

The Basic Components of Software-Hardware System for Modeling and Control of the Toroidal Plasma by Epsilon-Nets on Heterogeneous Mini-Supercomputers

Fedor S. Zaitsev^{1,2}, Alexei G. Shishkin^{1,2,*}, Andrei A. Lukianitsa^{1,2}, Egor P. Suchkov^{1,2}, Sergei V. Stepanov^{1,2} and Fedor A. Anikeev^{1,2}

¹ *Scientific Research Institute of System Development, Russian Academy of Sciences, Nakhimovsky prosp., 26 b.1, 117218, Moscow, Russia.*

² *Moscow State University, Department of Computational Mathematics and Cybernetics, Lenin's Hills, 119991, Moscow, Russia.*

Communicated by Jingfang Huang

Received 31 March 2017; Accepted (in revised version) 21 July 2017

Abstract. A significant step towards the comprehensive solution of the important new fundamental control problems of high-temperature plasma of toroidal configuration is made by utilizing heterogeneous mini-supercomputers of NIISI RAS (Scientific Research Institute of System Development, Russian Academy of Science). The basic components of hardware and software modeling system and automatic real-time plasma feedback control are developed and implemented. The system is named HASP CS (Hardware-Software Plasma Control System). It is based on a fundamentally new, more accurate and reliable control algorithm using a solution with a given accuracy of a number of inverse diagnostic problems by ϵ -nets. The possibility of an application of the proposed technique during discharge time to provide feedback control of the boundary and internal plasma parameters is demonstrated. It is shown that the complex allows investigating the processes in currently operated and designed fusion devices and solving problems of fusion application for energy production. The recommendations to the engineering implementation of plasma control systems are made. The directions for further research are formulated.

AMS subject classifications: 93C10, 93C95, 65N21

PACS: 52.55.Fa, 02.30.Yy, 02.30.Zz

Key words: Control problem, ϵ -net, plasma, tokamak, modeling.

*Corresponding author. *Email addresses:* zaitsev@cs.msu.ru (F. S. Zaitsev), shishkin@cs.msu.ru (A. G. Shishkin), andrei_luk@mail.ru (A. A. Lukianitsa), suchkov.egor@gmail.com (E. P. Suchkov), sergey.v.stepanov@gmail.com (S. V. Stepanov), snowfed@gmail.com (F. A. Anikeev)

1 Introduction

The leading world countries carry out intensive research on the transition to fusion power generation based on a secure, virtually inexhaustible source of energy.

The international thermonuclear tokamak reactor ITER (www.iter.org) is built by China, European Union, India, Japan, Korea, Russia and USA. The world's largest tokamak JET (www.euro-fusion.org), where in 1991 the first reaction of controlled thermonuclear fusion was carried out operates in EU. In the UK record plasma parameters were obtained on a large spherical tokamak MAST (<http://www.ccf.ac.uk>). In Germany ASDEX tokamak (<http://www.ipp.mpg.de/16195/asdex>) which has a well-developed plasma diagnostics operates. T-15 tokamak (National Research Center "Kurchatov Institute", <http://www.nrcki.ru/pages/main/6015/8698/7145/index.shtml>) is being built in Russia. The international community plans to construct the first fusion power plant DEMO in Japan.

Countries that possess the technology of controlled thermonuclear fusion (CTF) will be out of competition. The main difficulty in CTF study is a large range of temporal and spatial scales of the existing phenomena, as well as multi-dimensional dependency of process' characteristics on phase variables. It requires detailed study of plasma diagnostics and control systems, of engineering solutions in terms of accuracy, reliability, efficiency, safety and durability. Complete solution of these problems is impossible without the use of high-performance computing.

Plasma dynamics control is one of the main fundamental problems of theoretical and experimental study of fusion and of transition to the thermonuclear fusion power generation. However, control methods, especially for internal plasma parameters are poorly developed due to the need of using comprehensive mathematical models that address a number of complex ill-posed inverse problems of plasma diagnostics, development of massive knowledge-intensive software and application of high-performance computing. In practice, when constructing the controllers the semi-empirical greatly simplified approaches or prescribed (program) mode are commonly used, which lead to the bulk expensive work on the experimental choice of the parameters of control systems and to a large number of premature discharge disruptions during the research companies.

The basic components of system HASP CS (Hardware-Software Plasma Control System) were developed for comprehensive modelling of toroidal plasma control problems and support of real plasma operation. Here details of HASP CS are fully described for the first time in an easy accessible journal in English language.

The main objectives of this paper are: presentation of the concept and the main components of the new hardware and software system HASP CS, which allows modeling and control of the toroidal plasma by ϵ -nets on heterogeneous mini-supercomputers; description of the new boundary and internal plasma parameters real-time feedback control algorithms; a brief summary of the advanced graphical user interfaces; an example of system prototype operation.

Ongoing studies are focused on the creation of fundamentals of advanced technolo-

gies for the implementation of the essential components of a fusion power plant plasma control system. Deployment of these results in existing devices and future tokamak reactors will significantly increase the efficiency of expensive experimental studies, reduce their cost and the transition time to the thermonuclear energy.

The basic components of hardware and software system are developed in NIISI RAS (Scientific Research Institute of System Development, Russian Academy of Sciences) under the programs of the fundamental research of Presidium of RAS Nos. 43 and 17.

An important component of the system is the created in NIISI RAS jointly with other organizations the ϵ -nets method for solving with a given accuracy inverse problems of plasma diagnostics. This method makes it possible to meet the new challenges of the plasma in a tokamak modeling and precise real-time feedback control of discharge parameters. At present, no other research centers yet mastered the ϵ -nets technique and thus cannot solve full-scale problems of discharge control.

Calculations by ϵ -nets method implemented by the classical methods of parallel programming require the use of supercomputers of unprecedented performance, for example, "Lomonosov" of Moscow State University. However, developed in NIISI RAS new technology of heterogeneous CPU/GPGPU computing based on OpenCL standard [2, 3] allows using a mini-supercomputer at the expenditure of the complication of the parallel algorithm. Algorithm development costs are paid out by low cost and compactness of computational equipment. In addition, the low operating costs for the maintenance of mini-supercomputers allow allocating it entirely for the solution of the real time control problems. As well, the negative impact of time-sharing mode that inevitably arises during the operation of large supercomputer is excluded.

The possibility of using the proposed technique during discharge to provide feedback control of boundary and internal plasma parameters is demonstrated in the paper. It is shown that the developed system can be used in the study of processes in currently operated and newly designed fusion devices of all sizes and designs and for the solution of problems of thermonuclear fusion application for energy production.

2 The main components of the modeling and plasma control system

Software-hardware system is implemented on NIISI RAS supercomputers. User interacts with application software by using a set of graphical interfaces. A plasma model of different detailing level or a real plasma in a tokamak serve as a control object. The general scheme of the system is shown in Fig. 1. The arrows indicate the direction of information flows.

The system incorporates the basic tokamak components, its main plasma models and control systems such as the structural components (vacuum chamber, coils of toroidal and poloidal magnetic fields, etc.), various plasma diagnostics, magnetic and kinetic plasma control systems, actuating models, auxiliary plasma heating and current gen-

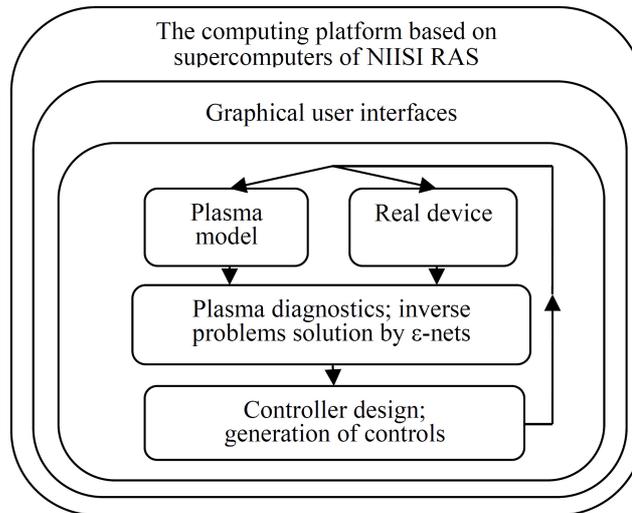


Figure 1: The main components of the system.

eration systems, divertors, etc. The system includes codes that simulate the processes occurring in a tokamak at different discharge stages; much attention was therefore given to the time synchronization of the sequence of code execution. The resulting computational load is automatically distributed across the network computers.

