Simplified Flip-Flop Gate Model for EEMI Injection

Luis Valbuena
University of New Mexico
Albuquerque NM, USA
lavalbuenar@unm.edu

Gregory L. Heileman
University of Arizona
Tucson AZ, USA
heileman@arizona.edu

Sameer Hemmady
University of New Mexico
Albuquerque NM, USA
shemmady@unm.edu

Edl Schamiloglu
University of New Mexico
Albuquerque NM, USA
edls@unm.edu

***Abstract*—We present a second order dynamical system to represent the behavior of a D flip-flop. We employ windowing functions and vector fields to replicate a characteristic found in [1],which resorts to switching. The model also takes into account metastable behavior, which can be exploited when studying software execution faults due to an undefined logical state. We conceived the noise injection to be additive noise targeting the transition between the stable equilibrium points. However, the model is flexible and many parameters can be changed to alter its behavior.**

***Index Terms*—Flip-Flop, modeling, EEMI.**

I. Introduction

The academic curiosity question of how the interaction of sensors/actuators with computing systems can render an adversarial vulnerability has evolved into practical relevance. There are various works where medical devices such as insulin injectors or pacemakers have been compromised to the point of reporting spurious levels of insulin, switching on/off the flow of insulin or producing shock commands [2]. Usually, these adversarial intrusions take advantage of the computing systems' ports of entry. It has also been reported that instructions in a cryptographic code can be skipped or duplicated by injecting clock glitches and underpowering the microcontroller [3]. However, there are no mathematical models describing how EEMI injection propagates through the electronic components or details about the specifics of the injection signal to cause certain instruction to fail at a particular point in time. Even though there are tractable models for inverters accounting for noise injection and it is possible to use them to build logic gates, like the `NOR` gate presented in [4], these models do not scale well when they are incorporated into hybrid modeling of a computing system. In this paper, we present a system of second-order differential equations that serves as our model of a flip-flop gate. This model is developed based on the ideas presented in [1] and vector fields from artificial potential fields. This is a normalized model both in time and space, and apart from being low dimensional and amenable for reachability analysis, employs windowing functions to account for metastability. This model does not require a hybrid approach for the transitioning of states as the windowing functions perform a smooth transition from each of the sinks present in the model.

This work supported by AFOSR Grant FA9550-15-1-0171.

II. Motivation

We have a 4-bit processor inside a simulation environment which can be tampered at any location inside the processor as well at any point in time [5]. However, we are interested in studying the propagation mechanism of EEMI injections in the entry ports of a processor to the location where it produces an exploitable fault; therefore, we are in need of dynamical models that interact with EEMI to affect the electronic components. From [6], we have derived dynamical models for logic gates that accounts for EEMI injection but we were lacking a model for components that store information, i.e., D flip-flops. Our starting point is [1] where there is a first-order flip-flop model gate, displayed in Fig. 1.

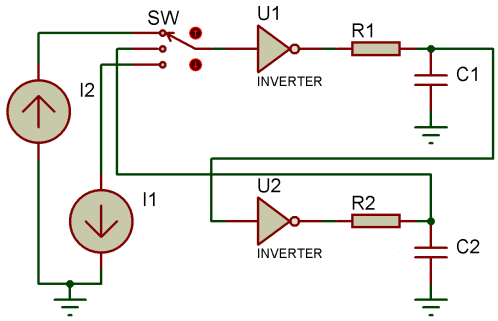


Fig. 1. First order flip-flop model. The `SW` component introduces a switching nature to the otherwise continuous dynamics.

This model is reported to generate the phase portrait displayed in Fig. 4 in [1], which suggests that a vector field can be proposed with the characteristics displayed on the mentioned figure, without resorting to switching techniques to achieve the same result. The remainder of this paper presents the mathematical elements required to build the flip-flop model, the properties that the model has, and simulation results.

III. Mathematical Formulation

The idea behind this formulation comes from locating the vector fields of two sinks at different locations in some space, similar to determining how the flow of a singularity is near a wall [7]: instead of dealing with the original problem, a virtual sink (source) of same intensity is located at the reflection of the original singularity with respect to the wall. The resulting vector field then has two sinks (sources) and also a streamline that has no component perpendicular to the wall, i.e., in our

case there are two stable equilibrium points and a saddle point, which resembles the features described in [1] where there are two equilibrium points that represent the logic states of the flip-flop gate as well as a saddle equilibrium point that represents the metastable state. However, trajectories near the equilibrium points do not behave like the ones observed on sinks; therefore, we select particular regions by using windowing functions and then conduct rotations on the vector field to match the phase portrait presented in [1].

The proposed bi-dimensional model for a flip-flop gate is presented in (1)

$$\begin{aligned}\dot{\mathbf{x}} =& L_{eqA}(\mathbf{x}) \begin{bmatrix} c\alpha & -s\alpha \\ s\alpha & c\alpha \end{bmatrix} (\mathbf{x}_{eqA} - \mathbf{x}) + \\ & L_{eqB}(\mathbf{x}) \begin{bmatrix} c\beta & -s\beta \\ s\beta & c\beta \end{bmatrix} (\mathbf{x}_{eqB} - \mathbf{x}) + \\ & \begin{bmatrix} K_1 \\ -K_2 \end{bmatrix} u(t) + \begin{bmatrix} K_3 \\ K_4 \end{bmatrix} \omega(t),\end{aligned} \tag{1}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ corresponds to the flip-flop states, $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$ are the locations for the equilibrium points that represent the logic states Q and $\bar{Q}$, and K_1, K_2, K_3 and K_4 are constants. A detailed explanation of the more complex terms in (1) is provided in the following subsections.

