Abstract—As a consequence of the continued increase in the worldwide demand of energy the need to achieve new sources of power has come out as a very important issue. In this sense, nuclear fusion has arisen to be a promising source of energy which has entailed an increasing interest in solving the different control problems existing in the nuclear fusion reactors such as tokamaks. The aim of this manuscript is to show how one of the existing codes that simulates the performance of tokamaks, the ASTRA code, can be integrated into the Matlab – Simulink tool in order to make more easier and comfortable the development of suitable controllers for tokamaks. As a demonstrative case study to show the feasibility and the goodness of the proposed ASTRA – Matlab integration, specifically, a simple PID controller for the loop voltage of a tokamak has been designed. The integration achieved represents an original and innovative issue in the tokamak control area and it provides new possibilities to the development and application of advanced control schemes to the standardized and widely extended ASTRA simulation code for tokamaks.

I. INTRODUCTION

The worldwide increasing demand of energy has highlighted the need to develop new sources of power. In this way, nuclear fusion has revealed to be a promising resource due to its several advantages, such as the abundance of fusion fuel, the safety of fusion process or the no harmful products resulting from the fusion reaction. One possible approach for the use of nuclear fusion as a source of power in order to supply the current demand is based in confine the plasma using electromagnetic forces, which is known as magnetic confinement. Nowadays, the most promising magnetic confinement system is the Tokamak (see [1]).

The Tokamak is a device characterized by a toroidal symmetry which was first developed in Russia in the 1960s (see Fig. 1). Due to the need of improving the performance of these devices, the interest in finding solutions for different control problems related to nuclear fusion that appear in Tokamaks has been increasing in the last years.

In this paper the application possibilities and utilities of the integration of the ASTRA code within the Matlab – Simulink tool are explained. With this purpose the achieved ASTRA – Matlab integration will be described and the particular case study of a PID controller, tuned using well – known Ziegler – Nichols tuning rules, will be presented since the use of these controllers is widely extended in industrial control systems (see [2]).

This paper is organized as follows. In Section II the ASTRA code used for modelling the Tokamak performance is presented and the equations and formulae used for constructing the model in this code are stated. In Section III the proposed ASTRA – Matlab integration is described. In Section IV a revision of existing PID theory is given and Ziegler – Nichols tuning rules are exposed followed by some simulations obtained from the combination of the implemented ASTRA – Matlab integration with a traditional PID control scheme designed according to those rules. Finally, in Section V, some concluding remarks are presented.

II. ASTRA CODE

A. General Description of the ASTRA Code

ASTRA (Automatic System for Transport Analysis) is a...
standard tool for generating computer codes to solve transport problems in magnetically confined plasmas (see [3]). It is a useful tool for the study of transport mechanisms in reactor-oriented facilities of Tokamaks. ASTRA solves coupled time-dependent 1-D transport equations for particles, heat and current and 2-D MHD (Magneto-hydrodynamic) equilibrium self consistently with realistic Tokamak geometry (see [4]).

In order to reach higher effectiveness, the flexibility provided by ASTRA allows the user to customize the code. This flexibility is built on the wide choice of standard relationships, functions and subroutines representing various transport coefficients, equilibrium solvers, methods of auxiliary heating (e.g. NBI) and other physical processes and data treatment in the Tokamak plasma. Another interesting characteristic of ASTRA is that it generates interactive codes. This means that not only can the user observe the time evolution of plasma parameters during the program execution, but he can also interrupt the execution or change the data presentation and control parameters influencing the course of modelling (see [3]).

Therefore, the ASTRA code is considered a transport code with a flexible programming system capable of creating numerical codes for predictive or interpretative transport modelling for stability analysis and for processing experimental data.

### B. ASTRA Background Equations and Formulae

In the ASTRA code, the magnetic system is considered toroidally symmetric and two coordinate systems are used: on the one hand, a cylindrical coordinate system \((r, \theta, z)\) with the polar axis coinciding with the major axis of the torus and, on the other, a coordinate system \((a, \theta, \zeta)\) associated to the magnetic geometry of the tokamak where \(a\) denotes the radial variable which is an arbitrary label of a magnetic flux surface, \(\theta\) is the poloidal angle and the toroidal angle is chosen \(\zeta = -\varphi\) (see [3]). Considering the local flux \(g\) expressed by (1) and the function of the magnetic surface \(F(a)\) which satisfies the diffusion equation given by (2), two functions of a single argument \(a\) defined by (3) and (4) have to be introduced.

