UMassAmherst

Monitoring the state of the physical plant in a CPS to detect and counter benign faults and malicious attacks

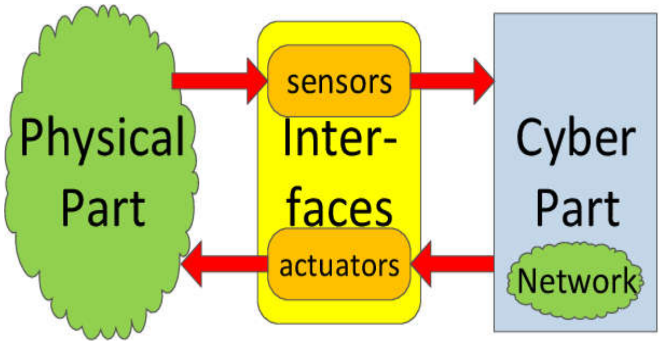
Israel Koren

University of Massachusetts at Amherst

Outline

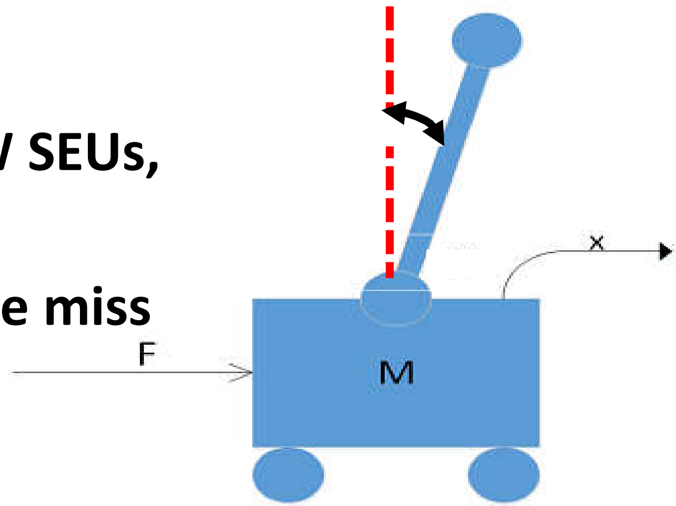
- **Objective – detect and counter faults in cyber physical systems**
- **Fault types**
  - **Benign faults – increase reliability**
  - **Malicious faults – increase security**
- **Known techniques to achieve these objectives**
  - **High overheads**
- **Our approach: Monitor the state of the physical plant**
  - **Fit the level of protection to the current state sub-space**
- **Main challenge: Determine in real-time the state subspace**
  - **Use Machine learning techniques**

Critical CPS applications



- **Many CPSs control life-critical applications**
  - **E.g., Aircrafts, Nuclear reactors, Smart Buildings, Automobiles, Medical Devices**
  - **Must support high levels of safety and provide timely response to benign and malicious faults**
- **Common techniques to detect and recover impose high overheads**
  - **Hardware, performance, power**
  - **Most focus on the cyber sub-system ignoring the physical plant**
- **Our approach: Detect faults and invoke adaptively proper countermeasures**

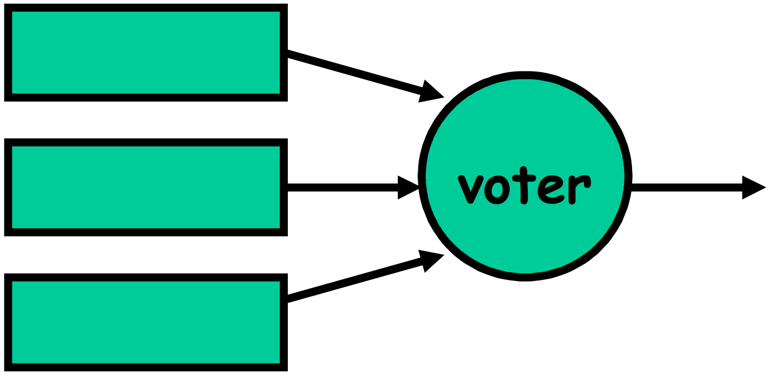
Failures in Cyber-Physical Systems

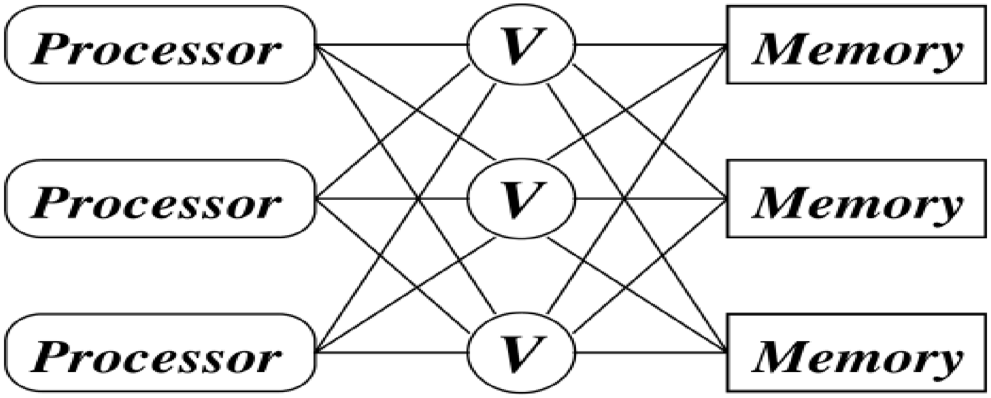


- **Computing side:**
  - **Erroneous computer outputs due to HW SEUs, SW bugs or maliciously modified SW**
  - **Computational delays causing a deadline miss**
- **Physical side:**
  - **Application specific**
    - **E.g., failure in an inverted pendulum: angle ≥ 90°**
  - **Safety Space Constraints (SSC): The conditions that the controlled plant must satisfy in order to operate safely**
    - **E.g., inverted pendulum: angle should be ≤ 0.5 rad, or 30°, otherwise it is unsafe**

