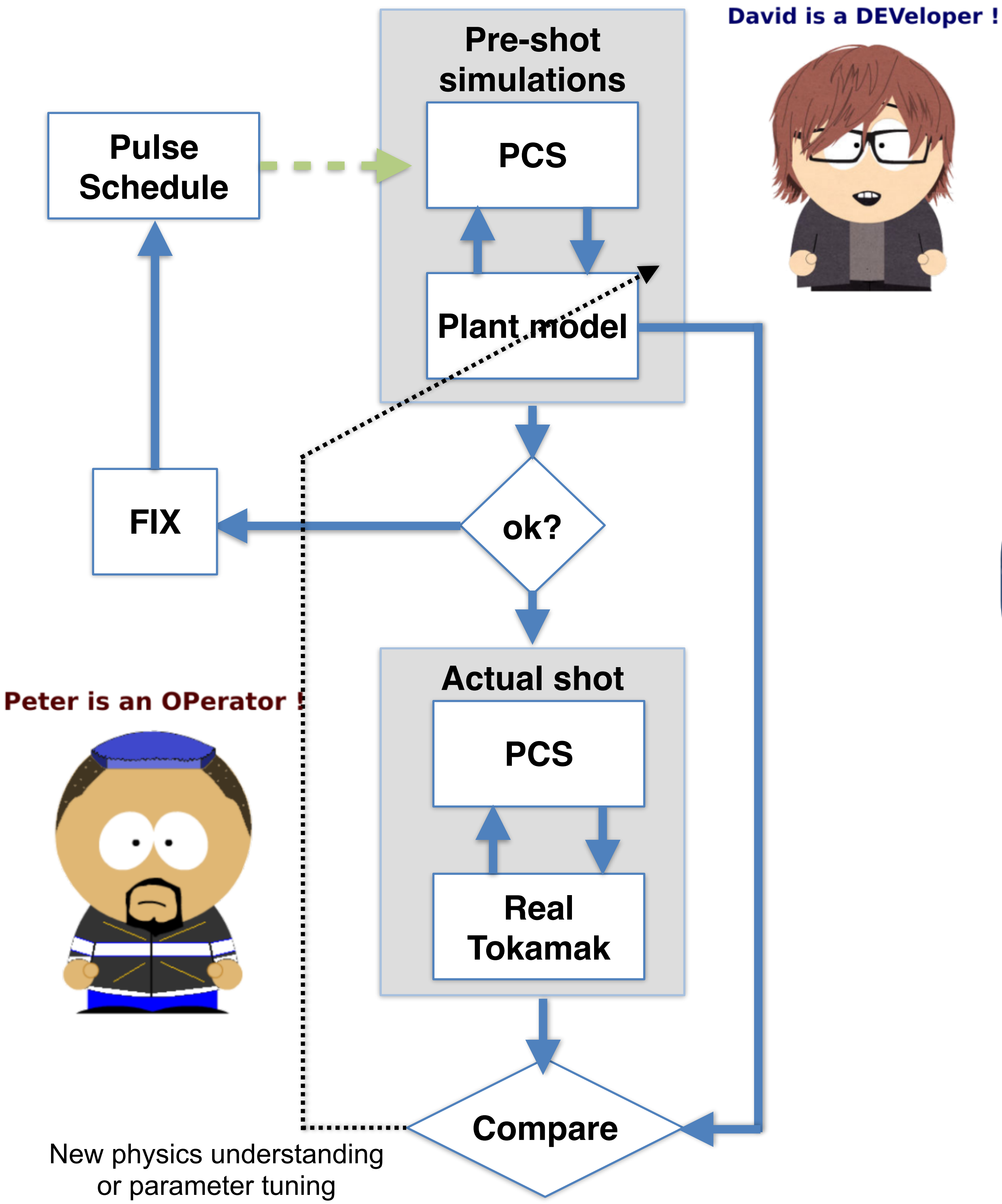
The DevOps solution

From: https://www.devops.ch/2017/05/10/devops-explained/

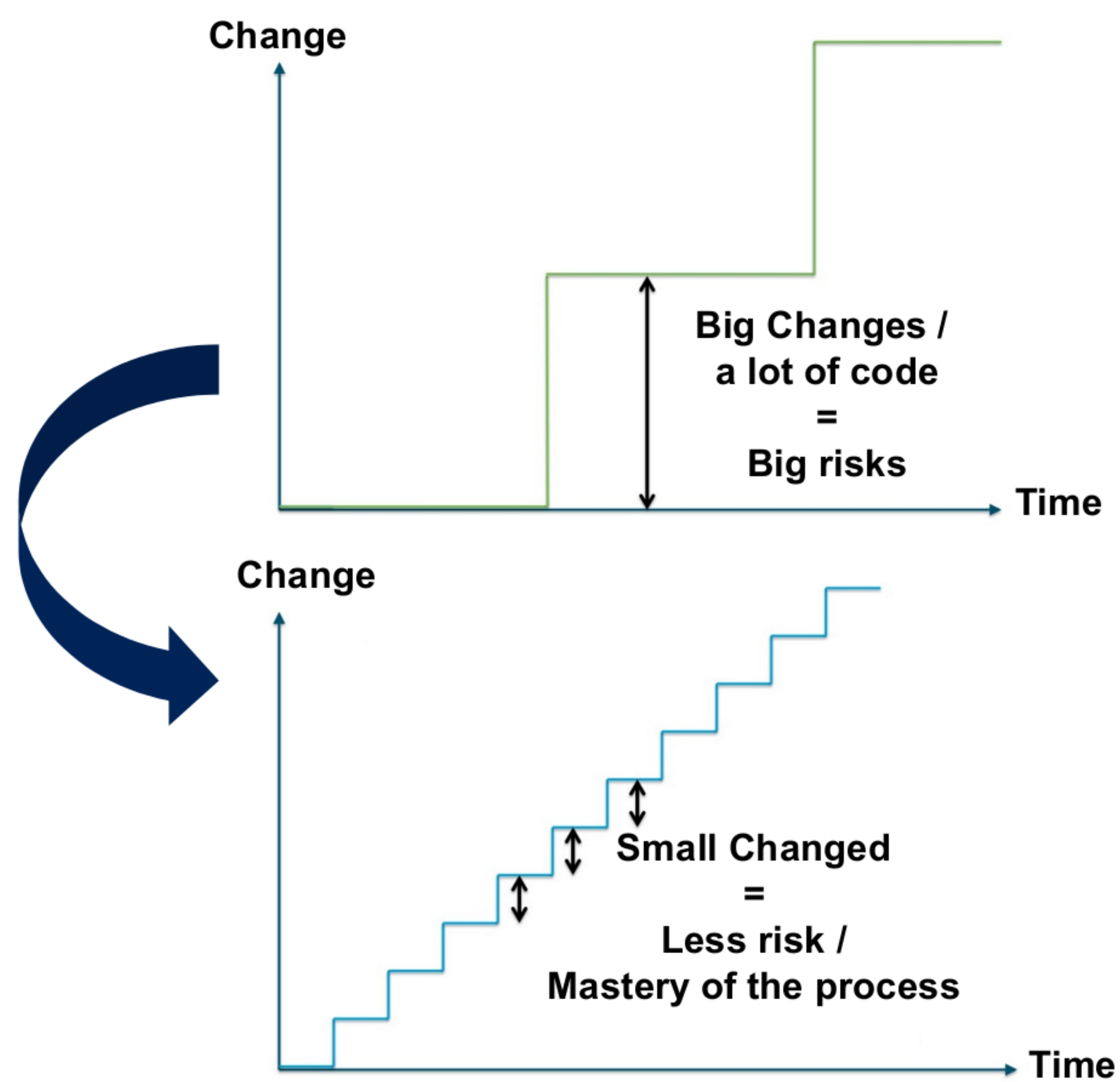


- **Dev: Automate (to the extent possible) all testing and deployment**
  - **Continuously test and deploy new software**
- **Ops: Provide platform for dev as close as possible to the real thing**
  - **The real-time control software environment + the models on which to test**
- **Run through this loop frequently**

