

Real-time physics-model-based simulation of the current density profile in tokamak plasmas

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2011 Nucl. Fusion 51 083052

(<http://iopscience.iop.org/0029-5515/51/8/083052>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 129.8.242.67

This content was downloaded on 10/06/2014 at 13:36

Please note that [terms and conditions apply](#).

Real-time physics-model-based simulation of the current density profile in tokamak plasmas

F. Felici, O. Sauter, S. Coda, B.P. Duval, T.P. Goodman,
J.-M. Moret, J.I. Paley and the TCV Team

École Polytechnique Fédérale de Lausanne (EPFL), Centre de Recherches en Physique des Plasmas, Association EURATOM-Suisse, 1015 Lausanne, Switzerland

E-mail: federico.felici@epfl.ch

Received 26 April 2011, accepted for publication 14 July 2011

Published 8 August 2011

Online at stacks.iop.org/NF/51/083052

Abstract

A new paradigm is presented to reconstruct the plasma current density profile in a tokamak in real-time. The traditional method of basing the reconstruction on real-time diagnostics combined with a real-time Grad–Shafranov solver suffers from the difficulty of obtaining reliable internal current profile measurements with sufficient spatial and temporal accuracy to have a complete picture of the profile evolution at all times. A new methodology is proposed in which the plasma current density profile is simulated in real-time by solving the first-principle physics-based equations determining its evolution. Effectively, an interpretative transport simulation similar to those run today in post-plasma shot analysis is performed in real-time. This provides real-time reconstructions of the current density profile with spatial and temporal resolution constrained only by the capabilities of the computational platform used and not by the available diagnostics or the choice of basis functions. The diagnostic measurements available in real-time are used to constrain and improve the accuracy of the simulated profiles. Estimates of other plasma quantities, related to the current density profile, become available in real-time as well. The implementation of the proposed paradigm in the TCV tokamak is discussed, and its successful use in plasma experiments is demonstrated. This framework opens up the possibility of unifying q profile reconstructions across different tokamaks using a common physics model and will support a wealth of applications in which improved real-time knowledge of the plasma state is used for feedback control, disruption avoidance, scenario monitoring and external disturbance estimation.

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Real-time knowledge of the internal state of a tokamak plasma, characterized by the current density profile as well as the kinetic (T_e , n_e , T_i) profiles and rotation profile, is of crucial importance for advanced operation of future devices including ITER. Until now, real-time knowledge of the internal plasma state has been based mainly on the particular set of temperature, density, rotation and magnetic pitch angle diagnostics available to each individual machine, sometimes in combination with constrained real-time Grad–Shafranov equilibrium reconstructions. Some recent work has employed statistical methods such as Bayesian analysis to provide rigorous techniques to merge separate diagnostic measurements into a single estimate for a given quantity [1]. Nevertheless, estimated quantities at a given time instant depend exclusively on diagnostic data gathered at that time point, possibly filtered to remove measurement noise.

In this paper, we present a novel paradigm for real-time reconstruction of the plasma state based on first-principle physics models. A physics-based model of internal plasma profile dynamics is solved while the plasma is physically evolving in the tokamak. This real-time simulation provides, within the validity boundary of the model, full knowledge of the plasma profiles with spatial and temporal resolution constrained only by the computational power of the hardware on which the simulation is executed, and not by the spatial or temporal resolution of diagnostics. In this framework, the available real-time diagnostic data can be included in a natural way to improve the accuracy of the estimate. This approach is known in the control engineering community as that of a *dynamic state observer* and is typically used to estimate unmeasured or poorly measured states of a dynamical system.

While the paradigm is general and can be applied, in principle, to any or all of the mentioned plasma profiles, this paper focuses on the current density profile (or, equivalently,

the safety factor (q) or poloidal flux (ψ) profile). We choose this profile for several reasons. First of all, the q profile is of prime importance for the plasma performance and macro-stability, since its shape determines the proximity to operational MHD limits, the presence of improved confinement (e.g. hybrid) regimes or transport barriers, and the appearance and location of tearing modes. Secondly, the physics of poloidal flux diffusion, which governs the current density profile evolution, is relatively well understood and reliable physics models exist on which to base first-principle simulations. Simulation results, therefore, can be trusted. This is in contrast to the more complex phenomena governing anomalous plasma energy transport, for example, for which first-principle models are much less developed and understood. Thirdly, the q profile happens to be a quantity which is particularly difficult to measure with sufficient spatial and temporal accuracy in real-time, even after years of development of diagnostics dedicated to do so, such as motional Stark effect (MSE) spectroscopy or polarimetry.

We propose to perform an interpretative transport simulation in real-time during the plasma discharge, of the same nature as is performed today post-shot by specialized transport codes (e.g. ASTRA [2], CRONOS [3], TRANSP [4, 5], JETTO [6], CORSICA [7]). A considerable amount of additional information obtained from these simulations, including the profiles of poloidal flux, current density, magnetic energy, rotational transform, magnetic shear, internal inductance, poloidal magnetic field, toroidal electric field, bootstrap current, ohmic current density, etc, becomes available in real-time and can be used for feedback control, disruption avoidance, scenario monitoring, and a wealth of other applications which will be discussed in some detail in section 2.

The basic idea underlying this paper was suggested earlier in work by Witrant *et al* [8], which presents a control-oriented current density profile model which would be suitable for real-time implementation. However, to our knowledge, such a model had until now never actually been implemented in a tokamak real-time system. Here, we present not only the methodology but also the implementation of this paradigm in the control system of the TCV tokamak [9] and its successful demonstration during plasma discharges. We show the successful resolution of the plasma poloidal flux transport equation with time steps of the order of 1 ms which is comfortably shorter than the global TCV current redistribution time (~ 150 ms for heated plasmas). The fact that this could be achieved for the relatively short characteristic time scales of the moderately sized TCV implies that this method can easily be applied to larger tokamaks with longer time scales.

The proof of principle and the main results obtained in TCV experiments are postponed until section 5. The first part of this paper presents the ingredients necessary to perform real-time current density profile simulations. Some are of a general nature and are valid independently of the particular tokamak on which the simulation is implemented. Others are more specific to the available diagnostics and implementation environment and will be different for each machine. The required ingredients are listed below with reference to the

section of this paper in which each is discussed in more detail.

- A first-principles-based *plasma physics model* that correctly describes the poloidal flux profile dynamics. In order to be applicable to advanced scenarios, this model must include at least plasma resistivity, bootstrap current and auxiliary current drive sources. The model we have implemented is based on the partial differential equation (PDE) describing the diffusion of poloidal flux as detailed in [10] and implemented in the ASTRA code [2], with the restriction that the shape of the flux surfaces and the toroidal flux they enclose maintained fixed in time. This implies that, at this stage, we do not solve the Grad-Shafranov equation in real-time, though we allow the value of the poloidal flux at each surface to vary in time. We use equations based on neoclassical calculations for the bootstrap current and neoclassical conductivity. Non-inductive current drive sources are modelled using *a priori* knowledge of the spatial distribution of the driven current, multiplied by the current drive efficiency and actuator power level. In section 3 we describe the model and the equations in some detail.
- A *numerical code* solving the poloidal flux diffusion PDE that is compatible with real-time implementation. For this purpose we have developed a new 1D transport code RAPTOR (RAPid Plasma Transport SimulatOR). RAPTOR uses finite elements to spatially discretize the PDE, yielding a high-order ODE for the finite-element coefficients. To achieve numerical stability, we use a mixed implicit-explicit temporal discretization scheme. With this choice, solving one time step involves constructing and solving a band-structured linear matrix equation as is briefly discussed in section 3.2. The numerical implementation and the necessary measures to make the solution as fast as possible are discussed in some detail in appendix A. Although material described in this section is essential for the successful real-time implementation of the algorithm, this rather technical section will appeal mostly to those interested in the details of the involved numerics.
- *Real-time measurements* of the plasma quantities that are required as inputs to the poloidal flux transport solver. These are (a) the total plasma current I_p or loop voltage V_{loop} and (b) the plasma kinetic profiles, at least $T_e(\rho)$ but preferably also $n_e(\rho)$ and $T_i(\rho)$ for bootstrap current calculations. The plasma current is available in real-time in practically all present-day tokamaks from a discrete set of magnetic measurements or from a Rogowski coil. The kinetic profiles are inherently more complicated to obtain and the availability of real-time diagnostics for each profile is different for each individual tokamak. In some cases, ECE, Thomson scattering, and/or charge exchange recombination spectroscopy diagnostics may be used in real-time. Since this is not the case for TCV, an alternative solution had to be found which uses other diagnostics that are available in real-time. In section 4.1 we present the particular solution that has been implemented: a set of line-integrated soft x-ray measurements and interferometer signals is combined and processed to yield real-time estimates of the profiles of n_e and T_e . This has proven to be one of the more difficult steps in the implementation

but crucial since the current profile dynamics is strongly influenced by the T_e profile evolution in particular.

- A *real-time computational platform* on which the transport code is implemented. Clearly, the real-time estimates of plasma quantities discussed above should be available at the start of each time step of the code execution. Furthermore, the outputs of the real-time simulation should be passed to other real-time processing entities for real-time analysis and control applications. The hardware and software specific details differ for each tokamak, but most advanced tokamak control systems today provide these capabilities to some degree. On TCV, the new distributed digital control system [11] provides the required platform on which the simulations are run using a CPU of relatively modest computational power. Information on the hardware and software used, and on how the real-time flux profile simulations are implemented in this system, is given in section 4.2.
- A *methodology to include additional measurements*, beyond the minimally required set, which may remove some uncertainty in the modeling and help to further constrain the solution. A first example has been implemented in the context of the TCV application, in which the loop voltage is used as redundant measurement to adapt the value of the effective charge Z_{eff} , a measure of the effective impurity content. This is presented in section 6.2.

As mentioned, we have successfully implemented and tested the real-time current density simulations on the TCV tokamak. The results of these simulations are discussed in section 5 where we compare the real-time computed profile information with off-line reconstructions obtained from the TCV Grad-Shafranov equilibrium reconstruction code LIUQE and from off-line interpretative transport modeling using ASTRA. The satisfactory comparison proves the validity of our implementation and opens the road towards use of the real-time reconstructed profiles for the multitude of applications described in section 2. We conclude this paper by discussing, in section 6, different methods to incorporate diagnostic measurements in a real-time simulation scheme as a means to handle the inevitable mismatch between the model used and the physical reality.

2. Applications of real-time tokamak plasma simulations

In this section, we will illustrate some of the major possible roles of a real-time simulation of the tokamak plasma evolution in an advanced tokamak control scheme. They are shown schematically in figure 1 and are discussed more in-depth below.

Physics-based estimate of the plasma state. The fundamental advantage of using a real-time simulation to obtain the plasma state is that the dynamic behaviour of the plasma, as dictated by the physics, is explicitly taken into account. Additional real-time diagnostics can complement and improve the quality of this estimate, as will be discussed in greater detail in section 6. The real-time simulation, by itself, can yield information

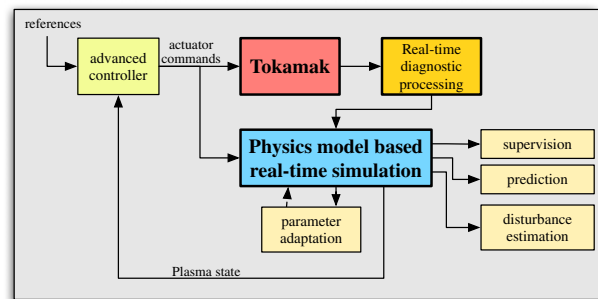


Figure 1. Diagram illustrating the envisaged role of a real-time simulation in a tokamak real-time control scheme. The real-time simulator simulates the plasma behaviour based on the same inputs as the actual tokamak. Measurements from the available real-time diagnostics are, after pre-treatment, used to help the simulation converge to an accurate plasma state. Mismatches between measured and expected values can be used either to estimate disturbances, or to adapt the model parameters in real-time. Application of the plasma state knowledge to scenario monitoring, prediction and feedback control is also drawn.

on an arbitrary spatial and temporal scale beyond the sample time and spatial resolution of diagnostic hardware. Using this state as a basis for real-time control (e.g. using the methods presented in [12]), rather than relying on individual measurements from specific diagnostics, carries great advantages. The state estimate is decoupled from the specificities of the diagnostics, which are different for each machine, increasing the portability and comparability of real-time algorithms across different machines. Furthermore, measurement noise present in the diagnostics is filtered out by the real-time simulation if this noise is not compatible with physically expected variations in the measured signals. Finally, quantities which cannot be directly measured can also be used for feedback control, for example the bootstrap current fraction can be used as a real-time controlled variable in advanced scenarios.

Physics parameter estimation and adaptation. In addition to estimating the plasma state as discussed above, other quantities can be estimated as well. Firstly, parameters in the physics model which are not precisely known may be estimated by incorporating redundant measurements (see section 6). Secondly, further physics calculations can be performed which do not pertain to calculating the state but give auxiliary quantities of interest. One example is to evolve a sawtooth crash criterion model as presented in [13, 14] in real-time since all quantities which go into the model are known from the plasma state. This would give real-time estimates of the proximity to the next sawtooth crash. Another example is to simulate the evolution of neoclassical tearing modes (NTMs) by evolving the modified Rutherford equation [20] in real-time. This would allow one to predict, for example, the amount of ECCD required to stabilize an already evolving NTM.

Supervision. This broad category includes all high-level supervision activities performed by the control system to maximize the scientific output of the tokamak while maintaining safe operation. In particular, the plasma state can be compared with expected trajectories to check

whether the plasma is still behaving in a well understood and expected way. Since the design of any control system is based on expectations of the plasma response to actuator inputs, an indication that the system is not behaving as expected from models would be cause for concern and possibly a reason to terminate the plasma in a safe manner. Apart from the safety aspect, supervision can have the goal of rescheduling the evolution of the single tokamak experiment during the shot as needed.

