LRP 404/90 May 1990

PLASMA SHAPE AND POSITION CONTROL IN HIGHLY ELONGATED TOKAMAKS

F. Hofmann and S.C. Jardin

(to be published in Nuclear Fusion)

PLASMA SHAPE AND POSITION CONTROL IN HIGHLY ELONGATED TOKAMAKS

F. Hofmann
Centre de Recherches en Physique des Plasmas,
Association Euratom-Confédération Suisse,
Ecole Polytechnique Fédérale de Lausanne, 21 Avenue des Bains,
1007 Lausanne, Switzerland.

S.C. Jardin
Plasma Physics Laboratory, Princeton University
P.O. Box 451, Princeton, New Jersey 08543, USA.

Abstract

Plasma shape and position control in elongated tokamaks is analyzed, using the TSC code. A new algorithm is presented which allows the shape evolution of a discharge to be entirely preprogrammed. The algorithm computes poloidal field coil voltages as functions of time, using magnetic measurements taken close to the vessel wall. No preprogrammed coil current wave-forms are required. By simulating the startup phase of typical TCV tokamak discharges, it is shown that the actual shape evolution follows the preprogrammed one very precisely. X-points can be specified at arbitrary positions. Active stabilization of the dominant vertical mode is achieved by a conventional PD feedback loop. The optimization of feedback coefficients, the tradeoff between shape accuracy and power dissipation in the poloidal field coils, as well as the computing power requirements for implementing the algorithm in a real experiment are discussed.

1. Introduction

One of the major challenges in operating large, elongated tokamaks is the triple requirement of accurate shape control, optimized vertical position control and efficient plasma current control. Several methods have been proposed [1-4] to solve this problem. In tokamaks with moderate plasma elongation, the passive stabilizers do not have to be very close to the plasma boundary, and some deviation between the actual and ideal plasma shapes can be tolerated. In this case it is generally sufficient to control the plasma current, radial and axial position and a few shape parameters (ellipticity, triangularity, etc.). However, when the plasma shape is more complicated and the elongation very large [3,4], a tightly fitting shell is required for passive stabilization and the actual plasma shape must correspond to the design shape with very good accuracy. Under these circumstances, it is usually assumed that the current in each of the poloidal field coils is the sum of a preprogrammed current and a correction current, the latter being a small fraction of the former. The preprogrammed currents can either be obtained from an MHD code, or inferred from a previous discharge. The correction currents, on the other hand, are computed in real time such as to adjust the plasma position and shape. In this approach, the accuracy with which a predetermined plasma shape can be produced depends to some extent on how well one is able to guess the plasma profile functions when computing the preprogrammed shaping currents. This implies that the establishment of a discharge in a new parameter regime may be a very time-consuming trial-and-error process.

In this paper, we propose a new method for plasma shape control which does not depend on preprogrammed coil currents. The method assumes, on the other hand, that the shape of the plasma boundary is preprogrammed, and coil currents and voltages are computed in real time from a comparison between the actual and preprogrammed plasma shapes.

The proposed algorithm has been tested by numerical simulation, using the TSC code [5]. Practical implementation of the algorithm in a tokamak requires a control system with

considerable real-time computing power. This should be possible in two tokamaks, presently under construction, i.e., Alcator C-MOD [6] and TCV [7].

2. Control Algorithm

The control algorithm proposed in this paper is shown schematically in Fig.1. The basic idea is to control the poloidal flux, and in some cases the poloidal magnetic field, at a number of points at the plasma boundary. The coordinates of these points are preprogrammed, and they define the desired evolution of the plasma boundary.

It is clear that fluxes inside the vacuum vessel cannot be measured directly. They have to be estimated from measurements taken at some distance from the plasma, usually close to the vacuum vessel wall. Several methods are available for this purpose. We are using an algorithm [8] based on a finite element reconstruction of the plasma current. This gives accurate flux values on the plasma boundary, i.e., it allows to determine the plasma shape with good precision.

Since we wish to simulate tokamak evolution as realistically as possible, we consider a discrete control system, i.e., we assume that the control variables ($\vec{V}_s$) are updated only at regular intervals and are left constant between updates. The time between updates, Δt_u, is essentially given by the finite bandwidth of the power supplies. It is typically between 0.2 and 2 ms. Measurements, however, are taken continuously, and we assume that, at the time of a voltage update, instantaneous measurements as well as those taken at the previous update are available as input to the control algorithm.

The control system shown in Fig.1 consists of four basic elements: The first (matrix A) uses measurements ($\vec{I}_s, \vec{I}_v, \vec{B}_w, \vec{\Psi}_w, I_p$) and preprogrammed functions ($\vec{R}_b, \vec{Z}_b, I_p, \Psi_{sh}$) to compute error signals ($\vec{\varepsilon}_\psi, \vec{\varepsilon}_B, \varepsilon_z, \varepsilon_{I_p}$). These error signals are then fed into PID controllers (second element) where they are mixed with their own time-derivatives and integrals. The third element (matrix M^{-1}) calculates the rate of change of the shaping currents ($\dot{\vec{I}}_s$) which will produce the maximum possible error reduction. The fourth element (matrix L) finally computes the coil voltages which are

necessary to produce the desired rate of change $\dot{\vec{I}}_s$.

The three matrices, A, M^{-1}, and L, will in general vary with time, since they depend on the preprogrammed functions. However, they do not depend on the plasma parameters (β_p, l_i, q_o, etc.) and no plasma simulation whatsoever is necessary to generate these matrices. This is important because any simulation is based on arbitrary assumptions which may or may not apply in a real experiment.

In the following sections, we shall describe the basic elements of the control algorithm in more detail.

2.1. Error Vectors

2.1.1. Shape Errors

Let us define the plasma shape by a number of points, distributed more or less uniformly along the plasma boundary. The coordinates of these points are preprogrammed functions of time, $\vec{R}_b(t)$, $\vec{Z}_b(t)$. The flux errors at these points can be expressed in terms of all currents in the system,

$$\vec{\varepsilon}_\psi = A_1\vec{X} + A_2\vec{I}_s + A_3\vec{I}_v \qquad (1)$$

where $\vec{X}$ represents the amplitudes of the plasma current elements, $\vec{I}_s$ the shaping coil currents and $\vec{I}_v$ the vacuum vessel currents. The elements of $\vec{\varepsilon}_\psi$ are defined as $\varepsilon_{\psi_i} = \Psi_i - \Psi_1$, where Ψ_i are the fluxes at the preprogrammed boundary points and Ψ_1 is the limiter flux. We assume, for simplicity, that the contribution of the OH transformer to the flux differences ε_{ψ_i} is negligible. The vector $\vec{X}$ can be expressed [8] in terms of the measurement vector $\vec{F}$, and hence eq.(1) can be written as

$$\vec{\varepsilon}_\psi = A_4\vec{F} \qquad (2)$$

where $\vec{F} = (\vec{I}_s, \vec{I}_v, \vec{B}_w, \vec{\Psi}_w)$, and A_4 is a matrix depending on $\vec{R}_b(t)$, $\vec{Z}_b(t)$. Flux loops and magnetic field probes are placed close to the vacuum vessel wall, as is the case in a real experiment. No "measurements" are taken in physically inaccessible areas. Vessel

currents are assumed to be derived from magnetic field measurements inside and outside the wall.

2.1.2. Volt-Second Error

In order to minimize the interaction between the shape control and plasma current control systems, it is often useful to preprogram the volt-seconds supplied by the shaping system. For this purpose, we compute the absolute value of the flux produced by the shaping coils, at a particular point inside the plasma, preferably close to the magnetic axis. The coordinates of this reference point are preprogrammed functions of time. We then compare the flux at this point with the prescribed shaping flux, ψ_{sh}, and obtain an additional flux error, $\varepsilon_{\psi\,sh}$, which is to be minimized simultaneously with the flux errors, $\vec{\varepsilon}_{\psi}$ (eq.1).

