Systems-theoretic Safety Assessment of Robotic Telesurgical Systems

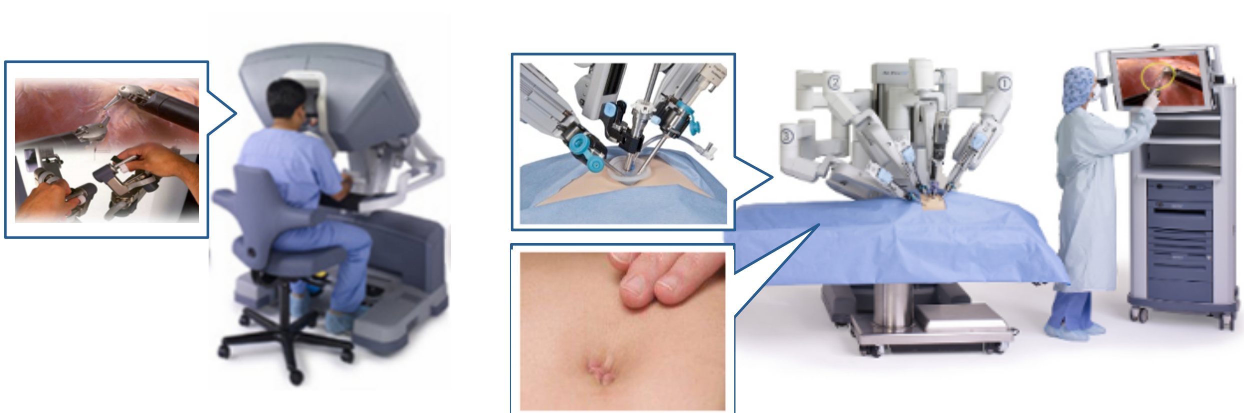
Homa Alemzadeh[1], Daniel Chen[1], Andrew Lewis[2], Zbigniew Kalbarczyk[1], Jaishankar Raman[3], Nancy Leveson[4], and Ravishankar Iyer[1]

[1]University of Illinois at Urbana-Champaign
[2]Applied Dexterity
[3]Rush University Medical Center
[4]Massachusetts Institute of Technology

Robotic Telesurgical Systems

- More than **1.75 million robotic procedures** since 2000
- Various surgical specialties:
  - Gynecology, Urology, General, Cardiothoracic, Head and Neck

Applied Dexterity, Biorobotics Lab, UW, 2007



Zeus Robot, First Intercontinental Surgery, 2003

Robotic Telesurgical Systems

- More than **1.75 million robotic procedures** since 2000
- Various surgical specialties:
  - Gynecology, Urology, General, Cardiothoracic, Head and Neck

- Over **10,600 adverse events** reported to the FDA
  - 9,382 (88.3%) involved device and instrument malfunctions
  - 536 system errors detected during procedures, leading to:
    - Manual system restarts (43%)
    - Conversion to non-robotic methods (61.5%)
    - Rescheduling (24.8%)

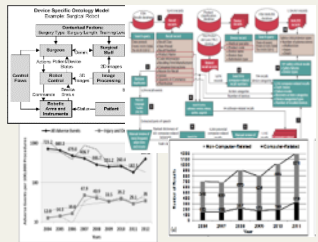
Better evaluation of safety mechanisms are needed.

Alemzadeh, H., et al., "Adverse Events in Robotic Surgery: A Retrospective Study of 14 Years of FDA Data," Technical Report (2015), *arXiv:1504.07135v2.*

Our Research

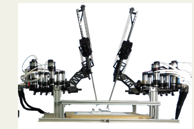
Analyzing Past Failures and Safety Incidents

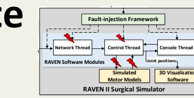
- Tools for automated analysis of incident reports
- Systems-theoretic accident modeling and analysis



Assessing Resilience to Safety Hazards

- Hazard analysis to identify unsafe scenarios and causal factors
- Software fault-injection to emulate realistic safety hazards

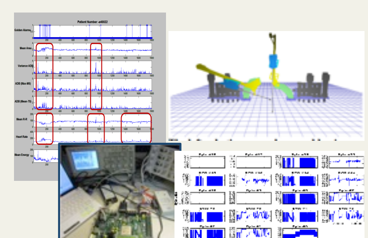




Safe and Secure Robotic Surgical Systems

Designing Safe and Secure Surgical Systems

- Tools for experimental safety and security assessment
- Safety monitors for early detection/mitigation of safety hazards and security exploits



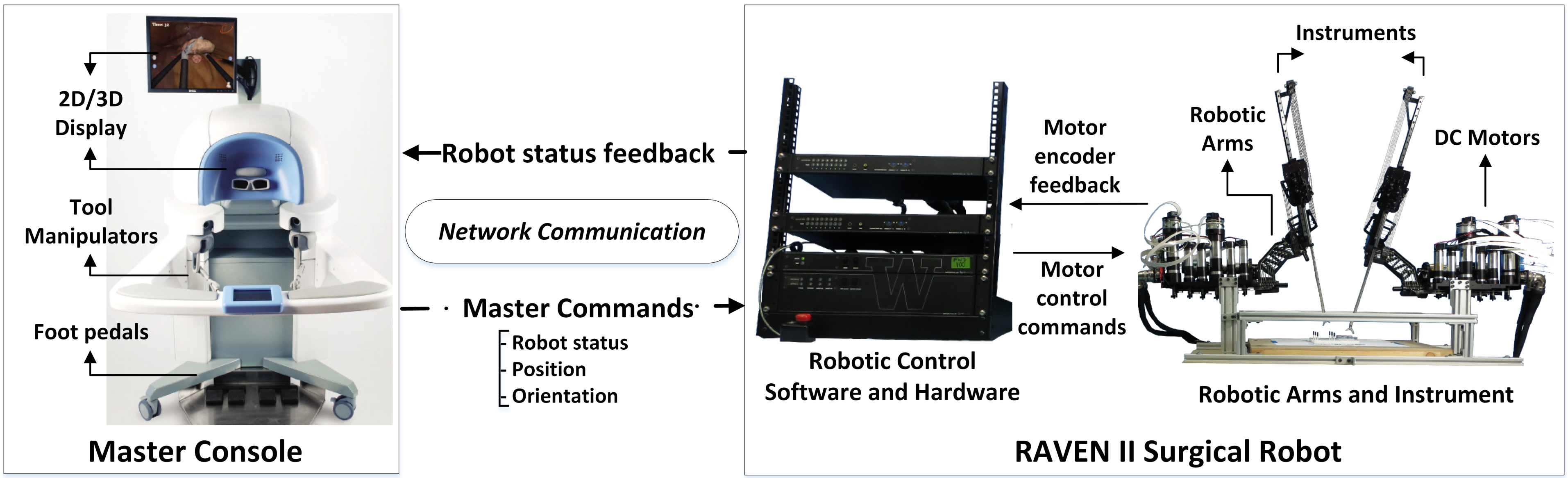
In this Paper…

A systems-theoretic approach to perform targeted fault injection to assess safety mechanisms of a surgical robot

