Carnegie Mellon

Software Rejuvenation For Secure Tracking Control Of Cyber-Physical Systems

Raffaele Romagnoli†, Bruce H. Krogh††, Bruno Sinopoli†††

†Carnegie Mellon University, ††Software Engineering Institute CMU, †††University of Washington in St Louis

Motivation

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

During SR, the system is in **open loop**, then this mechanism of protection may imply severe **issues** from the **control** perspective, such as **stability** and the inability to complete a mission (**tracking** performances).

Goal

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

Introduction

Software Engineering

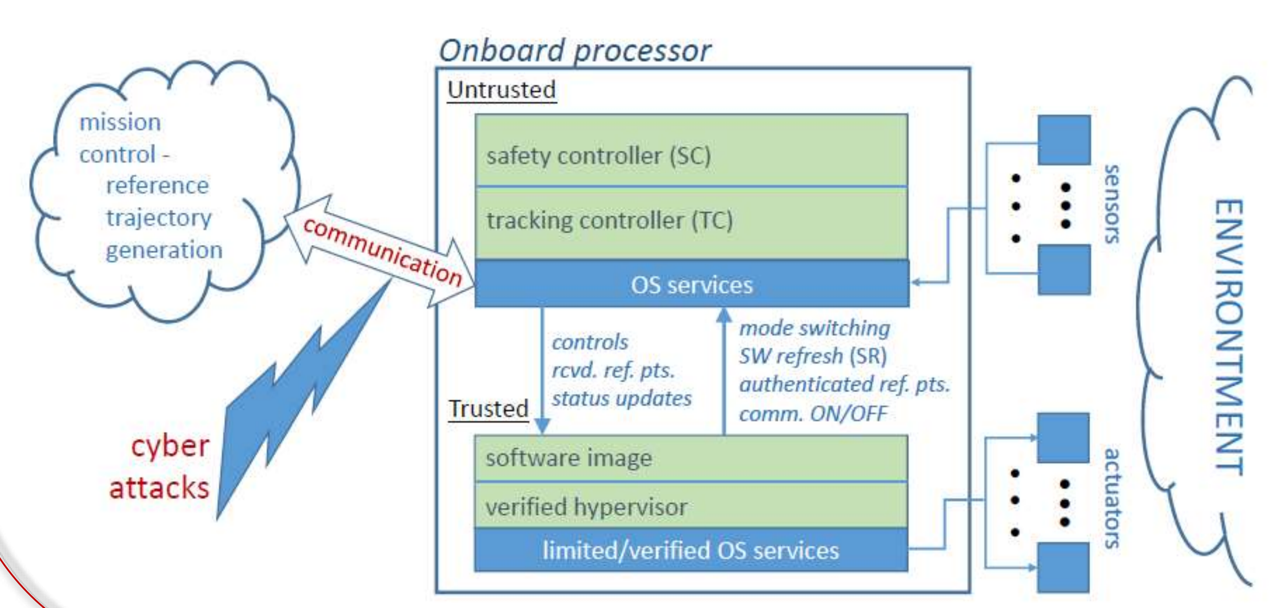
"...Software Rejuvenation is a periodic preemptive rollback of continuously running applications to prevent failures in the future." [Huang et al, 1995]

- I reboot;
- I restart the application from a clean internal state.

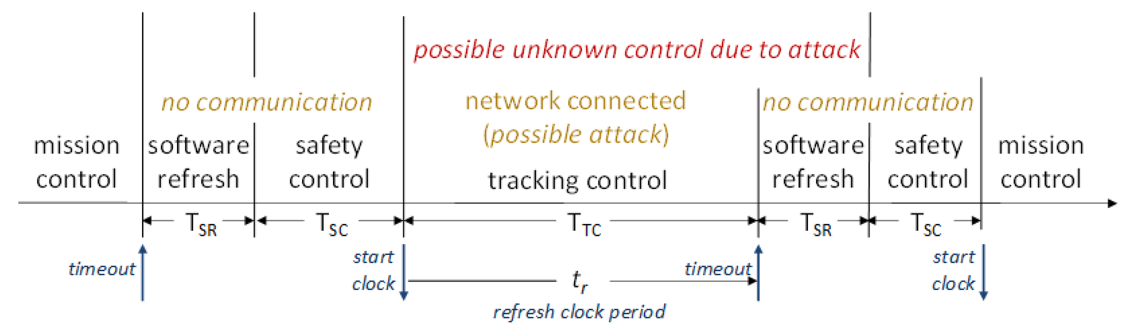
Control System

- I Fault Tolerant Control;
- I Secure Control of CPS [Abdi et al 2018, Arroyo et al 2017]

