# Design of Software Rejuvenation for CPS Security Using Invariant Sets

Raffaele Romagnoli*, Bruce H. Krogh** and Bruno Sinopoli***

*Dept. of Electrical and Computer Engineering, Carnegie Mellon University
**Software Engineering Institute, Carnegie Mellon University
***Department of Electrical and Systems Engineering, Washington University in St. Louis

2019 American Control Conference, July 10-12, 2019, Philadelphia, PA.

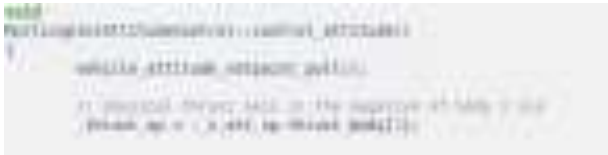
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0651

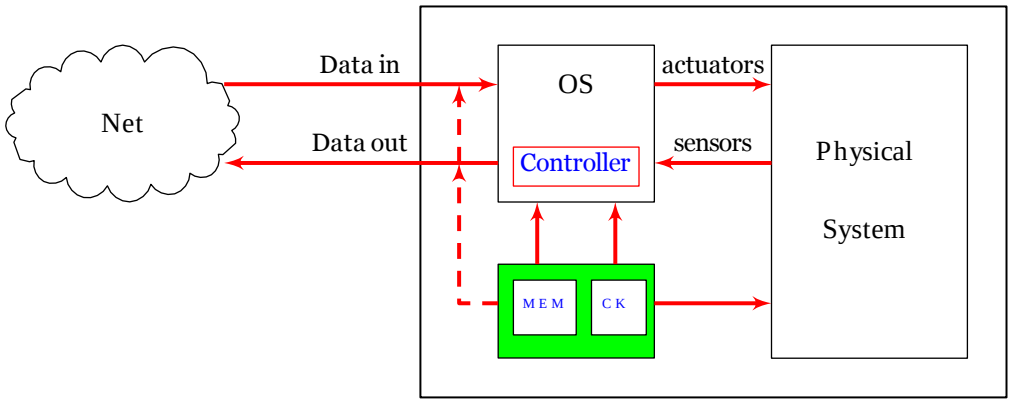