Prediction. As a final application, we mention the use of real-time predictive simulations, i.e. faster-than-real-time simulations. This provides estimates of the *future* plasma state based on knowledge of future actuator inputs. This has important applications in predicting whether actuator saturations or safety limits are about to be exceeded. The plasma state can also be monitored to provide advance warning when entering a regime with increased risk of disruptions: as recent focus has shifted from disruption mitigation to disruption prevention and avoidance, a real-time simulation, in particular of the q profile, may enable a much more precise prediction of the proximity to the stability limit and would allow the system to effect an appropriate mitigation or soft-stop response depending on the plasma state. This will be particularly crucial for large experiments like ITER where disruptions can have serious consequences. Taking this approach one step further, one can also optimize future actuator inputs by iteratively predicting the plasma state in a model-predictive control scheme as proposed by [16, 17].

With these applications in mind, we now move to describe the physics model for poloidal flux diffusion which has been implemented in this first version of a real-time physics-based simulation.

3. Simplified physics model for poloidal flux diffusion

In this section, we describe the simplified physics model that is to be solved in real-time. This is the principal ingredient of the real-time simulation framework and the choice of which physics to include or exclude certain physics strongly impacts the final current density profile estimate. We begin this section by recalling the 1D poloidal flux transport equation which is to be solved in real-time, and by showing the main simplifying assumptions which were made to lighten the computational complexity of the problem. Although some of the material presented here is based on [2, 10] we have included this material for completeness.

At the end of this section, we will also summarize the choices made in the numerical implementation of the physics model, while referring to the appendix for most details. We also mention the role of the initial conditions of the simulation and the time scales on which it will affect the solution.

While this section has some similarities with the work presented in [8], the model formulation and solution method is somewhat different. The key difference lies in the spatial discretization scheme, as we do not use finite differences but a finite-element approach, which allows us to formulate a variational form of the poloidal flux diffusion equation. Also, we have specifically focused on modeling electron cyclotron

current drive (ECCD) and have not modelled other forms of auxiliary current.

3.1. Poloidal flux diffusion equation

The derivations and definitions below, as well as the notation, follow the ASTRA documentation [2]. We start, as is customary, by defining a set of coordinates $\{\rho, \theta, \zeta\}$ where ρ is an arbitrary flux surface label, θ is the poloidal angle and ζ is the toroidal angle in the direction required for a right-handed system. The following standard relations can be derived, relating the main magnetic quantities to the two scalar functions $\psi(\rho)$ and $F(\rho)$:

$$\mathbf{B} = F \nabla \zeta + \frac{1}{2\pi} (\nabla \zeta \times \nabla \psi) \quad \text{Magnetic field} \quad (1)$$

$$\mathbf{j} = \frac{\nabla \zeta}{2\pi \mu_0} R^2 \nabla \cdot \left(\frac{\nabla \psi}{R^2} \right) + \frac{1}{\mu_0} (\nabla F \times \nabla \zeta) \quad \text{Current density} \quad (2)$$

$$\Phi = \frac{1}{2\pi} \int_V \mathbf{B} \cdot \nabla \zeta \, dV \quad \text{Toroidal flux} \quad (3)$$

$$F = R_0 B_0 - \frac{\mu_0}{2\pi} \int_{S_\theta} \mathbf{j} \cdot d\mathbf{S}_\theta \quad \text{Poloidal current function.} \quad (4)$$

Here R_0 is the major radius of the torus, B_0 is the vacuum toroidal magnetic field at R_0 and R is the distance from the vertical machine axis. S_θ is a toroidal ribbon orthogonal to $\nabla \theta$, extending from the plasma magnetic axis to a flux surface.

Choosing the flux surface label as the square root of the (unnormalized) toroidal flux:

$$\rho = \rho_{\text{tor}} = \sqrt{\frac{\Phi(\rho)}{\pi B_0}}. \quad (5)$$

we can derive the poloidal flux diffusion equation, describing the temporal diffusion of the poloidal flux $\psi(\rho, t)$ due to the plasma resistivity. Let us define flux surface averaged quantities as $\langle Q \rangle = \frac{\partial}{\partial V} \int Q dV$ and assume the flux surface averaged longitudinal Ohm's law

$$\sigma_{\parallel} E_{\parallel} = j_{\parallel} - j_{\text{BS}} - j_{\text{CD}}, \quad (6)$$

where E_{\parallel} , j_{\parallel} , j_{BS} and j_{CD} are defined following $j_{\parallel} = \langle \mathbf{j} \cdot \mathbf{B} \rangle / B_0$, and j_{BS} and j_{CD} are the bootstrap and auxiliary current density profiles, respectively. Assuming that $\partial B_0 / \partial t = 0$, the poloidal flux diffusion equation can be written as [2]

$$\sigma_{\parallel} \frac{\partial \psi}{\partial t} = \frac{R_0 J^2}{\mu_0 \rho} \frac{\partial}{\partial \rho} \left(\frac{G_2}{J} \frac{\partial \psi}{\partial \rho} \right) - \frac{V'}{2\pi \rho} (j_{\text{BS}} + j_{\text{CD}}), \quad (7)$$

where

$$J = \frac{F}{R_0 B_0}, \quad (8)$$

$$V' = \frac{\partial V}{\partial \rho}, \quad (9)$$

$$G_2 = \frac{1}{4\pi^2} V' \langle (\nabla \rho)^2 / R^2 \rangle. \quad (10)$$

Since ρ is the independent variable in this equation, it describes the diffusion of poloidal flux through a given background of toroidal flux. The spatial distribution of toroidal flux in the 2D

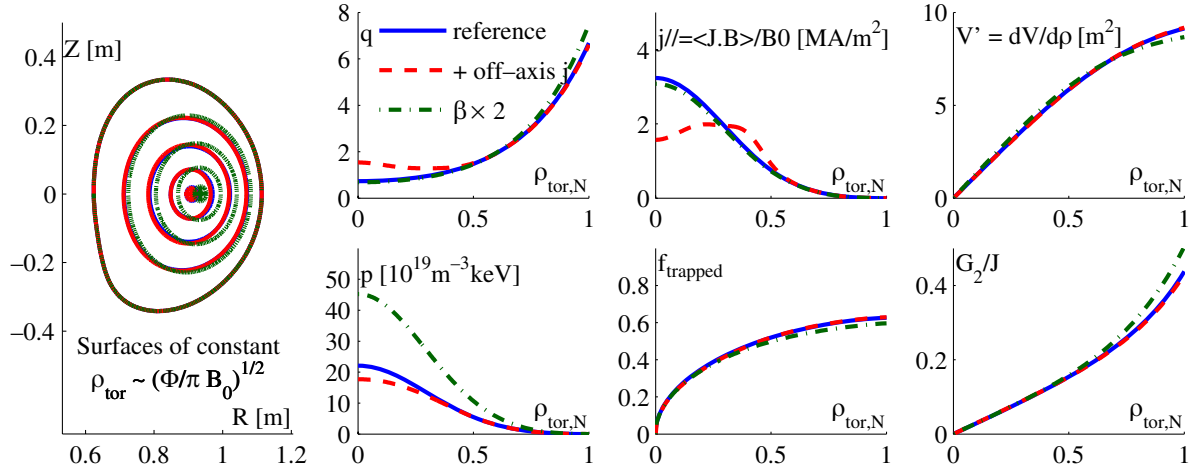


Figure 2. Set of TCV equilibria illustrating the effect of assuming fixed toroidal flux surfaces on the geometric terms V' , G_2/J appearing in (7). A reference equilibrium (blue, —) is perturbed by redistributing current density to give a reversed-shear q profile (red, ---), and by doubling the pressure, i.e. β (green, -·-). The variation in V' and G_2/J can be seen to be within 10% of their reference value. Also the trapped particle fraction, which governs the neoclassical contribution to j_{BS} , $\sigma_{||}$ and current drive efficiency, is hardly changed. The largest difference appears when changing the pressure through a change in Shafranov shift. The calculations were performed using the fixed-boundary equilibrium code CHEASE [18].

poloidal plane satisfies the Grad–Shafranov equilibrium, the source term of which is the toroidal current density profile j_{tor} (the first term on the rhs of (2)) which changes in time according to the radial poloidal flux redistribution governed by (7). Since, on the other hand, changes in the equilibrium cause variations in $\Phi \sim \rho^2$, J , V' and G_2 , the full magnetic configuration of the tokamak plasma is governed by (7), nonlinearly coupled to the Grad–Shafranov equation.

3.1.1. Solution method in existing transport codes. While an in-depth discussion of the working of modern transport codes is well beyond the scope of this article, it is useful to sketch a simple outline. The procedure for solving the flux diffusion equation in interpretative simulations (i.e. with prescribed temporal evolution of T_e , T_i and n_e) generally involves one or more iterations of the following steps:

- (i) The Grad–Shafranov equation, or an approximation thereof, is solved using the toroidal current density $j_{tor}(\rho)$ giving the spatial functions $\psi(R, Z)$, $\Phi(R, Z)$, hence $\rho(R, Z)$ on the poloidal plane. From this, one computes functions that are constant on a flux surface, such as ρ , $J(\rho)$ and geometric quantities $G_2(\rho)$, $V'(\rho)$. This step may be performed by a free or fixed-boundary calculation depending on whether the plasma boundary is specified or is to be determined based on the currents in the poloidal field (PF) coil set.
- (ii) The source terms $j_{CD}(\rho)$ and $j_{BS}(\rho)$, as well as the conductivity $\sigma_{||}(\rho)$, are computed based on the prescribed density and temperature profiles and models of auxiliary current drive injection systems. These calculations can be of varying complexity and may include geometric, kinetic and neoclassical effects.
- (iii) The poloidal flux diffusion equation (7) is solved for the next time step, yielding the flux profile $\psi(\rho, t)$ and a new $j_{tor}(\rho)$ to be used as input for the Grad–Shafranov step (i). The edge boundary condition for ψ at $\rho = \rho_{edge}$ is provided by imposing the total plasma current or loop voltage.

3.1.2. Fixed equilibrium assumption. As mentioned, in most existing codes the flux surface shapes are evolved by a Grad–Shafranov solver. The link to the Grad–Shafranov solution has not yet been included in the real-time model at this stage, although it could be an important extension. Instead, the shape of the poloidal flux surfaces as well as the toroidal flux Φ they enclose, are chosen once for one reference equilibrium and remain fixed in space and time. This restriction means that we are in practice (i) neglecting variations in diamagnetic effects which cause variations of $J(\rho)$, (ii) neglecting effects due to changes in the shapes flux surface shapes, primarily as a result of varying Shafranov shift as β evolves and (iii) assuming a fixed plasma boundary in time.

Note that this is a weaker statement than assuming the Grad–Shafranov equilibrium to be fixed, indeed we do allow the poloidal flux profile $\psi(\rho)$, and hence q , j_{tor} , etc to change in time. We merely fix the flux surface geometry and enclosed toroidal flux. By choosing a reference equilibrium we can treat arbitrary plasma shapes as long as they are not time-varying. Analysis of several plasma equilibria, shown in figure 2, indicates that unless the plasma β varies significantly with respect to the reference, the change in poloidal current density and Shafranov shift will be limited. Even a doubling of β results in a variation of G_2/J and V' of less than 10%. The consequence of this assumption is that in equation (7) $G_2(\rho, t) = G_2(\rho)$, $V'(\rho, t) = V'(\rho)$ and $J(\rho, t) = J(\rho)$ (equations (8)–(10)) are now fixed functions of ρ in time, evaluated once only based on a chosen Grad–Shafranov equilibrium. The only time-varying quantities apart from the dependent variable $\psi(\rho, t)$ are $\sigma_{||}(\rho, t)$, $j_{CD}(\rho, t)$ and $j_{BS}(\rho, t)$.

3.1.3. Current sources, conductivity and flux boundary conditions. The bootstrap current term j_{BS} as well as the parallel conductivity $\sigma_{||}$ are calculated using the expressions from [19, 20] which represent numerical fits of accurate neoclassical calculations for arbitrary shape, aspect ratio and collisionality.

The conductivity term is split into the Spitzer component multiplied by a neoclassical correction describing trapped particle effects:

$$\sigma_{\parallel} = c_{\text{neo}} \sigma_{\text{Spitz}}. \quad (11)$$

This correction term c_{neo} depends on the trapped particle fraction and collisionality, which in turn depend on the equilibrium and n_e , T_e profiles, respectively. The Spitzer conductivity is given by $\sigma_{\text{Spitz}} = 1.9012 \times 10^4 T_e [\text{eV}]^{3/2} / (Z N(Z) \ln \Lambda_e)$ where $Z = Z_{\text{eff}} = \sum_{p \in \text{ions}} Z_p^2 n_p / n_e$ is the effective charge and $N(Z) = 0.58 + 0.74 / (0.76 + Z)$ depends weakly on Z [19, 20].

For the bootstrap current we have, again from [19, 20] (note the factor of 2π due to a different definition of ψ)

$$j_{\text{BS}} = -2\pi J(\psi) R_0 p(\psi) \left[\mathcal{L}_{31} \frac{\partial \ln n_e}{\partial \psi} + R_{\text{pe}} (\mathcal{L}_{31} + \mathcal{L}_{32}) \frac{\partial \ln T_e}{\partial \psi} + (1 - R_{\text{pe}}) (\mathcal{L}_{31} + \alpha \mathcal{L}_{34}) \frac{\partial \ln T_i}{\partial \psi} \right]. \quad (12)$$

Here $R_{\text{pe}} = p_e / p$ is the ratio between electron and total pressure. Assuming $n_i = n_e$ and $\frac{\partial \ln T_i}{\partial \psi} = \frac{\partial \ln T_e}{\partial \psi}$ we rewrite this in the more convenient form involving derivatives of ρ :

$$j_{\text{BS}} = -\frac{2\pi J(\psi) R_0}{R_{\text{pe}}} \frac{\partial \rho}{\partial \psi} \left[\mathcal{L}_{31} \frac{\partial n_e}{\partial \rho} T_e + (\mathcal{L}_{31} + R_{\text{pe}} \mathcal{L}_{32} + (1 - R_{\text{pe}}) \alpha \mathcal{L}_{34}) \frac{\partial T_e}{\partial \rho} n_e \right]. \quad (13)$$

The profiles J , \mathcal{L}_{31} , \mathcal{L}_{32} , \mathcal{L}_{34} , α also depend on equilibrium and collisionality. Note also that since the T_e and n_e profiles derivatives with respect to ρ are used, the above expression involves the reciprocal of the flux profile gradient $(\partial \psi / \partial \rho)^{-1}$.

