Proceedings of the 57th Hawaii International Conference on System Sciences | 2024

Stabilizing D Flip-Flop Orbits with Self-Modifiable Differential Equations

Michael Stephen Fiske
Aemea Institute
mf@aemea.org

Abstract

Recent research has demonstrated electronic hardware attacks on pacemakers and insulin injectors. Injecting clock glitches can skip cryptographic instructions, defeating the security of the executing instructions. Typically, these attacks destabilize the dynamical behavior of the electronics.

For over 70 years, flip-flops have been a fundamental building block of digital computers. Our primary contribution applies self-modifiable differential equations to a D flip-flop model. In particular, meta operators can dynamically self-modify the flip-flop differential equations so that a noise attack is healed or ameliorated. Overall, we introduce new methods of healing a dynamical system that is performing a task.

1. Introduction

Recent research has shown how adversarial electronic interactions can disable systems. Insulin injectors have been sabotaged so that the spurious reporting of the insulin level erratically turns off and on the flow of insulin [1]. Electronically sabotaged pacemakers can produce dangerous shock commands [2]. Injecting clock glitches can skip cryptographic instructions [3].

Out-of-band signal injection attacks alter the measurements of sensors or actuator inputs at the hardware layer [4]. These attacks target the conversion process from a physical quantity to an analog property.

Typically, hardware attacks sabotage the intended dynamical behavior of the electronics. Moreover, there are currently no general mathematical models for describing how an extreme electromagnetic injection (EEMI) attack propagates in electronic components [5].

Current flip-flops are electronic circuits that have two stable physical states which can store information (0 or 1). A flip-flop's state is changed when signals are applied to one or more of its control inputs. Flip-flops are fundamental storage elements in sequential logic.

Fig. 1 shows a D flip-flop circuit, composed of four NAND gates and one NOT gate. CLK is the clock input. $D(t)$ is the data input value (0 or 1) of the D flip-flop at time t. At time t, $Q(t)$ is an output state (0 or 1), and $\overline{Q}(t) = 1 - Q(t)$ is an output state too.

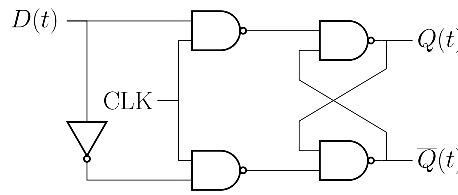

Fig. 1. D Flip-Flop Circuit

Table 1: D Flip-flop Logic

Time	CLK	$D(t)$	$Q(t+1)$	$\overline{Q}(t+1)$
t	Rising Edge	0	0	1
t	Rising Edge	1	1	0
t	Non-Rising	$\{0,1\}$	$Q(t)$	$1 - Q(t)$

Table 1 shows a D flip-flop's logic. According to row 3, if CLK is a non-rising edge at time t, then output state $Q(t+1)$ at time $t+1$ stays the same as $Q(t)$. According to rows 1 and 2, if CLK is a rising edge at time t, then the next output state $Q(t+1)$ at $t+1$ is set to the current value of the data input $D(t)$. If $D(t) = q$, then output state $Q(t+1)$ is set to q, where q is 0 or 1.

D flip-flops are commonly used in digital electronics to build computers and communication systems. Hence, our research focuses on a differential equations model of a D flip-flop. Our primary contribution applies a principle of self-modifiability to differential equations in order to stabilize a D flip-flop during a noise attack. A software simulation of the model is provided. Another contribution introduces meta variables as a new tool for detecting if a flip-flop orbit has been manipulated. Meta variables and meta operators can heal a flip-flop orbit disrupted by noise. Overall, new mathematical methods that heal a dynamical system are introduced.

URI: https://hdl.handle.net/10125/107276
978-0-9981331-7-1

2. Related Work

A differential equations model of a D flip-flop gate was proposed in [5], and was derived from a model in [6]. An experimental test circuit in [6] facilitated accurate and reproducible measurements of [6]'s model of flip-flops, built with CMOS [7] ASIC technology. The model in [5] is a starting point for our results. In [5, 6], they do not propose a self-modifiable dynamical system, nor error-correction for flip-flop circuits.

In [8], soft error code correction in flip-flops addresses errors due to neutrons generated from cosmic rays and alpha particles from packaging material. In [8], they do not address active electromagnetic attacks by Mallory, and self-modifiability is not used.

Self-modifiable machine instructions originated in [9, 10, 11]. In [12], a principle of self-modifiability and self-modifiable differential equations were first introduced. In [13, 14], hybrid systems use automata combined with differential equations to model complex processes. Hybrid systems do not have a notion of self-modifiability, nor variable spaces, nor meta operators that can dynamically add new variables and equations.

DNA repair has been extensively studied for over fifty years. DNA models explain repair at a biomolecular level [15]. Organisms have repaired their DNA for perhaps a few billion years.[1] Biomolecular DNA models do not propose self-modifiable differential equations.

"Self-modifying systems" were proposed for understanding the development of complexity in biology [16]. However, the mathematics described in [16] doesn't actually self-modify: e.g., there is no formalism for adding new variables and equations as time proceeds. Moreover, Turing machines are described in [16]'s "self-modifying systems", even though the rules governing a Turing machine program stay fixed.

3. A D Flip Flop Gate Model

A D flip-flop gate model[2] [5] is reviewed and explained. Fig. 2 shows a vector field of the model, generated by Julia's plotting software [19]. The vector field is plotted on a subset $[-2, 2] \times [-2, 2]$ of $\mathbb{R}^2$.

In Fig. 2, fixed points[3] $a = (1, -1)$ and $b = (-1, 1)$ represent two stable states[4] that this D flip-flop model can reach. The rest of this section defines the differential equations that produce the vector field in Fig. 2.

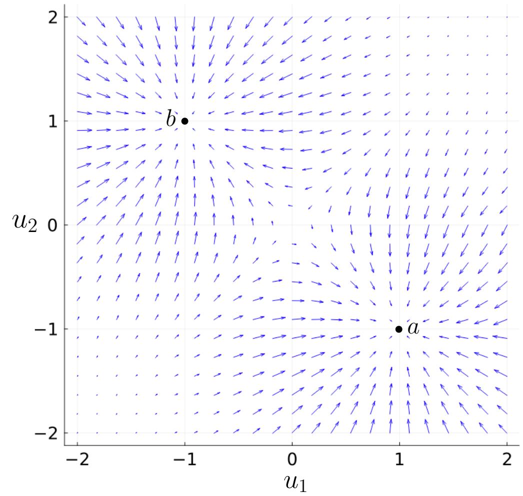

Fig. 2. D flip-flop vector field $\left(\frac{du_1}{dt}, \frac{du_2}{dt}\right)$

Let c, d and L_0 be constants that are real numbers. For a fixed point $p = (p_1, p_2)$ in the square $\{(x, y) \in \mathbb{R}^2 : -2 \le x, y \le 2\}$ and any point $u = (u_1, u_2) \in \mathbb{R}^2$, define metric $||p - u||^2 = (p_1 - u_1)^2 + (p_2 - u_2)^2$. Define a radially symmetric function centered at p as $L_p(u) = \dfrac{L_0}{1 + e^{c||p-u||^2 - d}}$. Function $L_p(u)$ helps define a differential equation (1) flip-flop model. (In sections 4 and 5, signal input and noise are added to the model.)

$$\frac{du}{dt} = L_a(u) \begin{pmatrix} \cos\alpha(u) & -\sin\alpha(u) \\ \sin\alpha(u) & \cos\alpha(u) \end{pmatrix} (a - u) \; + \; L_b(u) \begin{pmatrix} \cos\beta(u) & -\sin\beta(u) \\ \sin\beta(u) & \cos\beta(u) \end{pmatrix} (b - u) \quad (1)$$

$\alpha(u)$ and $\beta(u)$ have two favorable properties:

- Angles $\alpha(u)$ and $\beta(u)$ are smooth functions of u near the stable manifold $u_2 = u_1$ that separates the basins of attraction of stable fixed points a and b.
- Angle functions $\alpha(u)$ and $\beta(u)$ help realistically model flip-flop behavior measured in hardware.