# Introduction to the Problem

## Cyber-Physical Systems

## Attack Model



## Protected Hardware/Software Arch.

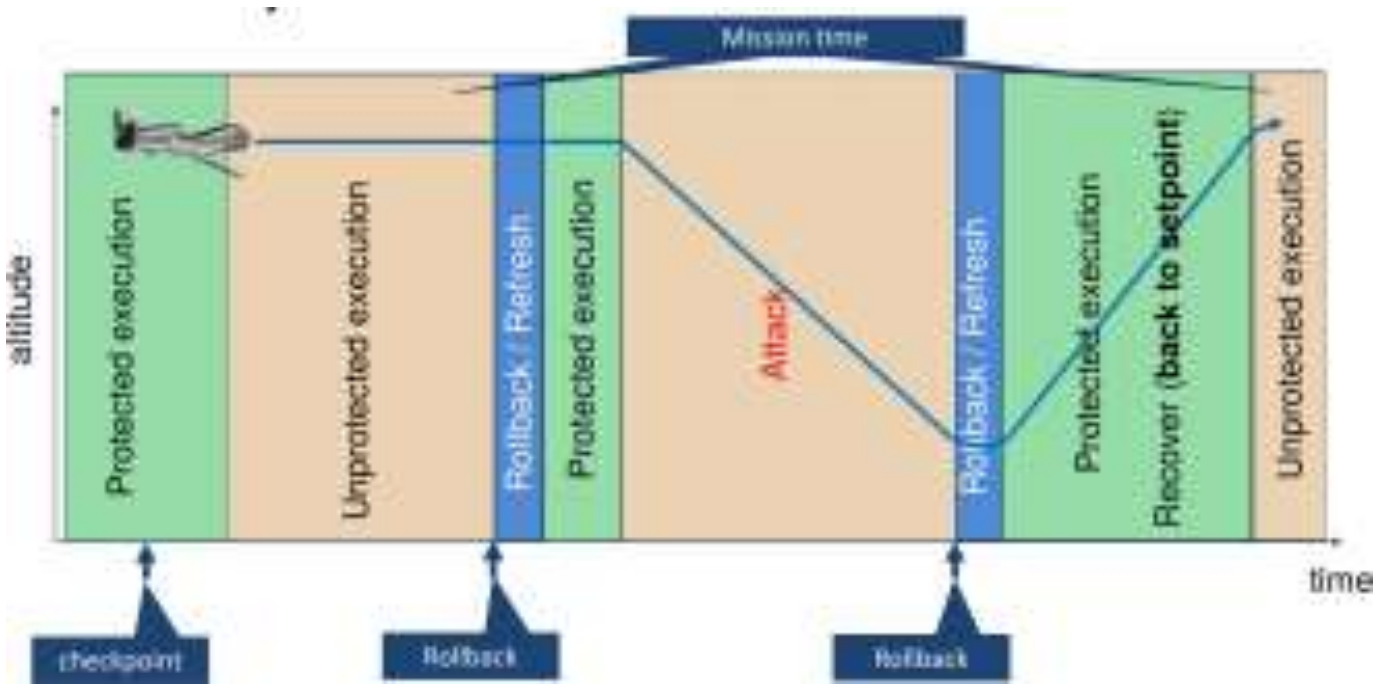


# Table of Contents

# Motivation and Goal

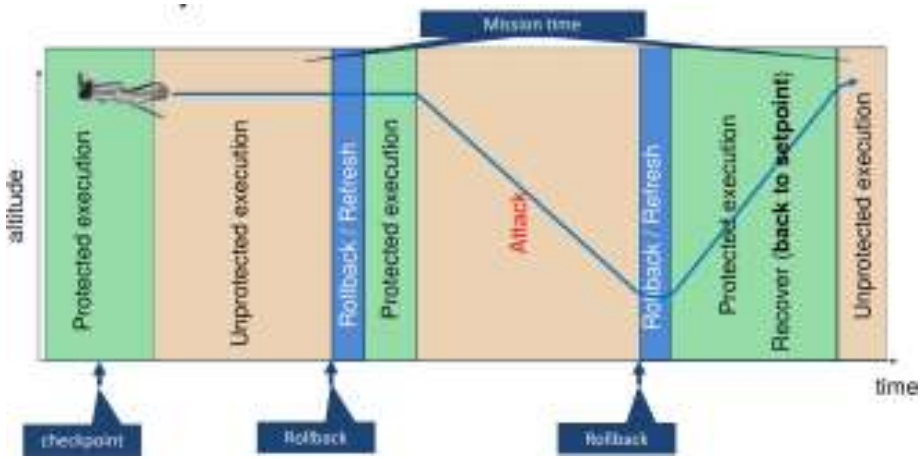
## Software Rejuvenation

**Software rejuvenation** (SR) protects **cyberphysical systems** (CSPs) against **cyber attacks** on the run time code by **periodically refreshing** the system with an uncorrupted software image.



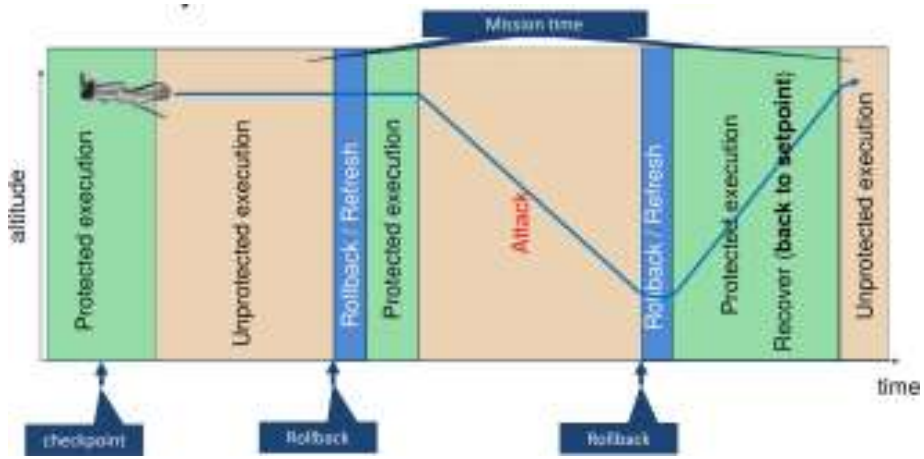
# Motivation and Goal

- How often rebooting the system?
- What are the effects on the control system?



# Motivation and Goal

- How often rebooting the system?
- What are the effects on the control system?