Promote frequent, rapid, small iterations



“If it’s hard, do it more often”



Outlook on software engineering aspects

- Controllers and models are ultimately software projects
  - Transition from demonstrations or in-house tools to ‘production’ - level codes
  - Role of open-source? -> leverage power of the community
- Software industry has developed methods for harnessing large collaborative software projects
  - Culture in fusion community has lagged behind, but is catching up
  - Promote this culture and educate ourselves on best practices / tools
- Essential role of software ‘digital twins’ for future tokamaks

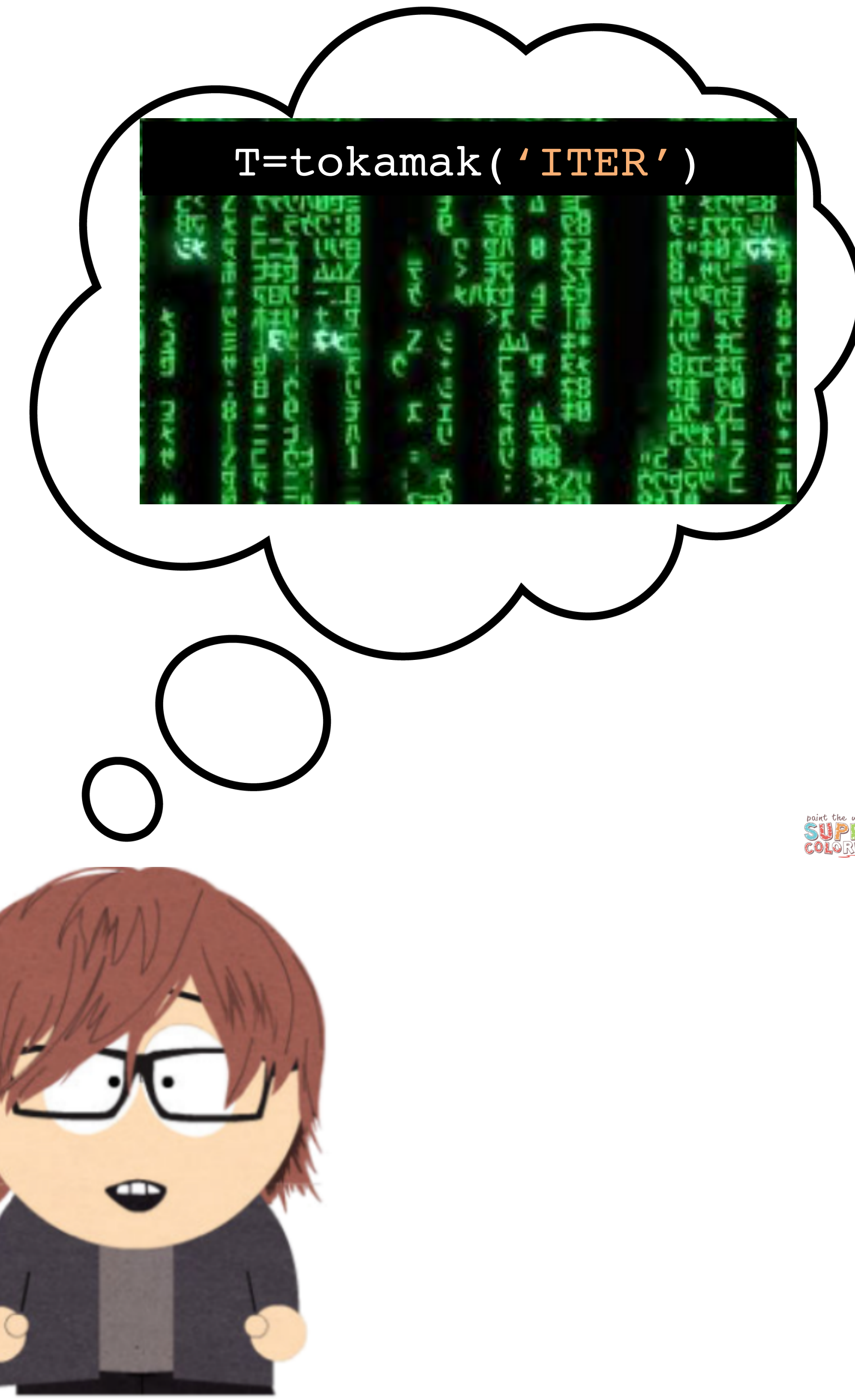
Physics view **Control design view**

T. Todd, in R. Dendy Plasma Physics p. 448 (1993)

Physics view

Control design view

Software Engineer view



T. Todd, in R. Dendy Plasma Physics p. 448 (1993)

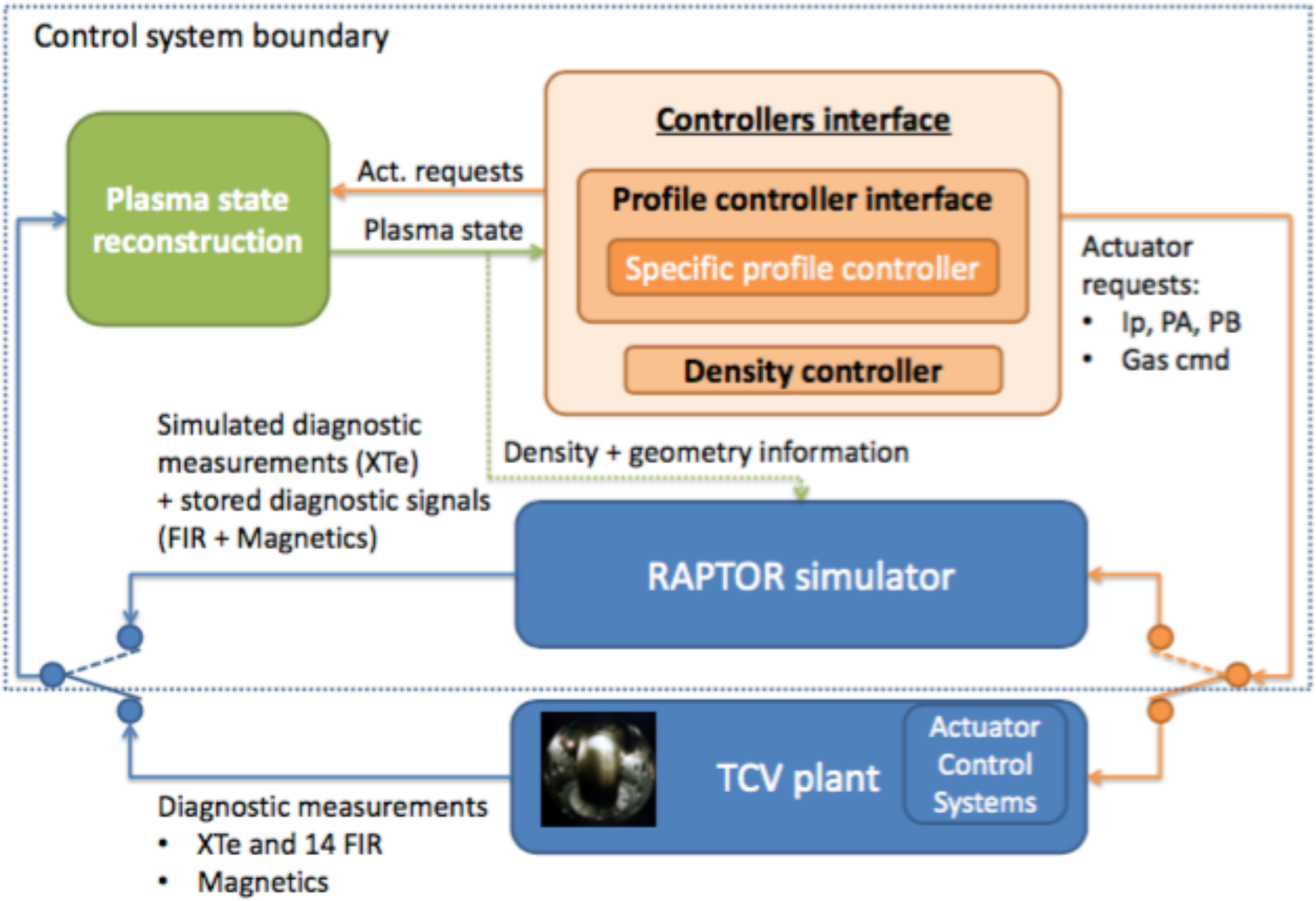
EPFL Graduate Course - Control & Operation of Tokamaks