The rest of this section defines functions that help build $\alpha(u)$ and $\beta(u)$. g_A and g_B are constructed from fixed points $A = (a_1, a_2)$, $B = (b_1, b_2)$, and constant κ. For $u = (u_1, u_2)$, define $g_A(u) = \frac{1}{2} + \frac{1}{2}\tanh(u_1 - \kappa a_1)$, and define $g_B(u) = \frac{1}{2} - \frac{1}{2}\tanh(u_1 - \kappa b_1)$.

For $u = (u_1, u_2)$ and fixed point $p = (p_1, p_2)$, define $\angle(p, u) = \text{atan2}(p_2 - u_2,\ p_1 - u_1)$.[5] Recall that

[1] Archaeans, single-celled microorganisms, have been detected in shales dating from 2.7 billion years ago.

[2] According to Dr. Valbuena [17], his model is based on an electromechanical D flip-flop so that orbits can be measured. Model constants can be selected to match the experimental data in [6] with high accuracy. An electromechanical D flip-flop was built because his calculations predicted that measurements, of a commercial D flip-flop, would occur over extremely small time horizons (picoseconds) [18].

[3] Sometimes fixed points are called equilibrium points.

[4] $Q(t)$ and $\overline{Q}(t)$ correspond to the 1st and 2nd coordinates of the fixed point: bit $0 \leftrightarrow -1$ and bit $1 \leftrightarrow 1$. In table 1, $D(t)$ is 0 or 1.

[5] $\angle(p, u)$ computes the angle between the x-axis and the vector $\vec{p} - \vec{u}$, where $\vec{p} = \langle p_1, p_2 \rangle$ and $\vec{u} = \langle u_1, u_2 \rangle$.

$$\text{atan2}(y,x) = \begin{cases} \arctan(y/x) & \text{if } x > 0 \\ \arctan(y/x) + \pi & \text{if } x < 0 \text{ and } y \geq 0 \\ \arctan(y/x) - \pi & \text{if } x < 0 \text{ and } y < 0 \\ \pi/2 & \text{if } x = 0 \text{ and } y > 0 \\ -\pi/2 & \text{if } x = 0 \text{ and } y < 0 \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0 \end{cases}$$

Define $\phi_B(u) = -\frac{\pi}{2}\tanh(u_2)$. Define $\gamma(u) = \pi - \phi_B(u)$. Define $\phi_A(u) = \text{atan2}\big(\sin\gamma(u), \cos\gamma(u)\big)$.

Define $\alpha(u) = \frac{1}{2}\ g_A(u)\ \big(\phi_A(u) - \angle(A, u)\big)$.

Define $\beta(u) = \frac{1}{2}\ g_B(u)\ \big(\phi_B(u) - \angle(B, u)\big)$.

Constants $c = 0.8$, $d = 1$, $L_0 = 2$, and $\kappa = 3$, and fixed points $a = (1, -1)$ and $b = (-1, 1)$ generated the vector field shown in Fig. 2. These constants were chosen, based on software simulations [19] of the model; some graphs and figures in [5] based on physical measurements from a electromechanical D flip-flop built in a lab [18]; and email communications [17].

For all values c in interval $[0.8, 1.3]$, d in $[0.4, 1.2]$, L_0 in $[1, 3]$, and κ in $[3, \infty)$, the vector field, in equation (1), "stays qualitatively close to the vector field" in Fig. 2: stable fixed points a and b do not move or vanish, and the two basins of attraction are separated by the line $u_2 = u_1$. At values $c = \frac{1}{2}$, $d = \frac{4}{5}$, $L_0 = 2$, $\kappa = 3$, fixed points a and b, on opposite sides of $u_2 = u_1$, vanish: a bifurcation occurs near $c = \frac{1}{2}$.

4. Adding Input to the Model

Fig. 2 shows a vector field of the D flip-flop model that has two stable outputs. In one output, a flip-flop's orbit reaches the attracting fixed point at $b = (-1, 1)$; in the second output state, the orbit reaches the attracting fixed point $a = (1, -1)$. For example, the first coordinate of b can represent a logical 0 bit; the first coordinate of a can represent a logical 1 bit.

For $u = (u_1, u_2)$, line $u_2 = u_1$ is a boundary of the system of equations, defined in equation (1), because it separates initial points $u(0)$ that eventually reach fixed point $a = (1, -1)$ and points that eventually reach fixed point $b = (-1, 1)$.[6] For all initial points $u(0) = (x_0, y_0)$ satisfying $x_0 < y_0$, the orbit of $u(0)$ converges to $b = (-1, 1)$. For all points $u(0) = (x_0, y_0)$ satisfying $x_0 > y_0$, the orbit of $u(0)$ converges to $a = (1, -1)$.

We add input signal $\mathcal{I}(t)$ with constants K_1, K_2:

$$\frac{du}{dt} = L_a(u)\begin{pmatrix}\cos\alpha(u) & -\sin\alpha(u)\\ \sin\alpha(u) & \cos\alpha(u)\end{pmatrix}(a-u) + L_b(u)$$
$$\begin{pmatrix}\cos\beta(u) & -\sin\beta(u)\\ \sin\beta(u) & \cos\beta(u)\end{pmatrix}(b-u) + \begin{pmatrix}K_1\\ K_2\end{pmatrix}\mathcal{I}(t) \quad (2)$$

[6] In electronics, $u_2 = u_1$ is sometimes called a *wall*; in dynamical systems, $u_2 = u_1$ is a *stable manifold* with saddle fixed point $(0, 0)$.

An example of $\mathcal{I}(t)$ is defined that illustrates how the input signal can toggle the flip-flop between fixed points $a = (1, -1)$ and $b = (-1, 1)$. For parameters T_0, period ρ and pulse width τ, define $r(t, T_0, \rho) = (t - T_0) \mod \rho$. Define $\psi(t, T_0, \rho, \tau) =$

$$\begin{cases} 1 & \text{if } r(t, T_0, \rho) \leq \tau \text{ and } \lfloor (t - T_0)/\rho \rfloor \text{ is even.} \\ -1 & \text{if } r(t, T_0, \rho) \leq \tau \text{ and } \lfloor (t - T_0)/\rho \rfloor \text{ is odd.} \\ 0 & \text{if } r(t, T_0, \rho) > \tau \end{cases}$$

$\mathcal{I}(t) = \psi(t, 16, 24, 1.8)$ is shown in Fig. 3.[7]

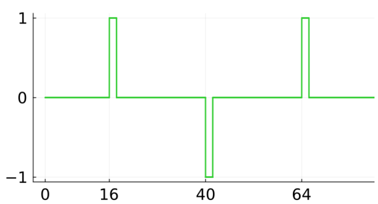

Fig. 3. Input Signal $\mathcal{I}(t)$. $T_0 = 16$. $\rho = 24$. $\tau = 1.8$.