In general, the auxiliary current drive can be written as a sum of contributing sources $\sum_i j_i(\rho, t)$ where $i \in \{\text{ECCD}, \text{ICCD}, \text{LHCD}, \text{NBCD}\}$. The only auxiliary current drive system on TCV is the electron cyclotron (ECCD) system. Its driven current density is modelled by an empirical formula

$$j_{\text{aux}}(\rho, t) = c_{\text{cd}} e^{-\rho^2 / 0.5^2} \frac{T_e}{n_e} e^{-(\rho - \rho_{\text{dep}})^2 / w_{\text{cd}}^2} P_{\text{EC}}(t). \quad (14)$$

Here we have modelled the deposition as a Gaussian with a given width, including effects of T_e / n_e on the current drive efficiency. An ad hoc term accounting for the inefficiency due to trapped particles towards the edge has been included. The efficiency is multiplied by the varying gyrotron power (P_{EC}). In all, this relation mimicks the more rigorous approach presented in [21]. A machine-dependent final scaling factor c_{cd} is determined from experimental experience to normalize to the experimentally measured values of total current drive. This factor is typically of order $10^{15} \text{ A} / (\text{eV m}^5 \text{ W})$ for the TCV case. Multiple ECCD contributions from multiple launchers can be included by summing various terms of this form. This way of modeling removes the need for real-time ray-tracing and wave-particle interaction calculations: these are summarized by analytic expression. Accurate, pre-calculated current drive profiles for a given set of equilibria and injection geometries could be pre-calculated using ray-tracing codes, combined in a look-up table and interpolated in real-time.

The boundary terms can be specified either via the plasma current

$$\frac{\partial \psi}{\partial \rho} \Big|_{\rho=\rho_e} = \frac{\mu_0}{G_2(\rho_e)} I_p(t) \quad (15)$$

or using the ohmic transformer flux $\Psi_{\text{OH}} = \int_t V_{\text{OH}} dt$:

$$\Psi_{\text{OH}}(t) = L_{\text{ext}} \frac{G_2}{\mu_0} \frac{\partial \psi}{\partial \rho} \Big|_{\rho=\rho_e} + \psi(t) \Big|_{\rho=\rho_e}, \quad (16)$$

where L_{ext} represents the mutual inductance between the ohmic primary and the plasma and V_{OH} is the inductive voltage from the ohmic transformer. Since I_p is available in real-time from magnetic measurements, the first option will generally be used for interpretative simulations.

3.1.4. Output quantities. From knowledge of the flux profile $\psi(\rho, t)$ and its spatial and temporal derivatives, profile quantities related to the magnetic configuration of the plasma within the last closed flux surface (LCFS) can be calculated.

Plasma current inside ρ surface (A)

$$I_{\text{pl}} = \frac{G_2}{\mu_0} \frac{\partial \psi}{\partial \rho}. \quad (17)$$

Toroidal current density (A m^{-2})

$$j_{\text{tor}} = \frac{2\pi R_0}{\mu_0 V'} \frac{\partial}{\partial \rho} \left(G_2 \frac{\partial \psi}{\partial \rho} \right). \quad (18)$$

Rotational transform, 1/safety factor

$$\iota = \frac{1}{q} = \frac{\partial \psi}{\partial \Phi} = \frac{1}{2\pi B_0 \rho} \frac{\partial \psi}{\partial \rho}. \quad (19)$$

Magnetic shear

$$s = \frac{2V}{q} \frac{\partial q}{\partial V} = \frac{2V}{\rho V'} \left(1 - \rho \frac{\partial^2 \psi}{\partial \rho^2} \left(\frac{\partial \psi}{\partial \rho} \right)^{-1} \right). \quad (20)$$

Poloidal field energy density (J m^{-3})

$$w_i = \frac{G_2}{2\mu_0 V'} \left(\frac{\partial \psi}{\partial \rho} \right)^2. \quad (21)$$

Magnetic energy inside ρ surface (J)

$$W_i = \frac{1}{2\mu_0} \int_0^\rho \left(\frac{\partial \psi}{\partial \rho} \right)^2 G_2 d\rho. \quad (22)$$

Unnormalized internal inductance (H)

$$L_i = 2W_i / I_{\text{pl}}^2. \quad (23)$$

Toroidal plasma loop voltage (V)

$$U_{\text{pl}} = \frac{\partial \psi}{\partial t}. \quad (24)$$

Parallel electric field (V m^{-1})

$$E_{\parallel} = \frac{2\pi \rho}{V'} U_{\text{pl}}. \quad (25)$$

Ohmic power density (W m^{-3})

$$P_{\text{OH}} = \frac{1}{2\pi R_0} U_{\text{pl}} j_{\text{tor}}. \quad (26)$$

Based on this list of profiles, a number of related quantities may be computed such as the bootstrap current fraction I_{BS} / I_p and externally driven current fraction I_{CD} / I_p . Using the ohmic power density P_{OH} (26), and since both electron and ion temperature, density and injected auxiliary power density profiles are assumed to be given, a real-time power

balance can be established using the electron energy transport equation [2, 10]:

$$V' \frac{\partial}{\partial t} [n_e T_e] = \frac{\partial}{\partial \rho} V' \left(\langle (\nabla \rho)^2 \rangle n_e \chi_e \frac{\partial T_e}{\partial \rho} \right) + V' P_e, \quad (27)$$

where χ_e is the electron heat diffusion profile and $P_e = P_{OH} + P_{aux} - P_{ei} - P_{rad}$. The electron-ion equipartition power P_{ei} is given by [10]

$$P_{ei} = n_e v_{eq} (T_e - T_i), \quad (28)$$

where $v_{eq} = 0.041 T_e^{-3/2} \sum_{p \in \text{ions}} n_{p[19]} Z_p^2 / A_p$ [10] is the neoclassical equipartition rate and Z_p and A_p are the charge and atomic mass number of each ion species, respectively. Using $n_i = (Z_C - Z_{eff}) / (Z_C - 1) n_e$ and the definition of Z_{eff} , in the case of deuterium plasmas with mainly carbon impurity we can rewrite v_{eq} as

$$v_{eq} = 0.041 T_e^{-3/2} \left(\frac{n_i}{A_i} + \frac{Z_C (n_e - n_i)}{A_C} \right). \quad (29)$$

The bremsstrahlung radiation losses are calculated using the simple formula $P_{rad} = 5.35 \times 10^{-5} Z_{eff} n_e^2 T_e^{1/2}$ [22]. We neglect losses due to cyclotron radiation, as they are usually negligible in the plasmas we have considered. Furthermore, as we do not model the ion temperature (which plays a minor role for the TCV plasmas studied), loss channels for the ion energy need not be considered.

Calculating the electron thermal energy $W_e = \int_\rho k_B n_e T_e V' d\rho$ and its time derivative, the instantaneous electron thermal diffusivity profile is given by

$$\chi_e(\rho, t) = \frac{\int_\rho \left(\frac{\partial W_e}{\partial t} - P_e \right) V' d\rho}{(V' \langle (\nabla \rho)^2 \rangle k_B n_e \frac{\partial T_e}{\partial \rho})}, \quad (30)$$

which can be calculated at each time step. The global energy confinement time is given by

$$\tau_E = \frac{R_{pe} W_e}{\int_\rho (P_{oh} + P_{aux} - P_{rad}) V' d\rho - R_{pe} dW_e/dt}, \quad (31)$$

from which confinement enhancement factors can be calculated for different scaling laws.

We can also compute time derivatives of profile quantities. For example

$$\frac{\partial I_{pl}(\rho)}{\partial t} = \frac{G_2}{\mu_0} \frac{\partial U_{pl}}{\partial \rho} \quad (32)$$

from which we can define a degree of ‘variability’ or, conversely ‘stationarity’ of this profile:

$$f_{ss, I_{pl}} = \left\| \frac{\partial I_{pl}(\rho)}{\partial t} \right\|_2 = \left[\int_0^{\rho_e} \left(\frac{G_2}{\mu_0} \right)^2 \left(\frac{\partial U_{pl}}{\partial \rho} \right)^2 d\rho \right]^{1/2}. \quad (33)$$

For other profile quantities, we can similarly define stationarity factors. They all turn out to be weighted norms of $\frac{\partial U_{pl}}{\partial \rho}$, and clearly a stationary state for all profiles is reached when the U_{pl} profile becomes flat, as expected¹ In this sense, these factors

¹ We use the term ‘stationary state’ to avoid confusion with the conventional terminology where ‘steady state’ refers to $U_{pl}(\rho) = 0$. In a stationary state the voltage profile is flat but not necessarily zero.

essentially represent a weighted least-squares distance from the stationary state (ss) solution. For example

$$f_{ss, t} = \left\| \frac{\partial t(\rho)}{\partial t} \right\|_2 = \left[\int_0^{\rho_e} \left(\frac{1}{2\pi B_0 \rho} \right)^2 \left(\frac{\partial U_{pl}}{\partial \rho} \right)^2 d\rho \right]^{1/2} \quad (34)$$

$$f_{ss, w_i} = \left\| \frac{\partial w_i(\rho)}{\partial t} \right\|_2 = \left[\int_0^{\rho_e} \frac{I_{pl}^2}{V'} \left(\frac{\partial U_{pl}}{\partial \rho} \right)^2 d\rho \right]^{1/2}. \quad (35)$$

These scalar values provide a practical measure of the degree to which the profiles have relaxed to their stationary conditions. In section 5 we provide an example of the evolution of $f_{ss, I_{pl}}$.

3.2. Numerical implementation using finite elements

The numerical solution of equation (7) is implemented using a finite-element method for the spatial discretization and a semi-implicit backwards Euler method for the temporal discretization. The code has been recently developed and named RAPTOR (RAPid Plasma Transport SimulatOR). The methodology describing the use of finite elements is discussed in detail in appendix A but a brief outline is given here. The flux at a discrete time step k : $t = t_k$ is approximated as

$$\psi(\rho, t_k) \approx \sum_{\alpha=1}^{n_{sp}} \hat{\psi}_\alpha(t_k) \Lambda_\alpha(\rho). \quad (36)$$

Where $\Lambda_\alpha(\rho)$ are the finite-element basis functions, chosen as cubic B-splines. To evolve equation (7) in time one needs, at each time step, to construct and solve a linear matrix-vector equation of the form

$$A_k \hat{\psi}_{k+1} = B_k \hat{\psi}_k + s_k, \quad (37)$$

where the matrices A_k and B_k and vector s_k must be assembled at each time step and depend on the time-varying terms $\sigma_{||}(\rho, t_k)$, $j_{BS}(\rho, t_k)$, $j_{aux}(\rho, t_k)$ and $I_p(t_k)$. $\hat{\psi}_k = [\hat{\psi}_1(t_k), \dots, \hat{\psi}_{n_{sp}}(t_k)]^T$ is the vector collecting the finite-element coefficients.

Considerable care has been taken to minimize the amount of computation time necessary for each time step (i.e. the part that has to be executed in real-time). The approach using finite elements differs from most other 1D plasma transport codes which use finite differences. The use of finite elements and the resulting weak form allows one to reduce the order of the derivatives involved. Indeed, only first derivatives of the finite elements are involved in the construction of A_k and B_k , while the original PDE (7) involves second-order spatial derivatives.

3.3. Role of the initial condition

Solving the parabolic PDE (equation (7)) requires an initial condition $\psi(\rho, 0)$. Even though I_p is known at the start of the simulation, one still has to choose the spatial distribution of the current density. The choice of the initial condition will affect the evolution of the profile until its effect has diffused away on the current redistribution time scale.

This may or may not pose a problem depending on the particular plasma scenario to be studied. If the plasma current ramp-up is followed by a period of stationary ohmic plasma

in which the current redistribution time is short, the profiles will rapidly converge to a stationary state and errors in the initial condition will at the same time disappear. If, on the other hand, early heating is applied lengthening the current redistribution time, it will take a longer time for the effect of the initial condition to become negligible.

On TCV, this does not pose a problem since (i) the current redistribution time is relatively short (~ 150 ms for heated plasmas, much shorter for ohmic plasmas, compared with a typical shot duration of ~ 2 s), and (ii) in the majority of TCV shots, there is always an ohmic phase early in the shot just after the plasma current ramp-up, which allows the profiles to go to a stationary state even before heating is applied. For larger tokamaks, however, the problem may be more significant since the current redistribution time may be a significant fraction, or may even be even longer than the total shot time. Also, many hybrid and advanced scenarios rely explicitly on early heating. A further complication may arise in the case where the plasma shape is also evolved during the ramp-up, as is envisaged in ITER.

To alleviate this, one can either rely on internal current density profile measurements as discussed in section 6.1 or use more advanced models of the initial phases of plasma formation and ramp-up to estimate the initial current density profile for initial (low-current) plasma and/or include some effects of the free-boundary equilibrium to account for varying shaping.

4. Implementation of the real-time simulation on TCV

In the previous section, we have described the physics model which is solved in real-time to obtain the poloidal flux profile and have introduced the RT-RAPTOR transport code used to simulate the model. In this section, we will show how this simulator has been implemented in practice in the TCV digital control system for real-time operation.

We have seen that, in order to calculate the conductivity, bootstrap current and auxiliary current one needs to have estimates of the profiles of T_e , n_e and T_i in real-time. The details of how they are obtained will be different for each tokamak, depending on the available real-time diagnostics. We will begin this section by describing the particular method that was implemented on TCV for the available real-time diagnostic set.

4.1. Real-time estimates of the kinetic profiles

Ideally, a tokamak equipped with real-time T_e , n_e and T_i profile diagnostics could feed this information directly to the real-time flux diffusion simulator. However, the ECE, Thomson scattering and charge exchange recombination spectroscopy systems available on TCV cannot provide real-time data at this time. We must therefore infer the profiles from diagnostics available in real-time, in the TCV case a set of 14 far-infrared interferometer (FIR) chords [23], 64 soft-x ray chords from the DMPX diagnostic [24] and 4 channels from a single-chord multi-filter x-ray diagnostic (XTe). The method to do this has been named ‘TENEX’ (TE and NE from X-rays).