A plasma model includes the following modules:

- Magnetohydrodynamic (MHD) equilibrium and its evolution: a slow change of plasma properties when the MHD equilibrium is retained at each time moment-SCoPE code [1].
- Transport of charged particles: evolution of heat and particle fluxes in the radial direction taking into account plasma heating, emission and absorption of energy-FPP-3D code [1].

A diagnostic processing module performs a boundary and internal plasma parameters reconstruction from the measurement data. It is based on the real-time solutions of inverse problems with a given inaccuracy by ϵ -nets method (FPP-3D, SCoPE, VIP, SDSS, NNTMM, CLUNAVT, DAMA and FIRE codes [1]).

Information about the plasma state is transferred to the module of control calculation that if necessary refines controller in real-time, generates control signals and transmits them to the plasma model or to actuators of real device (SCoPE code [1]).

The system is able to switch quickly between different models of plasma of various levels of complexity, as well as between the control of plasma model and of plasma itself in a real device.

3 Model of tokamak operation

Software-hardware system allows working with different models of tokamak operation. The model implemented in code SCoPE (Self Consistent Plasma Evolution) is used as a primary one [1]. Physical and mathematical components of the model are described in detail in the book [1]. The formulation of the model is cumbersome, so here only its general characteristics are given.

We consider time scales much longer than the development of MHD (Magneto Hydro Dynamic) instabilities time, i.e., from about hundredths or thousandths of a discharge time. SCoPE code model is well developed and verified [1] and allows to take self-consistently into account dozens of plasma behavior features and to describe discharges in the devices of various sizes and designs: ITER, JET, MAST, T-15.

The history of the development and application of SCoPE code dates back to more than 20 years. Amongst other features SCoPE code was developed to eliminate a number of existed in other codes drawbacks. In particular, SCoPE carefully takes into account the following items: effects of electromagnetic induction especially important for the solution of plasma control problems; free boundary plasma; arbitrary aspect ratio of the torus, i.e., the code can be applied to both conventional and spherical tokamaks; Spitzer and the neoclassical model of plasma conductivity; the presence of plasma currents driven by the pressure gradient; the ability to describe generation of a highly localized current in a plasma by RF waves and the injected neutral particles; the presence of the conductive vacuum chamber walls and of passive elements; description of plasma density and temperature by the transport equations or by the import of experimental measurements; various approaches to plasma control.

Furthermore, SCoPE code has interfaces for data input from other codes such as, for example, the kinetic code FPP-3D [1], the kinetic Monte Carlo code LOCUST [17], the code of equilibrium reconstruction EFIT [18], the evolution diagnostic code TRANSP [19].

In typical runs SCoPE whose flowchart is given in Fig. 2 uses the following input information.

- Device geometry, i.e., the location of chamber walls and of passive elements.
- Current in toroidal field coils $I_{rod}(t)$ as a function of time t .
- Poloidal field coil coordinates and currents in these coils. In the free boundary problem, these currents are redefined in SCoPE to achieve the desired shape and position of the plasma. Currents in poloidal field coils are used for calculation of the induced current in the plasma as well.
- The position of the solenoid coils and the currents therein. In the control problem the solenoid current is adjusted to bring plasma total current close to the given one. The currents in the windings of the solenoid are taken into account when calculating the induced current in the plasma.

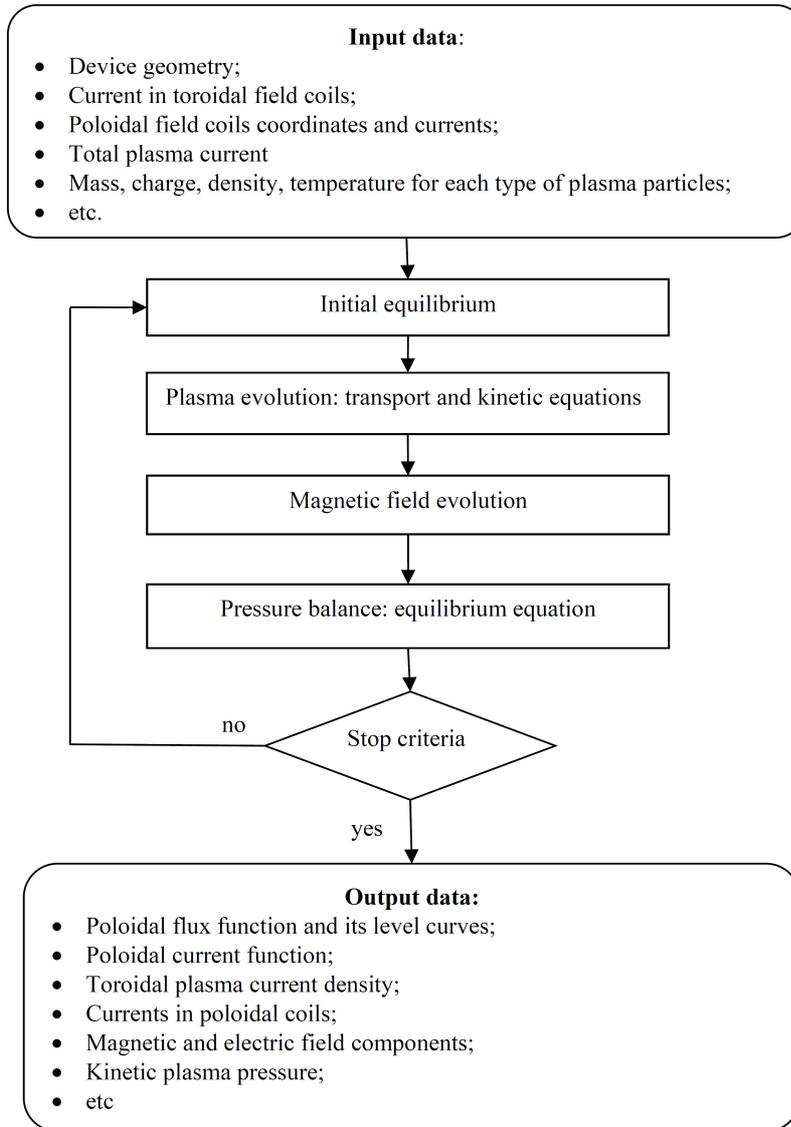


Figure 2: Flowchart of SCoPE code.

- The coordinates of points on the plasma boundary as functions of time. In a fixed boundary problem these points confine the plasma. In a free boundary problem the configuration closest to the specified boundary points is searched.
- Total plasma current $I_p(t)$. When the problem of total plasma current control is solved the difference between the calculated and given $I_p^0(t)$ current is minimized.
- Initial density of toroidal plasma current $j_\eta(t=0, R, Z)$.

- Mass m_α , charge e_α , density $n_\alpha(t,z)$ and temperature $T_\alpha(t,z)$ for each type α of plasma particles, where $z = \gamma/\gamma_\alpha$, γ_α -plasma boundary label. Kinetic pressure is calculated as $p(t,z) = \sum_\alpha n_\alpha(t,z)T_\alpha(t,z)$ or specified in input data, for example, from experimental measurements. As an alternative, SCoPE code can solve transport equation to define $n_\alpha(t,z)$ and $T_\alpha(t,z)$.
- NBI-current (Neutral Beam Injection) and additional density of fast ions driven by injection.

SCoPE generates a lot of important data on plasma, in particular, the following output data:

- Poloidal flux function $\Psi(t,R,Z)$ and its level curves (magnetic surfaces) $\Psi = const$ as functions of (t,R,Z) , where (R,Z) are cylindrical coordinates.
- Poloidal current function $F(t,\Psi)$ related to toroidal magnetic field $B_{tor} = F/R$.
- Toroidal plasma current density $j_\eta(t,R,Z)$.
- Safety factor $q(t,\Psi) = -\partial\Phi/\partial\Psi/(2\pi)$ where Φ is the function of flux of toroidal magnetic field.
- Magnetic axis coordinates in various time moments. Magnetic field value on magnetic axis.
- Currents in the coils of poloidal magnetic field as functions of time.
- Magnetic $\mathbf{B}(t,R,Z)$ and electric $\mathbf{E}(t,R,Z)$ field components.
- Kinetic plasma pressure $p(t,\Psi)$.