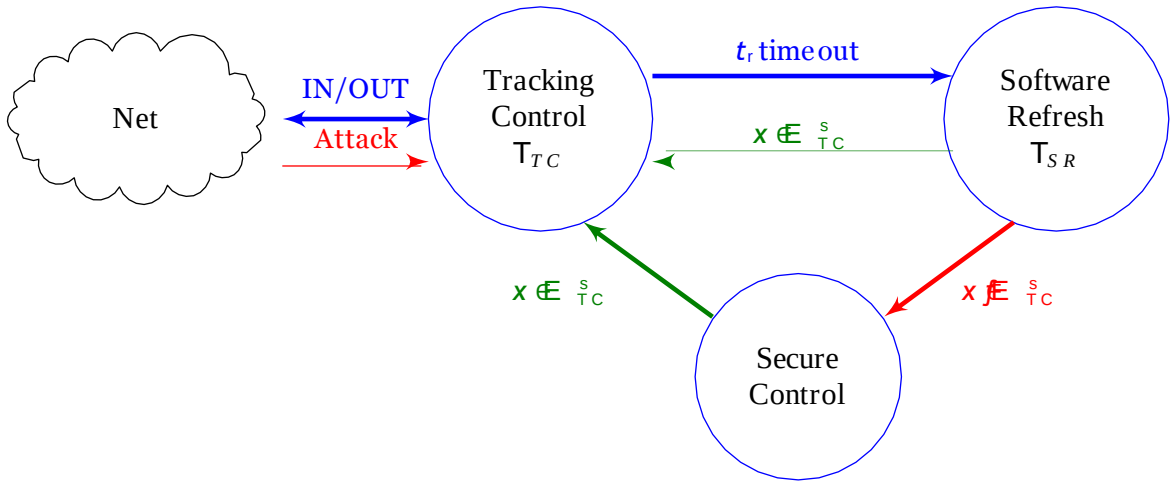



## Goal

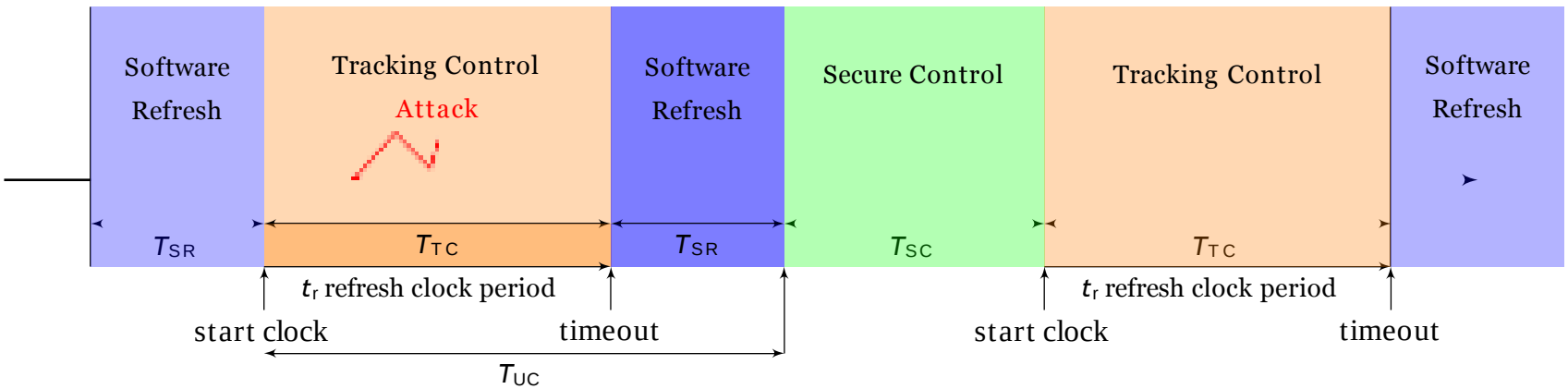
To propose a **secure tracking control** scheme based on **software rejuvenation** for nonlinear and linear systems and provide general conditions that guarantee the property of **safety** and **liveness**.

# Software Rejuvenation Operating Modes

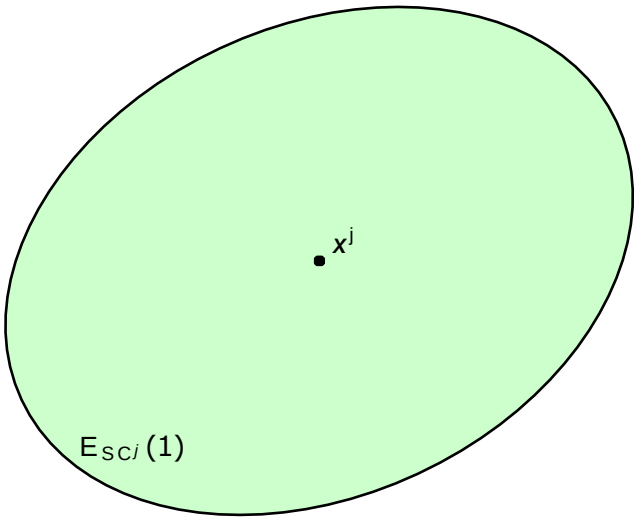
- Tracking Control (TC).
- Software Refresh (SR).
- Secure Control (SC).




