Fusion Engineering and Design 159 (2020) 111764

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

The MAST Upgrade plasma control system

Graham McArdle*, Luigi Pangione, Martin Kochan

UKAEA-CCFE, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK

ARTICLE INFO

Keywords:
MAST Upgrade
Plasma control
Software architecture
Multiple actuators
Virtual actuator
Functional chain

ABSTRACT

The plasma control system (PCS) for MAST Upgrade inherits most of the original MAST hardware / software architecture, which was based on an architecture developed by General Atomics. Whilst the digital control hardware has already had a mid-service upgrade on MAST, some additional I/O was required to support the numerous additional input signals, poloidal field (PF) coils and gas channels on MAST-U. The generic software infrastructure from General Atomics has been retained for MAST-U but the tokamak-specific algorithm software has been substantially re-written to support the additional capabilities of MAST-U, especially in the areas of gas injection control and coil current control. The software structure presented here has been designed to provide maximum flexibility to exploit the new features of MAST-U whilst maintaining a manageable degree of complexity in the operation of the system.

1. Introduction

MAST has been upgraded with, among other things, many new divertor coils and many gas injection valves compared to MAST [1]. The plasma control system (PCS) system is still based on the General Atomics (GA) infrastructure [2] as reported in [3] but has undergone a mid-service hardware upgrade to use an Intel-based server, PCIe to cPCI chassis and D-tAcq I/O hardware. For MAST-U we plan to use the same hardware platform and keep the General Atomics PCS infrastructure, but the tokamak-specific control algorithms needed to be replaced to handle the extra complexity.

Section 2 describes the new PF coils and gas actuators and the common challenge presented by their complex requirements. Section 3 describes the key aspects of the GA PCS architecture that influence our design choices, followed by the architectural design pattern chosen. Section 4 describes the general implementation of this design pattern for both PF control and gas control. Section 5 details how gas control is uniquely managed in MAST-U to provide the maximum configuration flexibility possible.

2. The new actuators for MAST-U

The upgrade to MAST features many changes, but for the purposes of this paper we focus mainly on two substantial areas: poloidal field coils and gas injection. In both cases the common concern for PCS development was the need to be able to execute multiple simultaneous control tasks where each task requires a different mix of multiple actuators, without inconsistency or conflict in actuator assignment.

2.1. PF coils

For MAST-U, the following changes were made to the PF coil system (see Fig. 1):

- P1 solenoid replaced with one of enlarged radius to double the flux swing and accommodate thicker TF limbs.
- Solenoidal Pc coil added to flatten the inner shape of the plasma, powered by the former P2 power supply.
- P4 and P5, the up/down symmetric coil pairs for combined equilibrium and shaping field, have been retained but P5 coils have been shifted toward the midplane to allow space for off-axis Neutral Beam injection.
- New P6 coil, an up/down antisymmetric coil pair for vertical stability and position control. Passive stabilisation rings have also been installed and the power supply has been replaced with a very fast IGBT system controlled by an FPGA device (to allow much faster control loops than PCS can handle). This means that PCS only provides configuration data to a separate vertical position and stabilisation control system that is decoupled through up/down symmetry from shape and radial position control.
- 8 new up/down symmetric pair coil sets (D1, D2, D3, Dp, D5, D6, D7, Px) installed in locations around the divertor to support exploration of various divertor configurations including Super-X Divertor (SXD). These are powered by new fast-switching IGBT

* Corresponding author.
E-mail address: graham.mcardle@ukaea.uk (G. McArdle).

https://doi.org/10.1016/j.fusengdes.2020.111764
Received 9 August 2019; Received in revised form 13 May 2020; Accepted 14 May 2020
Available online 16 June 2020
0920-3796/