Next edition: February 2023

contact federico.felici@epfl.ch for more information

Backup slides

Implementation of q profile + β control on TCV including plasma state reconstruction.



[E. Maljaars et al., Nucl. Fusion, vol. 57, no. 12, p. 126063, Dec. 2017]

TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

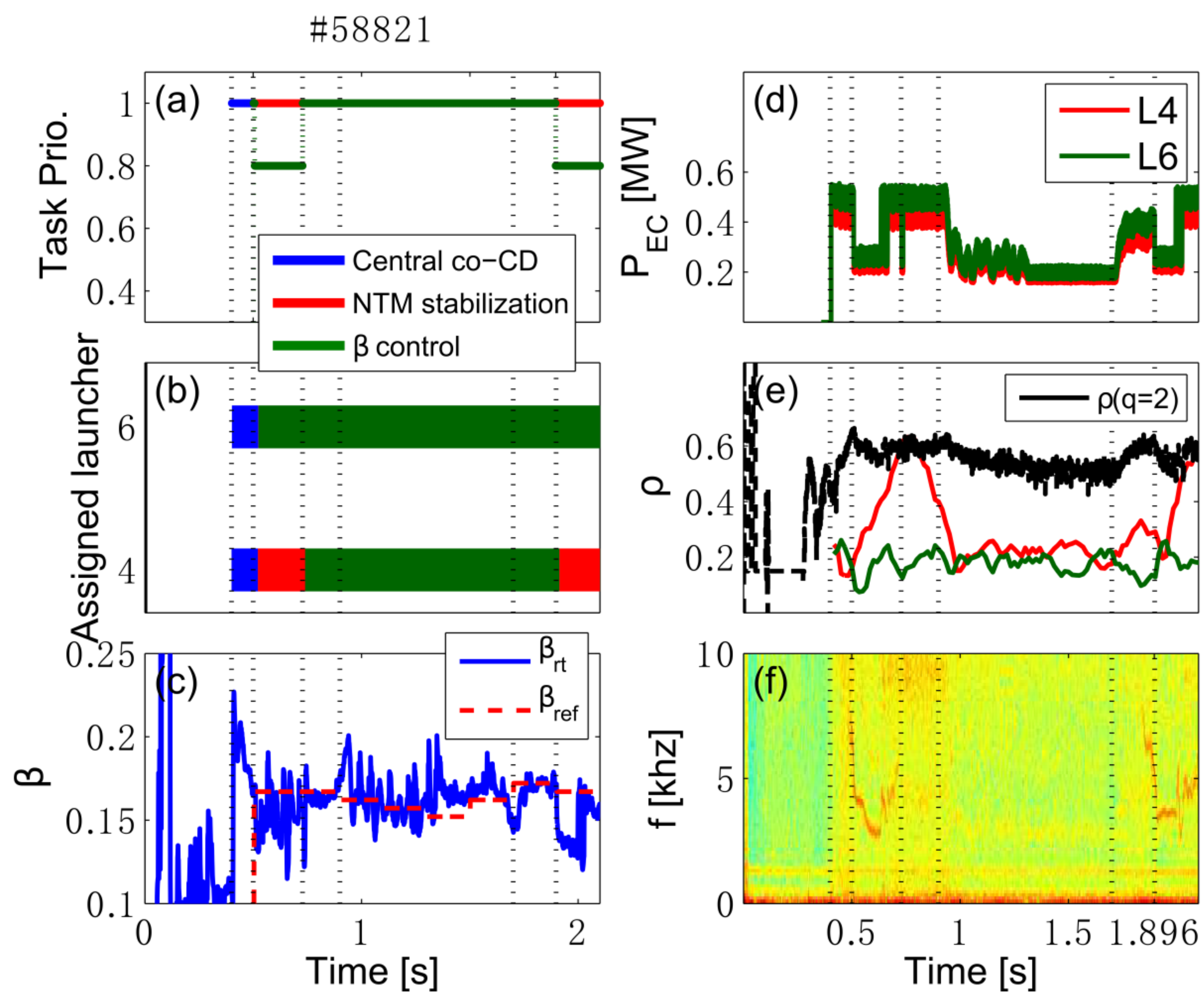
3 Tasks:

Task name	Activation
Central co-CD	[0.4s-0.55s]
2/1 NTM stabilization	[0.5s-2.5s] +NTM presence
β control	[0.5s-2.5s]

2 Actuators:

Actuator name	Type
EC launcher L4	co-CD (0.5MW)
EC launcher L6	co-CD (0.5MW)