2.1.3. Poloidal Field Errors

If we wish to produce a plasma having an X-point at a predetermined position (R_X, Z_X) we must require that the poloidal field vanishes at that point. The magnetic field error at the X-point can be written as

$$\vec{\varepsilon}_B = A_5\vec{X} + A_6\vec{I}_S + A_7\vec{I}_V \qquad (3)$$

where $\vec{\varepsilon}_B$ contains both B_R and B_Z components of all X-points which we wish to specify. In many cases, an X-point will coincide with one of the boundary points, as described above. The limiter flux, ψ_1, mentioned in section 2.1.1., is then replaced by the flux at the preprogrammed X-point location. We note that the poloidal field errors can be treated in exactly the same way as the flux errors. It should be remembered, however, that X-point control is intrinsically inaccurate due to the flatness of the flux function in the vicinity of the X-point.

2.1.4. Positional Errors

If we fix the plasma boundary, as described in section 2.1.1., above, the horizontal and vertical position of the plasma is also fixed. Nevertheless, it is useful to have a separate control system

for the vertical position, since the plasma is in most cases vertically unstable and vertical position control requires much more rapid feedback action than shape control does.

The vertical position can be measured in several different ways. A commonly used method is to measure the flux difference between two observation points, one situated near the top and the other near the bottom of the plasma cross section. The positional error is then assumed to be proportional to the measured flux difference. This method is perfectly adequate for near-circular or moderately elongated plasmas, but it may fail for highly elongated, D-shaped or bean-shaped plasmas [9], where the unstable "vertical" mode produces a tilting motion rather than a global vertical displacement.

A second method consists of evaluating the vertical position of the plasma current centroid,

$$Z_p = \frac{1}{I_p} \int j_p \, Z \, dR \, dZ \qquad (4)$$

In fact, this quantity can easily be obtained from the finite element reconstruction of the plasma current,

$$Z_p \cong \frac{\sum_i Z_i X_i}{\sum_i X_i} \qquad (5)$$

where X_i is the amplitude of the i-th element and Z_i is the vertical coordinate of its central node. The positional error is then obtained by comparing the value of Z_p as given by eq. (5) with some reference value, Z_p^o. This method can be used successfully in many cases, but it has one major shortcoming: If the plasma shape is preprogrammed, as outlined in section 2.1.1., above, and if the plasma is not up-down symmetric, then it is impossible to define a self-consistent reference signal Z_p^o which is independent of the plasma parameters (β_p, l_i, etc.). For this reason, we do not use eq.(5), but we define the positional error as a surface integral,

$$\varepsilon_Z = \frac{\pi(a+b)\int \sin\alpha \; \Delta\psi \; ds}{\mu_o I_p R_o \int ds} \qquad (6)$$

where a and b are the horizontal and vertical plasma minor radii, α is the angle between the tangent to the plasma boundary contour and the z-axis, $\Delta\psi$ is the measured flux error on the preprogrammed boundary, ds is the contour line element and R_o is the major plasma radius. This definition depends only on preprogrammed and measured quantities and is completely independent of plasma parameters. It gives us the positional error directly and does not require a $Z_p{}^o$ reference signal. In fact, the reference signal for the vertical position is implicitely given by the preprogrammed boundary points.

The definition (6) has the advantage that it is automatically consistent with the preprogrammed shape. This ensures that, if the plasma shape and vertical position are controlled by separate systems, these systems do not fight against each other, i.e.,they do not give conflicting instructions to the power supplies.

2.2. PID Controllers

The control algorithm presented here includes the option of sending all error signals through PID controllers (Fig.1.) These devices produce signals of the form

$$\vec{\varepsilon}_{PID} = \alpha\,\vec{\varepsilon} + \beta\dot{\vec{\varepsilon}} + \gamma\int_o^t \vec{\varepsilon}\,dt \qquad (7)$$

where the coefficients α, β and γ are given constants or preprogrammed functions of time. In the shape control and plasma current control branches of the system (Fig.1), PID controllers are generally not necessary. In fact, all simulations presented in this

paper assume that $\beta = \gamma = 0$ for shape and plasma current control. For position control however, proportional and derivative terms are absolutely necessary. The choice of optimum coefficients is discussed in section 3.3.

2.3. Rate of Change of Shaping Currents

Let us combine the flux and magnetic field errors into a single error vector, $\vec{\varepsilon}_S = (\vec{\varepsilon}_\psi, \vec{\varepsilon}_B)$. Using eqns. (1) and (3), the time drivative of $\vec{\varepsilon}_S$ can then be expressed as

$$\dot{\vec{\varepsilon}}_S = A_8 \dot{\vec{I}}_S + A_9 \dot{\vec{I}}_V + A_{10} \dot{\vec{X}} \tag{8}$$

where A_8, A_9 and A_{10} are known matrices. We now wish to determine a vector $\dot{\vec{I}}_S$ which gives us an $\dot{\vec{\varepsilon}}_S$ such that the measured errors are minimized. The problem is complicated by the fact that the vectors $\dot{\vec{I}}_S$, $\dot{\vec{I}}_V$ and $\dot{\vec{X}}$ are all coupled. The coupling between vacuum vessel and shaping currents is described by the vessel circuit equation and can be included in the analysis in a straightforward manner. Here, we neglect this effect in view of the fact that shape control is usually performed on time scales which are comparable or longer than typical decay times of vessel currents. If one wishes to control the plasma shape on shorter time scales, e.g. in presence of fast plasma phenomena, such as pellet injection, vessel effects should be taken into account.

The coupling between the plasma current distribution ($\vec{X}$) and the shaping and vessel currents, on the other hand, is much more complicated. It can be evaluated by a perturbation analysis [10], using an MHD equilibrium code. However, this requires a number of arbitrary assumptions concerning the plasma pressure and current profiles. In this paper, we choose not to include the plasma effects, since our aim is to construct a model-independent shape control algorithm. In summary, we neglect the last two terms on the r.h.s. of eq. (8) and write

$$\dot{\vec{\varepsilon}}_S \approx A_8 \dot{\vec{I}}_S \tag{9}$$

In principle, the elements of $\dot{\vec{I}}_S$ can be computed by assuming that the error vector $\vec{\varepsilon}_S$ should be reduced to zero within a time interval Δt,

$$\vec{\varepsilon}_S(t+\Delta t) = \vec{\varepsilon_S}(t) + \Delta t\, \dot{\vec{\varepsilon}}_S = 0 \qquad (10)$$

where $\vec{\varepsilon_S}(t)$ is the measured error vector at time t. Combining (9) and (10) then gives

$$\dot{\vec{I}}_S(t) = -\frac{1}{\Delta t} A_8^{-1}\, \vec{\varepsilon}_S(t) \qquad (11)$$

where Δt is an arbitrary parameter playing the role of a feedback gain. If the matrix A_8 is not square, A_8^{-1} represents a pseudo-inverse [11]. This approach, however is not always feasible in practice [12,13]. If, for example, the shaping coils are fed by individual power supplies, and if the distance between two adjacent coils is of the same order or less than the distance between coils and plasma boundary, the application of eq. (11) will lead to uncontrollably large coil currents of oscillating sign.

This problem can be avoided by a regularization procedure [13] which tends to smooth the shaping current distribution while minimizing the flux and magnetic field errors. To achieve this, we use a cost functional of the form

$$U = \vec{\delta}^T E\, \vec{\delta} + \vec{I}_S^T(t+\Delta t)\, D\, \vec{I}_S(t+\Delta t) \qquad (12)$$

where $\vec{\delta} = \vec{\varepsilon}_S + A_8\, \Delta t\, \dot{\vec{I}}_S$, $\vec{I}_S(t+\Delta t) = \vec{I}_S(t) + \Delta t\, \dot{\vec{I}}_S$ and D and E are diagonal matrices containing suitable weighting coefficients. The minimization of U with respect to variations in $\dot{\vec{I}}_S$ then leads to a linear system of equations which is readily solved for the elements of $\dot{\vec{I}}_S$.