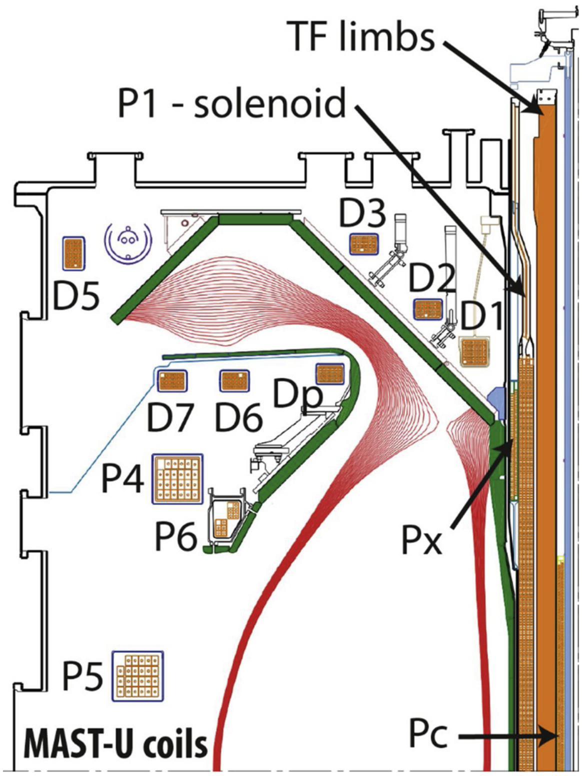

Fig. 1. MAST-U upper left vessel cross-section showing PF coil locations.

power supplies.

Since many new power supply command outputs were required and the existing equipment for transmitting voltage commands to the remote power supply buildings was obsolete, we designed and built new 4-channel "UDP DAC" devices. These contain an FPGA incorporating an IP core to hardware-decode Ethernet packets. By configuring the offset address of each device, all remote power supply voltage commands can be sent in a single UDP broadcast packet to all UDP DACs and each one outputs the 4 voltage commands found at its programmed offset within the command packet. This provided an efficient way of driving multiple remote power supplies with low latency.

It should be noted that whilst each PF coil has a particular primary purpose, their close proximity to each other and to the plasma chamber means they have a significant cross-coupling of their respective control influences. Even the solenoid produces enough stray field to require significant current in other coils to compensate its effect on plasma shape and divertor strike-points. Therefore, the control of each plasma shape parameter involves a different combination of many of these coils rather than simply allocating one coil to one “gap”. This introduces a high level of complexity and difficulty of understanding for the operator to successfully implement control of a given parameter without impacting many others.

2.2. The gas injection system

MAST-U was designed to provide high flexibility for gas injection, to allow for various experiments with changing the fuelling location and/or gas puffing for studying the impact on plasma performance and dissipation in the divertor, detachment control, radiation control, etc.

To meet these requirements, the system is comprised of 11 groups of valves, where each group provides a toroidally symmetric ring of valves injecting at each of the marked locations in the poloidal cross-section shown in Fig. 2.

The group locations have up/down symmetry but the High-Field Side midplane (HFS-m) valves are so close together that they are treated as a single group rather than an up/down pair of groups, hence 11 groups in total. Each up/down set of valves is supplied from one of 6 separate gas plena, each of which can be filled with any gas species available from the gas bottle rack connected to a shared gas fill manifold. The total number of gas injection valves in the system is 48.

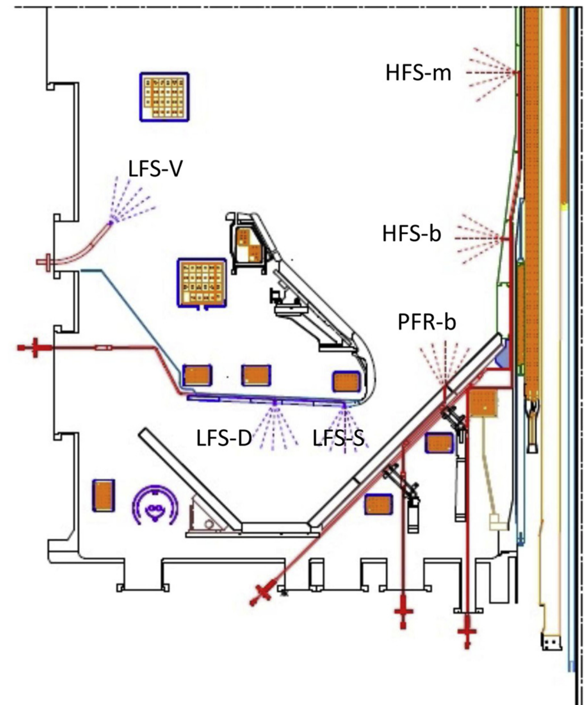

Fig. 2. MAST-U lower left vessel cross-section showing gas injection locations.

