A Formally Verified Plasma Vertical Position Control Algorithm

May Wu[1,2], Jessie Rosenberg[2], and Nathan Fulton[2]

[1] Massachusetts Institute of Technology
[2] MIT-IBM Watson AI Lab, IBM Research

Abstract. Tokamak fusion reactors generate energy by using a magnetic control system to confine hot plasma in a toroidal chamber. In large reactors, incorrect implementation of plasma stabilization algorithms can result in significant physical damage to the reactor. This paper explains how a combination of formal verification and numerical simulation can be used to analyze the safety of a vertical stabilization algorithm of a tokamak fusion reactor.

1 Introduction

Fusion reactors generate energy by capturing energy released when two atomic nuclei fuse together [1]. Fusion of atomic nuclei occurs when the nuclear force pulling the nuclei together exceeds the electrostatic force pushing them apart. Because the nuclear force only exceeds the electrostatic force over very short distances, fusion reactors must first strip away the electron clouds surrounding the nuclei. This is achieved by heating the fuel atoms to extremely high temperatures, resulting in a super hot and electrically charged ion cloud. To generate net positive energy, fusion reactors must first generate a plasma and then confine the plasma inside a reaction chamber. Confining the plasma requires careful control of its position, shape, and movement.

Tokamak reactors achieve this control objective by exploiting the charged nature of the plasma column. The charged plasma field is enclosed in a toroidal chamber, and magnetic coils are wrapped around the exterior of the chamber [8]. The plasma column in tokamak reactors is typically elongated vertically to increase fusion efficiency, but this results in a destabilizing force on the plasma. Vertical stabilization algorithms ensure that the plasma does not touch the top or bottom of the reaction chamber by controlling the vertical position of the elongated plasma. Vertical stabilization is one of the simplest but most important control problems in tokamak reactors.

The push for fusion reactors that produce more energy than they consume motivates the ongoing construction of very large tokamak reactors [39]. As tokamak reactors grow larger, safety interlocks for magnetic control algorithms become more important. Large reactors are extremely expensive, and improperly controlled plasma could permanently damage the reactor. Therefore, deploying an experimental control algorithm on a very large reactor requires extensive pre-validation. This need for extensive pre-validation slows down the deployment of

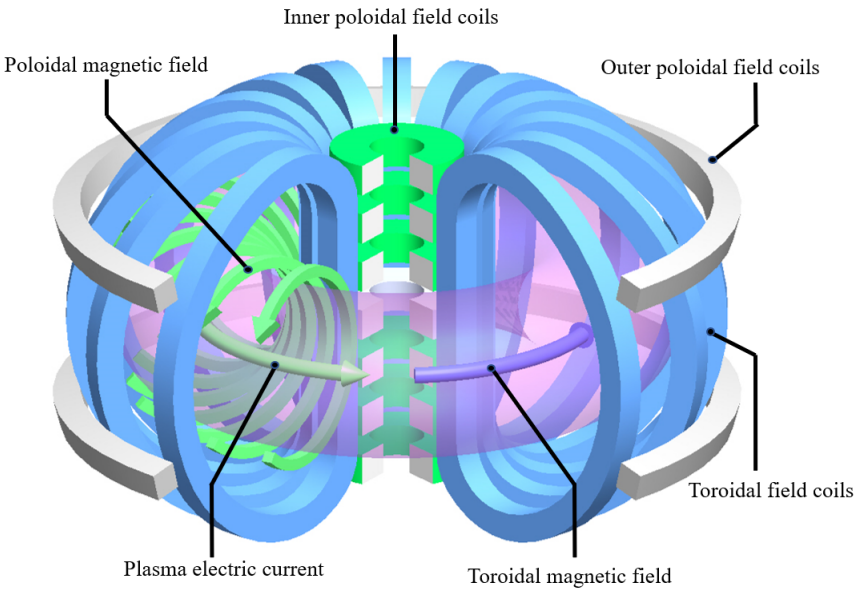

Fig. 1. A diagram of a Tokamak reactor with key features of the magnetic control system labeled, rendered by SolidWorks and based on a similar diagram in [8].

novel control algorithms, and poses a significant challenge when considering the use of control algorithms with black box machine learning components.

This paper considers the possibility of constructing software safety interlocks for the magnetic control systems of tokamak reactors. To illustrate the role the formal verification could play in supporting fusion research, we show how a hybrid systems theorem prover can be used to verify a vertical stabilization algorithm for an existing tokamak device. Because we are interested in enabling safe experimentation (e.g., via parameter tuning), we decompose the verification procedure into two phases: a first phase that reduces a parametric hybrid systems model to a non-parametric model, and a second phase in which a numerical ODE solver is used to check individual parameter choices for correctness.

The rest of this paper is organized as follows. Section 2 explains how tokamak fusion reactors work, precisely characterizes the vertical stabilization problem, and introduces the logic we use to specify and verify the vertical stabilization algorithm. Section 3 describes the T-15 vertical stabilization control algorithm that we verify. Section 4 presents our formalization of an established model for vertical plasma stabilization Section 5 discusses the details of our formal proof. Section 6 discusses related work, and Section 7 closes with a discussion of possible future work on formal methods for fusion reactors.

2 Background

This section introduces the vertical stabilization problem for tokamak reactors, assuming very little background in fusion or plasma control. We then introduce the logic and tool we use to prove the correctness of vertical stabilization.

2.1 Tokamak Reactors

Plasmas for nuclear fusion are composed of unbound electrons and ions at very high temperatures, and correspondingly high velocities. In order to achieve net fusion energy gain, the plasma must be confined and its shape carefully controlled. Magnetic confinement systems take advantage of the Lorentz force, wherein charged particles will spiral in helical paths around magnetic field lines. In the magnetic confinement design called a tokamak [32], shown in Figure 1 on page 2, toroidal coils wrap around the smaller circumference of a toroid and generate a magnetic field oriented along the larger circumference. This causes the charged particles to gyrate around that field, and induces confinement within the body of the toroid.