## Timeline

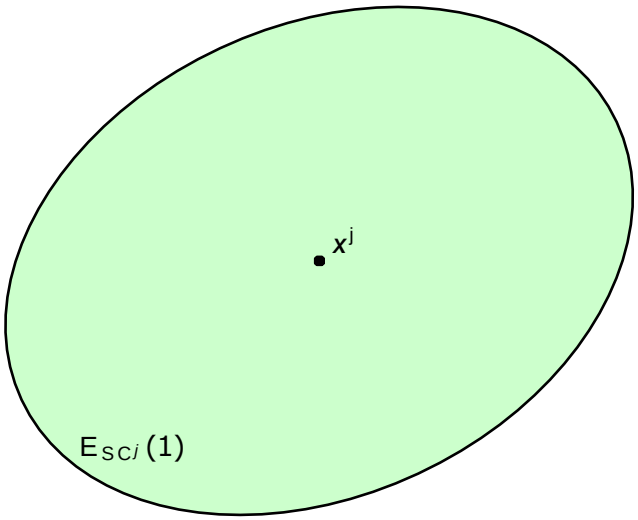


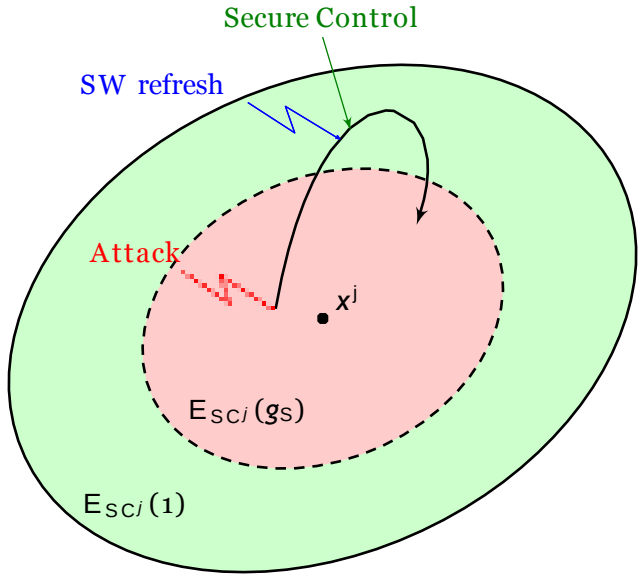
# Secure Control




Recoverable set: $E^j_{SC}(1)$

# Secure Control

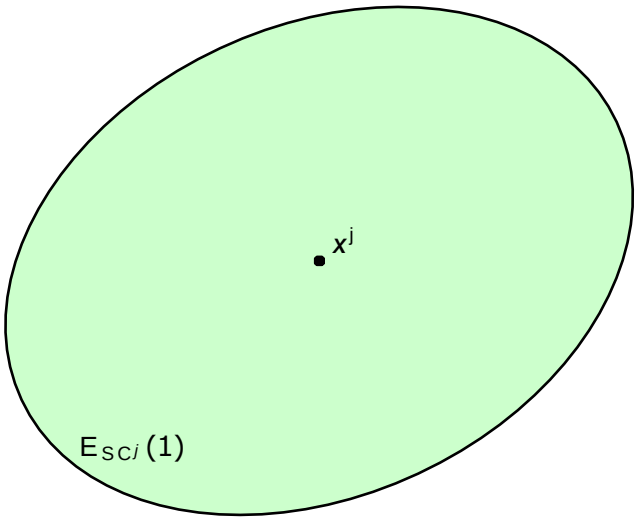


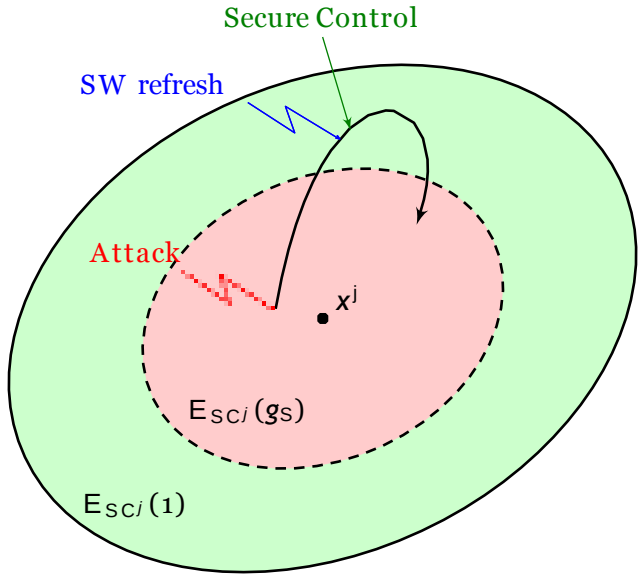




Recoverable set: $E^j_{SC}(1)$

Safety set: $E^j_{SC}(s_s) \subseteq s_s E^j_{SC}(1)$

# Secure Control




Recoverable set: $E^j_{SC}(1)$

Safety set: $E^j_{SC}(s_s) \subseteq s_s E^j_{SC}(1)$

Find $T_{UC}$, $s_s \in R$

$$R(T_{UC}; E^j_{SC}(s_s), U) \subseteq E^j_{SC}(1).$$

# Secure Control

**Controlled System**:

$$\dot{x} = f_\phi(x) = f(x, \phi(x))$$

where $\phi(x)$ is the state feedback controller.

**Lyapunov Function**:

$V_\phi : R^n \to R,\ N_{V_\phi}(x_{eq}) \subseteq N(x_{eq}), V(x_{eq}) = 0$ and

$\forall x \in N_{V_\phi}(x_{eq}) - \{x_{eq}\}$: (i) $V_\phi(x) > 0$, (ii)

$$\dot{V}_\phi(x) = \frac{\partial V}{\partial x} \cdot f(x_\phi) < 0$$

# Secure Control

**Controlled System**:

$$\dot{x} = f_\phi(x) = f(x, \phi(x))$$

where $\phi(x)$ is the state feedback controller.

**Lyapunov Function**:

$V_\phi : R^n \to R,\ N_{V_\phi}(x_{eq}) \subseteq N_\phi(x_{eq}), V_\phi(x_{eq}) = 0$ and

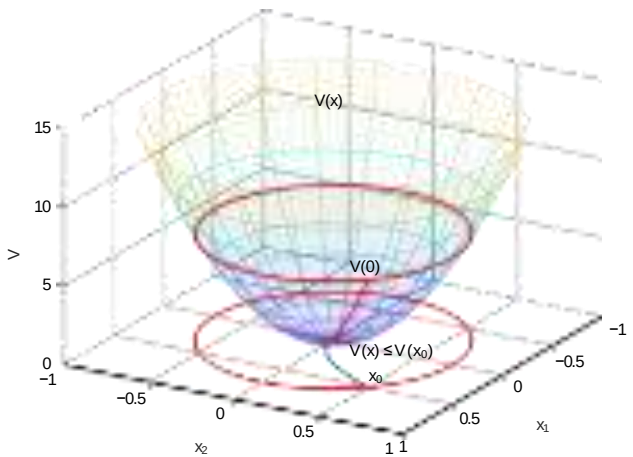
$\forall x \in N_{V_\phi}(x_{eq}) - \{x_{eq}\}$: (i) $V_\phi(x) > 0$, (ii)

$$\dot{V}_\phi(x) = \frac{\partial V}{\partial x} \cdot f_\phi(x) < 0$$

**Lyapunov level set**: For $s > 0$,

$$E_\phi(s) = \{x \in N_{V_\phi}(x_{eq}) | V_\phi(x) \le s\}. \qquad s \le 1$$

For any $0 < s \le 1$, $E_\phi(s)$ is an *invariant set*: $\forall t > 0, R(t; E_\phi(s), \phi) \subseteq E_\phi(s)$