4.1. A Toggled Flip-Flop Orbit

Fig. 4 shows an orbit of the flip-flop model with initial point $u(0) = (-1.7, 0.8)$ during the time period $\Gamma = [0, 17.8]$. During $u(t)$'s entire orbit, the constant values are $K_1 = 0.6$, $K_2 = -1.5$, $T_0 = 16$, period $p = 24$, and width $\tau = 1.8$. Input signal $\mathcal{I}(t) = \psi(t, 16, 24, 1.8)$, where $\mathcal{I}(16) = 1$. See Fig. 3.

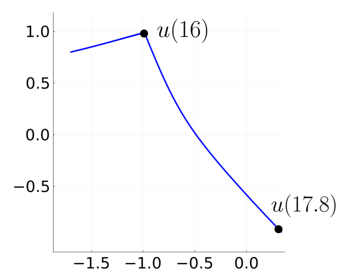

Fig. 4. Orbit of $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 17.8]$.

Point $u(16)$ is very close to stable fixed point $b = (-1, 1)$. $\mathcal{I}(t)$'s effect on $u(t)$'s orbit is shown in Fig. 4: at time $t = 16$, the orbit abruptly changes. When $\mathcal{I}(t) = 1$, term $(K_1\mathcal{I}(t), K_2\mathcal{I}(t))$ dominates the other terms in equation (2) that create the vector field in Fig. 2. During time period $[16, 17.8]$, the orbit crosses boundary $u_2 = u_1$, so $u(17.8)$ is in fixed point a's basin of attraction.

During time period $(17.8, 20]$, $\mathcal{I}(t) = 0$. $u(t)$'s orbit follows the vector field in Fig. 2. $u(20)$ is close to fixed point $a = (1, -1)$. See Fig. 5. During period $[20, 40)$, $\mathcal{I}(t) = 0$, so $u(t)$'s orbit approaches even closer to a.

[7] $\mathcal{I}(t)$ is normalized to 1. If an electronic flip-flop operates at 3 volts, then $\mathcal{I}(t) = 1$ may correspond to 3 volts.

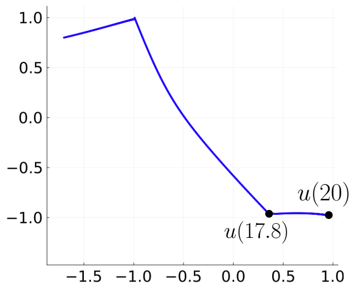

Fig. 5. Orbit of $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 20]$.

At time $t = 40$, $\mathcal{I}(40) = -1$ so there is an abrupt change in the orbit. During time period $[40, 41.8]$, Fig. 6 shows the orbit has crossed the boundary $u_2 = u_1$. Point $u(41.8)$ is in fixed point b's basin of attraction.

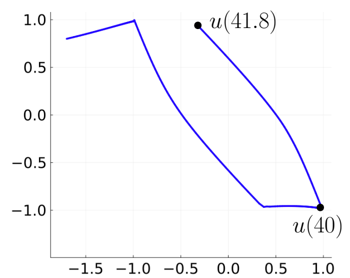

Fig. 6. Orbit of $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 41.8]$.

During period $[41.8, 45]$, $\mathcal{I}(t) = 0$. $u(t)$ follows Fig. 2's vector field and approaches $b = (-1, 1)$. See Fig. 7.

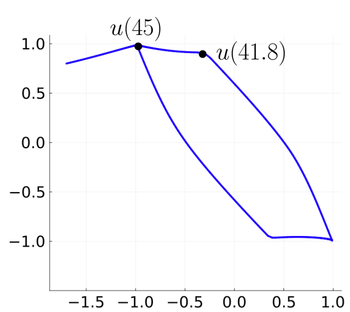

Fig. 7. Orbit of $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 45]$.

We explain how $\mathcal{I}(t)$'s toggling relates to table 1. $(K_1\mathcal{I}(t), K_2\mathcal{I}(t))$ models data input $D(t)$ combined with clock input CLK. $Q(t) = 0$, $\overline{Q}(t) = 1$ corresponds to $b = (-1, 1)$. $Q(t) = 1$, $\overline{Q}(t) = 0$ corresponds to $a = (1, -1)$. $\mathcal{I}(t) = 0$ corresponds to row 3 of Table 1. $\mathcal{I}(t) = 1$ at $t = 16$ corresponds to row 2, where $Q(t+1)$ is set to 1, as $u(20) \approx a$.[8] $\mathcal{I}(40) = -1$ corresponds to row 1, where $Q(t+1)$ is set to 0, as $u(t)$ is toggled to b.

If input constants K_1, K_2 are too small, then $|K_1|$ and $|K_2|$ must be substantially different. When $|K_1|$ and $|K_2|$ are small and $|K_1| \approx |K_2|$, $u(t)$'s orbit cannot be successfully toggled from one fixed point to the other fixed point because the vector field is close to zero near the saddle point $(0, 0)$. When $K_1 = 0.6$, $K_2 = -0.6$, $c = 0.8$, $d = 1$, $L_0 = 2$, $\kappa = 3$ with $u(0) = (-1.7, 0.8)$ as in Figs. 4-7, and $\tau = 1.8$ then $u(17.8) \approx (-0.484, 0.492)$. The flip-flop's orbit does not cross the boundary $u_2 = u_1$ and returns to fixed point b by time $t = 24$. Keeping the other constants the same, when $K_1 = 1.5$ and $K_2 = -1.5$ with $u(0) = (-1.7, 0.8)$, then $u(17.8) \approx (1.47, -1.47)$, so the orbit successfully crosses boundary $u_2 = u_1$.

Successful toggling depends upon time width τ of input $\mathcal{I}(t)$. For example, when $K_1 = 1.5$ and $K_2 = -1.5$ with $u(0) = (-1.7, 0.8)$ and $\tau = 1$, the orbit successfully crosses the boundary $u_2 = u_1$. When $\tau = 0.8$, the orbit does not cross $u_2 = u_1$ and by time $t = 24$, $u(24) \approx (-0.986, 0.987) \approx b$.

5. Noise Can Disrupt Flip-Flop Toggling

A noise pulse can disrupt flip-flop toggling. Noise term $\eta(t)$ is added to equation (2). K_3, K_4 are constants.

$$\frac{du}{dt} = L_a(u)\begin{pmatrix}\cos\alpha(u) & -\sin\alpha(u)\\ \sin\alpha(u) & \cos\alpha(u)\end{pmatrix}(a-u) \quad + \quad L_b(u)\begin{pmatrix}\cos\beta(u) & -\sin\beta(u)\\ \sin\beta(u) & \cos\beta(u)\end{pmatrix}(b-u) \quad + \quad \begin{pmatrix}K_1\\ K_2\end{pmatrix}\mathcal{I}(t) \quad + \quad \begin{pmatrix}K_3\\ K_4\end{pmatrix}\eta(t) \qquad (3)$$

Set $h(t, l, f) = \sin\big(2\pi f(t-l)\big)$. Set $g(t, l, \delta, k) = \tanh(k(t-l)) + \tanh(-k(t-l+\delta))$. See Fig. 8.

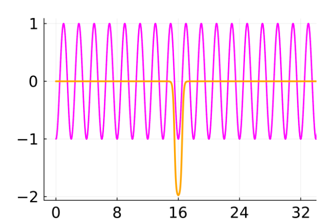

Fig. 8. $h(t, \frac{33}{2}, \frac{1}{2})$ is magenta. $g(t, \frac{33}{2}, 1, 5)$ is a pulse.