As TCV plasmas have only direct electron heating, the ion temperature T_i plays a relatively modest role in the sustainment of the bootstrap current and ions store a relatively small and constant fraction of the total plasma thermal energy. Lacking a real-time diagnostic to measure the T_i evolution, we fix this profile and take the value during the initial, ohmic phase of the discharge (before any auxiliary heating is applied), which yields, on average $T_i^{\text{ohmic}} \approx 0.7T_e^{\text{ohmic}}$ at the densities obtained in the discharges considered here.

The remaining problem is to estimate the T_e and n_e profiles. The central electron temperature T_{e0} is directly provided by the XTe diagnostic by taking the ratio between x-ray intensities measured with different thicknesses of beryllium filters. This diagnostic has been calibrated against Thomson scattering measurements in the past and is known to give a reasonable quality estimate. To estimate the T_e profile shape $\tilde{T}_e(\rho) \equiv T_e(\rho)/T_{e0}$, and the density profile $n_e(\rho)$, we rely on a combination DMPX and FIR chords. As the SXR intensity has the dependence $I_{\text{SXR}} \sim n_e^2 T_e^\alpha$ where $\alpha \sim 0.5$, the n_e and T_e profiles can in theory be derived from a nonlinear mapping from these two measurements. We have chosen to construct this mapping using machine-learning based techniques which are discussed in more detail in appendix B. In this approach, a set of shots with similar plasma shape is collected for which the n_e , T_e profiles are known from off-line post-shot Thomson scattering data. Using this shot database, a neural network is constructed which maps the data of the DMPX and FIR chord to profiles of $n_e(\rho)$ and $T_e(\rho)$. After training the network, it can be used in real-time to estimate these profiles from fresh diagnostic data.

Profiles estimated by TENEX in real-time for one example shot are shown in figure 3 and compared with Thomson scattering measurements for the same shot. We see that the T_e profile is relatively well reproduced by the neural network and the main difficulty resides in the estimation of $T_e(0)$ from the XTe diagnostic. On the other hand, both $n_e(0)$ and the n_e profile are well reproduced. We note here that this approach has been taken mainly because a direct T_e profile measurement is not available in real-time on TCV at this time. If such a measurement were available, the data could be used directly, avoiding the need for this scheme.

4.2. Implementation on the TCV digital control system

Having described the physics model RT-RAPTOR (Real-Time RAPTOR) code in section 3, and the TENEX real-time kinetic profile reconstruction in section 4.1, we turn in this section to their implementation in the TCV digital control system.

The new TCV digital control system [11] consists of multiple nodes, all of which have a Linux PC with a X86 CPU. Some nodes have analogue acquisition and output modules hosted in a compact-PCI crate. The present system contains four nodes, two of which are connected to a complementary set of diagnostics and which command a set of actuators. The other two nodes are for computation only and have no autonomous data acquisition or command capabilities. The nodes can pass data to each other in real-time utilizing a reflective (shared) memory. Data written into this memory by one node are automatically copied to the same memory space in all of the other nodes. Each node can run at a different

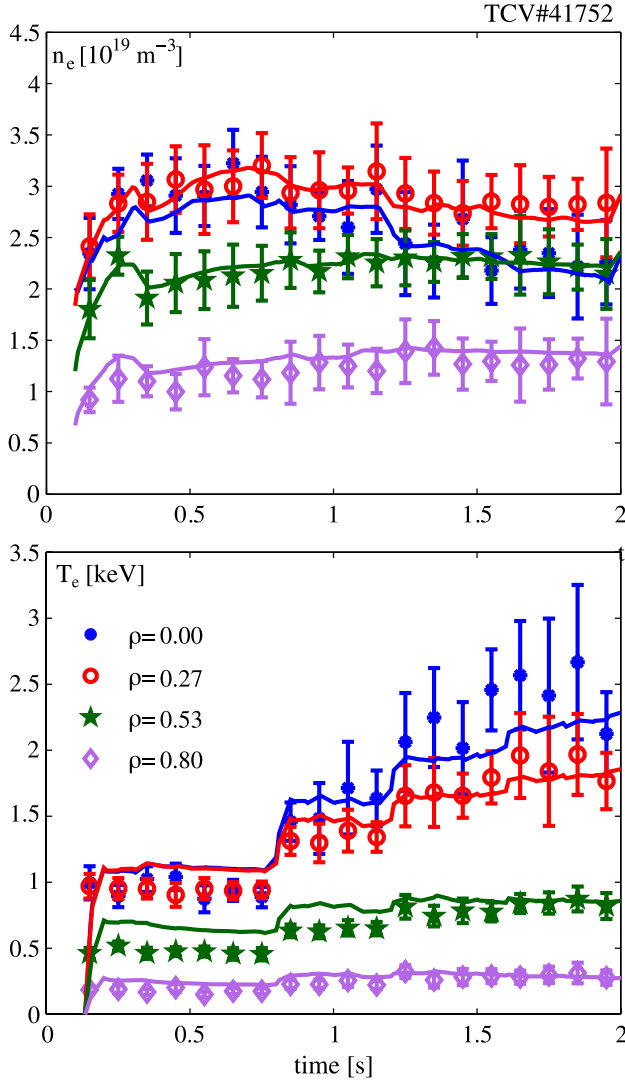


Figure 3. Comparison of post-shot fits of Thomson data for $T_e(\rho)$, $n_e(\rho)$ (points with error bars) and real-time estimates (solid lines) using TENEX to reconstruct the profiles from interferometer and x-ray diagnostics. Note that the value of $T_e(0)$ is separately determined using a dedicated real-time diagnostic and only the profile shape is determined by mapping from the multichannel soft-x ray diagnostic. During this shot the EC heating power was increased in steps, causing an increase in T_e and a hollow density profile due to pump-out effects.

clock rate, allowing algorithms of different complexity, and therefore with different timing requirements, to be run in parallel. One of the main advantages of the new system is the ability to program control algorithms using the Simulink block diagram language²; these are automatically converted to C code and compiled for execution on each individual RT node. Modifications to the Simulink diagram can be made in a matter of minutes and are often made in between TCV shots.

Exploiting this ease of programming, part of RT-RAPTOR has been implemented in Simulink, i.e. only that part of the computations, reduced to a minimum, that must be performed at each time step. The preparation for real-time computations is done in Matlab, including (i) loading a pre-defined equilibrium,

² www.mathworks.com.

(ii) pre-computing the finite-element basis functions and matrices required for differentiation and integration and (iii) preparing variables used later for assembling the elements of the linear system. By keeping most of this ‘bookkeeping’ out of Simulink, a relatively efficient block diagram is obtained.

The overall structure of the TCV implementation of the interpretative transport simulations is shown in figure 4. Each of the four rectangles in the figure corresponds to a distinct node which is described below

- The first node (RT01) has its ADCs connected to both FIR and DMPX diagnostics in the TCV hall. On this node, both the TENEX diagnostic mapping and the RT-RAPTOR algorithm are hosted. Feedback controllers which use the data from RT-RAPTOR are also hosted in this node but are not discussed in this paper. This node runs with a step time of 900 μ s which, for this set of algorithms, is close to the limit of its computational capabilities, and features a (relatively modest) 2.2 GHz Intel® Core™ 2 CPU.
- The second node (RT02) is connected to the full set of magnetic and coil current measurements. A software replica of the hybrid analogue TCV control system [25], which controls the plasma current, position, shape (partly) and density, runs here at a sample time of 100 μ s. In these experiments, this node provides real-time estimates of plasma current and loop voltage. These quantities are written into the reflective memory at each time step and are thus available to other nodes, in particular RT01 where $I_p(t)$ is used as a boundary condition for RT-RAPTOR.
- The third node is connected to the network, reads selected data from the shared memory and displays it in real-time on a screen in the control room.
- The fourth node is left in stand-by but will be used to perform other tasks such as hosting a real-time Grad-Shafranov solver at a future date.

5. Real-time simulation results and comparison with off-line ASTRA and LIUQE

The RT-RAPTOR algorithm was first tested during several plasma shots on the TCV tokamak. The digital real-time control system was not controlling the TCV plasma, but running the simulation in the background using the real-time acquired diagnostic data. As mentioned in the previous section, the time step for the simulation and profile reconstruction was 0.9 ms, more than two orders of magnitude below the TCV current redistribution time for heated plasmas ($\tau_{\text{crt}} \sim 150$ ms). While this could be improved further by using higher performance CPUs and by parallelization, it was sufficient for the given application. A grid size of 41 equally spaced radial points was used, ensuring a largely sufficient spatial resolution with respect to the spatial scales present.

It should be noted that, in the results presented, the collisionality dependent terms in the bootstrap and neoclassical conductivity expressions (c_{neo} in (11) and $\mathcal{L}_{31}, \mathcal{L}_{32}, \mathcal{L}_{34}, \alpha$ in (12)) were not evolved following changes to T_e and n_e profiles. These were calculated based on a reference T_e, n_e profile and kept constant thereafter. This effect could be included in the future since the collisionality-dependent terms and profiles are

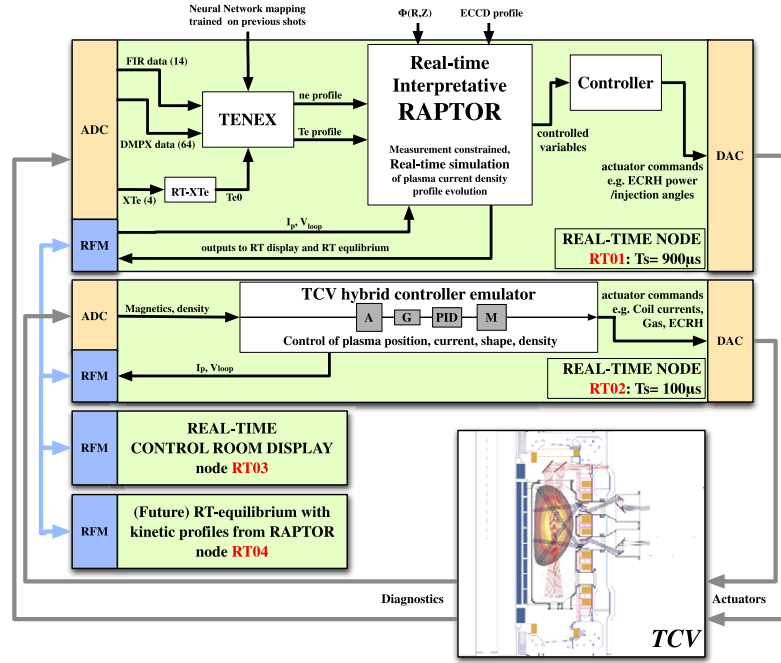


Figure 4. Diagram of the implementation of RT-RAPTOR in the distributed TCV digital control system. The main algorithm is run on node RT01 which is connected to the profile diagnostics needed to reconstruct the T_e and n_e profiles. Measurement of I_p is provided by a second node (RT02) which runs a functionally identical copy of the TCV analogue control system. Each node is connected to a different set of diagnostics via analogue to digital converters (ADCs) and commands a different (sometimes overlapping) set of actuators via DACs (Digital to Analog Converters). Information is exchanged via a real-time reflective (shared) memory (RFM). The third and fourth nodes (RT03, RT04) are now passively observing data generated from the other two nodes, but could be used for other applications in the future. Details of the architecture of the control system hardware and software can be found in [11].

related by analytical expressions in [19, 20], but were skipped in this first implementation for simplicity.

For these experiments, RT-RAPTOR adapts in real-time the value of Z_{eff} . As the measurement of V_{loop} , as measured by a flux loop outside the vessel, is redundant (since the boundary condition has already been specified via (15)), it can be compared with the simulation output $U_{\text{pl}}(\rho = \rho_e, t)$. The value of the Z_{eff} parameter is then increased or decreased, depending on whether the plasma voltage is overestimated/underestimated (corresponding to too high/too low Z_{eff} , respectively). This is implemented into the real-time loop in the form of a parameter adaptation law for Z_{eff} , using the simple expression

$$Z_{\text{eff},k+1} = Z_{\text{eff},k} + \gamma[V_{\text{loop,meas}} - U_{\text{pl}}(\rho_e)]. \quad (38)$$

Here, γ is the adaptation gain which is tuned empirically. A more general discussion of this idea is given in section 6.

To illustrate the capabilities and typical results of the code, we present two different types of shots. Time traces obtained in real-time are compared, where possible, to off-line data obtained from (a) the off-line Grad-Shafranov equilibrium code LIUQE [26], which reconstructs the plasma magnetic equilibrium using magnetic measurements and a constraint on the plasma stored energy via a DML (Diagnomagnetic loop) diagnostic, and (b) an off-line interpretative transport simulation performed using ASTRA [2]. The boundary and plasma current used in the ASTRA simulations were taken from the LIUQE equilibrium data, while kinetic profiles were taken from Thomson scattering measurements.

5.1. Plasma current ramps

As a first and basic test of the current diffusion simulation, we present an example plasma in which the current is forced to diffuse by repeatedly varying the plasma current I_p between 240 and 160 kA with different linear ramp rates. We use this example to focus on some of the magnetic quantities calculated by RT-RAPTOR. A selection of the real-time data coming from the RAPTOR flux profile reconstruction is shown in figure 5.

We mention four important events occurring during the evolution of this plasma:

- $t = 0.3$ s: the control loop for elongation feedback is closed, bringing the plasma to the required elongation. This leads to transients in the coil currents which induce some sharp variations in loop voltage.
- $t = 0.4$ s: the first current ramp is initiated, bringing the plasma current from 160 to 240 kA in 50 ms, and simultaneous switch-on of ECH power, leading to increased temperature and decreasing loop voltage due to the lower plasma resistivity.
- $t = 1.0$ s: a slower ramp-down is imposed, to bring I_p back to 160 kA in 100 ms. The lower current gives lower particle confinement such that a lower density is obtained thus higher temperature at constant auxiliary input power.
- $t = 1.6$ s: an even slower ramp-up back to 240 kA is performed in 200 ms.