However, the magnetic field $\vec{B}$ from toroidal coils is nonuniform across the diameter of the coils. Along with the curvature of the toroid, this results in forces to create a vertical charge separation between electrons and ions. That in turn induces an electric field $\vec{E}$ in the vertical direction, which causes the plasma to move towards the outer boundary of the toroid due to $\vec{E} \times \vec{B}$ drift. In order to compensate for this external drift, a magnetic field in the poloidal direction (around the small circumference of the torus) must be added, to reshape the circumferential magnetic field lines into helices. The degree of helicity, or the ratio of the number of toroidal circuits to poloidal circuts, gives the tokamak safety factor q, a measure of the stability of the tokamak design. In tokamaks the poloidal magnetic field is generated by a current driven through the plasma itself.

This basic tokamak design, which generates a plasma with a circular cross-section, suffices for theoretical confinement of the plasma. However, in order to improve performance, it is necessary to vertically elongate the plasma cross-section. There are several reasons for this. A vertically elongated plasma results in a higher safety factor q, enabling stable operation with a higher plasma current for a given tokamak geometry and toroidal magnetic field strength. Additionally, a vertically elongated plasma allows the placement of a divertor [21], which increases efficiency by removing impurities and fusion byproducts from the plasma while the reactor is operating. Moreover, tokamaks with particular elongated plasma shapes can operate in a high-confinement regime, with confinement times that can be 2 to 3 times longer than the standard low-confinement regime [22].

Plasma shaping is performed using an additional set of poloidal magnetic field coils placed outside of the toroidal coils, as shown in Figure 1 on page 2. If the plasma is vertically centered between the outer poloidal coils, then its vertical position is at equilibrium when the currents on the upper and lower

coil are the same. However, if the plasma is displaced vertically, this creates an instability, and the plasma is rapidly accelerated toward the upper or lower wall of the chamber. Therefore, an active feedback loop and vertical stabilization is necessary to maintain confinement and the desired plasma shape [8, 3].

2.2 Vertical Stabilization

The vertical stabilization methodology we consider is that used in the T-15 Tokamak [24]. In this design, a pair of outer poloidal field coils are positioned between the toroidal coil and the vacuum vessel of the plasma. Operating as part of a feedback control system, these coils generate a magnetic field distribution that can compensate for the plasma's vertical instability and bring the system back to equilibrium.

We follow the stabilization model of Mitrishkin et al. [28], which utilizes a multiphase thyristor rectifier, a type of high power switching device, as the actuator. The controller is based on a linear combination of physically measurable elements: the plasma vertical displacement, and the current and voltage of the outer poloidal coils. An unstable linear model is used for the plasma, as the position displacements are assumed to be small relative to the major or minor radii of the tokamak, and we use a linear rectifier model as well.

Relatively simple models have been chosen to illustrate the fundamental dynamics of the system, as a starting point for future work on more complex models. In using a linear model for the plasma, we make the assumption that the displacement of the plasma's position from equilibrium is relatively small in comparison to the plasma's major and minor radii. This model holds in the case in which the plasma starts near the equilibrium position, as calculated by numerical simulation for example, and that the plasma position remains in closed-loop control. As stated in Mitrishkin et al. [28], this is a common simplification for plasma control systems models in order to make the dynamics models tractable. We justify this assumption by noting that one condition of successful closed-loop operation is that the plasma must not deviate far enough from equilibrium to escape the small-signal regime. Such simplified models have been used successfully in operational plasma control systems for many years [28]. Mitrishkin et al. reference such models in use for the tokamaks T-11 and Globus-M, in operation for more than 10 years [28]. They also reference background work leading to the selection of the linear first-order rectifier model from a Russian study [5] and communications with the ASDEX Upgrade tokamak team [9].

The system of three equations used to model the stabilization system in the T-15 tokamak are: the plasma model,

$$\tau_{plasma}\frac{\partial Z}{\partial t} - Z_{ref} = K_{plasma}I,$$

the model for the outer poloidal field coils,

$$L\frac{\partial I}{\partial t} + RI = U,$$

and the multiphase thyristor rectifier model,

$$\tau_{rectifier}\frac{\partial U}{\partial t} + U = K_{rectifier}V.$$

Here, τ_{plasma} and K_{plasma} are the time constant and gain of the plasma model, $\tau_{rectifier}$ and $K_{rectifier}$ are the time constant and gain of the rectifier model, I and U are the current and voltage of the outer poloidal field coils, and Z is the plasma vertical position. R is the resistance of the control coil, and L is its inductance. The controller output V is specified by the state feedback synthesis method [44], determined by controller gains K_0, K_1, K_2, and K_3. By applying gains to each of the state variables and the reference plasma vertical position (Z_{ref}), the controller output voltage is determined:

$$V \doteq K_0 Z_{ref} - K_1 Z - K_2 I - K_3 U.$$

2.3 Differential Dynamic Logic

This paper uses differential dynamic logic ($\mathsf{d}\mathcal{L}$) to formally specify and verify the correctness of a vertical stabilization algorithm. Differential dynamic logic is a logic for specifying and verifying properties about hybrid time dynamical systems [35, 33] and has been previously used to verify properties about adaptive cruise control [27], aircraft collision avoidance [18], and SCUBA dive computers [4]. The terms of $\mathsf{d}\mathcal{L}$ are those of real arithmetic:

$$\theta ::= x \mid r \mid \theta \cdot \theta \mid \theta + \theta \mid \theta - \theta \mid \frac{\theta}{\theta}$$

where $x \in \text{Vars}$ is a variable, $r \in \mathsf{R}$ is a real number, and $\frac{\theta}{\theta}$ defined whenever the denominator is not equal to zero.

Differential dynamic logic is used to reason about reachability properties of hybrid programs. Hybrid programs are generated by the grammar following:

$$\alpha, \beta ::= x := \theta \mid \alpha; \beta \mid \alpha \cup \beta \mid ?\varphi \mid \alpha^* \mid \mathbf{x}' = \theta \& \varphi$$

where x is a variable, θ is a term, and φ is a formula (the grammar and meaning of $\mathsf{d}\mathcal{L}$ formulas is reviewed below). The meaning of hybrid programs is defined over mappings from variables to real values (these mappings are called states):

- The assignment program $x := \theta$ assigns to the variable x the value θ, leaving all other variables in the state unchanged.
- The nondeterministic choice program $\alpha \cup \beta$ transitions from an initial starting state s_0 to any new state that can be reached by executing either α or β from s_0.
- The sequential composition program $\alpha; \beta$ first runs the program α and then, from the resulting states, runs the program β.
- The test/assert program $?\varphi$ terminates if φ is false, or continues executing without any change to the state if φ is true.