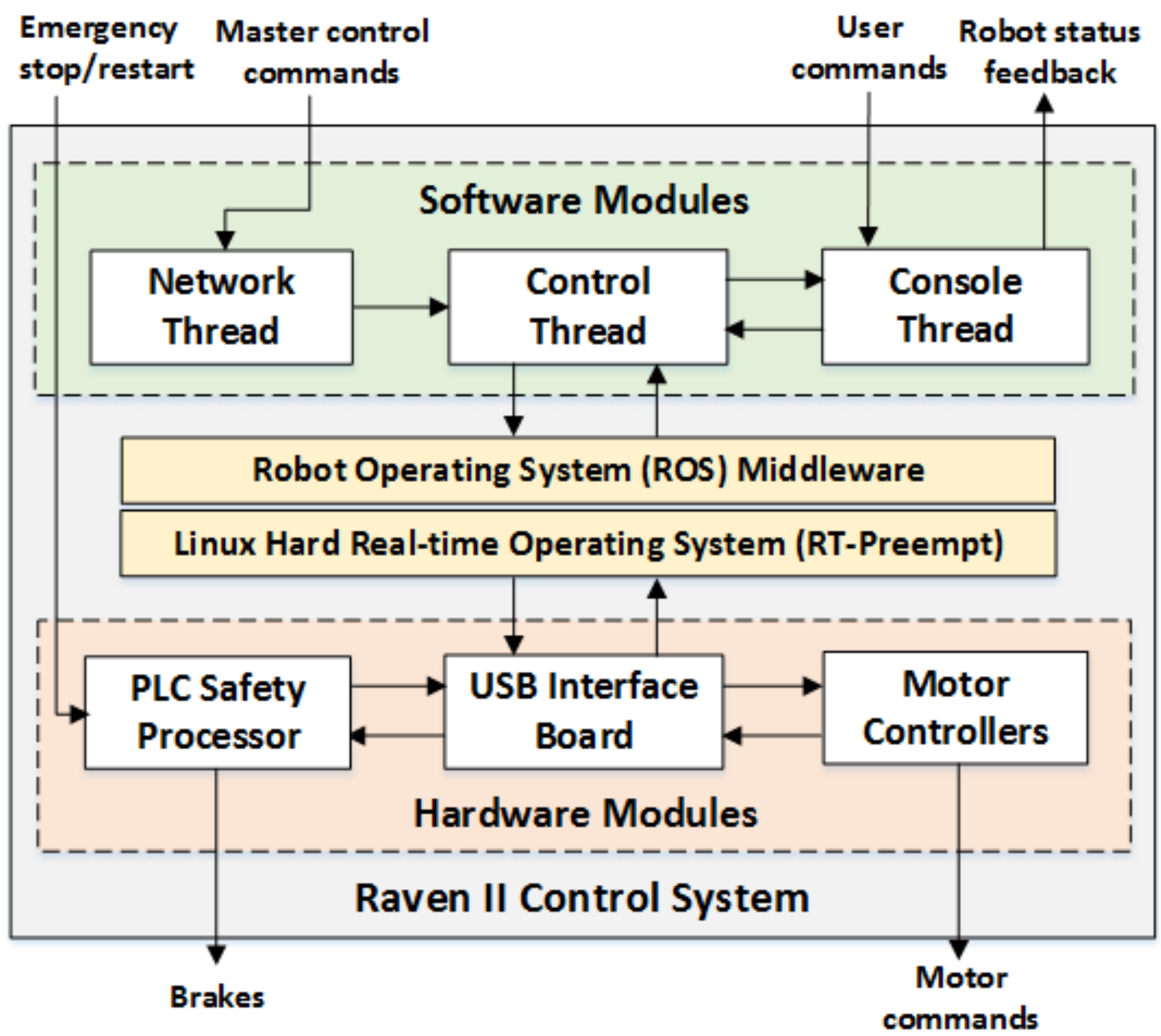
- Case study on **RAVEN II Surgical Robot**

- **Identify potential causes for unsafe control actions** (*safety scenarios*) using STPA
  - Including SW/HW interactions and human operator actions
- Targeted fault-injection to **emulate the identified safety scenarios** by inserting faults in the robot control software
- Quantifying the efficacy of safety mechanisms by identifying
  - **Undetected** safety scenarios
  - **Mitigated** safety scenarios