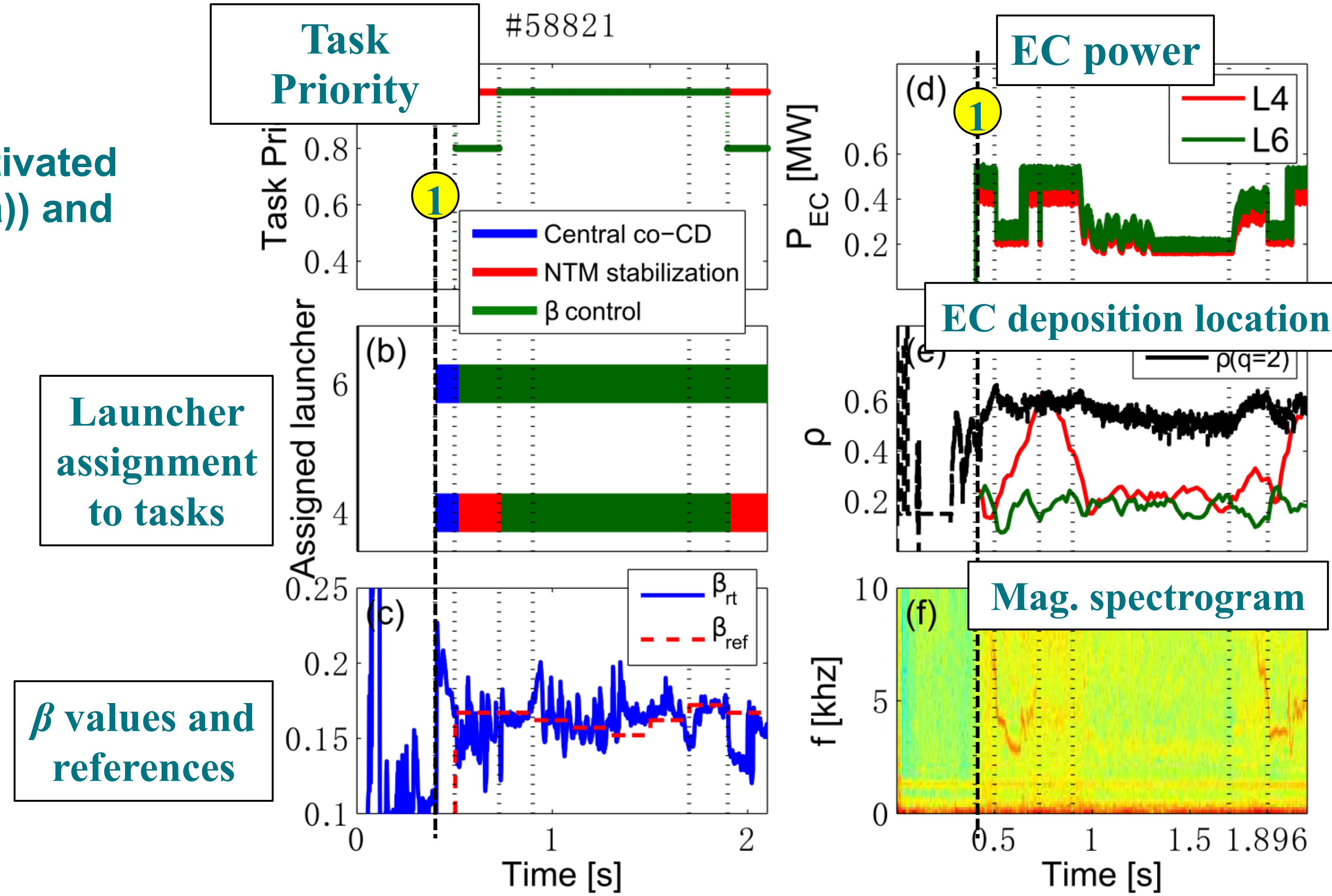
For more details:
[T. Blanken Nucl. Fus. 2019]
[T. Vu Fus. Eng. Des. 2019]



TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

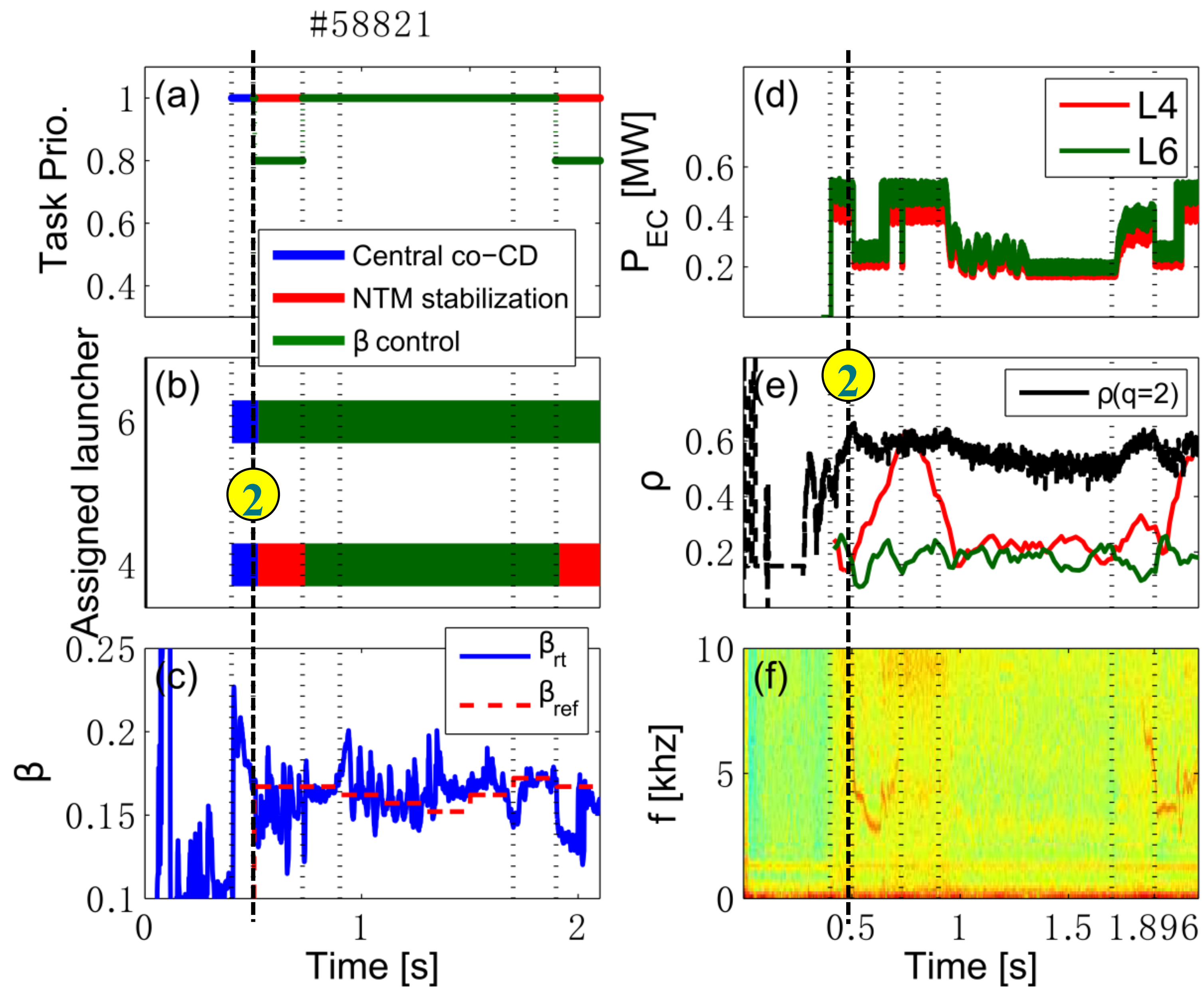
1

Central co-CD is the only activated task, gets priority 1 (panel (a)) and L4 and L6 (panel (b))