It should be noted that, in eq. (12), the shaping currents $\vec{I}_S$ play the role of error signals. In fact, the algorithm tends to reduce the

shaping currents if this is possible without producing unacceptable shape errors. It is, of course, impossible to reduce the errors to zero, for the following reasons: First of all, we must avoid excessively large shaping currents. Secondly, the number of elements in $\vec{\delta}$ is generally larger than the number of shaping currents. Finally, in cases where the plasma shape is programmed to evolve according to a given scenario, the power supplies may reach their current or voltage limits. The fact that the errors remain non-zero does not imply that the correction vector $\dot{\vec{I}}_S$ will always be finite. In the case of steady-state shape programming, $\dot{\vec{I}}_S$ will tend to zero and the asymptotic shape errors are related to the shaping currents by

$$A_g^T E \vec{\varepsilon}_s + D \vec{I}_s = 0 \qquad (13)$$

If vertical position control is to be accomplished by the shaping coils as well, the $\dot{\vec{I}}_S$ obtained from the shape optimization must be modified by an additional term, $\dot{\vec{I}}_S^V$, which is given by

$$\dot{\vec{I}}_S^V = \vec{C}^V \left[\alpha_z \varepsilon_z + \beta_z \dot{\varepsilon}_z + \gamma_z \int_o^t \varepsilon_z \, dt \right] . \qquad (14)$$

The vectror $\vec{C}^V$ is the product of a constant vector, producing a suitable radial field, and the plasma current I_p. ε_z is given by eq. (6) and α_z, β_z, γ_z are feedback coefficients. These are discussed in section 3.3.

2.4. Shaping Coil Voltages

Shaping coil voltages are computed using the circuit equations,

$$\vec{V}_s = R_s \vec{I}_s - \vec{V}_l + L_{ss} \dot{\vec{I}}_s \qquad (15)$$

where R_S and L_{SS} are the resistance and inductance matrices and $\vec{V}_l$ is the loop voltage produced by the OH transformer. If the coupling

between the shaping coils and the vacuum vessel is taken into account, we obtain

$$\vec{V}_s = R_s\vec{I}_s - \vec{V}_l + [L_{ss} - L_{vs}L_{vv}^{-1}L_{sv}]\dot{\vec{I}}_s - L_{vs}L_{vv}^{-1}[R_v\vec{I}_v - \vec{V}_l] \qquad (16)$$

where L_{vs} is the mutual inductance matrix between shaping coils and vacuum vessel elements, and L_{vv} and R_v are the self inductance and resistance matrices of the vessel elements, respectively.

In the work presented here, we always use eq. (15) for calculating the coil voltages. This is done for reasons of consistency with the assumptions leading to eq. (9).

3. Numerical Simulation

3.1. TCV Startup

As a first example, we consider the startup phase of a TCV tokamak discharge. TCV [7] poses extremely difficult control problems since it has a thick, continuous vacuum vessel, and the plasma elongation can reach values up to 3.0. We assume that the discharge is initiated by creating a circular plasma in the upper half of the vessel. During current ramp-up, the plasma is then deformed and stretched, until it reaches the desired D-shape. The scenario is asymmetric with respect to the midplane, in order to take advantage of the stabilizing effect of the top wall [14].

We simulate this scenario with the TSC code [5], using the control algorithm described in section 2. The result is presented in Fig.2., which shows flux plots at five stages during the evolution. The plasma shape is preprogrammed by prescribing the trajectories of 20 boundary points. These trajectories are chosen in such a way that the minimum plasma-wall distance on the top and on the two sides is constant during the whole evolution. Within the control algorithm, the plasma current is represented by 10 finite elements (vector $\vec{X}$ in eq.(1)), covering the rectangular area 0.64m<R<1.11m, -0.71m<Z<0.71m. The vacuum vessel is divided up into 28 elements, each carrying a discrete current. The decay time of the dominant mode for vertical stabilization is 6.7 ms. There are 28 flux loops, located outside the vessel, and 28 magnetic field probes, located just inside the vessel. The probes are oriented such that they only measure the field component parallel to the wall. Flux errors are computed from eq. (1), coil current increments are obtained by minimizing eq. (12) and the coil voltages are computed from eq. (15). The matrices D and E in eq. (12) are taken as $D=10^{-5}I$ and $E=I$, where I is the unit matrix. The plasma current follows a preprogrammed wave-form, increasing linearly from 200kA to 1.0MA within 160 ms. Coil currents are shown as functions of time in Fig.3. This figure covers the time interval between 50 and 90 ms, during which the plasma elongation evolves form $\kappa=1.625$ to $\kappa=2.125$.

We note that, in this example, the number of boundary points (20) exceeds the number of shaping coils (16). Therefore, it is impossible to match all the boundary points exactly, even when the smoothing term (D-matrix in eq.12) is zero. Nevertheless, a close examination of the actual and preprogrammed plasma shapes shows that they coincide to within a few mm. The shape quality can be varied by adjusting the coefficients of the D-matrix. When these coefficients are reduced, the shape accuracy improves but at the same time, the power dissipation in the poloidal field coils increases. In practice, the D-matrix is adjusted to find an optimum compromise, consistent with the requirements and constraints of the particular experiment. This point is illustrated in section 3.2.

Fig.4. shows typical flux errors at the 20 boundary points, expressed as a percentage of the total poloidal flux in the plasma. The flux errors were evaluated during the scenario shown in Fig.2., at time t=60 ms, when the plasma elongation has reached a value of 1.75. Fig.5. shows the evolution of the RMS flux error, averaged over all boundary points. It should be noted that the errors quoted here depend on the quality of the measurements. In our simulation, flux and magnetic field measurements are obtained by an interpolation, based on the discrete Ψ-matrix. The errors due to the finite mesh size are typically 1 to 2%. Measurements in real experiments with comparable accuracy should be possible. The influence of random measurement errors on the shape accuracy has been investigated elsewhere [8].

It should also be noted that the vertical instability is successfully stabilized throughout the evolution (Fig.2). Pure proportional feedback was used for shape control and pure derivative feedback for position control ($\alpha = \gamma = 0$ in eq.14). The two non-zero coefficients were optimized for a plasma with elongation, κ = 2.8, and they were kept constant throughout the run. κ = 2.8 is the critical elongation, in this scenario, at which the growth rate of the vertical instability reaches a maximum [15]. For $\kappa > 2.8$, the growth rate decreases, as the plasma approaches the bottom wall.

3.2. Shape Quality

In order to demonstrate the connection between shape quality and power dissipation in the poloidal field coils, we consider a plasma with peapod-shaped cross section. This plasma is obtained from a time-dependent TSC simulation, starting with a D-shaped equilibrium. The boundary points are programmed in such a way that they first make the transition from D to peapod, within 10 ms, and then remain fixed for another 30 ms. The result is shown in Fig.6. for two different values of the smoothing term (D-matrix in eq.(12)). It is seen that in the case with the larger D-matrix (Fig.6b), the undulations on the inboard side disappear almost completely and the power dissipation drops by a factor 2.

3.3. Optimization of Feedback Coefficients

Let us consider a vertically unstable plasma with elongation $\kappa=2$, as shown in the third plot of Fig.2. At time t=0, we displace all boundary points vertically by a small distance, ΔZ. We then let the system evolve, using steady-state programming. Fig.7. shows the vertical position of the magnetic axis as a function of time, for three different combinations of coefficients α_Z and β_Z (eq.14). Here, we are using four coils (i.e. the top and bottom coils of the inner and outer stacks of shaping coils) to control the vertical position. The coils inside the vacuum vessel (see Fig.1.) are not activated. Relative merits of coils inside and outside the vessel have been discussed elsewhere [16]. In the shape control branch, only proportional feedback is used, whereas in vertical position control, proportional and derivative terms are active ($\gamma_Z=0$ in eq.14). It should be noted that the shape control loop also contibutes to vertical position control. Therefore, when $\alpha_Z=0$, the effective proportional term for position control is determined by the gain in the shape control loop.

We note that, in the case shown in Fig.7a, the proportional term (α_Z) was chosen too large, producing considerable oscillation. In Fig.7b, the conditions are nearly optimal in the sense that the system is close to a state of critical damping. Finally, in Fig.7c, the velocity feedback coefficient (β_Z) was reduced by a factor 2. Stability is still achieved, but the control is much more sluggish. We conclude that the choice of feedback coefficients is not critical in the case considered here. This makes it possible to use constant coefficients for a complete startup evolution, as shown in Fig.2.

The evolution of the coil voltages for the case corresponding to Fig.7a is shown in Fig.8. No voltage limit has been applied in this case. Maximum values are seen to be close to ±50 V/turn. In the case of Fig.7b, maximum coil voltages are also about ±50 V/turn, whereas in the case shown in Fig.7c, they are reduced to approximately ±25 V/turn.