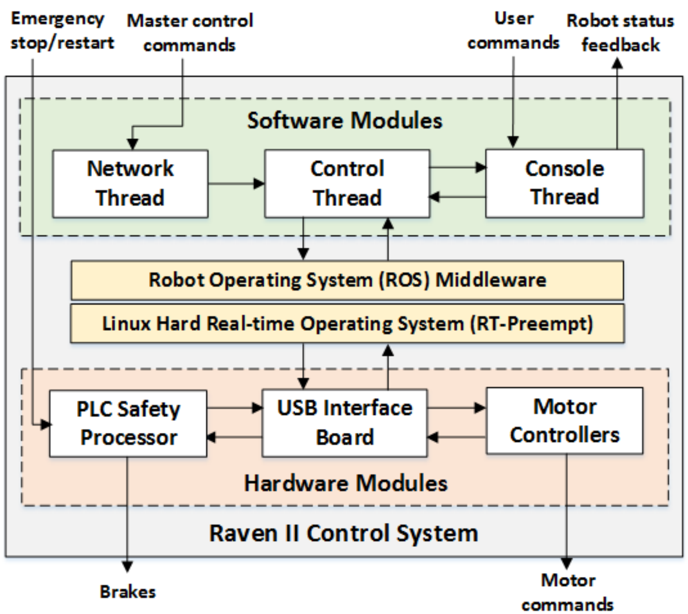
RAVEN II Telesurgical Robot

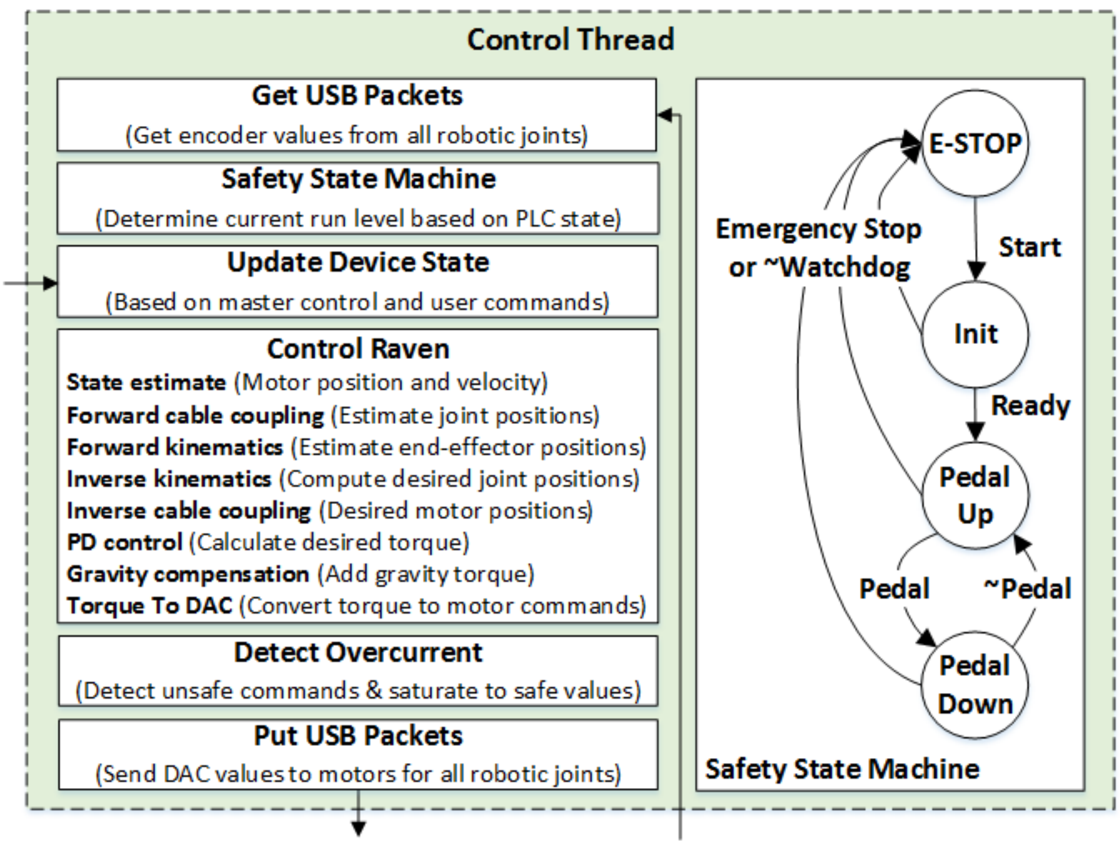


RAVEN II Control System



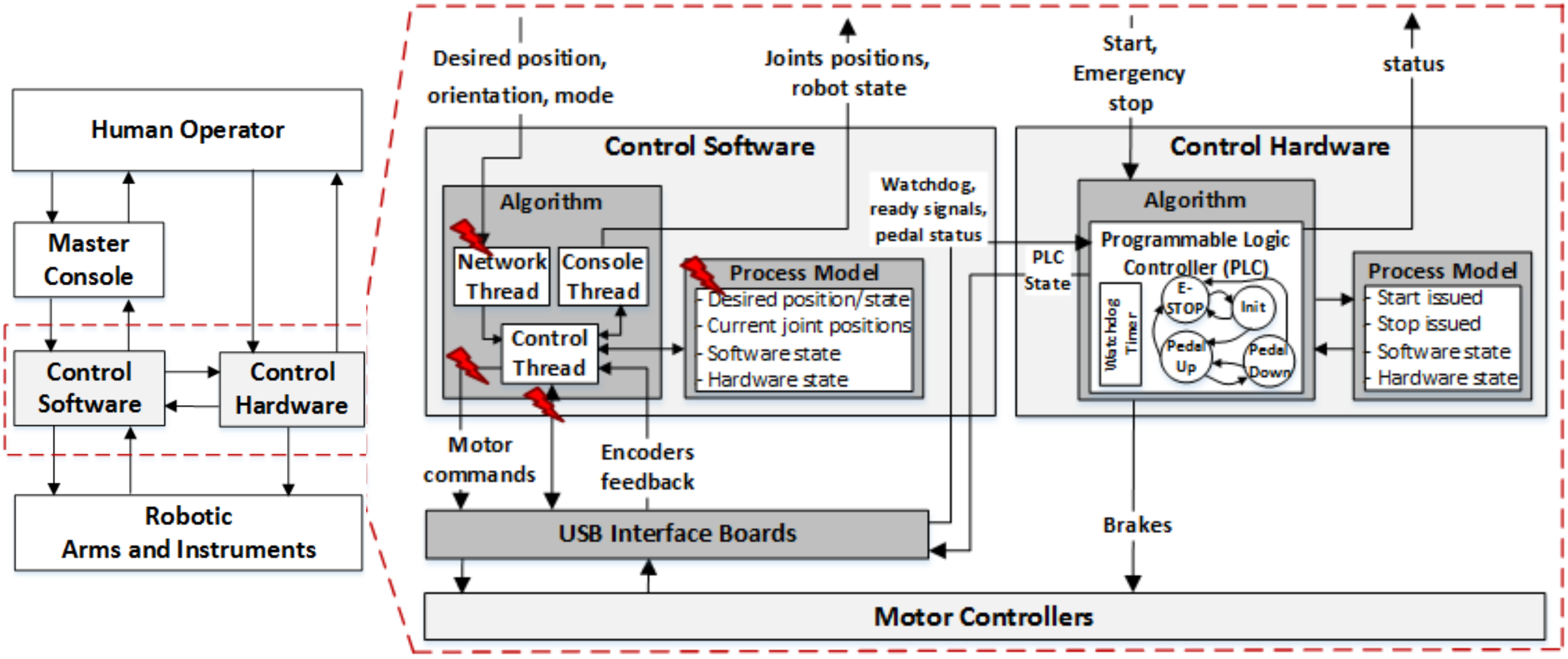
RAVEN II Control System





Safety Control Structure

Hardware and Software Control Loops



STPA Hazard Analysis
Accidents and Safety Hazards

Accidents:

A-1. Patient expires during or after the procedure.

A-2. Patient is injured or experiences complications during/after the procedure.

A-3. Surgical system or instruments are damaged or lost.

Hazards:

H-1. Robot arms/instruments move:

- to unintended location (H1-1),
- with unintended velocity (H1-2),
- at unintended time (H1-3).

H-2. Robotic arms or instruments are subjected to collision/unintended stress.

H-3. Robotic system becomes unavailable or unresponsive during procedure.