- The loop program α^* transitions from a state s_0 to any state that can be reached by executing α zero or more times. Looping is equivalent to a nondeterministic choice over a countable number of options:
  $\alpha^* \equiv \{\mathsf{NO\_OP} \cup \alpha \cup \alpha;\alpha \cup \alpha;\alpha;\alpha \cup \dots\}$
- The continuous evolution program $\mathbf{x}' = f\&\varphi$ follows the system of differential equations $\mathbf{x}' = f$ forward for any amount of time so long as φ remains true throughout.

The formulas of $\mathsf{d}\mathcal{L}$ form a first order logic for specifying reachability properties about hybrid programs. The grammar of $\mathsf{d}\mathcal{L}$ formulas follows:

$$\varphi, \psi ::= \varphi \wedge \psi \mid \neg\varphi \mid \forall x, \varphi \mid \exists x, \varphi \mid [\alpha]\varphi$$

where the meaning of $[\alpha]\varphi$ is that after every execution of α, φ is true.

Differential dynamic logic is implemented by the KeYmaera X theorem prover [13]. This paper uses KeYmaera X and, in particular, implements custom proof search scripts using the Bellerophon tactical programming language [12].

2.4 Example: Ball Suspended in Cylinder

We now use a highly simplified version of the plasma vertical stabilization problem to illustrate how $\mathsf{d}\mathcal{L}$ and hybrid programs are used to specify properties about dynamical systems. Consider a sphere suspended at the center of a cylinder. The cylinder's height is equal to its diameter, and the sphere's position may change. The controller must choose, at each control step, whether the first derivative of the sphere's position should increase or decrease. The controller's objective is to ensure that the ball does not touch the sides of the cylinder. This problem is illustrated in the diagram following:

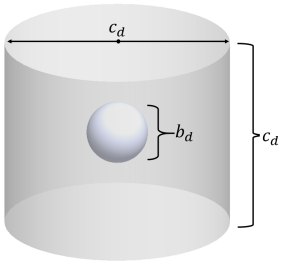

Fig. 2. Ball suspended in cylinder.

Denote by c_d the diameter and height of the cylinder, by c_r the radius of the cylinder, by $b_r = \frac{b_d}{2}$ the radius of the sphere, by b_p the offset of the ball's position from the midpoint of the cylinder, and by b_v the vertical velocity of the sphere. We will assume the ball is contained within the cylinder, so b_r is significantly smaller than c_r. The control program must choose, at each control step, a value

of $\frac{\partial b_p}{\partial t}$ that prevents the ball from touching the sides of the cylinder. The fact that that sphere never touches the cylinder's boundaries is expressible using the dℒ formula:

$$A > 0 \land B > 0 \land T > 0 \land b_p < c_r - b_r \rightarrow [\text{model}] b_p < c_r - b_r$$

where:

$$\begin{aligned} \text{model} &\equiv \{\text{ctrl}; t := 0; \text{plant}\}^* \\ \text{ctrl} &\equiv \text{ctrl}_A \cup b_v := 0 \cup \text{ctrl}_B \\ \text{ctrl}_A &\equiv ?b_p + AT < c_r - b_r; b_v := A \\ \text{ctrl}_B &\equiv ?b_p - BT < c_r - b_r; b_v := -B \\ \text{plant} &\equiv \{b_p' = b_v, t' = 1 \land t \leq T\} \end{aligned}$$

This simple example demonstrates how the dynamics of a moving object can be modeled in dℒ by referring to its offset from a fixed reference point (in this case, the center of the cylinder).

3 The T-15 Vertical Stabilization Controller

This section reviews the controller model for vertical stabilization of the T-15 tokamak via a multiphase thyristor rectifier. The model is substantially similar to the model presented in Mitrishkin et al. [28]. The system consists of four main components: the controller, the rectifier, the coil system, and the plasma of the tokamak. Figure 3 on page 8 illustrates the control scheme, wherein each component is represented as an individual control block.

The Controller block represents a feedback controller with four inputs and one output. The four inputs consist of the measured voltage output from the rectifier (U), the current in the outer poloidal field coils (I), the plasma vertical position (Z), and a vertical reference position Z_{ref}. As shown in Figure 3 on page 8, the controller amplifies the input signal Z_{ref} and the feedback signals U, I, and Z by controller gains $K0, \ldots, K3$. At the summing junction inside the controller, the resulting amplified Z_{ref} signal is added and the resulting amplified feedback signals are subtracted to produce the output signal V.

The controller gains $K_0, \ldots, K_3$ are tuned to achieve the desired performance. This paper considers an analysis that is parametric in one of these gains (K_0). Our analysis applies for an entire range of possible values of K_0 instead of, e.g., numerically checking each possible value. The goal of this partially parametric analysis is to lay the groundwork for a fully parametric analysis.

The control signal V is sent to the multiphase thyristor rectifier system of the tokamak, which is represented by the Rectifier block in Figure 3 on page 8. The multiphase thyristor rectifier system functions as the actuator for the vertical position control system. By regulating phases of thyristor bridges and utilizing a pulse-phase control circuit, the multiphase thyristor rectifier system outputs a regulated voltage (U) to control the tokamak's outer poloidal field