The design requirement was to be able to carry out any gas injection function from any group or weighted set of groups whilst other groups can be used simultaneously for other functions. This demanded a high level of flexibility in the software architecture whilst also posing the challenge of excessive complexity.

3. PCS software architecture

3.1. The General Atomics PCS framework

The MAST-U plasma control system is still based on the General Atomics PCS software architecture [2] that is used on DIII-D [4], NSTX [5], EAST [6] and KSTAR [7] tokamaks. It consists of a generic framework infrastructure shared by all tokamak installations, upon which the platform-specific and tokamak-specific code for each deployment is added. The infrastructure framework defines the concept of categories, which contain sequences of phases, to which algorithms relevant to that category can be attached, together with the configuration data specific to that phase and the chosen algorithm. More details can be found in [2] but the essential point for architectural design is that the top level of scenario design is the category. Therefore, the timing of what is executing in each category is independent of others. A fictitious example of this is shown in Fig. 3. This shows how each category can be used as a

category	pulse duration			
System	(Single immutable function and global data)			
PF control	pre-magnetise	breakdown	ramp-up	flat-top
Gas control	inactive	prefill	density control	density + XYZ
Other	inactive	algorithm A	algorithm B	algorithm A

Fig. 3. Representative category execution timing diagram for a fictitious example PCS installation.

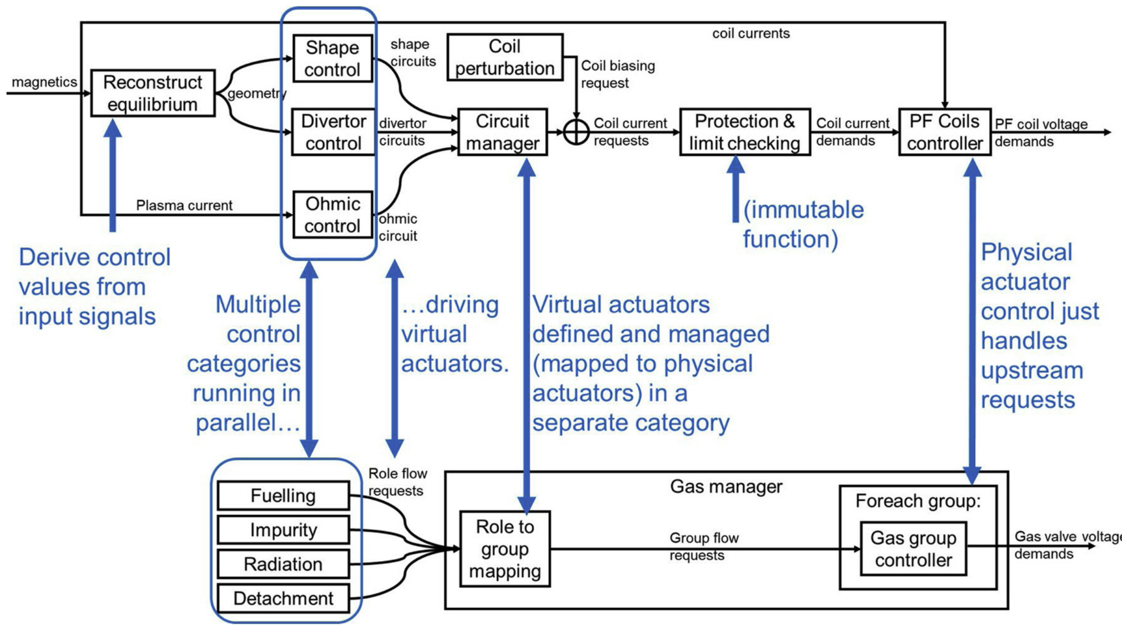

Fig. 4. Functional flow diagrams for PCS categories involved in (upper) PF coil control and (lower) gas control.