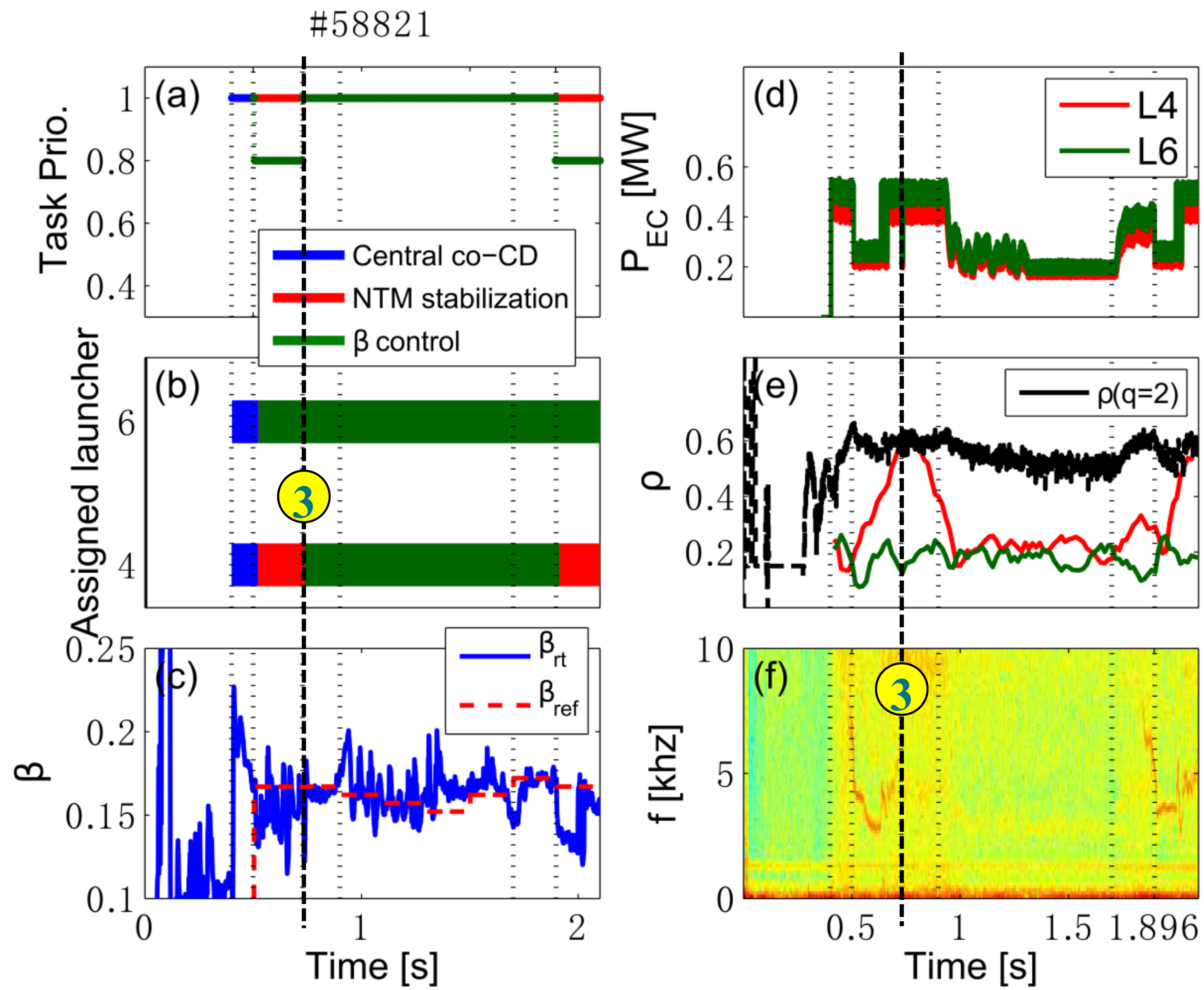
TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

2 2/1 NTM onset (panel (f)), NTM stabilization takes priority 1, requests 0.5MW and gets L4
β control is activated as well, requests 1MW, but gets only the remaining L6 due to its lower priority



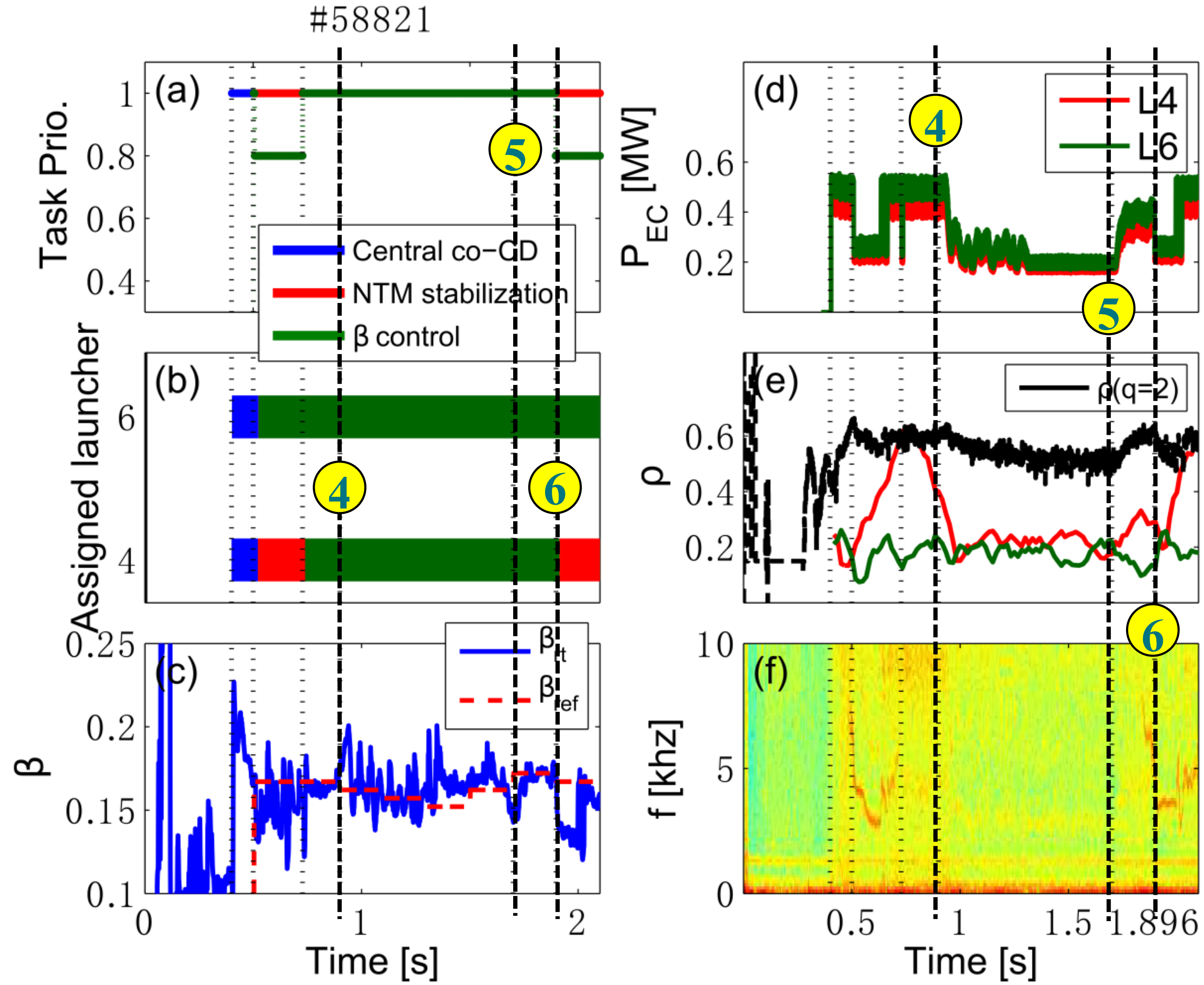
TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