Set $\chi(t, l, f, \delta, k, \mathcal{A}) = \mathcal{A} * g(t, l, \delta, k) * h(t, l, f)$. Fig. 9 shows an instance of $\eta(t) = \chi(t, \frac{33}{2}, \frac{1}{2}, 1, 5, \frac{3}{2})$. In Fig. 9, $l = \frac{33}{2}$ is the time location of the noise. $f = \frac{1}{2}$ is the frequency of function h. $\delta = 1$ is the time width of function g. Constant $k = 5$ sets the steepness of g on each side of time l. $\mathcal{A} = \frac{3}{2}$ is the amplitude.

[8] $Q(t+1)$'s time scale is distinct from $u(t)$'s time scale.

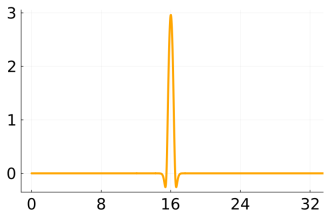

Fig. 9. Noise $\eta(t) = \chi(t, \frac{33}{2}, \frac{1}{2}, 1, 5, \frac{3}{2})$.

Figs. 10 and 11 show a noise injection on $u(t) = \big(u_1(t), u_2(t)\big)$ where $K_3 = 0$ and $K_4 = 1.0$. Input $\mathcal{I}(t) = \psi(t,\ 16, 24, 1.8)$. Noise $\eta(t) = \chi(t, \frac{33}{2}, \frac{1}{2}, 1,\ 5, \frac{3}{2})$ disrupts $u_2(t)$'s orbit during time period $[14.5, 17.5]$.[9]

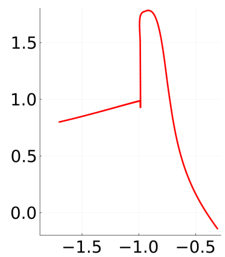

Fig. 10. Noise orbit $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 17.8]$.

Fig. 10 shows the effect on $u(0) = (-1.7, 0.8)$'s orbit. $u(17.8)$ still lies in fixed point b's basin of attraction. Compare Figs. 10 and 4.

During period $[17.8, 24]$, Fig. 11 shows that $u(0)$'s orbit converges to fixed point $b = (-1, 1)$. Noise $\eta(t)$, in Fig. 9, has disrupted $\mathcal{I}(t)$'s flip-flop toggling.

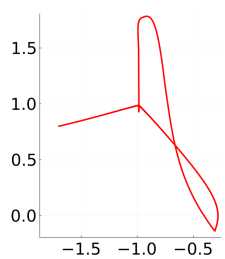

Fig. 11. Noise orbit $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 24]$.

6. Gate Model Assumptions

In our model, *Alice*'s goal is to hinder *Mallory* from sabotaging Alice's flip-flop computation. Assumptions about Mallory's sabotaging with noise are described after a comprehensive description of our extended model. Some of these assumptions defer hardware implementations to a subsequent paper.

[9] $|\eta(t)| < 10^{-19}$ whenever t lies in $[0, 14.5) \cup (17.5, +\infty)$.

7. Adding Healing Variables

When a simulated EEMI attacks the D flip-flop model, an effective attack creates enough noise for a long enough period of time so that the orbit is corrupted with respect to the boundary $u_2 = u_1$. The noise causes the output of the flip-flop to compute the incorrect value. A primary goal is to explore mathematical methods of healing a disrupted flip-flop orbit, since hardware implementations are beyond the scope of this paper.

An information theory [20] analogy is helpful in explaining our approach. Suppose Alice transmits 16 bits of signal to Bob. If any of the 16 bits are flipped due to noise in the transmission medium, Bob has no way of knowing if any of the bits have been flipped. To repair errors, Alice must transmit additional bits that help Bob with error code correction [21].

Similar to error correction, an extension of equation (3) adds new variables that heal the flip-flop orbit when it is disrupted by noise. Set $\nu_1(t) = \big(u_1(t),\ u_2(t)\big)$. For $i \geq 2$, set $\nu_i(t) = \big(u_{2i-1}(t),\ u_{2i}(t)\big)$. Define n differential equations, where each i is in $1, \dots, n$:

$$\frac{d\nu_i}{dt} = L_a(\nu_i) \begin{pmatrix} \cos\alpha_i(\nu_i) & -\sin\alpha_i(\nu_i) \\ \sin\alpha_i(\nu_i) & \cos\alpha_i(\nu_i) \end{pmatrix} (a - \nu_i) \ + \ L_b(\nu_i) \begin{pmatrix} \cos\beta_i(\nu_i) & -\sin\beta_i(\nu_i) \\ \sin\beta_i(\nu_i) & \cos\beta_i(\nu_i) \end{pmatrix} (b - \nu_i) \ + \ \begin{pmatrix} K_{i,1} \\ K_{i,2} \end{pmatrix} \mathcal{I}(t) + \begin{pmatrix} K_{i,3} \\ K_{i,4} \end{pmatrix} \eta_i(t) \quad (4)$$

Analogous to $\alpha(u)$ and $\beta(u)$ in section 3, angles α_1 and β_1 are computed between the two vectors derived from u_1 and u_2. If $i \geq 2$, angles α_i and β_i are computed between the two vectors derived from u_{2i-1} and u_{2i}.

When $n = 2$, two new variables u_3, u_4 and two more standard differential equations are added to the system. System $u(t) = \big(u_1(t), u_2(t), u_3(t), u_4(t)\big)$ is subject to equation (4). Set boundary conditions $u_1(0) = u_3(0)$ and $u_2(0) = u_4(0)$. Set input constants $K_{1,1} = K_{2,1} = 0.6$, and $K_{1,2} = K_{2,2} = -1.5$. Set noise constants $K_{1,3} = K_{2,3} = K_{2,4} = 0$, and $K_{1,4} = 1$.

Both $\nu_1(t)$ and $\nu_2(t)$ have the same stable fixed points $a = (a_1, a_2)$ and $b = (b_1, b_2)$. Hence, if noise injection does not occur, then the orbits of $\nu_1(t)$ and $\nu_2(t)$ are identical: $\nu_1(t) = \nu_2(t)$ for all t.

Assume u_1 and u_2 "perform the computation". u_1 and u_2 are called *signal variables*. u_3 and u_4 are called *repair* or *healing variables*.[10] For each i in $\{1, 2\}$, define meta differential equations: $\dfrac{d\omega_i}{dt} = \dfrac{du_{i+2}}{dt} - \dfrac{du_i}{dt}$ with initial conditions $\omega_1(0) = \omega_2(0) = 0$. ω_1 and ω_2 are

[10] Variables u_3 and u_4 are analogous to error correction bits.

meta variables. Define two thresholds $\theta_1, \theta_2 > 0$ and two meta operator execution times s_1, s_2.[11]

Let s_1 be the first time, where $|\omega_1(s)| \geq \theta_1$. Then

$$s_1 = \inf\Big\{\tau \geq 0 : \Big|\int_0^\tau \Big(\frac{du_3}{dt} - \frac{du_1}{dt}\Big)dt\Big| \geq \theta_1\Big\}. \qquad (*)$$

If s_1 exists, at time s_1 a meta operator $\mathcal{M}_1$ bound to ω_1 executes: $\mathcal{M}_1$ modifies $\frac{du_1}{dt} = \dots$ in equation (4).

Let s_2 be the soonest time such that $|\omega_2(s_2)| \geq \theta_2$.

$$s_2 = \inf\Big\{\tau \geq 0 : \Big|\int_0^\tau \Big(\frac{du_4}{dt} - \frac{du_2}{dt}\Big)dt\Big| \geq \theta_2\Big\}. \qquad (**)$$

If s_2 exists, then at time s_2 meta operator $\mathcal{M}_2$ bound to ω_2 executes: $\mathcal{M}_2$ modifies $\frac{du_2}{dt} = \dots$ in equation (4).