\[
g(a, \theta) = F(a)\psi(a, \theta) - \mathcal{D}(a, \theta)\nabla F(a) \tag{1}
\]

\[
\frac{\partial F}{\partial t} = \frac{\partial}{\partial a} \left( \frac{\partial}{\partial a} \left( (\nabla V)^2 \frac{\partial F}{\partial a} \right) - F(\nabla V \cdot \nabla ) \right) + S(a) \tag{2}
\]

\[
\mathcal{D}(a) = \frac{\langle (\nabla a)^2 \rangle}{\langle (\nabla V)^2 \rangle} \tag{3}
\]

\[
v(a) = \frac{\langle \nabla a \cdot \nabla \rangle}{\langle \nabla V \rangle} \tag{4}
\]

Assuming those definitions independent of the choice of the magnetic surface label \(a\) it is possible to rewrite (2) in the form used in ASTRA given by (5) and also the total flux and the average flux density on a magnetic surface by (6) and (7) respectively.

\[
\frac{\partial F}{\partial t} = \frac{\partial a}{\partial V} \left[ \frac{\partial V}{\partial a} \langle (\nabla a)^2 \rangle \left( D \frac{\partial F}{\partial a} - \frac{\langle (\nabla V)^2 \rangle}{\langle (\nabla a)^2 \rangle} \right) \right] + S(a) \tag{5}
\]

\[
\Gamma(a) = \frac{\partial V}{\partial a} \left( \frac{\langle (\nabla a)^2 \rangle}{\langle (\nabla V)^2 \rangle} (D \frac{\partial F}{\partial a} - \frac{\langle (\nabla V)^2 \rangle}{\langle (\nabla a)^2 \rangle} \right) \tag{6}
\]

The magnetic field can be obtained by (8) and the current density by (9), where \(\Psi\) and \(J\) are defined by (10) and (11) respectively, being \(R_0\) the distance from the axis of the torus to a fixed point in the plasma and \(B_0\) the vacuum magnetic field at the point where \(r = R_0\).

\[
B = \nabla \zeta + \frac{1}{2\pi} \left[ \nabla \Psi \times \nabla \zeta \right] \tag{8}
\]

\[
j = -\frac{\nabla \zeta}{2\pi \mu_0} \cdot \nabla \Psi + \frac{1}{\mu_0} \left[ \nabla I \times \nabla \zeta \right] \tag{9}
\]

\[
\psi = -\mathcal{U} = -\frac{1}{4\pi} \int B \cdot \nabla \vec{a}^3 x = \int B \cdot dS_0 \tag{10}
\]
\[ I = R_0 B_0 - \frac{\mu_0}{2\pi} \int_{S_0} j \cdot dS \]  

(11)

Although functions \( I \) and \( \Psi \) are surface functions, which means that they depend on space coordinates through the variable \( a \), since they describe an evolving plasma they also depend on time, that is why each of them can be used as radial coordinate instead of \( a \) (see [3]). Once at this point, it is convenient to introduce another two surface functions defined by the following equations: toroidal magnetic flux \( \Phi \) and the effective minor radius \( \rho \).

\[ \Phi = \int_{S_c} B \cdot dS = \frac{1}{2\pi} \int_{r_0}^{r_1} d^3 x \]  

(12)

\[ \rho = \sqrt{\left( \frac{\Phi}{\pi B_0} \right)^2} \]  

(13)

The plasma equilibrium in a tokamak is determined by the Grad – Shafranov equation (14), where \( \rho = \rho(\rho, \theta) \) represents the plasma pressure with the contribution of all plasma species and \( I \) is the diamagnetic current previously defined in (11).

\[ \Delta \nabla^2 \Psi = r^2 \frac{\nabla^2 \Psi}{r^2} = -4\pi^2 \left( \frac{\mu_0}{\rho^2} \frac{\partial \rho}{\partial \Psi} + I \frac{\partial I}{\partial \Psi} \right) \]  

(14)

The ASTRA code uses an special notation with the aim of simplifying the notation which is explained in detail in [3]. It also uses the transport equations shown in Table I, that may to be expressed in terms of thermodynamic forces taken as derivatives with respect to \( \rho \), which makes possible to write the equilibrium equation (14) in terms of the functions provided by transport equations as:

\[ \Delta \Psi = 2\pi \mu_0 R_0 \left( \frac{\mu_0}{B^2/B_0} \left( J_1 + R_0 B_0 \frac{\partial B_0}{\partial \rho} \right) - \frac{r^2}{B_0 R_0 \rho} \frac{\partial \rho}{\partial \rho} \right) \]  

(15)

