Health Monitoring in Small Satellite Design

Wendy A. Okolo, Ph.D.

Diagnostics and Prognostics Group
Discovery and Systems Health Area
Intelligent Systems Division
NASA Ames Research Center

Design of Small Satellite Missions for Planetary Studies
European Astrobiology Campus – Nordic Network of Astrobiology – Stockholm University
University of Tartu, Summer School. July 23rd – August 2nd, 2019. Tartu, Estonia

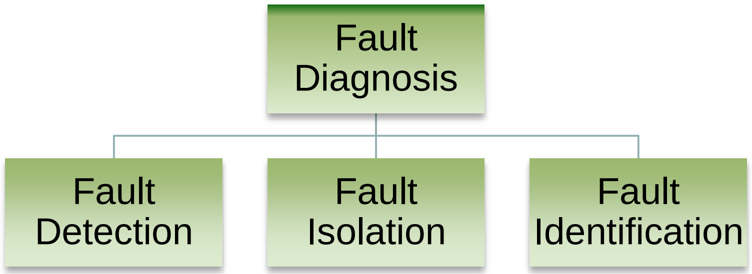
Outline

- Part 1: SHM Overview
    - Fault tree analysis
    - Diagnostics
        - Introduction
        - Methods
    - Prognostics
        - Introduction
        - Methods
        - Applications
- Part 2: SHM for Small Sats
    - Current State of the Art
    - Literature Review
        - Satellite Subsystem Failures
    - Component Modeling
        - Nominal behavior
        - Faulty Behavior
    - SHM Considerations in Mission Design

Systems Health Monitoring

What?

A set of capabilities, information and decision-making products – integrating technologies from systems engineering, reliability, data analytics, for diagnosis, prognosis and health management of complex systems



Why?

- Turning **Data** and **Health State** into **Information** then **Knowledge** then having the **Wisdom** to do something constructive with it
- Need to impact **Availability** and **Overall Enterprise Costs**
- “Do something that improves safety, increases autonomy, and reduces costs”

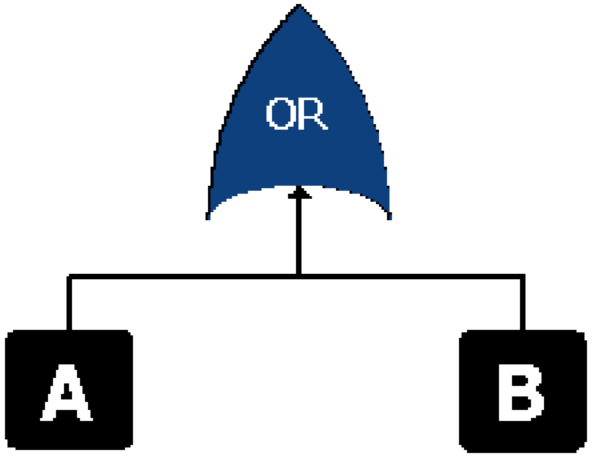
Fault Tree Analysis (FTA)

- Deductive, top-down, graphical analysis that shows failure path/failure chain
- Widely used in operations research and systems reliability
- Developed by Bell Telephone Labs for the USAF in 1962
- One of multiple techniques used for analysis
  - Reliability Block Diagrams (RBDs) – success oriented analysis
  - Fault trees – Failure oriented
- Fault trees: Logic block diagrams that display the state of a system (top event) in terms of the state of its components (basic events)
  - Built using gates (AND and OR gates) and events
  - If occurrence of **either** event causes system to fail → OR gate
  - If occurrence of **both** events cause system failure → AND gate

http://www.weibull.com/basics/fault-tree/index.htm

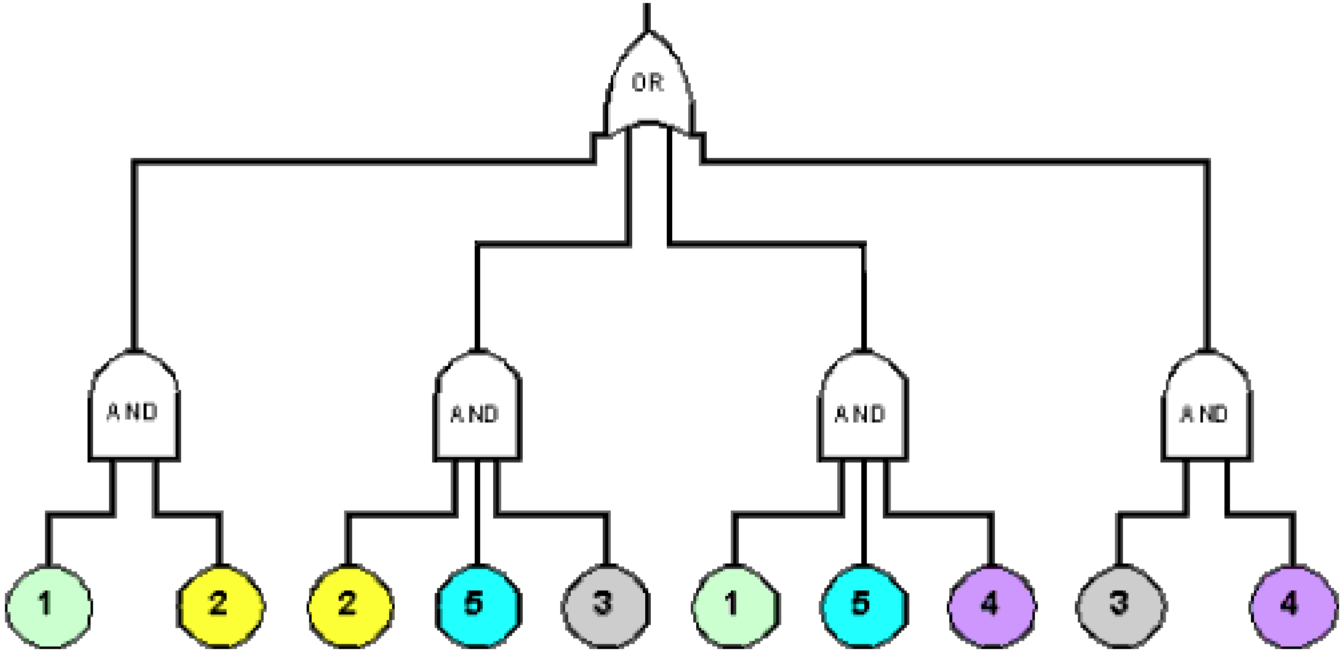
Fault Tree Analysis (FTA)

Fault tree below indicates that failure of A or B (occurrence of event A or B) causes system to fail



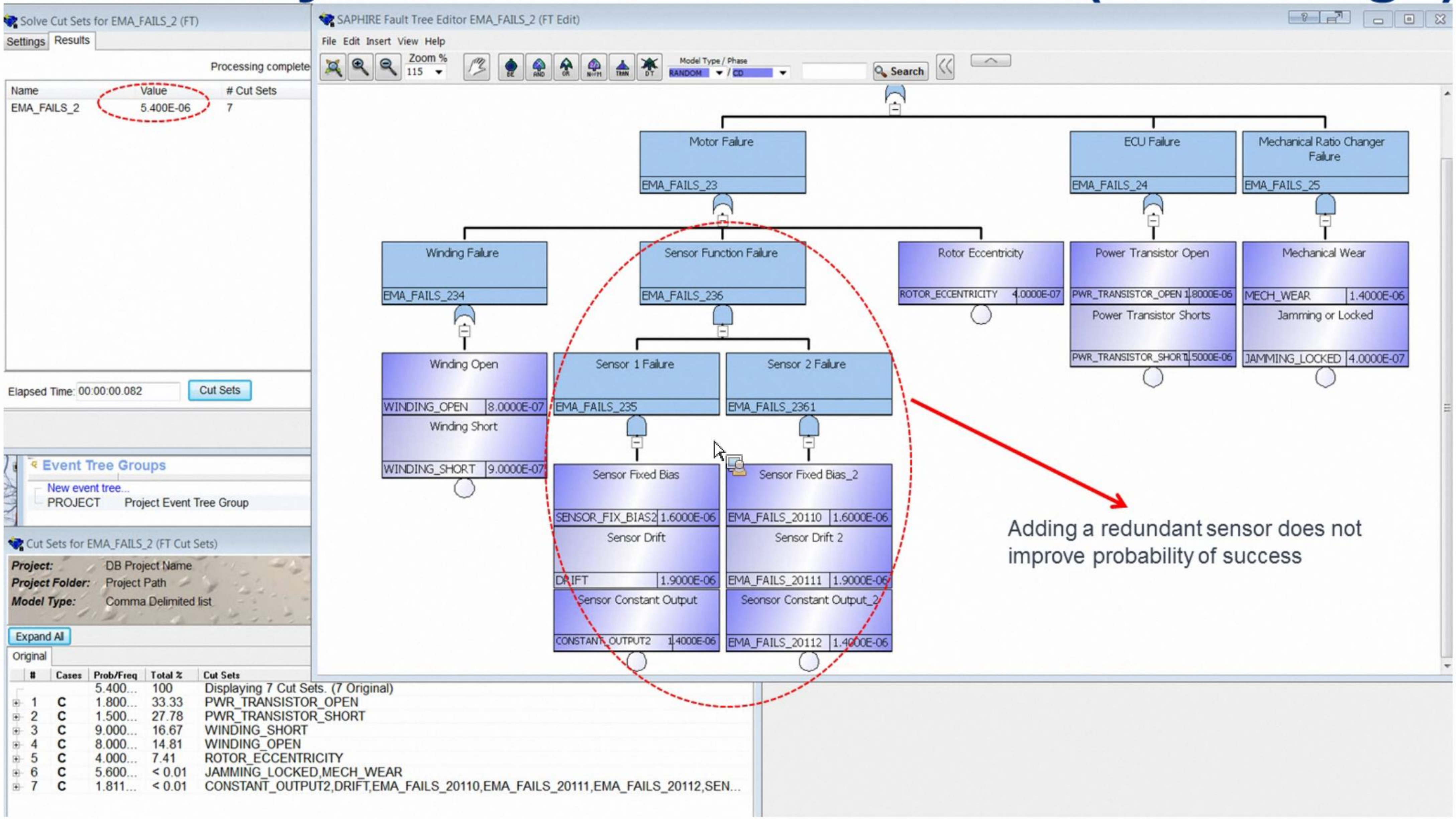
Fault tree below indicates that any of the following failures causes system to fail

Failure of component 1&2 OR 3&4 or 1,5,&4 or 2,5, &3



http://www.weibull.com/basics/fault-tree/index.htm

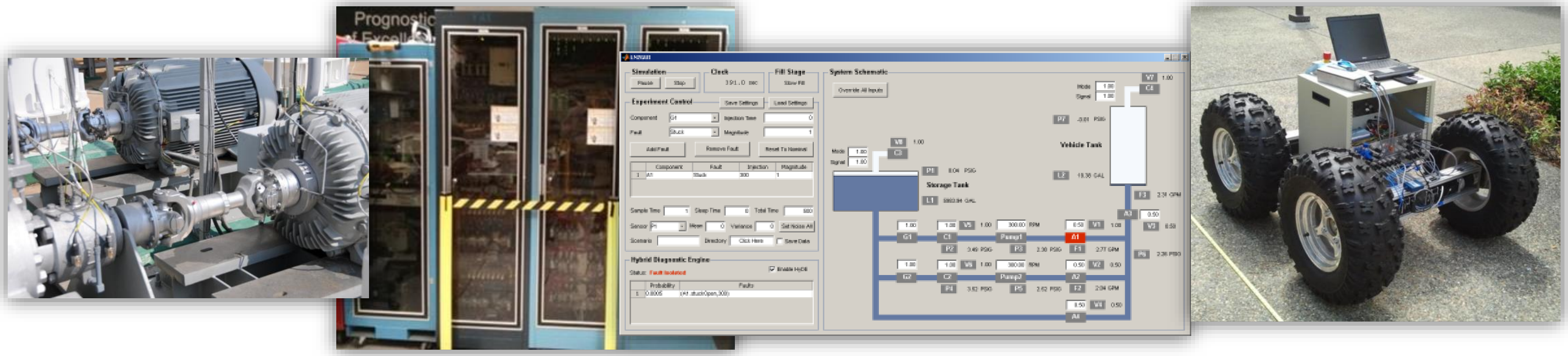
Redundancy assessment – Sensors (No change)



Adding a redundant sensor does not improve probability of success

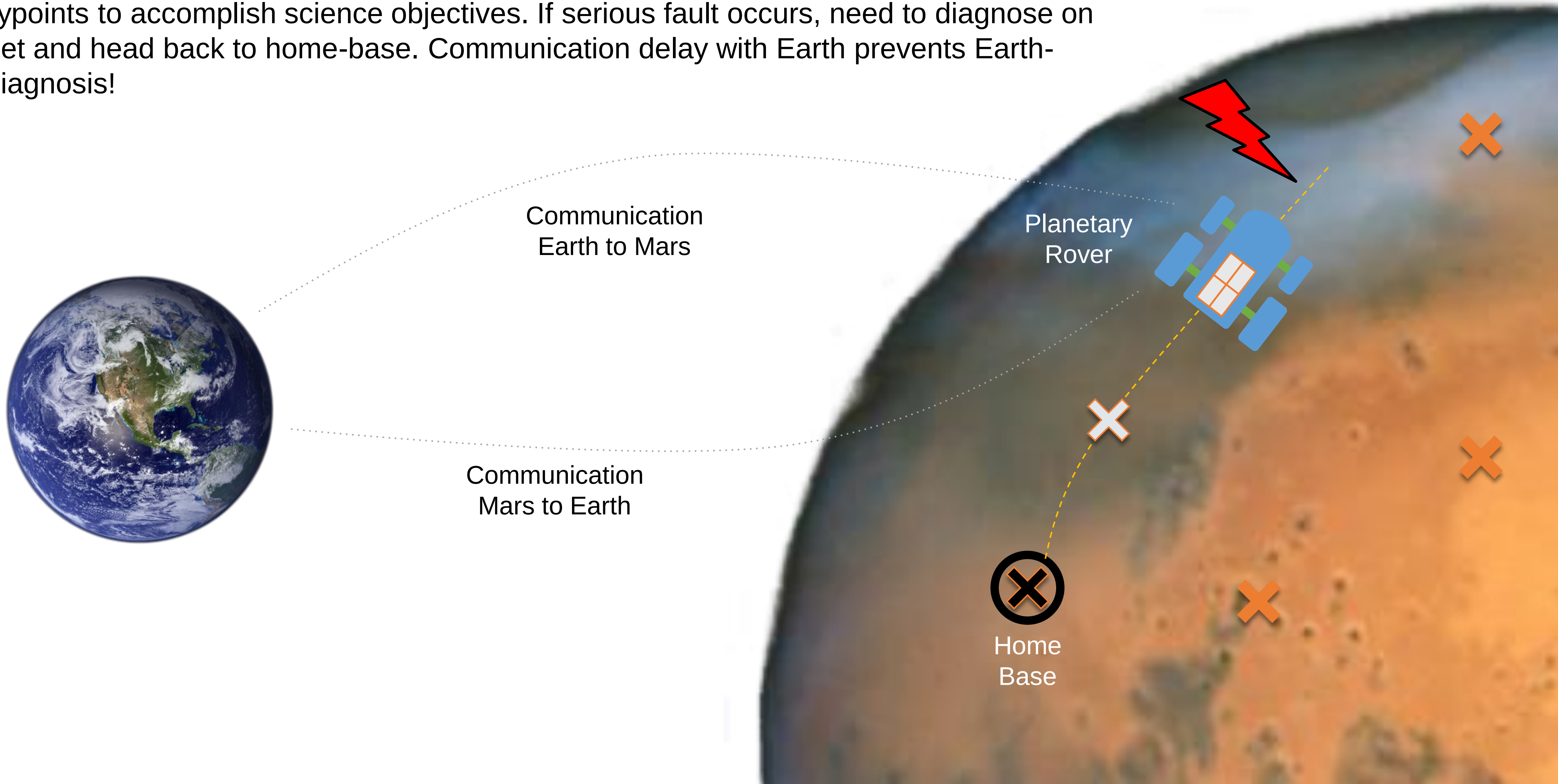
What is Diagnostics?

- Diagnosis = determining the nature and cause of something
- In a health management context, this “something” is a fault
  - Fault = an unexpected change in the dynamics of a system
  - Fault ≠ Failure!
  - Failure = a condition of the system in which it does not meet functional specifications
  - Faults can grow and lead to failure, or a fault may be significant enough in magnitude, or a severe enough change in configuration, such that the system will have failed (due to the occurrence of the fault)



Why Diagnostics?

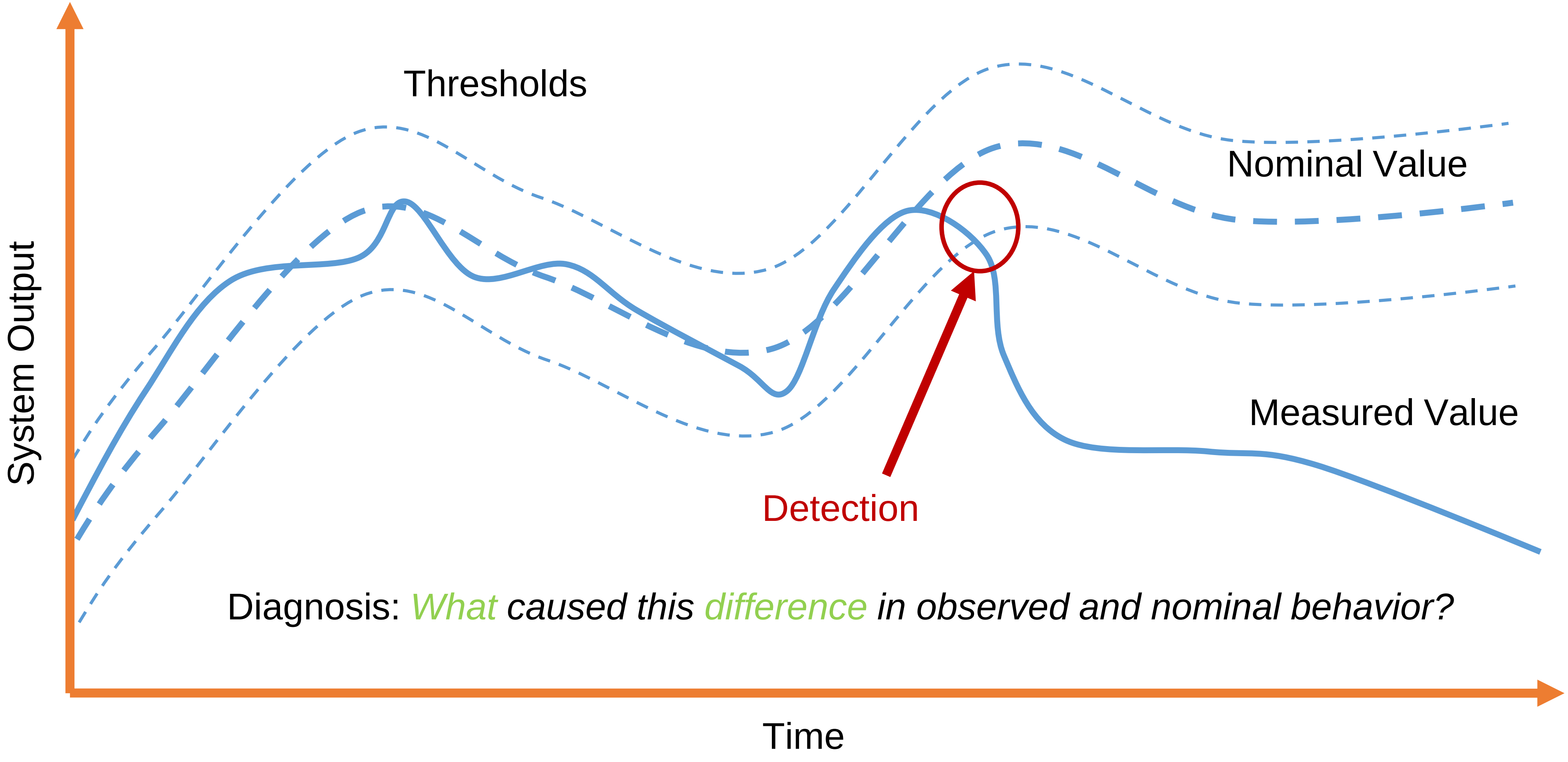
Example: Rover Mission
Visit waypoints to accomplish science objectives. If serious fault occurs, need to diagnose on the planet and head back to home-base. Communication delay with Earth prevents Earth-based diagnosis!



Why Diagnostics?

- Diagnostics informs decision-making
- Which components to repair/replace
- Inform fault mitigation
- Inform fault recovery
- Inform functional reallocation
- Diagnostics informs prognostics
  - What is (are) the dominant aging/degradation mode(s)
  - Can we complete operation(s)/mission in presence of fault?