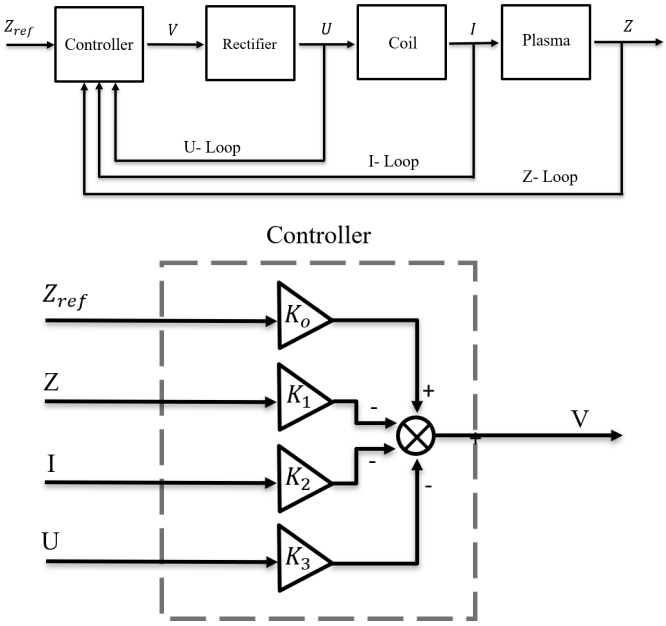

Fig. 3. System block diagrams for the vertical stabilization controller. **(a)** The tokamak is represented by a system block diagram. The overall system consists of a feedback controller, a multiphase thyristor rectifier, a control coil system, and tokamak plasma. These are represented, respectively, by the Controller, Rectifier, Coil, and Plasma system blocks. V is the output from the controller, U is the voltage measures at the output of the rectifier, I is the current measured from the output control coil system, and Z is the measure plasma vertical position. **(b)** The controller stabilizes the vertical plasma position by amplifying the input signal Z_{ref} and the feedback signals U, I, and Z by their respective gains $K0$ through $K3$. The resulting amplified Z_{ref} signal is added and the resulting amplified feedback signals are subtracted to produce the output signal V at the summing junction.

coil [28]. In our model, we generalized the behavior of the multiphase thyristor rectifier as a single system, instead of modeling the thyristor bridges separately. As discussed in Section 2.2, this assumption holds in the case of small deviations from equilibrium, where the system is assumed to be under closed-loop control. We use the rectifier time constant, $\tau_{rectifier}$ and the rectifier gain, $K_{rectifier}$ to describe the multiphase thyristor rectifier.

The voltage U is the drive signal for the outer poloidal field coil, which is represented by the Coil system block in the overall system block diagram. This voltage induces a current in the coils, modulated by its properties and environment. The output of the Coil system block is a current (I) regulated between an upper and lower operational limit.

Table 1. Meanings and physical constraints on variables occurring in the plasma vertical stabilization model.

Variable	Meaning	Constraints
Z_{ref}	Desired Plasma Position (Externally Specified)	$0 < Z_{min} < Z_{ref} < Z_{max}$
V	Controller Output	
K_0	Controller Gain Constant for Desired Plasma Position	> 0
K_1	Controller Gain Constant for Plasma Position	> 0
K_2	Controller Gain Constant for Current	> 0
K_3	Controller Gain Constant for Voltage	> 0
Z	Plasma Position	$0 < Z_{min} < Z < Z_{max}$
U	Rectifier Voltage	$0 < U_{min} < U < U_{max}$
I	Coil Current	$0 < I_{min} < I < I_{max}$
τ_{plasma}	Plasma Time Constant	> 0
$\tau_{rectifier}$	Rectifier Time Constant	> 0
K_{plasma}	Plasma Constant	> 0
$K_{rectifier}$	Rectifier Constant	> 0
R	Resistance of Coil	Device-specific
L	Inductance of Coil	Device-specific

Next we model the impact of the Coil system on the plasma itself. In addition to the outer poloidal field coil, a tokamak contains inner poloidal field coils, toroidal field coils, a primary transformer, ohmic heating coils, and other control coils depending on the tokamak design, as shown in Figure 1 on page 2. For the "Plasma" block in our KeYmaera X model, we abstract the response of the plasma and the feedback of these other control systems into one differential equation that describes the relationship between the plasma parameters and the final plasma vertical position, represented by the plasma time constant τ_{plasma} and the plasma gain K_{plasma}. This abstraction again holds in the stem remains in the small-signal regime near equilibrium and does not experience large deviations, in which case the complexities of the individual components would become apparent. For the T-15 tokamak, τ_{plasma} and K_{plasma} were estimated using a DINA plasma model [28, 23]. The output of the Plasma block arises from a sensor system that measures the plasma vertical position Z.

Table 1 presents a summary of the model variables, as well as their operational constraints. In addition to the variables already discussed, there are several constants that are used in our model. The resistance R and the inductance L of the outer poloidal control coil vary across different tokamak designs. For the T-15 tokamak that we study in this work, Mitrishkin et al. calculated the resistance and inductance of T-15 tokamak to be 0.09 Ω and 0.0042 H respectively.

4 The Model

This paper contributes a partially verified controller for T-15 vertical stabilization. The primary task is to verify that, for a particular choice of controller

gains, the plasma's position will not exceed a maximum threshold. The maximum thresholds represent the boundaries of model validity (in particular, the boundaries of model validity will be smaller than the boundaries of the tokamak chamber, so remaining within these bounds ensures that the plasma remains within the tokamak's chamber).

Table 1 reviews the variables used in our model. The goal of the control algorithm is to drive the plasma's position Z to the desired set=point Z_{ref}. The safety constraint proven in this paper is that the plasma's position Z stays below a maximum safe value Z_{max}.

```
1  ∃ K0, K1, K2, K3 .
2  τplasma = .0208 ∧ τrectifier = 0.0033 ∧ R = 0.09
3  ∧L = 0.0042 ∧ Krectifier = 2000
4  ∧0 = Zmin < z < Zmax < 0 ∧ z = i = u = 0
5  ∧ 0 ≤ Umin < Umax ∧ 0 ≤ Imin < Imax
6  →
7  [
8     Zref := *; ?Zmin ≤ Zref ∧ Zref < Zmax;
9     ?Z = Zref → z = −Kplasma/I ∧ I = U/R ∧ U = Krectifier/(K0 Zref − K1 Z − K2 I − K3 U) ;
10    {
11       Z' = Z/τplasma + Kplasma I/τplasma ,
12       I' = U/L − RI/L ,
13       U' = −U/τrectifier + Krectifier(K0 Zref − K1 Z − K2 I − K3 U)/τrectifier
14    }
15 ] ( zMin<z<zMax )
```

The model verified in this paper is listed above. The first nine lines state constraints on the physically realizable values for various parameters.