The profile of $I_{\text{pl}}(\rho, t)$ is plotted in figure 5(a) for distinct values of ρ , showing how the change in I_p (controlled by feedback) penetrates the plasma on the current redistribution

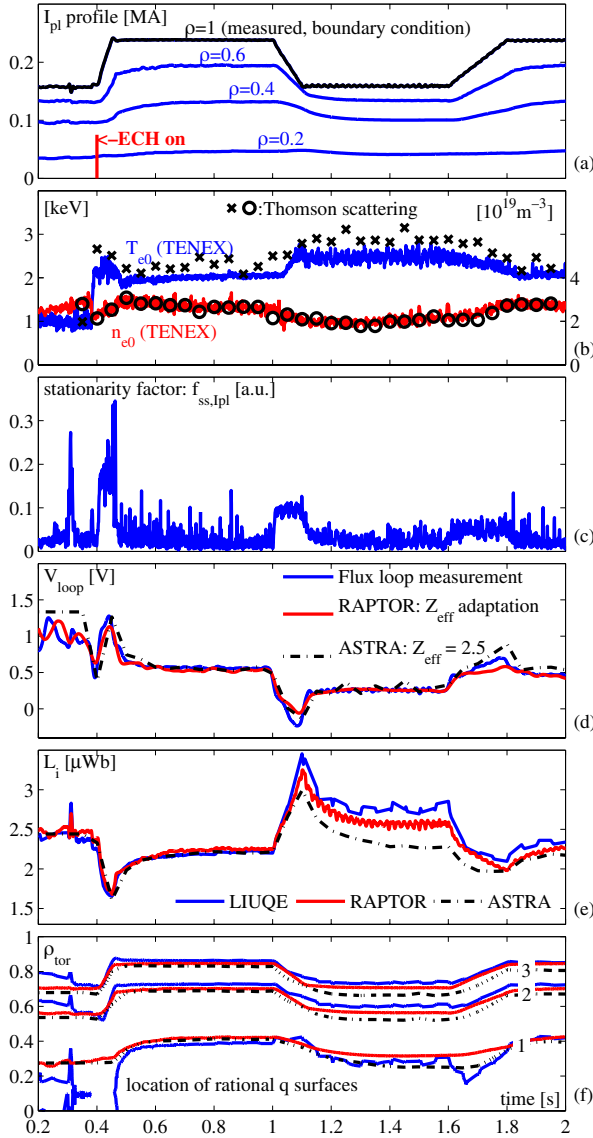


Figure 5. Illustration of real-time data output by the RAPTOR real-time simulation, on a shot with several changes in I_p (TCV#41479). Panel (a) shows the evolution of the $I_p(\rho)$ profile for different values of ρ . The edge value is constrained to the measured value as a boundary condition. The evolution of the profile for inner flux surfaces shows the inward diffusion of the plasma current. The second plot (b) shows the central T_e and n_e as estimated from the real-time XTe diagnostic which is used for calculating the conductivity and (small) bootstrap current. Also shown (c), is an indicator of the degree of stationarity of the I_p profile. A number of transient phases in the profile evolution, followed by decay to a stationary state on a current redistribution time scale, are evident. Comparing V_{loop} (d), L_i (e) and the q (f) surface locations to measured data and off-line reconstructions using the LIUQE equilibrium code and ASTRA interpretative transport modeling shows that the real-time results are very close to those obtained post-shot. The discrepancy in the L_i estimates for $t > 1.1$ s is due to an unwanted increase in the plasma elongation, but the time evolution is still qualitatively correct.

time scale as expected. The values of T_{e0} obtained from the XTe diagnostic are shown, in figure 5(b), to slightly underestimate the off-line Thomson measurements but are within reasonable margins of the measured values.

This plasma provides an opportunity to illustrate the stationarity for I_p as defined in (33). The reader should recall that it is the square norm of the profile $\frac{\partial}{\partial t} I_p(\rho)$ which is related to the spatial derivative of the voltage profile. A high value of $f_{ss,Ipl}$ indicates that the I_p profile is evolving, while a zero value indicates it is stationary. In figure 5(c), one can clearly distinguish a phase of constant f_{ss} while the ramp-up is being performed, and a subsequent relaxation on the resistive time scale after I_p has reached its target value. Note the longer phase of increased $f_{ss,Ipl}$ corresponding to the lower current ramp rate at $t = 1.6$ s, and the shorter, higher peak in variability for the rapid current ramp at $t = 0.4$ s. Indeed, this quantity shows the degree to which the current profile has relaxed to a stationary state.

Figure 5(d) compares the loop voltage from RAPTOR, where Z_{eff} was adapted, to the measured loop voltage. This same loop voltage is also compared with the loop voltage from a post-shot interpretative transport simulation using ASTRA, showing good agreement even in the transient phases. Finally, we also plot the internal inductance and radial location of rational q surfaces in figures 5(e) and (f). For the most part the real-time results show excellent agreement with the results from off-line analysis. The internal inductance estimates diverge somewhat during the second part of the shot, but this can be explained by the fact that the plasma shape (elongation) increases during this phase, in spite of the elongation feedback, due to one PF coil behaving unexpectedly for technical reasons. While this is taken into account in the equilibrium reconstruction from LIUQE, the RAPTOR result assumes a fixed equilibrium and is not aware of this change. Nevertheless, the estimate lies between the value estimated by LIUQE and ASTRA.

5.2. Auxiliary power steps

In a second shot we investigate confinement quantities such as W_{kin} , τ_E and confinement scaling factors, by imposing a sequence of steps of second harmonic X-mode (X2) ECH heating power shown in figure 6(a). The kinetic profile reconstruction for this shot was previously illustrated in figure 3. The power steps will yield changes to the plasma power balance, and RT-RAPTOR can be used to compute this power balance in real-time as we will see.

The sequence of EC power steps causes a change in electron temperature and an increase in (electron) thermal energy (figure 6(b)), but also increased conductivity leading to a redistribution of the current density and q surface locations (figure 6(f)), and decreased loop voltage (figure 6(d)).

The value of Z_{eff} was, like in the previous example, adapted in real-time in order to match the measured loop voltage. The measured loop voltage is also compared with the value obtained from an off-line ASTRA simulation using Thomson profile data and a fixed value of $Z_{eff} = 2.5$, and shows a very similar evolution. The downward drift in estimated Z_{eff} appears unrealistic (an increase is more likely with auxiliary power) but can be explained by errors in the T_e reconstruction. As can be observed from figure 3, the initial T_e is overestimated while the final T_e is underestimated. This would explain why the adaptive algorithm compensates by initially increasing the Z_{eff} estimate and decreasing it

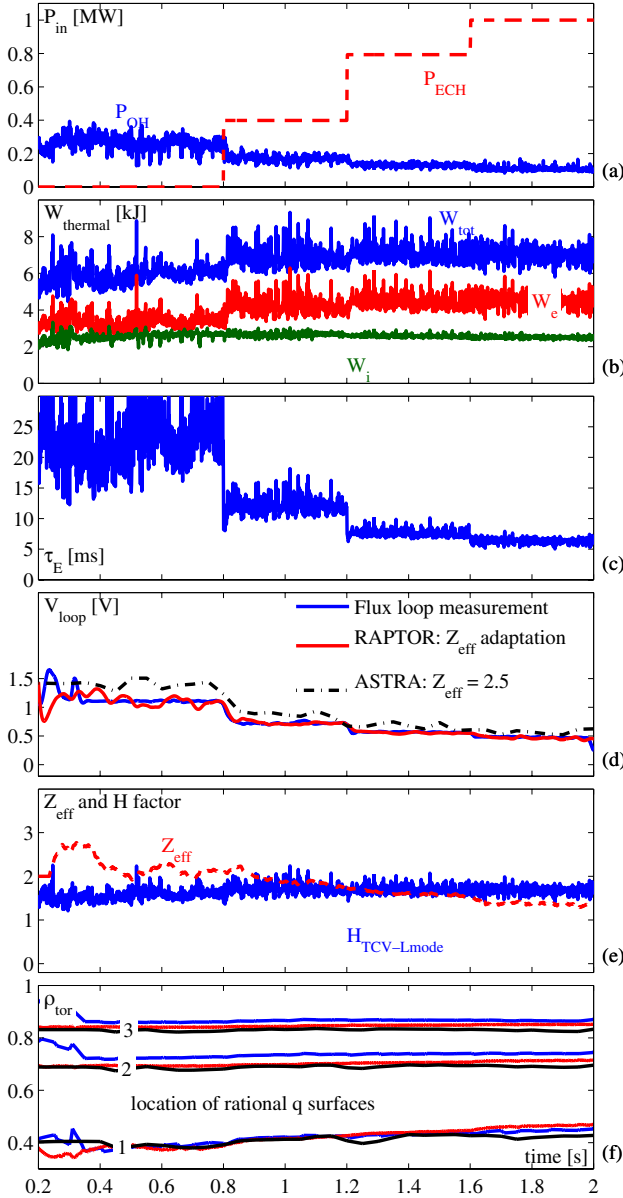


Figure 6. Illustration of several real-time signals available from the interpretative transport simulation done using RAPTOR during a TCV shot. During this particular shot, the ECH power is increased in steps and ohmic power drops due to the decreased loop voltage (a). At the same time the plasma energy content is seen to increase (b). The confinement decreases as expected for heated plasmas (c) while the $H_{\text{TCV-Lmode}}$ scaling factor stays approximately constant. The loop voltage (d) is shown as measured by a flux loop, as tracked by RAPTOR by adapting Z_{eff} (e) and from off-line ASTRA simulations with fixed Z_{eff} . The movement of the rational q surfaces due to the internal redistribution of current density is also shown for RAPTOR, ASTRA and from the LIUQE off-line equilibrium reconstruction code. Other parameters for this discharge (TCV#41752) are $I_p = 240$ kA, $\kappa = 1.5$, $\delta = 0.4$.

during the shot. This example illustrates that an accurate reconstruction of T_e is crucial for accurate estimation of Z_{eff} using this method. The noise in the confinement data can be explained as coming directly from the DMPX diagnostic which is used in the determination of T_e , where the high-frequency oscillations are largely result from sawteeth. Since the flux profile evolution evolves at a slower time scale, the simulation

naturally filters out any high-frequency content in the T_e signal, and no further filtering is necessary.

The extra auxiliary power causes a net decrease in confinement time as expected from known scaling laws (figure 6(c)). Comparing the confinement time with the TCV L-mode scaling, which follows the Rebut-Lallia-Watkins scaling [27, 28], reveals an $H_{\text{TCV-Lmode}} = \tau_E / \tau_{\text{RLW}}$ factor of just under 2 (figure 6(d)) as is typically the case when the EC power is deposited inside of the $q = 1$ radius. We stress again that the quantities shown here are now available in real-time during the shot, rather than post-shot from transport analysis, and may now be used for monitoring and control purposes.

5.3. Discussions and possibilities for improvement

All of the simulations shown above gave satisfactory results within error margins expected from diagnostics. Whenever the real-time simulation results have been compared with post-shot (off-line) simulations, good agreement was obtained taking into account the different information available in the various cases. The experience gained operating this real-time interpretative transport code during TCV tokamak experiments has, however, allowed us to highlight a number of possible improvements for the system as it is implemented today.

As illustrated in section 4.1, TCV currently lacks a real-time T_e profile diagnostic and the reconstruction is done using indirect means, heavily relying on a single chord T_{e0} estimate and multi-chord x-ray wire chamber. This has often led to errors in the T_e profile estimate which is a crucial quantity strongly influencing conductivity, bootstrap current and confinement estimates. Particularly if these real-time simulations are to be applied to advanced scenarios with significant bootstrap current fractions, the correct determination of the T_e profile with steep gradients will be crucial in simulating the correct current density profile, (although to some extent this requirement may be alleviated by using an ad hoc method to estimate the T_e gradients specifically in these plasmas). Secondly, the simulation currently requires a number of similar shots to be performed beforehand, with an adequate range of heating and density conditions to establish the neural-network mapping from real-time diagnostics to kinetic profiles. A new shot database must be constructed if the plasma shape or position is significantly altered. It would be a major advantage to have a real-time T_e and n_e profile measurement of high spatial accuracy to be able to improve on the results, avoiding the neural-network based fits described in section 4.1 entirely. Furthermore, any additional measurements of internal profile quantities would provide extra constraints and redundancy in the system and yield improved current density profile estimates. Finally, instead of using one static equilibrium, one could treat time-varying plasma equilibria by pre-calculating a set of equilibria based on the planned discharge parameters.

An outlook on how the present system could be extended and implemented on ITER and other tokamaks is given in section 7.

6. Handling model-reality mismatch

In the description of the physics model in section 3 and in the TCV implementation discussed so far, we assumed that the

model was perfect in the sense that it perfectly reproduces the physical evolution of the flux profile. In this section we will discuss, in a rather general way, how the inevitable model-reality mismatch can be treated. We will point out the possible uses of redundant diagnostic signals to pinpoint and correct errors of different origin, and will lay out three possible approaches. A discrepancy between measured and simulated quantities (residual) may roughly be categorized as due to an erroneous initial condition, to an incorrect choice of physics parameters, or due to unmodelled processes or disturbances. In the design of a more advanced real-time simulation, the choice must be made as to how to link each type of residual to a source of error. This is a design choice which varies depending on the available measurements and the degree of confidence placed on each part of the physics model. One possibility which has already been implemented as described in section 5 is to use redundant measurements to adapt uncertain physics parameters: in this case V_{loop} was used to adapt Z_{eff} . This section will generalize this idea as well as discussing alternatives.

6.1. Closed loop observer

As was mentioned in the introduction, the real-time simulations in this paper are known in control systems theory as dynamic state observers. Just as our simulations, they evolve the dynamic equations of the system. The standard way to handle measurements in the framework of state observers is to add a term to the equations which feeds back the error between measured system outputs and predicted system outputs with an observer gain. This observer gain is chosen such that the observer error (the difference between true and observer state) converges to zero following a specified dynamics dictated by the characteristics of the system and the observer gain. Our case, where no internal measurements for the flux profile are available, is equivalent to using an open-loop observer, i.e. choosing zero observer gain. Since the system is stable (from the diffusive nature of the equations), the observer error will decay to zero asymptotically by the same stable dynamics that governs the overall profile evolution. This property is known in the literature as *asymptotic observability* [29, chapter 6,7].