Optimum feedback coefficients can also be estimated theoretically, by considering a simplified model of a vertically unstable plasma in a conducting shell. This approach is discussed in Ref. [17].

3.4. Creation of X-Points

In divertor tokamaks, precise control of the separatrix location is extremely important, because the heat load distribution on the first wall depends critically on the position of the X-point. The control algorithm presented in section 2 is able to produce a vanishing poloidal field at any predetermined position (R_X, Z_X) within the vacuum vessel. If the point (R_X, Z_X) coincides with one of the boundary points, a magnetic separatrix is created. Of course, the preprogrammed boundary shape and the position of the X-point must be physically compatible, i.e., the boundary must exhibit a right angle at that point. The precision of the X-point location can be adjusted by varying those coefficients of the E-matrix (eq. 12) which correspond to the magnetic field and flux errors at the X-point.

Fig.9 shows a single-null divertor plasma in which an X-point

has been specified at R = 0.76 m, Z = -0.50 m . It is seen that the position of the X-point, as deduced from the flux contours, agrees well with the preprogrammed X-point location, indicated by a cross in Fig.9.

3.5. Computing Requirements

In this section, we estimate the computing power which is required to implement the proposed algorithm in a real experiment. As an example, let us consider a TCV startup scenario, as shown in Fig.2. The matrices A, M^{-1} and L contain 21x102, 18x36 and 18x18 elements, respectively. Consequently, each voltage update requires approximately 3100 multiplications and 3100 additions. These calculations are performed typically once every millisecond. TCV will be using an analog-digital hybrid control system, in which the matrices A, M^{-1} and L are supplied in digital form, whereas the input and output signals (i.e. measurements and coil voltages) appear in analog form. The matrices A and M^{-1} are functions of time, since they depend on the coordinates of the preprogrammed boundary points. These matrices can be updated at frequencies up to 800 Hz. The analog system bandwidth is over 5kHz. If the control system were built using digital technology exclusively, an array processor with a computing power of the order of 10 Mflops would be required.

4. Conclusion

A new algorithm for plasma shape and position control in tokamaks has been developed and implemented in the Tokamak Simulation Code (TSC). The algorithm computes poloidal field coil voltages such that the plasma shape evolves according to a preprogrammed scenario, irrespective of changes in the plasma parameters. No preprogrammed coil currents are used. X-points can be specified at arbitrary positions. Position control of vertically unstable plasmas is achieved by the same algorithm, by means of conventional PID controllers.

The new algorithm has been tested by simulating a number of TCV tokamak discharges. It is shown that the flux errors at the preprogrammed boundary points are of the order of 1% of the total poloidal flux within the plasma. By considering an elongated plasma with peapod-shaped cross section, the relation between shape accuracy and power dissipation in the poloidal field coils is illustrated.

Implementation of the algorithm in a real tokamak requires an on-line control computer capable of executing several matrix operations between updates of the coil voltages. Two tokamaks presently under construction (Alcator C-MOD and TCV) will be equipped with control systems which are sufficiently powerful to perform these operations in real time.

Acknowledgements

Stimulating discussions with Dr. F.B. Marcus on the numerical simulation of highly elongated plasmas are gratefully acknowledged.

This work was partly supported by EURATOM, the Ecole Polytechnique Fédérale de Lausanne and the Fonds National Suisse de la Recherche Scientifique. The research was sponsored by the Office of Fusion Energy, U.S. Department of Energy, under contract DE-AC02-76-CHO-3073.

References

[1] CISCATO, D., DE KOCK, L., NOLL, P., in Fusion Technology 1980 (Proc. 11th Symposium, Oxford, 1980), Vol. 2, EEC, Oxford (1981) 1033.

[2] LUXON, J., ANDERSON, P., BATTY, F., BAXI, C., BRAMSON, G., et al., in Plasma Physics and Controlled Nuclear Fusion Research 1986 (Proc. 11th Int. Conf. Kyoto, 1986), Vol. 1, IAEA, Vienna (1987) 159.

[3] BELL, R.E., FISHMAN, H., HATCHER, R., JARDIN, S.C., KESSEL, C.E., et al., 13th Syposium on Fusion Engineering, Knoxville, Tennessee, October 2-6, 1989.

[4] MARCUS, F.B., JARDIN, S.C. and HOFMANN, F., Phys. Rev. Lett. 55 (1985) 2289.

[5] JARDIN, S.C., POMPHREY, N. and DELUCIA, J., J. Comput. Phys. 66 (1986) 481.

[6] HUTCHINSON, I.H., BECKER, H., BONOLI, P., DIATCHENKO, N., FAIRFAX, S., et al., MIT Plasma Fusion Center Rept. PFC/RR-88-11.

[7] HOFMANN, F., JARDIN, S.C., MARCUS, F.B., PEREZ, A., TURNBULL, A.D., in Fusion Technology 1986, (Proc. 14th Symposium, Avignon, 1986), Vol.1, EEC, Oxford (1986) 687.

[8] HOFMANN, F. and TONETTI, G., Nuclear Fusion 28 (1988) 519.

[9] POMPHREY, N., JARDIN, S.C. and WARD, D.J., Nuclear Fusion 29 (1989) 465.

[10] ALBANESE, R., AMBROSINO, G., COCCORESE, E., GAROFALO, F., RUBINACCI, G., in Controlled Fusion and Plasma Physics (Proc. 16th Europ. Conf., Venice, 1989), Part I, Vol. 13B, 447.

[11] FORSYTHE, G.E., MALCOLM, M.A., MOLER, C.B., Computer Methods for Mathemetical Computations, Prentice-Hall, Englewood Cliffs, N.J., 1977.

[12] LACKNER, K., Comput. Phys. Commun. 12 (1976) 33.

[13] HOFMANN, F., Comput. Phys. Commun. 48 (1988) 207.

[14] HOFMANN, F., TURNBULL, A.D., MARCUS, F.B., Nuclear Fusion 27 (1987) 743.

[15] HOFMANN, F. and SCHULTZ, C.G., in Controlled Fusion and Plasma Physics (Proc. 16th Europ. Conf., Venice, 1989), Part I, Vol. 13B, 335.

[16] MARCUS, F.B., HOFMANN, F., JARDIN, S.C., NOLL, P., TONETTI, G., to be published in Nuclear Fusion.

[17] LAZARUS, E.A., LISTER, J.B., NEILSON, G.H., Nuclear Fusion 30 (1990) 111.

Figure Captions

Fig.1.
Schematic representation of control algorithm.

Fig.2.
Evolution of flux surfaces during TCV startup scenario. Radial and vertical coordinates are given in m. Radial profiles of poloidal flux, at the height of the magnetic axis, are also shown.

Fig.3.
Poloidal field coil currents (kA-turns) vs. time, during TCV startup evolution as shown in Fig.2. Coil identification shown in Fig.9.

Fig.4.
Flux errors on the preprogrammed boundary points as percentage of the total poloidal flux in the plasma, at t=60ms.

Fig.5.
RMS flux error, averaged over all boundary points, vs. time.

Fig.6.
Peapod-shaped plasma in TCV. Preprogrammed boundary points are indicated by plus signs. Radial and vertical coordinates in m.
a) $D=10^{-5}\ I$, $\Sigma\ I_s^2 = 0.570\ MA^2$, b) $D=3.0\times10^{-3}\ I$, $\Sigma\ I_s^2 = 0.266\ MA^2$.

Fig.7.
Vertical coordinate of magnetic axis (ZMAG) vs. time:
a) $\alpha_Z=20$, $\beta_Z=50$. b) $\alpha_Z=0$, $\beta_Z=50$. c) $\alpha_Z=0$, $\beta_Z=25$.

Fig.8.
Coil voltages (kV/turn) vs. time, corresponding to Fig.7a. Coil identification shown in Fig.9.

Fig.9.
Single-null divertor plasma in TCV. Radial and vertical coordinates in m. Cross indicates preprogrammed position of X-point.