The Basic Idea



Diagnosis: *What caused this difference in observed and nominal behavior?*

Nominal vs Faulty Behavior

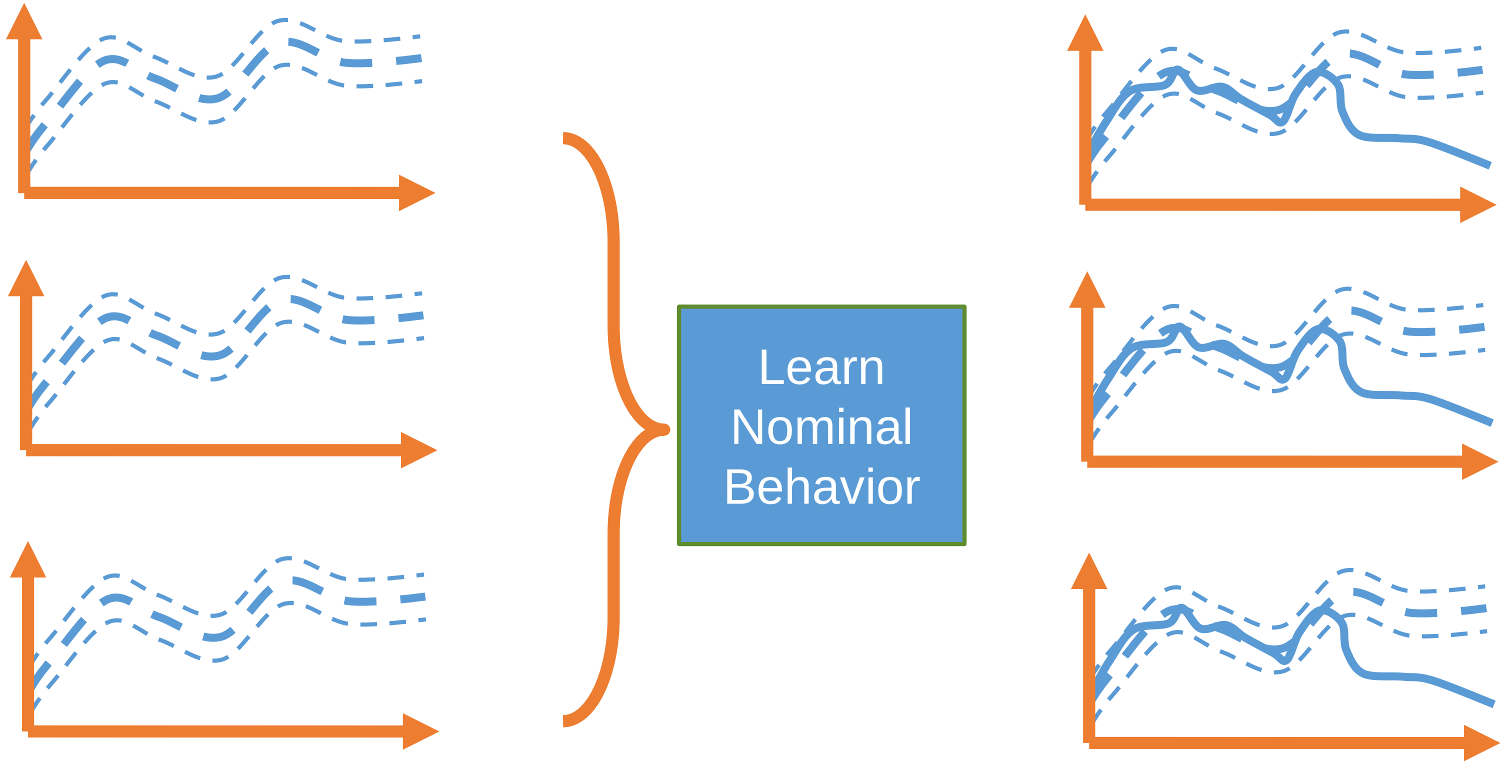
- What is “nominal” behavior?
  - Nominal is defined w/r/t some knowledge about the system, a reference behavior
  - Comes from expert knowledge, known operating limits, physics model, machine learning approaches, etc.
- In *model-based* diagnostics, the reference comes from a model that explicitly describes the nominal behavior
  - Models can be static or dynamic
  - Models can be used for simulation

Why Model-based Diagnostics?

Machine Learning Approach:
Detection: Classify between nominal and non-nominal behavior
Isolation: Classify between different classes of faulty behavior

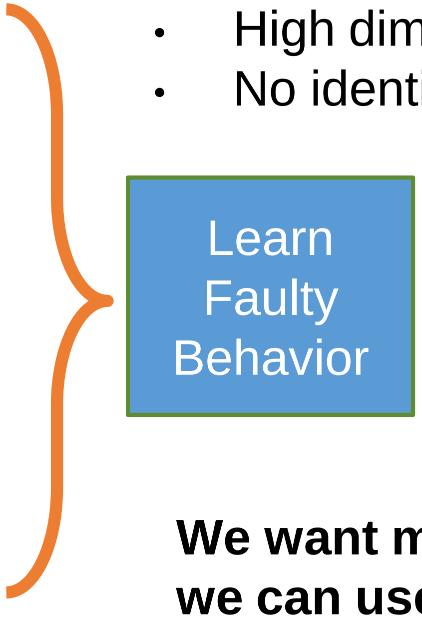
Problems:

- Lack of faulty data instances
- No explanatory power from models
- High dimensionality
- No identification



Instances of Nominal Behavior

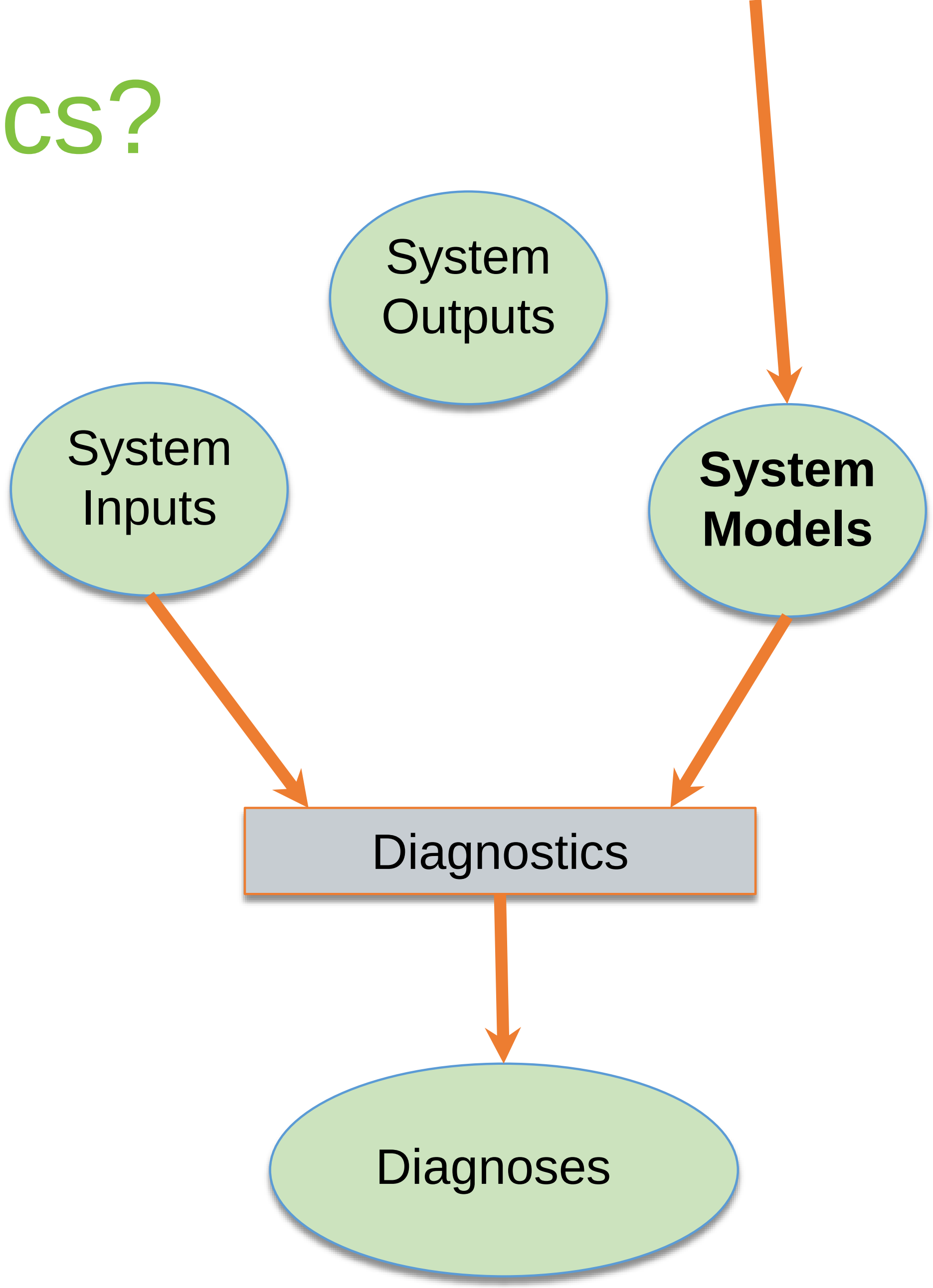
Instances of Faulty Behavior



We want models that we can use to *reason* over…

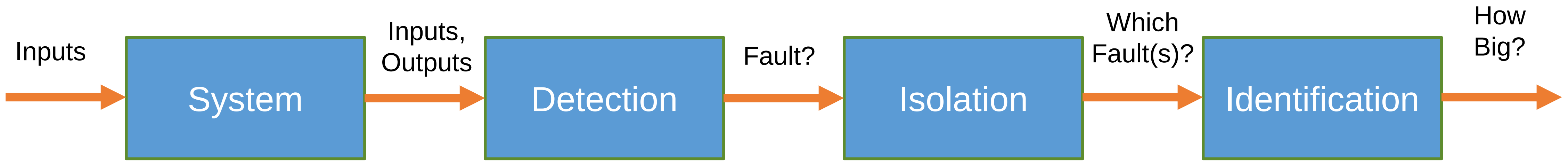
Why Model-based Diagnostics?

- Models have explanatory power
  - Causal reasoning
  - Explicit representation of faults
- Develop general model-based algorithms
  - Models used for diagnosis are *inputs*
  - Algorithms do not change for a new system, only the model changes



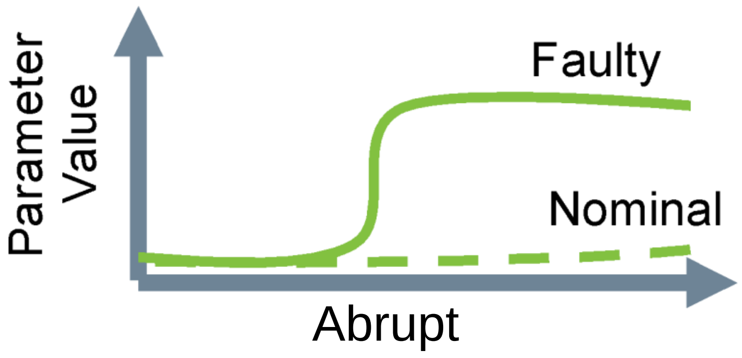
Constituent Problems

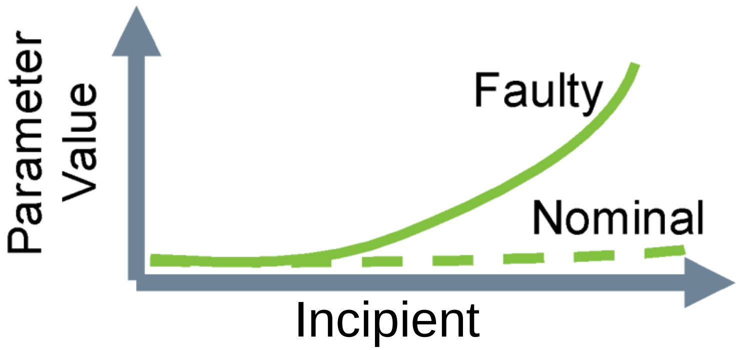
- Fault detection = determination of whether the system is not operating nominally
- Fault isolation = determination of the root cause(s) of the unexpected system behavior
- Fault identification = determination of the magnitude of the fault (if applicable)



Characterizing Faults

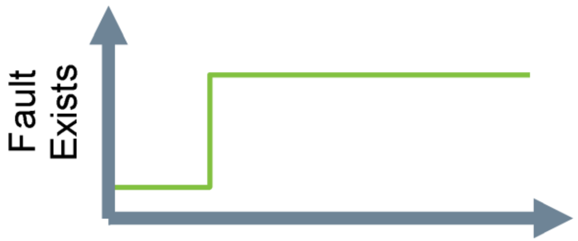
- Abrupt: Change in parameter value faster than the sampling frequency
- Incipient: Change in parameter value slower than the sampling frequency
  - Can be linear, exponential, or arbitrary degradation
  - Prognostics usually pertains to incipient faults
- Abrupt faults can be easier/faster to detect compared to incipient faults
- Dynamics of fault is different from dynamics of measurements/observations
  - E.g., abrupt fault can present incipient change in measurements



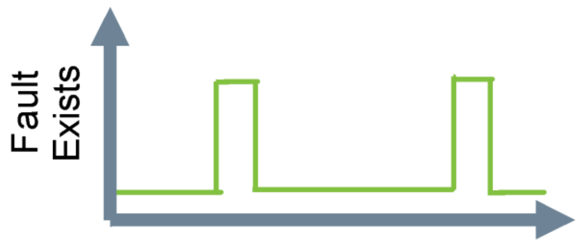


Characterizing Faults

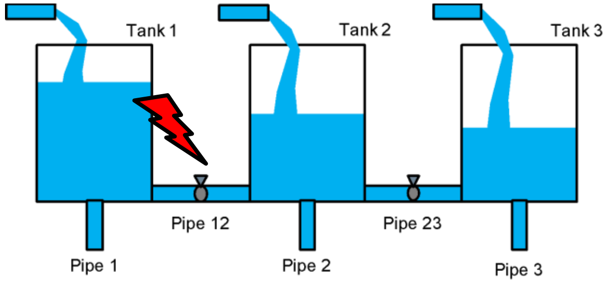
- Persistent vs Intermittent
  - Persistent: Once manifested, the fault persists
  - Intermittent: Fault manifests intermittently
- Discrete vs Parametric
  - Discrete faults involve undesired change in system or model structure, e.g., valve on Pipe 12 stuck closed
  - Parametric faults involve undesired change in system or model parameter, e.g., Pipe 2 is clogged



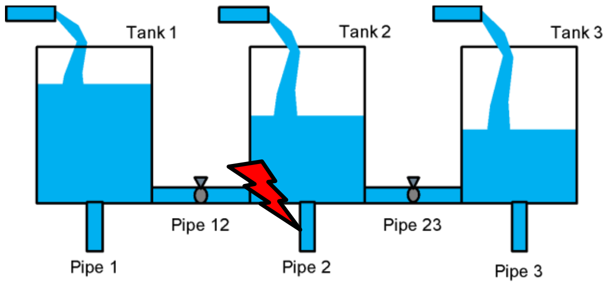
Persistent



Intermittent



Discrete



Parametric

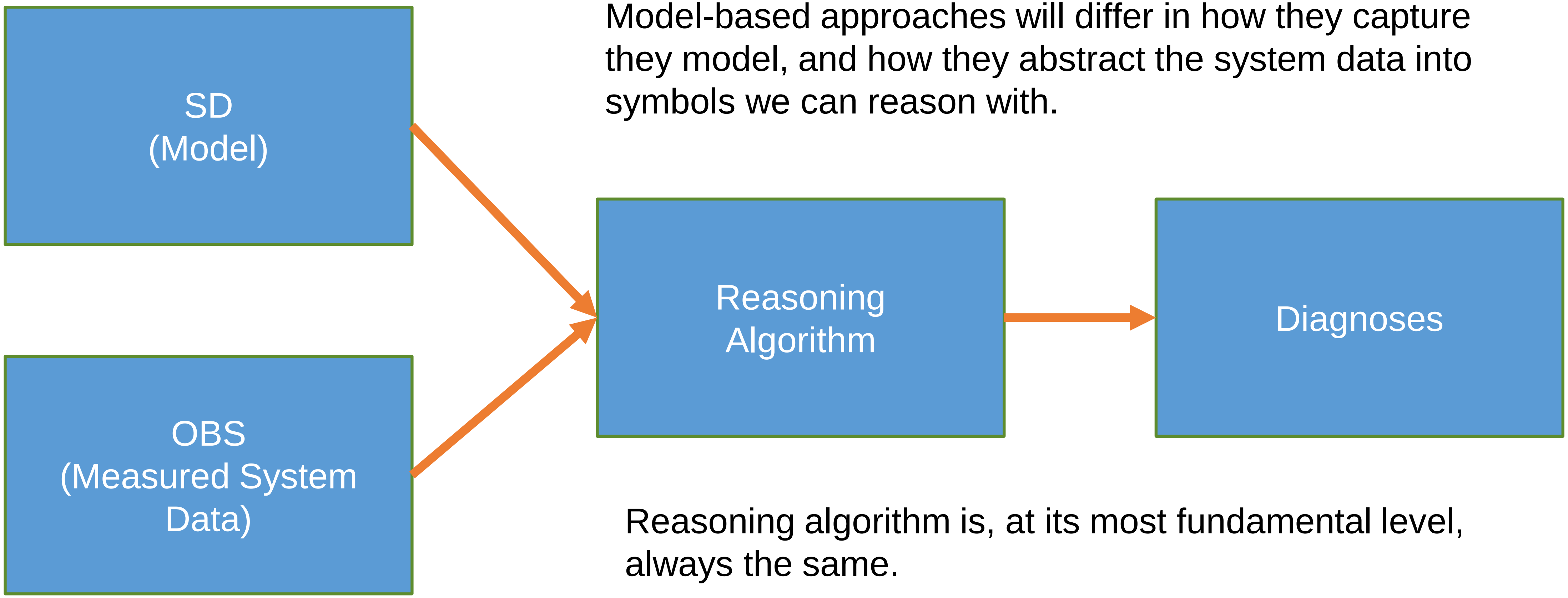
Diagnosis

- Some events are observable, and some are not
- We need to estimate what the possible state is, as that determines whether an unobservable fault event may or may not have occurred
- Valve example:
  - We have a sensor that reports the position of a valve at a regular interval, 0 for open and 1 for closed
  - Say that we observe the event sequence: Open, 0, 0, 0, Close, 1, 1
    - Is this nominal? Maybe
  - Say that we observe: Open, 0, 0, Close, 0, 0
    - Is this nominal? No

A reasoning algorithm would map events to diagnoses. This is basically "hard-coding" the reasoning algorithm.

Summary of Approach

Model-based approaches will differ in how they capture they model, and how they abstract the system data into symbols we can reason with.



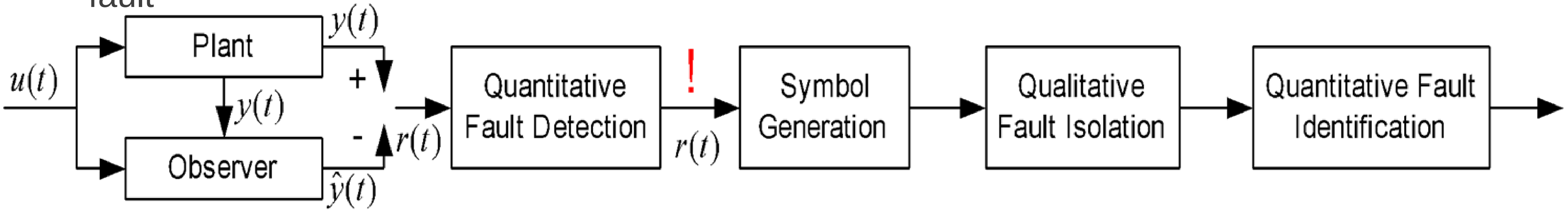
Reasoning algorithm is, at its most fundamental level, always the same.

Practical Considerations

- Such an approach doesn't work well for dynamic systems, and hides many issues
  - What about sensor noise?
  - How to represent dynamic behavior in this framework?
  - How to reason over time?
  - Computational complexity?
- However, the algorithmic approach is sound and forms the basis for most model-based diagnostic reasoning algorithms
  - Describe nominal and faulty behavior
  - Reason over discrepancies between nominal and observed behavior
  - Determine which faults would be consistent with the observations
  - This is always the approach in model-based diagnosis!

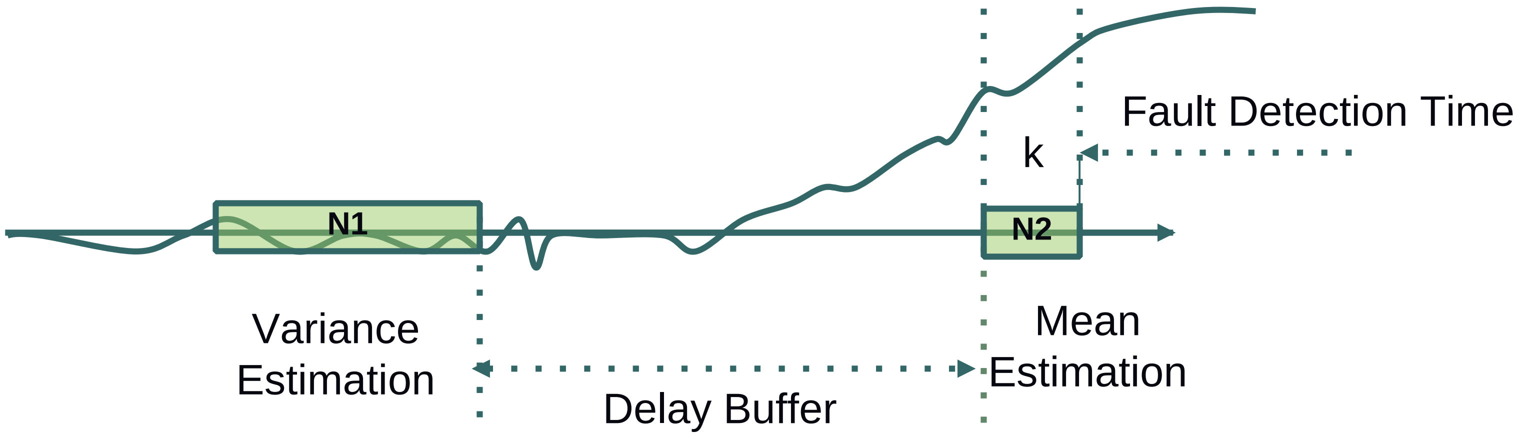
Continuous System Diagnosis: One Approach

- Tackles fault detection, isolation, and identification problems
- Assumes single, persistent parametric faults
  - Can be either abrupt or incipient
- Changes in parameter produces changes in system outputs w/r/t no parameter change
  - Eventually all (causally related) sensors effected
- Need to reason over those changes
  - Have a model of how measurements should deviate given different possible faults
  - Noting the order in which different measurement deviations are observed can also give us clues about the fault
  - Compare observed deviations to expected deviations for each fault candidate to diagnose true fault



Residual Generation and Fault Detection

- Residual Generation
  - Observer (eg, Kalman filter, unscented Kalman filter, particle filter) based on nominal local submodel computes nominal behavior as a reference
  - Residual computed as measured value minus reference value
- Fault Detection
  - Nominally residual is approximately zero
  - Fault detected when residual deviation from zero is statistically significant
  - Usually there is a delay between fault occurrence and fault detection – cannot be avoided

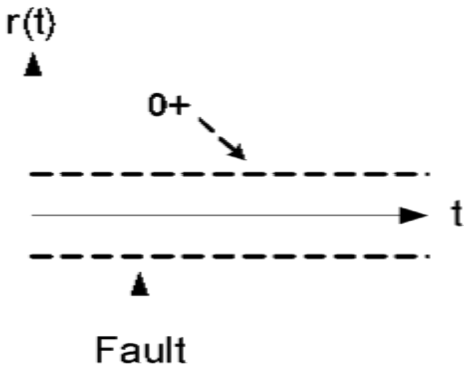


Fault Isolation

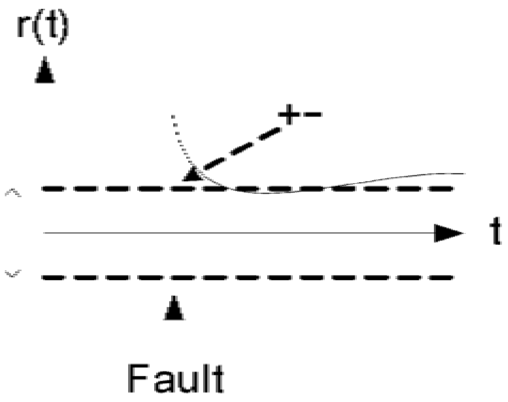
- Faults are isolated by comparing the qualitative deviation in measurements with the predicted fault signatures
  - Example: Consider fault set F = $\{C_1^-, R_2^+, C_2^-\}$ and measurement set M = $\{p_1, p_3\}$ all faults can be uniquely isolated
- Therefore, a system with faults F = $\{f_1, \dots, f_l\}$, and measurements M = $\{m_1, \dots, m_n\}$, is **diagnosable** if all single faults in F can be uniquely isolated using M
  - I.e., there is at least one distinguishing fault signature between f_i and all other faults in the system.