Line 1 existentially quantifies over the choice of gain $K_0, \ldots, K_3$. The constraint is eventually provided as input by the user and its correctness is checked using numerical simulation. The rest of the preconditions for the model, on Lines 2–4, express straight-forward constraints on the minimum and maximum value parameters and additionally set the values of constants to values appropriate for the T-15 reactor.

Line 8 models the choice of a new reference vertical position for the plasma. We assume that the reference position is provided as input from an external control module. In our model, we simply assert that the external module provides a reference value that is within the safety envelope.

Line 9 specifies the values that Z, I, and U should have when the plasma's vertical position is at the reference point. The condition presented is a conjunction formed by two simplified z and U equations and is derived via Ohm's Law letting $z = z_{ref}$ and $I = U/R$. The entire program terminates when the condition in Line 9 is false. This is a condition on the valid choices of $K_0, \ldots, K_3$;

i.e., this line should be understood as constraining the choice of gains for the controller, not the dynamics of the rest of the system.

Line 11 to Line 13 displays the model equations from Section 2.2 rewritten with the derivative term on the left hand side. Line 15 models the condition we want to ensure, which is to maintain z between z_{Min} and Z_{Max}.

5 The Analysis

Our model and corresponding analysis decompose the plasma's dynamics into two phases: an initialization phase and a steady-state phase. The graphs on page 12 visualize these two phases for the Z, I, and U variables.

Our proof is formalized in the KeYmaera X theorem prover. This section provides an intuition for how the geometry of the system relates to our formal derivations. We also comment on how the geometric intuitions underlying our informal description of the formal proof are encoded in the proof assistant. Before discussing our proofs, we recall a few inference rules from the proof calculus of $\mathsf{d}\mathcal{L}$.

5.1 The Proof Calculus of Differential Dynamic Logic

The primary proof techniques used for this construction are *differential cuts*, *differential induction*, and *differential weakening*. We briefly recall these proof techniques; our treatment is not exact, but contains enough formality that the reader will understand our proof. A full development of the proof calculus is presented in [36].

Differential cuts are similar to cuts in propositional logic [34]. Unlike propositional logic, $\mathsf{d}\mathcal{L}$ does not admit cut elimination; i.e., differential cuts strictly increase the deductive power of the logic. To prove that φ is an invariant of an ODE c restricted to the domain F, it suffices to find some G such that G is an invariant and, additionally, φ in an invariant of c restricted to the domain $F \wedge G$. As an inference rule,

$$\frac{\Gamma \vdash [c\&F]G \qquad \Gamma \vdash [c\&F \wedge G]\varphi}{\Gamma \vdash [c\&F]\varphi} DC$$

Differential induction proves invariants about differential equations by reasoning about the Lie derivatives of formulas. Geometrically, differential induction proves that a property φ is true throughout the flow of an ODE $\mathbf{x}' = \theta$ by establishing that, at every point in the vector field, derivatives point into the set φ. As an inference rule,

$$\frac{\Gamma, F \vdash \varphi \qquad \vdash F \rightarrow [\mathbf{x}' := \theta](\varphi')}{\Gamma \vdash [\mathbf{x}' = \theta\&F]\varphi} DI$$

Finally, differential weakening simply states that the domain constraint on an ODE is itself an invariant of the ODE. Stated as an inference rule,

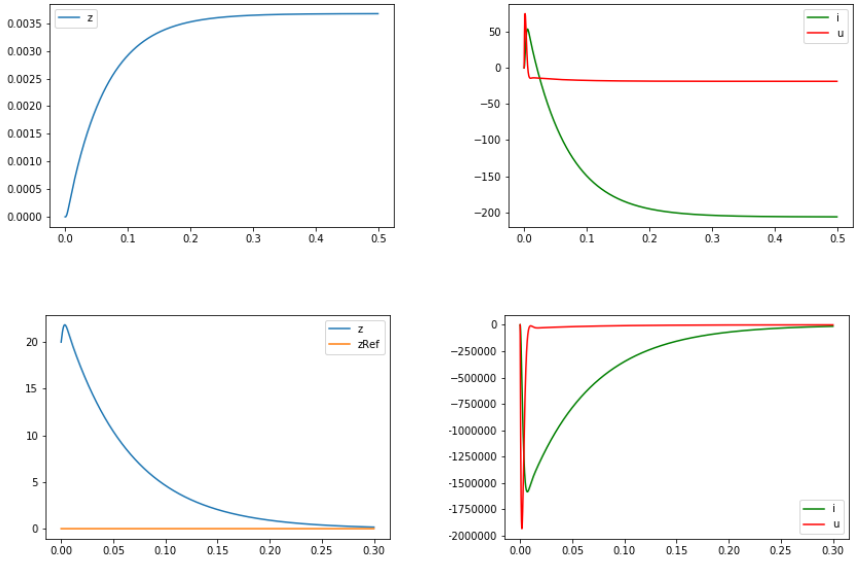

Fig. 4. Above: the dynamics of Z, I, and U as Z approaches Z_{ref} for parameter values $L = 0.0042$, $R = 0.09$, $Z_{ref} = 0.002$, $\tau_{plasma} = 0.0208$, $\tau_{rectifier} = 0.0033$, $K_{plasma} = 0.0000178$, $K_{rectifier} = 2000$, $V = 0.1$, and $\mathbf{K} = [100.0, 243.3287, 0.0032, 0.0013]$. These are the parameters that were used in Matriskin et al. [28]. Below: similar simulation but with $Z_{ref} < Z$. The data for these figures were generated using the ScyPy `scipy.integrate.odeint` function [42] and the figures were rendered by the `matplotlib` library [20]. Notice that the graph of Z is scaled differently from the graphs of I and U so that its magnitude is large enough to see.

$$\frac{\Gamma \vdash \varphi \qquad \Gamma_{\text{const}} \vdash F \rightarrow \varphi}{\Gamma \vdash [\mathbf{x}' = \theta \& F]\varphi} DW$$

where Γ_{const} is the subset of formulas in Γ that do not mention any of the variables occurring primed in $\mathbf{x'} = \theta$.[3]

In addition to these inference rules that allow reasoning about reachability properties about differential equations, we will denote by R the inference rule which proves φ whenever φ is in the modality-free fragment of $\mathsf{d}\mathcal{L}$. This fragment of $\mathsf{d}\mathcal{L}$ is the first-order logic over real-closed fields and it is decidable via quantifier elimination; the decidability result is due to Tarski [41] and its effective algorithm to Collins [7].