C. ASTRA definition for the loop voltage

The toroidal loop voltage, which can be understood as the quantity measured by a fixed toroidally symmetric loop of a constant major radius \( r_i = (r_i, z_i) \), can be defined by (16), where \( u_\psi \) represents the local velocity of the constant flux surface (see [3]).

\[ U_{\text{tor}} = \frac{\partial \psi}{\partial t} \bigg|_{r_i} = -u_\psi \cdot \nabla \psi \bigg|_{r_i} \]  

(16)

Considering that \( U_{\text{tor}} \) can be rewritten as shown in (17), it can be seen that, due to the first term on the right hand side, the \( U_{\text{tor}} \) is related to an average longitudinal electric field \( E_l \) on the magnetic surface \( \Phi \) making possible to define \( U_\parallel \) by (18), where \( G_3 \) concerns to the special notation of ASTRA detailed in [3].

\[ U_{\text{tor}} = \frac{\partial \psi}{\partial t} \bigg|_{r_i} = \frac{\partial \psi}{\partial t} \bigg|_{r_i} = \varphi \frac{\partial \psi}{\partial t} \bigg|_{r_i} + \mu \frac{\partial \Phi}{\partial t} \bigg|_{r_i} \]  

(17)

\[ U_\parallel = 2\pi R_0 E_\parallel = JG_3 \frac{\partial \psi}{\partial t} \bigg|_{r_i} = JG_3 \left( \frac{\partial \psi}{\partial t} \bigg|_{r_i} - \pi \rho^2 \mu B_0 \right) \]  

(18)

Moreover, taking into account that for ideally conducting plasmas the average longitudinal electric field on the magnetic surface is zero and that (18) describes freezing of the fluxes \( \psi \) and \( \Phi \) in one another, (18) can be turn into (19). All those considerations lead to the conclusion that the poloidal flux \( \psi \) moves through the toroidal flux \( \Phi \) only to its resistive dissipation within the flux surface measured by \( U_\parallel \). In other words, \( U_\parallel \) represents the diffusive flow of the poloidal flux \( \psi \) through the toroidal flux \( \Phi \).

\[ \mu(u_\psi - u_{\Phi}) \frac{V}{4\pi^2 R_0} U_\parallel = 0 \]  

(19)

Using (18) another flux surface quantity \( U_{pl} \), which can be understood as seen in the coordinate system moving together with a \( \rho \) surface, can be defined as shown in (20). Form that expression it can be concluded that the relative motion of the \( \Phi \) surface with respect to the \( \rho \) surface is due to the variation of the external magnetic field \( B_0 \), which in most present day tokamaks can be considered as time-independent ( \( \dot{B}_0 = 0 \)). For this reason, flux surface labels \( \rho \) and \( \Phi \) are physically equivalent and \( U_{pl} \) is closed to \( U_\parallel \). Therefore, for practical applications the use of \( U_{pl} \) is more convenient having into account that in steady state it does not vary over the minor radius \( \rho \) (\( \dot{\Phi} = 0 \)) (see [3]).

\[ U_{pl} = \frac{\partial \psi}{\partial t} \bigg|_{\rho} = \frac{U_\parallel}{G_3} + 2\pi \rho \mu B_0 (u_\psi - u_{\Phi}) \nabla \rho \]  

(20)

Now, for the study of the second term on the right hand side in (17) other considerations have to be taken into account. Since that the pressure of the toroidal magnetic field in a tokamak is much higher than the plasma pressure, the toroidal flux can only be changed noticeably when the external magnetic fields vary. This variation causes plasma movement as a whole, but if any of these movements is made faster than characteristic transport times then the plasma is compressed adiabatically which leads to (21) where the “magnetic” variable \( \rho \) and the “geometric” variable \( V \) are related.

\[ -u_\rho \cdot \nabla \rho = \frac{\partial \rho}{\partial t} \bigg|_{r_i} = \frac{\partial \rho}{\partial V} \left( \frac{\partial V}{\partial t} \bigg|_{r_i} - \frac{\partial V}{\partial t} \bigg|_{\rho} \right) \]  

(21)

In the general case, using (17) and (21) the toroidal loop voltage can be expressed by (22), where the left hand side is
the loop voltage as measured in an experiment, the first term on the right hand gives the resistive component, the second one describes the evolution of the plasma surface due to a variation in plasma para- or dia-magnetism and the remaining two terms can be associated with the adiabatic compression in minor and major radii respectively. Therefore, the measured loop voltage $U_{\text{tor}}$ can be expressed in terms of the calculated quantity $U_{\text{pl}}$ by (23).