STPA Hazard Analysis
Unsafe Scenarios

Unsafe scenarios: the set of system conditions under which the control actions could possibly be unsafe and lead to hazards.

i) a required control action was *not performed*

ii) a control action was performed *in a wrong state*

iii) a control action was performed *at an incorrect time*,

iv) a control action was performed *for an incorrect duration*,

v) a control action was provided, but *not followed by the controlled process*

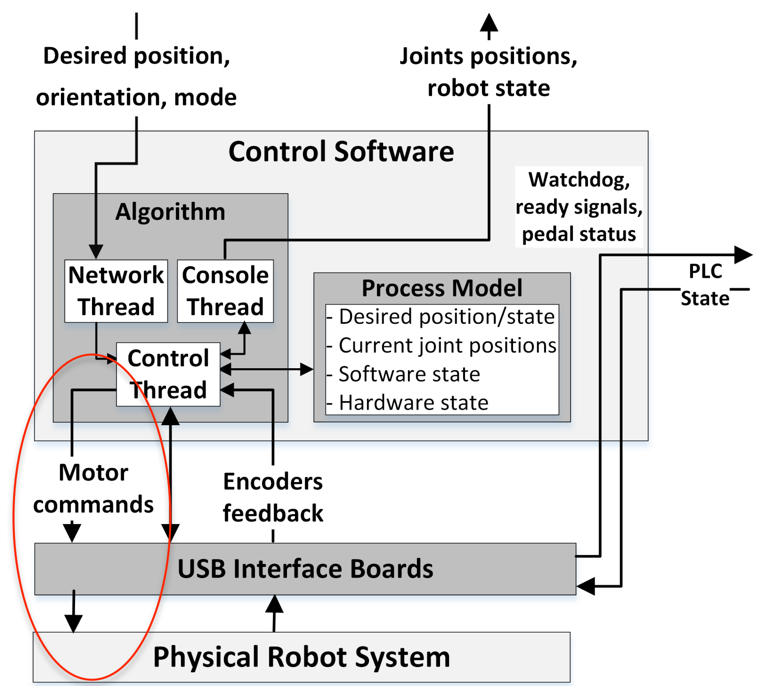
STPA Hazard Analysis
Example Unsafe Scenario

ii) a control action was performed *in a wrong state*

A motor command is ***provided*** by control software when the ***user desired joint position is at a large distance from the current joint position***

Potential hazard: **H1-2**

Robot arms/instruments will move with an unintended velocity



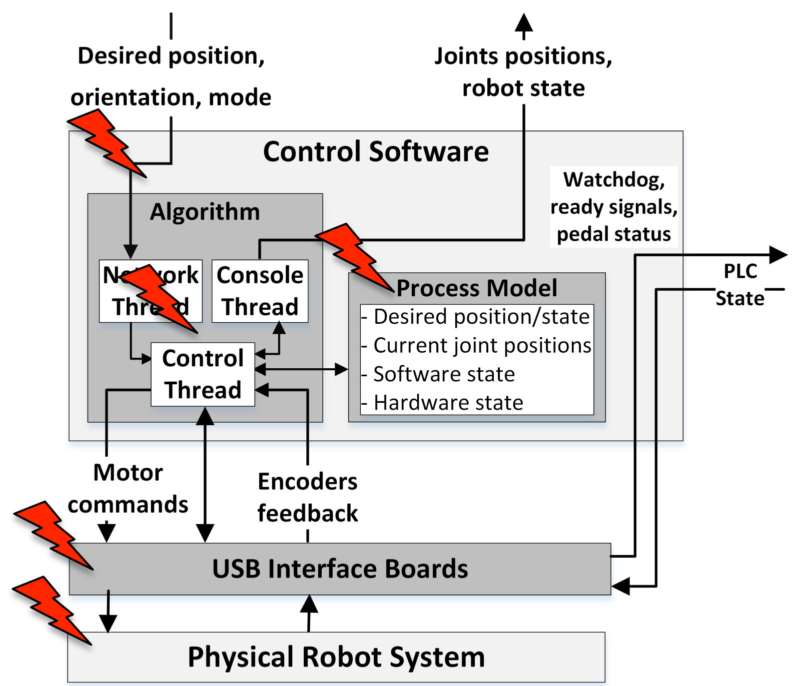
STPA Hazard Analysis
Example Unsafe Scenario

ii) a control action was performed *in a wrong state*

A motor command is ***provided*** by control software when the ***user desired joint position is at a large distance from the current joint position***

Potential causes:

- Incorrect console inputs
- Faulty control algorithm
- Incorrect process model
- Faulty USB communication
- Physical system malfunction



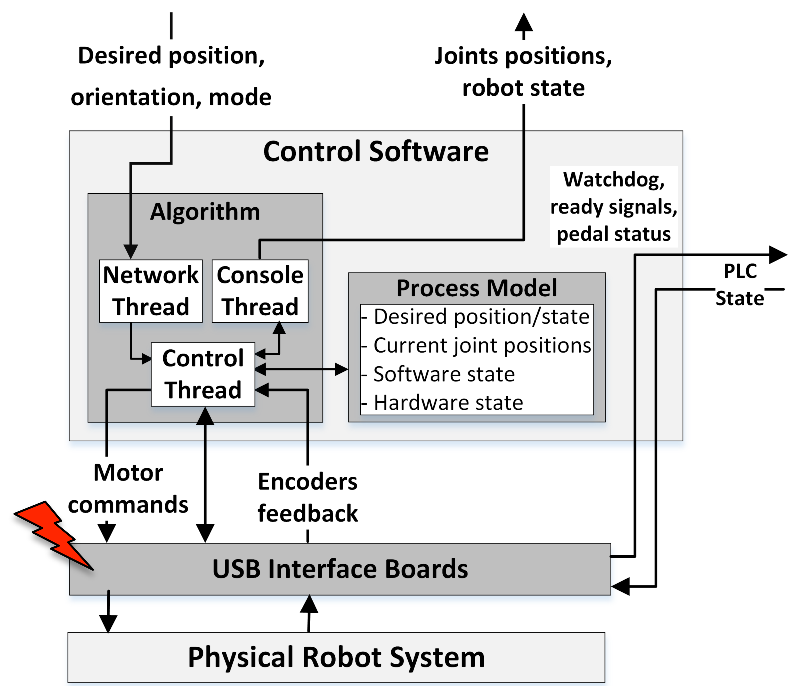
Software Fault Injection Strategies

Injection targets in the robot control software:

- Target functions and variables
- Injection triggers

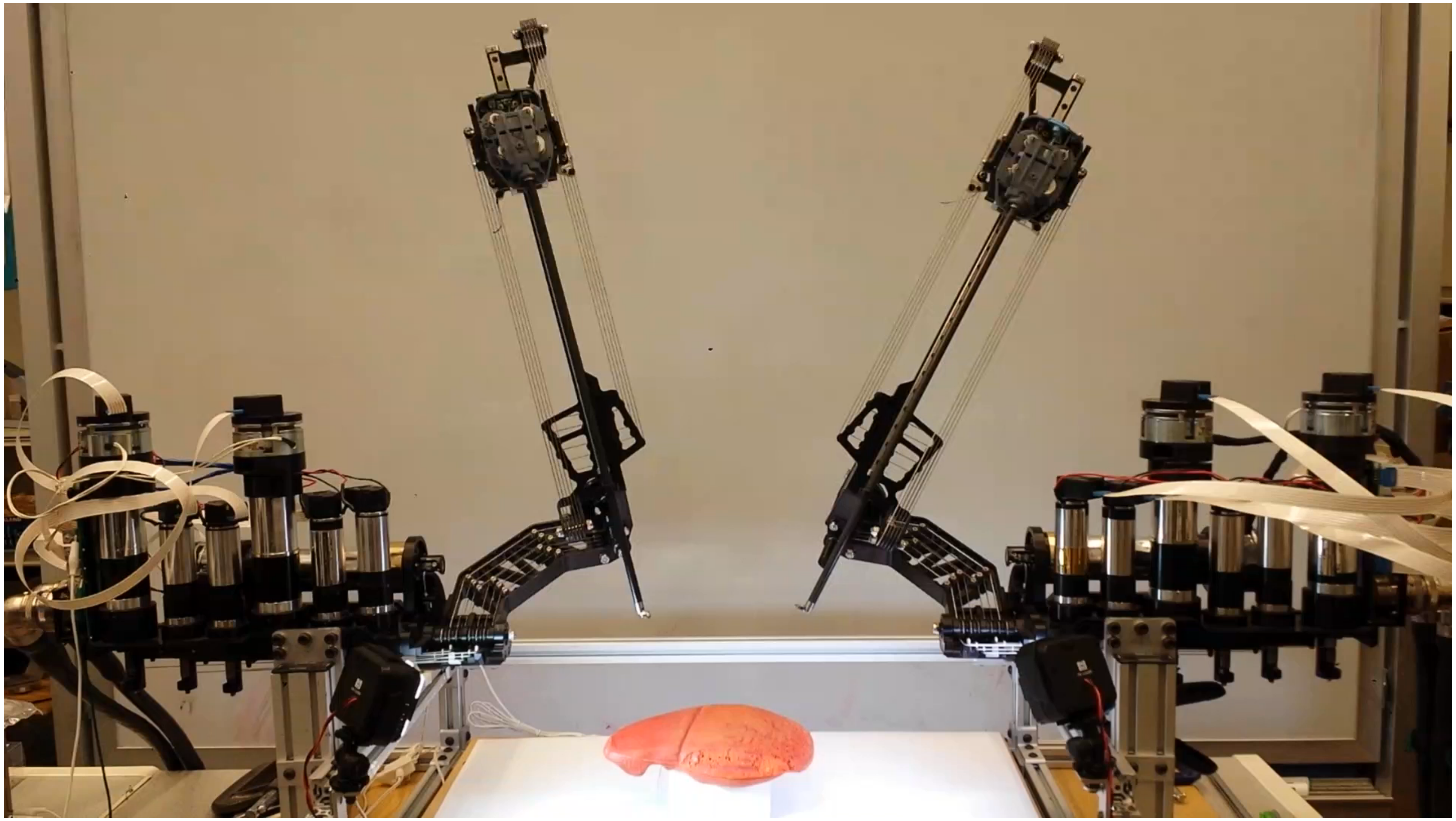
Example: Faulty USB communication

- Function: *putUSBPacket*
- Variables: *Joints current commands*
  [Stuck At Random Value]
- Triggers: *robot_state = Homing*
  robot_state = Pedal Up



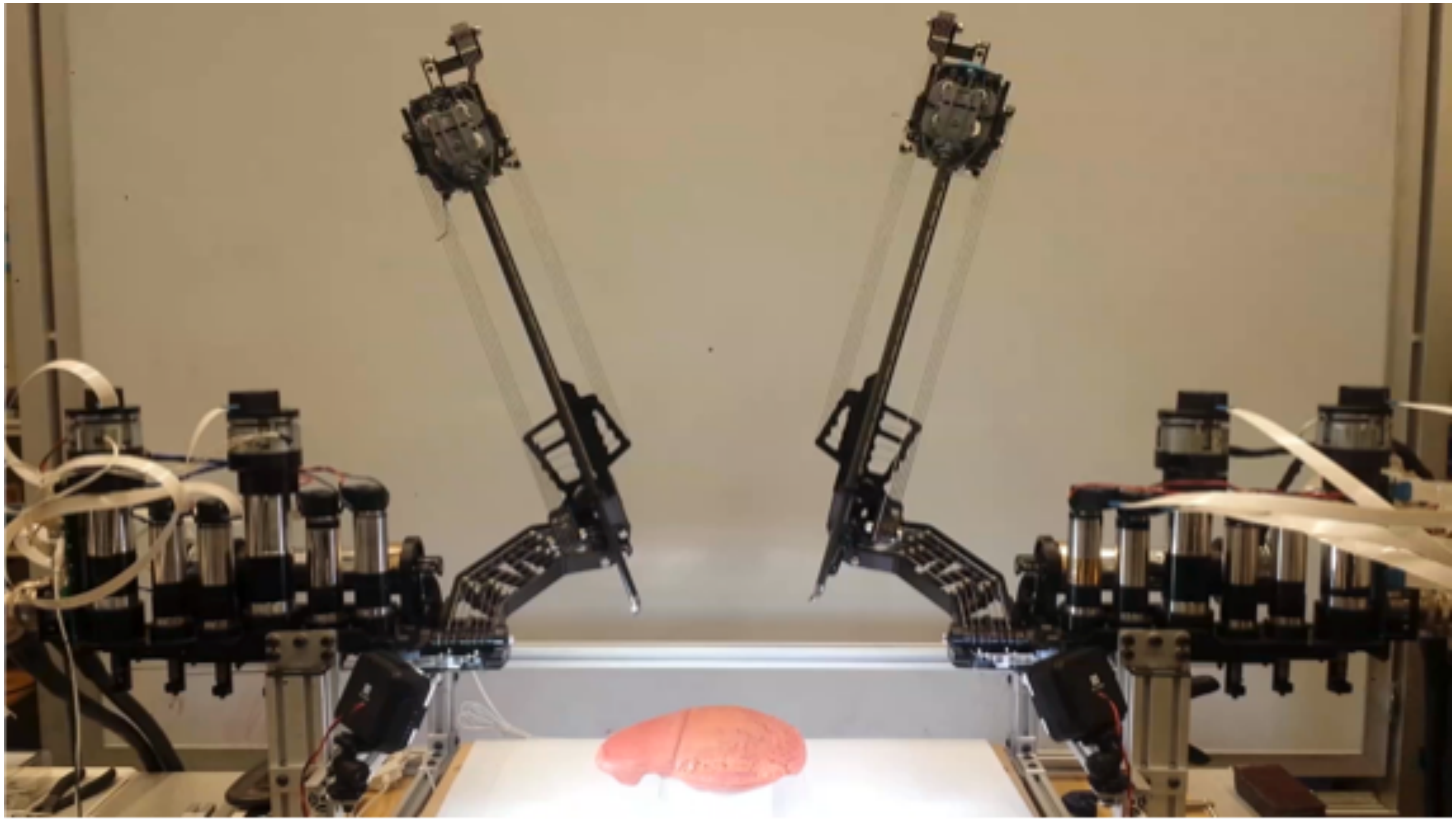
Abrupt Jump (H1)