A number of advanced numerical methods [1] is implemented in SCoPE. They ensure the calculation of plasma evolution with the required accuracy.

The approaches of object-oriented programming are used. The modern version of the code is made up of about 150 modules of more than 60 thousand lines in Fortran 2008.

Historically SCoPE was created as a sequential program. The time of a computational experiment, especially for discharges with complex dynamics and large plasma cross-section, can amount to tens of hours. For example, the calculation of plasma evolution in ITER takes from 10 to 20 hours on a single 4-core processor Intel Core i7. Therefore, the parallelization is necessary for large-scale computational modeling of plasma control.

SCoPE code is optimized for efficient use in heterogeneous computing systems of NIISI RAS. MPI technology (Message Passing Interface) in the latest version of MPICH2 package is used in SCoPE.

In addition to MPI technology the tuning of parameters of the implemented in SCoPE algorithms was made to reduce the run time without a loss of accuracy. As a result, the

typical run time of the equilibrium evolution calculation was reduced by almost five times at a processor Intel Core i7 with 4 cores.

Examples of hardware and software system with a use of SCoPE code as a plasma model are shown below.

4 General description of ϵ -nets method

In many applications, there arises the necessity of solving ill-posed problems. Such problems play a particularly important role in the problem of controlled thermonuclear fusion (CTF) because high temperatures prevent measurements directly inside plasma.

The input data of an ill-posed problem are usually given with some error. Moreover, a small variation of the input data of an ill-posed problem may cause a large change of its output data [4]. Methods for solving ill-posed problems usually seek one solution and do not answer the question of whether there are other solutions substantially different from the found one and corresponding to the same input data within the limits of a given error. This strongly complicates obtaining reliable results.

Relatively recently the method of ϵ -nets was developed [1,5] for numerically solving a certain class of ill-posed problems making it possible to find all strongly different solutions or substantiate their absence with given inaccuracy ϵ . In the general case, this method is suitable for numerically analyzing the ill-posedness of a problem with error ϵ , namely, studying the existence and uniqueness of solutions and stability with respect to the input data. In the limit $\epsilon \rightarrow 0$, the functions which form an ϵ -net give the entire initial class functions which makes it possible to improve the accuracy of analysis in the sense of approximation of the initial class of functions by decreasing ϵ .

At present, the method of ϵ -nets is the only method which makes it possible to solve a whole series of the most important problems of plasma diagnostics in controlled fusion, namely, choose a solution of the problem of plasma equilibrium reconstruction with approximate input data which is closest to a real physical process, compute the error (confidence intervals) of reconstructions, investigate the effectiveness of various diagnostics (various additional conditions) in the determination of a solution adequate to reality and estimate the required accuracy of measurements.

The real-time feedback plasma control systems are based on the solution of diagnostics problems. Most important among them are the problems of boundary and internal plasma parameters control. The method of ϵ -nets gives a unique way to solving such control problems.

In the simplest cases, the ϵ -nets method makes it possible to obtain instructive results from measurement processing on a usual personal computer without any accelerators. However, improving the accuracy and speed of computations requires applying high performance computers.

The parallel algorithms implementing the ϵ -nets method for two types of computer architecture, with central processors (CPU) and graphic accelerators (GPGPU) have been

developed in NIISI RAS and their properties have been studied [1].

CPU-architectures usually employ the MPI and MPI/OpenMP technologies of parallel computation. Such architectures are used in most of the modern supercomputers, including "Lomonosov" and "Blue Gene/P" at Moscow State University, which have $\sim 10^4$ CPU-cores and in exaflops computing complexes with $\sim 10^6$ cores being designed.

Hybrid architectures use special-purpose graphic or other accelerators operated by the OpenCL, CUDA, OpenACC and other technologies. Such architectures make it possible to achieve high performance on a relatively cheap and compact hardware at the expense of algorithm complication.

We consider the following class of problems. In a domain Ω consider the equation

$$L(\mathbf{x}, u) = b(\mathbf{x}, f(u)), \quad (4.1)$$

where $\mathbf{x} \in \Omega$, L is a given operator, b is a given function and $u = u(\mathbf{x})$ and $f = f(u)$ are unknown functions. It is known that $u \in [0, 1]$ and $f(u)$ is contained in a given strip $g^-(u) \leq f(u) \leq g^+(u)$, $d^-(u) \leq f'(u) \leq d^+(u)$. The additional conditions to Eq. (4.1)

$$l_1(\mathbf{x}, u) = b_1(\mathbf{x}, u), \quad \mathbf{x} \in \Gamma, \quad (4.2a)$$

$$\frac{\|l_2 \mathbf{x}, u - b_2(\mathbf{x}, u)\|}{\|b_2(\mathbf{x}, u)\|} \leq \delta \quad (4.2b)$$

are added. Here, Γ is the boundary of the domain Ω ; l_1 , l_2 and b_1 , b_2 are known operators and functions; and δ is a given relative error. As a rule, of most interest is the uniform norm which is defined as the least upper bound over $\Omega \cup \Gamma$. We assume that condition (4.2a) ensures the possibility of solving Eq. (4.1) in the case where the function $f(u)$ is known. We refer to problem (4.1), (4.2a) with given $f(u)$ as the direct problem. Condition (4.2b) is added for solving problem (4.1)-(4.2b) with unknown function $f(u)$. Such a problem is called the inverse problem.

It is required to find all strongly different solutions $(u, f(u))$ of the inverse problem (4.1)-(4.2b). We do not specify the term "strongly (substantially) different solutions". A criterion for selecting substantially different pairs $(u, f(u))$ depends on the objective of a particular study. In many cases, a noticeable difference in the quantitative or the qualitative behavior of the functions $f(u)$ and/or u in some set of points \mathbf{x} is considered.

The algorithms presented below are easy to modify for solving a more general problem in which the right-hand side of (4.1) contains a set of one-dimensional functions $b(\mathbf{x}, f_1(u), f_2(u), \dots, f_{n_f}(u))$ rather than only one function and the other operators and functions in (4.1)-(4.2b) are multidimensional. The setting of the problem may contain several conditions of the form (4.2a) or (4.2b). The method can also be generalized to the case of a vector function u .

We assume that a method for solving the direct problem (4.1), (4.2a) is known. The main idea of the solution of the inverse problem (4.1)-(4.2b) is to construct a finite set of functions covering some class of functions $f(u)$ with a given absolute error ϵ (i.e., an ϵ -net), solve the direct problem (4.1), (4.2a) for each element of this set, verify the fulfillment of condition (4.2b) and select strongly different solutions according to a certain criterion.

Sequential algorithms for constructing ϵ -nets were presented in [1, 5]. ϵ -nets of polynomials, splines and their combinations were considered. In [1, Section 2.4], detailed proofs of assertions about the number of elements and values of other parameters of ϵ -net for which a given error ϵ is achieved were presented and strategies of application of ϵ -nets were described. Settings and solutions of particular inverse and control problems are given in [1, Sections 1.7, 1.8, 2.3, 2.5 and 4.9].

The technique of ϵ -nets is an enumeration algorithm for finding function best satisfying given conditions. The method is based on solving a separate problem for each element of an ϵ -net. A condition for obtaining a highly accurate solution requires an ϵ -net of a large size and, therefore, a large amount of computations. However, the full independence of problems for different elements of the ϵ -net makes it possible to completely parallelize the algorithm [1].

The construction of an ϵ -net itself can be parallelized as well. But this problem is not so interesting because its solution requires significantly less computations. Moreover, some set of ϵ -nets can be once constructed and used many times for solving problems from a certain class with different input data.

The parallelization of the method of ϵ -nets for CPU and hybrid GPGPU architectures has some specific features. The memory and performance of one processor (or core) in CPU/MPI systems (already functioning or being designed) are sufficient for solving the problem for one or several ϵ -net elements without interprocessor exchanges which simplifies the algorithm and improves its efficiency. In GPU architectures the local memory of a thread is usually small and exchanges between different types of memory or special computing algorithms are needed. However, the number of threads may be larger by two or three orders than the number of processors in CPU systems which ensures a fast acceleration. In addition, the expenditures for designing a more complex algorithm are compensated by the significantly lower cost and the compactness of GPGPU systems.