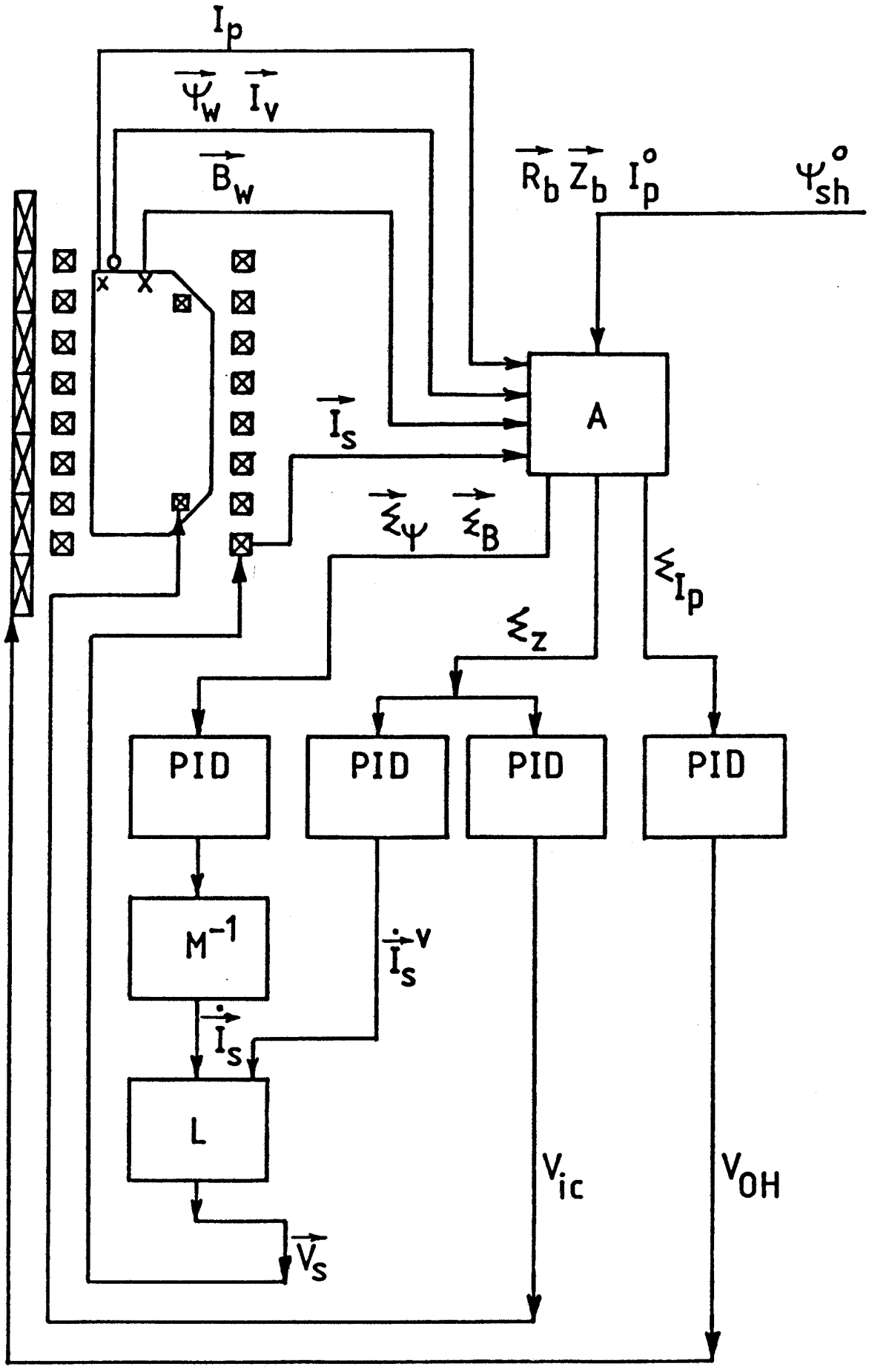

I_p
$\vec{\Psi}_w$ $\vec{I}_v$
$\vec{B}_w$
$\vec{R}_b$ $\vec{Z}_b$ I_p^o
Ψ_{sh}^o
A
$\vec{I}_s$
$\vec{\varepsilon}_\Psi$ $\vec{\varepsilon}_B$
ε_{I_p}
ε_z
PID
PID
PID
PID
M^{-1}
$\dot{\vec{I}}_s^v$
$\dot{\vec{I}}_s$
L
V_{ic}
V_{OH}
$\vec{V}_s$

FIG. 1

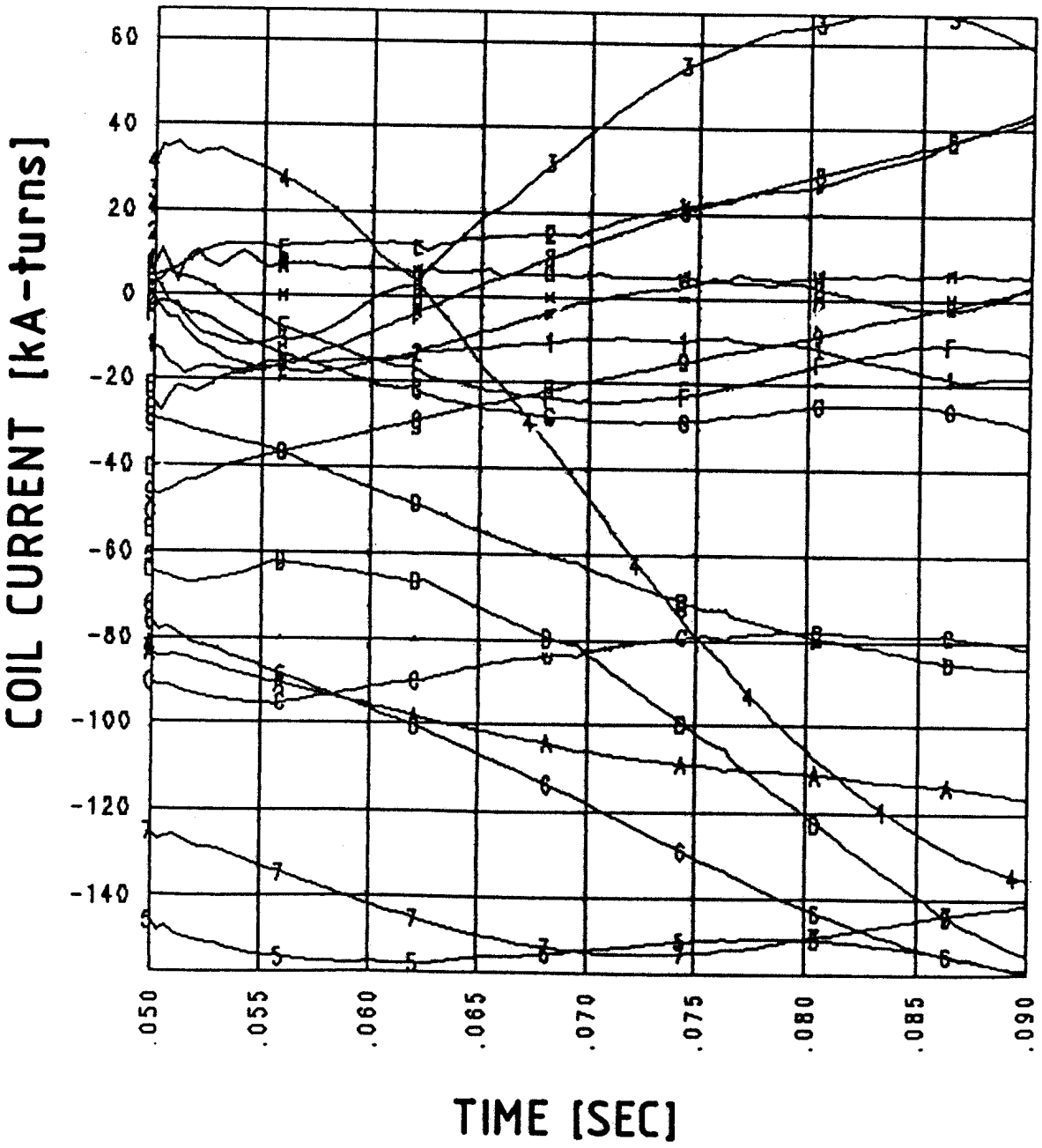

COIL CURRENT [kA-turns]
60
40
20
0
-20
-40
-60
-80
-100
-120
-140
.050
.055
.060
.065
.070
.075
.080
.085
.090
TIME [SEC]