Fault	p_1	p_3
~~C_1^-~~	~~+-~~	~~0+~~
~~R_2^+~~	~~0+~~	~~0+~~
C_2^-	0+	+-

Fault Signature Matrix



Pressure p_1



Pressure p_3

Fault Identification

- Parameter estimation problem
  - Identify new (fault) parameter value, given observed faulty behavior
  - Several algorithms solve this problem
- One approach:
  - Determine an estimation window
    - Use data from *before* t_d (detection time) to t (current time)
  - Run an observer through that window, with the state vector augmented with the fault parameter (joint state-parameter estimation)
- Alternate approach
  - Derive submodel expressing unknown parameter as function of known/measured variables

Challenges

- Modeling!
  - At what level of abstraction should one model?
  - How to combine results from different levels of abstractions?
- Online simulation
  - Problems with dynamic systems: initial conditions
- What is the source of complexity?
  - Complex systems or large systems (# components)
  - Exponential number of multiple fault candidates

Adapted from Anibal Bregon’s talk on “Consistency-Based Diagnosis” at PHME14, Nantes, France.

Selected Bibliography

- Armengol, J., Bregon, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., Olive, X., Pulido, B., and Travé-Massuyès, L. Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09, pages 1480–1485, Barcelona, Spain, 2009.
- Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M. Diagnosis and FaultTolerant Control. Springer, 2006.
- Biswas, G., Cordier, M., Lunze, J., Travé-Massuyè, L., and Staroswiecki, M. Diagnosis of complex systems: bridging the methodologies of the FDI and DX communities. IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, 34(5):2159–2162, 2004.
- Cordier, M., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., and Travé- Massuyès, L. Conflicts versus Analytical Redundancy Relations: a comparative analysis of the Model-based Diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics, 34(5):2163–2177, 2004.
- Guckenbiehl, T. and Schaefer-Richter, G. SIDIA: Extending prediction based diagnosis to dynamic models. In Proceedings of the 1st International Workshop on Principles of Diagnosis, DX90, pages 74–82, Stanford, California, USA, 1990.

Selected Bibliography

- Krysander, M., Aslund, J., and Nyberg, M. An Efficient Algorithm for Finding Minimal Over-constrained Sub-systems for Model-based Diagnosis. IEEE Trans. on Systems, Man, and Cybernetics. Part A: Systems and Humans, 38(1):197–206, 2008.
- Loiez, E. and Taillibert, P. Polynomial temporal band sequences for analog diagnosis. In Proceedings of the 15th International Joint Conference on Artificial Intelligence, IJCAI97, pages 474–479, Nagoya, Japan, 1997.
- Mosterman, P. and Biswas, G. Diagnosis of continuous valued systems in transient operating regions. IEEE Transactions on Systems, Man, and Cybernetics, 29(6):554– 565, 1999.
- Staroswiecki, M. Fault Diagnosis and Fault Tolerant Control, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems. Eolss Publishers, Oxford, UK, 2002.
- Struss, P. Model-based diagnosis for industrial applications. In ColloquiumApplications of Model-based Reasoning, Savoy Place, London, United Kingdom. Institute of Electrical Engineers (IEE), 1997.
- Travé-Massuyès, L. and Milne, R. Gas-Turbine Condition Monitoring Using Qualitative Model-Based Diagnosis. IEEE Expert, 12(3):22–31, 1997.
- I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Designing Distributed Diagnosers for Complex Continuous Systems," IEEE Transactions on Automation Science and Engineering, vol. 6, no. 2, pp. 277-290, April 2009.
- I. Roychoudhury, M. Daigle, A. Bregon, and B. Pulido, "A Structural Model Decomposition Framework for Systems Health Management," in Proceedings of the 2013 IEEE Aerospace Conference, March 2013.
- A. Bregon, M. Daigle, and I. Roychoudhury, "An Integrated Framework for Model-Based Distributed Diagnosis and Prognosis," in Proceedings of the Annual Conference of the Prognostics and Health Management Society 2012, Minneapolis, MN, September 2012.
- I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Distributed Diagnosis in Uncertain Environments Using Dynamic Bayesian Networks," in Proceedings of the 18th Mediterranean Conference on Control and Automation (MED'10), Marrakech, Morocco, June 2010.

Selected Bibliography

- Travé-Massuyés, L., Escobet, T., and Olive, X. Diagnosability analysis based on component supported analytical redundancy relations. IEEE Trans. Syst. Man Cy. A., 36(6), 2006.
- Washio, T., Sakuma, M., and Kitamura, M. A new approach to quantitative and credible diagnosis for multiple faults of components and sensors. Artificial Intelligence, 91(1):103–130, 1997.
- Williams, B. and Millar, B. Automated decomposition of Model-based Learning Problems. In Proceedings of the 7th International Workshop on Principles of Diagnosis, DX96, pages 258–266, Val Morin, Quebec, Canada, 1996.
- Williams, B. and Millar, B. Decompositional Model-based Learning and its analogy to diagnosis. In Proceedings of the 15th National Conference on Artificial Intelligence, AAAI98, Madison, Wisconsin, USA, 1998.
- M. Daigle, A. Bregon, X. Koutsoukos, G. Biswas, and B. Pulido, "A Qualitative Event-based Approach to Multiple Fault Diagnosis in Continuous Systems using Structural Model Decomposition," Engineering Applications of Artificial Intelligence, vol. 53, pp. 190-206, August 2016.
- M. Daigle, I. Roychoudhury, and A. Bregon, "Qualitative Event-based Diagnosis Applied to a Spacecraft Electrical Power Distribution System," Control Engineering Practice, vol. 38, pp. 75-91, May 2015.
- M. Daigle, A. Bregon, and I. Roychoudhury, "Distributed Prognostics Based on Structural Model Decomposition," IEEE Transactions on Reliability, vol. 63, no. 2, pp. 495-510, June 2014.
- A. Bregon, M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, and B. Pulido, "An Event-based Distributed Diagnosis Framework using Structural Model Decomposition," Artificial Intelligence, vol. 210, pp. 1-35, May 2014.
- M. Daigle, X. Koutsoukos, and G. Biswas, "An Event-based Approach to Integrated Parametric and Discrete Fault Diagnosis in Hybrid Systems," Transactions of the Institute of Measurement and Control, Special Issue on Hybrid and Switched Systems, vol. 32, no. 5, pp. 487-510, October 2010.
- M. Daigle, I. Roychoudhury, G. Biswas, X. Koutsoukos, A. Patterson-Hine, and S. Poll, "A Comprehensive Diagnosis Methodology for Complex Hybrid Systems: A Case Study on Spacecraft Power Distribution Systems," IEEE Transactions of Systems, Man, and Cybernetics, Part A, Special Section on Model-based Diagnosis: Facing Challenges in Real-world Applications, vol. 4, no. 5, pp. 917-931, September 2010.
- M. Daigle, X. Koutsoukos, and G. Biswas, "A Qualitative Event-based Approach to Continuous Systems Diagnosis," IEEE Transactions on Control Systems Technology, vol. 17, no. 4, pp. 780-793, July 2009.

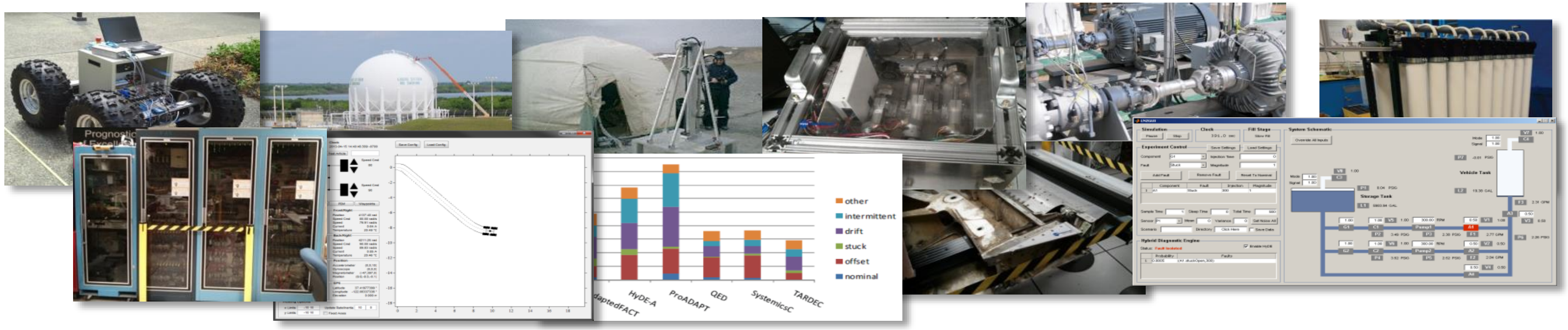
Additional Information Sources

- Conferences
  - PHM: http://www.phmsociety.org/
  - DX: http://dx-2014.ist.tugraz.at/
  - IJCAI: http://ijcai.org/
  - Safeprocess (part of IFAC organization): http://safeprocess15.sciencesconf.org/
  - IFAC world conference: http://www.ifac2014.org/
- Journals
  - Artificial Intelligence Journal
  - International Journal of the PHM Society (IJPHM)
  - Journal of AI Research
  - IEEE Transactions On Systems, Man and Cybernetics
  - AI Communications
  - Control Engineering Practice
  - Engineering Application on Artificial Intelligence
  - …

Adapted from Anibal Bregon’s talk on “Consistency-Based Diagnosis” at PHME14, Nantes, France.

Diagnostics Summary

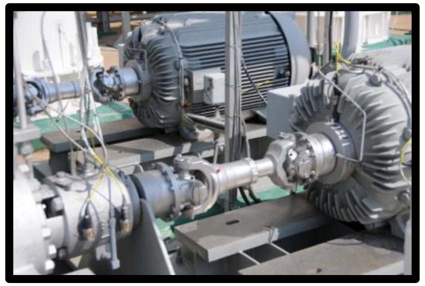
- Diagnostics answers the following questions
  - Is anything broken? – Fault Detection
  - What is broken? – Fault Isolation
  - How bad is the fault? – Fault Identification
  - What can we do about it? – Fault Mitigation, Recovery, & Decision-making
- Benefits include
  - Enhanced system safety: faults are mitigated before catastrophic events
  - Enhanced system performance: supports maintaining system goals in the presence of faults
  - Reduces costs: supports smart troubleshooting and maintenance
- Core focus areas include
  - Algorithm development: hybrid systems diagnosis, multiple fault diagnosis, distributed diagnosis, integrated diagnosis and prognosis, uncertainty management
  - Tool maturation: Livingstone, Hybrid Diagnostic Engine (HyDE), Qualitative Event-based Diagnosis (QED)
  - Verification and validation: metric development, diagnostic algorithm evaluation framework (DXF), diagnostic competitions (DXC)
  - Application to real systems: electrical power distribution systems, planetary rovers, propellant loading systems, environmental control and life support systems, autonomous drilling, electromechanical actuators



PROGNOSTICS

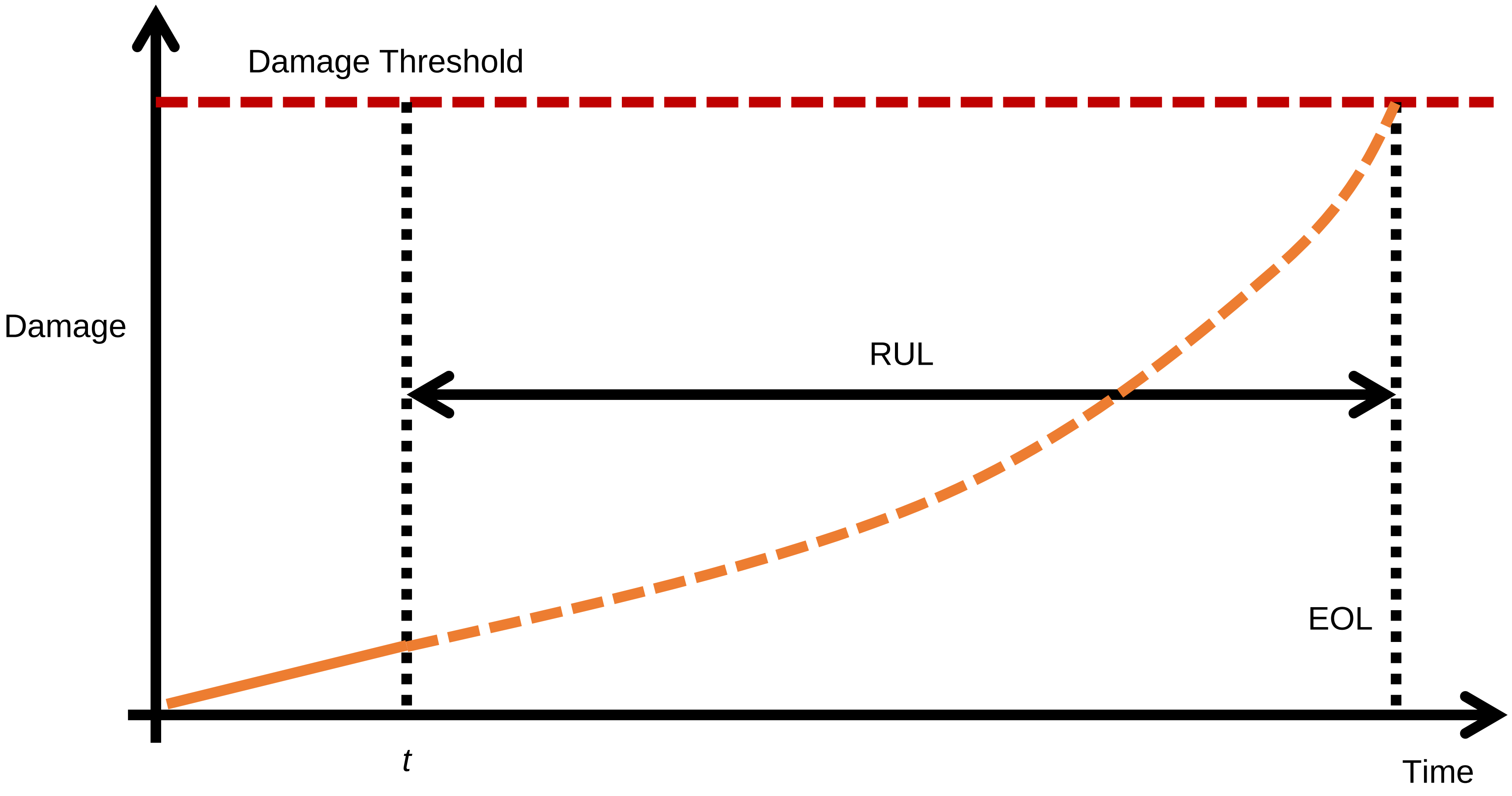
Definitions

- Prognostics = the problem of predicting the future state of a system
  - In the context of this lecture, interested in states that represent failure
- Model-based prognostics = an approach to prognostics in which a model of the system is used for prediction
  - First principles
  - Physics-based
  - Neural network
  - Etc.



- For the purposes of this lecture, we are interested in predicting *failure states*
  - EOL = end of life (time of failure)
  - RUL = remaining useful life (time until failure)

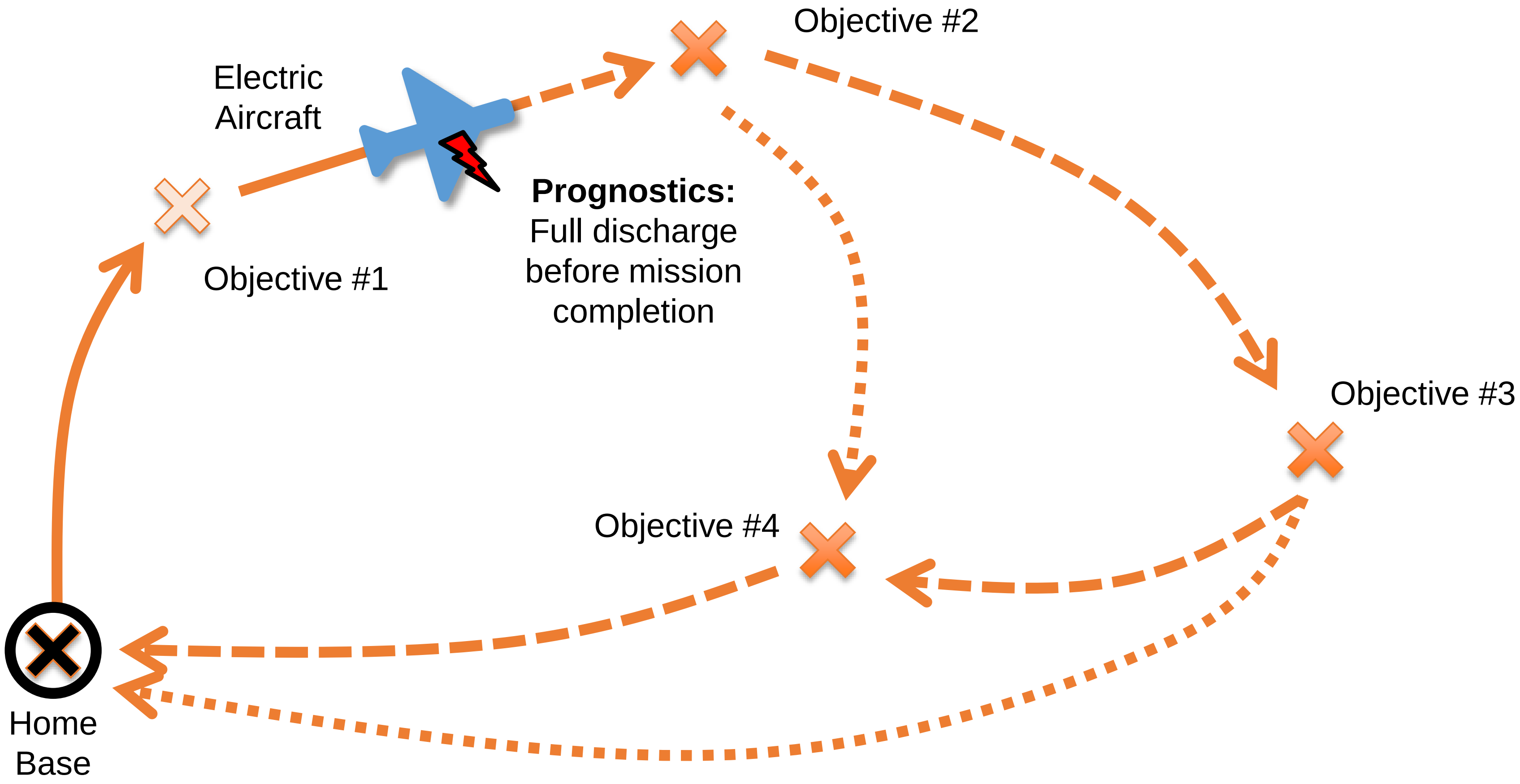
The Basic Idea



Why Prognostics?

Example: UAV Mission

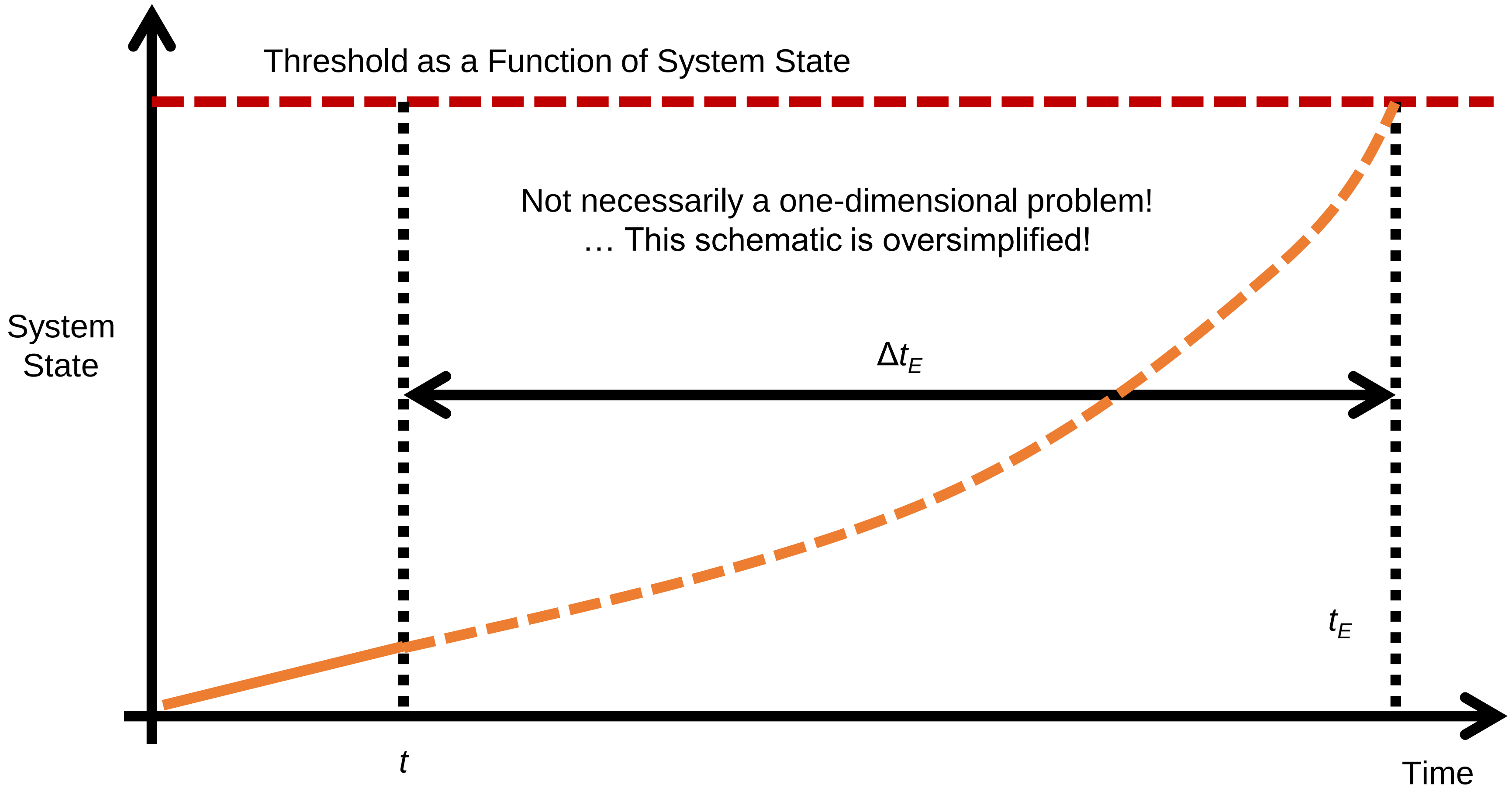
Visit waypoints to accomplish science objectives. Predict aircraft battery end of discharge to determine which objectives can be met. Based on prediction, plan optimal route. Re-plan if prediction changes.