[3] There are also conditions on the occurrences of these variables in φ; however, in our case, those conditions are irrelevant because φ is simply a formula of first-order logic over real arithmetic and there are therefore no conditions. Platzer's uniform substitution calculus provides a full discussion of the static semantics of $\mathsf{d}\mathcal{L}$ [35].

5.2 Proof Structure

The vertical stabilization process, subject to appropriate control, has two phases: an *initialization phase* and a *steady state* phase. Our formal proof decomposes into these two phases. During the initialization phase, Z approaches Z_{ref} monotonically. Proving safety within this phase requires establishing that the system operates within a trapping region that confines Z below Z_{ref}. Eventually, the system reaches a steady state where $Z = Z_{ref}$ and where I and U are also invariant. We begin by describing our proof for the steady state phase.

5.3 Safety in the Steady State

The steady-state phase of the dynamics is the region where $Z = Z_{ref}$. Proving that the system is safe at the steady state is simple because the position of the plasma is invariant within this region and the values of the other system variables are constant. Therefore, the steady-state phase is fully characterized by the intersection of the z, i, and u nullclines.

The z nullcline where $z = Z_{ref}$ gives a constraint that relates the position of the plasma to the coil current (I):

$$\frac{\partial z}{\partial t} = 0 = \frac{Z_{ref}}{\tau_{plasma}} + \frac{K_{plasma} I}{\tau_{plasma}} \tag{1}$$

$$\frac{Z_{ref}}{\tau_{plasma}} = \frac{-K_{plasma} I}{\tau_{plasma}} \tag{2}$$

$$Z_{ref} = -K_{plasma} I \tag{3}$$

Similarly, the I nullcline is a constraint that relates the position of the plasma to the rectifier voltage:

$$\frac{\partial I}{\partial t} = 0 = \frac{U}{L} - \frac{RI}{L} \tag{4}$$

$$\frac{U}{L} = \frac{RI}{L} \tag{5}$$

$$I = \frac{U}{R} \tag{6}$$

Finally, the U nullcline constrains the choice of controller gains in terms of the current state of the system:

$$\frac{\partial U}{\partial t} = \frac{-U}{\tau_{rectifier}} + \frac{K_{rectifier} V}{\tau_{rectifier}} \tag{7}$$

$$U = \frac{1}{1 + K_3 K_{rectifier}} V \tag{8}$$

Notice that the control branch corresponding to this steady state asserts that each of these equations hold; we abbreviate the conjunctions of Equations (3), (6), and, (8) as SSA:

$$\begin{aligned}\mathsf{SSA} \equiv & U = \frac{1}{1 + K_3 K_{rectifier}} V \\ & \wedge I = \frac{U}{R} \\ & \wedge Z_{ref} = -K_{plasma} I\end{aligned}$$

Composing these nullclines gives a fixedpoint where $Z = Z_{ref}$. We introduce a new soundness-critical proof rule to KeYmaera X which allows us to reason about this fixed-point:

$$\frac{\Gamma, y = y_0 \vdash [\mathbf{x}' = \mathbf{f}(\mathbf{x}) \wedge y = y_0]P, \Delta \qquad \Gamma \vdash (\mathbf{x} = \mathbf{0})'}{\Gamma, y = y_0 \vdash [\mathbf{x}' = \mathbf{f}(\mathbf{x})]P, \Delta} DFP$$

where $\mathbf{x}$ and $\mathbf{f}$ are vectors. I.e., if $y = y_0$ initially and the derivative of each primed variable is 0 initially, then $y = y_0$ after any flow. Given this extension, we can then easily prove that the system is safe whenever it enters the fixed-point in a safe configuration:

$$\dfrac{\dfrac{\dfrac{*}{\Gamma_{\text{const}}, Z_{min} < z_0 < Z_{max} \vdash x = x_0 \rightarrow Z_{min} < Z < Z_{max}}\mathsf{R}}{\Gamma, Z_{min} < z_0 < Z_{max}, \mathsf{SSA} \vdash [\mathsf{plant} \wedge x = x_0] Z_{min} < Z < Z_{max}} DW}{\Gamma, Z_{min} < z_0 < Z_{max}, \mathsf{SSA} \vdash [\mathsf{plant}] Z_{min} < Z < Z_{max}} DFP$$

where Γ_{const} are the formulas in Γ that only mention variables that do not occur primed in the ODEs plant. In this case, that includes the assumption from our controller's assertion that $Z_{ref} < Z_{max}$. The result of the proof involves proving Δ_1, which is $\Gamma, \mathsf{SSA} \vdash [\mathsf{plant}] Z < Z_{ref}$.

5.4 Proving the System Remains Safe while Approaching the Reference Value

The remainder of the proof involves showing that the controller induces a trapping region that keeps Z above (or below) Z_{ref}, and furthermore that this trapping region is sufficient to ensure that $Z_{min} < Z < Z_{max}$.

Instead of establishing this property globally for an infinite set of possible controller gains, we instead observe that each variable at this point in the proof will have a specific value chosen by an experiment designer. The existential quantifiers on Line 1 will have already been instantiated with values known to satisfy Line 9. Therefore, any trusted numerical integrator can be used to simulate the full system dynamics out to the fixed point, at which point global reachability is established using the technique described in the previous section.

6 Related Work in Formal Methods

To the best of the authors' knowledge, formal methods tools have not been previously applied to the verification of plasma control algorithms in tokamak devices. Therefore, our related work discussion focuses on:

1. applications of formal methods in powerplant control systems,
2. other hybrid systems verification tools also capable of proving properties about control systems for fusion reactors.

6.1 Applications of Formal Methods in Power Plants and Similar Control Systems

The Cyber-Physical Systems and Formal Methods research communities have developed many approaches toward verifying industrial control systems. A thorough survey of the past half century of research in this area is hardly possible, so we focus instead on formal methods for industrial control systems that might be relevant to future work in formal methods for plasma control systems.