placeholder for execution of interchangeable functions, where the choice of phase to execute at any time defines (via the attached algorithm) what functions will be executed. It also shows that the category acts as a unit of concurrency, since only one algorithm can be active at any time, therefore multiple categories are required to be able to execute multiple tasks concurrently.

This architectural feature demands care in the use of data exchange between categories, because it implies that the active algorithm on just one side of a data exchange could switch to something else independently of the other side. For the fictitious example in the figure, if the ***breakdown*** algorithm in the *PF control* category relies on data from ***algorithm A*** in the *Other* category, the software designer must consider the scenario where the *Other* category switches to a phase using ***algorithm B*** whilst the ***breakdown*** algorithm is still active in *PF control*.

3.2. Architectural choices for MAST-U

When considering how to manage the complexity of the "many actuator" problem for both PF and gas within the framework of the GA PCS infrastructure, the following design decisions were made:

Specific categories would own a specific set of actuators, becoming the definitive source of drive signals for those actuators.

This would avoid possible conflicts from multiple controllers attempting to drive the same actuator. How the actuators are driven depends on the active algorithm in that category and its configuration data.

A "functional chain" of categories have been defined that implement separate steps in the data flow sequence from measurement input to actuator output.

The ability to switch the active algorithm in one category independently of others is exploited to break down the complexity of multi-actuator control into a set of processing steps, each of which is handled by a separate category. Instead of implementing just one PF control category that tries to do everything (and likewise for gas), a modular approach is taken where separate categories are provided for each stage of the processing chain from input to output. This allows each processing stage to use any of a choice of algorithm implementations (mostly) independently of the other stages. The data interfaces between successive stages of the chain are statically defined to mitigate concerns about consistency of data transfers when the running algorithm in one category changes to something else asynchronously from other categories in the chain. Greater flexibility is possible by combining different permutations of compatible sets of algorithm choices along the functional chain than could be achieved by developing a separate all-in-one algorithm for every conceivable control permutation.

Multiple parallel control categories control separate sets of virtual actuators, which are then mapped to physical actuators via a separate virtual-to-physical mapping category

This further decomposes the complexity of what needs to be controlled and how. For each of the various simultaneous use cases for the PF coils or gas valves, separate groups of virtual actuators are defined. Each virtual actuator is an abstract control output that modifies the feature of interest to that use case. A separate control category owns the group of virtual actuators for each use case and all such control categories run concurrently, driving their virtual actuators. A virtual-to-physical mapping category defines these virtual actuators in terms of combinations of the physical actuators (PF coils or gas valves) that can provide the desired effect of each virtual actuator. Because they are in separate categories the virtual-to-physical actuator mapping can change independently of the virtual actuator control algorithms.

The next chapter shows how these design principles are implemented in each of the PF coil and gas control cases.

4. PF and gas control implementation

4.1. PF coil control

The functional chain of categories for PF control is shown in the upper half of Fig. 4. It is expected that PF coils will be used for 3 primary use cases: ohmic heating (loop voltage or plasma current), plasma boundary (position and shape) and divertor leg control. These have been split into separate categories, each of which is allocated a set of virtual actuators in the "circuit manager" category. In most cases the plasma boundary and divertor leg properties will not be directly measured but will require an upstream reconstruction algorithm. This is also put in its own category to allow flexibility in the choice of implementation of reconstruction independently of control. The currently available algorithm is the Local Expansion MAST Upgrade Reconstruction (LEMUR) code, which uses local flux extrapolation methods [8,9] to locate the plasma boundary and divertor legs.