Faulty USB packets sent to the I/O Boards



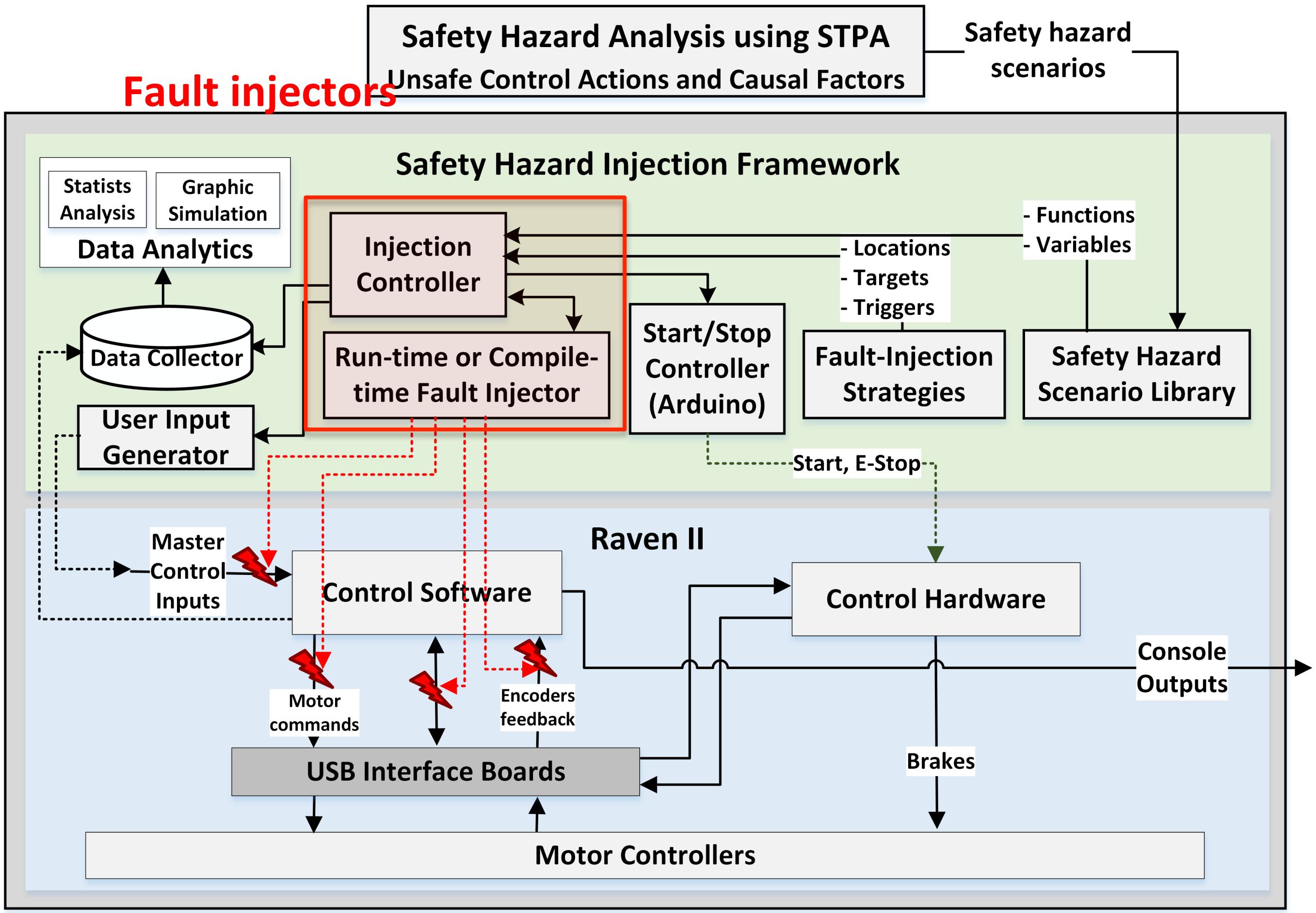
Link to the video: https://www.dropbox.com/s/rrx6f74xful38on/Sudden_Jump.mp4?dl=0