Electrical & Computer ENGINEERING

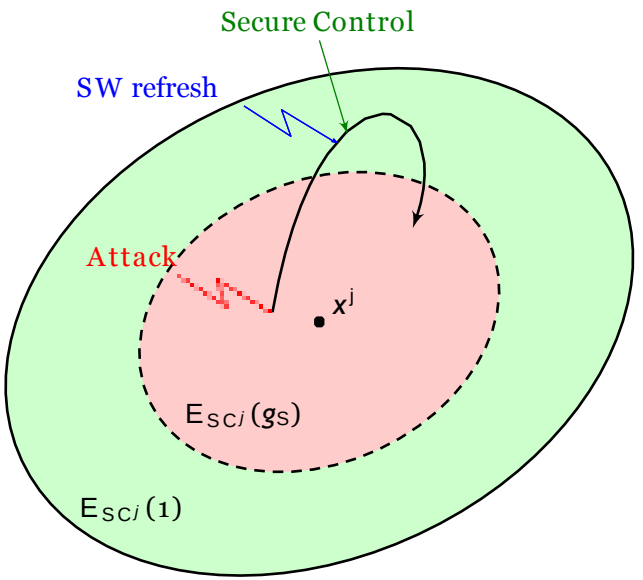
Carnegie Mellon University

# Secure Control

## Proposition 1

Given $\dot{x} = f_\phi(x) \coloneqq f(x, \phi(x))$ with stabilizing controller $\phi$ for equilibrium state $(x_{eq}, \phi(x_{eq}))$ and Lyapunov function $V_\phi(x)$ as defined above, given $s > 0$ for any $s < s^j \leq 1$ $\exists \gamma > 0$ $s.t.$ $\forall t \geq (s^j - s)\gamma^{-1}$,

$$R(t; E_\phi(s^j), \phi) \subseteq E_\phi(s).$$

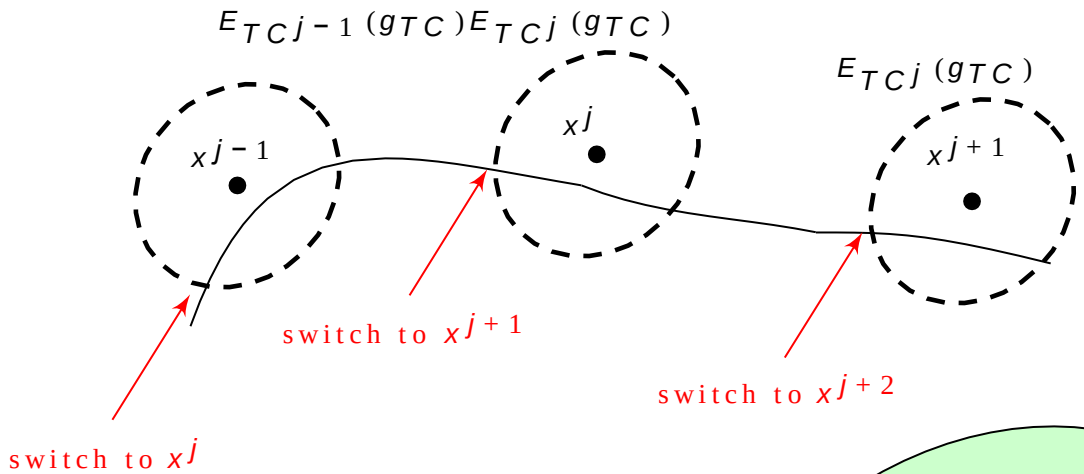



## Proposition 2

For any $U \subseteq U$ and any $0 < s < s^j \leq 1$,
$\exists T_U > 0$ $s.t.$ $R(t; E_\phi(s), U) \subseteq E_\phi(s^j)$ $\forall t < T_U$.

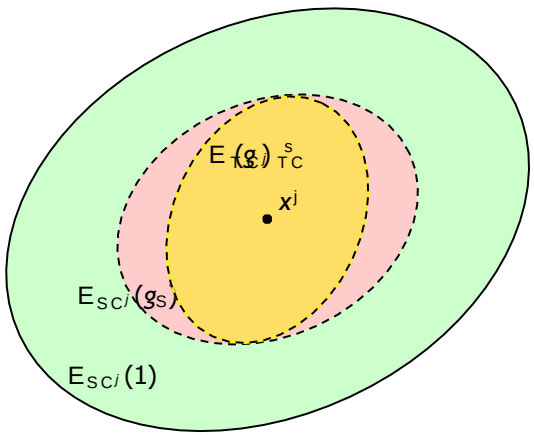
# Tracking Control

**Idea**: sequence of equilibrium points $x_0, x_1, \ldots, x^{j-1}, x^j, \ldots$




## Goals

- Safety transition $x^j \mapsto x^{j+1}$
- Liveness (normal operation)

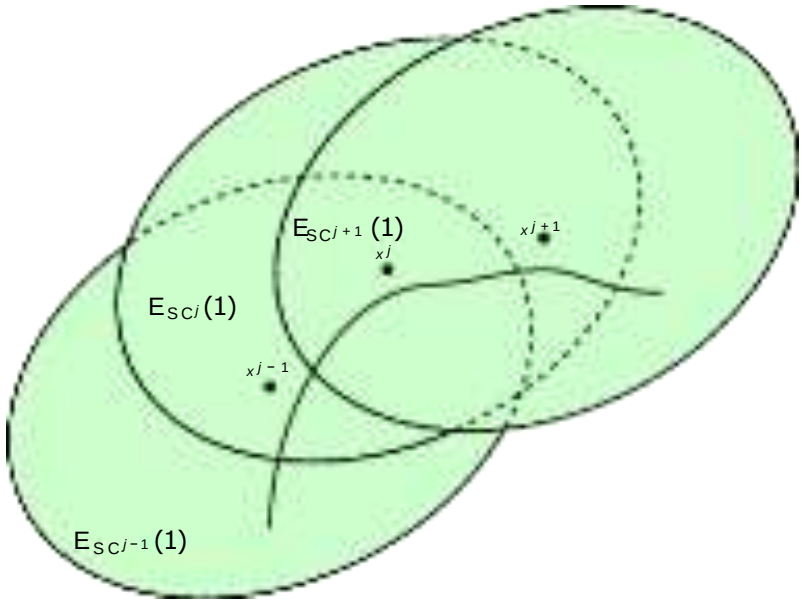



$$E_{TCj}(s_{TC}) = (s_{TC})E_{TCj}(s_{TC}) \subseteq E_{TCj}(s_{TC})^{s} \quad (s_{TC}) < 1.$$