FIG. 3

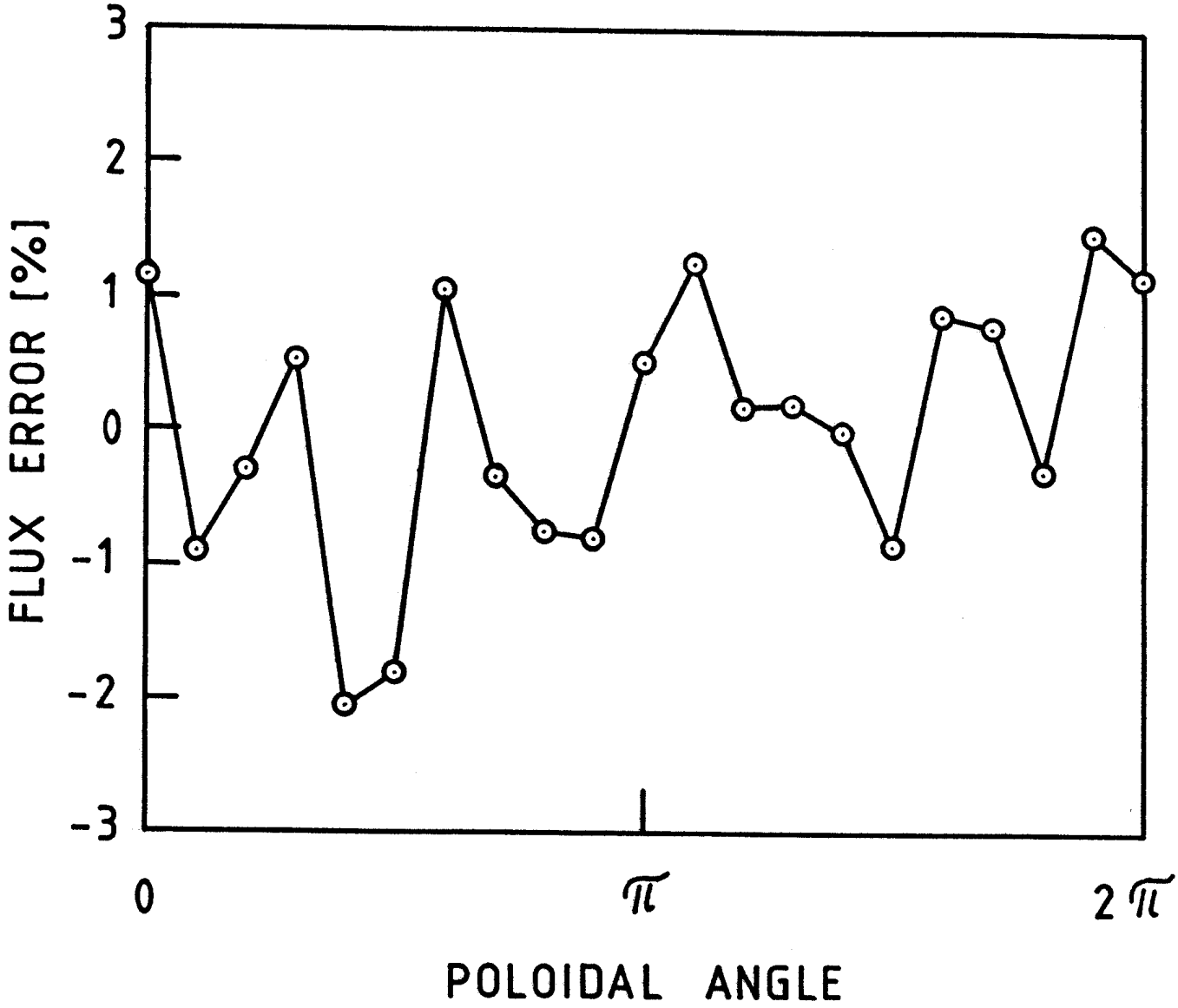


FIG. 4

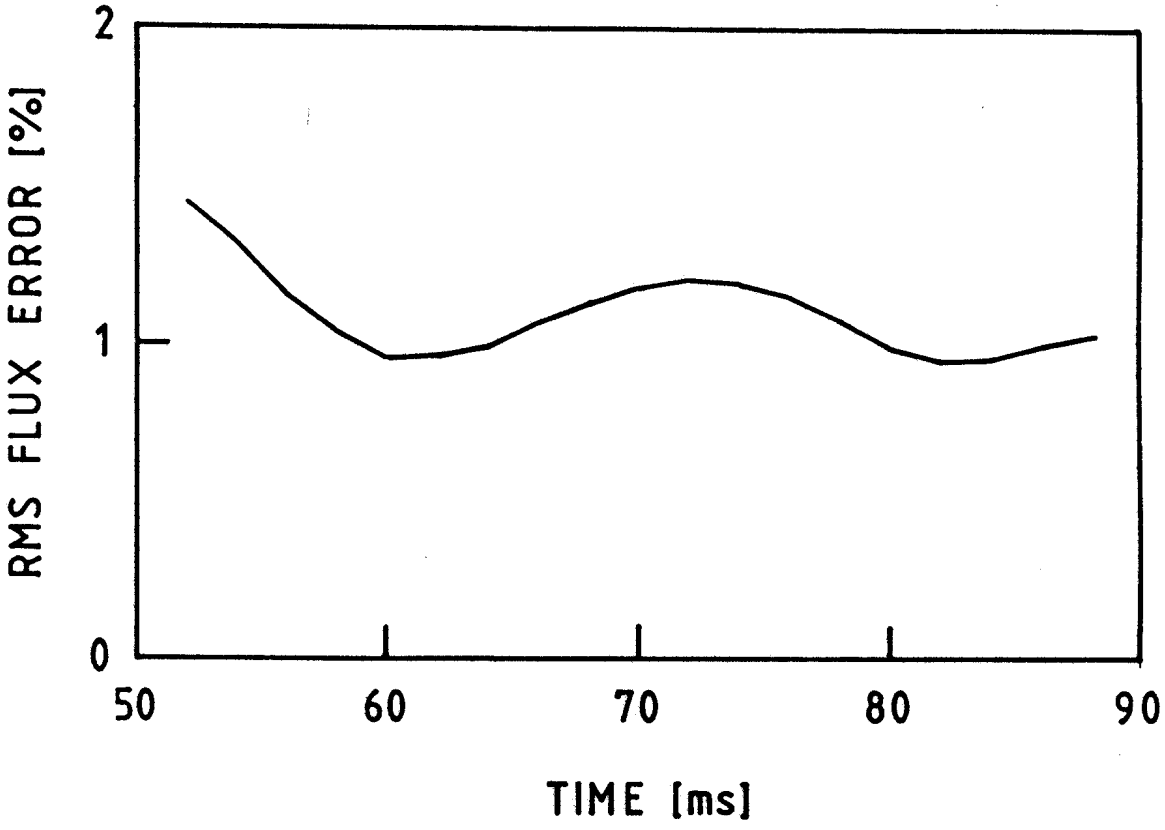


FIG. 5

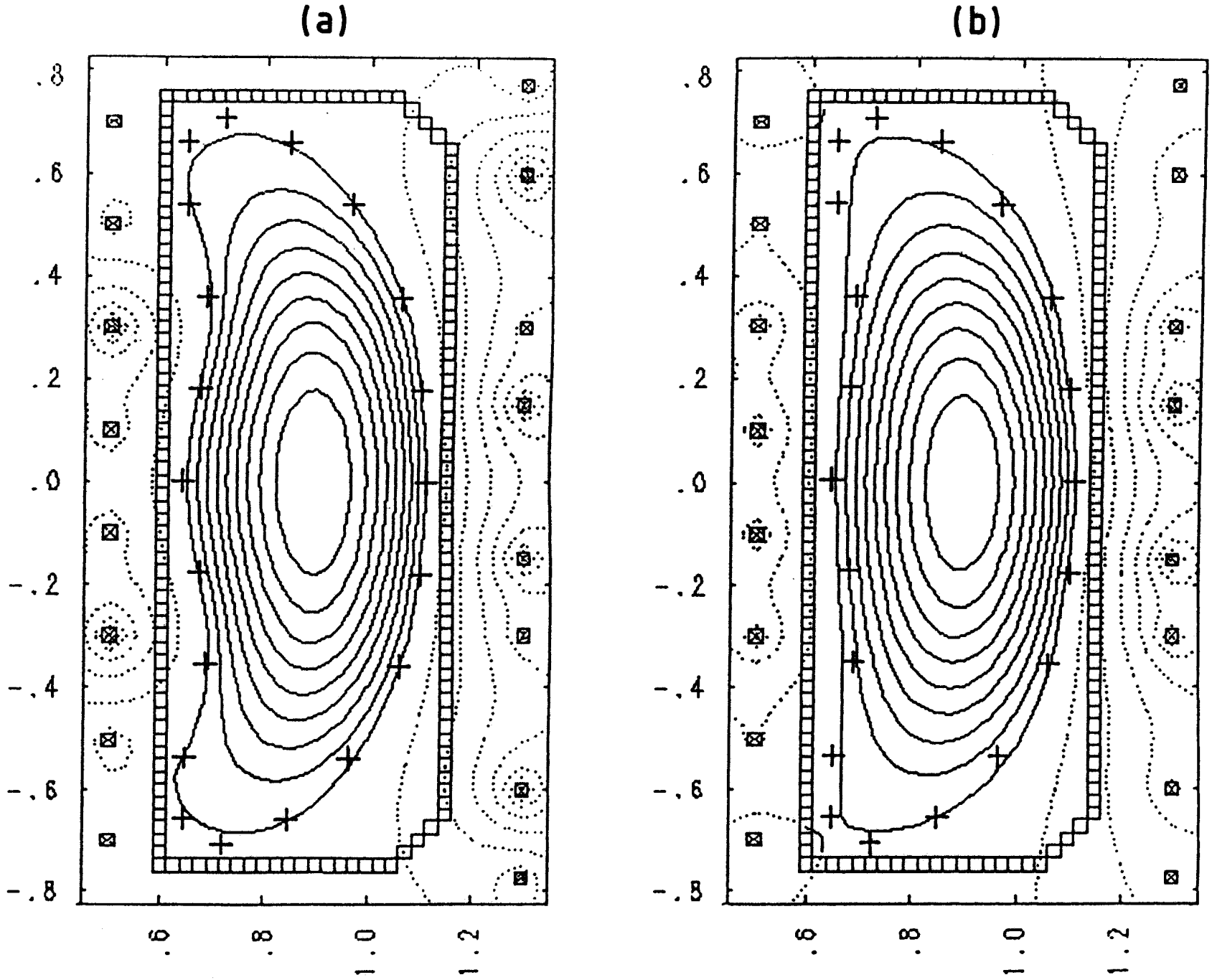
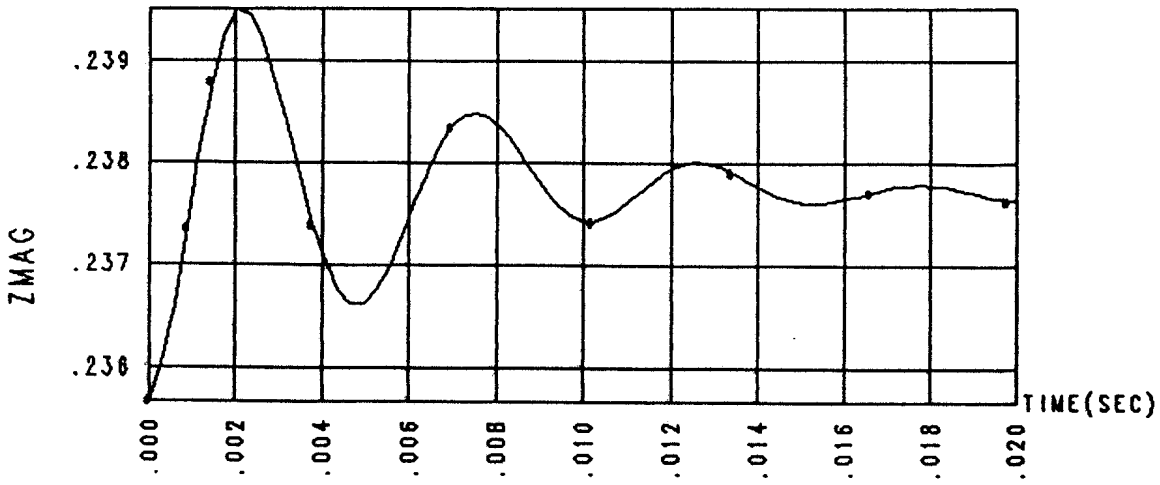
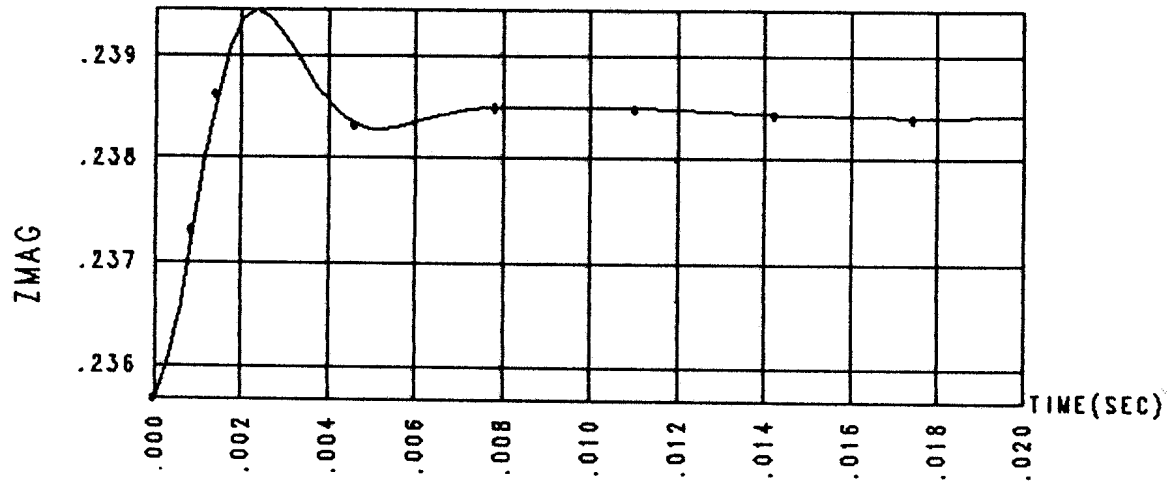


FIG. 6

(a)


ZMAG
.239
.238
.237
.236
.000