3 **NTM stabilized, β control task takes priority 1, gets L4 and L6**



TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

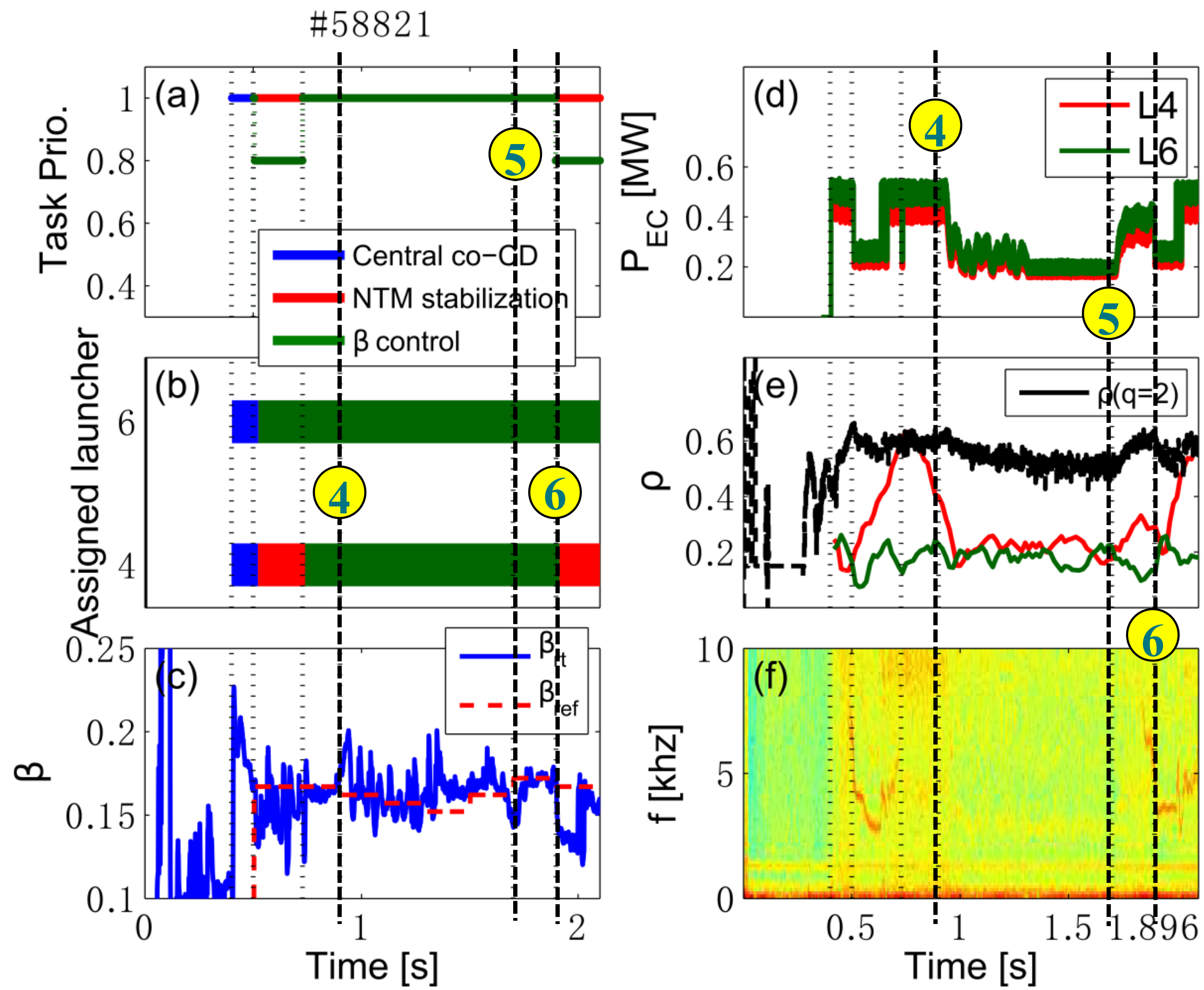
4 - 5 **β control only, with both L4 and L6**



TCV example: simultaneous NTM stabilization and β control with real-time task prioritization

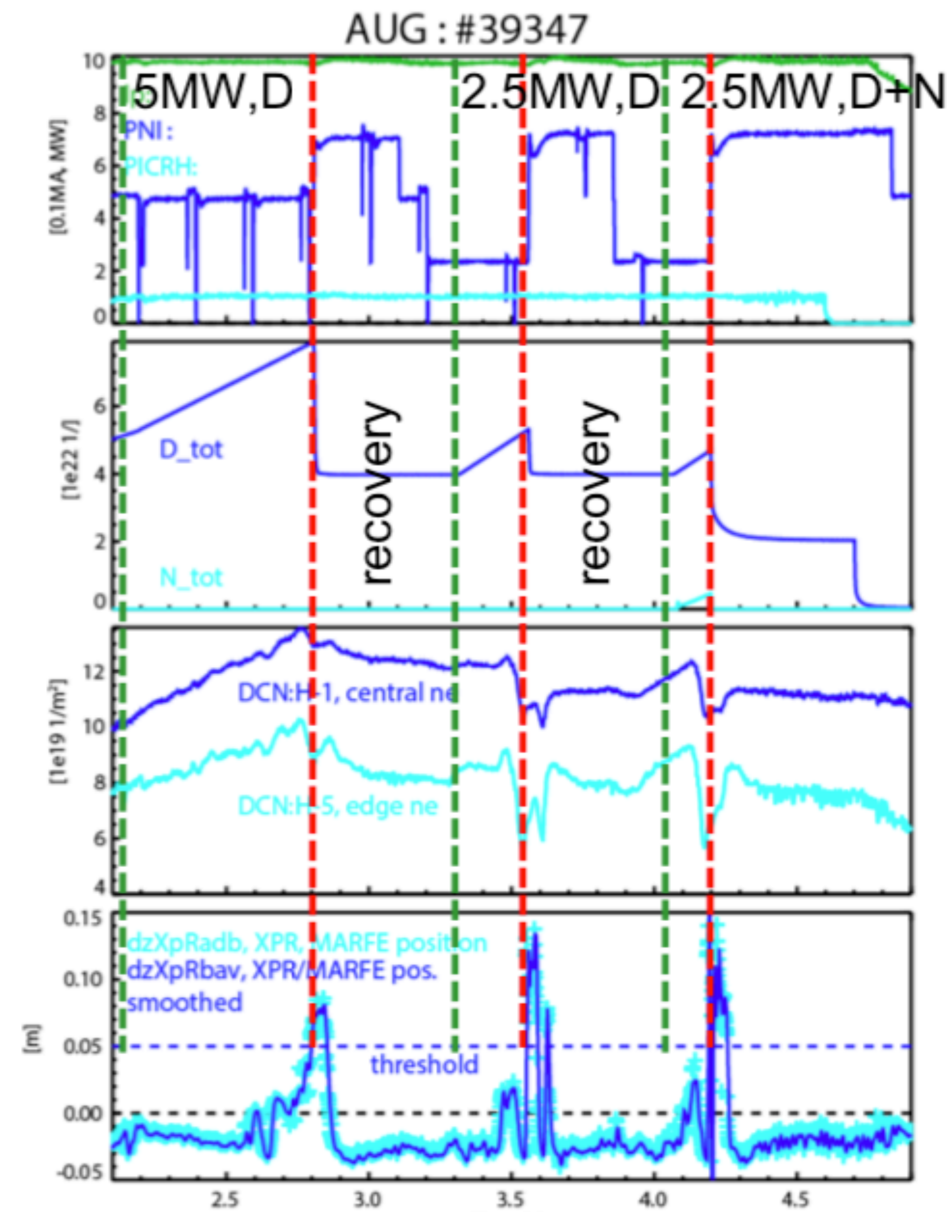
4 - 5 β control only, with both L4 and L6