If internal measurements of some function of ψ were available, we could construct a closed-loop observer by adding an observer error feedback term to the flux diffusion equation chosen such that the observer error decays on some fast time scale, faster than the diffusion time of the system. Using a closed-loop observer allows us, in theory, to reduce the effects of measurement noise and imperfect initial condition. This approach is natural when modeling errors are assumed to be small: the observer can then effectively filter out measurement noise. State observers for PDEs are extensively treated in [30] which may be used as a starting point for further work in this direction.

6.2. Adaptation of model parameters

If high-quality, trustworthy measurements are available, but the parameters in the physics model are less well known, it may be appropriate to adapt the model parameters in order to reduce the residual error. The key difference between the closed-loop observer and the parameter adaptation approach

is that in the first case one directly corrects the estimate of the state, while in the second case one corrects the parameters of the model generating the state estimate. We already mentioned in section 5 the particular case of adapting the effective charge Z_{eff} by comparing the V_{loop} measurement with the simulated value. Of course the above methodology assumes that all of the other model parameters and measurements are correct. In practice, errors in, for example, the reconstruction of T_e would perturb our estimate of Z_{eff} . A more complete approach would attempt to make consistent estimations of all quantities, including measurement and parametric uncertainties. Also, adding further redundant measurements clearly increases the number of parameters which can be constrained and improves the quality of the reconstructions. Full analysis and development of this approach is beyond the scope of this paper and is left as future work.

6.3. Disturbance estimation and fault detection

Finally, we discuss a third option for classifying a mismatch between measurements and simulations, that of a disturbance or fault. In the case of a disturbance, some unexpected or unmodelled effect causes the simulations and measurements to diverge. By assuming a particular structure for the disturbance, its magnitude and distribution can be computed from the residual error. Similarly, one can construct a typical signature of the residual which would arise from faults in a particular actuator or diagnostic. Correlating the occurring errors with the signature of each possible fault can aid in model-based diagnosis of problems in real-time. This field of model-based fault detection and identification (FDI) has been little explored for tokamaks so far but may be of great interest for the future.

We conclude this section by mentioning that the three approaches mentioned above can be treated in a unified manner by e.g. the extended Kalman filter method, where both the system states and parameters are estimated [31]. This has recently been proposed [32] for estimating transport parameters. The full development of an adaptive, closed-loop state observer including disturbance rejection and fault detection would be a very interesting avenue for research.

7. Real-time simulation on ITER and other tokamaks

Before concluding this paper, we provide an outlook on the possible future application of real-time simulations on ITER and other (existing and future) tokamaks. ITER will certainly have a real-time Grad-Shafranov equilibrium solver, allowing a real-time update of the flux surface geometry based on magnetic measurements, without having to choose a fixed equilibrium. The reconstructed current density profile can also be fed back into the real-time equilibrium code, providing a better profile than can be obtained from a purely diagnostics-based approach, not being limited by diagnostic spatial or temporal resolution. Additionally, any available real-time internal current density profile measurements such as MSE or polarimetry should be included in a closed-loop observer as outlined in section 6.1, rather than fed directly to a q profile feedback controller. When the density profile is available, real-time ray tracing would give the ECCD distribution providing

better current drive modeling. Self-consistent modeling of other heating sources such as NBI could also be included. While the combination of the above features would require significant computational effort, one should note that they can each be performed separately on different, dedicated computers, and require only exchange of the final results. As such, we foresee no technical obstacles to deploying this technique to ITER.

The feasibility of the proof-of-principle implementation on TCV, with its relatively short time scales, shows that today's large tokamaks could certainly use this approach. Real-time equilibrium reconstructions codes (e.g. RT-EFIT [33] or EQUINOX [34]) exist and most tokamaks have multiple real-time diagnostics available [35, 36]. The extra computational effort required to include time-varying flux surface shapes is non-negligible, but will be offset by the longer time scales of larger devices, making the implementation viable. For any tokamak, the accuracy of the real-time simulations will largely depend on the quality and resolution of the available kinetic profile measurements and their availability in real-time, which is a crucial condition for the method to be applicable.

In the introduction, we mentioned that the real-time simulation paradigm is applicable, in principle, to any profile including kinetic (T_e , n_e , ...) profiles, and would hold similar advantages as for the current density profile. However, kinetic profiles are more difficult to simulate in real-time, partly because they evolve more rapidly but especially because first-principle models are not yet sufficiently developed to allow accurate and rapid simulation. Ad hoc models could be used in these cases, similar to those used in [37], for example. These models would have to be tuned depending on the plasma scenario, but given the experience operating today's tokamaks, reasonable results can be expected. Fortunately, high-quality diagnostic measurements of kinetic profile quantities are often available, so a real-time simulation would be able to rely more heavily on the diagnostics and less on the model than would be the case for the current density profile. Although the TCV energy confinement times (~ 1 ms) probably preclude kinetic profile simulation on TCV with the present architecture, the time scales involved on today's large tokamaks, and even more so on ITER, would probably be sufficiently long for kinetic profile simulations. Further research and better understanding of kinetic profile transport could also yield more tractable, accurate and general models.

Finally, real-time prediction and model-predictive control, as mentioned in section 2, requires faster-than-real-time predictive simulation capabilities, which may be beyond today's computing power for small tokamaks and may be at the limit of the capabilities for existing tokamaks. However for the ITER time scales this will most likely be possible, especially when relying on ad hoc models as mentioned above and when simplifying the dynamical model, for example by local linearization.

8. Conclusions

In this paper we have presented a first implementation of a real-time simulation of the diffusion of poloidal flux in a tokamak plasma. This allows full reconstruction of the plasma state including quantities otherwise difficult to measure, such as the

internal current density distribution, parallel electric field and many more, and yields reconstructions with higher accuracy, temporal and spatial resolution than can be obtained from diagnostic measurements alone. We have discussed multiple applications of such an implementation, ranging from feedback control, estimation and monitoring of physics parameters of an ongoing tokamak discharge, disruption avoidance and predictive control.

The simulation is based on the numerical solution of a PDE describing the radial diffusion of poloidal flux, employing real-time kinetic profile measurements to calculate the conductivity and bootstrap current and using the plasma current as a boundary condition. In this initial implementation, the underlying 2D Grad-Shafranov equilibrium does not evolve during the simulation. This assumption only has a small effect as long as the actual equilibrium does not deviate too far from the chosen reference equilibrium.

The real-time simulations use a new finite-element code called RAPTOR (RAPid Plasma Transport simulatOR) which is embedded in the real-time control system of the TCV tokamak and uses real-time estimates of the kinetic profiles T_e and n_e provided indirectly by a nonlinear mapping of data from several diagnostics. The first experiments have shown very good agreement of the quantities reconstructed in real-time by RT-RAPTOR with those from post-shot calculations, the latter computed with the Grad-Shafranov equilibrium code LIUQE and interpretative transport simulations using ASTRA. For a shot with varying plasma current, the inward diffusion of the current density distribution, and evolution of the q profile are well reproduced. It has also been shown that during a shot with increasing auxiliary power, the confinement properties such as confinement time and H factor can be calculated. This shows that the real-time simulation is capable of producing, even assuming a fixed equilibrium, the evolution of the internal current density profile with similar quality as is obtained from off-line analysis. It is found that a correct estimate of the electron temperature profile is essential due to the sensitivity of the poloidal flux transport to this quantity, and a high-quality diagnostic measuring this profile is beneficial.

In this new paradigm, physics-based real-time simulation and real-time diagnostics work in synergy, with the diagnostics providing crucial data and constraints to the simulation and the simulation providing the basis for real-time validation of the diagnostic's measurements. More real-time diagnostic measurements would improve the accuracy of the reconstructed profiles, as well as further constrain some of the unknown or uncertain physics parameters governing the poloidal flux transport. Synergy with real-time equilibrium reconstruction codes is also fruitful: the toroidal current density estimate can be fed back to the Grad-Shafranov code. This current density will have a higher spatial accuracy than can be obtained from a discrete set of internal measurements and the internal profile description will not be limited to the space spanned by a small number of basis functions. Linking the 2D flux map with the 1D profiles in this way yields a complete and self-consistent picture of the plasma state. In this sense, the approach we put forward ideally complements existing tools and methods, unifying available measurements with known physics.

In summary, this paper offers a new framework for real-time estimation of tokamak current density profiles for control

applications. Full knowledge of the plasma state obtained this way could now be exploited on existing experiments to improve the reliability, repeatability, performance, and safety of present and future devices. The presented approach is fully general and could readily be deployed on other tokamaks provided a given set of real-time capabilities are available.

Appendix A. Numerical implementation

This appendix will treat some details of the numerical implementation, including the spatial and temporal discretization of the poloidal flux diffusion equation as well as some measures which have been taken to render the problem computationally efficient for real-time applications. Appendix A.1 will treat the spatial discretization of the PDE using finite elements, appendix A.2 will briefly show the time discretization employed. Appendix A.3 will show some details of how the numerical integration necessary for the real-time computations is treated using Gaussian quadrature. Finally appendix A.4 will show an example of how output quantities such as the rotational transform can be calculated efficiently based on knowledge of the poloidal flux profile.

A.1. Finite-element method

Consider the general case of a time-varying, inhomogeneous PDE of the form

$$m(\rho, t) \frac{\partial y}{\partial t} = \frac{\partial}{\partial \rho} \left[g(\rho, t) \frac{\partial y}{\partial \rho} \right] + k(\rho) j(\rho, t), \quad (\text{A.1})$$

where $y(\rho, t) : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$, $0 \leq \rho \leq \rho_e$ and $t_0 \leq t \leq t_f$. The infinite-dimensional problem in ρ is transformed into a finite dimensional problem using the finite-element approach (for which a vast literature exists, e.g. [38]). An important advantage of using finite element methods is that the basis functions can easily be modified, and also that the order of spatial derivatives of the elements involved are, as we shall see, one order lower than the order of the PDE.

First, approximate the solution by

$$y(\rho, t) \approx \sum_{\alpha=1}^{n_{\text{sp}}} \hat{y}_{\alpha}(t) \Lambda_{\alpha}(\rho). \quad (\text{A.2})$$

We choose polynomial B-splines [39] on a finite support as finite elements Λ_{α} . Choosing a set of (possibly irregularly spaced) *knots* $0 = x_1 < \dots < x_N = \rho_e$ the basis functions are defined recursively as

$$\Lambda_{\alpha}^0(\rho) = 1 \text{ if } x_{\alpha} \leq \rho < x_{\alpha+1}, \quad 0 \text{ otherwise}, \quad (\text{A.3})$$

$$\Lambda_{\alpha}^p = w_{\alpha-1}^p \Lambda_{\alpha-1}^{p-1} + (1 - w_{\alpha}^p) \Lambda_{\alpha}^{p-1}, \quad (\text{A.4})$$

$$w_{\alpha}^p = \frac{\rho - x_{\alpha}}{x_{\alpha+p} - x_{\alpha}}. \quad (\text{A.5})$$

We obtain non-periodic splines by defining additional knots on the domain boundary: $x_{-p+1} = \dots = x_1$ and $x_{N+p} = x_{N+p-1} = \dots = x_N$. For a given spline order and set of knots we can construct $n_{\text{sp}} = p + N - 1$ unique splines. Note that as a result of the finite support of the elements, $\Lambda_{\alpha}^p(\rho) \neq 0$ if and only if $x_{\alpha-p} < \rho < x_{\alpha+1}$.

Substituting this into (A.1)

$$\sum_{\alpha=1}^{n_{\text{sp}}} m \frac{d\hat{y}_{\alpha}(t)}{dt} \Lambda_{\alpha}(\rho) = \sum_{\alpha=1}^{n_{\text{sp}}} \hat{y}_{\alpha}(t) \frac{\partial}{\partial \rho} \left[g \frac{\partial \Lambda_{\alpha}(\rho)}{\partial \rho} \right] + k j, \quad (\text{A.6})$$

where we have dropped the (ρ, t) dependences for notational simplicity, we can construct the weak form by projecting the equation for each α onto a trial function Λ_{β} and integrating over the domain

$$\begin{aligned} \sum_{\alpha=1}^{n_{\text{sp}}} \frac{d\hat{y}_{\alpha}(t)}{dt} \int_0^{\rho_e} m \Lambda_{\beta} \Lambda_{\alpha} d\rho \\ = \sum_{\alpha=1}^{n_{\text{sp}}} \hat{y}_{\alpha}(t) \int_0^{\rho_e} d\rho \left(\Lambda_{\beta} \frac{\partial}{\partial \rho} \left[g \frac{\partial \Lambda_{\alpha}}{\partial \rho} \right] \right) \\ + \int_0^{\rho_e} d\rho \Lambda_{\beta} k(\rho) j, \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} \sum_{\alpha=1}^{n_{\text{sp}}} \frac{d\hat{y}_{\alpha}(t)}{dt} \underbrace{\int_0^{\rho_e} m \Lambda_{\beta} \Lambda_{\alpha} d\rho}_{M_{\beta\alpha}(t)} \\ = - \sum_{\alpha=1}^{n_{\text{sp}}} \hat{y}_{\alpha}(t) \underbrace{\left[\int_0^{\rho_e} g \frac{\partial \Lambda_{\beta}}{\partial \rho} \frac{\partial \Lambda_{\alpha}}{\partial \rho} d\rho \right]}_{=D_{\beta\alpha}} \\ + \underbrace{\left[g \Lambda_{\beta} \frac{\partial y}{\partial \rho} \right]_0^{\rho_e}}_{=l_{\beta}} + \underbrace{\left[\int_0^{\rho_e} \Lambda_{\beta} k j d\rho \right]}_{=s_{\beta}}. \end{aligned} \quad (\text{A.8})$$

$$\quad (\text{A.9})$$

Note that by integrating by parts, we were able to reduce the order of the maximum radial derivative to be evaluated. This yields an equation for each β of the form

$$\sum_{\alpha=1}^{n_{\text{sp}}} M_{\beta\alpha} \frac{d\hat{y}_{\alpha}}{dt} = \sum_{\alpha=1}^{n_{\text{sp}}} -D_{\beta\alpha} \hat{y}_{\alpha} + l_{\beta} + s_{\beta}, \quad (\text{A.10})$$

which can be written in the matrix form as

$$\mathbf{M} \frac{d\hat{\mathbf{y}}}{dt} = -\mathbf{D} \hat{\mathbf{y}} + \mathbf{l} + \mathbf{s}, \quad (\text{A.11})$$

where $\mathbf{M} \in \mathbb{R}^{n_{\text{sp}} \times n_{\text{sp}}}$, $\mathbf{D} \in \mathbb{R}^{n_{\text{sp}} \times n_{\text{sp}}}$, $\mathbf{l} \in \mathbb{R}^{n_{\text{sp}}}$, $\mathbf{s} \in \mathbb{R}^{n_{\text{sp}}}$ and $\hat{\mathbf{y}} \in \mathbb{R}^{n_{\text{sp}}}$. As a consequence of the finite support of the basis functions, the matrices have limited bandwidth. Furthermore, the finite-element basis functions Λ_{α} and Λ_{β} are usually chosen such that $\Lambda_{\alpha} = \Lambda_{\beta} \forall \alpha = \beta$, in which case the matrices become symmetric.