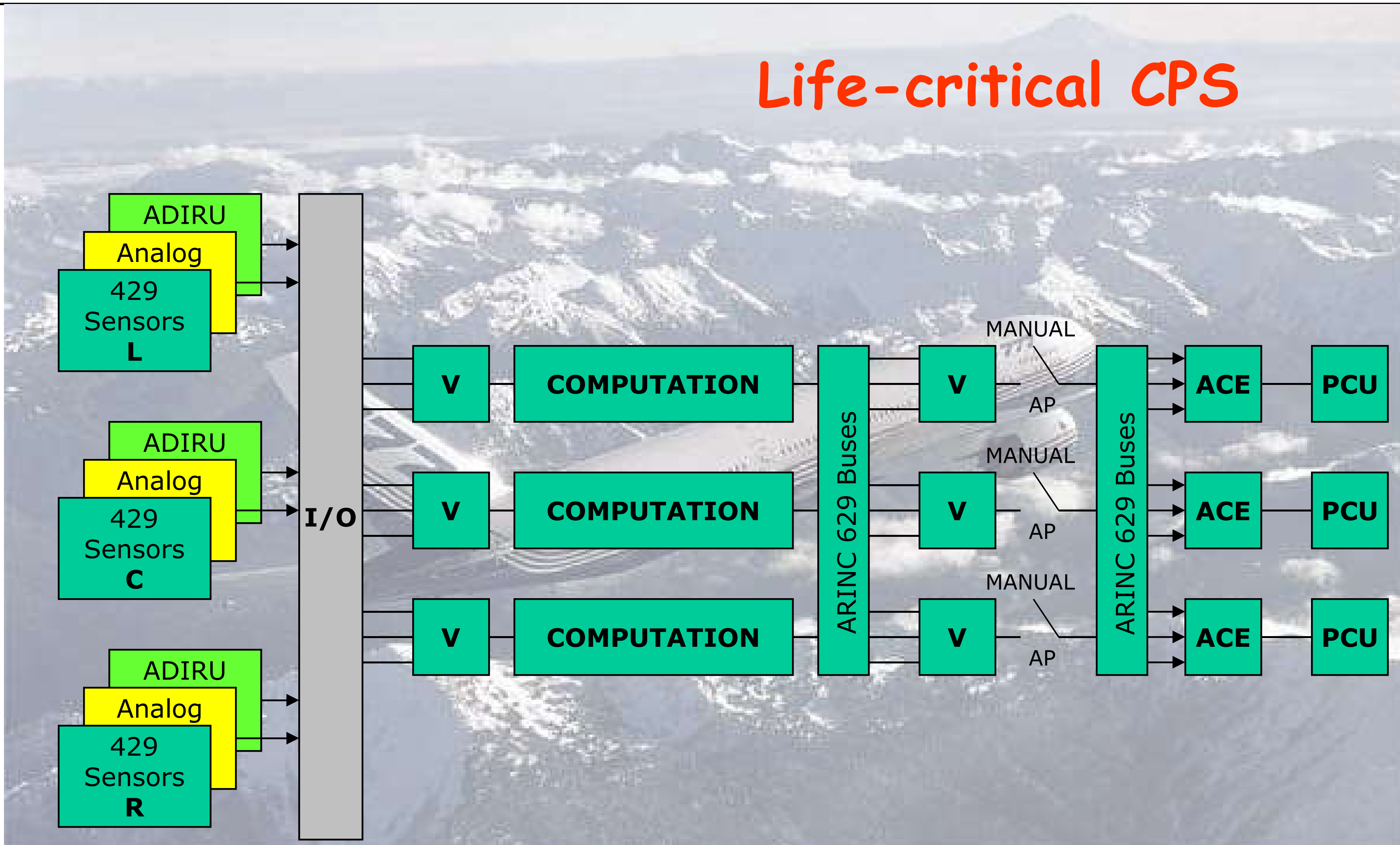
Fault Tolerance (FT) in CPS

- **Traditional FT - continuous massive redundancy**
  - **Duplex: two copies of a task running on two cores, can detect faults**
  - **TMR: three copies of a task running on three cores, can mask a single erroneous result**

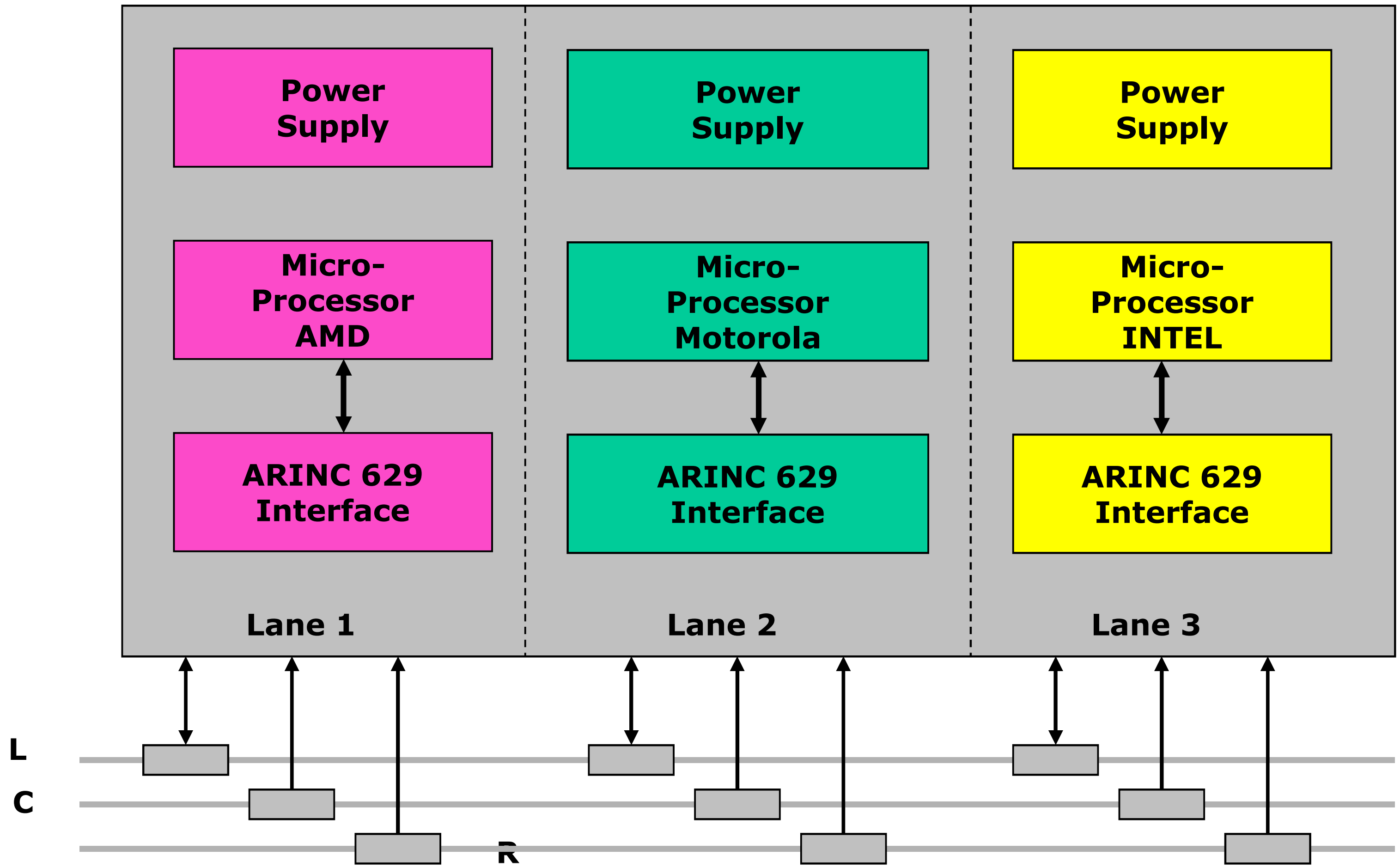




Example: Boeing 777 (early design)



TMR with design diversity



UMassAmherst

Our approach – Adaptive Fault Tolerance

- **Plant state based adaptive FT:**
  - **If the plant is deep within its safe region, can withstand some erroneous control inputs**
  - **In such a state, a lower level of FT can be deployed**
  - **Need a definition of**
    - **Safe region**
    - **How to determine whether the plant is "deep" in the safe region**

Physical Plant's Safe State Space (S^3)

- **Definition:** The sub space of the states of the physical system that meet the SSC (determined by the application engineer)
- **A point is in S^3 if:**

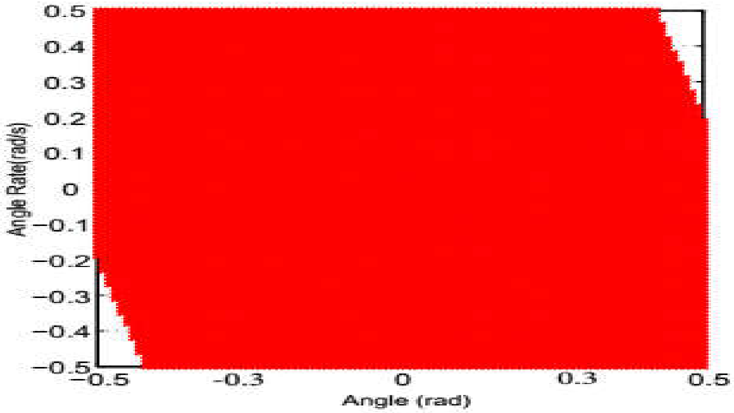
SSC: Safety Space Constraints

  - 1. The plant satisfies the SSCs at the present time, and
  - 2. Based on
    - (1) the controlled plant control laws,
    - (2) the control algorithm used,
    - (3) the actuator limitations,
    - (4) the control task execution rate, and
    - (5) the limits of the operating environment impact

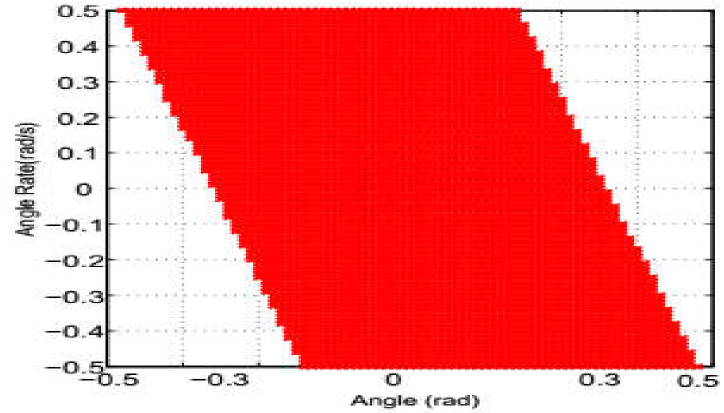
    the plant will continue to satisfy these constraints up to a given horizon, as long as correct control inputs are applied