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
TIME(SEC)

(b)


ZMAG
.239
.238
.237
.236
.000
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
TIME(SEC)

(c)

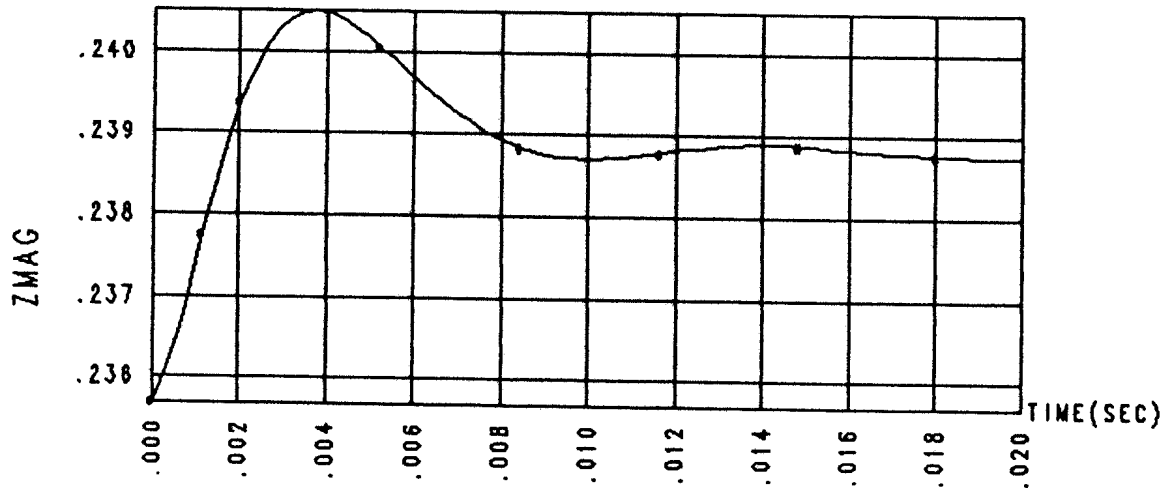

ZMAG
.240
.239
.238
.237
.236
.000
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
TIME(SEC)