$$U_{\text{tor}} = \frac{1}{JG_3} U_{\text{pl}} + 2\pi \rho \mu B_0 \frac{\partial \rho}{\partial t} \left|_{V} \right. + \pi r^2 \mu B_0 + 2 \pi \rho \mu B_0 \frac{\partial V}{\partial t} \left|_{V} \right.$$

(22)

$$U_{\text{tor}} = U_{\text{pl}} + 2\pi \rho \mu B_0 \frac{\partial \rho}{\partial t} \left|_{V} \right. + \pi r^2 \mu B_0 \frac{\partial V}{\partial t} \left|_{V} \right.$$  \hspace{0.5cm} (23)

Finally, if $U_{\text{tor}}$ is evaluated at $\eta = r_B$, a point which belongs to the edge flux surface with a fixed volume $V = V_B$ due to the boundary conditions so that $\left(\frac{\partial V_B}{\partial \eta}\right) = 0$, (24) can be obtained, where the last term on the right hand side is quite small but nonzero because the flux inside the plasma changes with any change in the profiles of the plasma pressure or current density and it is provided by a solution of the Grad – Shafranov equation.

$$U_{\text{plB}} = U_{\text{tor}} \bigg|_{\eta = r_B} = U_{\text{pl}} (\rho_B) + 2\pi B_0 \rho_B \mu (\rho_B) \frac{\partial \rho_B}{\partial t} \bigg|_{\eta = r_B} + 2\pi B_0 \rho_B \mu (\rho_B) \frac{\partial V_B}{\partial t} \bigg|_{\eta = r_B}$$

(24)

III. ASTRA - MATLAB INTEGRATION

The final aim of this work is to develop different controllers for Tokamak reactors in a more convenient and efficient way as it has been indicated in the introduction. Keeping this target in mind, as a first step towards achieving this objective a widely used standard transport code, ASTRA, has been embedded into Matlab tool. With the purpose of achieving the ASTRA code integration into Matlab, it has been necessary to automate the ASTRA running although the possibility of the interaction with the user hasn’t been eliminated (see Fig.2). The ASTRA – Matlab integration achieved makes possible to control the loop voltage by controlling the plasma current at each step. The significance of this work lies in that there are already existing Matlab codes in the market lacking current profiles but with several Tokamak models. Therefore, the benefits of this work are twofold: on the one hand the control for the currents can be tested in Matlab via the Simulink toolbox and on the other hand an inner second closed loop may be implemented coupling ASTRA with another Tokamak model in order to control other variables as, for example, the vertical displacement of the plasma.

In order to show the possibilities of the integration proposed, a simple PID controller, which will be explained next, has been taken as a case study. The suitability of a PID controller choice for this demonstrative example has been based on the broad use of this kind of controllers in real plants. At this respect, a study of 2008 showed that “based on a survey of over eleven thousand controllers in the heavy industries, 97% of regulatory controllers utilize PID feedback” (see [4]).

IV. CASE STUDY: PID LOOP VOLTAGE CONTROL

A. Introduction to PIDs and Ziegler – Nichols Tuning Method

In this section, a short background of PID theory is given in order to make easier and more comfortable the reading of this manuscript even if PID (Proportional Integral Derivative) control strategies are well – known and their use is widely extended.

Although PID control is one of the earlier control strategies it is still very useful in real plants (see [2]). A PID controller is a generic control loop feedback mechanism which tries to correct the error between a measured process variable and a desired reference setpoint by calculating and applying a corrective action in order to adjust the process properly. The typical structure of a PID control can be expressed by equation (25) where $u(t)$ is the input signal to the plant model or the control signal, $e(t) = r(t) - y(t)$ is the error signal, $r(t)$ and $y(t)$ the reference and output signals respectively and $K_p$, $K_i$ and $K_d$ are the proportional, integral and derivative gains (see [6]).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$

(25)
Considering that the pure derivative action is never used it is possible to rewrite (25) in a more common way using the Laplace transformation representation as shown in (26), where $T_i$ and $T_d$ represent the integral and derivative time constants respectively, considering that $K_i = K_p / T_i$ and $K_d = K_p T_d$ , and $\alpha$ is the parameter which limits the derivative action avoiding the undesirable effects of the derivative action.

$$U(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{s T_d}{1 + \frac{T_d}{\alpha}} \right) E(s)$$

(26)

The Ziegler-Nichols tuning rules, which were introduced in 1942, provide a practical and systematic way of tuning PID loops so as to improve the performance and obtain PID parameters in an easier way (see [7]). The tuning rules relay on two different techniques, one of these tuning techniques is based in the use of open-loop measurements and the other one in the use of the closed-loop and an automated mode for the controller.