6 NTM is detected and NTM stabilization takes priority 1



Asynchronous response - intervene when threshold is exceeded

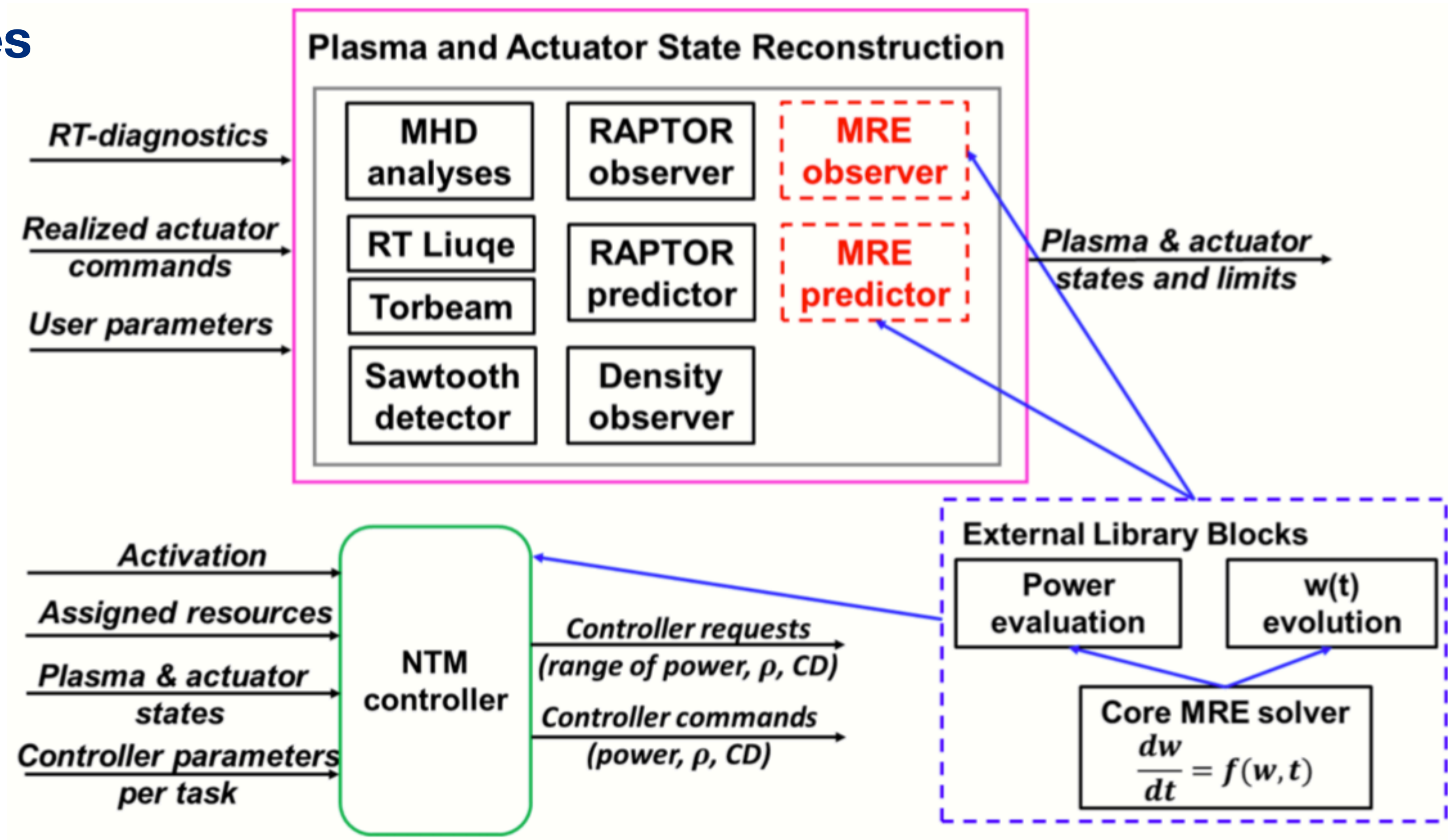
- **Deviate from 'nominal' scenario to 'recover' the discharge**
  - Should catch 'most' of remaining 1% cases
- **Detect and track multiple events simultaneously**
- **Need to track various events:**
  - Exceeding of limits related to proximity control
  - (N)TM presence / locked modes
  - Sawteeth, Minor disruptions
  - ELMs, Impurity influx
  - MARFE onset
  - (Real-time detectors needed for all these quantities..)
- **Respond by targeted recovery actions, or ramp-down**
- **Leave as few cases as possible for DMS triggering**



Repeated recovery of discharge based on MARFE position monitoring, acting on gas & heating [B. Sieglin, M. Maraschek, M. Bernert ASDEX Upgrade]