FIG.7

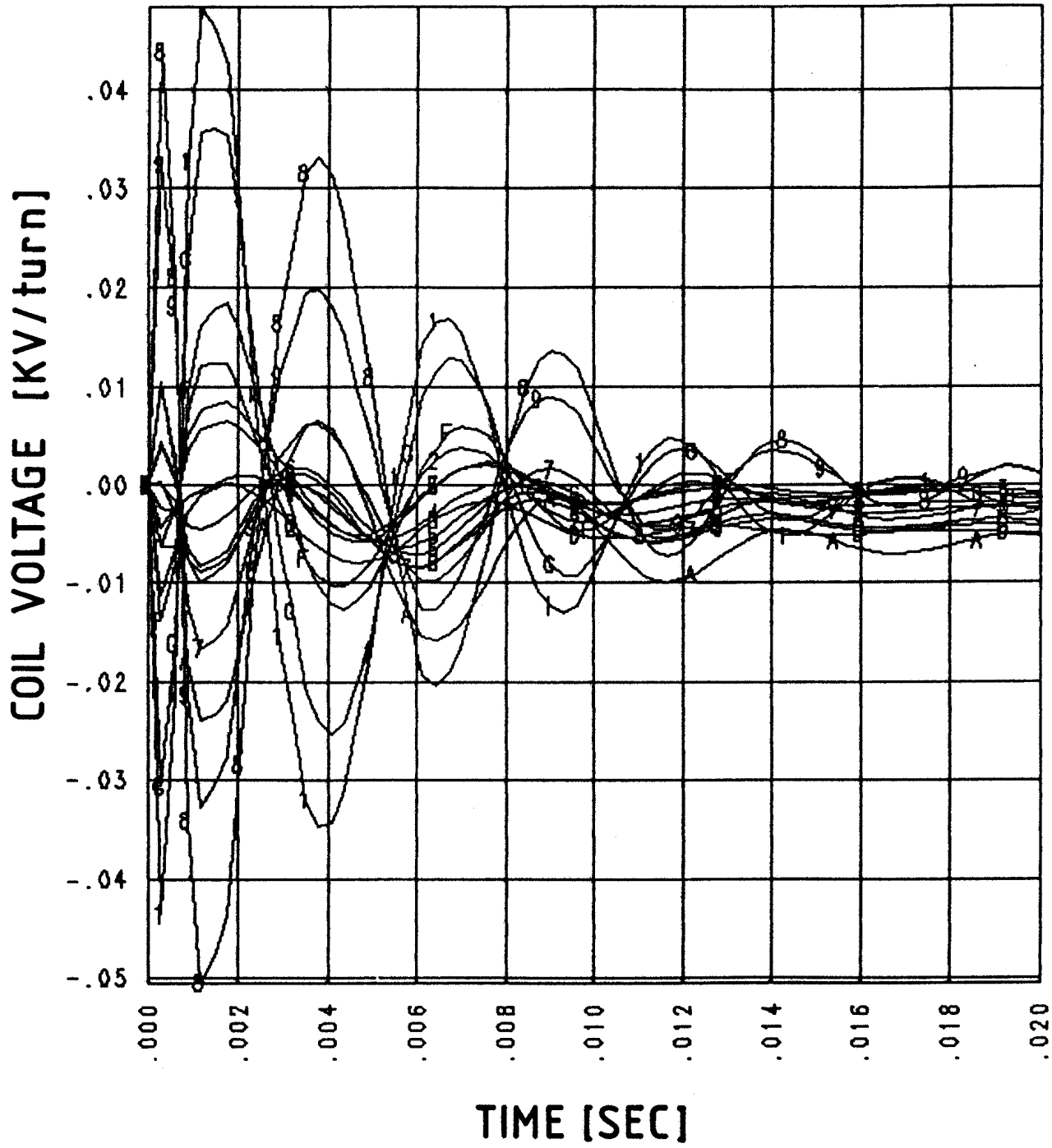

COIL VOLTAGE [KV/turn]
.04
.03
.02
.01
.00
-.01
-.02
-.03
-.04
-.05
.000
.002
.004
.006
.008
.010
.012
.014
.016
.018
.020
TIME [SEC]

FIG. 8

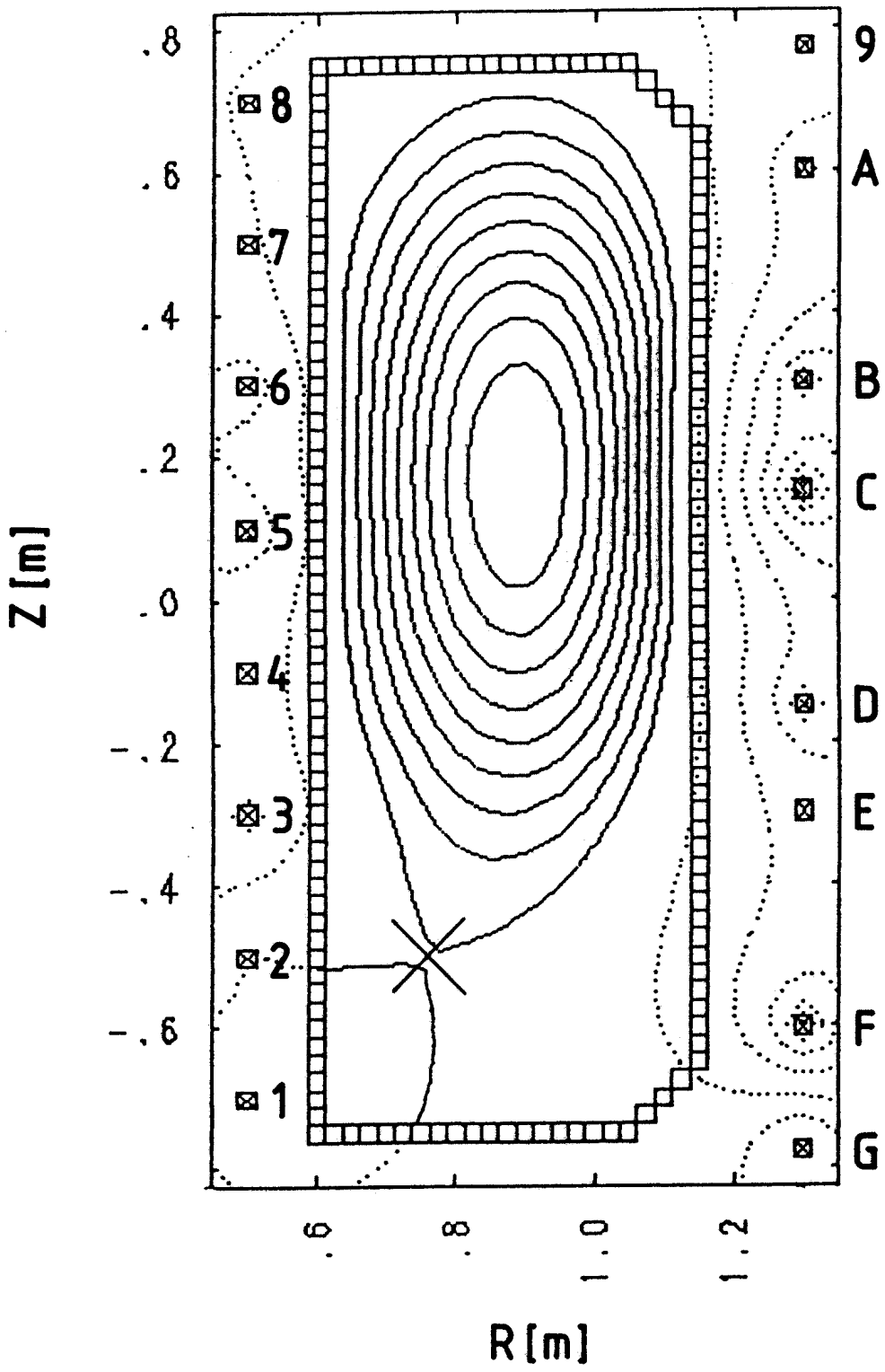


FIG. 9