# Tracking Control

## Safety

$$x(t) \in \bigcap_{j=0}^{J} E_{SCj}(1)$$

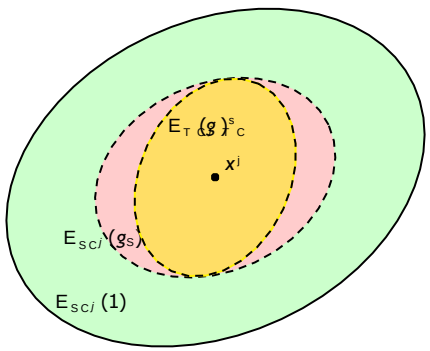



**Theorem**. The system is safe under software rejuvenation if the following conditions are satisfied:

i. $x(0) \in E_{TC1}(s^s_{TC})$;

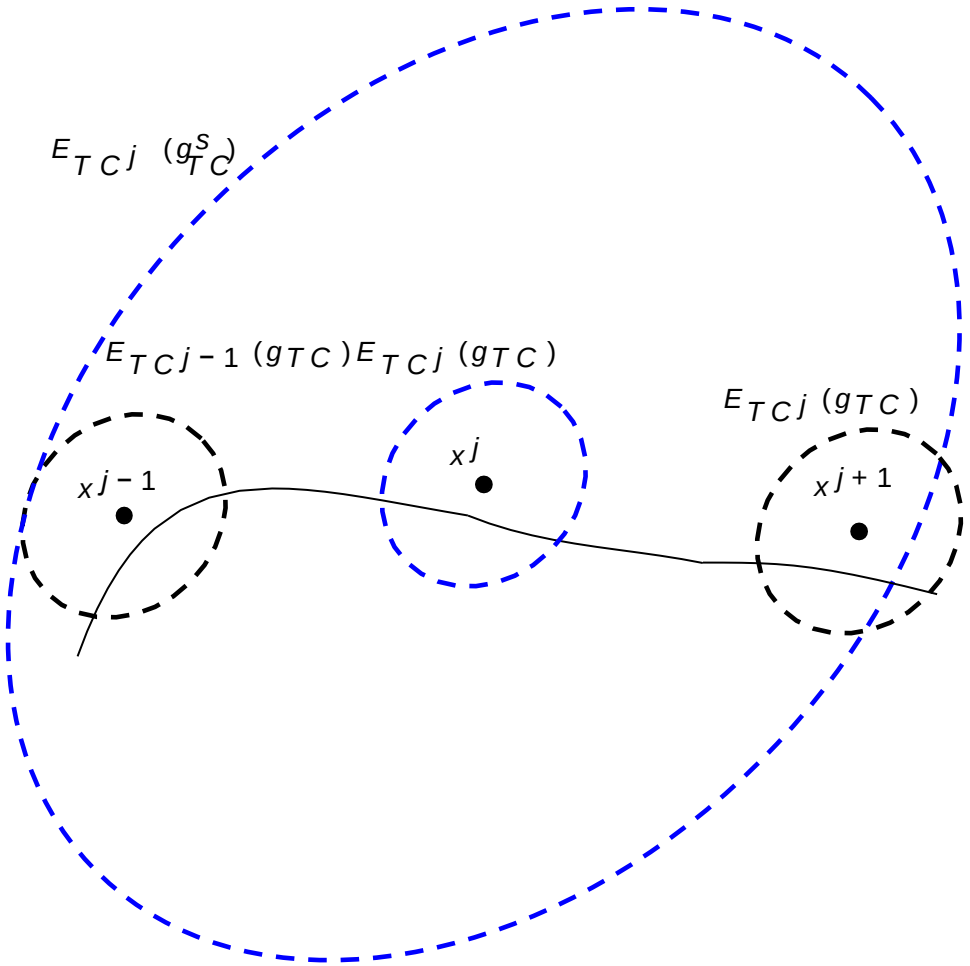
ii. $E_{TCj}(s^s_{TC}) \subset E_{SCj}(s_{SC})$;

iii. $R(t; E_{SCj}(s_{SC}), U) \subseteq E_{SCj}(1)\ \forall t \in [0, T_{TC} + T_{SR}]$;



# Tracking Control

iv. $E_{TC\,j}(s_{TC}) \subset E_{TC\,j+1}(s^{S}_{TC})$.