Example: S^3 for inverted pendulum, horizon 15sec

Control task period

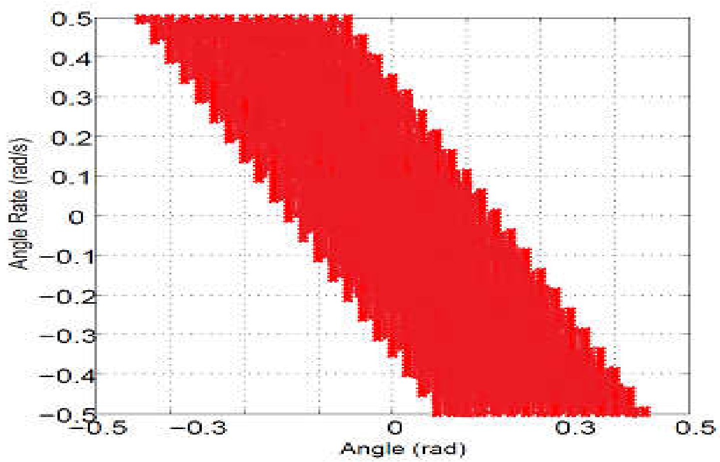


(a) Period = 10ms

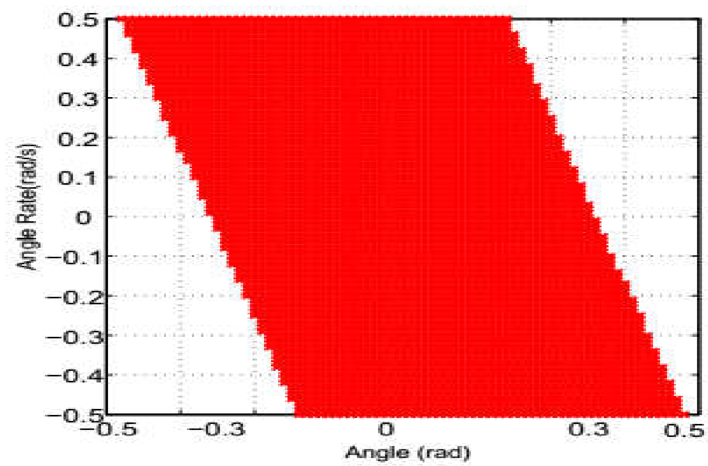


(b) Period = 30ms

Actuator bound



(c) $u_{lim} = 2.5$ N



(f) $u_{lim} > 10$ N

Sub-spaces of S^3

- **S1: Even if the controller generates the worst-case control input until the next iteration of the control task, the plant will not leave its S^3**
- **S2: If the controller generates a default output (e.g., zero or repeat the previous output), the plant remains in S^3**
- **S3: If the controller produces an incorrect output, the plant is not guaranteed to stay in S^3**
- **(Benign) Fault Tolerance implications:**
  - **S1: No fault-tolerance is required**
  - **S2: It is sufficient for the computer to be fail-stop**
  - **S3: Faull fault-masking is necessary**

Security in CPS

- **Unique characteristics of CPS**
  - Limited computing resources
  - Often limited power
  - Often inaccessible location
  - Network connectivity
  - Physical exposure
- **Vulnerabilities**
  - Network intrusion
  - Exhaustion attack
  - Information theft
  - Modifying software (code injection or reprogramming)
  - Physical tampering (side channel attacks)
  - Modifying sensor output

Classifying Security Threats in CPS

- **Distinguish between two malicious objectives**
  - **Harming physical plant operation vs.**
  - **Stealing propriety information**
    - **Stealing information - well-known threat in general computing systems**
    - **Various cryptographic schemes can be employed**
- **Threats to the physical plant operation can be detected by**
  - **Common techniques to detect intrusion & software modification**
    - **E.g., code analyzers, anomaly detection, sandboxing**
    - **Often have a high overhead for constrained CPSs**
    - **Never achieve 100% coverage as new attacks are developed (hard to update countermeasures)**

Our approach to deal with security threats in CPS

- Must first detect the threat and if possible recover

- Monitor the state of the plant and identify marginal states
  - The marginal state is likely to be the result of a fault
  - The exact nature of the fault is unknown
    - (1) A benign fault requiring fault tolerance measures
      - E.g., execute two copies of the control task on two cores
    - (2) A malicious attack on the control task
      - Must use a different version of the control task

Counteracting security threats in CPS

- Assume first that it is a benign fault – duplicate the control task
  - If the state remains marginal – replace the current version of the control task by a second version
  - Second version should follow a simpler control algorithm
    - More robust, shorter execution time but lower quality
    - Can be useful even for dealing with benign SW bugs
  - If the plant state is still marginal execute emergency procedure
    - Use a default control (even an open-loop scheme)
    - Inform remote operator
- Detecting a threat to the safe operation of the physical plant is the most significant step

Challenge: Determine current sub-space in real-time

- Given the current state of the physical plan how to decide which sub-space it belongs to?
  - Storage constraints
  - Timing constraints
- Use machine learning schemes to identify boundaries between sub-spaces
  - Hopefully requiring only a few parameters
- Standard Machine Learning (ML) algorithms for classification problems:
  - E.g., Logistic Regression, Neural Networks, Support Vector Machine (SVM)

Safety Critical Issues

- Can not guarantee 100% classification accuracy
- Need a way to make it conservative
- Misclassification from S1 to S2 or even S3 is allowed, only wasting computing resource; from S3 to S1 is not allowed
- Classification algorithms produce a 1 if the calculated probability is greater than a threshold
  - Default 0.5
- Can iteratively adjust this threshold value, until no dangerous misclassifications exist

Real-Time Task Optimization - example

- **Inputs:**
  - Number of available copies & number of versions for each task
  - Power consumed by each version of every task
  - Current temperature of each processor $T_{proc}(t)$
- **Output:**
  - Preferred version for each task (Note: need to generate classifier for each version of every task)
- System objective: e.g., minimize aging of processors due to high operating temperature
  - All circuit fault mechanisms rates exponential in T (e.g., electro-migration, dielectric breakdown and stress migration)
  - Thermal Age Acceleration Factor (TAAF)

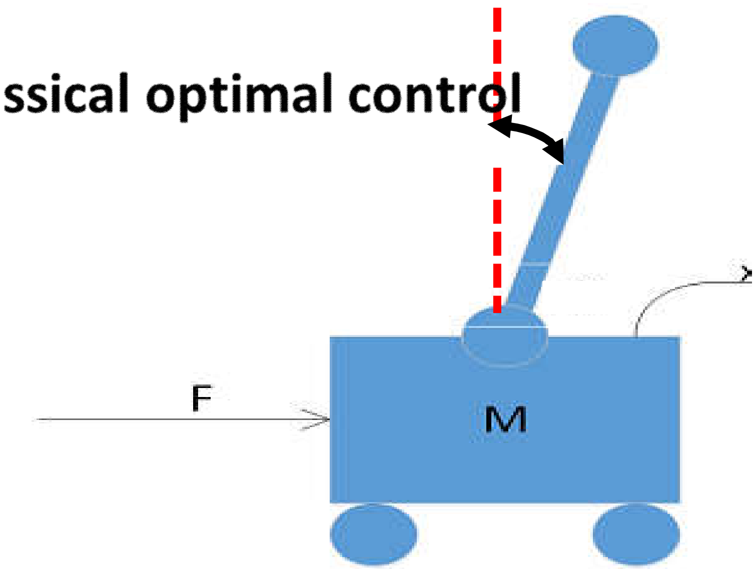
$$TAAF = e^{\left(-\frac{E_a}{kT_{proc}(t)}\right)}$$

Examples of online plant state classification

- **Inverted pendulum**
- **Anti-lock Braking System (ABS) in a car**
- **Highway platoon**
- **Humanoid Robot**