Attack Model and Architecture



How SR works



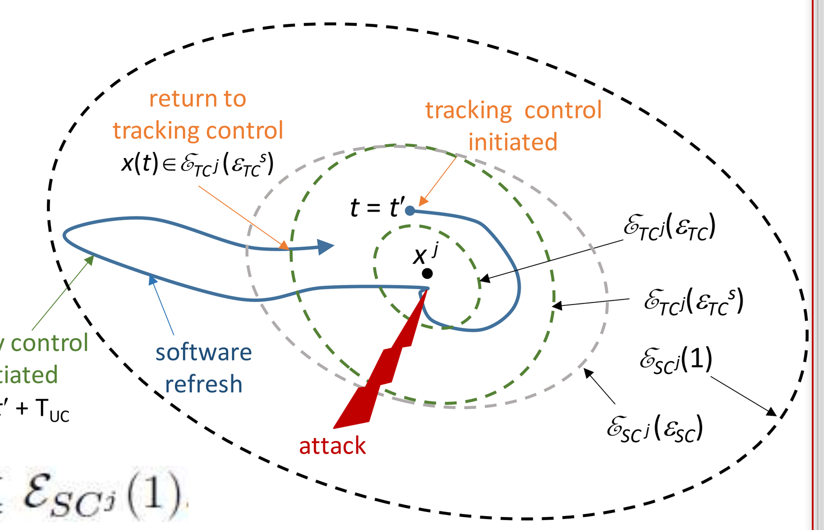
The refresh clock period has to guarantee that for any control input, the system cannot leave the safety set.

Safety Control

$\mathcal{E}_{SC^j}(1)$

$\mathcal{E}_{SC^j}(\epsilon_{SC})$

$\mathcal{R}(T_{UC};\mathcal{E}_{SC^j}(\epsilon_{SC}),U) \subseteq \mathcal{E}_{SC^j}(1)$



Lyapunov Functions and Invariant Sets

Controlled system: $\dot{x} = f_\varphi(x) \triangleq f(x,\varphi(x)),$

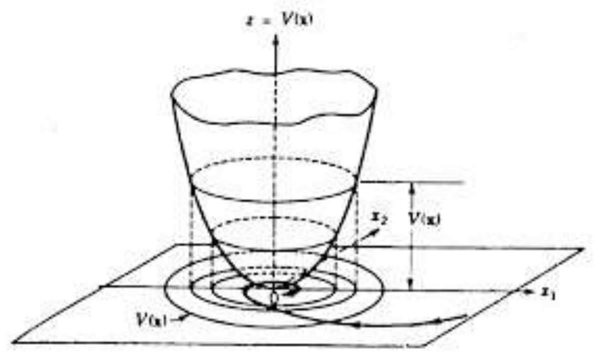
Lyapunov Function: $V_\varphi : \mathbb{R}^n \to \mathbb{R}$, $V_\varphi(x_{eq}) = 0$, $\dot{V}_\varphi(x) = \frac{\partial V}{\partial x} \cdot f_\varphi(x) < 0.$

Lyapunov level Set: $\mathcal{E}_\varphi(\epsilon) = \{x \in \mathcal{N}_{V_\varphi}(x_{eq}) | V_\varphi(x) \leq \epsilon\}$

Positively Invariant Set:

$\forall\ t > 0,\ \mathcal{R}(t;\mathcal{E}_\varphi(\epsilon),\varphi) \subseteq \mathcal{E}_\varphi(\epsilon).$

$0 < \epsilon' < \epsilon,\ \mathcal{E}_\varphi(\epsilon') \subset \mathcal{E}_\varphi(\epsilon)$