A. Intensity of the stable equilibrium points

Usually, a sink can be written as:

$$\mathbf{v}(\mathbf{x}, \mathbf{x_0}) = \frac{1}{\|\mathbf{x_0} - \mathbf{x}\|^n} (\mathbf{x_0} - \mathbf{x}), \tag{2}$$

for some equilibrium point $\mathbf{x_0}$ and some constant n. However, the scaling done by the norm is unbounded as $\mathbf{x} \to \mathbf{x_0}$; then, if (2) is part of a differential equation, the norm will turn into a stiff differential equation. Therefore, with the purpose of bounding the scaling of the vector pointing towards the equilibrium point, we employ functions $L_{eqI}(\mathbf{x})$ and $L_{eqI}(\mathbf{x})$(See (3)).

$$L_{eqI}(\mathbf{x}) = \frac{L_0}{1 + e^{k\|\mathbf{x}_{eqI} - \mathbf{x}\|^2 - d}}, \text{ for } I = \{A, B\}, \tag{3}$$

which is the logistic function. Note that the term $\|\mathbf{x}_{eqI} - \mathbf{x}\|^2$ makes $L_{eqI}(\mathbf{x})$ radially symmetric with respect to an axis located at $\mathbf{x}_{eqI}$. The constant d provides some offset from the equilibrium $\mathbf{x}_{eqI}$ and L_0 is some tunning constant. The value of k modifies how fast the logistic function varies, which allows us to set it big enough to make L_{eqI} vanish to zero when $\mathbf{x}$ is far away from the symmetry axis of the function, like for example, the other equilibrium point. Consider Fig. 2, where we have chosen a value for k big enough such that the value of $L_{eqA}(\mathbf{x})$ does not sabotage the value of $L_{eqB}(\mathbf{x})$ and vice versa. The addition of $L_{eqA}(\mathbf{x})$ and $L_{eqB}(\mathbf{x})$ tends to zero for values of $\mathbf{x}$ sufficiently far from both $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$. The intensity $L_{eqA}(\mathbf{x})$ only influences the vector field of the sink created at $\mathbf{x}_{eqA}$ and the same goes for $L_{eqB}(\mathbf{x})$ and $\mathbf{x}_{eqB}$.

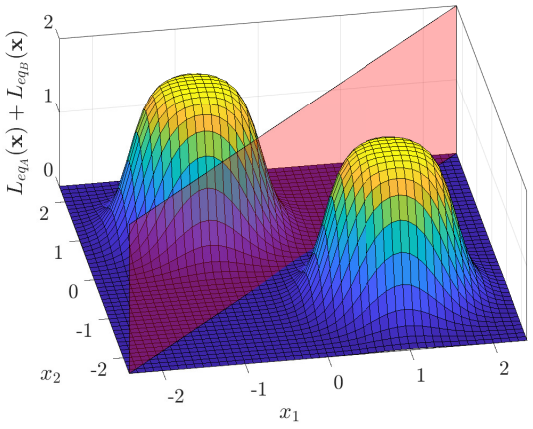


Fig. 2. Functions $L_{eqA}(\mathbf{x})$ and $L_{eqB}(\mathbf{x})$ added together where the value of k is chosen so the intensity of a vector field from one sink does not interfere with the other. The red translucent region represents the locus where the component of the added vector field does not have a component in the direction of the segment from $\mathbf{x}_{eqA}$ to $\mathbf{x}_{eqB}$.

B. Rotation matrices

Rotation matrices are full rank and preserves the norm; hence, the mapping they conduct keeps the properties of the domain, i.e., the stability properties of an equilibrium point are maintained on the range. The terms $c(\cdot)$ and $s(\cdot)$ in (1) represent the trigonometric functions $\cos(\cdot)$ and $\sin(\cdot)$. We define the constant rotation

$$\begin{bmatrix} {}_R x_1 - {}_R x_{eqI,1} \\ {}_R x_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} (\mathbf{x} - \mathbf{x}_0), \tag{4}$$

for $\mathbf{x}_0$ a point colinear to $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$, where ${}_R x_{eqI,2}$ equals zero for this rotation. The vector $[{}_R x_1 - {}_R x_{eqI,1}, {}_R x_2]^T$ is the argument of the functions defined below. The rotation angles α and β are defined as:

$$\begin{aligned}\alpha &= \frac{1}{2} g_A({}_R x_1 - {}_R x_{eqA,1})(\phi_A - \measuredangle(\mathbf{x}_{eqA} - \mathbf{x})), \\ \beta &= \frac{1}{2} g_B({}_R x_1 - {}_R x_{eqB,1})(\phi_B - \measuredangle(\mathbf{x}_{eqB} - \mathbf{x})),\end{aligned} \tag{5}$$

with g_A and g_B defined as

$$\begin{aligned} g_A &= \frac{1}{2}(1 + \tanh K_6({}_R x_1 - {}_R x_{eqA,1})), \\ g_B &= \frac{1}{2}(1 - \tanh K_6({}_R x_1 - {}_R x_{eqB,1})),\end{aligned} \tag{6}$$

where $\measuredangle(\cdot)$ is the angle of the vector present in the argument and K_6 is another tuning constant. The purpose of (6) is to

select the region of space for which the rotation given by (5) should take place, see Fig. 3. In the case of $\mathbf{x}_{eqA}$, the region where the rotation angle α is enabled consists of all the points from $\mathbf{x}_{eqA}$ in the direction of $\mathbf{x}_{eqB}$. It is functions g_A and g_B that are in charge of allowing the merging of the two vector fields from the sinks without resorting to switching or introducing singularities [8].