The virtual actuators for PF coils are "virtual circuit currents". Each virtual circuit is defined by a column vector of physical coil currents with set proportions, where the combination of coil currents is designed to uniquely control one plasma parameter of interest and maintain reasonable decoupling from the others. The column vectors of each virtual circuit can be assembled into a transformation matrix[1] that maps the vector of virtual actuator requests to a vector of coil current requests. This allows each controller to own its own virtual actuators for its own control functions, but the virtual circuit manager combines them all to just one set of physical actuator control requests. A warning can be generated if any two virtual circuits lack orthogonality by checking the dot product of their unit vectors.

This is quite similar to how the "M matrix" is used in the Isoflux algorithm [10] that is deployed in many other tokamaks that use PCS, but the key difference here is that the circuit matrix is implemented in a separate (circuits) category from the shape controller, so it would be possible either to switch between multiple instances of "isoflux-like" algorithm without having to replicate the M matrix settings between them all or conversely to keep the same isoflux algorithm running whilst changing the "M matrix" to track changes in the response of the equilibrium to coil currents.

It was decided that the definition of virtual circuits would only be done by control experts and made available for selection at run time by the operator. This prevents the production of a plethora of variants and duplicate definitions of the same virtual circuit. The "circuit manager" in the figure is the category containing the circuit mapping algorithm and simply applies the transformation from virtual actuator output to physical coil current. Since it is separate from the control algorithms this allows the mapping to be changed during the pulse whilst the control algorithm remains the same, or the mapping function can remain the same whilst the control algorithm is changed. This allows the handling of transitions required for tracking the solenoid swing or the movement of the divertor strike point over the top of a divertor coil.

To allow for a high level of abstraction of what a virtual circuit might be controlling (e.g. a gap dimension, a geometry descriptor such as elongation, flux expansion factor, etc.), each virtual circuit definition includes not only the vector of coil current coefficients but also a type code to specify which plasma property it is to be used for. Validation checks on the pulse configuration settings ensure that the upstream control algorithm is compatible with the type of virtual actuator provided to it by the circuit manger. Likewise, the control algorithm checks its consistency with the upstream reconstruction algorithm to ensure that the property to be controlled will also be reconstructed at run time.

The units of a virtual circuit are not current but whatever is appropriate to the quantity being controlled. Thus, the reconstruction algorithm produces a measurement of gap, flux, strike point etc and the controller for this quantity simply applies a control law to demand a rate of change of this quantity. The virtual circuit for this quantity is defined in terms of the currents required in each coil to change that quantity by one unit. Therefore, the application of the virtual to physical mapping function also transforms units of output to a demand for changes in coil currents.

It should be noted that all controllers operate in terms of 'delta change' rather than absolute values. The circuit manager therefore also must integrate up the history of change requests to produce a set of absolute coil current requests (actually both delta and absolute coil currents are passed to the coil controller to provide full state information)

There is also a coil perturbation category added to provide the ability to add bias / trim signals directly to individual coils for the purpose of developing new virtual circuits or other ad hoc requirements.

Note that consistent coil control protection is maintained regardless of active upstream controller by passing all coil current requests through an immutable protection function belonging to the system category. This validates and limits absolute coil current requests to avoid entering an operating region that would trigger the plant realtime protection system to stop the pulse.

4.2. Gas valve control

The function chain of categories involved in gas valve control is shown in the lower half of Fig. 4.

The virtual actuators for gas are "role flow requests". However, unlike coil currents, each gas group can only be assigned to one role due to some significant departures from the "virtual circuit" analogy.[2]

There are separate control categories for each role type: Fuelling, Impurity, Radiation and Detachment, with multiple virtual actuators for each category (except fuelling, since this is used to control the plasma density, a singular term).

A role request can be shared across multiple gas groups with a defined sharing ratio. In the case of gas control the "gas manager" is a single category handling both the mapping of virtual "roles" to physical gas groups and also the distribution of group flow to the individual gas valves in each group (A separate category could be set up to do the latter but there's no expected requirement to change the function separately). This flexible handling of gas control capability is detailed in the following chapter.