Meta operator 1 is defined below.

Meta Operator 1. *Lock Variables* u_1, u_2

if $|\omega_1(s_1)| \geq \theta_1$ OR $|\omega_2(s_2)| \geq \theta_2$
 for i in $\{1,2\}$ {
  $\big(u_1(s_i), u_2(s_i)\big)$ is the current point.
  set $\frac{du_i}{dt}(u_1, u_2, s) = 0$ for $s \in [s_i, s_i + \theta_i]$.
}

Set $\theta_1 = \theta_2 = 0.01$. Fig. 12 shows (u_1, u_2)'s orbit when meta operator 1 is bound to ω_2 and θ_2. Fig. 4 shows $(u_3(t), u_4(t))$'s orbit.[12] In meta operator 1, variables u_3 and u_4 lock (u_1, u_2)'s orbit near fixed point b. If there is an AND gate to compare to (u_3, u_4), the result can detect a noise injection on (u_1, u_2).

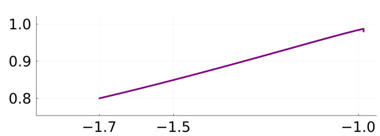

Fig. 12. Meta Procedure 1. $u(0) = (-1.7, 0.8)$. $\Gamma = [0, 17.8]$. Noise orbit of (u_1, u_2).

Variables (u_3, u_4) help heal an attack on (u_1, u_2).

Meta Operator 2. *Self-Modify Vector Field*

if $|\omega_1(s_1)| \geq \theta_1$ {
  $\big(u_1(s_1), u_2(s_1)\big)$ is the current point.
  for all $s \geq s_1$ such that $|\omega_1(s)| \geq \theta_1$:
   set $\dfrac{du_1}{dt}(u_1, u_2, s) = \dfrac{du_3}{dt}(u_1, u_2, s)$.
}
if $|\omega_2(s_2)| \geq \theta_2$ {
  $\big(u_1(s_2), u_2(s_2)\big)$ is the current point.
  for all $s \geq s_2$ such that $|\omega_2(s)| \geq \theta_2$:
   set $\dfrac{du_2}{dt}(u_1, u_2, s) = \dfrac{du_4}{dt}(u_1, u_2, s)$.
}

In a simulation of meta operator 2, the same noise injection is used as section 5 that caused the flip-flop toggling to fail (Fig. 11). The noise constants are $K_{1,3} = 0$; $K_{1,4} = 1$, and $K_{2,3} = K_{2,4} = 0$. During time period $[16, 17.8]$, noise pulse $\eta(t) = \chi(t, \frac{33}{2}, \frac{1}{2}, 1, 5, \frac{3}{2})$, shown in Fig. 9, attempts to disrupt u_2's orbit when input signal $\mathcal{I}(t) = \psi(t, 16, 24, 1.8)$.

Fig. 13 shows that meta operator 2 with $\theta_1 = \theta_2 = 0.2$ successfully heals this noise injection in variable u_2. By $t = 24$, the orbit $\big(u_1(t), u_2(t)\big)$ successfully toggles and reaches $a = (1, -1)$. Compare Figs. 13 and 11.

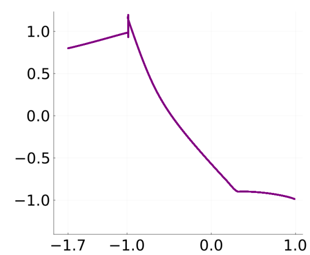

Fig. 13. Horizontal axis: u_1. Vertical axis: u_2. Meta Operator 2. $\theta_1 = \theta_2 = 0.2$. $\Gamma = [0, 24]$.

Observe that $K_{2,3} = K_{2,4} = 0$, so healing variables u_3, u_4 are not attacked with noise. Also, $K_{1,4} = 1$, so variable u_2 is disrupted by noise near $t = 16$.[13]

If the thresholds are set to $\theta_1 = \theta_2 = 1$ and the time width δ of $\eta(t)$ is set to 1.4, then meta operator 2 no longer can heal the noise disruption. For these same values, $\theta_1 = \theta_2 = 1$ and $\delta = 1.4$, if width τ of $\mathcal{I}(t)$ increases from 1.8 to 2.0, then meta operator 2 successfully heals the stronger noise injection, even with very weak thresholds $\theta_1 = \theta_2 = 1$. There are two effects. Meta operator 2 heals u_2's orbit at $t \approx 15.63$. The second effect is that $\eta(t)$ is effectively 0 after $t = 17$, and $\mathcal{I}(t) = 1$ until $t = 18$. See Fig. 14.

Consider that meta variables ω_1, ω_2 integrate the difference between variables u_1, u_3 and u_2, u_4, respectively: $\frac{d\omega_i}{dt} = \frac{du_{i+2}}{dt} - \frac{du_i}{dt}$, where i is 1 or 2. If $|\omega_2| \geq \theta_2 = 1$, then $|u_2 - u_4| \geq 1$. This means it is likely that the (u_1, u_2) and (u_3, u_4) orbits are on opposite sides of the line $u_2 = u_1$, implying that u_2's orbit was disrupted from its correct orbit.

[11] Meta variables and meta operators are more comprehensively defined in the appendix. Section 10.1 defines a detectable event. Set $\mathcal{C}_i = (-\infty, -\theta_i] \cup [\theta_i, \infty)$ for i in $\{1, 2\}$. Set $f\Big(\omega, \dfrac{d\omega}{dt}, t\Big) = \dfrac{d\omega}{dt}$. $X = \mathbb{R}$. Then (*) and (**) satisfy condition (***), defined in 10.1.

[12] $(u_1(t), u_2(t))$'s orbit is the same as Fig. 4 when there is no noise.

[13] In Fig. 9, the peak of pulse $\eta(t)$ is near $t = 16$. The peak amplitude of noise pulse $\eta(t)$ is over 5 times larger than the peak amplitude of the noise pulses, used by [5] in figure 6b.

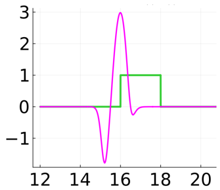

Fig. 14. $\mathcal{I}(t)$ is green. $\tau = 2$. $\eta(t)$ is magenta. $\delta = 1.4$

Meta operator 3 increases the pulse width of $\mathcal{I}(t)$.

Meta Operator 3. *Self-Modify Input Signal*

`if` $|\omega_1(s_1)| \geq \theta_1$ `OR` $|\omega_2(s_2)| \geq \theta_2$

`increase pulse width` τ `of input` $\mathcal{I}(t)$

Meta operator 4 combines meta operators 2 and 3.

Meta Operator 4. *Heal*

`Operator 2 changes the vector field.`
`Operator 3 changes the input signal.`

`if` $|\omega_1(s_1)| \geq \theta_1$

`execute` s_1 `parts of Operators 2 & 3`

`if` $|\omega_2(s_2)| \geq \theta_2$

`execute` s_2 `parts of Operators 2 & 3`

8. Decentralizing Healing Variables

In this section, the role of each variable is decentralized over time. A noise injection only lasts for a short period of time.[14] Decentralization of a task between variables addresses the weakness that a sophisticated Mallory will attempt to attack the weakest part of the system. Hence, in meta operators 1, 2, 3 and 4, if Mallory knows or guesses which variables perform the healing, Mallory will likely attack the healing variables.