The purpose of angles ϕ_A and ϕ_B is to account for the desired angle of the vector field while $\measuredangle(\mathbf{x}_{eqI} - \mathbf{x})$ gives the current angle of the vector field; then the difference between them, as it can be seen on (5), is the required angle rotation necessary to achieve the desired angle of the vector field. Depiction of ϕ_A and ϕ_B is presented in Fig. 3.

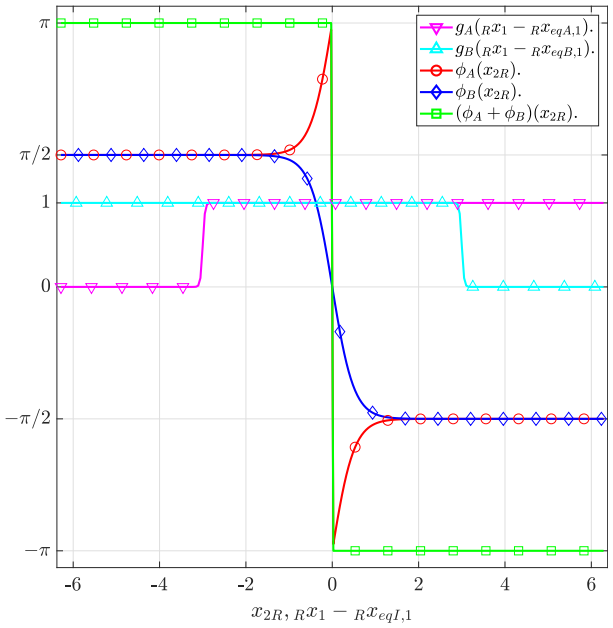


Fig. 3. Windowing functions g_A and g_B. Their purpose is to enable the rotations given by $\phi_A - \measuredangle(\mathbf{x}_{eqA} - \mathbf{x})$ and $\phi_B - \measuredangle(\mathbf{x}_{eqB} - \mathbf{x})$, respectively. Also in this graph are the angles ϕ_A and ϕ_B. they are the measure of how much the vector field of the sinks should rotate to attain the behavior shown in Fig. 4.

Functions ϕ_A, ϕ_B, $\measuredangle(\mathbf{x}_{eqA} - \mathbf{x}))$ and $\measuredangle(\mathbf{x}_{eqB} - \mathbf{x}))$ are given by

$$\begin{aligned}
\phi_A &= \text{atan2}(s\gamma, c\gamma), \text{with } \gamma = \pi + \frac{\pi}{2}\tanh(G_R x_2),\\
\phi_B &= -\frac{\pi}{2}\tanh(G_R x_2),\\
\measuredangle(\mathbf{x}_{eqA} - \mathbf{x})) &= \text{atan2}({}_R x_{eqA,2} - {}_R x_2, {}_R x_{eqA,1} - {}_R x_1),\\
\measuredangle(\mathbf{x}_{eqB} - \mathbf{x})) &= \text{atan2}({}_R x_{eqB,2} - {}_R x_2, {}_R x_{eqB,1} - {}_R x_1).
\end{aligned} \tag{7}$$

It is important to point out that the ranges for $\measuredangle(\mathbf{x}_{eqA} - \mathbf{x}))$ and $\measuredangle(\mathbf{x}_{eqB} - \mathbf{x}))$ are the interval $(-\pi, \pi)$; then, the range of the differences $\phi_A - \measuredangle(\mathbf{x}_{eqA} - \mathbf{x})$ and $\phi_B - \measuredangle(\mathbf{x}_{eqB} - \mathbf{x})$ are $(-\pi/2, \pi/2)$, which leads α and β to have a range $(-\pi/4, \pi/4)$.

C. Control and perturbation inputs

Function $u(t)$ in (1) is the input of the system and it dictates the transition between the two poles that represents a logical value. This function embeds the behavior or input channels *Set* and *Data* present in a D flip-flop where once the *Set* channel is "HIGH", the D flip-flop stores whatever value is present in *Data*. Then, a positive combination of unit-step functions separated by some infinitesimal time is generated when the rising edge of the *Set* signal occurs and the *Data* signal is at logical "HIGH". Similarly, a negative combination of unit-step functions takes place when the *Data* signal is at logical "LOW" and the rising edge of the *Set* signal arrives. The idea behind this approach is to force a transition from one stable equilibrium point to another by a trigger signal of sufficient amplitude and duration that guarantees that the trajectory of the state can abandon the region of influence of its current sink [9], see Fig. 6(b). Function $\omega(t)$ is the input of some perturbation signal considered to be additive noise. In this paper we have adopted the noise injection from [4] as:

$$\begin{aligned}
\omega(t) &= \frac{V_{EMI}}{2}\sum_{k=1}^{N} h(t, T_k)g(t, T_k),\\
h(t, T_k) &= \sin\left(2\pi f(t - T_k)\right),\\
g(t, T_k) &= \tanh\left(K(t - T_k)\right) + \tanh\left(-K(t - T_k + w_d)\right).
\end{aligned} \tag{8}$$