5. Gas control detailed case study

Fig. 5 shows in more details the configuration capability of the gas management in PCS. The virtual actuator management is achieved primarily by the GUI item shown in the top centre. Each of the 11 gas groups can be allocated to any of the defined role flow request virtual actuators with a percentage sharing option. If more than one is allocated to the same role, the percentage contributions to that role are validated to ensure they add to 100 %. The upstream control algorithm in the category that can drive that virtual actuator will simply request a flow rate for that "role", and it is divided across the allocated groups for that role by the stated percentages. Since the gas management is separate from the control algorithm, it's possible to simply create additional phases in the gas category timing to redefine which gas groups fulfil a certain role at any given time. This can be used for example to maintain a constant fuelling control algorithm for the whole pulse whilst choosing different fuel injection locations for the prefill, start up, limiter and divertor phases of the plasma. Other categories such as radiation and detachment are each given two role flow virtual actuators to control so that the upper and lower divertor locations can be controlled independently by the same algorithm. The control categories are shown in the lower left of the figure, each driving the role flow virtual actuators into the mapping function provided by the gas category to convert these requests to group flow requests as shown.

[1] In fact, these column vectors typically originate by extraction from a control matrix that was obtained by inverting the 'sensitivity' matrix that is obtained from modelling of the influence of coil current perturbations on the plasma parameters of interest in the linear space around a chosen equilibrium scenario.

[2] It is neither intended nor likely possible to build a "sensitivity matrix" for the gas valves for each control function and then "invert" this matrix to produce a set of orthogonal gas control circuits. Indeed, the first stumbling block is that gas valves can only add gas, not remove it. Hence the intent of the gas mapping function is to provide flexibility to, for example, migrate the fuelling from 100% HFS-m group to a 50:50 split of HFS-t/b (top/bottom) groups and then to a 50:50 split of the PFR-t/b groups as the plasma evolves from early formation to limiter to full divertor, or to transition the detachment fuelling from LFS-S to LFS-D as the divertor strike point evolves toward higher radius, again possibly with a 50:50 upper/lower split. Most practical gas applications will therefore have very sparse "matrix mappings", especially due to the segregation of roles by gas species and it was considered more manageable and meaningful to constrain the gas group assignment in this way.