Homing Failure (H3)
Faulty USB packets received from the I/O boards

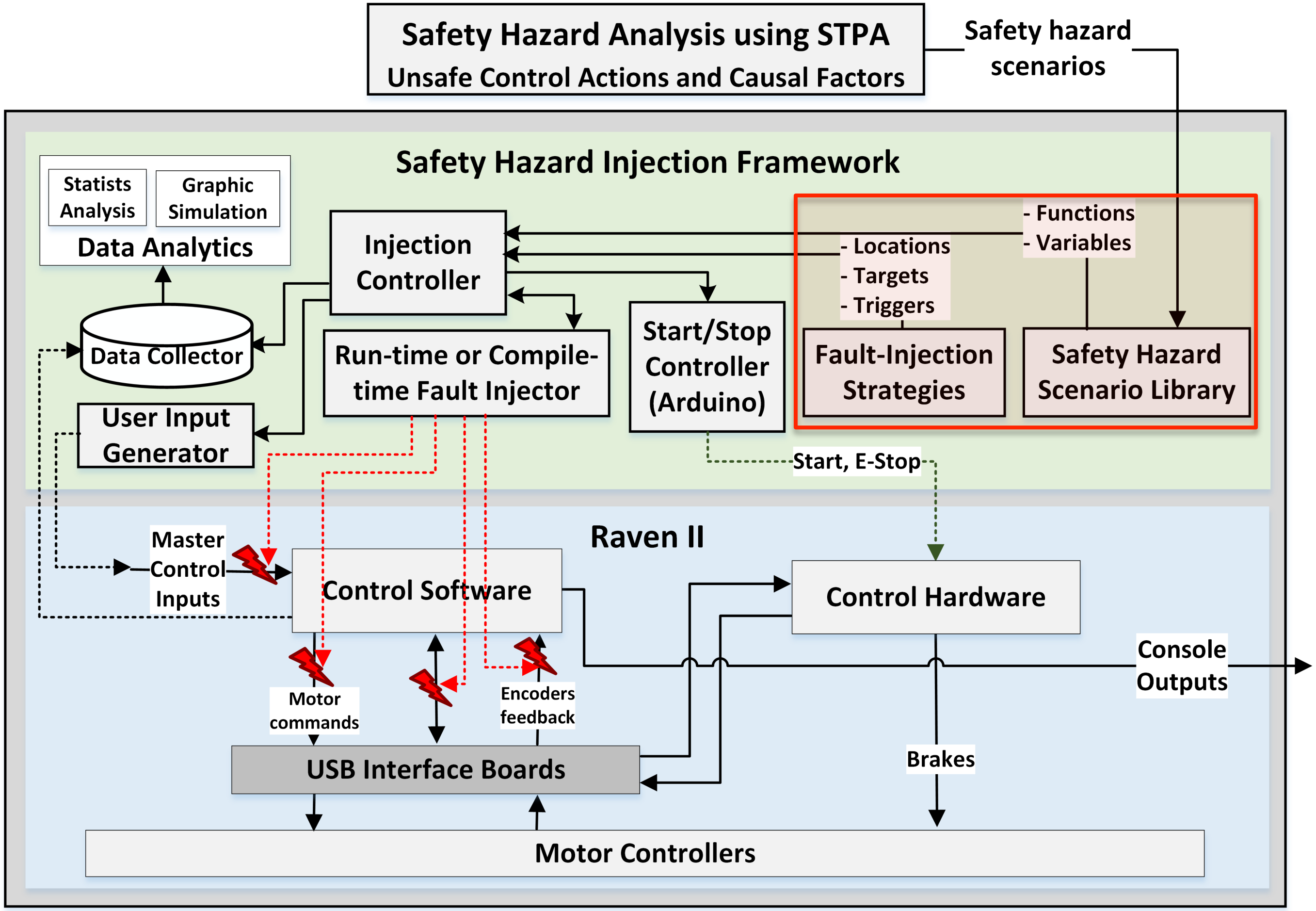


Link to the video: https://www.dropbox.com/s/0wa9evgwfj9nr6k/Repeated_Homing.mp4?dl=0