To address this *single point (variable) of failure*, the variables should heal each other in one or more *variable cycles*. Decentralization enables each variable to have a dual purpose: signal and healing. After showing how meta operator 2 fails on a noise attack on its healing variable, meta operator 5 is defined that is more robust to noise. Lastly, we explain variable cyclicity and how changes in the noise parameters and number of variables affects the efficacy of meta operator 5.

A noise attack on meta operator 2 is examined. A successful attack provides an example that demonstrates the intuition of *Mallory attacking the healing variable*. Fig. 15 shows two noise pulses $\eta_2(t) = \chi(t, l, f, \delta_2, k, \mathcal{A})$. and $\eta_3(t) = \chi(t, l, f, \delta_3, k, \mathcal{A})$ with

[14]The case, when the noise is generated for the duration of the whole computation, is not addressed: this is analogous to physically destroying a computer. A priority is to hinder Mallory from manipulating the computation to do something that benefits Mallory: e.g., flipping a bit in a flip-flop to subvert a cryptography algorithm.

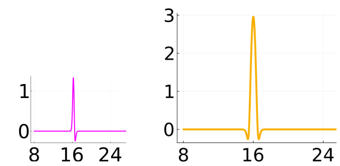

Fig. 15. Two Noise Pulses $\eta_i(t) = \chi(t, l, f, \delta_i, k, \mathcal{A})$. $l = \frac{33}{2}$. $f = \frac{1}{2}$. $\delta_2 = \frac{1}{3}$. $\delta_3 = 1$. $k = 5$. $\mathcal{A} = \frac{3}{2}$.

parameters $l = \frac{33}{2}$, $f = \frac{1}{2}$, $\delta_2 = \frac{1}{3}$, $\delta_3 = 1$, $k = 5$, and $\mathcal{A} = \frac{3}{2}$. In Fig. 15, η_2 is magenta; η_3 is the larger pulse.

Fig. 16 shows η_2, η_3 attacking variables u_2, u_4, respectively. Meta operator 2 fails with $\theta_1 = \theta_2 = 0.2$.

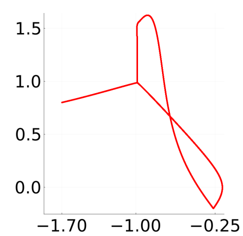

Fig. 16. Horizontal axis: u_1. Vertical axis: u_2. Meta Operator 2 Fails. Variables u_2, u_4 attacked.

To address the failure of meta operator 2 when both u_2 and u_4 are attacked, an example is extended to six variables $(u_1, u_2, u_3, u_4, u_5, u_6)$. In equation (4), $i = 1, 2,$ or 3. Similar to section 7, set $K_{1,1} = K_{2,1} = K_{3,1} = 0.6$ and $K_{1,2} = K_{2,2} = K_{3,2} = -1.5$, so the same constants are used with input $\mathcal{I}(t)$.

Define meta variables so that the signal and healing are decentralized over the variables. Let ω_1, ω_2, and ω_3 be the meta variables with initial conditions $\omega_1(0) = \omega_2(0) = \omega_3(0) = 0$. All 3 thresholds are equal, so just use θ. Meta operator 5 depends upon 3 meta variable equations: $\frac{d\omega_1}{dt} = \frac{du_4}{dt} - \frac{du_2}{dt}$; $\frac{d\omega_2}{dt} = \frac{du_6}{dt} - \frac{du_4}{dt}$; and $\frac{d\omega_3}{dt} = \frac{du_2}{dt} - \frac{du_6}{dt}$.

Meta Operator 5. *Decentralized Variables*

`if` $|\omega_1(s_1)| \geq \theta$ `AND` $|\omega_3(s_3)| \geq \theta$

$$\frac{du_2}{dt} = \frac{1}{2}\Big(\frac{du_4}{dt} + \frac{du_6}{dt}\Big).$$

`if` $|\omega_1(s_1)| \geq \theta$ `AND` $|\omega_2(s_2)| \geq \theta$

$$\frac{du_4}{dt} = \frac{1}{2}\Big(\frac{du_2}{dt} + \frac{du_6}{dt}\Big).$$

`if` $|\omega_2(s_2)| \geq \theta$ `AND` $|\omega_3(s_3)| \geq \theta$

$$\frac{du_6}{dt} = \frac{1}{2}\Big(\frac{du_2}{dt} + \frac{du_4}{dt}\Big).$$

Fig. 17 shows that meta operator 5 successfully heals the orbit on the same two pulse (Fig. 15) noise attack that caused meta operator 2 to fail (Fig. 16).[15]

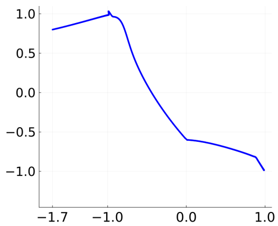

Fig. 17. Horizontal axis: u_1. Vertical axis: u_2. Meta Operator 5 Succeeds.

Variables u_2, u_4, and u_6 have cyclic interdependencies. In meta operator 5, $\frac{du_2}{dt}$ depends upon u_4, u_6; $\frac{du_4}{dt}$ depends upon u_2, u_6; and $\frac{du_6}{dt}$ depends upon u_2, u_4.[16]

Intuitively, when noise η_3 attacks u_4, u_2 and u_6 restore u_4. When η_2 attacks u_2, then u_4 and u_6 still agree so they can heal u_2. Depending upon the time, u_2, u_4, and u_6 play different roles. This is why six variables can ameliorate a two pulse attack on u_2 and u_4.

Keeping η_3's parameters fixed, if δ_2 lies in $(\frac{1}{2}, \frac{3}{4}]$, meta operator 5 successfully heals the noise. When $\delta_2 = 0.76$ and $\theta_1 = \theta_2 = 0.2$, a bifurcation[17] is about to occur. See Fig. 18. Operator 5 fails at $\delta_2 \geq 0.77$.

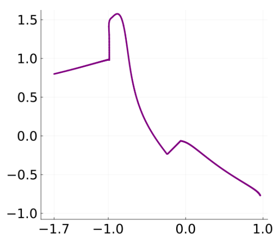

Fig. 18. Horizontal axis: u_1. Vertical axis: u_2. Meta Operator 5 Succeeds. $\delta_2 = 0.76$

The relative timing of noise pulses is also critical. If pulses η_2 and η_3 are separated in time, operator 5 is far more robust: when $\delta_2 = 1$, and η_2's time location $l = \frac{37}{2}$, meta operator 5 successfully heals the noise, as shown on the left side of Fig. 19.[18]

As the number of variables n increases beyond 8, there are more potential variable interdependencies that

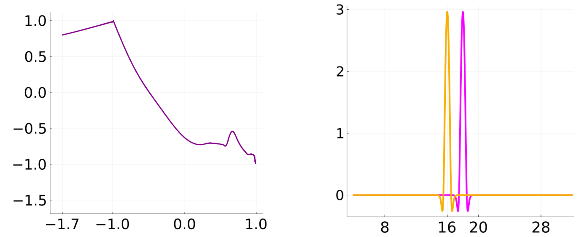

Fig. 19. Meta Operator 5 Succeeds. $\delta_2 = \delta_3 = 1$

can create robust healing. If $n \geq 8$, noise attacks require more pulses on multiple variables.[19] Also, it is far more challenging for Mallory to inject noise pulses so that on multiple variables their time locations overlap.