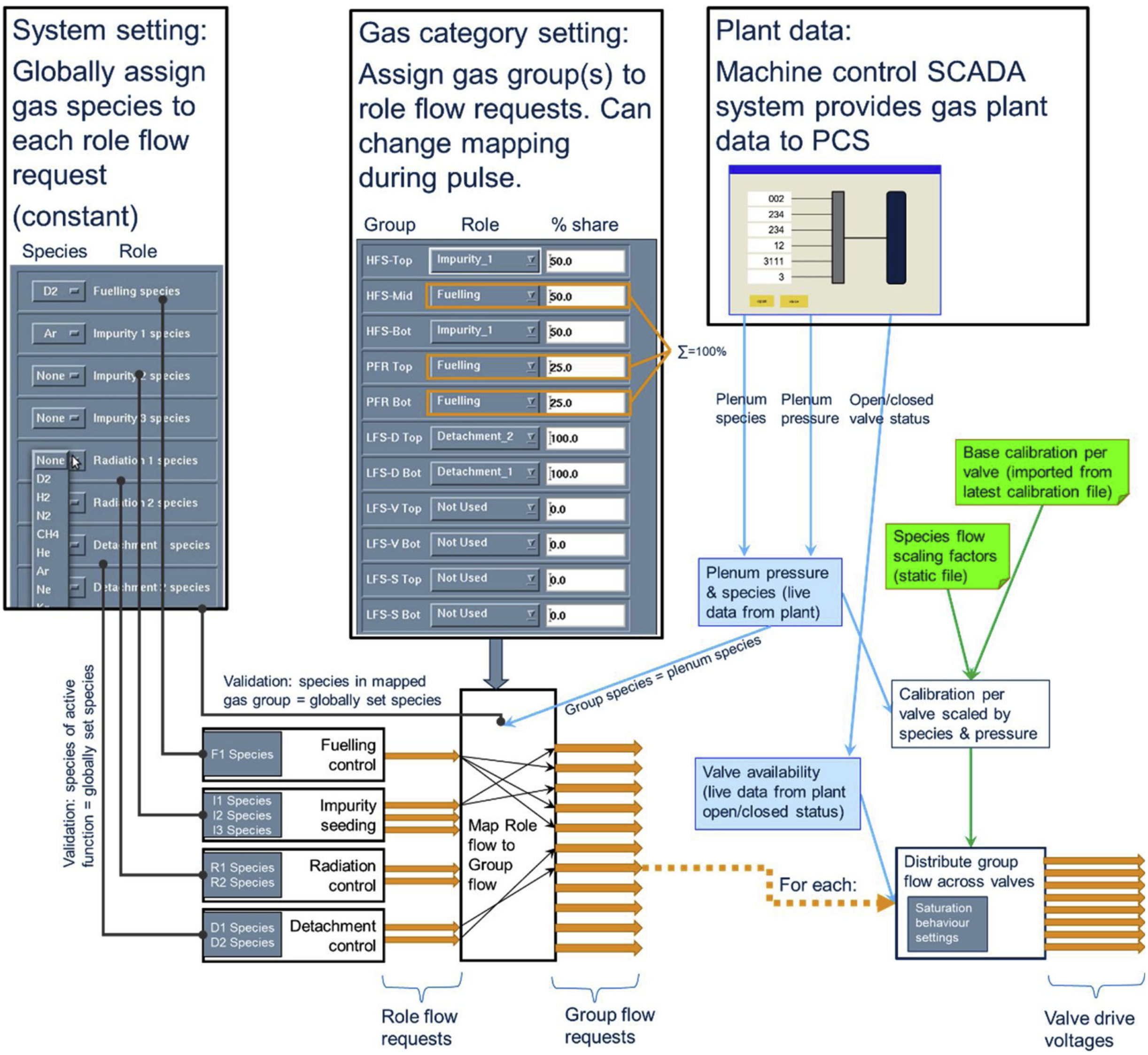

Fig. 5. Diagram showing the detailed information flow in the gas control functional chain.

Since it is possible to switch the allocation of gas groups to different roles during a pulse and it's also possible to switch control algorithms for each role type during the pulse, strong validation logic has been put in place. A global setting is made in the system category (upper left of the figure) to define which gas species will be assigned to each role virtual actuator that is to be used in the pulse. This can only have one definition for the whole pulse because the system category is required to be constant. Each phase of the role control categories is required to declare what species the active algorithm presumes to use for that role, and it is cross-checked against the global setting to ensure consistency. This is particularly useful if cherry-picking a partial restore of settings from a previous pulse into just one category or phase of the current pulse, where one may accidently copy in a controller selection that was originally tuned for a different gas species.

PCS also receives live gas plant state data from the machine control Supervisory Control And Data Acquisition (SCADA) system, including which gas species is present at what pressure in each of the 6 gas plena. By knowing which plenum supplies each gas group it therefore knows the gas species currently physically present in each gas group. This is cross-checked in the role to group mapping assignment to ensure that the gas species in the gas group allocated to a role is the same as that in the global setting of gas species for that role for all phases of the gas category.

Having achieved a validated mapping from roles to gas groups, the requirements for gas role control design are much simpler: one can focus on design of an algorithm for density control, detachment control or radiation control where the output is simply a drive to a virtual "fuelling gas flow" or "detachment gas flow" or "radiation gas flow" actuator without having to consider the implementation of that flow request in the gas valves.