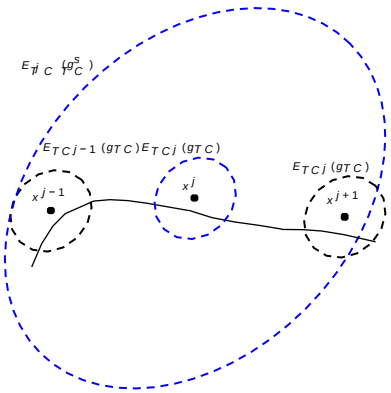
# Tracking Control

## Liveness

Given reference points $x^1, \ldots, x^J$, the system is *live* if there exists a sequence of times $t_1, \ldots, t_J$ where $0 < t_1 < \cdots < t_J < \infty$ such that

$$x(t) \in E_{TCj}(s_{TC}^S) \quad \forall t \in [t_j, t_{j+1}), j = 0, \ldots, J,$$

where $t_0 = 0$ and $t_{J+1} = \infty$.

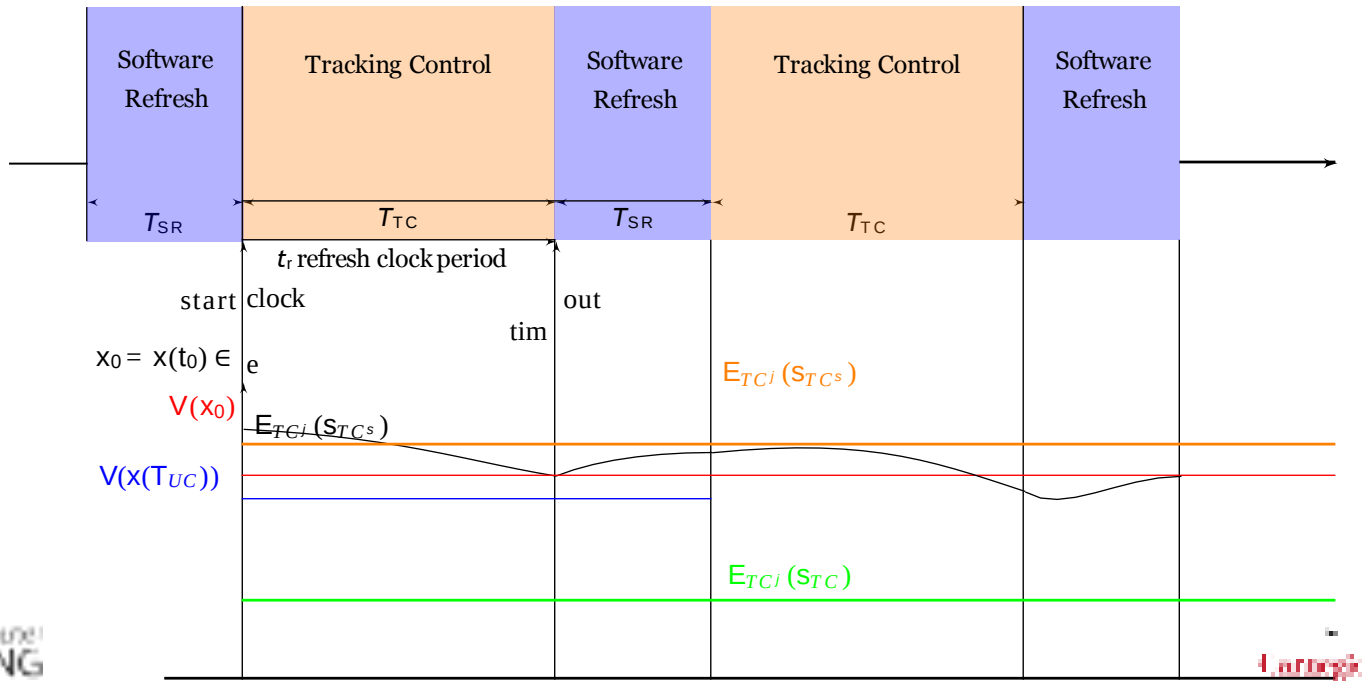



**Assumptions**: no attack, the control input during $T_{SR}$ is equal to the last value provided before software refresh.

# Tracking Control

**Theorem**. When there are no cyber-attacks, the system is live under software rejuvenation if, in addition to the conditions in Th. (safety), $0 < s_{TC} < s^s_{TC}$ and $\exists \delta > 0$ such that $\forall x \in E^j_{TCj}(s^s_{TC}) - E_{TCj}(s_{TC})$ and $\forall x^{jj} \in \underset{0 \le t \le T_{SR}}{} R(t; x(T_{TC}; x^j, TC^j), U)$,

$$V_{TCj}(x^j) - V_{TCj}(x^{jj}) \ge \delta$$



# Tracking Control

## Linear Systems

$$E_{TCj}(s_{TC}) = \{ x \mid \|(x - x_j)\|^2_{P_{TCj}} \leq s_{TC} \}$$

## Safety

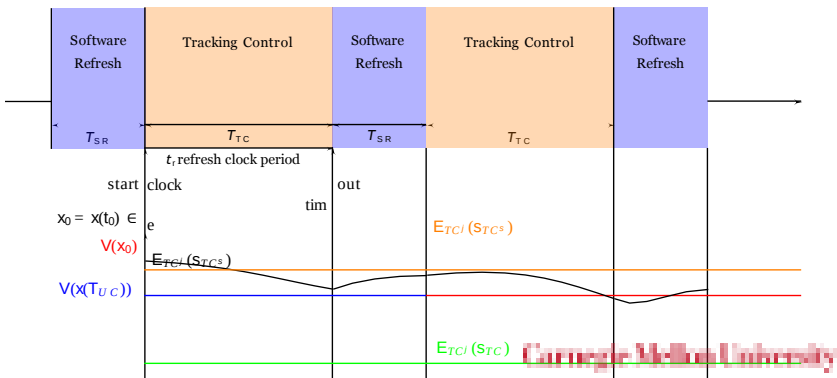
$$\forall x \in E_{TCj}(s_{TC}) \Rightarrow x \in E_{TCj+1}(s_{TC}).^{s}$$