Inverted Pendulum

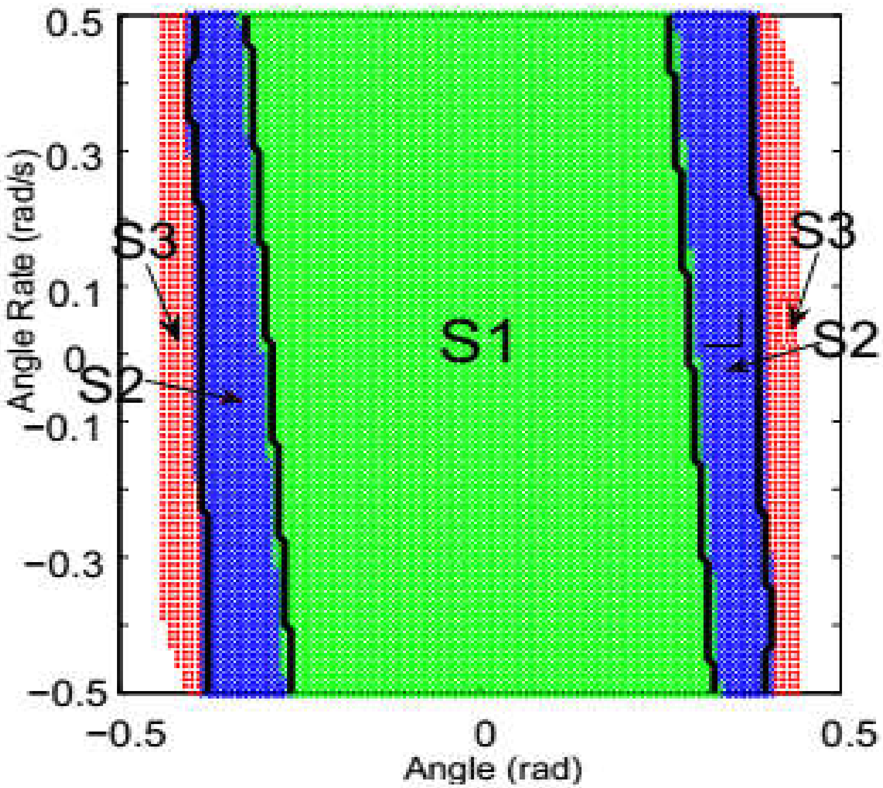
- **Real-time control algorithm:**
  - **Linear Quadratic Regulator (LQR) - classical optimal control algorithm**

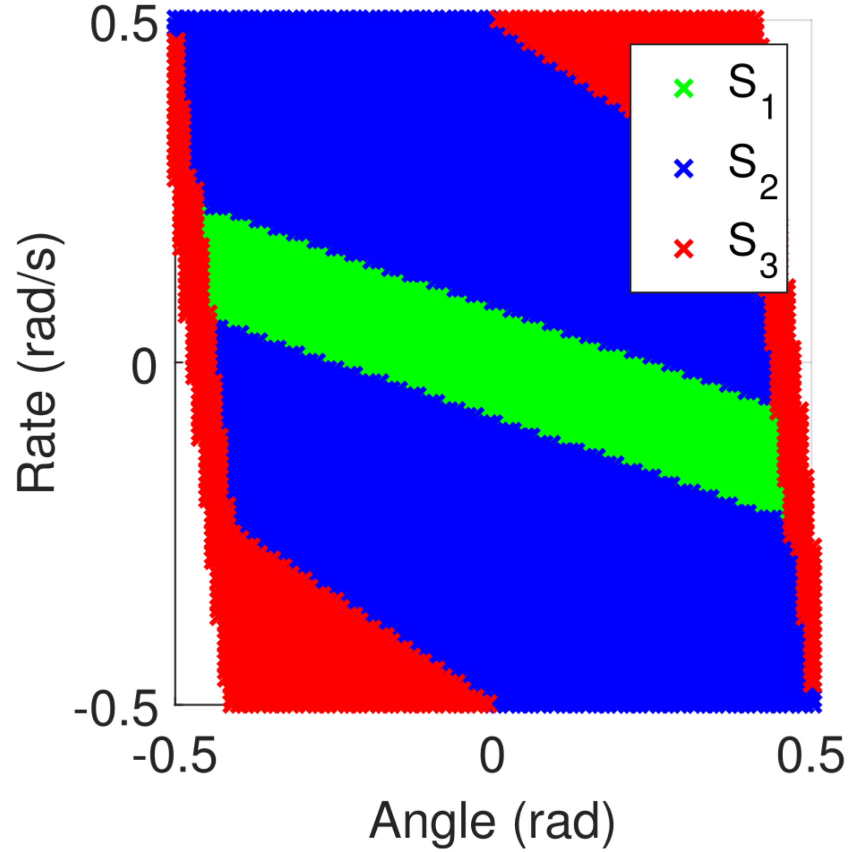


- **Safe State Constraints (SSC):**
  - $-0.5 \le \emptyset \le 0.5\ rad$

- **Upper and Lower Bounds of the control force:** $\pm$**40 N**

Sub Spaces and Decision Boundaries





Max Cart velocity→0

Training Algorithms (Inverted Pendulum)

	LR	*NN*	*SVM*
Trained Parameters Size	15	153	788
Training Accuracy	85.8%	99.92%	99.98%

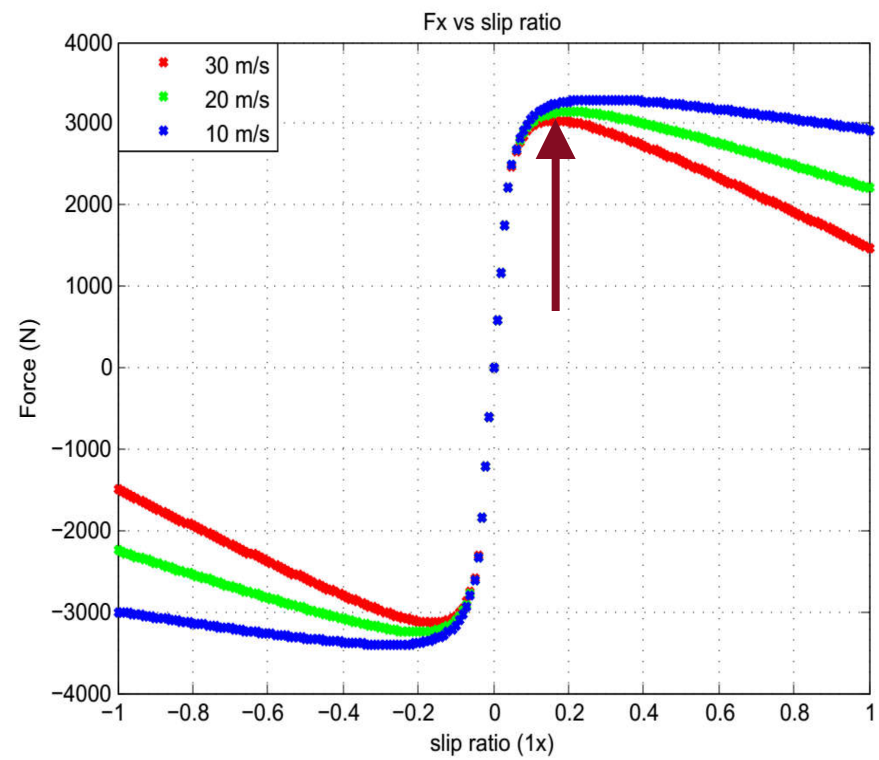
COMPARISON OF LEARNING ALGORITHMS FOR INVERTED PENDULUM

Angle	*Angle Rate*	*Predicted*	*Actual*
-0.3900	-0.1800	3.0000	2.0000
-0.3100	0.1600	3.0[illegible] S3	2.0[illegible] S2
0.3100	-0.1600	3.0[illegible]	2.0[illegible]
0.3900	0.1800	3.0000	2.0000

TRAINING PERFORMANCE OF NEURAL NETWORKS FOR INVERTED PENDULUM

Anti-Lock Braking System (ABS)

- **Prevent wheels from locking up during hard braking**
- **Also maximize braking forces generated by the tires to get small stop distance**
- **The most important parameter is the Slip ratio**