Why Prognostics?

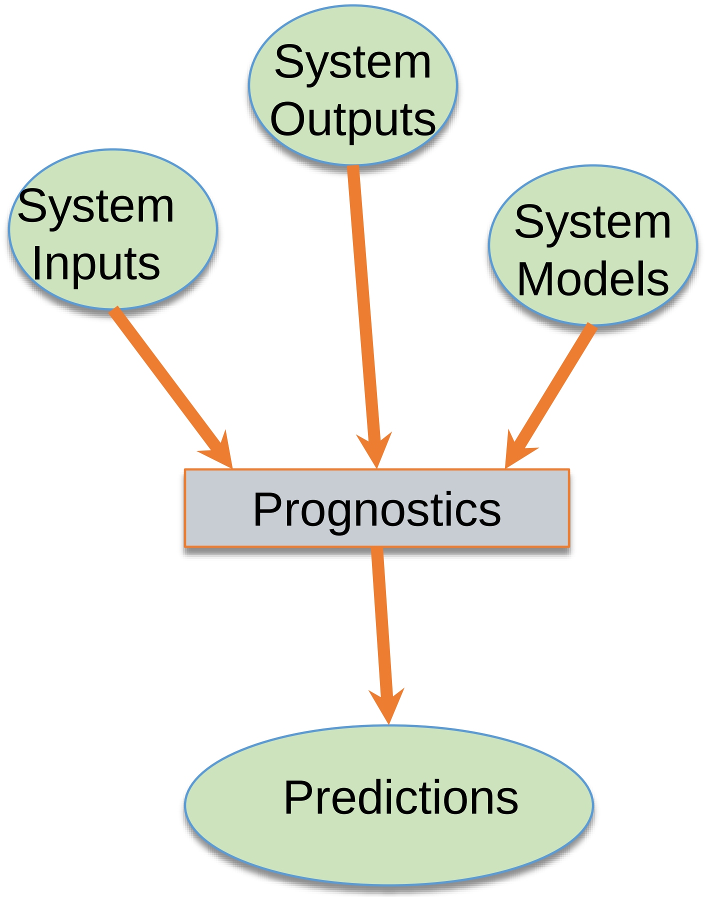
- Prognostics can enable
    - Adopting condition-based maintenance strategies, instead of time-based maintenance
    - Optimally scheduling maintenance
    - Optimally planning for spare components
    - Reconfiguring the system to avoid using the component before it fails
    - Prolonging component life by modifying how the component is used (e.g., load shedding)
    - Optimally plan or re-plan a mission
- System operations can be optimized in a variety of ways

The Basic Idea Revisited



Why Model-Based Prognostics?

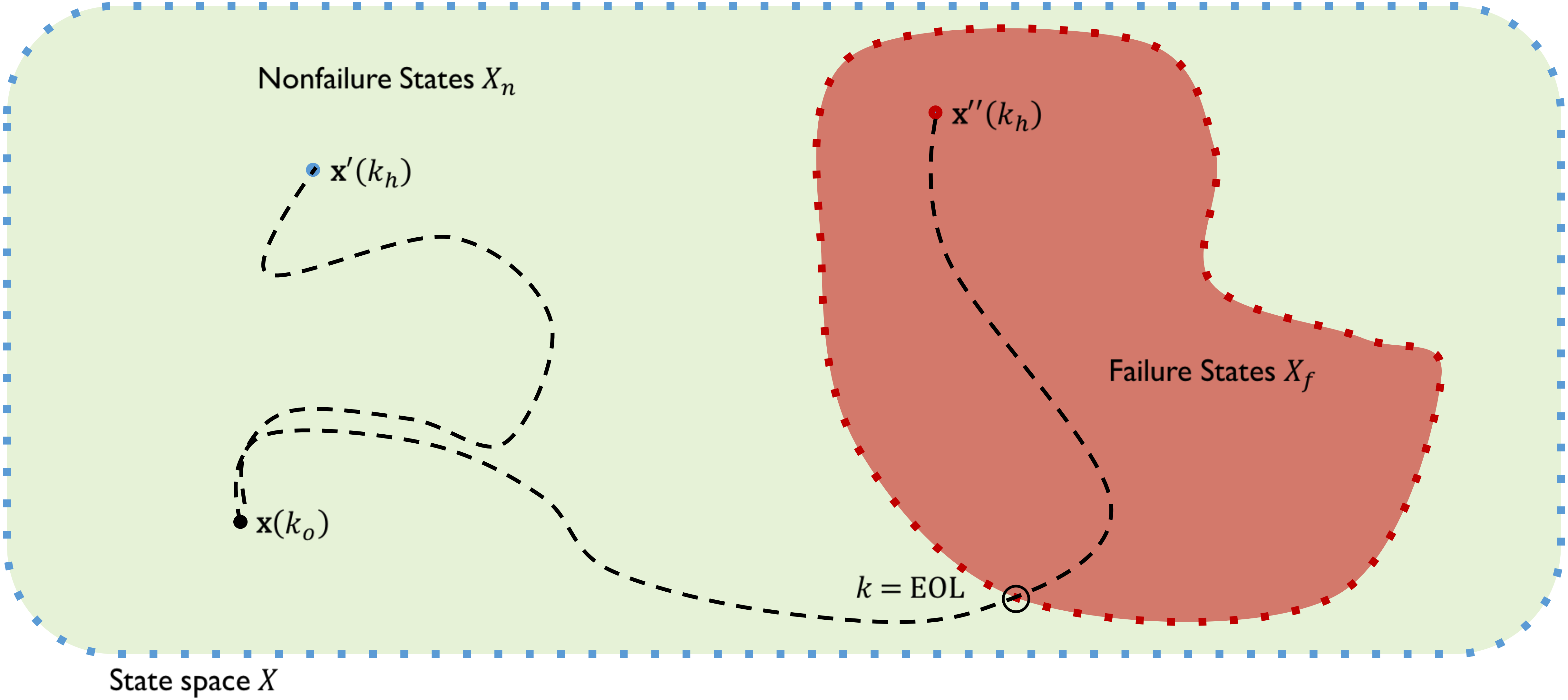
- With model-based algorithms, models are inputs
  - This means that, given a new problem, we use the same general algorithms
  - Only the models should change
- Model-based prognostics approaches are applicable to a large class of systems, given a model
- Approach can be formulated mathematically, clearly and precisely



System Model

- Assume system can be modeled using
  - $\mathrm{x}(k+1) = \mathrm{f}\big(\mathrm{x}(k), \mathrm{u}(k), \mathrm{v}(k)\big)$
  - k is the discrete time variable
  - x is the state vector
  - u is the input vector
  - v is the process noise vector
  - f is the state update equation
- Define a function that partitions state-space into nonfailure and failure states
  - $T_f\colon \mathbb{R}^{n_x} \rightarrow \{true, false\}$
  - That is, $T_f\big(\mathrm{x}(k)\big)$ returns true when it is a failure state, false otherwise

Concept



Nonfailure States X_n
$\mathbf{x}'(k_h)$
$\mathbf{x}''(k_h)$
Failure States X_f
$\mathbf{x}(k_o)$
$k = \mathrm{EOL}$
State space X

Prognostics Model Library

- Available at https://github.com/nasa/PrognosticsModelLibrary
- Implements Model and PrognosticsModel classes
  - Model
    - Encapsulates state, input, and output equations, and model parameters
    - Enforces consistent equation interfaces
    - Input equations are used to define future input trajectories
    - Includes function to simulate a model
  - PrognosticsModel
    - Extends Model class with threshold equation (i.e., T_f)
    - Includes function to simulate a model to the threshold (i.e., to compute EOL)
- Contains example models for batteries (equivalent circuit and electrochemistry models), centrifugal pumps, and pneumatic valves

Prognostics Algorithm Library

- Available at https://github.com/nasa/PrognosticsAlgorithmLibrary
- Provides an Observers package
  - Includes KalmanFilter, ExtendedKalmanFilter, UnscentedKalmanFilter, and ParticleFilter classes
- Provides a Prognosis package
  - Includes a Predictor class
  - Includes a Prognoser class
    - Encapsulates an Observer and a Predictor
  - Relies on PrognosticsModel class
- Includes examples for using the different filters, predictor, and prognoser

Data sets available for download

Available at https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/

Randomized Battery Usage Data Set
Publications using this data set

Description	Batteries are continuously cycled with randomly generated current profiles. Reference charging and discharging cycles are also performed after a fixed interval of randomized usage in order to provide reference benchmarks for battery state of health.
Format	
Datasets	+ Download Randomized Battery Usage Data Set 1 (1285 downloads) + Download Randomized Battery Usage Data Set 2 (936 downloads) + Download Randomized Battery Usage Data Set 3 (906 downloads) + Download Randomized Battery Usage Data Set 4 (4217 downloads) + Download Randomized Battery Usage Data Set 5 (825 downloads) + Download Randomized Battery Usage Data Set 6 (890 downloads) + Download Randomized Battery Usage Data Set 7 (857 downloads)
Dataset Citation	B. Bole, C. Kulkarni, and M. Daigle "Randomized Battery Usage Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA
Publication Citation	B. Bole, C. Kulkarni, and M. Daigle, 'Adaptation of an Electrochemistry-based Li-Ion Battery Model to Account for Deterioration Observed Under Randomized Use', Annual Conference of the Prognostics and Health Management Society, 2014

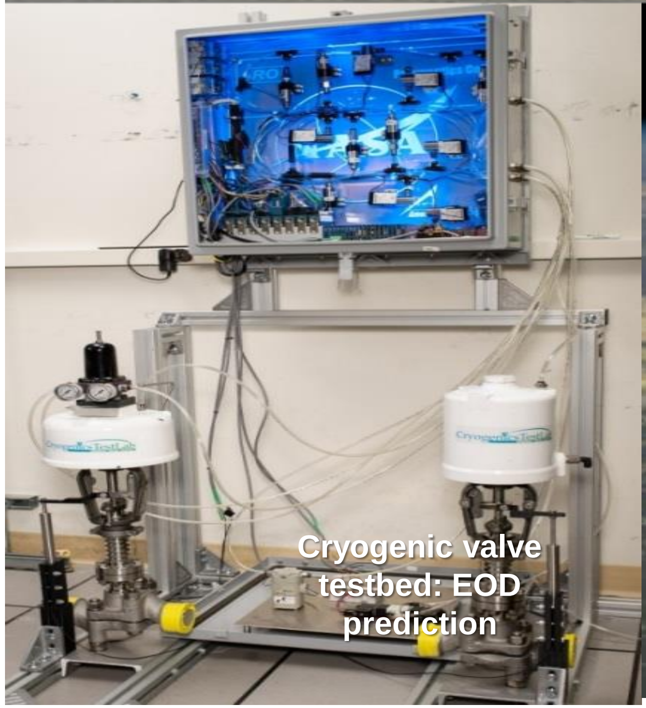
HIRF Battery Data Set
Publications using this data set

Description	Battery Data collected from the Experiments on the Edge 540 Aircraft in HIRF Chamber. Refernce document can be downloaded here
Format	The set is in .mat format and has been zipped.
Datasets	+ Download HIRF Battery Data Set 1 (184 downloads) + Download HIRF Battery Data Set 2 (127 downloads) + Download HIRF Battery Data Set 3 (131 downloads) + Download HIRF Battery Data Set 4 (125 downloads) + Download HIRF Battery Data Set 5 (149 downloads) + Download HIRF Battery Data Set 6 (135 downloads)
Dataset Citation	C. Kulkarni, E. Hogge, C. Quach and K. Goebel "HIRF Battery Data Set", NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA
Publication Citation	Verification of a Remaining Flying Time Prediction System for Small Electric Aircraft. Edward F. Hogge, Brian M. Bole, Sixto L. Vazquez, Jose R., Annual Conference of the Prognostics and Health Management, PHM 2015

Fielded Applications

Edge 540T subscale electric aircraft: EOD, reaming flight time prediction, SOH

Rover testbed: EOD, SOH and remaining driving distance prediction

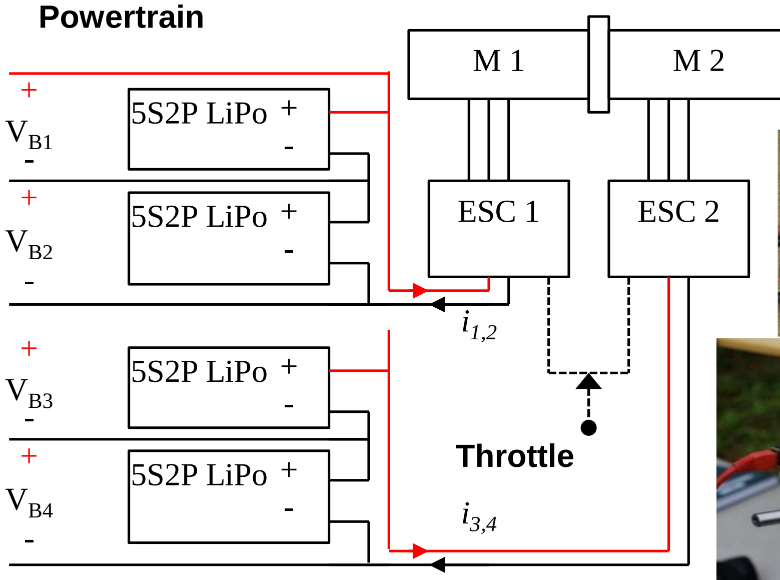


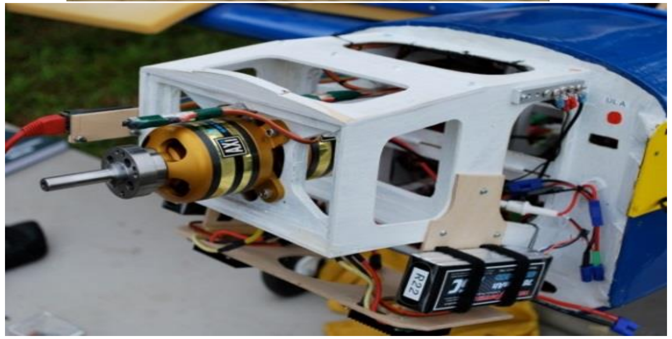
Cryogenic valve testbed: EOD prediction

Orion EFT-1 mission: SOC estimation, EOD prediction, mission success probability computation

Edge 540-T

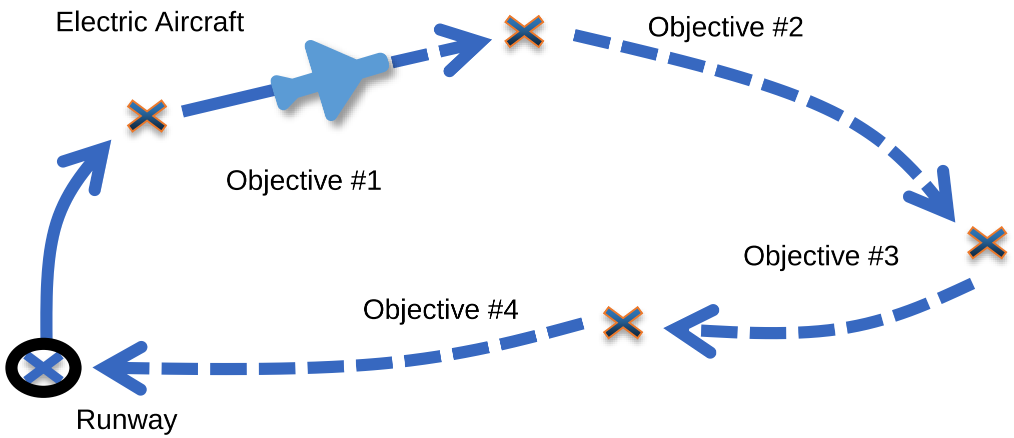
- Subscale electric aircraft operated at NASA Langley Research Center
- Powered by four sets of Li-polymer batteries
- Estimate SOC online and provide EOD and remaining flight time predictions for ground-based pilots





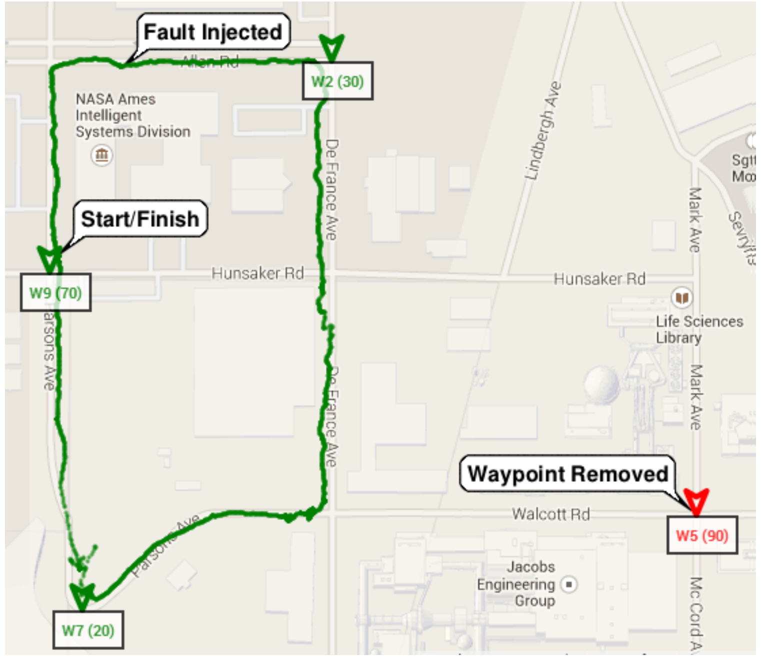
Edge UAV Use Case

- Piloted and autonomous missions, visiting waypoints
- Require 2-minute warning for EOD so pilot/autopilot has sufficient time to land safely
  - This answer depends on battery age
  - Need to track both current level of charge and current battery age
  - Based on current battery state, current battery age, and expected future usage, can predict EOD and correctly issue 2-minute warning



Rover

- Planetary rover testbed at NASA Ames Research Center
  - 24 lithium ion batteries, two parallel sets of 12 in series
  - Batteries power 4 motors, one for each wheel (skid steering)
- Rover operated in two driving modes
  - Unstructured driving
    - Rover is driven freely by an operator, without prior knowledge of actions
  - Structured driving
    - Rover has a given mission, to visit a set of waypoints
    - Rover moves along, visiting waypoints
    - End-of-discharge prediction is required in order to ensure the given set of waypoints can be visited, and if not, to replan the route to optimize mission value

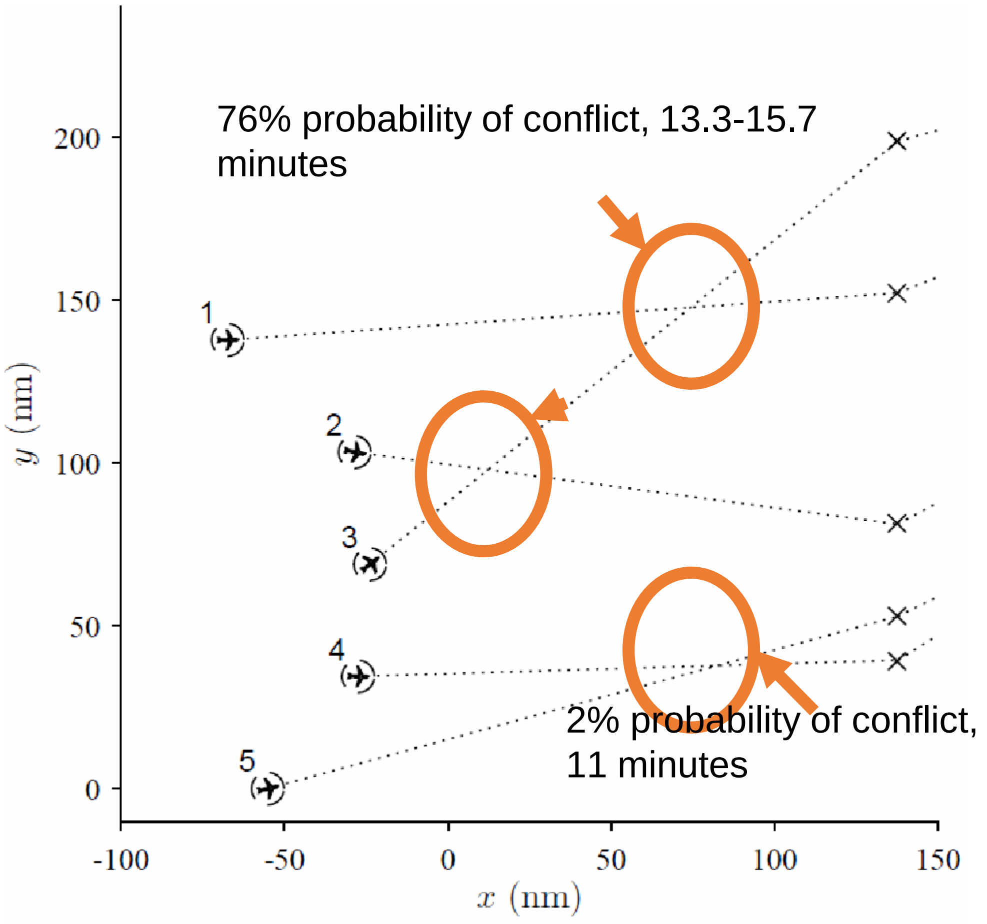


Ref : A. Sweet et al “Demonstration of Prognostics-Enabled Decision Making Algorithms on a Hardware Mobile Robot Test Platform”, PHM 2013

National Airspace Safety

- In the National Airspace System (NAS), can assign labels representing loss of separation (conflicts) between aircraft
  - – Predict time of conflict
  - – Predict probability of conflict
- Can assign labels for other unsafe events
  - – Convective weather encounter
  - – Wake vortex encounter
  - – Low fuel
  - – Etc.