The idea of the parallelization algorithm for the ϵ -net method on CPU/MPI computers is relatively simple. It is necessary to distribute the problems for all elements of the ϵ -net over processors, solve the problem for each element and select a solution best satisfying the given conditions. The processing time of one ϵ -net element may be different for different elements; therefore the acceleration and efficiency of the hardware will improve, if the computations are distributed uniformly with processor loads taken into account.

A parallel method for OpenCL, CUDA and OpenACC program models is constructed in a similar way. It is supplemented by an exchange algorithm for different types of memory and a program solving sparse systems of linear algebraic equations adapted to a particular accelerator.

The parallel algorithms of the ϵ -nets method described above were implemented for both CPU/MPI and CPU/GPGPU technologies in the SDSS code (Substantially Different Solution Searcher) [1,5]. The total length of the SDSS code is more than 30000 lines in Fortran 2008. Fortran makes it possible to use, in addition to MPI, the built-in SIMD parallel technology for accelerating computations as well as its advantageous tools for processing numerical data structures and flexible tools for specifying the precision of computer

representation of numbers. In the GPU version of the SDSS C++ language was used for handling the memory of accelerators and calling numerical algorithms specialized for accelerators.

SDSS code based on ϵ -nets technique allows solving a variety of inverse problems of plasma parameters reconstruction.

5 Plasma boundary reconstruction by ϵ -net method

Plasma control methods are based on solving the inverse problems of plasma parameters reconstruction according to the measurement data. The problem of a plasma reconstruction can be divided into two subtasks [1, 5]: plasma boundary reconstruction and reconstruction of internal plasma parameters. The former one is considered below; the latter one is described in the next section. The construction of solutions for both problems is based on applying a corresponding modification of the ϵ -net method.

Mathematically the problem of plasma boundary reconstruction is formulated as follows. Consider the Grad-Shafranov operator [1] in cylindrical coordinates

$$\Delta^* \psi \equiv R \frac{\partial}{\partial R} \left(\frac{1}{R} \frac{\partial \psi}{\partial R} \right) + \frac{\partial^2 \psi}{\partial Z^2}.$$

Outside plasma the following equation holds true

$$\Delta^* \psi = -\mu_0 R \sum_{i=1}^L J_i(t) \delta(R - R_{c,i}) \delta(Z - Z_{c,i}), \quad (5.1)$$

where $(R_{c,i}, Z_{c,i})$ are the coordinates of the poloidal field coils, $J_i(t)$ are the corresponding system.

The values of the poloidal current flux function ψ_k are known at the measurement points (R_k, Z_k) located outside the plasma, $k = 1, \dots, K_e$.

Plasma boundary Γ_p , which is defined as a closed contour line of function ψ of maximum width, must be found. In other words, plasma boundary is a such closed contour line that the segment of R axis inside of it has the maximum width. We must add extra conditions to Eq. (5.1):

$$\psi|_{\Gamma_p} = \psi_b, \quad \psi|_{\Gamma_w} = \psi_w, \quad \Delta^* \psi = -\mu_0 R j_{\text{SOL}},$$

where ψ_b is the value of ψ at the plasma boundary, ψ_w is the value of ψ near the chamber walls and passive elements Γ_w ; j_{SOL} is the current density in the region bordering the plasma. Values ψ_b and ψ_w are determined in the course of solving the problem, j_{SOL} is set according to a separate model.

Other constraints could be added such as D-alpha diagnostics data, plasma video footage, the presence of plasma limiters, prior information on plasma boundary shape

near X-point, etc. The ϵ -net method allows one to fully account many additional conditions relatively easily.

The problem statement has new features. It uses the information about the values of the poloidal flux near the chamber walls and passive elements besides the information about the currents in the control coils and solenoid and the region bordering the plasma.

The proposed here numerical method for solving the problem is based upon reducing it to a system of nonlinear algebraic equations, linearization and solving a set of linear algebraic systems.

Let us search for a polynomial function

$$\psi = \sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho^i z^j, \quad \rho = R^2 - R_0^2, \quad z = Z - Z_0,$$

and for plasma boundary in a parametric form

$$\begin{cases} R = R(\xi), \\ Z = Z(\xi), \end{cases} \quad \xi \in [-\pi, \pi].$$

Besides points $P_{e,k}$, $k = 1, \dots, K_e$, consider: K_v points $P_{v,k}$, $k = K_e + 1, \dots, K_e + K_v$ in vacuum; K_b points $P_{b,k} = (R(\xi_k), Z(\xi_k))$, $k = K_e + K_v + 1, \dots, K_e + K_v + K_b$ at the plasma boundary; K_w points $P_{w,k}$ near the chamber walls and passive elements. We get the following system of equations:

$$\sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho_k^i z_k^j = \psi_k, \quad k = 1, \dots, K_e; \quad (5.2a)$$

$$\Delta^* \left(\sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho_k^i z_k^j \right) = 0, \quad k = K_e + 1, \dots, K_e + K_v; \quad (5.2b)$$

$$\begin{cases} \sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho_k^i z_k^j = \psi_b, \\ R = R(\xi_k), \\ Z = Z(\xi_k), \end{cases} \quad k = K_e + K_v + 1, \dots, K_e + K_v + K_b; \quad (5.2c)$$

$$\Delta^* \left(\sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho_k^i z_k^j \right) = -\mu_0 R_k J_k(t), \quad k = K_e + K_v + K_b + 1, \dots, K_e + K_v + K_b + L; \quad (5.2d)$$

$$\sum_{i=0}^N \sum_{j=0}^M a_{ij} \rho_k^i z_k^j = \psi_w, \quad k = K_e + K_v + K_b + L + 1, \dots, K_e + K_v + K_b + L + K_w, \quad (5.2e)$$

where (ρ_k, z_k) corresponds to $P_{e,k}$, $P_{v,k}$, $P_{b,k}$, $P_{c,k}$, $P_{w,k}$ points.

Consider a certain class of curves $(R(\xi), Z(\xi))$, i.e., consider the class of plasma boundaries. For each curve of this class equations (5.2a)-(5.2e) allow finding function

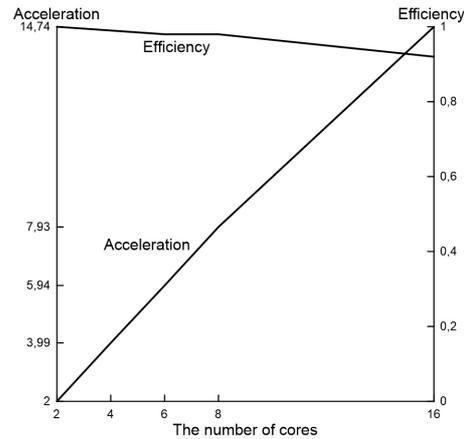


Figure 3: Speedup and efficiency on one CPU as function of the number of cores used in this CPU.

$\psi(R, Z)$ outside plasma by solving system of linear algebraic equations. Thus, the problem is linearized in this sense. The construction of the boundary class $R(\xi)$, $Z(\xi)$ is performed via ϵ -nets.

The system of linear equations for unknown constants a_{ij} , ψ_b , ψ_w has a rectangular matrix and can be solved, for instance, with Tikhonov regularization. The regularization parameter is chosen so that it satisfies the condition of the minimum residual of system (5.2a)-(5.2e). In order to increase or decrease the role of one set of equations in respect to another one can minimize a functional composed of residual sums of weighted sets of equations. Additionally, it is possible to achieve a more precise description of a particular part of the boundary, e.g., near X -point, by setting greater weight for a corresponding part of the residual for (5.2c). Going through all of the plasma boundaries of the specified class we obtain different values of the given functional. The required boundary ($R(\xi), Z(\xi)$) is the boundary minimizing the functional.

The algorithm for finding the plasma boundary with ϵ -net method was parallelized for the heterogeneous architecture of NIISI RAS mini-supercomputers with Intel Xeon processors. Each process of the algorithm requires ~ 100 Mb RAM. Hard disk space requirement is ~ 500 Mb. The computational performance is ~ 180000 elements of the ϵ -net per second which is more than enough for using the proposed method in real-time for a system with feedback control of a plasma boundary. By comparison, the performance on a regular PC is ~ 1000 elements of the ϵ -net per second. The speedup and efficiency of the MPI parallel algorithm are shown in Figs. 3 and 4.