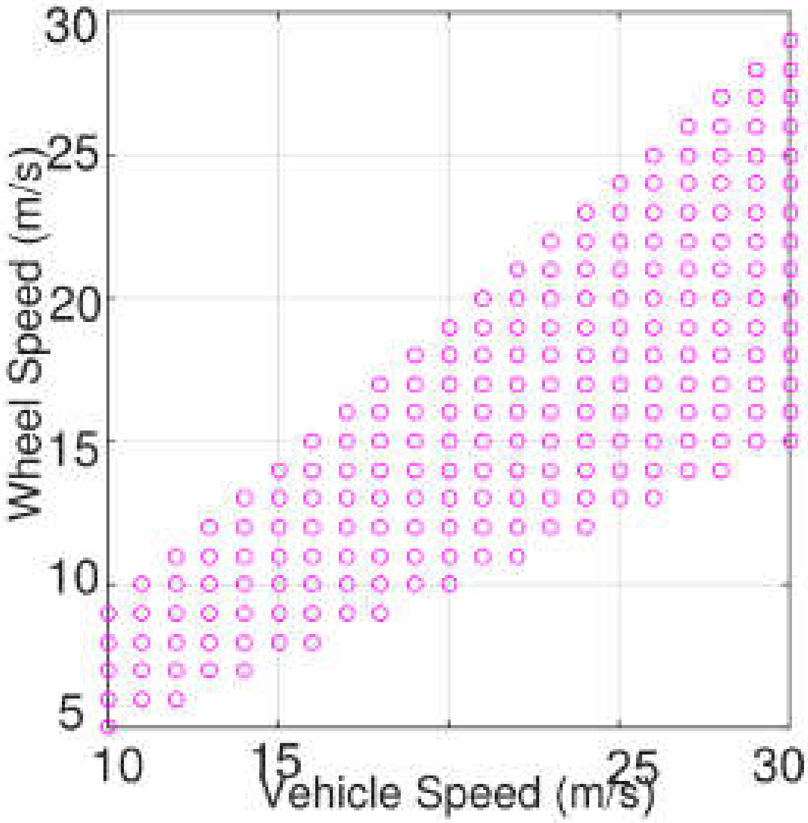


- **Slip_ratio**: $\sigma_x = \frac{r_{eff}\omega_w - \dot{x}}{\dot{x}}$
- ω_w **is the wheel speed,** $\dot{x}$ **is the car speed**
- **Largest longitudinal friction force is achieved for a slip value around 0.15**

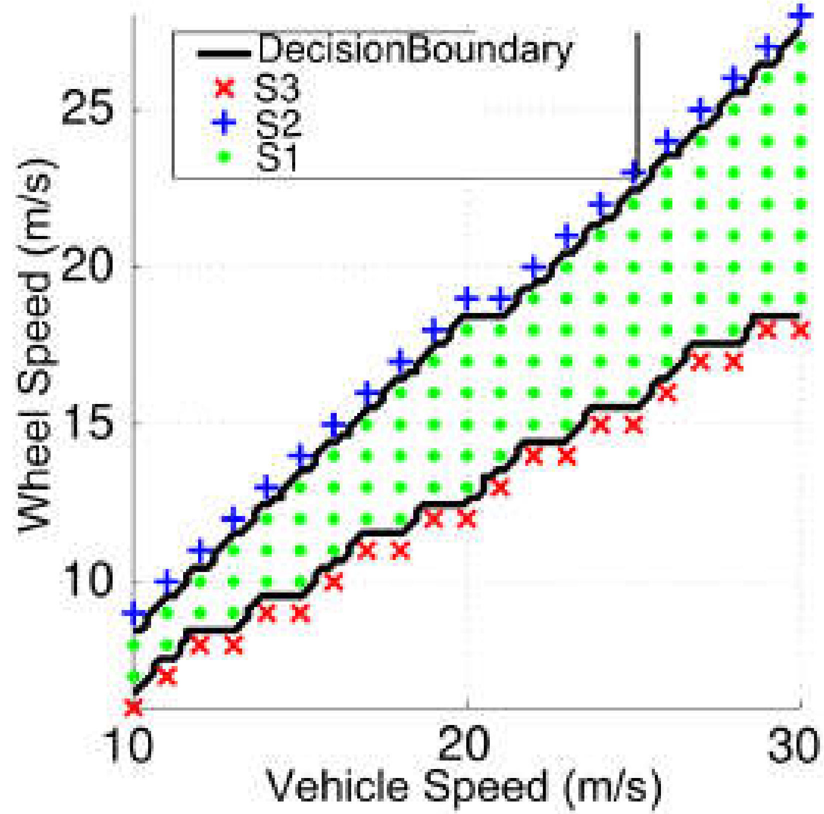
ABS in a Car

- State vector
  - [vehicle speed v, wheel speed ω]
- Real-time control algorithm:
  - Proportional Integral Derivative (PID)
- SSC: in order to have a final stopping distance smaller than a threshold, the slip ratio must be within a certain range
  - Slip ratio = [0.05, 0.25]

SSC and the state sub-spaces



(a) SSC



(b) Sub-spaces in S^3

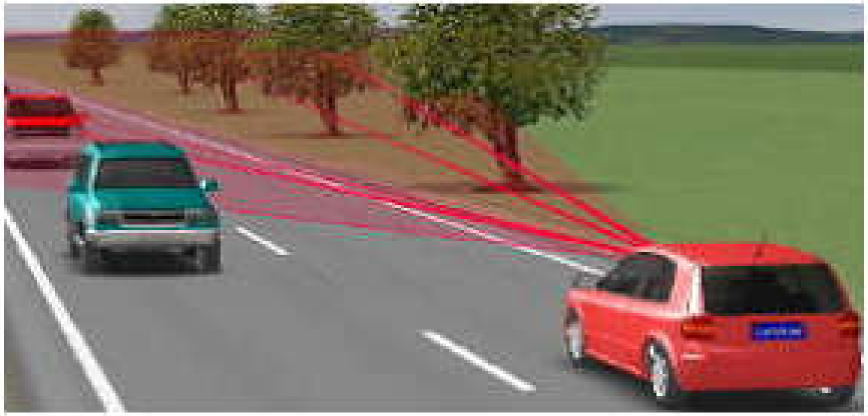
Training process for 3 ML techniques

- **Logistic Regression, Neural Networks, SVM all achieved 100% training accuracy**
- **Using 15, 93 or 138 parameters**

Algorithm	*LR*	*NN*	*SVM*
No. of Parameters	15	93	138
Accuracy	100%	100%	100%

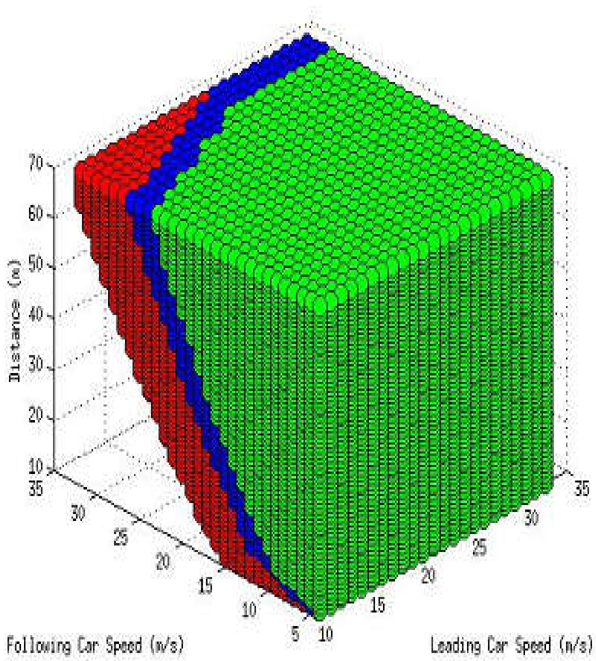
Platoon System (Automated Highway)

- **An example of an application with multiple individuals systems communicating with each other**
- **Carsim: a commercial software for automotive design, can simulate automated highway - integrated with our SW tool**