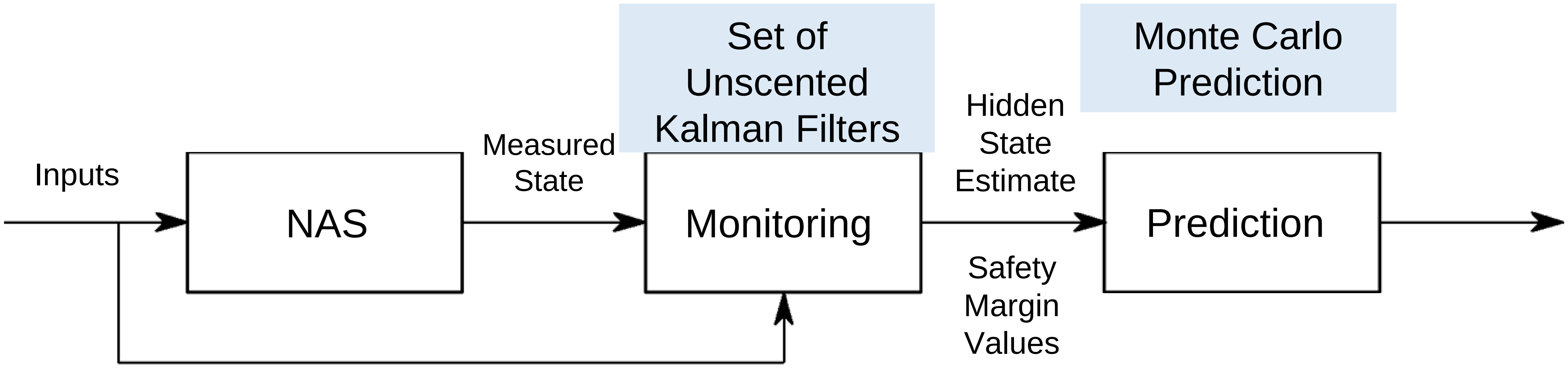
– Provide real time assessment of safety & risk



Safety Modeling

- What categories of events can occur?
  - Loss of separation, wake vortex encounter, convective weather encounter, sector demand violation, etc.
- What conditions define the occurrence of the event?
  - Defined as some function of the NAS state
  - Example 1: Loss of separation between A1 and A2 occurs when the horizontal separation is less than 5 nautical miles and the vertical separation is less than 1000 ft
  - Example 2: Sector demand is too high when the number of aircraft in a sector meets or exceeds the capacity limit
- How do we compute the safety margin w/r/t an event?
  - Margin = {“distance” to event threshold}/threshold and expressed as a percentage
  - Therefore, Margin is 0% when event is present
- How do we compute aggregate safety margins?
  - Example: Average safety margins over all potential events

Computational Architecture



Predicted State

Predicted Safety Margins

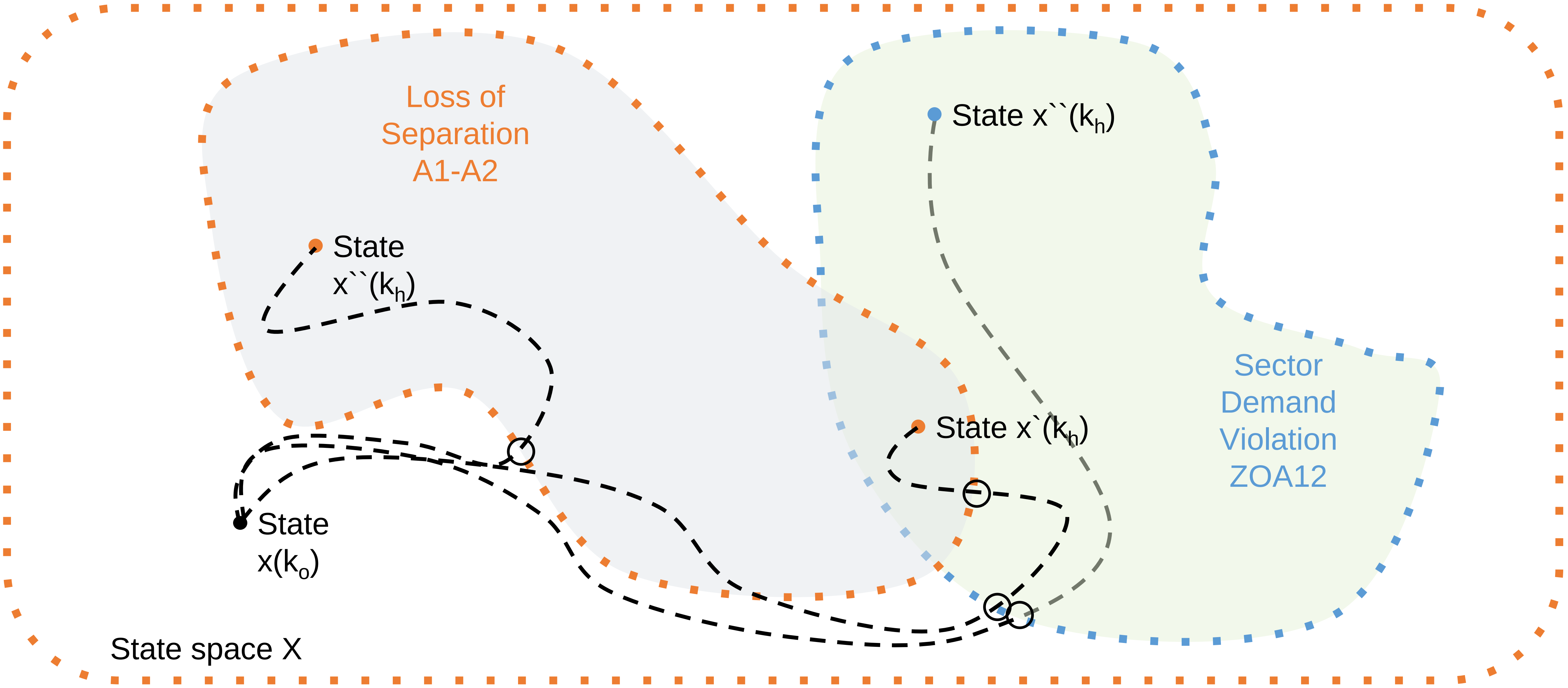
Predicted Times of Occurrence of Unsafe Events

Probability of Future Occurrence of Unsafe Events (in next *x* minutes)

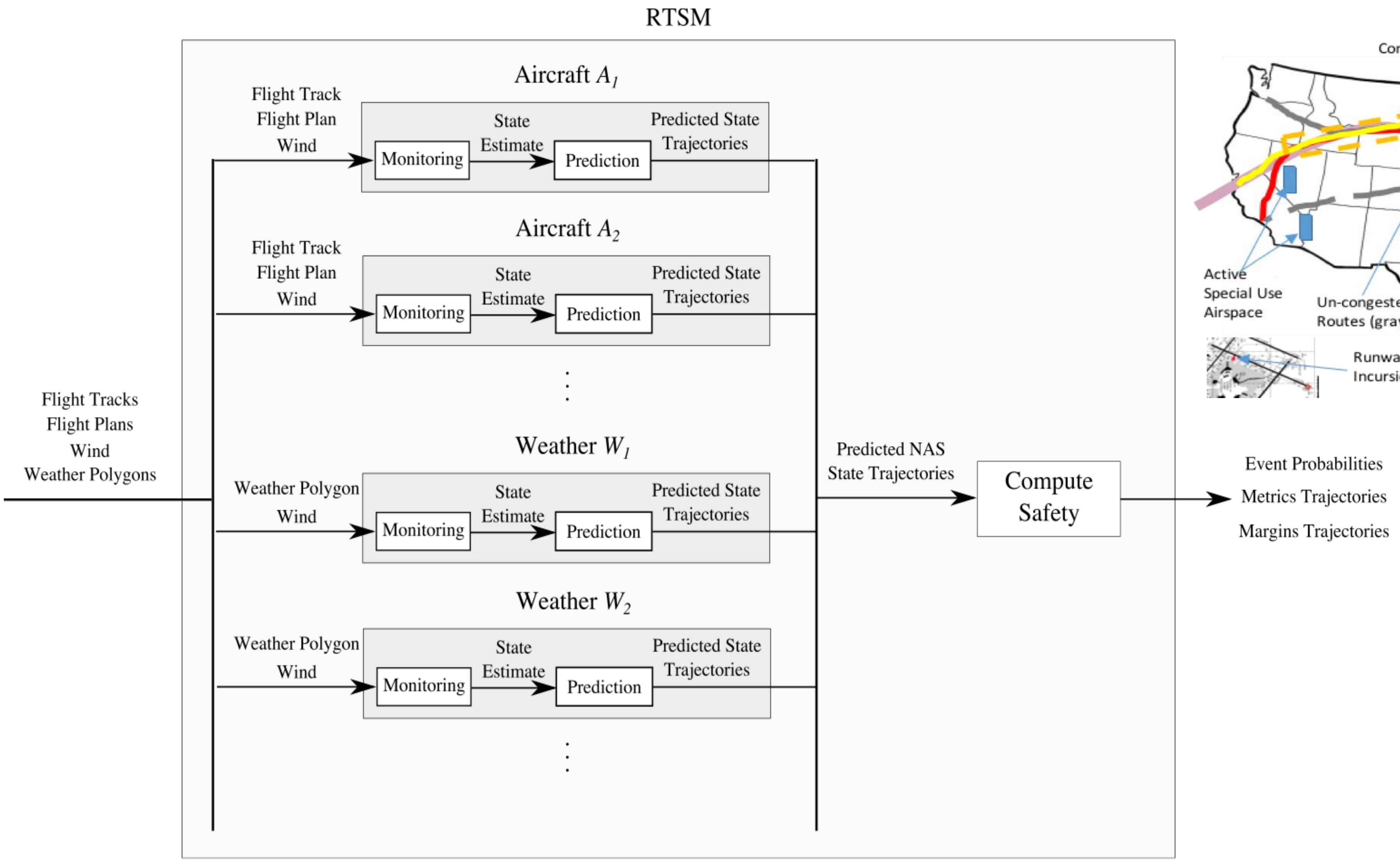
Model-based framework
- First principles models of NAS components (aircraft dynamics, weather, wake vortex, etc.)
- Safety metrics & thresholds

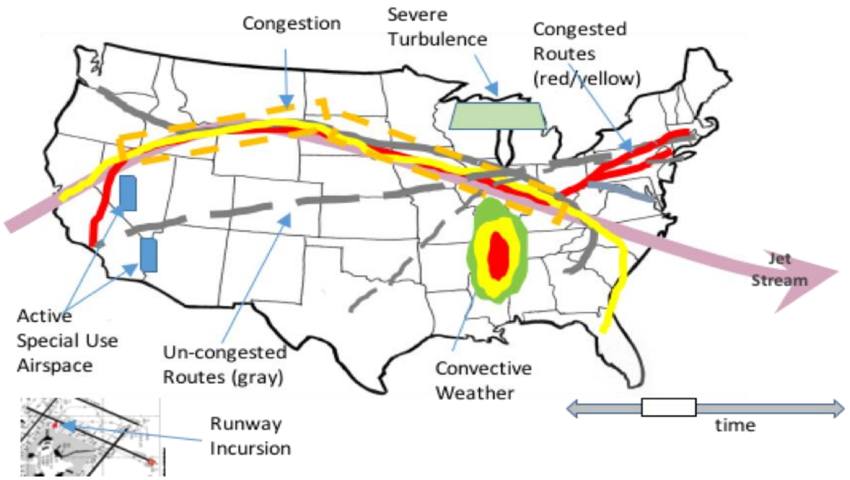
Computation can be distributed to different regions of the NAS and consolidated for system-level safety assessment

Conceptual Framework



Distributed Computational Architecture





Prognostics Summary

- Presented prognostics framework, algorithms, and applications
- Key takeaways:
  - Modeling is key – both dynamics of the system and representation of uncertain inputs to the prediction problem
  - Uncertainty is inherent to the problem and cannot be ignored
  - Future input uncertainty is often most significant and its representation should include as much knowledge about future operation of the system as is known
- Framework and models implemented and available in open-source packages

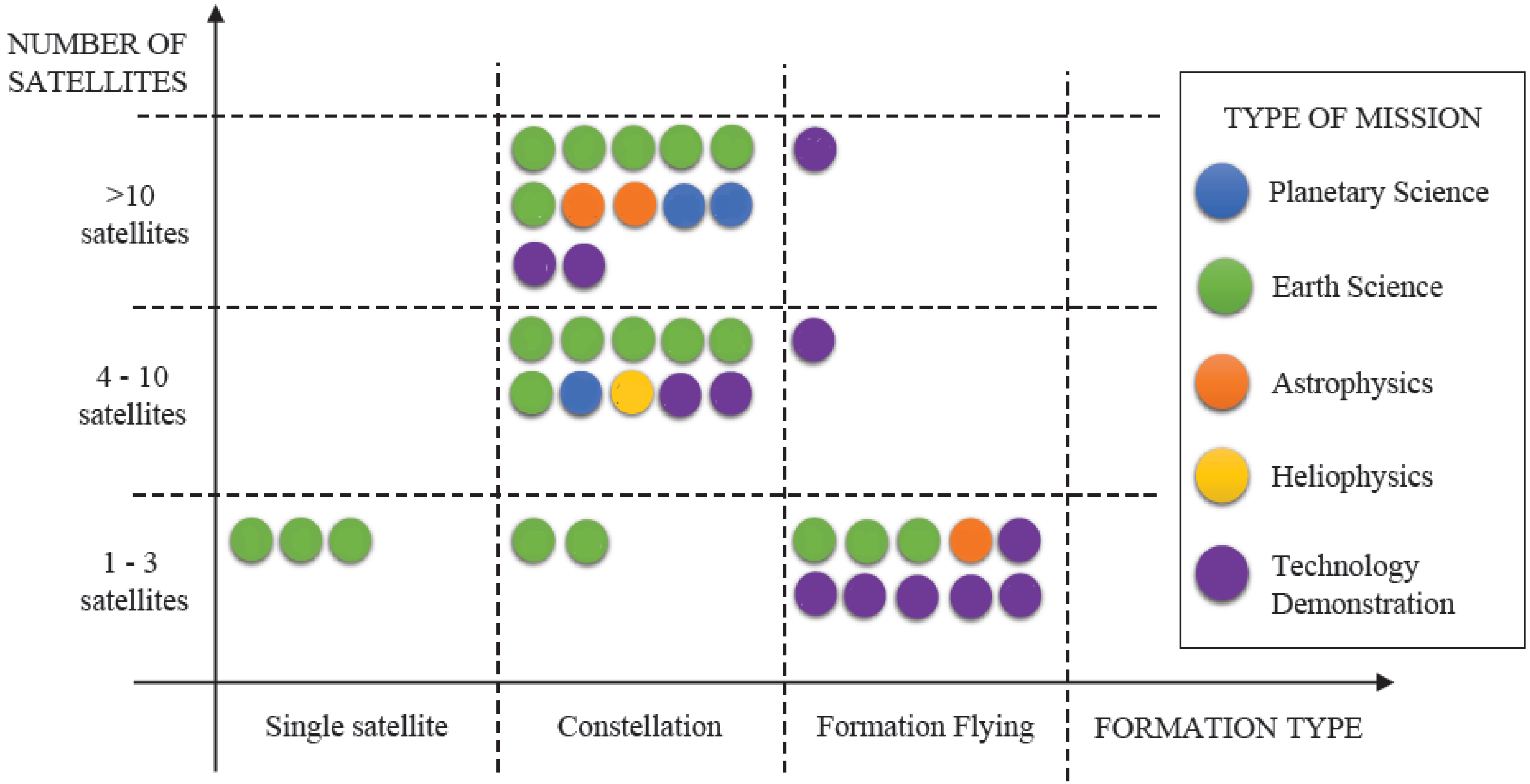
Systems Health Monitoring – Satellite Applications

State of the Art

- Livingstone L1 (Ames/JPL): FDR system on the Deep Space 1 Satellite
  - Onboard planner, multi-threaded executive, & model based FDR system
- Advanced FDIR Study: ESA project to improve on-board satellite failure diagnostics
  - Probabilistic reasoning using Bayesian networks and model-based based diagnostics
- Livingstone L2 (Ames)
  - Longer durations in comparison to predecessor and installed on the Earth Observing One (EO-1) remote sensing satellite. Uses CASPER & SCL for execution of commands
- Smart FDIR (ESA, Alenia Spazio, & Politecnico di Milano)
  - Detects anomalies and identifies faults. Testbeds for evaluation are electrical power subsystem & attitude control subsystem.
- ARPHA Study (ESA Thales/Alenia Italy): Anomaly Resolution & Prognostics Health Management Study for Autonomy.

Satellite Mission Breakdown

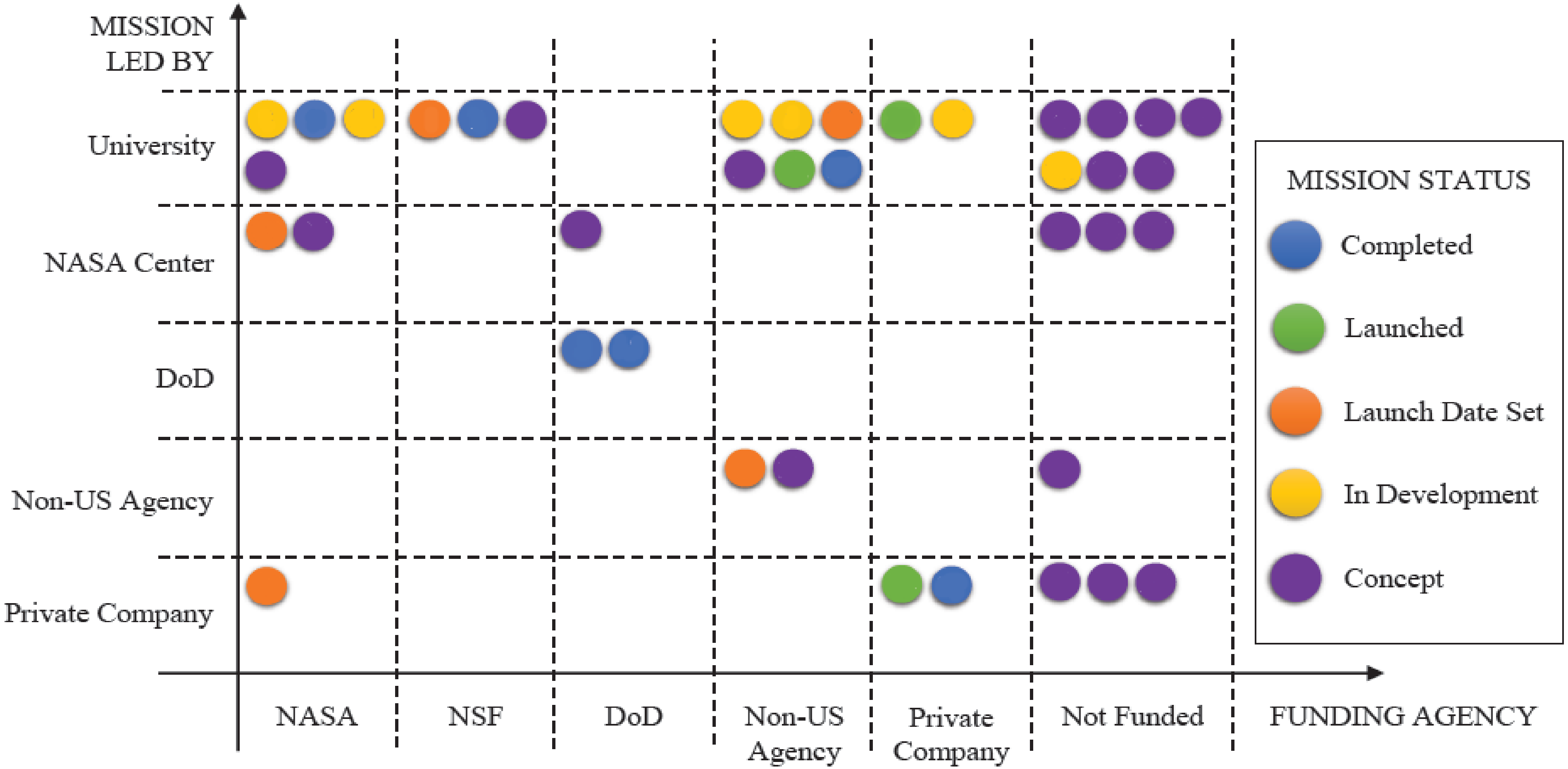
Break-down of thirty-nine multi-satellite missions based on their mission type, formation type and number of satellites



1. Bandyopadhyay, S., Foust, R., Subramanian, G. P., Chung, S.-J., & Hadaegh, F. Y. (2016). Review of formation flying and constellation missions using nanosatellites. *Journal of spacecraft and rockets*.
2. Bandyopadhyay, S., Subramanian, G. P., Foust, R., Morgan, D., Chung, S.-J., & Hadaegh, F. (2015). *A review of impending small satellite formation flying missions.* Paper presented at the 53rd AIAA Aerospace Sciences Meeting.

continued

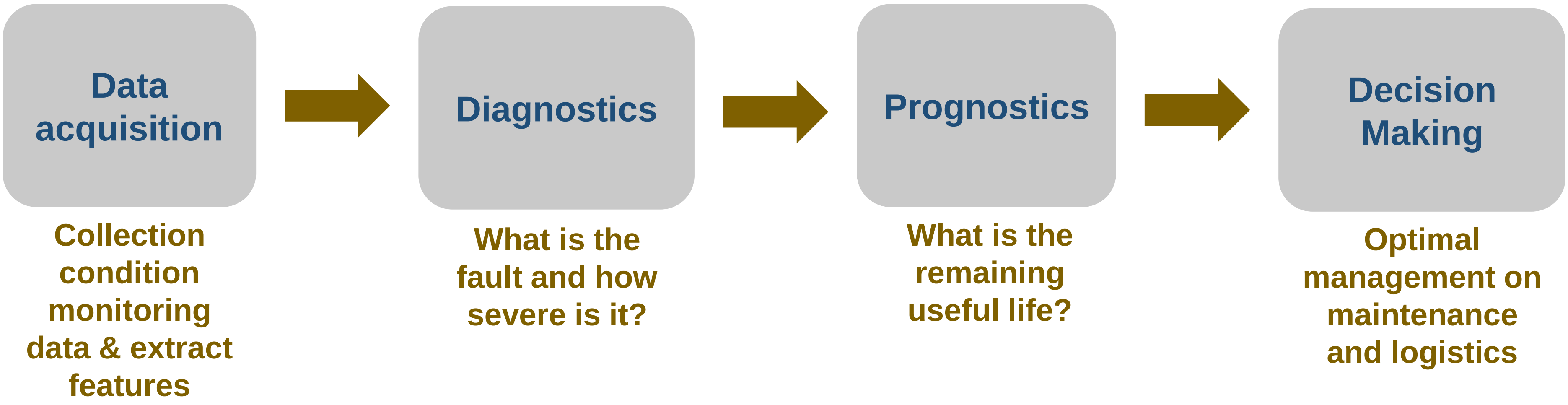
Break-down of thirty-nine multi-satellite missions based on their mission type, formation type and number of satellites



1. Bandyopadhyay, S., Foust, R., Subramanian, G. P., Chung, S.-J., & Hadaegh, F. Y. (2016). Review of formation flying and constellation missions using nanosatellites. *Journal of spacecraft and rockets*.
2. Bandyopadhyay, S., Subramanian, G. P., Foust, R., Morgan, D., Chung, S.-J., & Hadaegh, F. (2015). *A review of impending small satellite formation flying missions.* Paper presented at the 53rd AIAA Aerospace Sciences Meeting.