Outlook: towards resource-aware NTM control

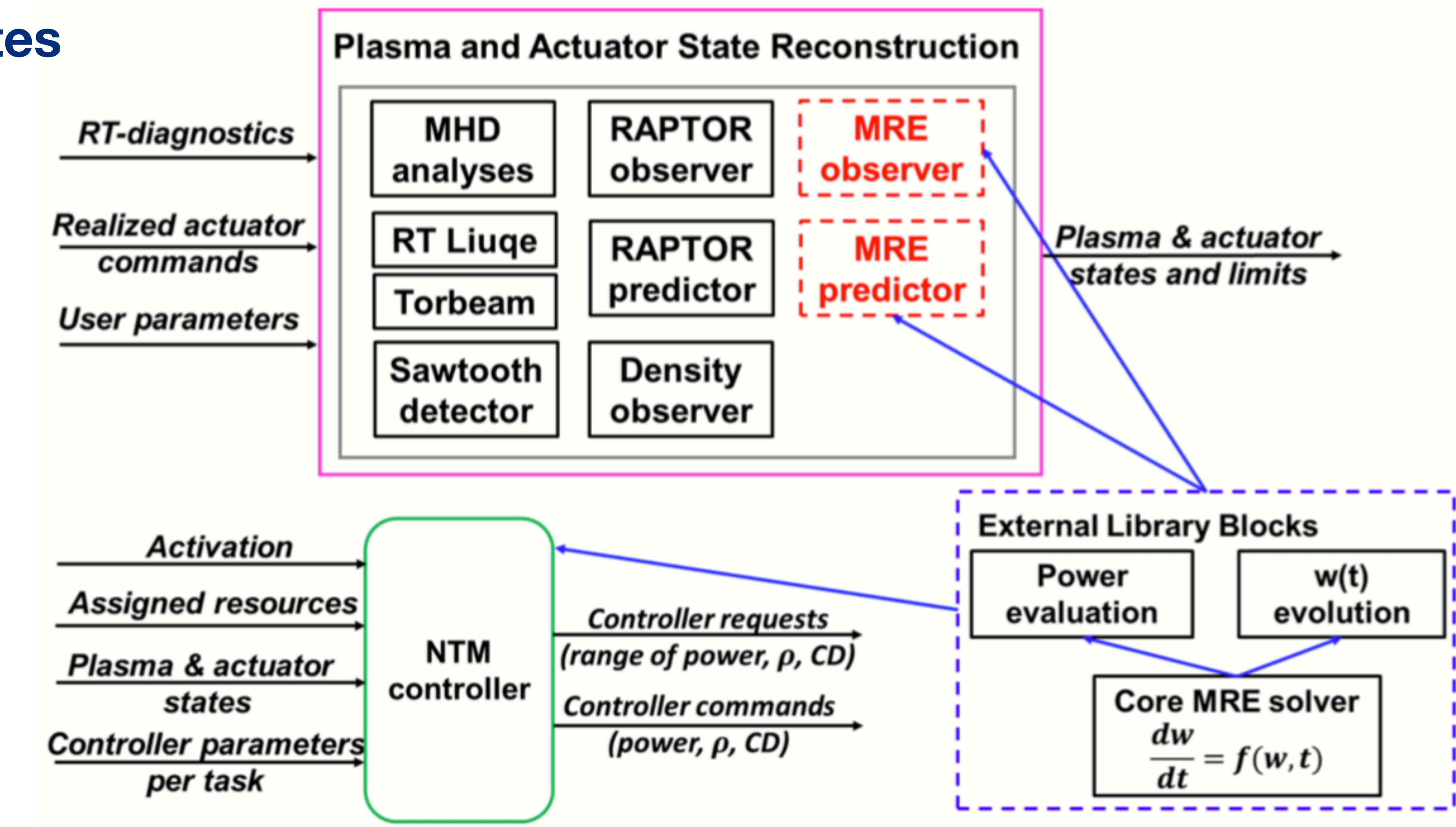
- **First: Modified Rutherford Equation (MRE) model for $w_{NTM}(t)$**
  - **Including empirical $\Delta'(w)$ for TCV.**
  - **Reproduces island width evolution $w(t)$ from $w=0$ to $w=w_{sat}$**
    - **[M. Kong, NF 2019]**
- **Solving MRE in PCS - resource-aware NTM controller**
  - **Estimate required power & deposition location for NTM preemption**
  - **Estimate required power for NTM suppression**
  - **Continuously update estimates based on plasma state**



Outlook: towards resource-aware NTM control

- First: Modified Rutherford Equation (MRE) model for $w_{NTM}(t)$
  - Including empirical $\Delta'(w)$ for TCV.
  - Reproduces island width evolution $w(t)$ from $w=0$ to $w=w_{sat}$
    - [M. Kong, NF 2019]
- Solving MRE in PCS - resource-aware NTM controller
  - Estimate required power & deposition location for NTM preemption
  - Estimate required power for NTM suppression
  - Continuously update estimates based on plasma state

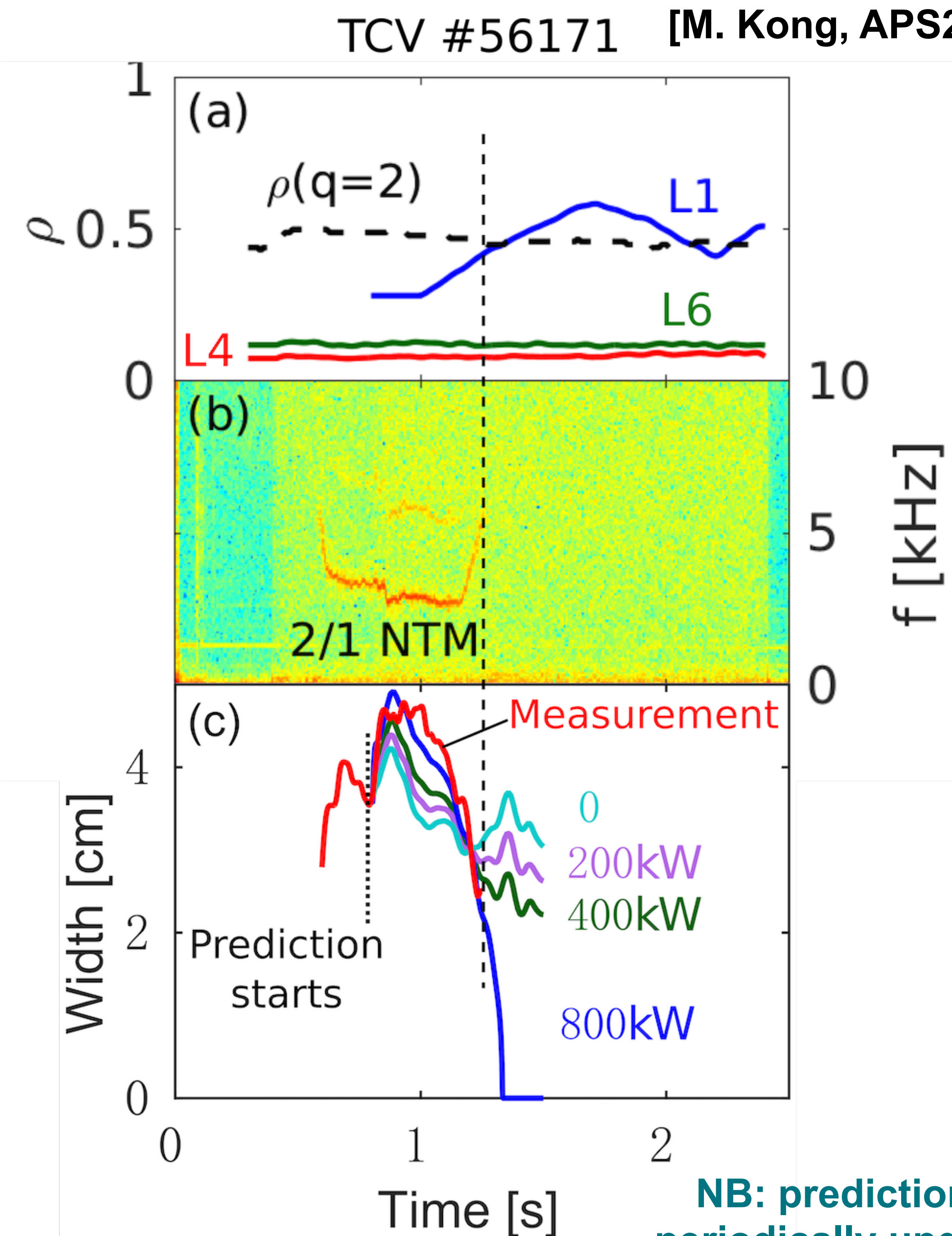
No need to change PCS architecture,
just add new components in the right place.



Simulation of real-time MRE-based control of NTMs: continuously predict $w_{ntm}(t)$ evolution

- **TCV experiment:**
  - **Sweep 800kW EC beam across q=2 surface.**
  - **NTM stabilized when ρ_{dep} crosses $\rho_{q=2}$**
- **Simulation using MRE model:**
  - **Predict w(t) time evolution for different EC power levels.**
  - **Predicts NTM stabilization at expected time for this power level.**
  - **Predicts that lower power would not have stabilized the mode.**

[M. Kong, APS2019]



NB: prediction is periodically updated!