Two Fundamental Propositions

Proposition 2.1: Given system (1) with stabilizing controller φ for equilibrium state $(x_{eq},\varphi(x_{eq}))$ and Lyapunov function $V_\varphi(x)$ as defined above, given $\epsilon > 0$ for any $\epsilon < \epsilon' \leq 1\ \exists\ \gamma > 0\ \ni \forall\ t \geq (\epsilon' - \epsilon)\gamma^{-1}$,

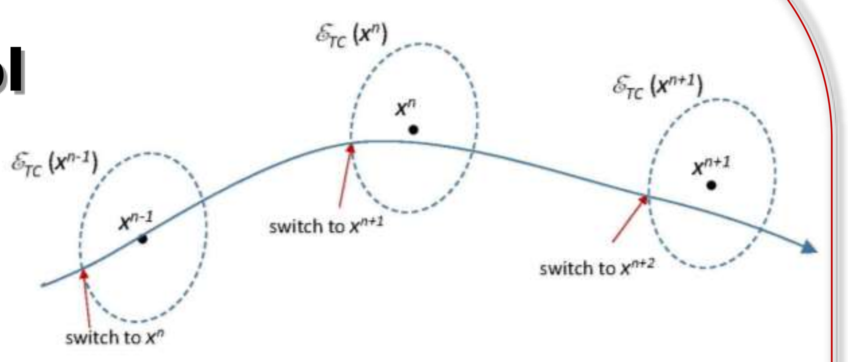
$$\mathcal{R}(t;\mathcal{E}_\varphi(\epsilon'),\varphi) \subseteq \mathcal{E}_\varphi(\epsilon). \quad (10)$$

Proposition 2.2: For any $U \subseteq \mathcal{U}$ and any $0 < \epsilon < \epsilon' \leq 1$, $\exists\ T_U > 0\ \ni\ \mathcal{R}(t;\mathcal{E}_\varphi(\epsilon),U) \subseteq \mathcal{E}_\varphi(\epsilon')\ \forall\ t < T_U$.

Tracking Control

$\mathcal{E}_{TC^j}(\epsilon^s_{TC}) \subseteq \mathcal{E}_{SC^j}(\epsilon_{SC})$

sequence of reference points $x^1,\ldots,x^J$



$$\mathcal{E}_{TC^j}(\epsilon_{TC}) \subset \mathcal{E}_{TC^{j+1}}(\epsilon'),\ \exists t^{j+1} > 0 \ni \forall\ t \geq t_{j+1},$$
$$\mathcal{R}(t;\mathcal{E}_{TC^j}(\epsilon_{TC}),TC^{j+1}) \subset \mathcal{E}_{TC^{j+1}}(\epsilon_{TC}).$$

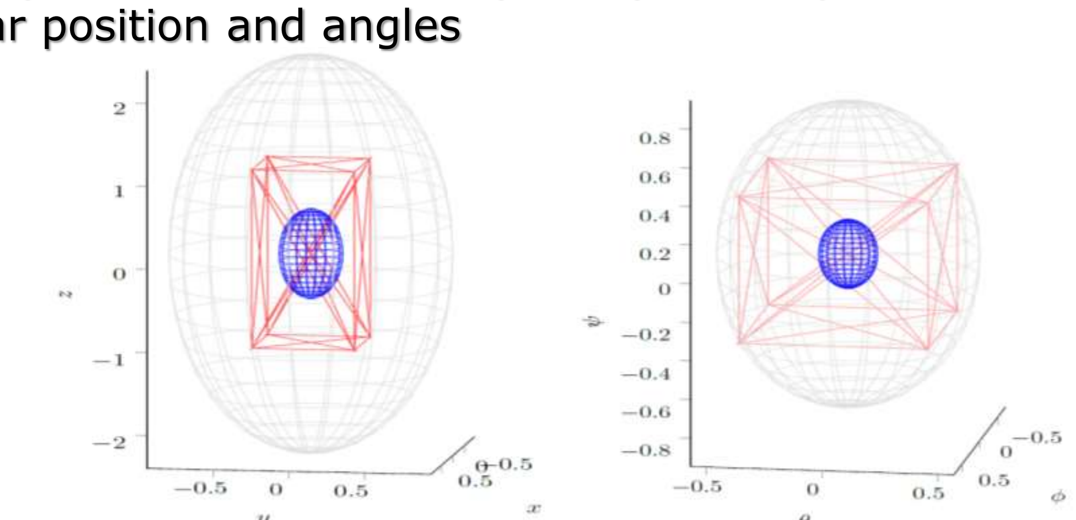
Safety and Liveness

- *safety*: when $x^j \to x^{j+1}$, the system has to be in $\mathcal{E}_{TC^{j+1}}(\epsilon^s_{TC})$;
- *liveness*: in presence of software refresh, the tracking controller has to drive the system to $\mathcal{E}_{TC^j}(\epsilon_{TC})$.