We now write the weak form of the poloidal flux transport equation (7). By inspection, we see that we can cast (7) into the form (A.1) by choosing

$$m(\rho, t) = \rho \sigma_{\parallel} \mu_0 / J^2 R_0, \quad (\text{A.12})$$

$$g(\rho) = G_2 / J, \quad (\text{A.13})$$

$$k(\rho) = -V' \mu_0 / 2\pi J^2 R_0, \quad (\text{A.14})$$

$$j(\rho, t) = j_{\text{BS}} + j_{\text{aux}}. \quad (\text{A.15})$$

The flux profile is written as a sum of finite elements:

$$\psi(\rho, t) \approx \sum_{\alpha=1}^{n_{\text{sp}}} \hat{\psi}_{\alpha}(t) \Lambda_{\alpha}(\rho) \quad (\text{A.16})$$

Then the weak form is written in the matrix form as

$$\mathbf{M}_\psi \frac{d\hat{\psi}}{dt} = -\mathbf{D}_\psi \hat{\psi} + \mathbf{l}_\psi + \mathbf{r}_\psi, \quad (\text{A.17})$$

where the elements of the matrices are

$$[\mathbf{M}_\psi]_{\beta\alpha} = \int_\rho \frac{\sigma_\parallel \mu_0}{J^2 R_0} \rho \Lambda_\beta \Lambda_\alpha d\rho, \quad (\text{A.18})$$

$$[\mathbf{D}_\psi]_{\beta\alpha} = \int_\rho \frac{G_2}{J} \frac{\partial \Lambda_\beta}{\partial \rho} \frac{\partial \Lambda_\alpha}{\partial \rho} d\rho, \quad (\text{A.19})$$

$$[\mathbf{r}_\psi]_\beta = \int_\rho \frac{-V' \mu_0}{2\pi J^2 R_0} j(\rho, t) \Lambda_\beta d\rho \quad (\text{A.20})$$

and for the boundary term

$$[\mathbf{l}_\psi]_\beta = \left[\frac{G_2}{J} \frac{\partial \psi}{\partial \rho} \Lambda_\beta \right]_{\rho_e} - \left[\frac{G_2}{J} \frac{\partial \psi}{\partial \rho} \Lambda_\beta \right]_0 \quad (\text{A.21})$$

The second term on the right-hand side is zero since $G_2(0) = 0$. The first term of the right-hand side of this expression depends on the chosen boundary condition at the edge. If we choose to fix the boundary flux derivative via the total current $\partial\psi/\partial\rho|_{\rho_e} = \mu_0 I_p / G_2$, we obtain (recalling $J = 1$ at the edge) $[\mathbf{l}_\psi]_\beta = \mu_0 I_p \Lambda_\beta|_{\rho_e}$ which is nonzero only for the last basis $\beta = n_{sp}$. Note that the boundary condition at $\rho = 0$ is not explicitly imposed, but will appear as a natural solution of the physical problem. We can collect the vectors on the right-hand side of (A.17) writing $\mathbf{s}_\psi = \zeta \mathbf{l}_p + \mathbf{r}_\psi$ where ζ is a vector of zeros except for the last element which is μ_0 . Then (A.17) becomes

$$\mathbf{M}_\psi \frac{d\hat{\psi}}{dt} = -\mathbf{D}_\psi \hat{\psi} + \mathbf{s}_\psi, \quad (\text{A.22})$$

where the source term in the flux diffusion equation can therefore be considered as a sum of the effect of injected (auxiliary and bootstrap) current density and a boundary source depending on I_p .

A.2. Time discretization

To discretize the continuous-time equation (A.22), we start by choosing a time grid $t = [t_0, \dots, t_k, \dots, t_M]$. We write a general Crank–Nicholson-type discretization scheme

$$\mathbf{M}_\psi(t_{k+\frac{1}{2}}) \left(\frac{\hat{\psi}_{k+1} - \hat{\psi}_k}{\Delta t} \right) = -\mathbf{D}_\psi(t_{k+\frac{1}{2}}) (\theta \hat{\psi}_{k+1} + (1-\theta) \hat{\psi}_k) + \mathbf{s}_\psi(t_{k+\frac{1}{2}}), \quad (\text{A.23})$$

which is second-order accurate. For practical reasons, we approximate the terms above such as to obtain a stable numerical scheme requiring no iterations, at the expense of being less accurate. This is achieved by choosing

$$\mathbf{M}_\psi(t_{k+\frac{1}{2}}) \approx \mathbf{M}_\psi(t_k) = \mathbf{M}_{\psi,k}, \quad (\text{A.24})$$

$$\mathbf{s}_\psi(t_{k+\frac{1}{2}}) \approx \mathbf{s}_\psi(t_k) = \mathbf{s}_{\psi,k}, \quad (\text{A.25})$$

$$\mathbf{D}_\psi(t_{k+\frac{1}{2}}) = \mathbf{D}_\psi(t_k) = \mathbf{D}_\psi \quad (\text{time-independent}), \quad (\text{A.26})$$

$$\theta = 1. \quad (\text{A.27})$$

This gives the following difference equation, implicit in time:

$$(\mathbf{M}_{\psi,k} + \Delta t \mathbf{D}_\psi) \hat{\psi}_{k+1} = (\mathbf{M}_{\psi,k}) \hat{\psi}_k + \Delta t \mathbf{s}_{\psi,k} \quad (\text{A.28})$$

This linear system is to be solved for each time step yielding an update $\hat{\psi}_{k+1}$. Note that the elements $\mathbf{M}_{\psi,k}$ and $\mathbf{s}_{\psi,k}$ of this linear system are all calculated based on the current time step, k , avoiding the need to iterate: a step forward in time to step $k+1$ is made based only on information available at time step k . Experience suggests that when the time step is taken sufficiently small with respect to the characteristic time scales of the equations, errors introduced by the approximations (A.24)–(A.27) are acceptable. In our case, a time step of 1 ms was found adequate. Each time step requires the solution of the linear system (A.28), the left-hand side of which contains a banded, positive definite, symmetric matrix such that the problem can be efficiently solved by LDL^T (Cholesky) decomposition [40]. In the next subsection we turn to the problem of how to efficiently calculate the terms $\mathbf{M}_{\psi,k}$, \mathbf{D}_ψ and $\mathbf{s}_{\psi,k}$.

A.3. Numerical integration using Legendre–Gauss quadrature

The integrals (A.18), (A.19) and (A.20) are evaluated numerically using Gaussian quadrature [41]. Note that due to the finite support of the basis functions the terms are identically zero except pairs of indices α, β satisfying $\alpha - p \leq \beta \leq \alpha + 1$ where p is the order of the spline. For this set of combinations of α, β , the integrals can be computed efficiently using Gaussian quadrature. On the i th interval $[x_i, x_{i+1}]$ between two knots, define the set of n_g Gauss points $x_{i,j}^g$ for $j = [1, 2, \dots, n_g]$, then define the function $z(x) = (2x - x_{i+1} - x_i) / (x_{i+1} - x_i)$ which linearly maps the interval $x \in [x_i, x_{i+1}]$ to $z \in [-1, 1]$, now the j th Gauss point is the point for which $z(x_{i,j}^g)$ is the j th zero of the normalized Legendre polynomial of order n_g . The corresponding set of weights $w_{i,j}^g$ are given by $w_{i,j}^g = 2(1 - z(x_{i,j}^g)^2)^{-1} [L'_{n_g}(z(x_{i,j}^g))]^{-2}$ where L'_{n_g} is the derivative of that same Legendre polynomial. Then we can approximate the integral

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx \sum_{j=1}^{n_g} w_{i,j}^g f(x_{i,j}^g). \quad (\text{A.29})$$

Now one can approximate, for example, integral (A.18) as follows:

$$\begin{aligned} \int_\rho m(\rho, t) \Lambda_\alpha(\rho) \Lambda_\beta(\rho) d\rho \\ \approx \sum_{i=i_{\min}}^{i_{\max}} \sum_{j=1}^{n_g} w_{i,j}^g \Lambda_\alpha(x_{i,j}^g) \Lambda_\beta(x_{i,j}^g) m(x_{i,j}^g, t). \end{aligned} \quad (\text{A.30})$$

Here i_{\min} and i_{\max} are the indices of the intervals delimiting the domain where $\Lambda_\alpha > 0$. Using n_g points exactly integrates up to polynomial order $2n_g - 1$, thus choosing $n_g = 4$ provides exact integration of the products of up to order 7 (Note that the integrands include a product of two cubic splines which gives order ≥ 6). Thus one needs to evaluate radial quantities on Nn_g radial points. Separating the time-varying part of $m(\rho, t) = \tilde{m}(\rho) \sigma_\parallel(\rho, t)$ where $\tilde{m}(\rho) = (\mu_0 \rho / (J(\rho)^2 R_0))$ we can write each element of $\mathbf{M}_{\psi,k}$ as an inner product

$$[\mathbf{M}_\psi(t_k)]_{\beta\alpha} \approx \xi_{\alpha,\beta}^T \bar{\sigma}_\alpha(t), \quad (\text{A.31})$$

where

$$\xi_{\alpha,\beta} = \begin{pmatrix} w_{i_{\min},1}^g \Lambda_\alpha(x_{i_{\min},1}^g) \Lambda_\beta(x_{i_{\min},1}^g) \tilde{m}(x_{i_{\min},1}^g) \\ \vdots \\ w_{i_{\min},n_g}^g \Lambda_\alpha(x_{i_{\min},n_g}^g) \Lambda_\beta(x_{i_{\min},n_g}^g) \tilde{m}(x_{i_{\min},n_g}^g) \\ w_{i_{\min}+1,1}^g \Lambda_\alpha(x_{i_{\min}+1,1}^g) \Lambda_\beta(x_{i_{\min}+1,1}^g) \tilde{m}(x_{i_{\min}+1,1}^g) \\ \vdots \\ w_{i_{\min}+1,n_g}^g \Lambda_\alpha(x_{i_{\min}+1,n_g}^g) \Lambda_\beta(x_{i_{\min}+1,n_g}^g) \tilde{m}(x_{i_{\min}+1,n_g}^g) \\ \vdots \\ w_{i_{\max},n_g}^g \Lambda_\alpha(x_{i_{\max},n_g}^g) \Lambda_\beta(x_{i_{\max},n_g}^g) \tilde{m}(x_{i_{\max},n_g}^g) \end{pmatrix}, \quad (\text{A.32})$$

and

$$\bar{\sigma}_\alpha(t) = \begin{pmatrix} \sigma_{\parallel}(x_{i_{\min},1}^g, t) \\ \vdots \\ \sigma_{\parallel}(x_{i_{\min},n_g}^g, t) \\ \sigma_{\parallel}(x_{i_{\min}+1,1}^g, t) \\ \vdots \\ \sigma_{\parallel}(x_{i_{\min}+1,n_g}^g, t) \\ \vdots \\ \sigma_{\parallel}(x_{i_{\max},n_g}^g, t) \end{pmatrix}. \quad (\text{A.33})$$

The row vectors $\xi_{\alpha,\beta}^T$ for all combinations (α, β) corresponding to a nonzero element of M_ψ can be combined into a band-structured matrix Ξ . Then defining $\sigma(t)^T = [\sigma_{\parallel}(x_{1,1}^g)^T, \dots, \sigma_{\parallel}(x_{N-1,n_g}^g)^T]^T$, i.e. the vector containing the values of σ_{\parallel} on all Gauss points on all intervals, we can compute *all* the elements of M_ψ with a single matrix-vector product $\Xi \bar{\sigma}(t)$. The elements are then assigned to their corresponding indices. The key advantage is that the entire matrix Ξ can be pre-computed once the equilibrium, knots and finite elements have been defined. In the real-time part, it is only necessary to calculate $\sigma_{\parallel}(t)$ on the grid of Nn_g Gauss points, perform one matrix multiplication and assign the resulting elements to their respective indices positions in the matrix to construct M_ψ .

Generalizing this approach to the other terms (A.19) and (A.20), we see that D_ψ can be entirely pre-calculated as no terms in the integrand depend on time, while r_ψ requires calculating j_{BS} , j_{EC} at the Gauss points and can be obtained in a similar way as described before for M_ψ .