The constant N represents the number of noise injections over the time horizon under analysis. Function $h(t, T_k)$ is the noise injection itself; however, we embed it with function $g(t, T_k)$ to avoid non-smoothness. The time offset T_k is a random variable that tells the location in time of the injection but for the purpose of this paper we use T_k to locate the EEMI injection in time. The constant f is the frequency of the noise.

IV. Features of the Flip-Flop Model

We demonstrate the properties of the system presented in (1) such as the location of its equilibrium points, as well as the stability properties. We are going to require the equilibrium points $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$ satisfy:

$$\begin{aligned}
\mathbf{x}_{eqA} = r\begin{bmatrix}0\\1\end{bmatrix}, \quad \mathbf{x}_{eqB} = r\begin{bmatrix}1\\0\end{bmatrix}, \quad r \in \mathbb{R};\\
\text{hence, } \mathbf{x}_{eqA} - \mathbf{x}_{eqB} = r\begin{bmatrix}-1\\1\end{bmatrix}.
\end{aligned} \tag{9}$$

Lemma IV.1. *Given the system* (1) *over the space* $\mathbb{R}^2$ *with* $u(t) = 0$ *and* $\omega(t) = 0$*, its equilibrium points are* $\mathbf{x}_{eqA}$, $\mathbf{x}_{eqB}$, *and* $\mathbf{x}_m = (1/2)[\mathbf{x}_{eqA} + \mathbf{x}_{eqB}]$.

Proof. We are left with the system

$$\begin{aligned}
\dot{\mathbf{x}} =& L_{eqA}(\mathbf{x})\begin{bmatrix}c\alpha & -s\alpha\\ s\alpha & c\alpha\end{bmatrix}(\mathbf{x}_{eqA} - \mathbf{x}) +\\
& L_{eqB}(\mathbf{x})\begin{bmatrix}c\beta & -s\beta\\ s\beta & c\beta\end{bmatrix}(\mathbf{x}_{eqB} - \mathbf{x}).
\end{aligned} \tag{10}$$

We verify that $\dot{\mathbf{x}} = 0$ by replacing $\mathbf{x}$ with the candidate equilibrium points and corroborating that the mathematical expression on the right of (10) is identically zero. For $\dot{\mathbf{x}} = 0$, take $\mathbf{x} = \mathbf{x}_{eqA}$ and the first term at the left of (10) is identically zero. The second term vanishes to zero as the constant k in Section III-A is chosen big enough to force $L_{eqI}(\mathbf{x})$ to go to zero. A very similar approach arises when taking $\mathbf{x} = \mathbf{x}_{eqB}$.

For the case of $\mathbf{x}_m = (1/2)[\mathbf{x}_{eqA} + \mathbf{x}_{eqB}]$, the rotation necessary for the arguments of (5) yields $g_A = g_B = 1$, and $\phi_A = \pi$, $\phi_B = 0$, $\measuredangle(\mathbf{x}_{eqA} - \mathbf{x}_m) = \pi$, and $\measuredangle(\mathbf{x}_{eqB} - \mathbf{x}_m) = 0$. Note also that $\mathbf{x}_m$ is equidistant to $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqb}$; hence, $L_{eqA} = L_{eqB} = c$ for some constant c. Then (10) reduces to

$$\begin{aligned}\dot{\mathbf{x}} &= c\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}(\mathbf{x}_{eqA} - \mathbf{x}_m) + c\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}(\mathbf{x}_{eqB} - \mathbf{x}_m),\\ &= c\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\Big(\mathbf{x}_{eqA} + \mathbf{x}_{eqB} - 2\mathbf{x}_m\Big) = 0.\end{aligned}$$

□

Theorem IV.2. *The equilibrium points $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$ for system* (10) *are asymptotically stable.*

Proof. We start by rewriting (10) as

$$\dot{\mathbf{x}} = \sum_{I=\{A,B\}} L_{eqI}(\mathbf{x})\begin{bmatrix}c\theta(I) & -s\theta(I)\\ s\theta(I) & c\theta(I)\end{bmatrix}(\mathbf{x}_{eqI} - \mathbf{x}), \quad (11)$$

with $\theta(A) = \alpha$, and $\theta(B) = \beta$. Note that the properties for functions $L_{eqI}(\mathbf{x})$ presented in Section III-A are useful given that $L_{eqI}(\mathbf{x})$ vanishes for $\mathbf{x}$ sufficiently far from $\mathbf{x}_{eqI}$ like, for instance, the proximity of the other equilibrium point; hence, we focus our attention on to the argument of the summation. Then, for the local system:

$$\dot{\tilde{x}} = L_{eqI}(\tilde{\mathbf{x}})\begin{bmatrix}c\theta(I) & -s\theta(I)\\ s\theta(I) & c\theta(I)\end{bmatrix}(\mathbf{x}_{eqI} - \tilde{\mathbf{x}}),$$

where $\tilde{\mathbf{x}}$ belongs to the vicinity of $\mathbf{x}_{eqI}$. The change of variables $\mathbf{y} = \tilde{\mathbf{x}} - \mathbf{x}_{eqI}$ leads to

$$\dot{\mathbf{y}} = L_{eqI}(\mathbf{y} + \mathbf{x}_{eqI})\begin{bmatrix}c\theta(I) & -s\theta(I)\\ s\theta(I) & c\theta(I)\end{bmatrix}(-\mathbf{y}).$$