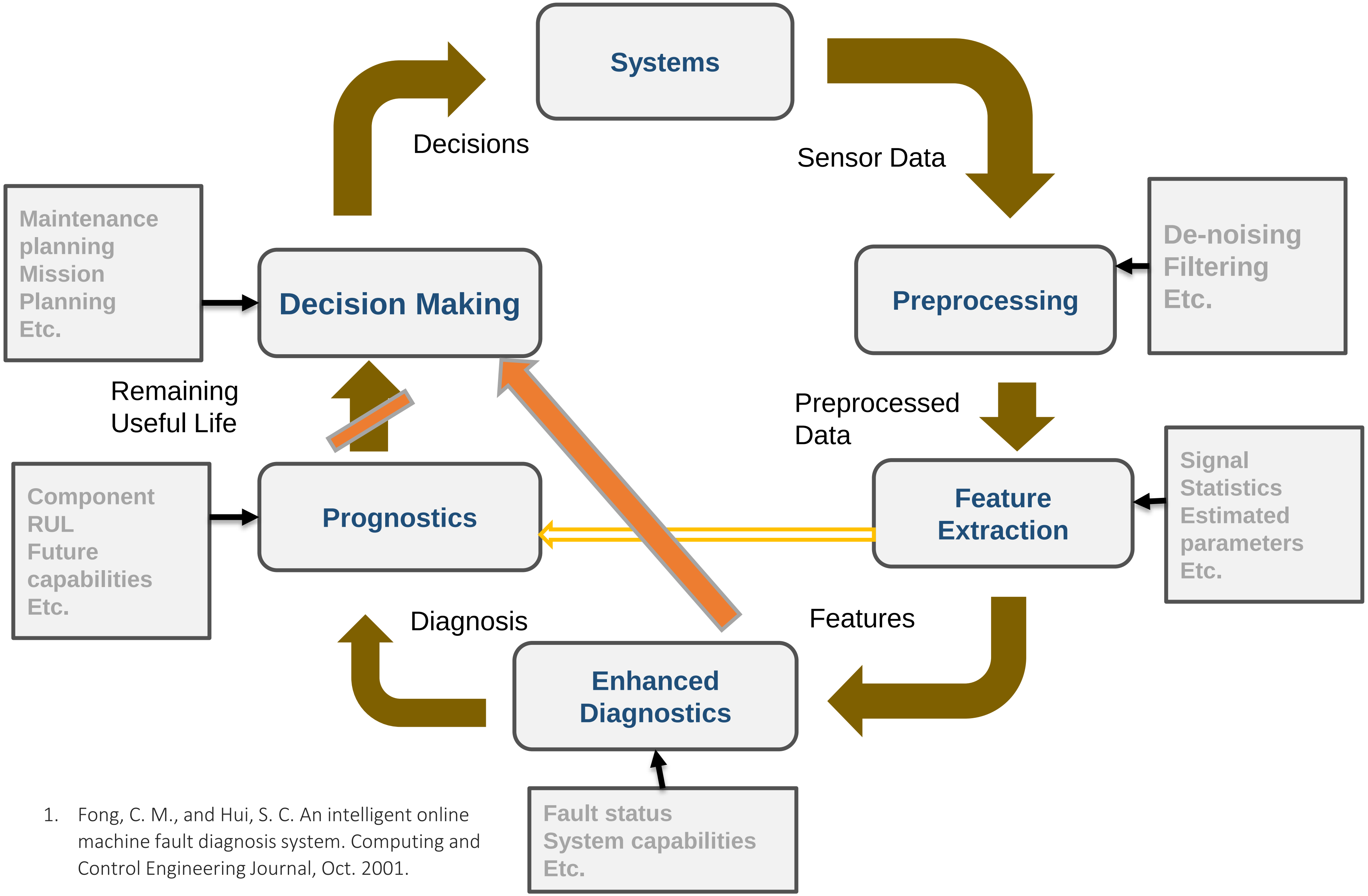
Autonomy Goals for Satellite Missions

- Self-requirements
  - self-trajectory
  - self-protection
  - self-scheduling
  - self-reparation
- Knowledge
- Awareness
- Monitoring
- Adaptability
- Dynamicity
- Robustness
- Resilience
- Mobility



1. Vassev, E., & Hinchey, M. (2014). *Autonomy Requirements Engineering for Space Missions*: Springer.

Satellite and Systems Health Monitoring

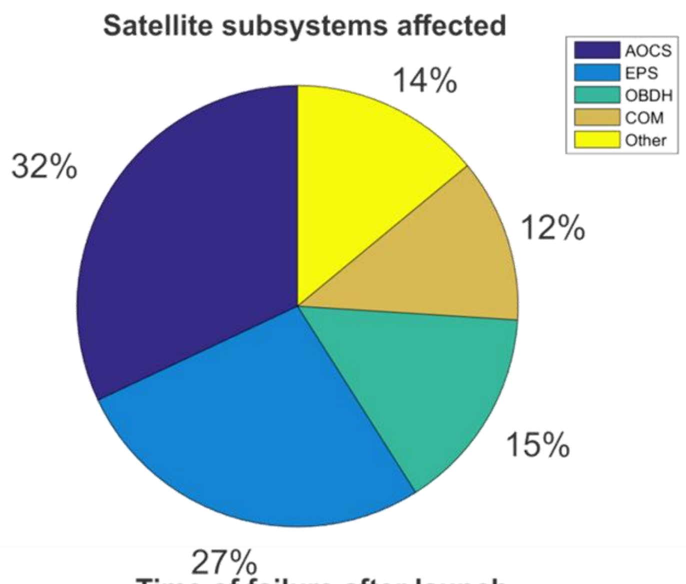


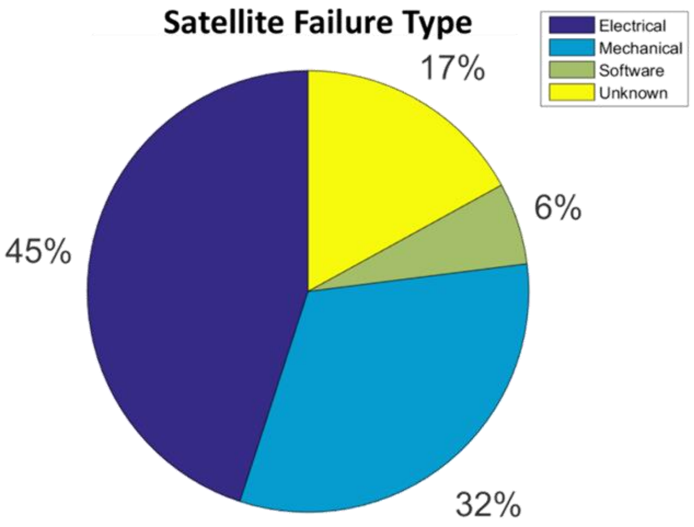
1. Fong, C. M., and Hui, S. C. An intelligent online machine fault diagnosis system. Computing and Control Engineering Journal, Oct. 2001.

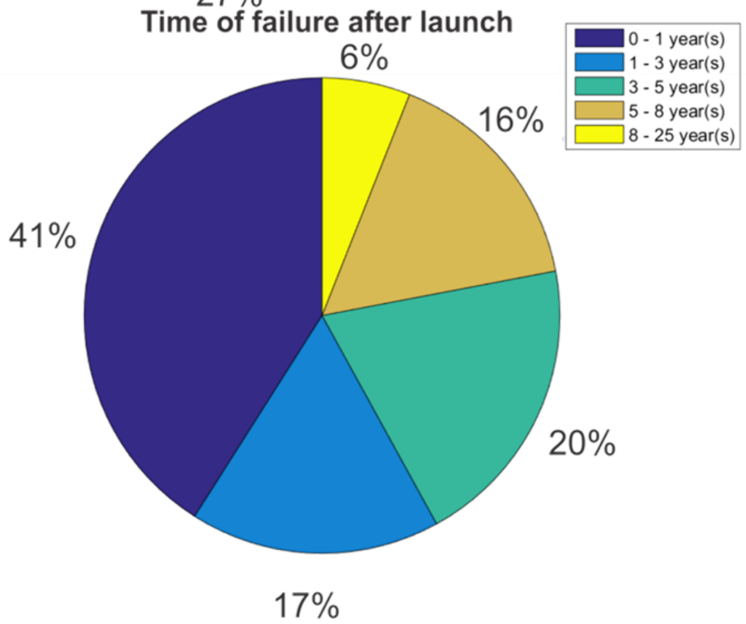
Satellite Subsystems

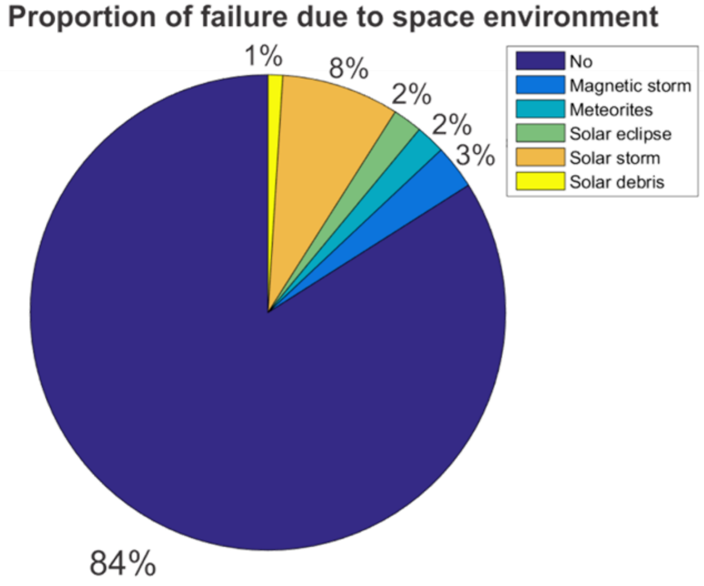
- On-board Data Handling System (OBDH)
- Power System (EPS)
- Communication System - Inter-Satellite Link & Data Downlink and/or Uplink (COM)
- Thermal Control System (TCS)
- Structure (MECH)
- Attitude and Orbit Control System (AOCS)
- Payload (PL)

Satellite Subsystems Anomalies and Failures



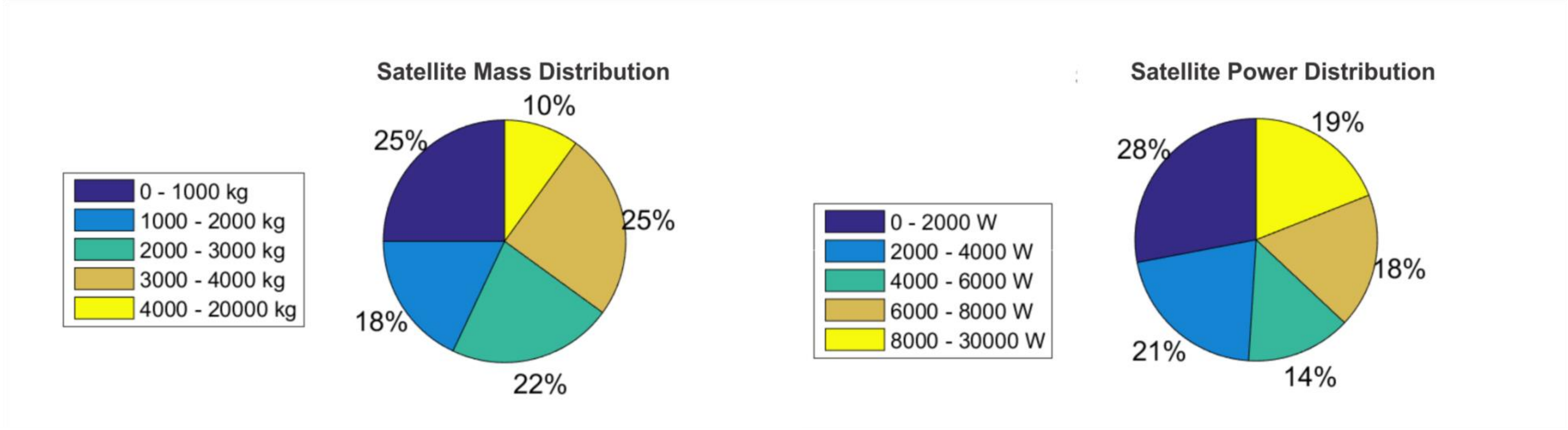


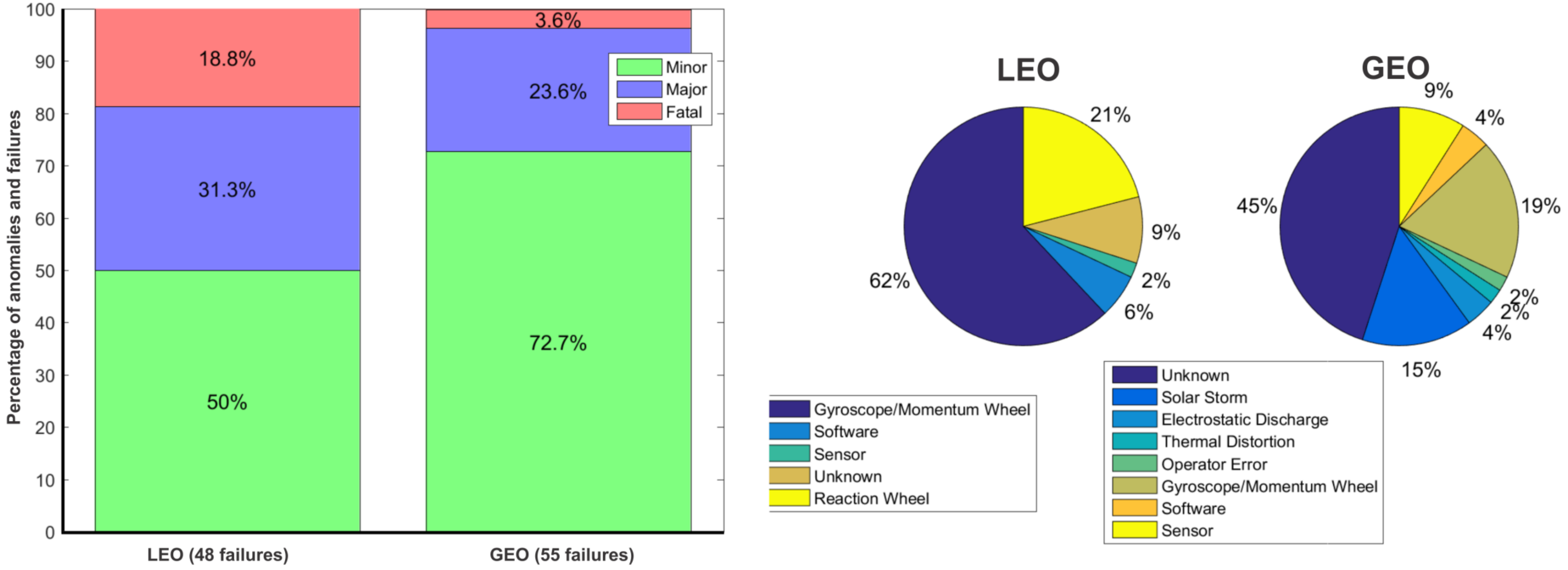




1. Tafazoli, M. (2009). A study of on-orbit spacecraft failures. *Acta Astronautica, 64*(2), 195-205. .

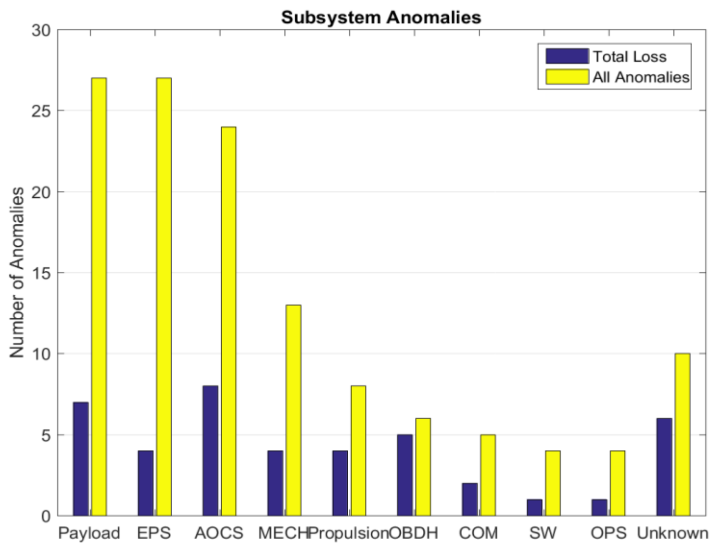
continued

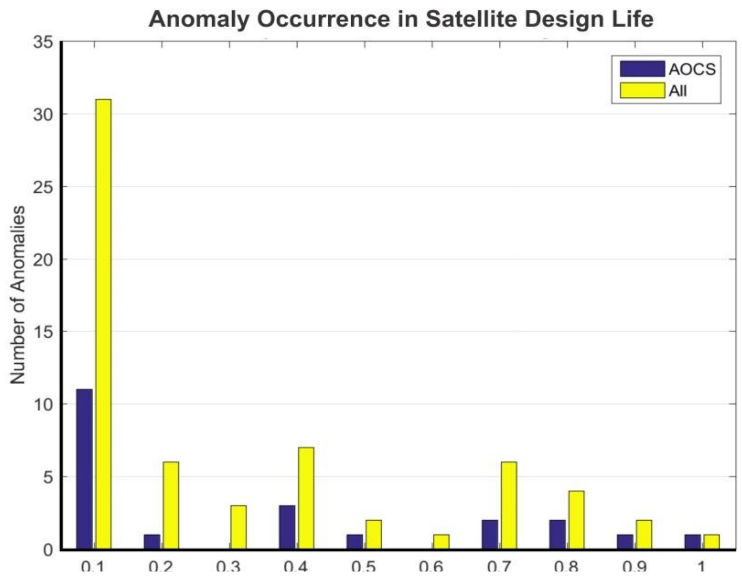


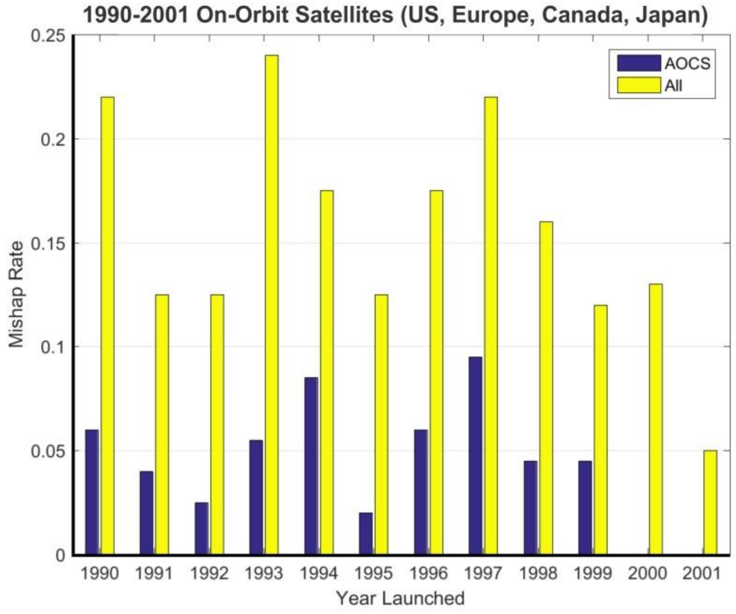


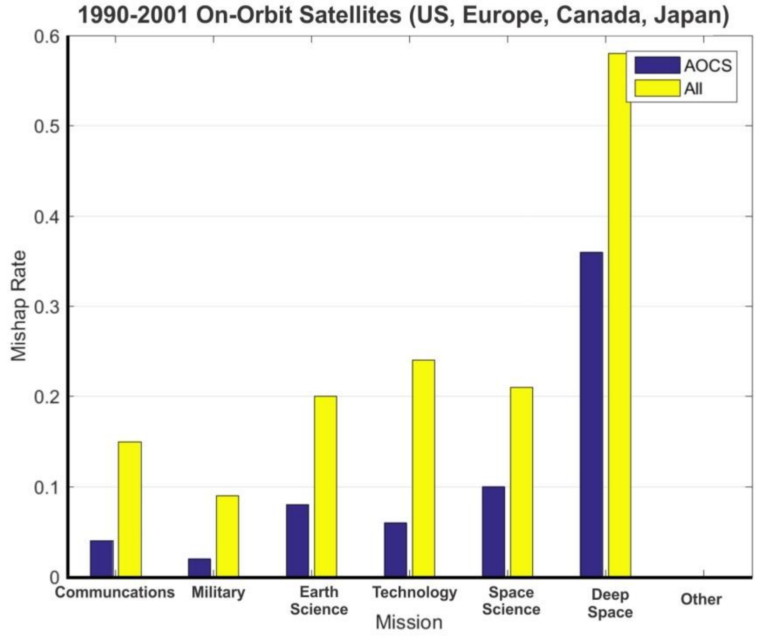
1. Wayer, J. K., Castet, J. F., & Saleh, J. H. (2013). Spacecraft attitude control subsystem: Reliability, multi-state analyses, and comparative failure behavior in LEO and GEO. *Acta Astronautica*, *85*, 83-92.