9. A Hardware Gedankenexperiment

A gedankenexperiment is proposed that tests feasibility of building meta operator 2. [20] Imagine two copies C_1 and C_2 of the flip-flop circuit in Fig. 1. Data input lines $D(t)$ and clock inputs CLK are the same for C_1 and C_2. Let $Q_i(t)$ and $\overline{Q_i}(t)$ be the two outputs of C_i, i in $\{1, 2\}$. There is an additional circuit S that receives $D(t)$, CLK, $Q_1(t)$, $\overline{Q_1}(t)$, $Q_2(t)$, and $\overline{Q_2}(t)$ as inputs. $Q_1(t)$ and $\overline{Q_1}(t)$ correspond to signal variables u_1 and u_2. $Q_2(t)$ and $\overline{Q_2}(t)$ correspond to healing variables u_3 and u_4. C_1 also receives outputs from S.

Meta operator 2's definition describes how S should behave. For i in $\{1, 2\}$, S should compute $\frac{d\omega_i}{dt} = \frac{du_{i+2}}{dt} - \frac{du_i}{dt}$. If thresholds θ_1 or θ_2 are exceeded, then S sends output that self-modifies circuit C_1. The final part of a gedankenexperiment tests under what conditions S can self-modify C_1 so that a noise injection doesn't disrupt C_1's orbit, per section 7.

10. Summary & Research Questions

A previously proposed differential equation model of a D flip-flop was our starting point. A noise attack was demonstrated that can disrupt a flip-flop orbit. Meta variables and meta operators were introduced as a mathematical tool for building self-modifiable differential equations. We showed how meta operators can heal a flip-flop orbit, disrupted by a noise injection, and identified that signal functionality and healing should be decentralized over standard and meta variables. A gedankenexperiment was proposed on how to build a meta operator that can self-modify a circuit.

[15] In the simulations of Fig. 16 and Fig. 17, noise constants $K_{1,3} = K_{2,3} = 0$ and $K_{1,4} = K_{2,4} = 1$.

[16] If variable y *depends upon* variable x, write $x \to y$. From the 3 equations, there are six variable 2-cycles: all $u_i \to u_j \to u_i$ such that $i \neq j$ and $\{i, j\} \subset \{2, 4, 6\}$. There are six variable 3-cycles: all $u_i \to u_j \to u_k \to u_i$ such that sets $\{i, j, k\} = \{2, 4, 6\}$.

[17] A bifurcation occurs when the orbit does not cross $u_2 = u_1$.

[18] Per footnote 7, the peak amplitudes of η_2, η_3 correspond to 9 volts, which is substantially above a typical range of 1.2 to 5 volts for commercial flip-flops, yet meta operator 5 still heals the orbit.

[19] This means Mallory requires more energy.

[20] If meta operator 2 is feasible in hardware, then we predict that the implementation can be extended to more elaborate meta operators.

In digital hardware design and theoretical computer science, *the final result of a computation is a fixed point.* After a Turing machine halts [22], the final result is the machine's state and symbols stored on the tape. A Turing machine's halting configuration is represented by a fixed point. (See pages 390–392 in [11].)

Fixed points are simpler objects in dynamical systems theory. Besides fixed points, nonlinear systems can have periodic orbits, limit cycles, dense orbits, and chaotic attractors. It is unknown whether chaotic attractors can help build self-modifiability that is resistant to attacks. It is also unknown what role quantum physics can play in building self-modifiable dynamical systems.

Future research should explore these questions:

- Can self-modifiable differential equations be implemented in electronics? Can a hardware flip-flop be built that self-modifies to hinder noise attacks?
- Can a chaotic attractor decentralize signal and healing among multiple variables?
- Can bifurcation theory provide insight on how to design new meta operators?
- Can quantum entanglement help self-modify a quantum dynamical system?

Acknowledgments

We are extremely grateful for Edl Schamiloglu and Luis Valbuena's helpful answers to our questions; helpful comments from the peer reviewers; and Sam Malachowsky for calling attention to row hammering.

References

[1] Li C., Raghunathan A., Jha N.K. (2011) "Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system." *IEEE Intl. Conf. on e-Health Networking, Applications and Services*.

[2] Halperin D., Heydt-Benjamin T.S., et. al. (2008) "Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses." *IEEE Symposium on Security and Privacy*.

[3] Korak T., Hoefler M. (2014) "On the Effects of Clock and Power Supply Tampering on Two Microcontroller Platforms." *Workshop on Fault Diagnosis and Tolerance in Cryptography*. 8–17, Sept. 2014.

[4] Giechaskiel I., Rasmussen K.B. (2019) "Taxonomy and Challenges of Out-of-Band Signal Injection Attacks and Defenses." CoRR abs/1901.06935. `https://arxiv.org/abs/1901.06935`

[5] Valbuena L., Heileman G.L., Hemmady S., Schamiloglu E. (2019) "Simplified Flip-Flop Gate Model for EEMI Injection." *Intl. Conf. on Electromagnetics in Advanced Applications*. 845–850.

[6] Horstmann J.U., Eichel H.W., Coates R.L. (1989) "Metastability behavior of CMOS ASIC flip-flops in theory and test." *IEEE Journal of Solid-State Circuits*. 24(1), 146–157, Feb. 1989.

[7] Tsividis Y. (1999) *Operation and Modeling of The MOS Transistor*. Oxford Univ. Press. 2nd Edition.

[8] Mitra S., et. al. (2007) "Soft Error Resilient System Design through Error Correction." *IFIP Intl. Federation for Info. Processing*. 249, Springer, 143–156.

[9] Fiske M.S. (2007) "Active Element Machine Computation." U.S. Patent 8,712,942. April 24, 2007.

[10] Fiske M.S. (2019) "Quantum random self-modifiable computation." Logic Colloquium 2019. Prague, Aug. 11–16. *Bulletin of Symbolic Logic*. 25(4), 510–511.

[11] Fiske M.S. (2021) "Random Self-Modifiable Computation." *Trans. on Computational Science and Computational Intelligence*. Springer Nature, 375–393.

[12] Fiske M.S. (2023) "Dynamical Systems that Heal." *Proceedings of the 56th Hawaii Intl. Conf. on System Sciences*. Lahaina, Maui. 6685–6694, Jan. 3, 2023.

[13] Wikipedia. (2024) *Hybrid Systems*. `bit.ly/47TWjmR`

[14] Lygeros J. (2004) *Lecture Notes on Hybrid Systems*. `https://aemea.org/HICSS2024/ref/lygeros.pdf`

[15] Friedberg E.C., Walker G.C., et. al. (2006) *DNA Repair and Mutagenesis*. ASM Press, 2nd Edition.

[16] Kampis G. (1991) *Self-Modifying Systems in Biology and Cognitive Science*. Pergamon Press.

[17] Valbuena L. (2023) Email. April-September 2023.

[18] Valbuena L. (2020) *Software Execution under Extreme Electromagnetic Interference*. Ph.D. Thesis. `digitalrepository.unm.edu/ece_etds/553/`

[19] Bezanson J., et. al. (2017) "Julia: A Fresh Approach to Numerical Computing." *SIAM review*. 59(1), 65–98.

[20] Shannon C.E. (1948) "A Mathematical Theory of Communication." *Bell Technical J*. 27(3), 379–423.

[21] Hamming R.W. (1950) "Error Detecting and Error Correcting Codes." *Bell Technical J*. 29(2), 147–160.

[22] Turing A.M. (1936) "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proc. London Math Society*. Series 2. 42 (3, 4), 230–265.

[23] Tao T. (2011) *An Introduction to Measure Theory*. Volume 126. American Mathematical Society.