6 Reconstruction of internal plasma parameters with the ϵ -net method

As previously noted the problem of plasma parameters reconstruction can be divided into two subtasks: plasma boundary reconstruction and the reconstruction of internal

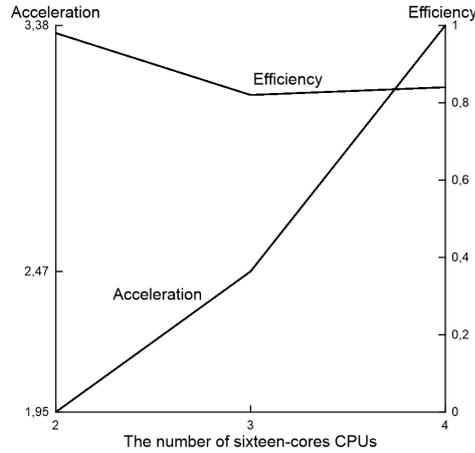


Figure 4: Speedup and efficiency on several CPUs with sixteen cores used in each CPU.

plasma parameters. The current section is devoted to the latter one.

The mathematical statement of the problem of the reconstruction of internal plasma parameters is rather cumbersome, for this reason it is not presented here. The statement of the problem and the methods for its solution are described in full detail in Sections 1.8.1, 2.5, 2.5.7, 4.10 of book [1] and in [5]. Here only a general approach to solving the inverse problem of the reconstruction of internal plasma parameters with the ϵ -net method is presented.

The flowchart of algorithm is shown in Fig. 5. It consists of the following steps:

1. The plasma boundary reconstruction via solving the exterior problem (see the previous section). Plasma boundary Γ_p and poloidal flux function ψ found at this stage are used at steps 2-6 as the input when solving the internal problem.
2. The reconstruction of kinetic plasma pressure p and poloidal current function F (related to toroidal magnetic field $B_{tor} = F/R$) by one of the methods described in [1, Section 2.5]. At this stage there is no demand for the high-precision reconstruction of functions p and F because these functions are only used at step 3 to set the boundaries of the area in which the ϵ -net is to be constructed. So, one can utilize the solution of a direct problem with a particular approximation of functions p and F instead of solving the inverse problem.
3. Setting the domains around functions p and F for which other functions satisfying the error of input data (measurements) are to be sought. The construction of an ϵ -net for functions p and an ϵ -net for functions F with SDSS code.
4. The solution of a standard direct problem with a prescribed plasma boundary for the Grad-Shafranov equilibrium equation for each set of elements p and F of the corresponding ϵ -nets. Finding the set of triples (ψ_k, p_k, P_k) .

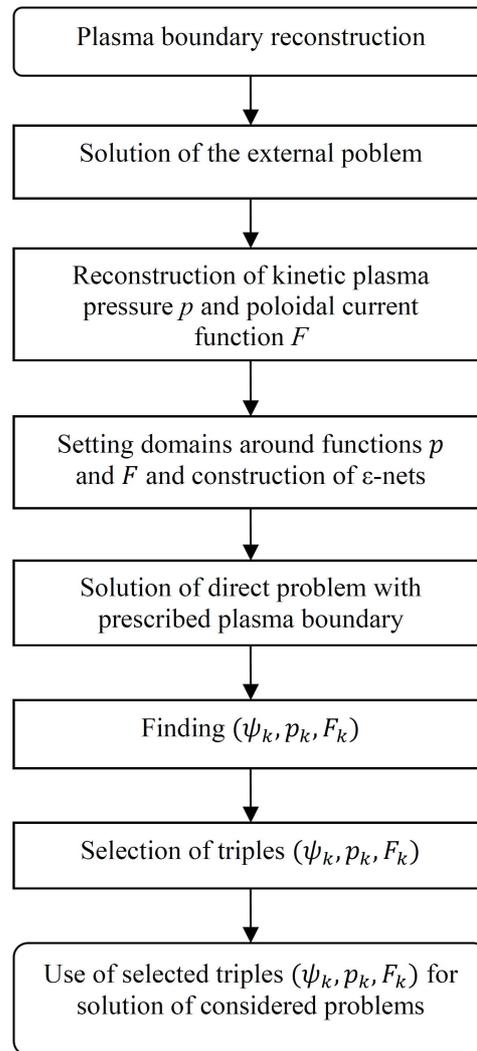


Figure 5: Flowchart of algorithm for reconstruction of internal plasma parameters.

5. The selection of triples (ψ_k, p_k, F_k) satisfying basic subsidiary conditions and one or another combination of special subsidiary conditions [1].
6. The use of selected triples (ψ_k, p_k, F_k) for the solution of the most important problems:
 - The calculation of an error range (confidence interval) for the reconstructions of internal plasma parameters.
 - The analysis of existence of significantly different solutions of the inverse problem which satisfy one or another measurements error (input data error).

- The examination of the efficiency of additional diagnostics (additional conditions) in finding the inverse problem solution corresponding to the actual physical process.
- The calculation of the maximum allowed measurements error of the diagnostic equipment.
- The determination of the optimal number and positioning of the measurement sensors.
- The studying of the influence of the plasma model modifications on the confidence interval size. Including: (a) anisotropic plasma pressure, (b) taking into account the measurements of the loop voltage, (c) the incorporation of Ohm's law for the current density evolution in plasma into the model, (d) the presence of the iron core, (e) the presence of passive elements, (f) the axial asymmetry of plasma equilibrium.

The reconstruction of p and F with minimum error is of the main interest in the problem of the internal plasma parameters control. Knowing the difference between the reconstructed (p, F) and the desired values one can consider the problem of automatic control of (p, F) profiles (using the influence of the High Frequency field and/or injection) as well as the problems of current density profiles and the safety factor.

The devised method is perhaps the first ever method of practical use which allows for feedback control of internal plasma parameters in real-time based on solving the inverse problem for the equilibrium equation with prescribed precision. That is made possible by the parallel algorithms of the ϵ -net method [1].

It was shown that with ~ 1000 measurements per minute the problem of plasma equilibrium reconstruction can be solved the same number of times with $\sim 8\%$ precision in the class of quadratic polynomials or quadratic splines. Therefore the methods proposed can be used right away in real-time systems for the automatic feedback control of plasma boundary and position. The employment of the next generation computers with higher performance will increase the precision of the real-time problem solving and will broaden the available classes of functions.

7 New algorithm for plasma control

Plasma control problems in tokamaks are complicated by a number of factors that should be taken into account while building controllers:

- The behavior of the plasma is nonlinear.
- Experimental measurements, numerical models and methods contain inaccuracies.
- There exists quite a strong delay. For example, the current in the plasma is not changed instantly under the influence of controls and some time is required for

penetration of the electromotive force into the plasma. In terms of control theory such an object is called an object with a delay.

- The conductivity of the plasma at the periphery is much lower than in the plasma center. Therefore, to create a current in the plasma one must substantially change the current in the solenoid. However, excessive electromotive force (electric voltage) while penetrating into the central plasma regions can lead to excessively high current increase due to the higher conductivity of the plasma in the center.
- Time-varying magnetic field induces currents in the walls of the vacuum chamber and in the passive elements. These currents can exert a strong influence on the behavior of the plasma.
- Due to the engineering constraints the current increment in the solenoid coils and in the poloidal magnetic field coils should not be too large.
- Plasma can drastically change its state, for example, due to the development of instabilities.

Unlike traditional approaches to CTF, in addition to the use of ϵ -nets, the algorithm allows building the controller dynamically in real time based on data about the plasma state in the previous time moments.

Let h_1, \dots, h_N be the differences between desired and available plasma parameters or rather some object approximating the plasma. For example, h_1, \dots, h_{N-2} are distances between the points through which a plasma boundary must go and the corresponding points of the obtained boundary, h_{N-1} is the difference between the required value of total toroidal current and actually available, h_N is the difference between the increments of the given current and actual current.

Based on a priori information that \mathbf{J} currents in poloidal coils have an effect mainly on the shape and position of the plasma and their time derivatives $\dot{\mathbf{J}}$ have an effect on the full toroidal plasma current I_p , we assume the following representation for the functional dependence of the error

$$\mathbf{h} = \mathbf{h}(t, I_p(t), \dot{\mathbf{J}}(t), \mathbf{J}(t)).$$

Derivatives of $\dot{\mathbf{J}}(t)$ enter only I_p as they mainly effect I_p and not the position of the plasma boundary. The explicit dependence of \mathbf{h} and I_p on t takes into account the possibility of changing these quantities in time due to the different from $\mathbf{J}(t)$ and $\dot{\mathbf{J}}(t)$ parameters of the object, for example, due to the temperature or plasma density.