continued





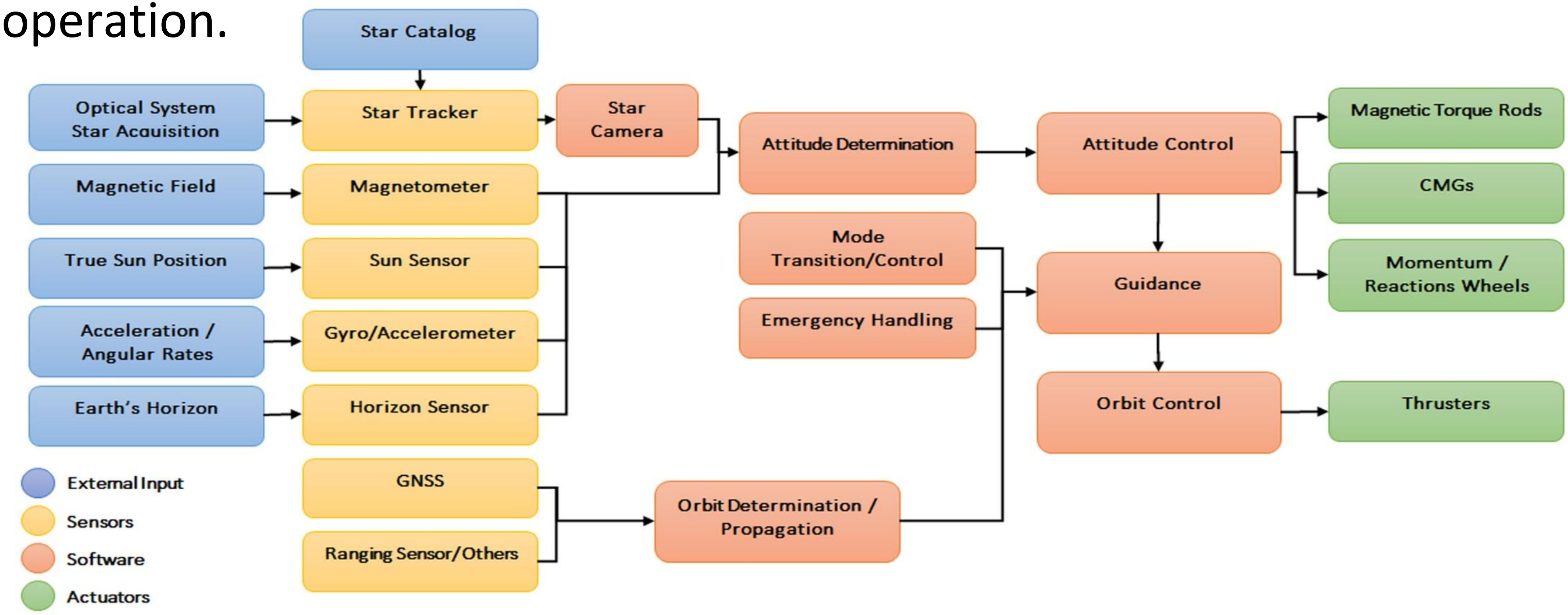


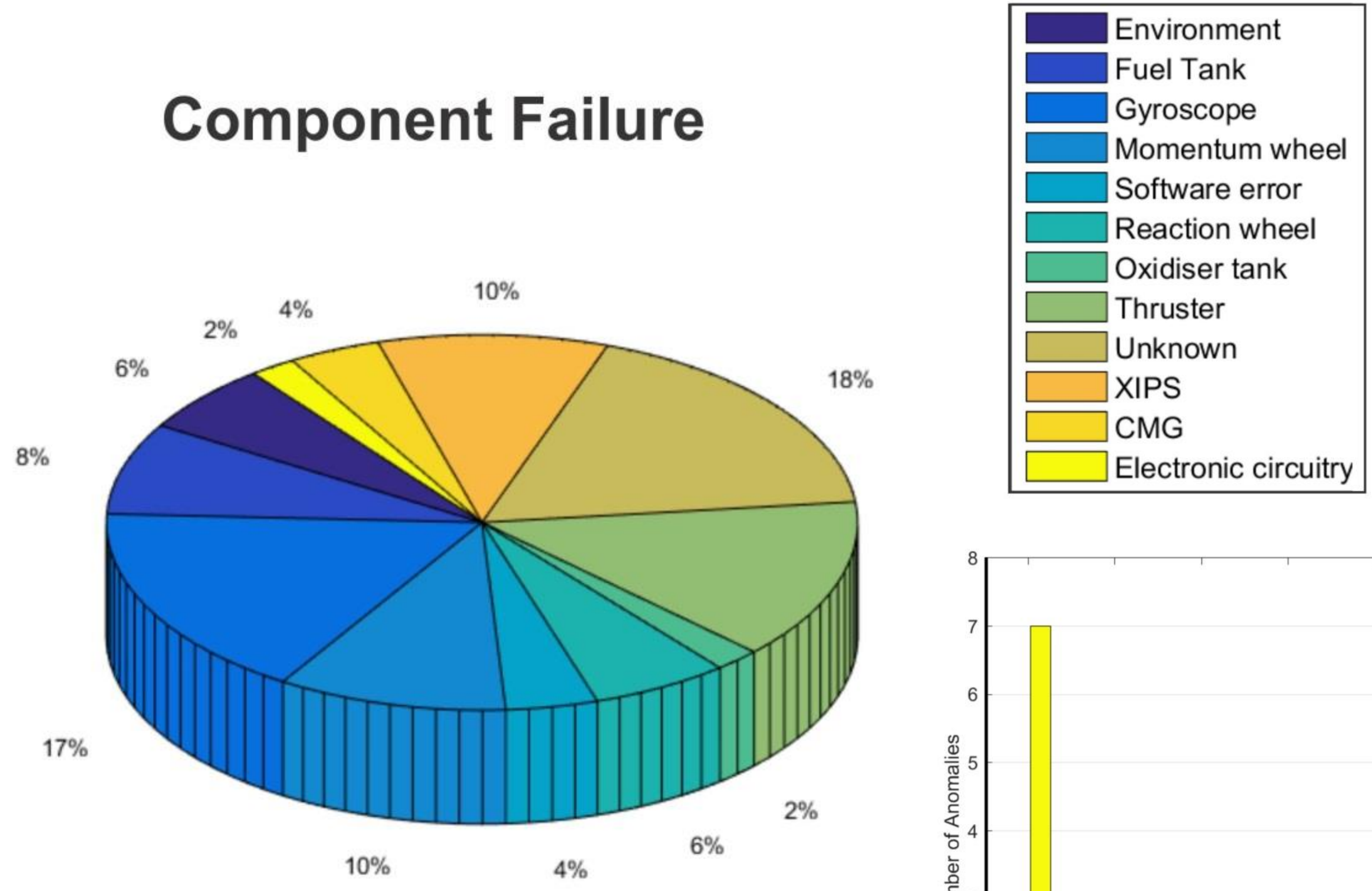


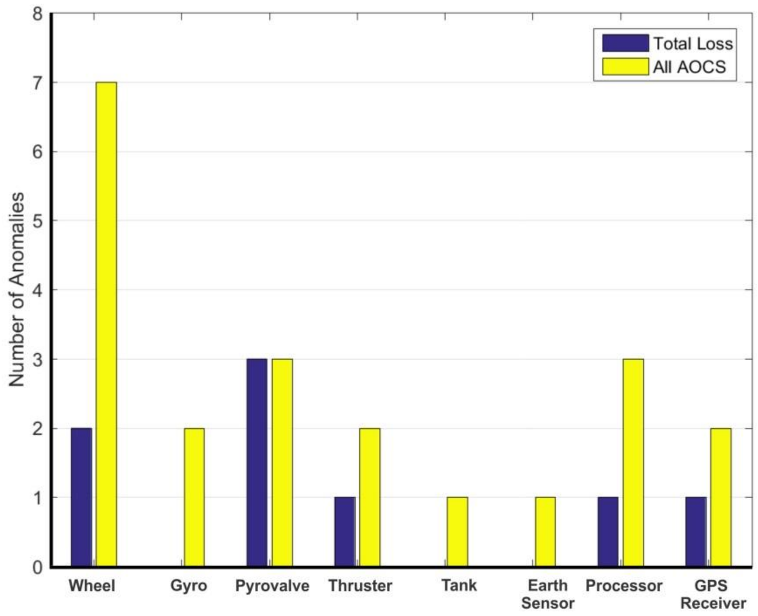
1. Robertson, B., & Stoneking, E. (2003). Satellite GN & C anomaly trends. *Advances in the Astronautical Sciences, 113*, 531-542.

Attitude and Orbit Control System

- Attitude and Orbit Control System, Attitude Determination System (ADS), or Attitude Determination and Control Subsystem (ADCS).
- Used to stabilize and orient spacecraft in desired directions during operation.



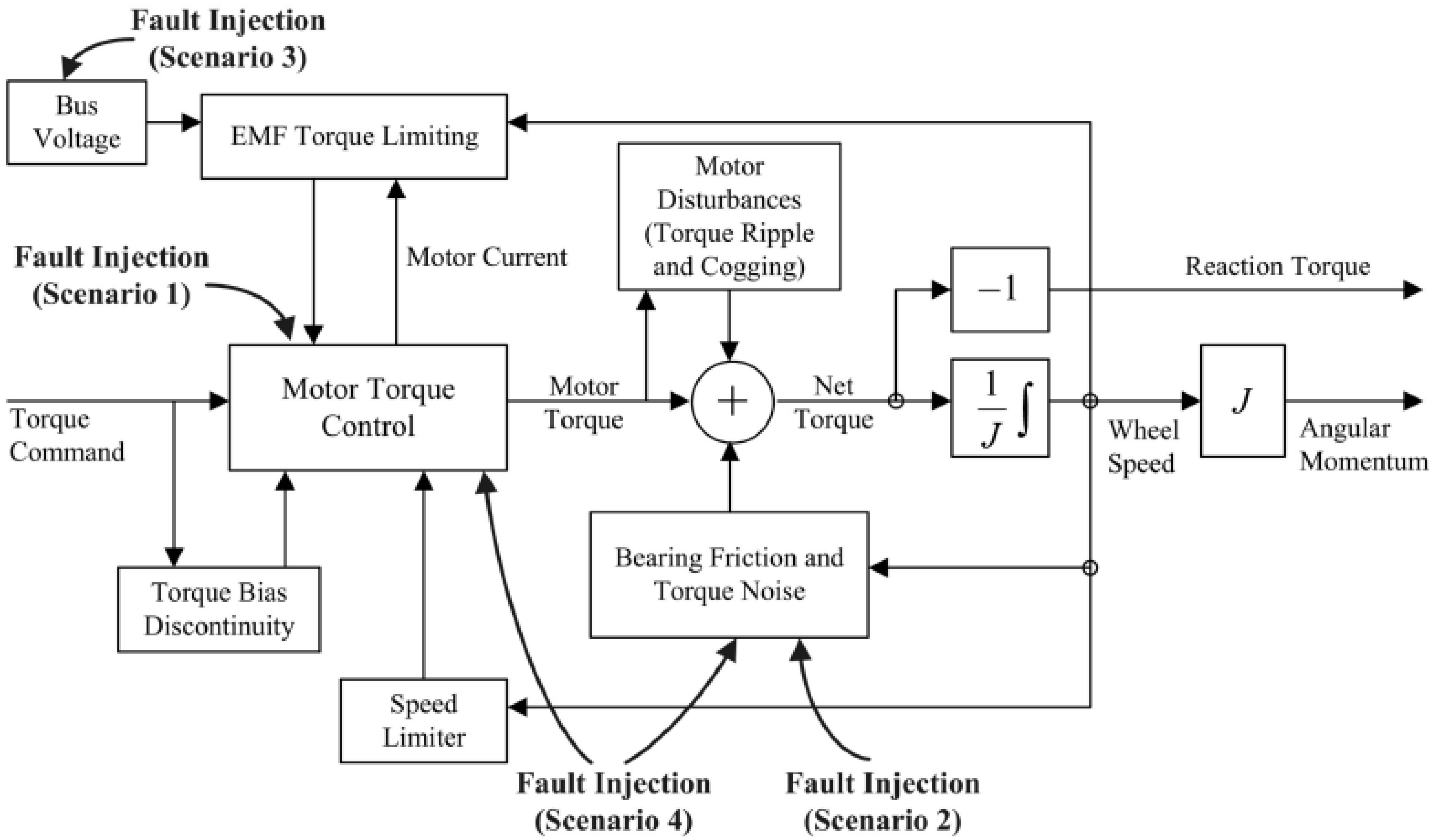




1. Tafazoli, M. (2009). A study of on-orbit spacecraft failures. *Acta Astronautica, 6*
2. Robertson, B., & Stoneking, E. (2003). Satellite GN & C anomaly trends. *Advances in the Astronautical Sciences, 113*, 531-542.

Basic SHM Idea for AOCS Components

- Data transmission for deep space missions is limited
- Algorithms to diagnose faults and possibly predict the remaining useful life of AOCS components.
- For a single satellite, SHM is useful/important for mission success
- For a satellite formation, proper identification of faults coupled with an estimate of RUL can enable
  - Mission success
  - Mission reconfiguration/re-planning within RUL window
- AOCS contributes to about 32% of all on-board failures
- 84% of all AOCS anomalies and failures are related to design & operations
- SHM will enable robust and resilient design



Fault Injection
(Scenario 3)
Bus Voltage
EMF Torque Limiting
Motor Current
Fault Injection
(Scenario 1)
Motor Disturbances (Torque Ripple and Cogging)
−1
Reaction Torque
Torque Command
Motor Torque Control
Motor Torque
+
Net Torque
$\frac{1}{J}\int$
Wheel Speed
J
Angular Momentum
Torque Bias Discontinuity
Bearing Friction and Torque Noise
Speed Limiter
Fault Injection
(Scenario 4)
Fault Injection
(Scenario 2)

Implementation

Determine component for study

Obtain modeling sources

Model nominal and faulty operation

Component Determination

Actuators

- Magnetorquers
- Reaction Wheels
- Momentum Wheels
- Control Momentum Gyros (CMGs)
- Thrusters

Pointing accuracy

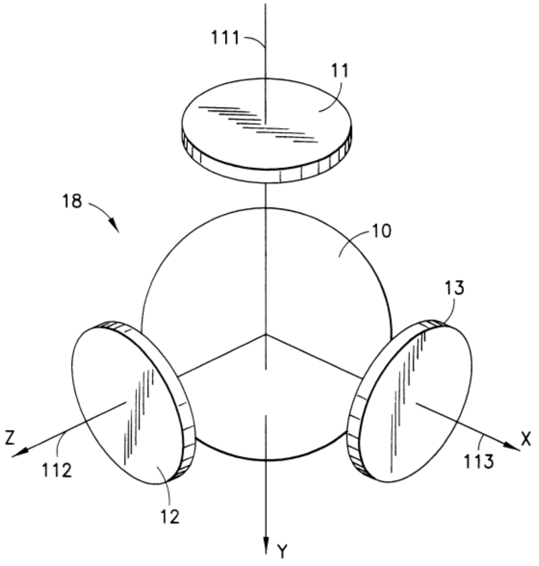
Low energy consumption

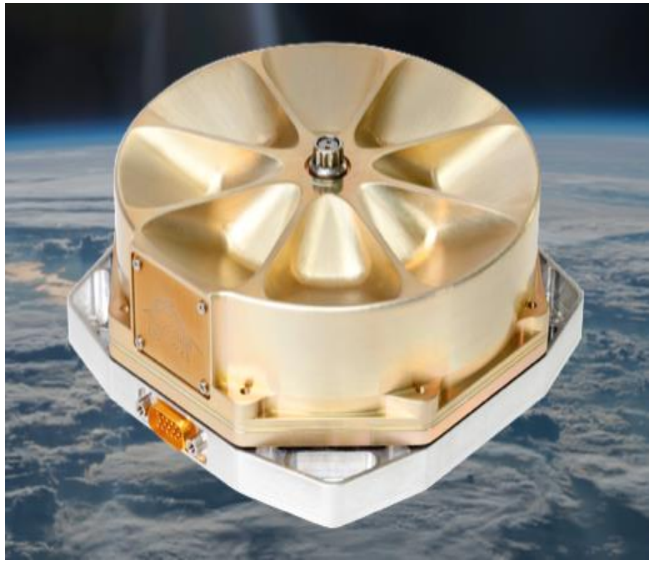
Known cause of mission decay and failure

- Relevance: is it suitable for future satellite missions?
- History: does the component have records of failure?
- Benefits: what are the pros of using this actuator?

Reaction Wheels

- The Reaction Wheel Assembly (RWA) is an actuator that consists of a flywheel attached to a brushless DC motor.
- They produce a torque that is applied to the spacecraft to correct its position.
- A minimum of three reaction wheels, one per body axis, is required to maintain attitude
- Reaction wheels are used for zero-momentum control or momentum-bias control on ADCS, the two forms of three-axis control



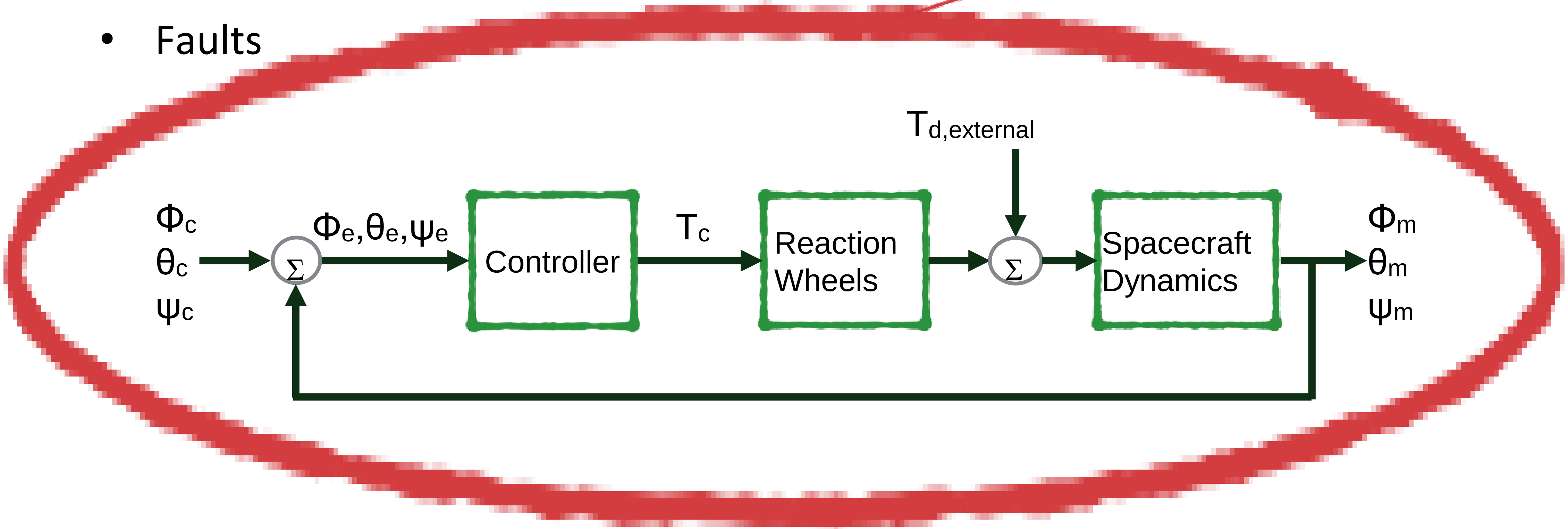


Source: Blue Canyon Technologies

Modeling Sources

- What needs to be determined?
  - Parameters
  - Models
  - Faults

Schematic of an AOCS actuated by RWA



Equations

- Mechanical:

Motor torque

Friction torque

$$\boxed{T_m} - \boxed{T_f} + \boxed{T_d} = J\dot{\omega}$$

Disturbance torque

- Electrical:

Resistance Voltage

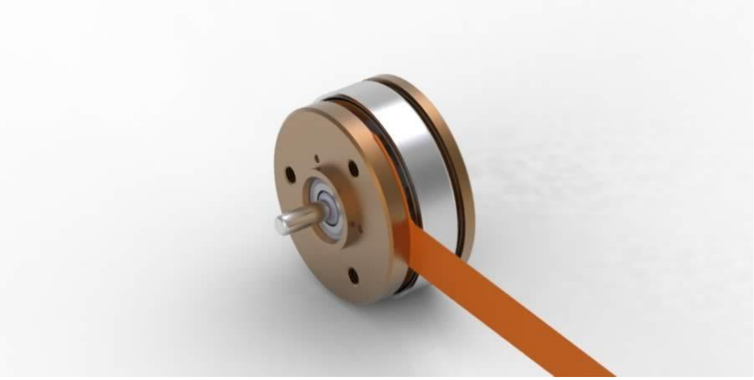
$$\boxed{V_R} = \boxed{V_S} - \boxed{V_{emf}}$$

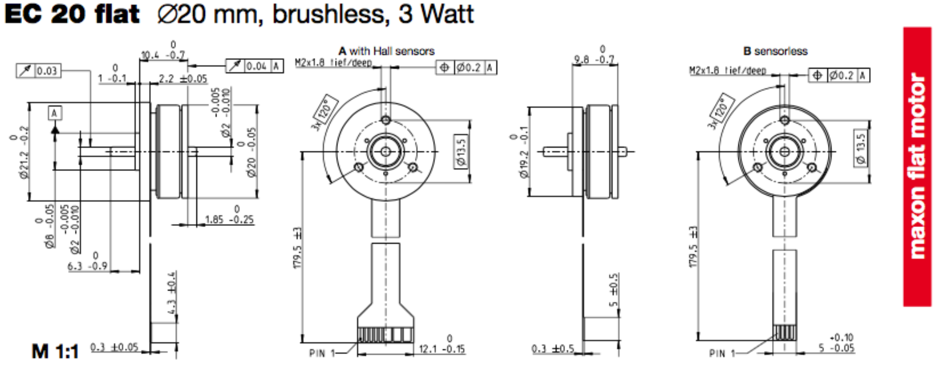
Commanded Voltage

Back EMF Voltage

$$V_R = k\int(i_c - i_m) - k_{emf}\omega$$

Specifications (Maxon Motor)





Parameter	Notation	Value
Torque constant	k_m	5.88 mNm/A
Back EMF constant	k_{emf}	5.89 mNm/A
Inertia of rotor	J	1.12x10^-6 kg-m^2
Resistance	R	6.67 Ω
Number of poles	N	8
Viscous friction coefficient	b	5.1965x10^{-7} Nms
Static Imbalance	s	1.2 g-mm
Dynamic Imbalance	d	20 g-mm^2
Gain	k	220 V/A*s

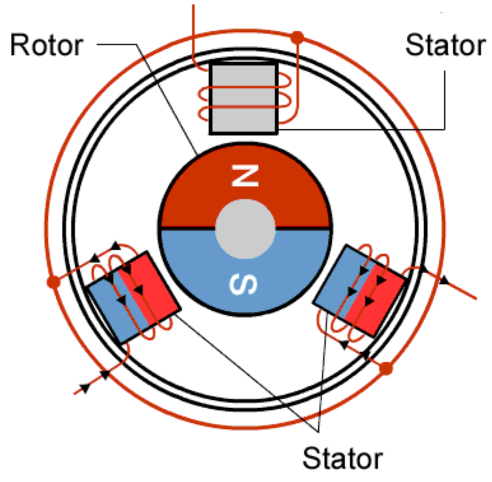
Source: Maxon Motor

Motor disturbances

- **Torque Ripple**: result of the drive torque being a superposition of rectified sine waves. The torque ripple of a motor with a greater number of poles is at a higher frequency, where it is less problematic.

$$f_2(\omega) = B\sin(3Nt\omega)$$

(f_2): where B is the amplitude of the cogging torque, N is the number of poles, and ω is the angular speed



- **Cogging torque**: result of the magnets in the rotor moving past a ferromagnetic stator. Present regardless of whether a torque is applied. Absent from RWA that have no ferromagnetic materials in the stator.

$$f_1(\omega) = C\sin\left(\frac{Nt}{2}\omega\right)$$

(f_1): where C is the amplitude of the cogging torque, N is the number of poles, and ω is the angular speed

Markley, F. L. & Crassidis, J.L. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*, 195-205. Springer.
Khorasani, K, & Sobhani-Tehrani, E. (2009). *Fault Diagnosis of Nonlinear Systems Using a Hybrid Approach.* Springer.