The second part of the gas algorithm is to convert the gas group flow request into individual gas valve voltage requests for each valve in the group. This requires several additional pieces of information. First, each gas valve has its own calibration coefficient of flow rate per applied volt for a standard deuterium gas pressure. This needs to be scaled according to the actual gas pressure (as reported from the SCADA plenum readings) and gas species (read from a static file containing species scaling coefficients). The gas plant state from the SCADA system also identifies which of the valves in the group have been isolated, e.g. due to a leak or a fault. The total maximum available flow from the group is reduced by such unavailability, but another consideration is the requirement to achieve toroidally symmetric flow across all the valves in the group. The operator must first accept the constraint that if a sharing ratio is defined between multiple groups, the maximum flow rate for that virtual actuator will be limited by whichever group will saturate first in its weighted contribution to that role. Within a single group however, the operator can choose (with a selection flag in the GUI) what happens when the valve with the lowest maximum throughput in the group reaches saturation. If the primary objective is to maintain symmetric flow, then the whole group will saturate at this point and the maximum available flow of the group is defined as the maximum flow of this weakest valve multiplied by the number of available valves in the group. On the other hand, if greater flow capability is considered necessary at the expense of loss of symmetry, this option can be selected. The flow distribution will still divide the group flow request evenly across the available valves up to the point of one of them saturating, but for a higher group flow request it will redistribute the flow request to the more capable valves after one or more are saturated. The maximum flow capability of the gas group in this case is then the sum of the maximum flow of each of the available valves.

The above implementation provides the maximum capability to exploit all the capabilities of the MAST-U gas system within sensible operational constraints.

6. Conclusion

In this paper we have presented the architectural design for the MAST-U PCS, implemented on the General Atomics PCS framework. We dealt with the presence of many actuators for coil and gas control, and the need to change controller behaviour in multiple ways during a pulse. We managed these challenges by chaining together PCS categories and the use of “virtual actuators” to aggregate/arbitrate/route multiple upstream controller demands to finite downstream actuator commands. In doing so we achieved a powerful configuration capability while managing underlying complexity and ensuring consistency-checking against configuration errors.

We believe we have created a framework that can stay abreast of the evolution of the MAST-U machine as new capabilities are developed.

CRediT authorship contribution statement

Graham McArdle: Conceptualization, Methodology, Supervision, Writing - original draft, Visualization. **Luigi Pangione:** Software, Investigation, Formal analysis. **Martin Kochan:** Software, Validation, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has been(part-) funded by the RCUK Energy Programme [grant number EP/T012250/1]. To obtain further information on the data and models underlying this paper please contact PublicationsManager@ukaea.uk.

References

[1] W. Morris, et al., MAST Upgrade divertor facility: a test bed for novel divertor solutions, IEEE Trans. Plasma Sci. 46 (5) (2018) 1217–1226.
[2] B.G. Penaflor, et al., A structured architecture for advanced plasma control experiments, Lisbon, Portugal, Proc. of the 19th Symposium on Fusion Technology, vol. 1, 1996, p. 965.
[3] G. McArdle, J. Storrs, First results from the MAST digital plasma control system, Fusion Eng. Des. 71 (2004) 59–64.
[4] B.G. Penaflor, et al., Extending the capabilities of the DIII-D Plasma Control System for worldwide fusion research collaborations, Fusion Eng. Des. 84 (2009) 1484–1487.
[5] M.D. Boyer, et al., Plasma boundary shape control and real-time equilibrium reconstruction on NSTX-U, Nucl. Eng. 58 (2018) 036016.
[6] B.J. Xiao, D.A. Humphreys, M.L. Walker, A. Hyatt, J.A. Leuer, D. Mueller, et al., EAST plasma control system, Fusion Eng. Des. 83 (2008) 181–187.
[7] Sang-Hee Hahn, et al., Achievements and lessons learned from the operation of KSTAR plasma control system upgrade, Fusion Eng. Des. 130 (2018) 16–20.
[8] D.P. O'Brien, et al., Local expansion method for fast plasma boundary identification in JET, Nucl. Fusion 33 (3) (1993) 467–474.
[9] Yong Guo, et al., A local expansion method applied to fast plasma boundary reconstruction for EAST, Plasma Phys. Control. Fusion 53 (2011) 105015.
[10] F. Hoffman, S.C. Jardin, Plasma shape and position control in highly elongated tokamaks, Nucl. Fusion 30 (10) (1990) 2013–2022.