For the open-loop tuning technique the tuning formula is obtained when the plant model can be approximated by a first-order plus dead time as shown in (27). The open-loop technique is based on the study of the step response of the plant which results in an S-shaped curve with no overshoot called reaction curve (see Fig.3.a, see [8]). This S-shaped curve is characterized by three constants, the delay or dead time $L$, the time constant $T$ and the process gain $k$. Using this tuning technique by a trial-and-error process the parameters of the controller are computed as shown in Table II (see [9]).

$$G(s) = \frac{k}{1 + sT} e^{-sL}$$

(27)

In the closed-loop technique the controller is set in automatic mode, but with the integral and derivative actions shut off. In order to obtain suitable parameters for the controller, its gain is increased until a sustained oscillation occurs in the process variable (see Fig.3.b, see [8]). The period of those oscillations is called the ultimate period $P_u$ and the smallest controller gain that can cause such oscillations is called the ultimate gain $K_u$. Once those two parameters are measured the tuning parameters of the controller can be computed using expressions shown in Table II (see [9]).

### B. Design of the PID for the loop voltage

With the purpose of showing how the integration of the ASTRA code within the Matlab – Simulink tool can help in the design of controllers for Tokamaks a first attempt to design a PID for the loop voltage of a Tokamak has been made. For the tuning of the parameters of the controller, $K_p$, $K_i$ and $K_d$, the Ziegler – Nichols tuning rules have been used. First of all, the open loop tuning technique was tried but the S – shaped curve obtained shows some overshoot which obliges to discard this tuning technique. Therefore it’s necessary to apply the closed loop tuning technique for which the block diagram shown in Fig.4.a is used, where the MATLAB Function block contains the M-file which takes care of the automatic running of the ASTRA. In this case, the integral and derivative actions are maintained shut off, while the gain of the PID block is increased gradually and smoothly until the sustained oscillations shown in Fig.4.b are achieved. From that last figure the period of oscillation $P_u = 0.0023$ and the minimum gain $K_u = 10.5$ for which those oscillations are computed. After that, these values were refined in an experimental trial and error basis for our case study target, obtaining the definitive controller gains:

<table>
<thead>
<tr>
<th>Open-loop Technique</th>
<th>Close-loop Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_p$</td>
<td>$K_p$</td>
</tr>
<tr>
<td>$T_i$</td>
<td>$2L$</td>
</tr>
<tr>
<td>$T_d$</td>
<td>$0.5L$</td>
</tr>
</tbody>
</table>
Using those parameters for the PID and the block diagram shown in Fig.4.a, some simulations for the closed loop system were obtained as shown in Fig.4.c, where it can be observed the evolution of the loop voltage for the proposed reference tracking problem. In particular, it is seen how the controlled loop voltage reaches the desired reference value.

\[ K_p = 6.3 \quad K_i = 5478.3 \quad K_d = 0.0018 \]

V. CONCLUSIONS

Due to the current worldwide increasing demand of energy, the need of developing new sources of power has emerged as a relevant issue. In this way, fusion energy has arisen as a promising source of power which has lead to a increasing interest and higher efforts employed in finding solutions for the different control problems existing in present Tokamaks. With the purpose of finding suitable controllers, diverse models for the simulation of tokamak performance have been developed and studied since the first Tokamak devices where conceived in 1960s.

The aim of this paper has been to show the feasibility of the proposed ASTRA – Matlab integration in order to use it as a tool for the development of controllers for Tokamak reactors in an easy and unified way. The first step has been to embed the standardized ASTRA simulation code into the Matlab tool in order to achieve more suitable controllers for Tokamaks. With this purpose the ASTRA consecutive running has been automated which allows using the Simulink toolbox for testing different controllers. The use of that toolbox provides the user the ability to try and test different controllers in an easier and more convenient way with the final aim of achieving the development and application of advanced control schemes to the widely extended and standardized ASTRA code for Tokamaks.

In order to show the effectiveness of the proposed integration a demonstrative case study based on a traditional PID has been presented. Since this kind of controllers are well – known and their use is widely extended in industrial control systems, their election as a case study hasn’t been accidental but it exposes the reliability and feasibility of the proposed integration for future applications.

ACKNOWLEDGMENT

The authors are very grateful to UPV/EHU and MICINN (formerly MEC) by its support through research project GIU07/08, and projects DPI2006-01677 and DPI2006-00714, respectively. They are also grateful to the Basque Government by its partial support through the research projects S-PE07UN04 and S-PE08UN15. The authors also would like to thank Professor Pereverzev from the Max Planck Institute for having facilitated the use of ASTRA.

REFERENCES