Our candidate Lyapunov function is $V(\mathbf{y}) = (y_1^2 + y_2^2)/2$ and its time derivative is

$$\begin{aligned}\dot{V}(\mathbf{y}) &= \nabla V \cdot \dot{\mathbf{y}}\\ &= -L_{eqI}(\mathbf{y} + \mathbf{x}_{\mathbf{eqI}})\begin{bmatrix}y_1 & y_2\end{bmatrix}\cdot\begin{bmatrix}c\theta(I) & -s\theta(I)\\ s\theta(I) & c\theta(I)\end{bmatrix}\begin{bmatrix}y_1\\ y_2\end{bmatrix},\\ &= -L_{eqI}(\mathbf{y} + \mathbf{x}_{\mathbf{eqI}})c\theta(I)(y_1^2 + y_2^2),\\ &\leq 0,\end{aligned}$$

for $|c\theta(I)| \geq 0$, which implies $|\theta(I)| \leq \pi/2$. Taking into account the ranges obtained for α, and β in Section III-B, we conclude that $\mathbf{x}_{eqA}$ and $\mathbf{x}_{eqB}$ are asymptotically stable. □

Theorem IV.3. *The equilibrium point $\mathbf{x}_m$ is a saddle point, i.e., it is unstable.*

Proof. We show that there are two invariant regions intersecting at $\mathbf{x}_m$. A subset E is said to be invariant with respect to some dynamics $\dot{\mathbf{x}} = f(\mathbf{x})$ if $x_0 \in E$ and the trajectory $x(t) \in E, \forall t > 0$, [9]. Then, for one of these regions we show that the trajectory diverges from $\mathbf{x}_m$, while in the other region the trajectory converges to $\mathbf{x}_m$ instead.

Consider the region

$$E_1 = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{x} = \mathbf{x}_m + s[-1 \quad 1]^T, s \geq 0, s \in \mathbb{R}\},$$

where $\phi_A = \pi$, $\phi_B = 0$, $\measuredangle(\mathbf{x}_{eqA} - \mathbf{x})) = \pi$, $\measuredangle(\mathbf{x}_{eqB} - \mathbf{x})) = 0$. Then, system (10) becomes

$$\begin{aligned}\dot{\mathbf{x}} =& L_{eqA}\left(\mathbf{x}_m + s\begin{bmatrix}-1\\ 1\end{bmatrix}\right)\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\left(\mathbf{x}_{eqA} - \mathbf{x}_m - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right) +\\ & L_{eqB}\left(\mathbf{x}_m + s\begin{bmatrix}-1\\ 1\end{bmatrix}\right)\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\left(\mathbf{x}_{eqB} - \mathbf{x}_m - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right),\\ =& c_1\left(\mathbf{x}_{eqA} - \mathbf{x}_m - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right) + c_2\left(\mathbf{x}_{eqB} - \mathbf{x}_m - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right),\\ =& c_1\left(\frac{1}{2}(\mathbf{x}_{eqA} - \mathbf{x}_{eqA}) - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right) +\\ & c_2\left(\frac{1}{2}(\mathbf{x}_{eqB} - \mathbf{x}_{eqA}) - s\begin{bmatrix}-1\\ 1\end{bmatrix}\right),\\ =& \left(\frac{c_1 + c_2}{2}\right)\mathbf{x}_{eqA} - \left(\frac{c_1 + c_2}{2}\right)\mathbf{x}_{eqB} - 2\left(\frac{c_1 + c_2}{2}\right)s\begin{bmatrix}-1\\ 1\end{bmatrix},\\ =& \left(\frac{c_1 + c_2}{2}\right)\left(\mathbf{x}_{eqA} - \mathbf{x}_{eqB} - 2s\begin{bmatrix}-1\\ 1\end{bmatrix}\right),\\ =& \left(\frac{c_1 + c_2}{2}\right)(r - 2s)\begin{bmatrix}-1\\ 1\end{bmatrix},\end{aligned}$$

which makes the trajectory $\mathbf{x}(t)$ belongs to E_1. In a similar approach we can also formulate

$$E_2 = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{x} = \mathbf{x}_m + s[-1 \quad 1]^T, s \leq 0, s \in \mathbb{R}\},$$

for which we would obtain

$$\dot{\mathbf{x}} = \left(\frac{c_1 + c_2}{2}\right)(r + 2s)\begin{bmatrix}-1\\ 1\end{bmatrix}.$$