Using Taylor expansion of \mathbf{h} and introducing the notation

$$\begin{aligned} \delta \mathbf{J}(t_n) &\equiv \dot{\mathbf{J}}(t_n) \tau = \mathbf{J}(t_n) - \mathbf{J}(t_{n-1}) + \mathcal{O}(\tau^2), \\ \delta \dot{\mathbf{J}}(t_n) &\equiv \frac{d\dot{\mathbf{J}}(t_n)}{dt} \tau = \dot{\mathbf{J}}(t_n) - \dot{\mathbf{J}}(t_{n-1}) + \mathcal{O}(\tau^2), \\ \tau &\equiv t_{n+1} - t_n, \end{aligned}$$

with an accuracy up to terms of second order $\mathcal{O}(\tau^2)$ we have

$$\begin{aligned} & \mathbf{h}[n+1] \\ = & \mathbf{h}[n] + \frac{\partial \mathbf{h}[n]}{\partial t} \tau + \frac{\partial \mathbf{h}[n]}{\partial I_p} \left(\frac{\partial I_p[n]}{\partial t} \tau + \frac{\partial I_p[n]}{\partial \mathbf{j}} \delta \mathbf{j}[n] \right) + \frac{\partial \mathbf{h}[n]}{\partial \mathbf{J}} \delta \mathbf{J}[n], \end{aligned}$$

where, for brevity sake, the time step number is indicated in square brackets.

Taking into account

$$\delta \mathbf{J}[n] \equiv \mathbf{J}[n] \tau = (\mathbf{J}[n] + \mathbf{J}[n-1] - \mathbf{J}[n-1]) \tau = (\mathbf{J}[n-1] + \delta \mathbf{J}[n]) \tau + \mathcal{O}(\tau^2),$$

we obtain for linearized error the following expression with an explicit value $\delta \mathbf{j}[n]$ which we choose as a control

$$\begin{aligned} & \mathbf{h}[n+1] \\ = & \mathbf{h}[n] + \frac{\partial \mathbf{h}[n]}{\partial \mathbf{J}} \tau \mathbf{J}[n-1] + \tau \left(\frac{\partial \mathbf{h}[n]}{\partial t} + \frac{\partial \mathbf{h}[n]}{\partial I_p} \frac{\partial I_p[n]}{\partial t} \right) \\ & + \left(\frac{\partial \mathbf{h}[n]}{\partial I_p} \frac{\partial I_p[n]}{\partial \mathbf{j}} + \frac{\partial \mathbf{h}[n]}{\partial \mathbf{J}} \tau \right) \delta \mathbf{j}[n]. \end{aligned} \quad (7.1)$$

We will describe dynamics of the controlled object by a state vector \mathbf{x} that satisfies the discrete equation

$$\mathbf{x}[n+1] = A\mathbf{x}[n] + B\mathbf{u}[n], \quad (7.2)$$

where \mathbf{u} is a control vector.

Taking into account a priori information about the factors having effect on the boundary and plasma current we will use

$$\mathbf{u}[n] = \delta \mathbf{j}[n]$$

as a control vector. The state vector \mathbf{x} and matrices A, B will be constructed using the linearized formula for \mathbf{h} .

Let

$$\mathbf{x}[n] = \begin{pmatrix} \mathbf{h}[n] \\ \tau \mathbf{J}[n-1] \\ \tau \left(\frac{\partial \mathbf{h}[n]}{\partial t} + \frac{\partial \mathbf{h}[n]}{\partial I_p} \frac{\partial I_p[n]}{\partial t} \right) \end{pmatrix}.$$

With this choice of vector \mathbf{x} the state of the object is characterized by a deviation of \mathbf{h} parameters from given ones, by increment of control currents $\tau \mathbf{j}[n-1] = \delta \mathbf{J}[n-1]$ at previous time moment and by increments associated with explicit dependencies of \mathbf{h} and I_p on time.

We will consider

$$A = 0.999 \begin{pmatrix} I & C_1 & I \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix},$$

$$C_{1,ik} = \left(\frac{\partial h_i[n]}{\partial J_k} \right), \quad i=1, \dots, N, \quad k=1, \dots, N_{coil},$$

as a matrix A . Here I is a unit matrix of a corresponding dimension, N_{coil} is a number of components of vector \mathbf{J} . In the absence of control $\delta \mathbf{J}[n]$ the top rows of the matrix A provide the fulfillment of Eq. (7.1), the middle and bottom row do not change the corresponding elements of the state vector. The coefficient of 0.999 provides stability of the object (7.2).

We will take

$$B = \begin{pmatrix} C_2 \\ \tau I \\ 0 \end{pmatrix}, \quad C_{2,ik} = \left(\frac{\partial h_i[n]}{\partial I_p} \frac{\partial I_p[n]}{\partial \mathbf{J}_k} \right) + \tau C_{1,ik},$$

$$i=1, \dots, N, \quad k=1, \dots, N_{coil},$$

as a matrix B .

The group of upper rows of the matrix B provide the fulfillment in (7.2) of Eq. (7.1) for \mathbf{h} . The group of middle rows in (7.2) yields the identity with the accuracy of $\mathcal{O}(\tau^3)$, since in the left-hand side we have $\tau \dot{\mathbf{J}}[n]$ and in the right-hand side we have

$$\tau \dot{\mathbf{J}}[n-1] + \tau \delta \mathbf{J}[n] = \tau \cdot (\dot{\mathbf{J}}[n-1] + \mathbf{J}[n] - \dot{\mathbf{J}}[n-1] + \mathcal{O}(\tau^2)) = \tau \mathbf{J}[n] + \mathcal{O}(\tau^3).$$

The group of bottom rows in (7.2) gives us the following expression

$$\tau \left(\frac{\partial \mathbf{h}[n+1]}{\partial t} + \frac{\partial \mathbf{h}[n+1]}{\partial I_p} \frac{\partial I_p[n+1]}{\partial t} \right) = \tau \left(\frac{\partial \mathbf{h}[n]}{\partial t} + \frac{\partial \mathbf{h}[n]}{\partial I_p} \frac{\partial I_p[n]}{\partial t} \right), \quad (7.3)$$

which means the invariance during transition to time layer $n+1$ of the increment of value in parentheses associated with the explicit dependence of \mathbf{h} and I_p on time.

Considered matrices A and B mark out the object properties that are weakly dependent on time, the vector \mathbf{x} gives emphasis on its dynamic condition and the vector \mathbf{u} singles out the object control. Process (7.2) describes the object evolution when the Eq. (7.1) is fulfilled and increments $\tau \dot{\mathbf{J}}[n]$ and (4.2b) are conserved. The object evolving according to Eq. (7.2) approximates the plasma behavior and from the control point of view emphasizes on some of its key properties.

A relation between a control and a state vector for the process (7.2) can be expressed as

$$\mathbf{u}[n] = -K[n] \mathbf{x}[n], \quad (7.4)$$

where K is a gain matrix with a dimensionality of $N_{coil} \times 3N$.

The active control is usually carried out by currents in poloidal coils \mathbf{J}_{PFC} and in solenoid \mathbf{J}_{SOL} : $\mathbf{J} = \mathbf{J}_{PFC} + \mathbf{J}_{SOL}$. In tokamaks currents $\mathbf{J}_{PFC} = \mathbf{J}_{PFC,1}, \dots, \mathbf{J}_{PFC,N_{PFC}}$ influence primarily the shape and position of the plasma while $\mathbf{J}_{SOL} = \mathbf{J}_{SOL,1}, \dots, \mathbf{J}_{SOL,N_{SOL}}$ effects primarily the total plasma current.

One way of these derivatives computing is to determine $\mathbf{h}(t, I_p(t), \mathbf{j}(t), \mathbf{J}(t))$ for increment only in one of the arguments and to calculate the difference ratio. However, this method implies the possibility of object transition into a state with one incremented parameter. It may cause some difficulties both in reality and in the numerical model, since, for example, the real object or its model may not exist with increments in only one parameter. Moreover, the numerical model requires the calculation of object state for each increment, which greatly increases the computation time. Therefore, when defining the derivatives, it is desirable to use a different approach, for example, the data about object state in previous moments of time.