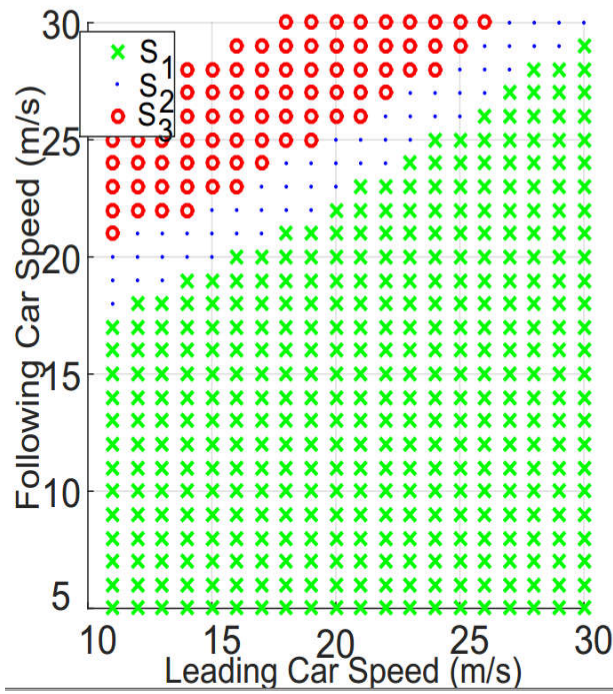


- **Experiments:**
- **A leader-follower system**
  - **Ensure safety - do not allow cars to collide**
  - **Following car uses a sensor to measure distance from leading car**
  - **Leading car sends its speed wirelessly to following car**

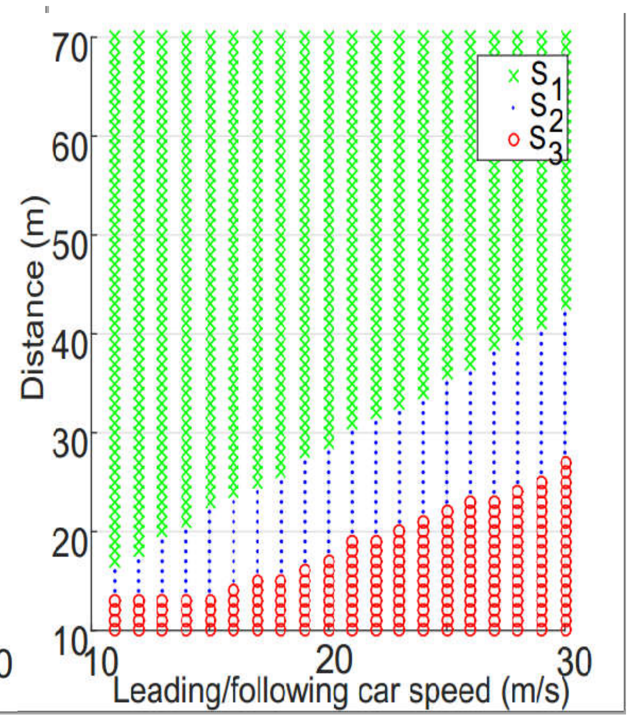
Sub-spaces of the Following Car



(a) 3D Plot for Sub-spaces



(b) Cross Section Plot with Distance Fixed at 40 meters

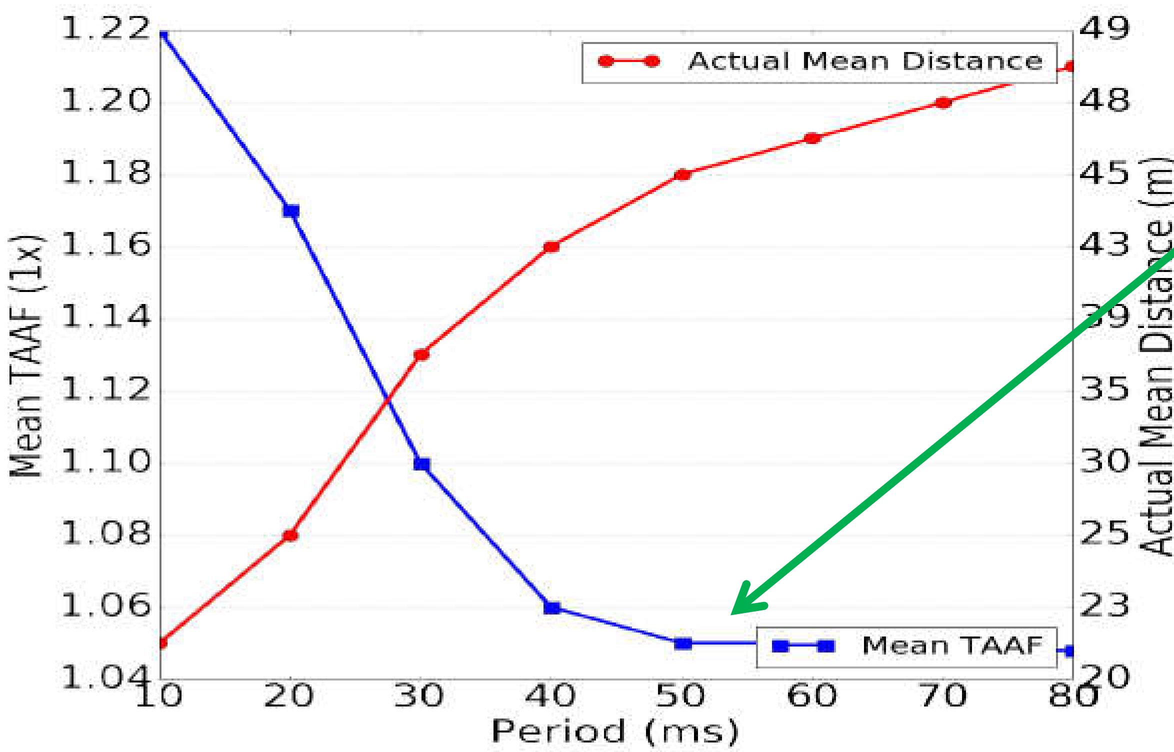


(c) Cross Section Plot with Leading Car Speed Same as Follower

Platoon Case Study with Multiple Task Versions

- **Each control task has two versions:**
  - **Version 1 (complex version):**
    - **Constant Time Gap algorithm for Adaptive Cruise Control**
    - **Period: 10 ms to 80 ms**
  - **Version 2 (the simple version):**
    - **PID with pre-determined desired velocity and distance**
- **The distance between two cars is the quality of control constraint**
- **The version for the control task will switch during the drive depending on the current sub-space**

Trade-off between Reliability & Quality of Control



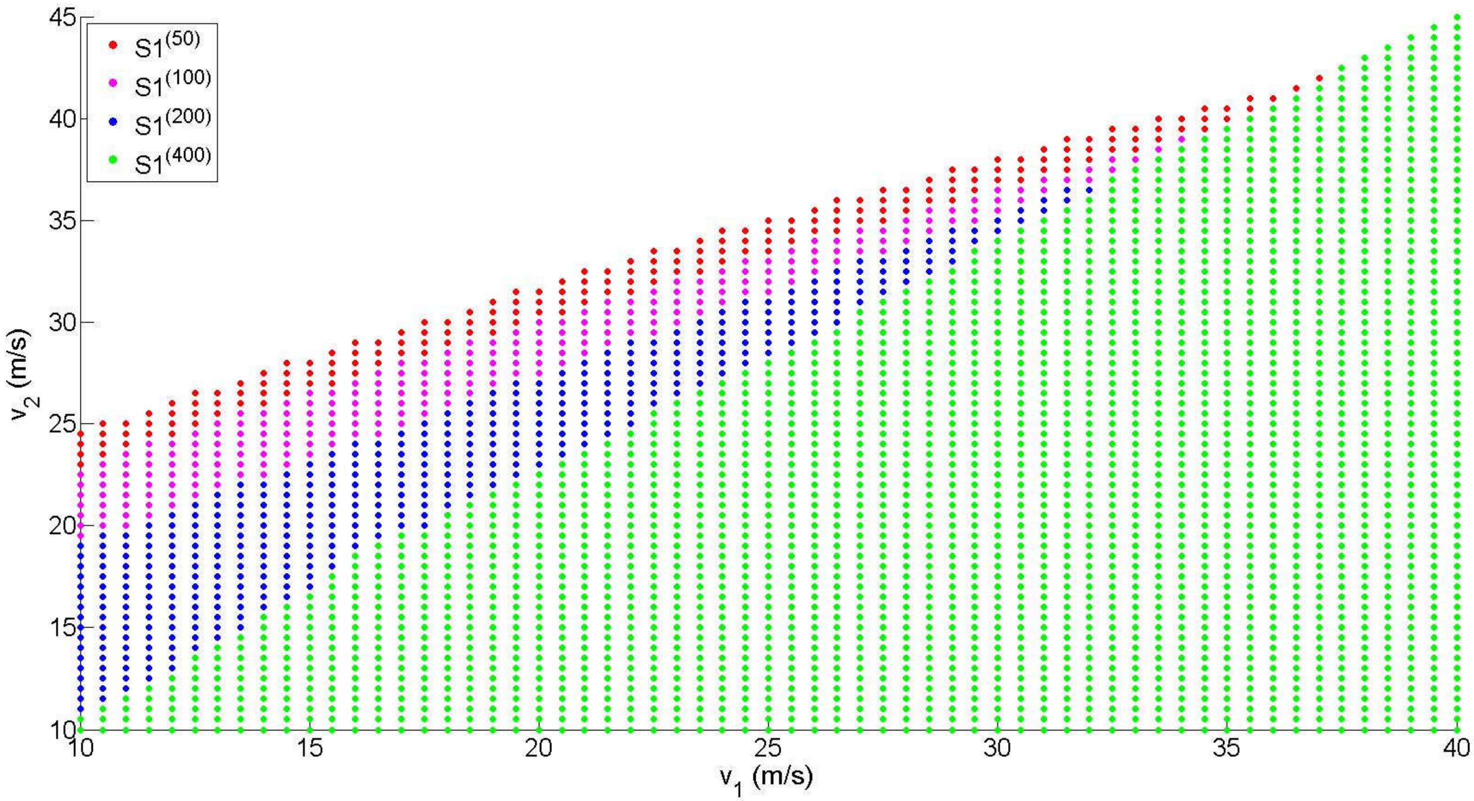
Execute the simpler version (PID) more often