Note that these two slopes are zero at the equilibrium points $\mathbf{x}_A$ and $\mathbf{x}_B$. These two points also belong to E_1 and E_2, as $\mathbf{x}_m$ belongs to both E_1 and E_2. We can combine the results for $\dot{x}$ for the sets E_1 and E_2 so it becomes

$$\dot{\mathbf{x}} = \left(\frac{c_1 + c_2}{2}\right)s(r - 2|s|)\begin{bmatrix}-1\\ 1\end{bmatrix}. \quad (12)$$

Because of Lemma IV.1, we need to incorporate $\mathbf{x}_m$; hence, the vector field (12) also becomes zero for $s = 0$, i.e., for $\mathbf{x}_m$. We have that $s(r - 2|s|) \geq 0$ for all $s \in U$ with $U = \{(-\infty, -r/2] \cup [0, r/2]\}$ while $s(r - 2|s|) < 0$ for $V = \{(-r/2, 0) \cup (r/2, \infty)\}$. Therefore, for $|s| < r/2$, the slope (12) diverges from $\mathbf{x}_m$, rendering this equilibrium point unstable.

Another region of interest consists of

$$D = \{\mathbf{x} \in \mathbb{R}^2 | \mathbf{x} = \mathbf{x}_m + s[1 \quad 1]^T, \rho \geq 0, \rho \in \mathbb{R}\},$$

where the system (10) becomes

$$\dot{\mathbf{x}} = L_{eqA}\left(\mathbf{x}_m + \rho\begin{bmatrix}1\\1\end{bmatrix}\right)\begin{bmatrix}c\alpha & -s\alpha\\ s\alpha & c\alpha\end{bmatrix}\left(\mathbf{x}_{eqA} - \mathbf{x}_m - \rho\begin{bmatrix}1\\1\end{bmatrix}\right) + L_{eqB}\left(\mathbf{x}_m + \rho\begin{bmatrix}1\\1\end{bmatrix}\right)\begin{bmatrix}c\beta & -s\beta\\ s\beta & c\beta\end{bmatrix}\left(\mathbf{x}_{eqB} - \mathbf{x}_m - \rho\begin{bmatrix}1\\1\end{bmatrix}\right).$$

Note that the subspace D is equidistant to $\mathbf{x}_A$ and $\mathbf{x}_B$. As a consequence, $\alpha + \beta = 0 \implies c\beta = c\alpha, s\beta = -s\alpha$, then

$$\begin{aligned}
\dot{\mathbf{x}} =& c\begin{bmatrix}c\alpha & -s\alpha\\ s\alpha & c\alpha\end{bmatrix}\left(\mathbf{x}_{eqA} - \mathbf{x}_m - \rho\begin{bmatrix}1\\1\end{bmatrix}\right) + \\
& c\begin{bmatrix}c\alpha & s\alpha\\ -s\alpha & c\alpha\end{bmatrix}\left(\mathbf{x}_{eqB} - \mathbf{x}_m - \rho\begin{bmatrix}1\\1\end{bmatrix}\right), \\
=& c\begin{bmatrix}c\alpha & -s\alpha\\ s\alpha & c\alpha\end{bmatrix}\left(r\begin{bmatrix}0\\1\end{bmatrix} - \left(\frac{r}{2}+\rho\right)\begin{bmatrix}1\\1\end{bmatrix}\right) + \\
& c\begin{bmatrix}c\alpha & s\alpha\\ -s\alpha & c\alpha\end{bmatrix}\left(r\begin{bmatrix}1\\0\end{bmatrix} - \left(\frac{r}{2}+\rho\right)\begin{bmatrix}1\\1\end{bmatrix}\right), \\
=& c\left(r(c\alpha - s\alpha)\begin{bmatrix}1\\1\end{bmatrix} - (r+2\rho)\begin{bmatrix}1\\1\end{bmatrix}\right), \\
=& -c(rs\alpha + 2\rho c\alpha)\begin{bmatrix}1\\1\end{bmatrix}.
\end{aligned}$$

At $\mathbf{x} = \mathbf{x}_m$, $\rho = 0$ and $s\alpha = 0$ which makes $\dot{\mathbf{x}} = 0$. Meanwhile, for $|\rho| << 1$, $|s\alpha| << 1$, $c\alpha \approx 1$; hence, the term $2\rho c\alpha$ determines the sign of $\dot{\mathbf{x}}$. Finally, for $\rho < 0$, $\dot{\mathbf{x}} > 0$ and $\rho > 0$, $\dot{\mathbf{x}} < 0$, which points outs that trajectories on D converge to $\mathbf{x}_m$. □

V. Simulation Results

The phase portrait of system (1) when $u(t) = 0$ and $\omega(t) = 0$ is presented in Fig. 4. Note that there are two asymptotically stable equilibrium points at $[-1 \quad 1]^T$ and $[1 \quad -1]^T$. The metastable state is represented by the point in between these two equilibrium points and it displays both a stable and an unstable trajectory. It is worth pointing out that the vector field at the bottom right and the top left corners points to the stable equilibrium point radially, while in the region in the middle the vector field becomes parallel to the positive diagonal, and in the vicinity of the negative diagonal the vector field points to the stable equilibrium points. This is the result of (5) and (6), which allows us to merge the properties of radial vector fields such as sinks as well as the vector fields for stable nodes where the rate of approach to the equilibrium is different according to the direction.