Formal methods have been used extensively for verification and validation of nuclear fission power plants. Wassyng and Lawford report on a large-scale verification project at the Darlington nulear power plant [43] and Németh et al. applied coloured petri nets to the verification of a primary-to-secondary leaking safety procedure [30]. Lahtinen's thesis provides a through survey of these and other formal methods efforts in the nuclear domain [26].

Most of the work on nuclear safety verification focuses on ensuring the safety of an already mature system. Therefore, much of the effort in these projects goes into modeling and verifying the dynamics of large-scale control systems such as programmable logic controllers. This paper focuses on the verification of controller design rather tha the verification of concrete implementations; therefore, work on verifying programmable logic controllers, such as the toolkits of Garcia et al. [17] and Pakonen et al. [31], are highly complementary to the work in this paper.

In addition to the obvious focus on nuclear fusion, another fundamental difference between our work and prior work on formal verification of control systems is the intended mode of use. Most applications of formal methods focus on mature domains where the fundamental design principles are well-understood and the primary problem is ensuring the correct implementation of a closed-loop system. Fusion, on the other hand, remains an unsolved problem. Perhaps the primary contribution of this paper is the simple suggestion that domain-specific and light-weight formal methods tools could be very useful to fusion researchers.

6.2 Hybrid Systems Case Studies and Tools

Although no other papers consider verification of fusion reactors, the dynamical system studied in this paper is mathematically similar to systems studied in other tools.

Reachability analysis tools for continuous and hybrid systems are capable of analyzing the linear system studied in this paper. For example, Althoff et al. introduce an approach toward reachability analysis of linear systems with uncertain parameters using matrix zonotopes and interval matrices [2]. The Flow* tool uses Taylor models for reachability analysis of nonlinear systems [6]. The dReal [16] and dReach [25] tools provide more automated analyses by framing hybrid systems reachability in terms of δ-decision procedures, and tools such as PHAver [10] and SpaceEx [11] are also well-suited to verification of hybrid systems. Automated analysis of both linear and nonlinear systems is also possible within KeYmaera X. The Pegasus tool introduced a set of nonlinear control benchmark problems, some of which exhibit dynamics similar to those studied in this paper [40].

Although our work used the KeYmaera X tool, fusion control systems might pose interesting verification challenges for other hybrid systems tools. Our use of KeYmaera X was motivated by two considerations. First, the authors' familiarity with this tool's implementation details enabled us to rapidly make the changes to the tool required to enable our analysis technique. Second, our plan for future work includes modeling more complex aspects of the fusion reactor's control systems and synthesizing safety interlocks that enable the use of reinforcement learning for these systems. KeYmaera X provides a method for composing verification results [29]. Because data-driven methodologies can play an important role in plasma design [37, 38], KeYmaera X's support for incorporating safety interlocks into learning systems [14] – especially in cases where some aspects of the control system are not captured by a first-principles model [15] or represented in explicitly modeled quantities [19] – provides another motivation for its use as a platform for this work.

7 Conclusion

Net positive fusion energy is an unsolved problem, and control design is a fundamental component of any fusion reactor. Experimenting with new controller designs is risky on large reactors where failures can cause millions of dollars in damage and even set back progress in the field. Our hope is that the formal methods community can contribute a set of robust safety interlocks of modern fusion reactors. This paper makes a first small step toward that vision by demonstrating that safety properties about vertical stabilization algorithms are possible to state and analyze using a combination of theorem proving and numerical integration.

Acknowledgments

We thank Cristina Rea, Darren Garnier, and other members of the MIT Plasma Science and Fusion Center for their helpful conversations. We also thank the anonymous reviewers for their helpful feedback.

Bibliography

[1] A.G.Peeters. *The Physics of Fusion Power*. 2008.

[2] Matthias Althoff, Bruce H. Krogh, and Olaf Stursberg. *Analyzing Reachability of Linear Dynamic Systems with Parametric Uncertainties*, pages 69–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[3] G. Ambrosino and R. Albanese. Magnetic control of plasma current, position, and shape in tokamaks: a survey or modeling and control approaches. *IEEE Control Systems Magazine*, 25(5):76–92, 2005.

[4] Viren Bajaj, Karim Elmaaroufi, Nathan Fulton, and André Platzer. Verifiably safe scuba diving using commodity sensors: Work-in-progress. In *Proceedings of the International Conference on Embedded Software Companion*, EMSOFT '19, New York, NY, USA, 2019. Association for Computing Machinery.

[5] A. A. Bulgakov. A new theory of controlled rectifiers. 1970 (in Russian).

[6] X. Chen and S. Sankaranarayanan. Decomposed reachability analysis for nonlinear systems. In *2016 IEEE Real-Time Systems Symposium (RTSS)*, pages 13–24, 2016.

[7] George E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *J. Symb. Comput.*, 12(3):299–328, 1991.

[8] Gianmaria De Tommasi. Plasma magnetic control in tokamak devices. *Journal of Fusion Energy*, 38(3):406–436, 2019.

[9] A. Kallenbach for the ASDEX Upgrade Team and the EUROfusion MST1 Team. Overview of asdex upgrade results. *Nuclear Fusion*, 57, 2017.

[10] Goran Frehse. PHAVer: algorithmic verification of hybrid systems past HyTech. *STTT*, 10(3):263–279, 2008.

[11] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. SpaceEx: Scalable verification of hybrid systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *23rd CAV, 2011*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.

[12] Nathan Fulton, Stefan Mitsch, Brandon Bohrer, and André Platzer. Bellerophon: Tactical theorem proving for hybrid systems. In Mauricio Ayala-Rincón and César A. Muñoz, editors, *Interactive Theorem Proving - 8th International Conference (ITP 2017)*, volume 10499 of *LNCS*, pages 207–224. Springer, 2017.

[13] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy P. Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538. Springer, 2015.

[14] Nathan Fulton and André Platzer. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In Sheila McIlraith and Kilian Weinberger, editors, *Proceedings of the Thirty-Second*

AAAI Conference on Artificial Intelligence (AAAI 2018), pages 6485–6492. AAAI Press, 2018.