A.4. Calculating simulation outputs by matrix multiplication

The output quantities listed in equations (17), (26) are functions of spatial and temporal derivatives of $\psi(\rho, t)$. Since ψ is given by (A.16), they can be calculated on an arbitrary grid once the finite-element coefficient vector $\hat{\psi}$ is known. For example, the case of the rotational transform $\iota(\rho) = 1/(2\pi B_0 \rho) \partial \psi / \partial \rho$. Defining the matrix, which can be precalculated,

$$C_\iota = \frac{1}{2\pi B_0} \begin{pmatrix} \frac{1}{x_1} \frac{\partial}{\partial \rho} \Lambda_1(x_1) & \cdots & \frac{1}{x_1} \frac{\partial}{\partial \rho} \Lambda_{n_{sp}}(x_1) \\ \vdots & \ddots & \vdots \\ \frac{1}{x_N} \frac{\partial}{\partial \rho} \Lambda_1(x_N) & \cdots & \frac{1}{x_N} \frac{\partial}{\partial \rho} \Lambda_{n_{sp}}(x_N) \end{pmatrix}, \quad (\text{A.34})$$

where $[x_1, \dots, x_N]$ are the knots, such that $C_\iota \hat{\psi}_k$ gives the vector of values of ι at all knots at $t = t_k$, allows efficient

computation of the rotational transform using a single matrix multiplication for each time step. A further refinement is necessary: since $x_1 = 0$ ($\rho = 0$, i.e. the axis), evaluating the top row gives undefined values $0/0$. To evaluate the on-axis ι we resort to l'Hopital's rule and differentiate both numerator and denominator in the expression: $\lim_{\rho \downarrow 0} 1/(2\pi B_0 \rho) \partial \psi / \partial \rho = \lim_{\rho \downarrow 0} 1/(2\pi B_0) \partial^2 \psi / \partial \rho^2$ thus the top row of C_ι is replaced by $[\frac{\partial^2}{\partial \rho^2} \Lambda_1(x_1), \dots, \frac{\partial^2}{\partial \rho^2} \Lambda_{n_{sp}}(x_1)]$.

In completely analogous fashion, we can define matrices for the other output quantities. Quantities which are to be evaluated on the Gauss points can be obtained from a suitably redefined matrix, the rows of which are evaluated on the Gauss points instead of the knots.

Surface and volume integrals, necessary for computing quantities such as W_i and I_{BS} , are recast as integrals over ρ . Taking a general example, $Q_i(\rho) = \int_0^\rho q(\rho) d\rho$ can be evaluated using Gaussian quadrature weights described in the previous section as $Q(x_i) = \sum_{j=1}^l \sum_{j=1}^{n_g} w_{i,j}^g q(x_{i,j}^g)$ which, after some algebra, can also be recast into a matrix multiplication involving a lower-triangular matrix.

A.5. Algorithm breakdown

Having described these computationally efficient methods to calculate the time-varying matrices and the outputs at each time step, we now give an overview of a typical interpretative (RT-)RAPTOR calculation:

- (i) Define spline order, knot locations, time grid and other model parameters, compute Gauss grid points
- (ii) Choose a reference equilibrium, compute fixed geometric terms.
- (iii) Compute all pre-calculable matrices.
- (iv) Choose an initial condition $\hat{\psi}_0$.
- (v) Execute the time loop for each time step $k = [0, M]$
 - Calculate time-varying profiles: given values of $\hat{\psi}$, T_e , n_e and their spatial derivatives at time step k on the grid of Gauss points, calculate σ_{\parallel} (11), j_{BS} (13), j_{aux} (14) on the same points.
 - Assemble the linear system: use the method described in appendix A.3 to calculate $M_{\psi,k}$ and $r_{\psi,k}$, also compute $s_{\psi,k} = r_{\psi,k} + \zeta I_{p,k}$.
 - Solve the linear system (A.28) using Cholesky factorization to obtain $\hat{\psi}_{k+1}$
 - Compute necessary outputs as shown in appendix A.4
- (vi) Compute additional post-run outputs if necessary.

All the pre- and post-processing steps, respectively, (i)–(iv) and (vi), are implemented in Matlab. The real-time step (v) has been implemented both in Matlab and Simulink. The Simulink version can be converted to C code, automatically compiled for the TCV control system and is the part that runs in real-time during a TCV shot.

Appendix B. Real-time estimates of kinetic profiles

This appendix provides further details on the nonlinear neural-network mapping used to reconstruct profiles of $\tilde{T}_e (=T_e(\rho)/T_{e0})$ and n_e based on DMPX (soft x-ray) and FIR (line-integrated density) measurements, briefly introduced in section 4.1.

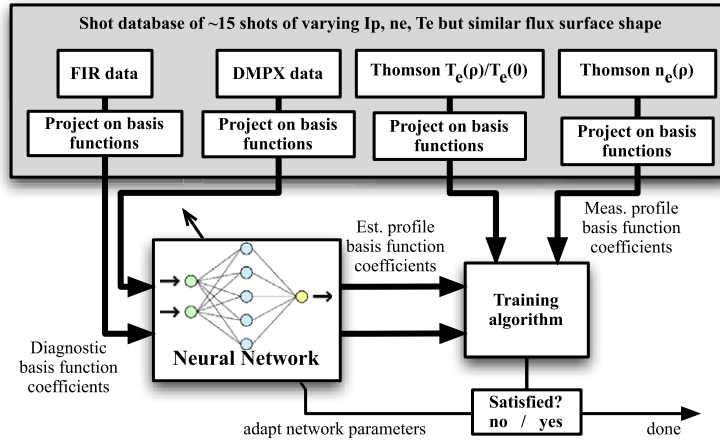


Figure B.1. Schematic representation of the training phase of the ‘TENEX’ algorithm for determining estimates of T_e and n_e profiles from DMPX soft-x ray and FIR interferometer diagnostics. The basis function coefficients for the T_e/T_{e0} and n_e profile are the output of a neural network, trained on data from previous shots.

We first build a database of several shots spanning a range of different heating conditions but similar shape. In practice, about 15 shots, corresponding to approximately 300 time points, were found sufficient as long as the heating conditions were sufficiently diverse. We collect both the post-shot processed and fitted Thomson profiles for $T_e(\rho)$ and $n_e(\rho)$ every 20 ms, and the (lowpass filtered) raw data from all DMPX and FIR channels corresponding to the same time points. The data are collected into matrices, the columns of which contain data for each time point, for example

$$\mathbf{M}_{\text{FIR}} = [\mathbf{m}_{\text{FIR}}(t_1), \dots, \mathbf{m}_{\text{FIR}}(t_p)] \in \mathbb{R}^{n_d \times n_p}, \quad (\text{B.1})$$

where n_p represents the total number of time points available in the database, n_d is the number of diagnostic channels and $\mathbf{m}_{\text{FIR}}(t_1)$ is the vector of acquired FIR values (one data point for each chord) at the time t_1 . Similarly we construct matrices \mathbf{M}_{DMPX} , \mathbf{M}_{n_e} and \mathbf{M}_{T_e} .

To reduce the dimensionality of the problem, the data are parametrized by writing each data vector as a sum of basis functions

$$\mathbf{m}(t_j) = \sum_{i=1}^{n_{\text{bas}}} b_i d_i(t_j). \quad (\text{B.2})$$

The basis functions are determined as the minimum least-squares error fit to the data by computing the singular value decomposition of each matrix and retaining the dominant n_{bas} vectors spanning the matrix column space. Typical values for n_{bas} are 3 or 4 and this number may be different for each diagnostic. Thus, each matrix can be approximated as $\mathbf{M} \approx \mathbf{B}\mathbf{D}$, where $\mathbf{B} \in \mathbb{R}^{n_d \times n_{\text{bas}}}$ is a matrix containing the first n_{bas} orthogonal singular vectors and $\mathbf{D} \in \mathbb{R}^{n_{\text{bas}} \times n_p}$ is the matrix of basis function coefficients. Now for each data vector $\mathbf{m}(t_j)$ the corresponding basis function coefficients are given by $\mathbf{d}(t_j) = \mathbf{B}^+ \mathbf{m}(t_j)$ where $\mathbf{B}^+ = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T$ denotes the pseudo-inverse of \mathbf{B} . This procedure gives us a set of input and output vectors

$$\begin{aligned} \mathbf{d}_{\text{in}}(t_j) &= \begin{bmatrix} \mathbf{d}_{\text{FIR}}(t_j) \\ \mathbf{d}_{\text{DMPX}}(t_j) \end{bmatrix} \in \mathbb{R}^{2n_{\text{bas}}}, \\ \mathbf{d}_{\text{out}}(t_j) &= \begin{bmatrix} \mathbf{d}_{n_e}(t_j) \\ \mathbf{d}_{T_e}(t_j) \end{bmatrix} \in \mathbb{R}^{2n_{\text{bas}}}. \end{aligned} \quad (\text{B.3})$$

$\forall j \in [1, \dots, n_p]$ (time points in the database). We now attempt to find the nonlinear mapping $\mathcal{N} : \mathbb{R}^{2n_{\text{bas}}} \rightarrow \mathbb{R}^{2n_{\text{bas}}}$ such that $\sum_j \|\mathbf{d}_{\text{out}}(t_j) - \mathcal{N}(\mathbf{d}_{\text{in}}(t_j))\|_2^2$ is minimized. For \mathcal{N} we use a neural network which is trained on the available data set using standard techniques [42] such as the Levenberg–Marquardt optimization algorithm, separating the data set into training, validation and test samples with suitable stopping criteria based on the validation set to avoid over-fitting. Once the neural-network parameters have been optimized the network can be used to map a fresh input set of real-time DMPX and FIR measurements to estimated profiles of $n_e(\rho)$ and $\tilde{T}_e(\rho) = T_e(\rho)/T_{e0}$ profile. The overall method is illustrated schematically in figure B.1.

Acknowledgments

We gratefully acknowledge discussions with Drs S. Brunner, T.-M. Tran and J.B. Lister, and thank the referees for providing useful suggestions for improving the original manuscript. This work is supported in part by the Swiss National Science Foundation.

© Euratom 2011.

References

- [1] Svensson J. *et al* 2008 *Plasma Phys. Control. Fusion* **50** 085002
- [2] Pereverzev G.V. *et al* 2002 ASTRA automated system for transport analysis in a tokamak *Technical Report* 5/98 IPP Report
- [3] Artaud J. *et al* 2010 *Nucl. Fusion* **50** 043001
- [4] Hawryluk R.J. 1980 *Physics of Plasmas Close to Thermonuclear Conditions* vol 1, ed B. Coppi p 19
- [5] Budny R. *et al* 1992 *Nucl. Fusion* **32** 429
- [6] Genacchi G. *et al* 1988 JETTO: a free boundary plasma transport code RT/TIB 5 ENEA
- [7] Crotinger J. *et al* 1997 *Technical Report* UCRL-ID-126284 LLNL
- [8] Witrant E. *et al* 2007 *Plasma Phys. Control. Fusion* **49** 1075
- [9] Hofmann F. *et al* 1994 *Plasma Phys. Control. Fusion* **36** B277
- [10] Hinton F.L. *et al* 1976 *Rev. Mod. Phys.* **48** 239

- [11] Paley J. *et al* 2010 *17th IEEE-NPSS Real Time Conf. (RT) (Lisbon, Portugal)* pp 1–6 (doi:[10.1109/RTC.2010.5750487](https://doi.org/10.1109/RTC.2010.5750487))
- [12] Moreau D. *et al* 2008 *Nucl. Fusion* **48** 106001
- [13] Porcelli F. *et al* 1996 *Plasma Phys. Control. Fusion* **38** 2163
- [14] Sauter O. *et al* 1999 *Proc. Joint Varenna–Lausanne Int. Workshop on Theory of Fusion Plasmas Varenna (Italy, 31 August–4 September 1998)* vol ISPP-18 pp 403–8 <http://infoscience.epfl.ch/record/123899>
- [15] Sauter O. *et al* 2002 *Plasma Phys. Control. Fusion* **44** 1999
- [16] Ou Y. *et al* 2007 *Fusion Eng. Des.* **82** 1153
- [17] Ou Y. *et al* 2008 *Plasma Phys. Control. Fusion* **50** 115001
- [18] Lutjens H. *et al* 1996 *Comput. Phys. Commun.* **97** 219
- [19] Sauter O. *et al* 1999 *Phys. Plasmas* **6** 2834
- [20] Sauter O. *et al* 2002 *Phys. Plasmas* **9** 5140
- [21] Prater R. 2004 *Phys. Plasmas* **11** 2349
- [22] Wesson J. 2004 *Tokamaks (International Series of Monographs on Physics vol 118)* 3rd edn (New York: Oxford Science) ISBN 0-19-8509227
- [23] Barry S. 1999 The extension of the FIR interferometer of TCV to a polarimeter and measurements of the Faraday rotation caused by the poloidal magnetic field *PhD Thesis* National University of Ireland, Cork
- [24] Sushkov A. *et al* 2008 *Rev. Sci. Instrum.* **79** 023506
- [25] Lister J. *et al* 1997 *Fusion Technol.* **32** 321
- [26] Hofmann F. *et al* 1988 *Nucl. Fusion* **28** 1871
- [27] Rebut P. *et al* 1989 *Proc. 12th Int. Conf. on Plasma Physics and Controlled Nuclear Fusion Research 1988 (Nice, 1988)* vol 2 (Vienna: IAEA) p 191
- [28] Coda S. *et al* 2000 *Plasma Phys. Control. Fusion* **42** B311
- [29] Sontag E.D. 1998 *Mathematical Control Theory (Texts in Applied Mathematics vol 6)* 2nd edn (New York: Springer)
- [30] Smyshlyaev A. *et al* 2010 *Adaptive Control of Parabolic PDEs* (Princeton, NJ: Princeton University Press)
- [31] Anderson B. *et al* 1979 *Optimal Filtering* (Mineola, NY: Dover)
- [32] Xu C. *et al* 2010 *IEEE Trans. Plasma Sci.* **38** 359
- [33] Ferron J. *et al* 1998 *Nucl. Fusion* **38** 1055
- [34] Blum J. *et al* 2008 *J. Phys.: Conf. Ser.* **135** 012019
- [35] Felton R. *et al* 2005 *Fusion Eng. Des.* **74** 561
- [36] Barana O. *et al* 2007 *Fusion Eng. Des.* **82** 1023
- [37] Albajar F. *et al* 2005 *Nucl. Fusion* **45** 642
- [38] Hughes T.J.R. 1987 *The Finite Element Method* (Englewood Cliffs, NJ: Prentice-Hall)
- [39] de Boor C. 2001 *A Practical Guide to Splines (Applied Mathematical Science vol 27)* (New York: Springer)
- [40] Golub G.H. *et al* 1996 *Matrix Computations* (London: The Johns Hopkins University Press) ISBN: 0-8018-5413-X
- [41] Press W.H. *et al* 1996 *Numerical Recipes* 2nd edn (New York: Cambridge University Press)
- [42] Haykin S. 1999 *Neural Networks: A Comprehensive Foundation* 2nd edn (Upper Saddle River, NJ: Prentice-Hall)