As pointed out earlier, the model proposed in (1) is normalized both in state and time; therefore, manipulation of the constants presented in Section III is required to match a realistic model. However, simulation result involving time are presented in Fig. 6(a) and Fig. 6(b). The star marker present in Fig. 6(a) indicates the initial condition chosen for this particular simulation. We can see that the trajectory automatically falls on the stable equilibrium point $[-1 \quad 1]^T$, and it stays there until the signal $u(t)$ triggers the system to the other stable domain of attraction, i.e., $[1 \quad -1]^T$. The trajectory in Fig. 6(a) moves counterclockwise, which makes

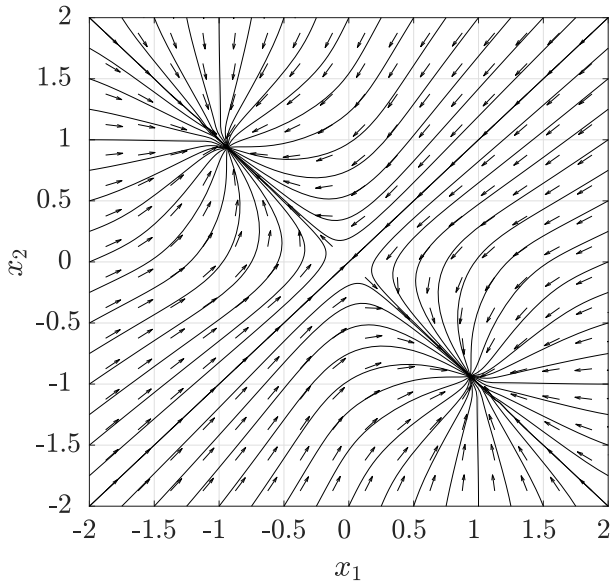


Fig. 4. Phase portrait of the systems (1) for $u(t) = 0$ and $\omega(t) = 0$.

it evident that the trigger signal $u(t)$ need only be sufficient to change the state of the system to the other domain of attraction, so once $u(t)$ is silent, the system drifts to the closest stable equilibrium point.

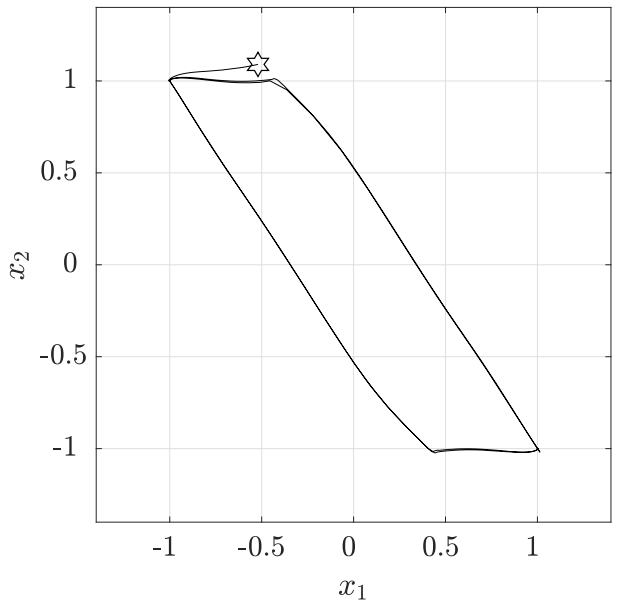
Another simulation with noise injection as additive noise is presented in Fig. 6(a) and Fig. 6(b). This time, the noise injection signal $\omega(t)$ is presented as the signal at the very top of Fig. 6(b), and it takes place at two points in time. First, the noise signal is present at a state where the trajectory of system (1) in on the stable equilibrium point $[-1 \quad 1]^T$. The noise is present on the state of system but the trajectory remains on the domain of attraction of the equilibrium point. At a later time, the noise signal is injected when the input signal $u(t)$ attempts to trigger a transition to the other equilibrium point, causing $u(t)$ not to be strong enough to provoke the transition. It is also visible in Fig. 6(a) where there is an internal loop on the original trajectory.

VI. Conclusions

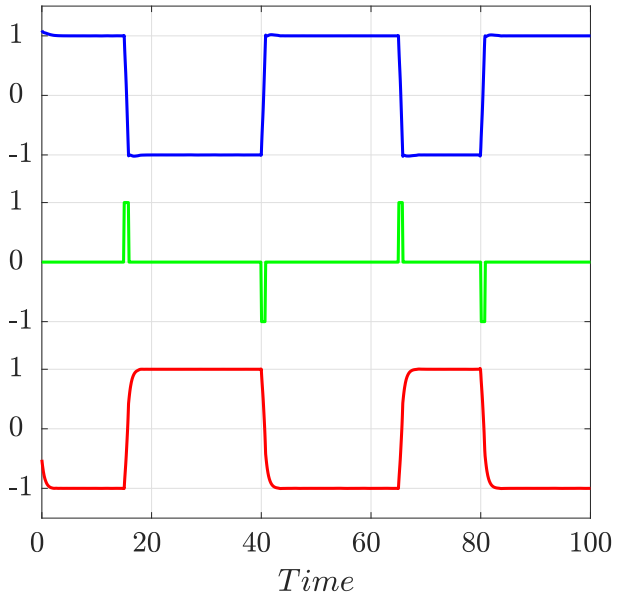
We have presented a bi-dimensional model that replicates the phase diagram of a first order flip-flop. Our model is smooth and does not resort to switching strategies, which comes in handy on hybrid modelings where the amount of discrete states is already big. The model also takes into account the metastability behavior, which can be exploited when studying software execution faults due to an undefined logical state. We conceived the noise injection to be additive noise targeting the transition between the stable equilibrium points. However, the model is flexible and many parameters can be changed to alter its behavior.