[15] Nathan Fulton and André Platzer. Verifiably safe off-model reinforcement learning. In Tomás Vojnar and Lijun Zhang, editors, *Part I of the Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Held as Part of the European Joint Conferences on Theory and Practice of Software (TACAS 2019)*, volume 11427 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2019.

[16] Sicun Gao, Soonho Kong, and Edmund M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In Maria Paola Bonacina, editor, *24th International Conference on Automated Deduction (CADE-24)*, volume 7898 of *LNCS*, pages 208–214. Springer, 2013.

[17] Luis Garcia, Stefan Mitsch, and André Platzer. HyPLC: Hybrid programmable logic controller program translation for verification. In Linda Bushnell and Miroslav Pajic, editors, *ICCPS*, pages 47–56, 2019.

[18] Khalil Ghorbal, Jean-Baptiste Jeannin, Erik Zawadzki, André Platzer, Geoffrey J. Gordon, and Peter Capell. Hybrid theorem proving of aerospace systems: Applications and challenges. *J. Aerospace Inf. Sys.*, 11(10):702–713, 2014.

[19] Nathan Hunt, Nathan Fulton, Sara Magliacane, Nghia Hoang, Subhro Das, and Armando Solar-Lezama. Verifiably safe exploration for end-to-end reinforcement learning. *arXiv preprint arXiv:2007.01223*, 2020.

[20] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[21] G. Janeschitz, K. Borrass, G. Federici, Y. Igitkhanov, A. Kukushkin, H.D. Pacher, G.W. Pacher, and M. Sugihara. The iter divertor concept. *Journal of Nuclear Materials*, 220-222:73 – 88, 1995. Plasma-Surface Interactions in Controlled Fusion Devices.

[22] M Keilhacker. H-mode confinement in tokamaks. *Plasma Physics and Controlled Fusion*, 29(10A):1401–1413, oct 1987.

[23] R.R. Khayrutdinov and V.E. Lukash. Studies of plasma equilibrium and transport in a tokamak fusion device with the inverse-variable technique. *Journal of Computational Physics*, 109(2):193 – 201, 1993.

[24] GS Kirnev, VA Alkhimovich, OG Filatov, VV Ilin, DP Ivanov, PP Khvostenko, SV Trubnikov, AE Ugrovatov, EP Velikhov, VA Vershkov, et al. Superconducting tokamak t-15 upgrade. In *FT/P7-3, Proceedings of the 21st IAEA Fusion Energy Conference*, 2006.

[25] Soonho Kong, Sicun Gao, Wei Chen, and Edmund M. Clarke. dReach: δ-reachability analysis for hybrid systems. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015*, volume 9035 of *LNCS*, pages 200–205. Springer, 2015.

[26] Jussi Lahtinen. *Model Checking Large Nuclear Power Plant Safety System Designs: Dissertation.* PhD thesis, Aalto University, Finland, 2016. BA1606 SDA: SHP: SASUNE Nuclear Project code: 108550 165 p. + app. 75.

[27] Sarah M. Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In *FM 2011: Formal Methods - 17th International Symposium on Formal Methods, Limerick, Ireland, June 20-24, 2011. Proceedings*, pages 42–56, 2011.

[28] Yuri V. Mitrishkin, Evgeniia A. Pavlova, Evgenii A. Kuznetsov, and Kirill I. Gaydamaka. Continuous, saturation, and discontinuous tokamak plasma vertical position control systems. *Fusion Engineering and Design*, 108:35–47, 2016.

[29] Andreas Müller, Stefan Mitsch, Werner Retschitzegger, Wieland Schwinger, and André Platzer. Tactical contract composition for hybrid system component verification. *STTT*, 20(6):615–643, 2018. Special issue for selected papers from FASE'17.

[30] E. Németh, T. Bartha, Cs Fazekas, and K. M. Hangos. Verification of a primary-to-secondary leaking safety procedure in a nuclear power plant using coloured petri nets. May 2009.

[31] A. Pakonen, T. Mätäsniemi, J. Lahtinen, and T. Karhela. A toolset for model checking of plc software. In *2013 IEEE 18th Conference on Emerging Technologies Factory Automation (ETFA)*, pages 1–6, 2013.

[32] A. Pironti and M. Walker. Fusion, tokamaks, and plasma control: an introduction and tutorial. *IEEE Control Systems Magazine*, 25(5):30–43, 2005.

[33] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reas.*, 41(2):143–189, 2008.

[34] André Platzer. The complete proof theory of hybrid systems. In *LICS*, pages 541–550. IEEE, 2012.

[35] André Platzer. A uniform substitution calculus for differential dynamic logic. In *CADE*, 2015.

[36] André Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer, Cham, 2018.

[37] C Rea, R S Granetz, K Montes, R A Tinguely, N Eidietis, J M Hanson, and B Sammuli. Disruption prediction investigations using machine learning tools on DIII-d and alcator c-mod. *Plasma Physics and Controlled Fusion*, 60(8):084004, jun 2018.

[38] Cristina Rea and Robert S. Granetz. Exploratory machine learning studies for disruption prediction using large databases on diii-d. *Fusion Science and Technology*, 74(1-2):89–100, 2018.

[39] Y Shimomura, R Aymar, V Chuyanov, M Huguet, R Parker, et al. Iter overview. *Nuclear Fusion*, 39(9Y):1295, 1999.

[40] A. Sogokon, S. Mitsch, Y. K. Tan, K. Cordwell, and A. Platzer. Pegasus: A framework for sound continuous invariant generation. In Maurice ter Beek, Annabelle McIver, and José N. Oliviera, editors, *FM*, volume 11800 of *LNCS*, pages 138–157. Springer, 2019.

[41] Alfred Tarski. A decision method for elementary algebra and geometry. 1948.

[42] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua

Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[43] Alan Wassyng and Mark Lawford. Lessons learned from a successful implementation of formal methods in an industrial project. In Keijiro Araki, Stefania Gnesi, and Dino Mandrioli, editors, *FME 2003: Formal Methods*, pages 133–153, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[44] Robert L Williams, Douglas A Lawrence, et al. *Linear state-space control systems*. John Wiley & Sons, 2007.