[24] Munkres J.R. (1975) *Topology*. Prentice-Hall.

[25] Thue A. (1914) "Probleme über Verändergungen von Zeichenreihen nach gegebenen Regeln." *Christiana Videnskabs-Selskabs Skrifter*. I. 10.

[26] Herrero-Collantes M., Garcia-Escartin J.C. (2017) "Quantum random number generators." *Reviews of Modern Physics*. 89(1), 015004, APS.

[27] Hopcroft J.E., Ullman J.D. (1979) *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley.

Appendix

The appendix describes self-modifiable differential equations that can add new variables and new equations. Two notions are fundamental:

1. A formal language specifies how to self-modify a differential equation with *meta operators*.

2. A *meta variable* helps detect an event. A detectable event triggers an execution of a *meta operator*. A meta operator alone is not sufficient for defining self-modifiability. A self-modifiable dynamical system must also know at what time a meta operator executes.

10.1. Meta Variables

A *meta variable*, a *physically detectable event*, and a *meta execution time* are defined. *Standard variables* are variables that occur in an ordinary or partial differential equation. x and t are standard variables in
$$i\hbar\frac{\partial\Psi(x,t)}{\partial t} = \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x,t)\right]\Psi(x,t).$$

$\omega_1, \dots \omega_n$ represent meta variables, and $u_1, \dots, u_n$ are standard variables.[21] $\dfrac{d\omega_i}{dt} = f_i(u_1, \dots, u_n)$ is a *meta equation*, where each f_i is a function.

Let X be a measurable [23], topological space [24], where derivatives exist. At time t, a standard variable's value $u_i(t)$ and a meta variable's value $\omega_i(t)$ both lie in X. Let $f : X \times X \times [0, \infty) \to X$ be a function, where f's arguments are $\omega(t)$, $\frac{d\omega}{dt}$ and t, respectively. Call $\mathcal{C} \subseteq X$ a *detectable set*. A *physically detectable event* occurs if
$$\int_{t=0}^{t=s} f\Big(\omega(t),\ \frac{d\omega}{dt},\ t\Big)\, dt \quad \text{lies in } \mathcal{C} \text{ at time } s. \qquad (***)$$

Define *meta execution time* τ_ω as the infimum[22] of all times s that satisfy (***). A meta operator $\mathcal{M}_\omega$, bound to meta variable ω, executes at time τ_ω.

10.2. Meta Operators

Meta operators help build self-modifiable differential equations. A *system* is a set of differential equations, e.g., $\mathcal{S} = \{\frac{du_1}{dt} = 0,\ \frac{du_2}{dt} = 0,\ \frac{du_3}{dt} = u_1 + u_2 - u_1u_2 - u_3\}$. *Create operator* $\mathcal{C}$ creates an empty system, and assigns a name with syntax $\mathcal{C}$(`time`, `name`). $\mathcal{C}(0, \mathcal{S})$ creates an empty system $\mathcal{S} = \{\}$ at time 0.

Initialize operator $\mathcal{I}$ declares a variable with a name and its type; assigns an initial value; and places the variable in a system. $\mathcal{I}$'s syntax is $\mathcal{I}$(`time`, `variable_name`, `variable_type`, `initial_value`, `system_name`). For example, $\mathcal{I}(0, u_i$, `standard`, 0, $\mathcal{S})$ creates u_i at time 0; defines u_i as a standard variable; and assigns u_i the initial value of 0 (i.e., $u_i(0) = 0$). Argument $\mathcal{S}$ places variable u_i in system $\mathcal{S}$.

Adjoin operator $\mathcal{A}$ adjoins a new differential equation to a system. $\mathcal{A}$'s syntax is $\mathcal{A}$(`time`, `equation`, `system_name`). $\mathcal{A}$ is executed at a time specified by the first argument. In some cases, `time` is explicitly stated. In other cases, `time` is the greatest lower bound of all times s which satisfy condition (***) in section 10.1. For example, execute 7 meta operators shown below:
$\mathcal{C}(-1, \mathcal{S})$ $\quad \mathcal{I}(0, u_1$, `standard`, $0, \mathcal{S})$
$\mathcal{A}(0, \frac{du_1}{dt} = 0, \mathcal{S})$ $\quad \mathcal{I}(0, u_2$, `standard`, $0, \mathcal{S})$
$\mathcal{A}(0, \frac{du_2}{dt} = 0, \mathcal{S})$ $\quad \mathcal{I}(0, u_3$, `standard`, $0, \mathcal{S})$
$\mathcal{A}(0, \frac{du_3}{dt} = u_1 + u_2 - u_1u_2 - u_3, \mathcal{S})$. Afterward, system $\mathcal{S} = \{\frac{du_1}{dt} = 0,\ \frac{du_2}{dt} = 0,\ \frac{du_3}{dt} = u_1 + u_2 - u_1u_2 - u_3\}$.

Replace operator $\mathcal{R}$ replaces a variable with an equation or variable, or $\mathcal{R}$ replaces an equation with another equation. $\mathcal{R}$'s syntax is $\mathcal{R}$(`time`, `old_exp`, `new_exp`, `grammar`, `system_name`). The argument `time` behaves the same as `time` in the adjoin operator. Sometimes the 2nd argument `old_exp` represents the current variable that will be replaced by a new variable or equation, indicated by the 3rd argument `new_exp`. Sometimes `old_exp` represents an equation that will be replaced by a new equation `new_exp`. The 4th argument `grammar` is a pattern matching scheme. For replacement to occur, an expression in `old_exp`, must be accepted by a grammar, specified in `grammar`. It may be a semi-Thue grammar [25].[23] If `grammar` is $\emptyset$ or omitted, a replacement occurs at time specified by `time`.

10.3. Variable Spaces

Meta operators can remove and add variables to equations. *Variables spaces* specify how meta operators add new variables or remove variables. Variable spaces are useful when they create a new variable, by measuring a quantum random event [26], which can help self-modifiable differential equations evolve and increase their complexity as time proceeds.

Let $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{R}^{\geq 0}$ be the counting, real numbers and non-negative reals, respectively. Define *variable space* $\mathcal{V} = \{u_1, u_2, \dots, u_n, \dots\}$ in order to add a new standard variable u_i to a differential equation. Define *dual variable space* $\overline{\mathcal{V}} = \{\overline{u}_i : i \in \mathbb{N}\}$ to remove an existing standard variable u_j in $\mathcal{V}$ from an equation. λ is the empty variable.[24] Set $\mathcal{W} = \{\lambda\} \cup \mathcal{V} \cup \overline{\mathcal{V}}$.

Define a *variable operator* $\Phi : \mathbb{R}^{\geq 0} \to \mathcal{W}$ that adds or removes variables from a system $\mathcal{S}$. If $\Phi(t_i) = u_i$, then meta operator $\mathcal{I}(t_i, \Phi(t_i)$, `standard`, 0, $\mathcal{S})$ adds standard variable u_i to system $\mathcal{S}$ if u_i doesn't already exist in $\mathcal{S}$. For our purposes, a system is a set of ordinary or partial differential equations. For some systems that represent a physical model, a restriction can be placed on Φ: e.g., $\{r \in \mathbb{R}^{\geq 0} : \Phi(r) \in \mathcal{V} \cup \overline{\mathcal{V}}\}$ is *countable*.

[21] Variables have type *standard* or type *meta*.
[22] The infimum of a set of real numbers is the greatest lower bound.
[23] Also, see pages 220-223 in [27].
[24] λ is an analog of empty string ϵ in formal language theory [27].