Let us consider the linearized error (7.1) in time layer $n-1$. We have a system of N linear algebraic equations with $2NN_{coil}$ unknown values of $\partial \mathbf{h}[n-1]/\partial \mathbf{j}$, $\partial \mathbf{h}[n-1]/\partial \mathbf{J}$. Let us find and extrapolate them by Taylor's expansion formula at the time n and then construct the desired matrices A and B .

The system for derivatives of \mathbf{h} is underdetermined. In principle, it can be solved by a regularization method. However, we can reduce the number of unknown variables with the help of a priori information on the error of \mathbf{h} and obtain a system with a square matrix.

We take into account the fact that the control coil PFC has an influence primarily on its nearest plasma boundary region and will control the position of the boundary in the point closest to the coil. Also we will take into account that the solenoid has a weak effect on the position of the boundary. Then

$$\partial h_i[n-1]/\partial J_k = 0 \quad \text{for } i \neq k, \quad i \leq N_{PFC}, \quad k > N_{PFC}.$$

Besides we consider the full solenoid current $\mathbf{J}_{SOL, N_{SOL}} = 1$ instead of the currents in each solenoid coil and take into account that the major contribution to I_p is due to the derivative \mathbf{J}_{SOL}

$$\partial I_p[n-1]/\partial J_k = 0 \quad \text{for } k \leq N_{PFC}.$$

As for \mathbf{h} we consider only the deviation from the preset position of the boundary and from the specified total current.

As a result we obtain $N = N_{PFC} + 1$ equations with $N_{PFC} + 1$ unknowns.

The problem of specifying the initial values of derivatives of \mathbf{h} can be solved in several ways: by using a priori data; by using the increment for each argument of \mathbf{h} separately; by organizing of sub-iterations, repeating the first step up to obtaining the stationary values of derivatives.

Algorithms of control of only plasma shape or only total plasma current can be developed in similar manner. In the absence of control of currents in the solenoid $J_{SOL,i}$ these

currents must be set beforehand and in the absence of control of currents in the poloidal coils it is necessary to specify $J_{PFC,i}$.

To build a controller LQR (Linear-Quadratic Regulator) for the control of system described by Eq. (7.1), we need to find the state vector $\mathbf{x}[n]$ and control vector $\mathbf{u}[n]$ that minimize the following functional

$$I(\mathbf{u}) = \frac{1}{2} \sum_{n=1}^{\infty} (\mathbf{x}^T[n] Q \mathbf{x}[n] + \mathbf{u}^T[n] R \mathbf{u}[n]),$$

where Q and R are given matrices of dimension $N \times N_{coil}$ and $3N \times N_{coil}$ accordingly. The choice of Q and R provides a compromise between the magnitude of the errors when obtaining the desired state and the value of control.

At every relatively small time interval plasma model can be considered quasi-stationary so the search for the control vector $\mathbf{u}[n]$ can be reduced to the search for a gain matrix K (see, e.g., [6–8])

$$\mathbf{u}[n] = -K\mathbf{x}[n],$$

where

$$K = (B^T P B + R)^{-1} B^T P A,$$

P is a solution of a discrete algebraic Riccati equation (DARE)

$$P = Q + A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A.$$

The solution of Riccati equation P allows finding the performance index as well

$$I_{\min} = \frac{1}{2} \mathbf{x}^T[0] P \mathbf{x}[0].$$

The problem of existence and uniqueness of the solution of the LQR-control problem is studied, for example, in [7, 8]. Methods for solving Riccati equations are presented in [9–11]. The controller H_{∞} was applied as well, see for example, [6, 8, 12, 13, 15]. The H_{∞} refers to the so-called robust control methods that provide a good quality of control when mathematical model of the object is not known with sufficient accuracy or there are disturbances.

It is convenient to perform initial construction and testing of controllers using well-known Matlab software package. However, for automatic real-time feedback control of the plasma one should use a programming language that enables extremely fast construction of the controller and calculation of the signals for control devices. The fast calculation of a controller and the controls is implemented in Fortran 2008 language. The software is adapted for efficient operation with high speed on heterogeneous supercomputers of NIISI RAS. One video card can give GPGPU acceleration in up to 100 times, which is especially important for solving problems in real time.

Figs. 6 and 7 illustrate the application of system HASP CS to plasma modeling and control for the ITER device www.itpa.ipp.mpg.de parameters close to Scenario 2 and for T-15 tokamak in a scenario based on the data from [16].

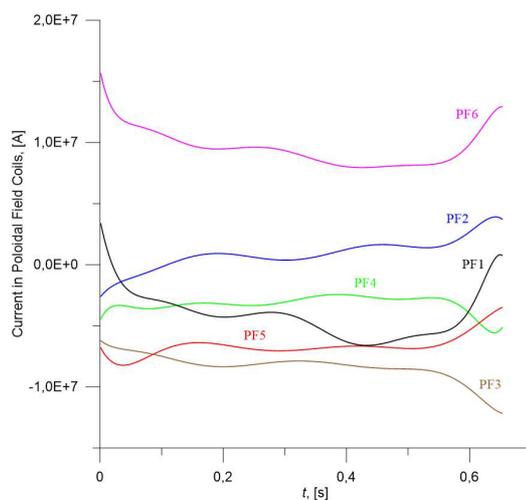


Figure 6: ITER poloidal coil currents obtained using HASP CS.

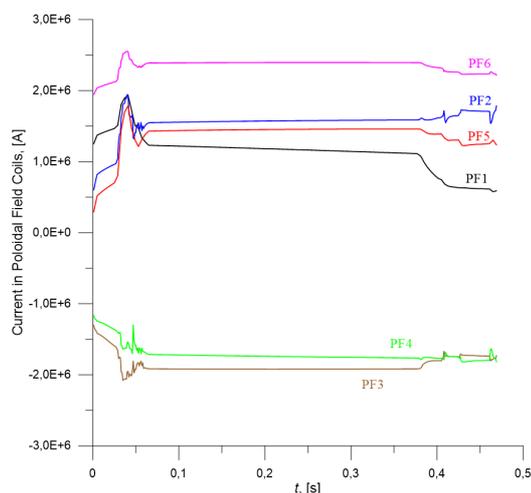


Figure 7: The same as in Fig. 4 for T-15 tokamak.

8 Graphical user interfaces

The development and study of plasma control systems include handling of a large number of models, organizing large input and output data streams, performing a number of calculations using distributed computing. To improve overall users productivity and research efficiency user-friendly graphical interfaces (GUI), data and processes visualization and animation tools are required.

As a part of the system HASP CS design the following tools have been developed: VD (Virtual Discharge) – environment for configuration and execution of computational

processes, VShell – system for data analysis and visualization, EnetComposer – a GUI application for ϵ -nets construction and visualization, VDcontrol – a GUI application for visualization and study of plasma control systems. All the mentioned systems are described briefly below.

VD is an interactive graphical user interface for configuration, tuning, execution and monitoring of computational processes. It is written purely in Java to get platform independent graphics. The size of the code is more than 20000 lines.

The basic concept of VD is "Drag & Drop" principle. All calculation sequences and configurations can be simply done using an interactive constructor interface with no need to manually modify codes and perform other complicated routine operations. VD has an intuitive GUI. A computational process is described as a directed graph. The vertices of the graph are the various blocks define the computational processes and their parameters. The edges connecting vertices define a sequence of calculations and methods of data exchange.

Advanced numerical codes can generate hundreds and thousands of output files. In order to analyze and visualize such data manually one must keep in mind its structure, format, units and define a plot title, axes labels, description of curves and so on. A number of universal general purpose graphic packages are often used. But many of them require proprietary and not always simple input data format, specifying a plurality of settings and even some code written in a custom internal language. Such tools are good for one-time advanced visualization but are usually ineffective or not suitable for monitoring of computations, especially remote ones, real-time filtration, visualization and animation of large number of functions.

VShell was developed to overcome above mentioned problems and work in cases when distributed systems are used. VShell is a substantial advancement of SCoPEShell framework [16]. Depending on its configuration VShell uses one of the following plotting tools: GnuPlot, Scavis, jzy3d or JFreeChart.