## Liveness

$$\tilde{A}_j^T P_{TCj} \tilde{A}_j - P_{TCj} < 0.$$

where $\tilde{A}_j = \hat{A}_j \bar{A}_j$, $\bar{A}_j = e^{A_{SCj} T_{TC}}$, $\hat{A}_j = (A_d - B_d K^j)$.
$A_d = e^{A T_{SR}}$ and $B_d = \int_0^{T_{SR}} e^{A(T_{SR} - \tau)} B d\tau$.

$$\|x(T_{UC})\|^2_{P_{TCj}} < \|x_j(0)\|^2_{P_{TCj}}.$$



# Example: Quadrotor

- 6DOF → 12 state variables

Linear design

- Linearize at equilibrium;
- Assume full state available;
- LQ state feedback design;
- Reference point = Equilibrium point.

# Example: Quadrotor

- 6DOF → 12 state variables

Linear design

- Linearize at equilibrium;
- Assume full state available;
- LQ state feedback design;
- Reference point = Equilibrium point.

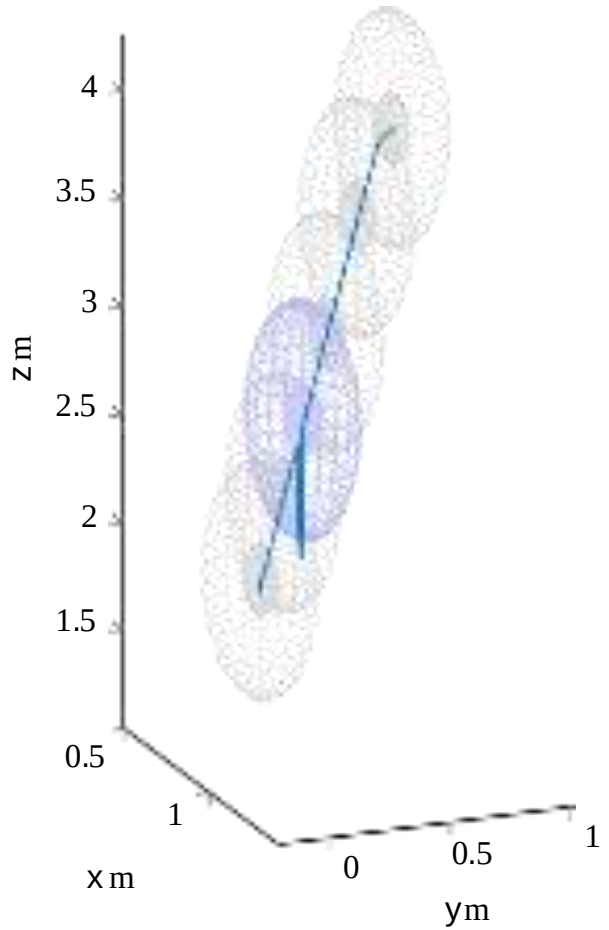
## Simulation

jMAVSim simulator. Turn-off attack.

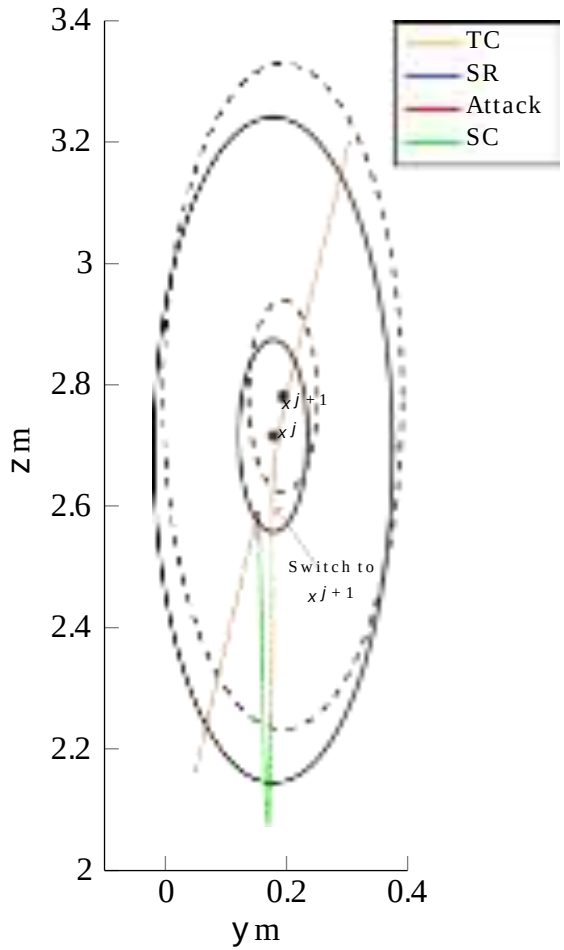
# Example: Quadrotor

# Example: Quadrotor

**Tracking**. JMAVSim simulator. Turn-off attack. Projections of $E_{TCj}(s_{TCs})$ and $E_{TCj}(s_{TC})$




(a)




(b)

# Summary

- Overview of software rejuvenation
- Description of algorithm for tracking control systems
- Summary of theoretical results from control theory
- Demonstration in simulation for nonlinear drone application

**References**


- R.Romagnoli, B.H. Krogh, B. Sinopoli. **Design of SoftwareRejuvenation for CPS Security Using Invariant Sets**. IEEE 2019 American Control Conference.