Safety Hazard Injection Framework

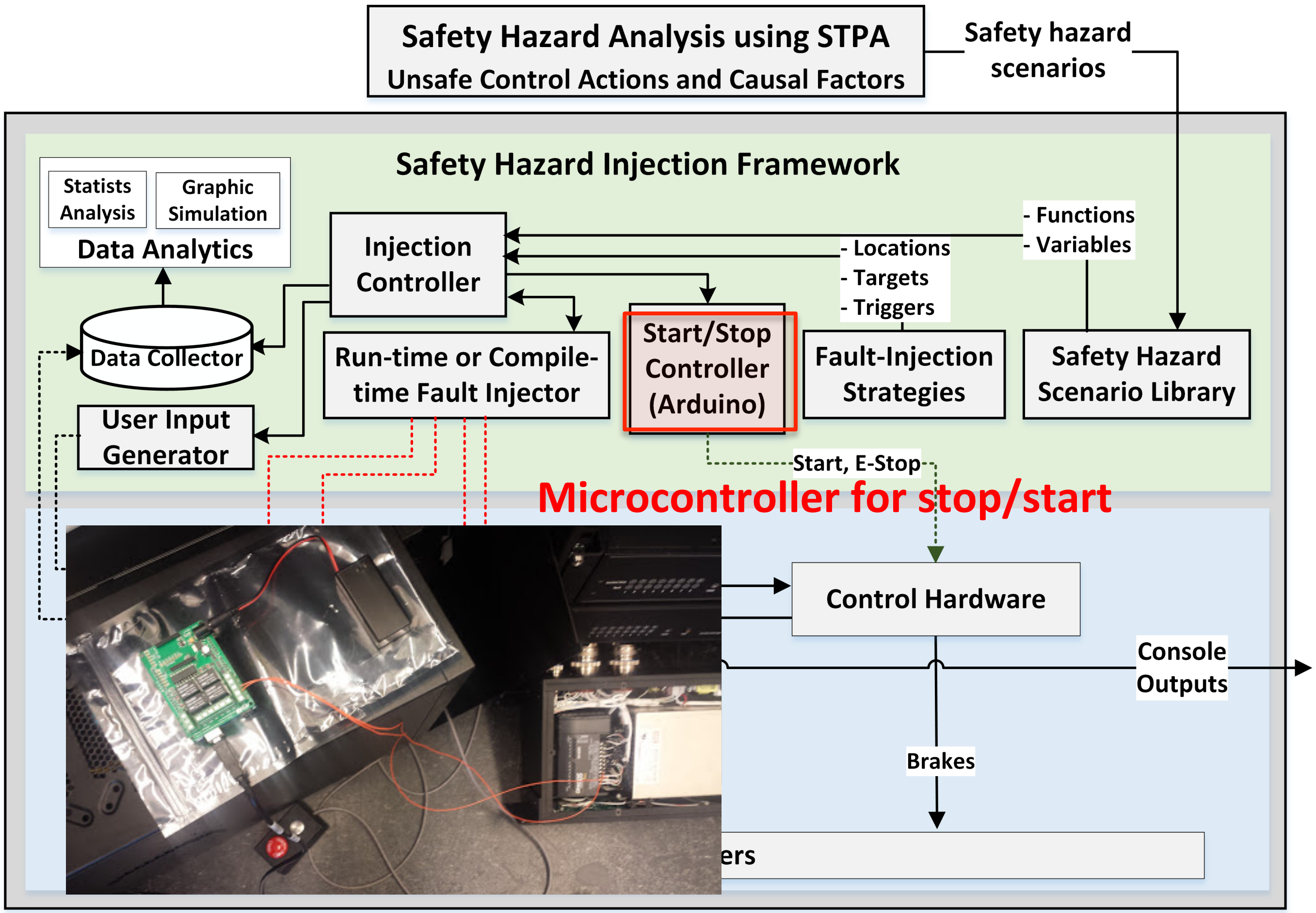


Safety Hazard Injection Framework



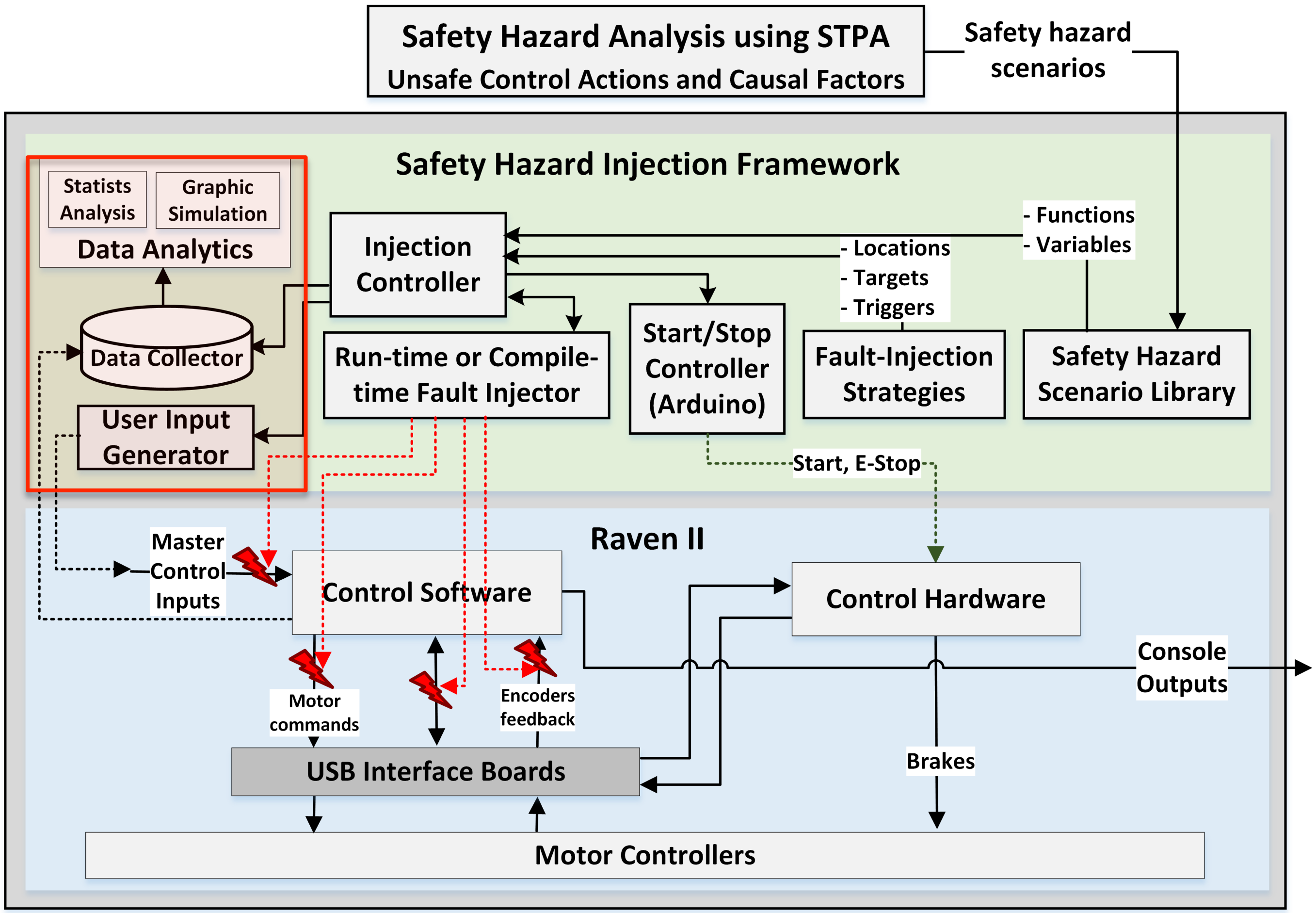
Injection targets

Safety Hazard Injection Framework



Safety Hazard Injection Framework

Trajectory generator and results analysis



Experiments

- Emulated 45 safety scenarios by injecting faults into the RAVEN control software

Injected Software Fault *Target Function*: Variables [Fault Type, Values]	No.	Observed System Behavior	Hazard
network_process: Position and Orientations [Stuck At Out of Range]	20	During Homing: No impact After Homing in Pedal Down: IK-failure, small jumps, no movements with no E-STOP, E-STOP	H1 H3
network_process: Foot Pedal Status [Stuck At 0, StuckAt 1]	20	During Homing: No impact After Homing: Does not start movement if Stuck At 0, No impact if Stuck at 1.	

- 25 locations within 13 software functions
- A total of 368 targeted fault-injections
- Each scenario repeated > 10 times to achieve high confidence in the observed behavior

Results:
Undetected Safety Hazards (1)

Unintended Robotic Movement (H1) – Abrupt Jumps

Unintended Collision or Mechanical Stress (H2) – Cable breakage

Example scenarios:

- Inputs:
  - ***Intermittent out of range*** values injected into the position, orientation, and foot pedal variables
- Control algorithm:
  - Random torque values injected to the joints current commands
  - Stuck-at faults injected to the estimated motor ***velocities***
- USB interface:
  - Faulty packets ***sent to*** the motor controllers

Results:
Undetected Safety Hazards (2)

Unresponsive Robotic System (H3) –
Stuck at emergency stop or software error

Example scenarios:

- Inputs:
  - ***Stuck-at faults*** injected into the position, orientation, and foot pedal variables
- Control algorithm:
  - Stuck-at or intermittent faults injected to the estimated motor ***positions***
- USB interface:
  - Stuck-at or intermittent faults injected to the packets ***received from*** the motor controllers (never finishes homing)

Real Incidents in Robotic Surgery

Examples from FDA MAUDE database

Report # (Year)	Summary Event Description from the Report	Potential Causal Factors (ID in Table 3)	Observed Behavior (Hazard)	Patient Impact
2120175 (2011)	During a hysterectomy procedure, the left master controller did not have full control of the maryland bipolar forceps instrument, resulting in non-intuitive motion and causing a small bleed on the patient's uterine tube.	Master console calibration issue (i)	Non-intuitive movement (H2)	Small bleed on patient's uterine tube
2663924 (2012) 2589307 (2012)	Approximately 3.5 hours into a pancreatectomy procedure, multiple instances of non-recoverable system error code #23 was experienced and the surgeon was unable to control the patient side manipulator (psm) arms.	Communication failure between master console and robot (i)	Non-recoverable system error (H3)	Converted to open surgery after 3.5 hours

Lessons Learned

Vulnerabilities in the Safety Mechanisms:

a) Lack of monitoring mechanisms for the initialization (homing) process.

b) No safety mechanisms for monitoring the USB board communications.

c) No hardware detection mechanisms for unsafe motor commands.

d) No feedback from the motor controllers and brakes to the PLC

Robust Safety Mechanisms:

a) Robot movements cannot start without the start signal from the operator

b) PLC engages the brakes upon loss of watchdog ("E-STOP") or foot pedal signals from software

c) Software only sends the pedal signal to the PLC when the foot pedal is pressed and it is not in "E-STOP" or "Init" states.

d) Software checks the status of PLC on every cycle to immediately follow the state transitions of the robotic hardware.