VShell is integrated into VD environment.

The ϵ -net construction and visualization package EnetComposer is also implemented in Java. This application allows experimenting with various ϵ -nets of polynomials and splines and select optimal classes of functions to be used in SDSS code.

Application VDcontrol was developed to visualize the solution of boundary feedback control problem allowing user to select a controller type, to show plasma boundary, the poloidal flux level lines, currents in plasma, solenoid and poloidal field coils, plasma boundary reconstruction errors bars and other parameters.

VDcontrol GUI makes it possible to view an animation of the plasma evolution. A fragment of VDcontrol GUI is presented in Fig. 8. The vertical cross-section of plasma in the (R, Z) plane is shown. Poloidal field coils positions are marked with numbers. Thin curves give contour lines of the poloidal flux function ψ . The deviation of obtained boundary from the desired is calculated at points on the thick curve. Calculated (blue curve) and desired (green curve) plasma boundaries practically coincide.

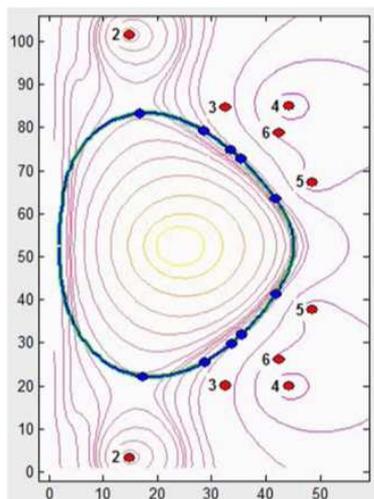


Figure 8: Visualization of control problem solution. Tokamak vertical cross-section. Normalized units.

9 Conclusions

A considerable step forward in the field of high-temperature toroidal plasma control is made with the help of NIISI RAS heterogeneous mini-supercomputers. Base components and toolkits of the software-hardware system HASP CS for modeling and automatic plasma real-time feedback control are designed and implemented.

The system is based on fundamentally new, more accurate and robust control algorithm, which uses solution of a number of diagnostic inverse problems with a given inaccuracy by the ϵ -nets method. Parallel algorithms implementing the ϵ -nets method for CPU and hybrid CPU/GPGPU NIISI RAS architectures were developed using MPI, OpenCL and OpenMP and were studied. Code SCoPE, which gives advanced plasma model, is parallelized with MPI. All this allows speedup of HASP CS system on supercomputers in ~ 10 -100 times with much more acceleration of particular algorithms, such as ϵ -net ones.

It is demonstrated that parallel approaches can be used for solution of the real-time feedback control problem for boundary and internal plasma parameters and for detailed diagnostic data processing between pulses. Low price and operating costs of CPU/GPGPU complex make it possible to dedicate it solely to real-time control problems.

It is shown that the software-hardware system HASP CS is applicable for studies of processes in both running devices and being designed thermonuclear fusion reactors and in general for solving problems of fusion energy production.

Development of the software infrastructure of heterogeneous computing based on OpenCL standard as well as promising new generation hybrid NIISI RAS supercomput-

ers will make it possible to solve new 2D and 3D plasma diagnostics problems, to extend the class of functions used in ϵ -nets and to improve the accuracy.

The studies of plasma control processes done using HASP CS system allow formulating the following important recommendations for plasma control system design:

1. The developed methodology for the plasma control can be recommended for implementation in tokamaks of different size and construction including ITER and T-15. The use of NIISI RAS supercomputers and HASP CS system for real-time generation of device control signals is reasonable.
2. Real-time plasma boundary control problem can be efficiently treated on the basis of the inverse problem solution by the ϵ -nets method using only magnetic diagnostics and information about the device design.
3. Real-time feedback control problem for plasma internal parameters can be efficiently solved by the ϵ -nets method as well. However, having only magnetic diagnostic data is not enough for automatic plasma parameters control whatever the inverse problem solution method is. It is necessary to use various diagnostics simultaneously: MSE (Motional Stark Effect), polarimetry, interferometry, kinetic pressure and other measurements giving the information about distributions of functions over the flux surfaces.
4. The complexity of different radial profiles in ITER requires accurate study of each operation regime, e.g., by using the proposed here control methodology together with different codes for modeling tokamak plasma dynamics. Commonly used semi-empirical greatly simplified approaches for construction of controllers can lead to bulk expensive work on the experimental search of the control system parameters while launching ITER and to a large number of premature discharge disruptions during research campaigns.

The completion of the described software-hardware system HASP CS development will create a base for the next level plasma control problems such as: formation of the predefined initial plasma configuration during gas breakdown; control in critical situations: suppression of arose instabilities, prediction and prevention of unplanned discharge disruptions, overheating of vacuum chamber construction elements; safe discharge completion; improvement of thermonuclear energy output and economic efficiency of power plants.

Acknowledgements

The authors are grateful to the head of NIISI RAS, Academician V. B. Betelin, the head of NIISI RAS department A. G. Kushnirenko for support and discussions and to P. B. Bogdanov for assistance in software modification for NIISI RAS heterogeneous computers. Research was partly funded by Grant I.33 of Presidium of Russian Academy of Sciences.

References

- [1] F. S. Zaitsev, *Mathematical Modeling of Toroidal Plasma Evolution*, English edition, M.: MAKS Press, 2014.
- [2] NIISI computers OpenCL, <http://devgurus.amd.com/thread/159457>.
- [3] O. Yu. Sudareva and P. B. Bogdanov, Heterogeneous programming in the framework of OpenCL standard, Proc. of XV International Conf. "Supercomputing and mathematical modeling", October 2014, Sarov, 125–127, 2014.
- [4] A. N. Tikhonov and V. Ya Arsenin, *Methods for Ill-Posed Problems*, M.: Nauka, 1986.
- [5] R. Coelho, S. Matejcek, P. McCarthy, E. P. Suchkov and F. S. Zaitsev, EU-IM Team and ASDEX Upgrade Team. Evaluation of epsilon-net calculated equilibrium reconstruction error bars in the European integrated modeling platform, *Fusion Sci. Technol.*, 69(3) (2016), 611–619.
- [6] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analyses and Design*. Chichester: John Wiley and Sons, 2000.
- [7] K. Ogata, *Discrete-Time Control Systems*, New Jersey, Englewood Cliffs: Prentice Hall International Inc., 1995.
- [8] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Oxford: Clarendon Press, 1995.
- [9] A. J. Laub, A Schur Method for Solving Algebraic Riccati equations, *IEEE Trans. Auto. Contr.*, AC-24: 913–921, 1979.
- [10] W. M. Wonham, On a matrix Riccati equation of stochastic control, *SIAM J. Contr.*, 6 (1968), 681–697.
- [11] V. Sima, *Algorithms for Linear-Quadratic Optimization*, Pure and Applied Mathematics: A Series of Monographs and Textbooks, vol. 200, Marcel Dekker, Inc., New York, 1996.
- [12] M. Green and D. Limebeer, *Linear Robust Control*, Prentice Hall, 1994.
- [13] D. McFarlane and K. Glover, *Robust Controller Design Using Normalised Coprime Factor Plant Descriptions*, Springer-Verlag, 1990.
- [14] P. Hr. Petkov, D. W. Gu and M. M. Konstantinov, Fortran 77 routines for Hinf and H2 design of linear discrete-time control systems, Report 99-8, Department of Engineering, Leicester University, April 1999.
- [15] E. A. Azizov, V. A. Belyakov, O. G. Filatov, E. P. Velikhov and T-15MD Team, Status of Project of Engineering-Physical Tokamak, 23rd International Atomic Energy Agency (IAEA) Fusion Energy Conf., Daejeon, South Korea, FTP/P6-01, 2010.
- [16] D. P. Kostomarov, S. V. Stepanov and A. G. Shishkin, Virtual Discharge: Integrated Modeling Environment for Supporting Numerical Experiments with Gas Discharge Plasmas, *Doklady Mathematics*, 90(3) (2014), 688–691.
- [17] R. J. Akers, et al., Neutral beam heating in the START spherical tokamak, *Nuclear Fusion*, 42 (2002), 2–135.
- [18] Code EFIT: <https://fusion.gat.com/theory/Efit>.
- [19] Code TRANSP: <http://w3.pppl.gov/transp/>.