TAAF: Thermal Age Acceleration Factor

Comparing classification schemes - Platoon

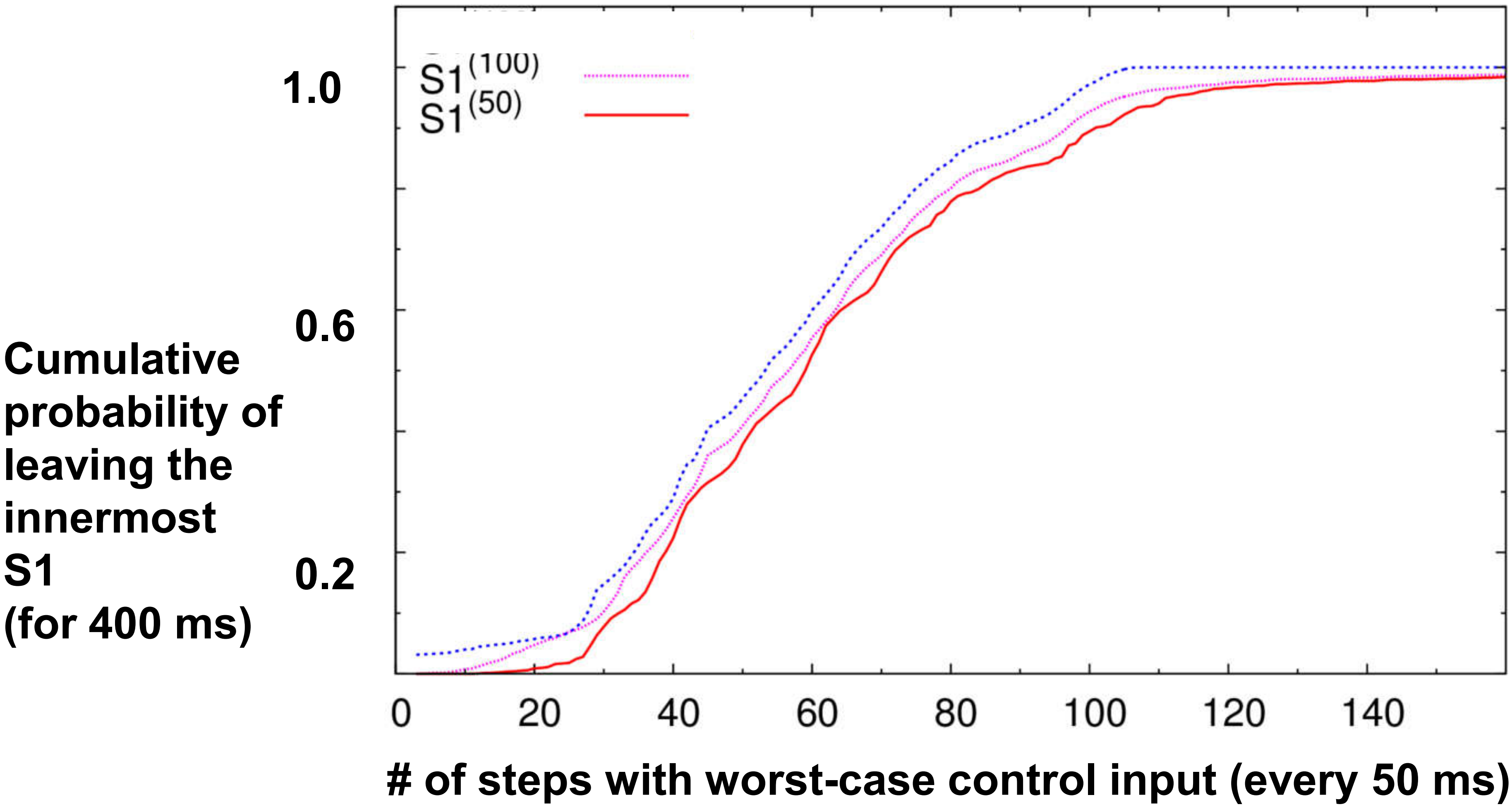
Algorithm	*LR*	*NN*	*SVM*
No. of Parameters	15	153	788
Accuracy	78.56%	99.58%	99.62%