References

[1] J. U. Horstmann, H. W. Eichel, and R. L. Coates, "Metastability behavior of cmos asic flip-flops in theory and test," *IEEE J. Solid-State Circuits*, vol. 24, no. 1, pp. 146–157, Feb 1989.

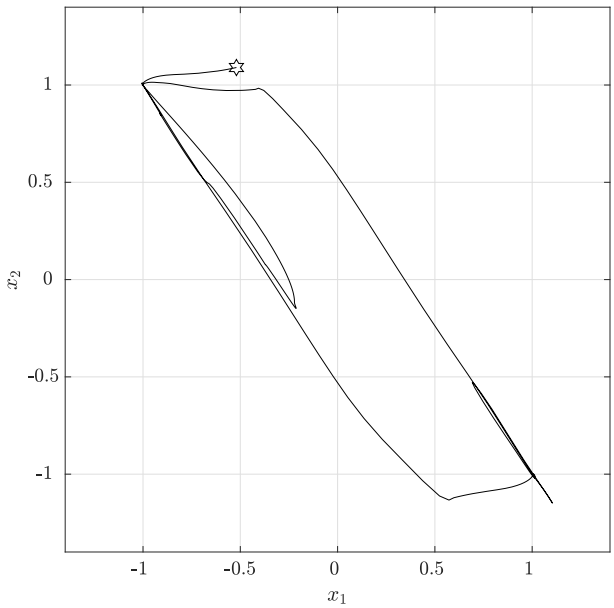


(a)

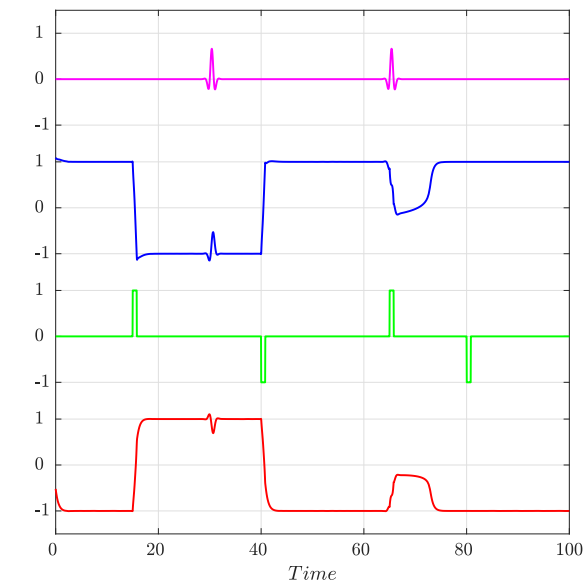


(b)

Fig. 5. Simulation results, (a) Trajectories of (1) in the state plane. The star marker is an arbitrary point chosen as the initial condition and , (b) Output signal $x_1 = Q$, input signal $u(t)$ and output signal $x_2 = \bar{Q}$ in ascending order vs. time.



(a)



(b)

Fig. 6. Simulation results with noise injection as additive noise, (a) The trajectories of (1) in the state plane now has an internal loop that correspond to the noise injection signal $\omega(t)$. The disruption is more evident on (b), where the transition from $\bar{Q}$ to Q is frustrated.

[2] I. Giechaskiel and K. B. Rasmussen, "Sok: Taxonomy and challenges of out-of-band signal injection attacks and defenses," *CoRR*, vol. abs/1901.06935, 2019. [Online]. Available: http://arxiv.org/abs/1901.06935

[3] T. Korak and M. Hoefler, "On the effects of clock and power supply tampering on two microcontroller platforms," in *Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC)*, Sept 2014, pp. 8–17.

[4] L. Valbuena, G. L. Heileman, S. Hemmady, and E. Schamiloglu, "Predicting deviations in software execution paths due to emi injection via reachable sets and random delays," in *2018 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, Sep. 2018, pp. 578–581.

[5] L. Valbuena, G. Heileman, S. Hemmady, and E. Schamiloglu, "A testbed for simulating electromagnetic effects on software execution," in *2017 IEEE International Conference on Circuits and Systems (ICCS)*, Dec 2017, pp. 26–31.

[6] C. Pouant, F. Torrs, A. Reineix, P. Hoffmann, J. Raoult, and L. Chusseau, "Modeling and analysis of large-signal rfi effects in mos transistors," *IEEE Trans. Electromagn. Compat.*, pp. 1–10, 2018.

[7] P. K. Kundu and I. M. Cohen, *Fluid Mechanics*. Elsevier, 2008.

[8] L. A. Valbuena Reyes and H. G. Tanner, "Flocking, formation control, and path following for a group of mobile robots," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1268–1282, July 2015.

[9] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.