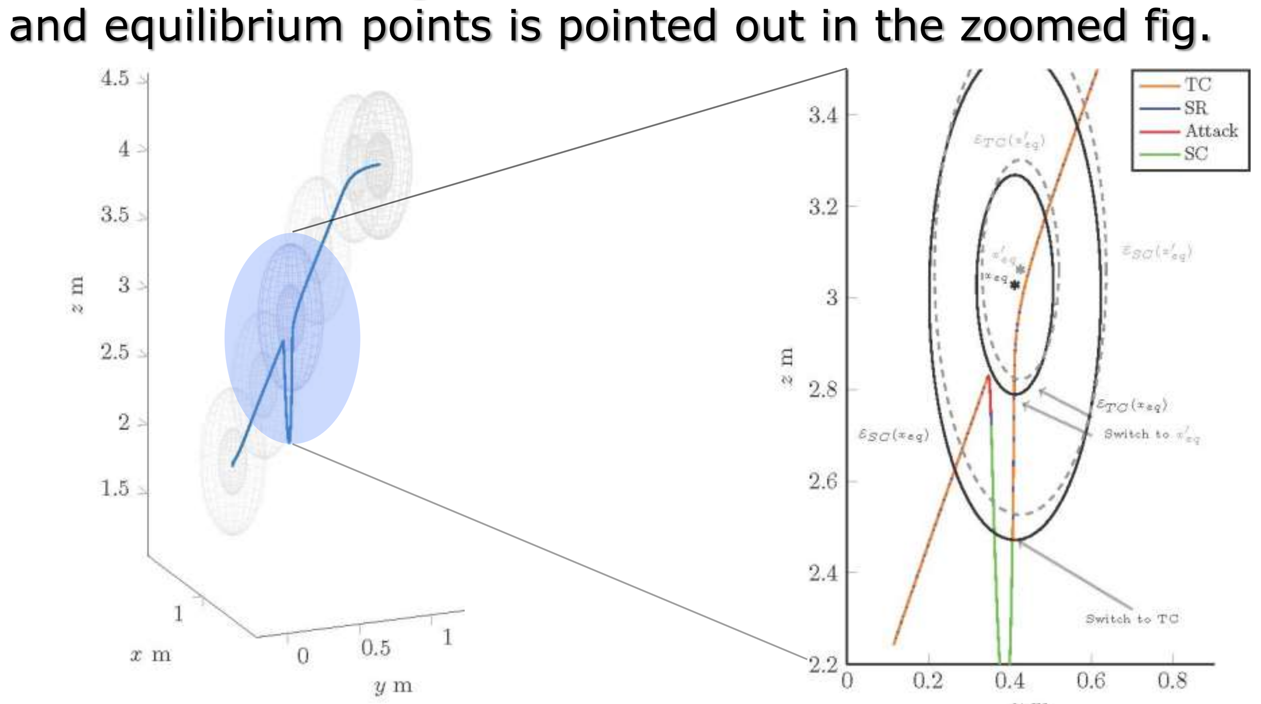
Example: quadrotor

- 6 DOF quadrotor using the PX4 jMAVSim quadrotor simulator
- Linearized model

Safety sets and Reachability Analysis for quadrotor's linear position and angles



Simulation results in presence of an attack. Details about the switching between the serveral controllers and equilibrium points is pointed out in the zoomed fig.



Carnegie Mellon University
Software Engineering Institute

Electrical & Computer ENGINEERING

Copyright 2019 Carnegie Mellon University. All Rights Reserved. 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.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution. Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and "No Warranty" statements are included with all reproductions and derivative works. External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu. * These restrictions do not apply to U.S. government entities. Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University. DM19-0067