Check for benign faults and then for malicious ones

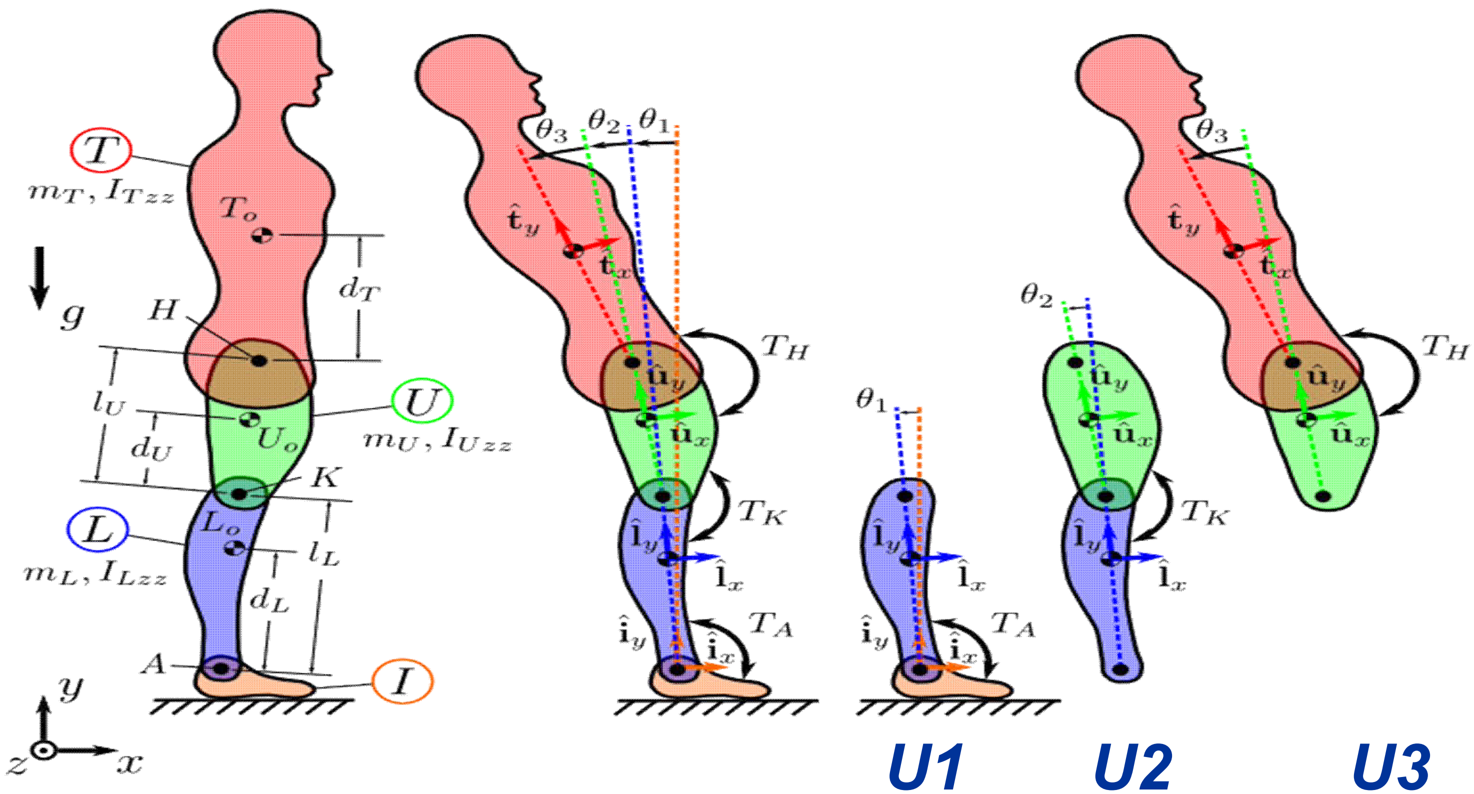


Task_period=50ms; Innermost S1 defined for Task_period=400ms

of steps with wrong control to exit innermost S1

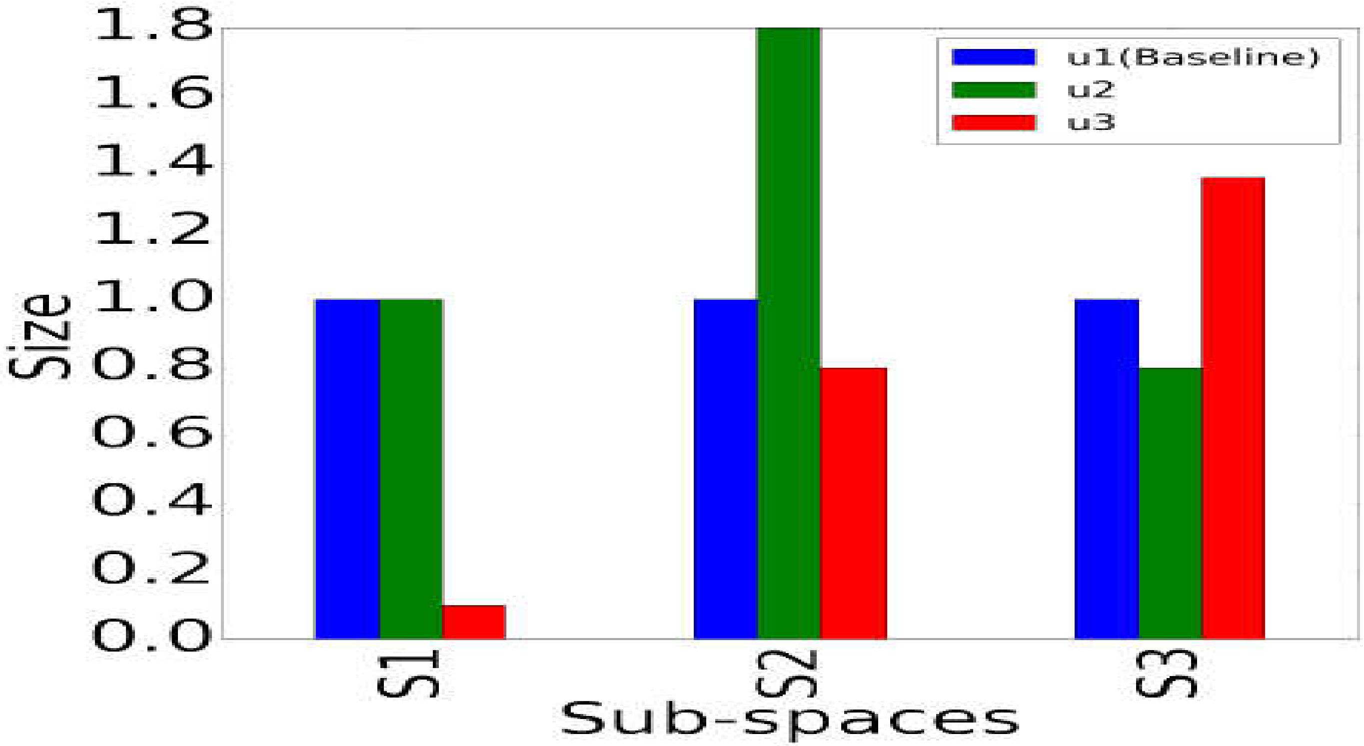


Humanoid Robot



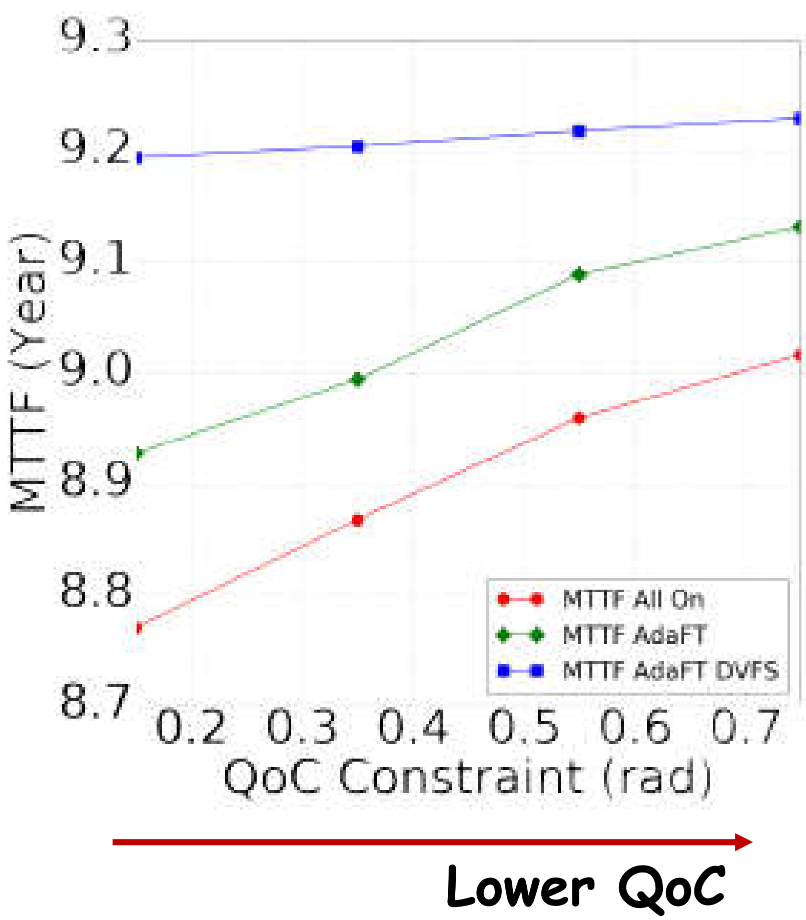
Three control tasks, *U1*, *U2* and *U3*, adjusting the torques at the ankle, knee and hip, respectively

Sub-spaces and classification schemes



Algorithm	Neural Network	Random Forest	Decision Tree
Accuracy	99.6%	97%	97%
Prediction Time	3ms	1ms	0.0096ms

Reliability vs Quality of Control (QoC)



Developed the AdaFT tool that includes the classification process and system optimization to determine the tasks' version and rate

Conclusions

- Benefits of monitoring the current state of the physical plant
  - Achieve high reliability at a lower cost
  - Detect malicious attacks targeting the physical plant's operation (rather than attempts to access proprietary information)
    - Such attacks are dangerous in a CPS
  - Allow recovery from some malicious attacks
    - Can always detect and invoke emergency response
  - Must have an efficient scheme to classify the state sub-space in real-time