Flywheel Imbalances

- **Static imbalance**: condition that the wheel's center of mass is not on the axis of rotation.

$$F_c = ma_c = m\frac{v^2}{r} = mr\Omega^2,$$

- **Dynamic imbalance**: condition that the axis of rotation of the wheel is not on the principal axis.

$$\tau = mrd\Omega^2.$$

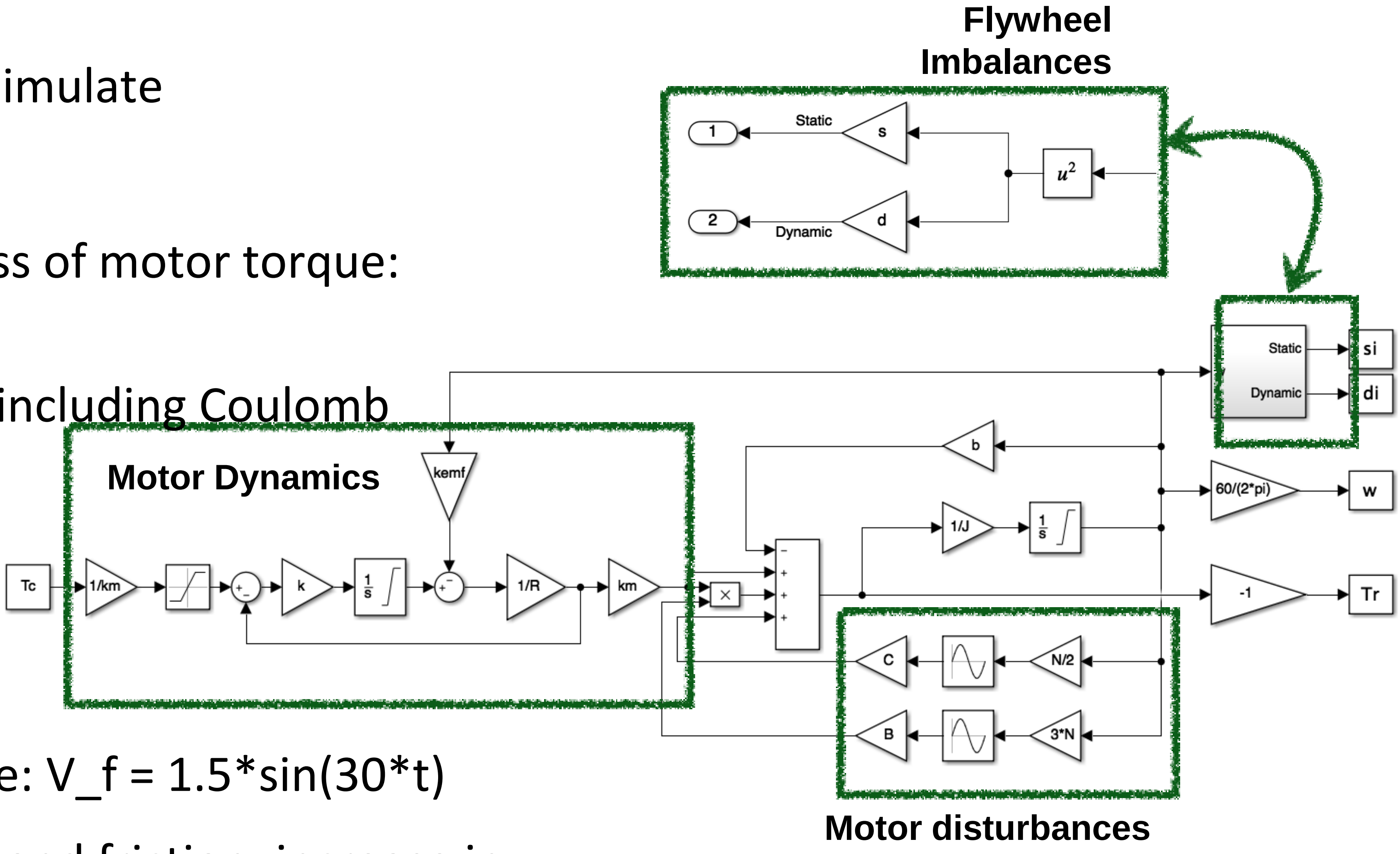
Markley, F. L. & Crassidis, J.L. (2014). *Fundamentals of Spacecraft Attitude Determination and Control*, 195-205. Springer.

Shields, J et. al (2017). Characterization of CubeSat Reaction Wheel Assemblies. *Journal of Small Satellites, 6*(1), 565-580.

Simulation Model & Scenarios

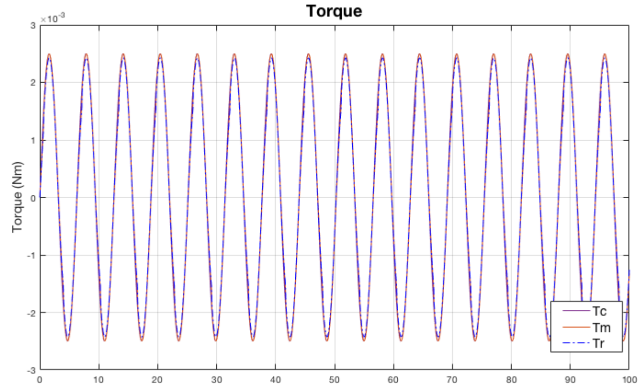
Using different inputs, simulate

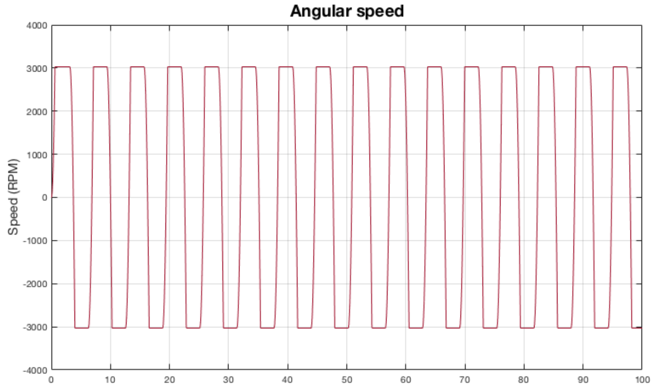
1. Nominal operation
2. Loss of effectiveness of motor torque: $k_m^{f1}=0.7k_m^n$
3. Change in friction: including Coulomb friction, c=0.00103
4. Ripple and cogging

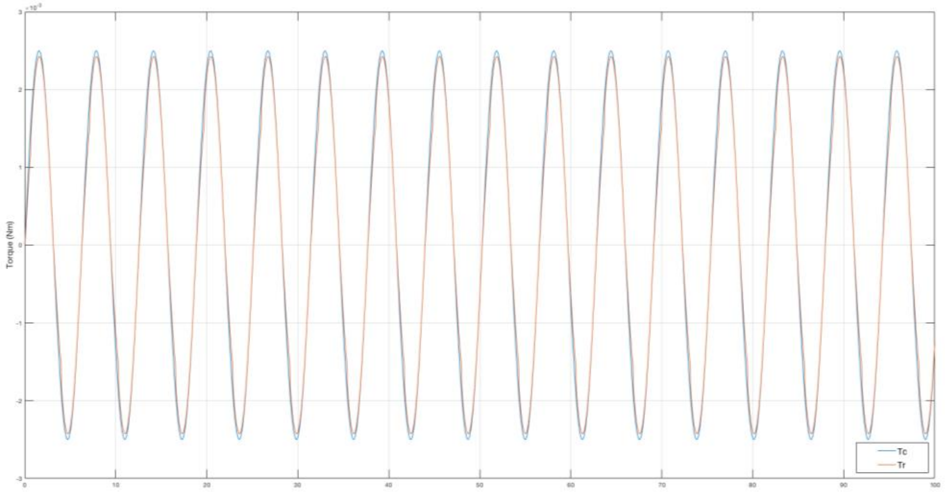


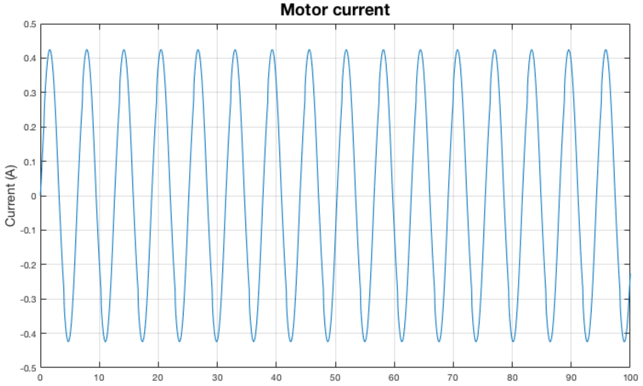
1. Voltage disturbance: V_f = 1.5*sin(30*t)
2. Increase in current and friction: increase in friction and $k_m^{f4}=3k_m^n$

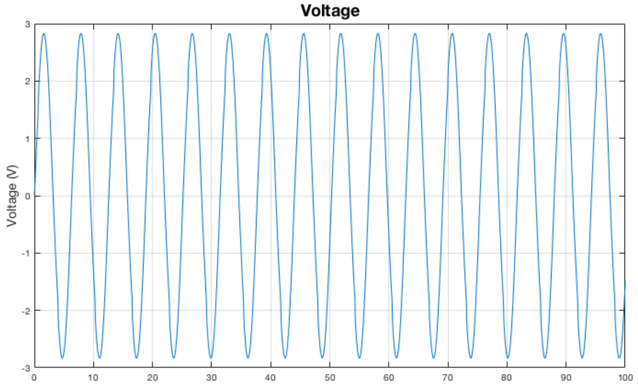
Nominal Operation



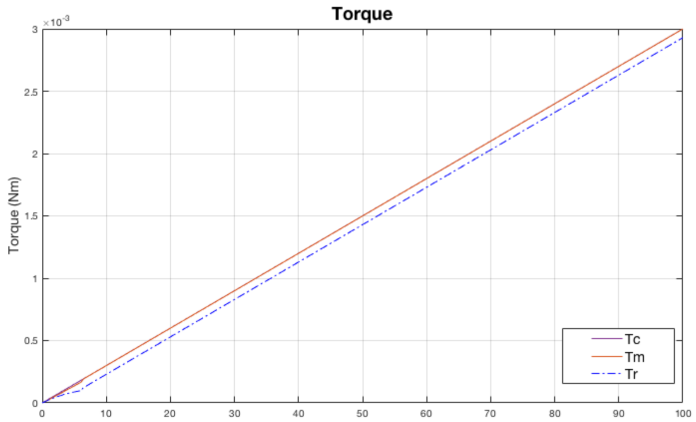


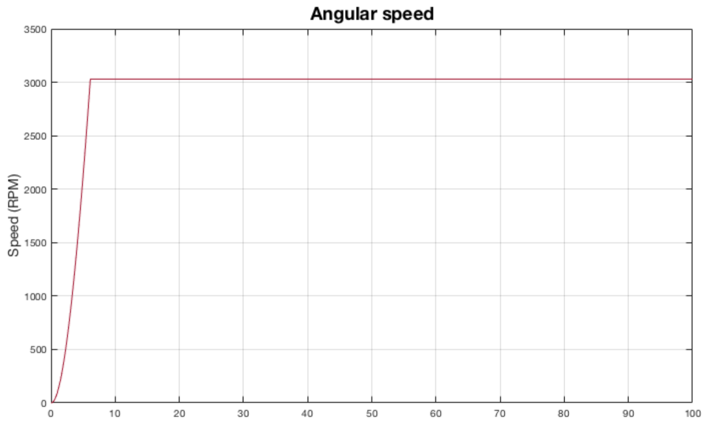


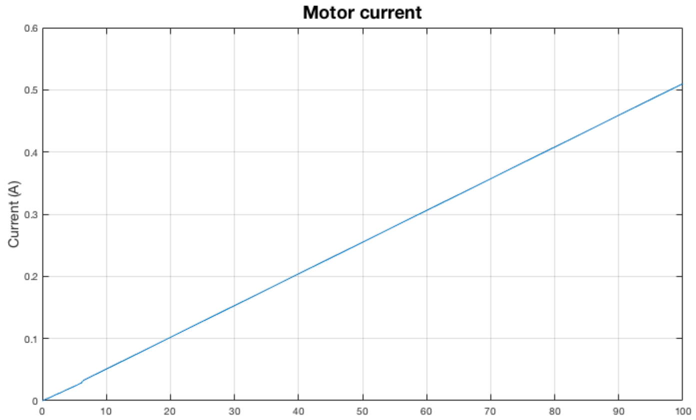


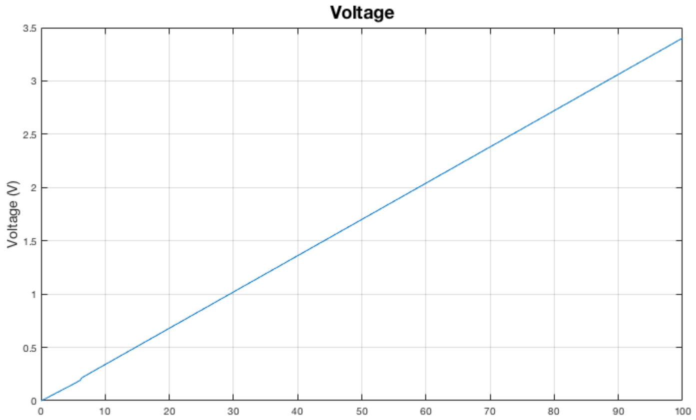


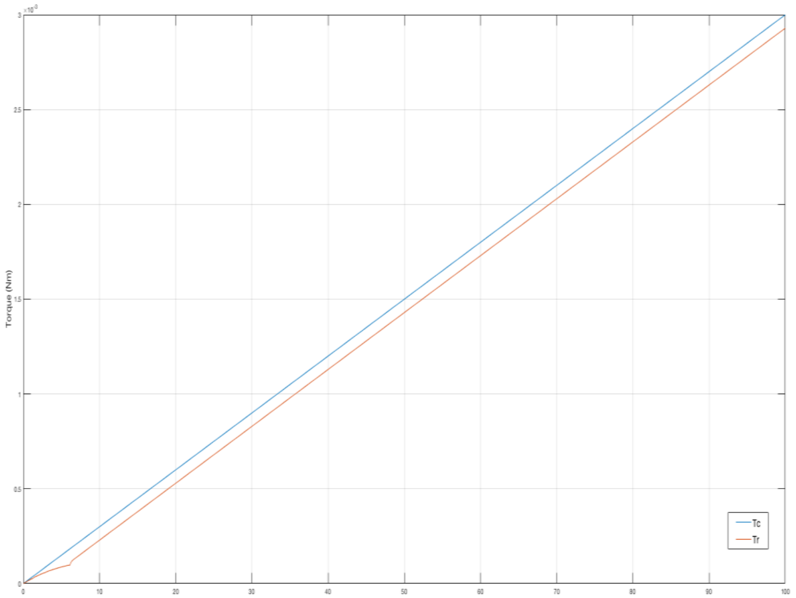
Nominal Operation



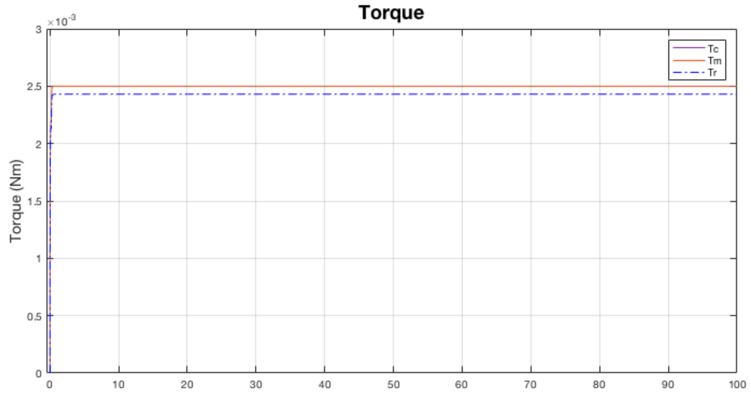


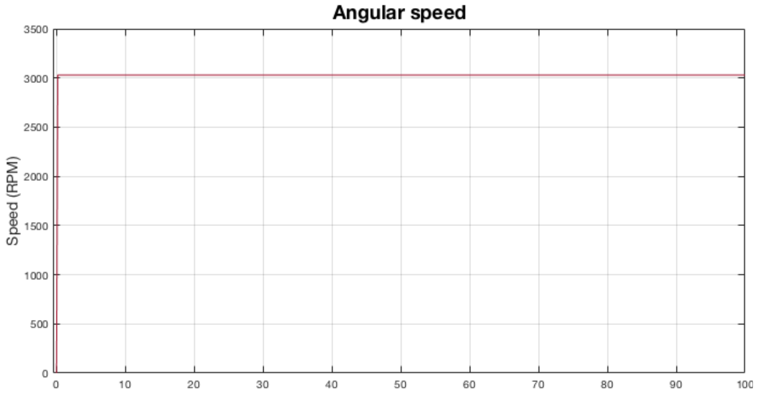


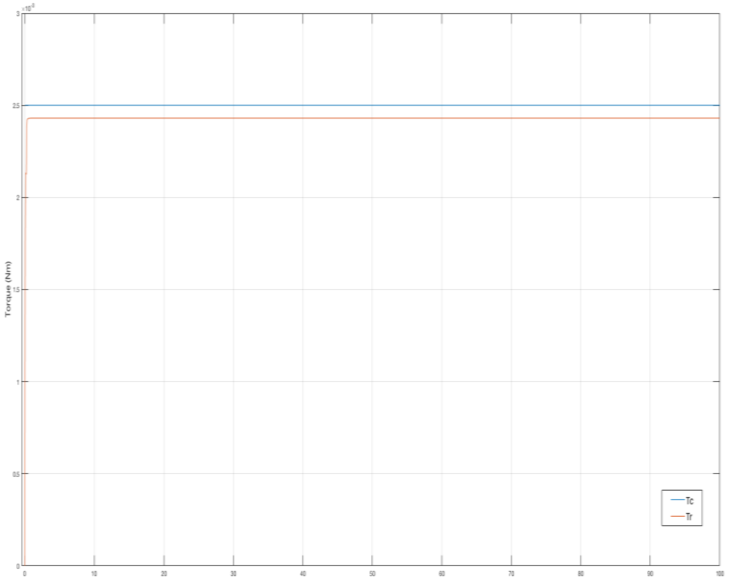


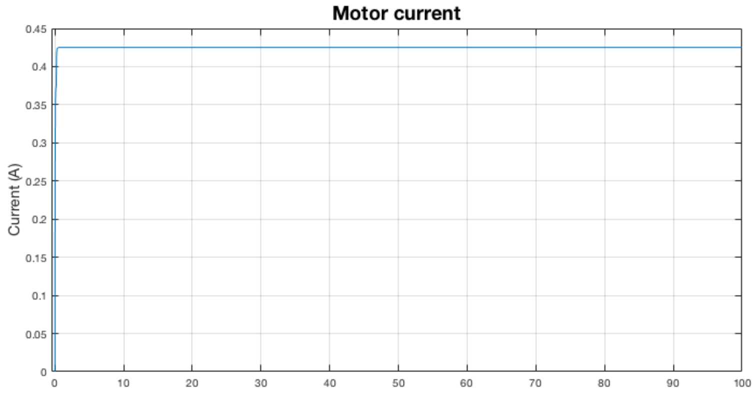


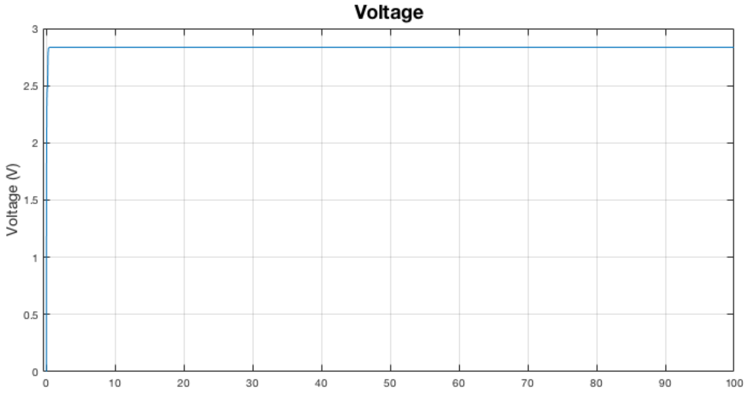
Nominal Operation







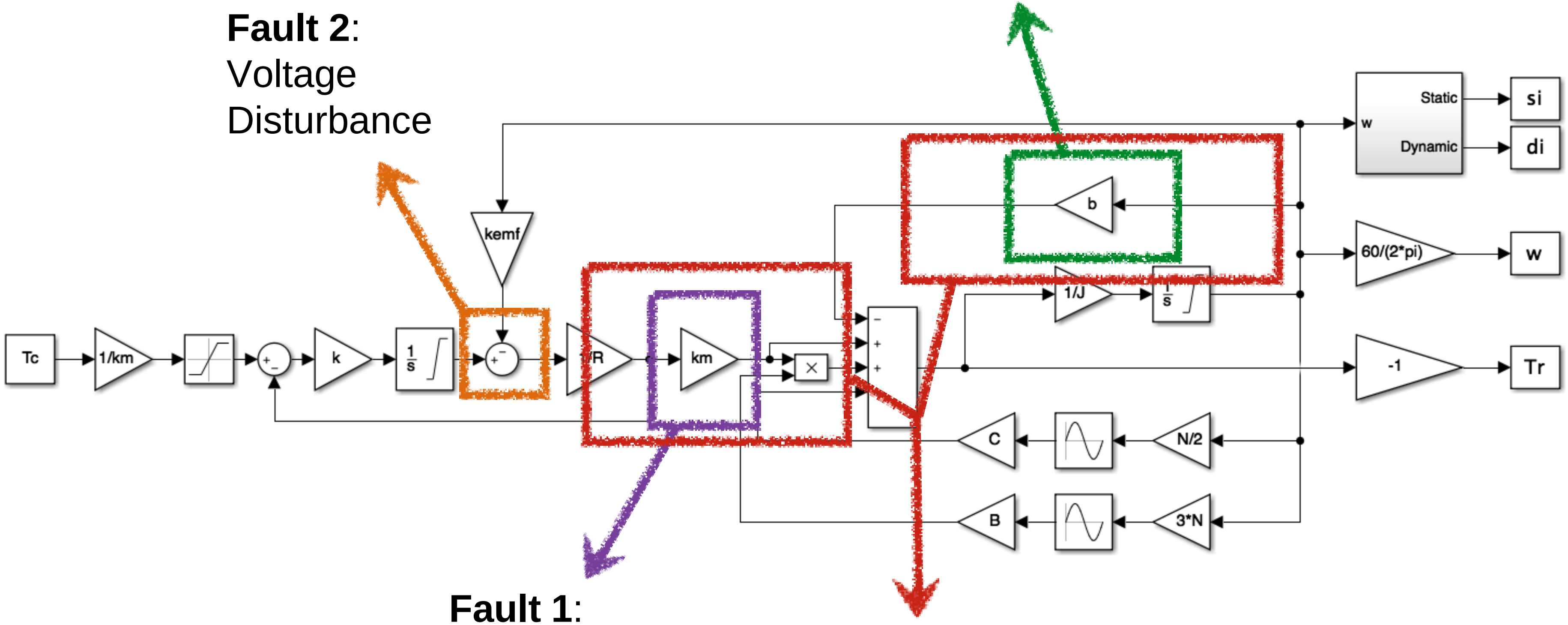




Faulty Operation

Fault 3:
Increase in
friction

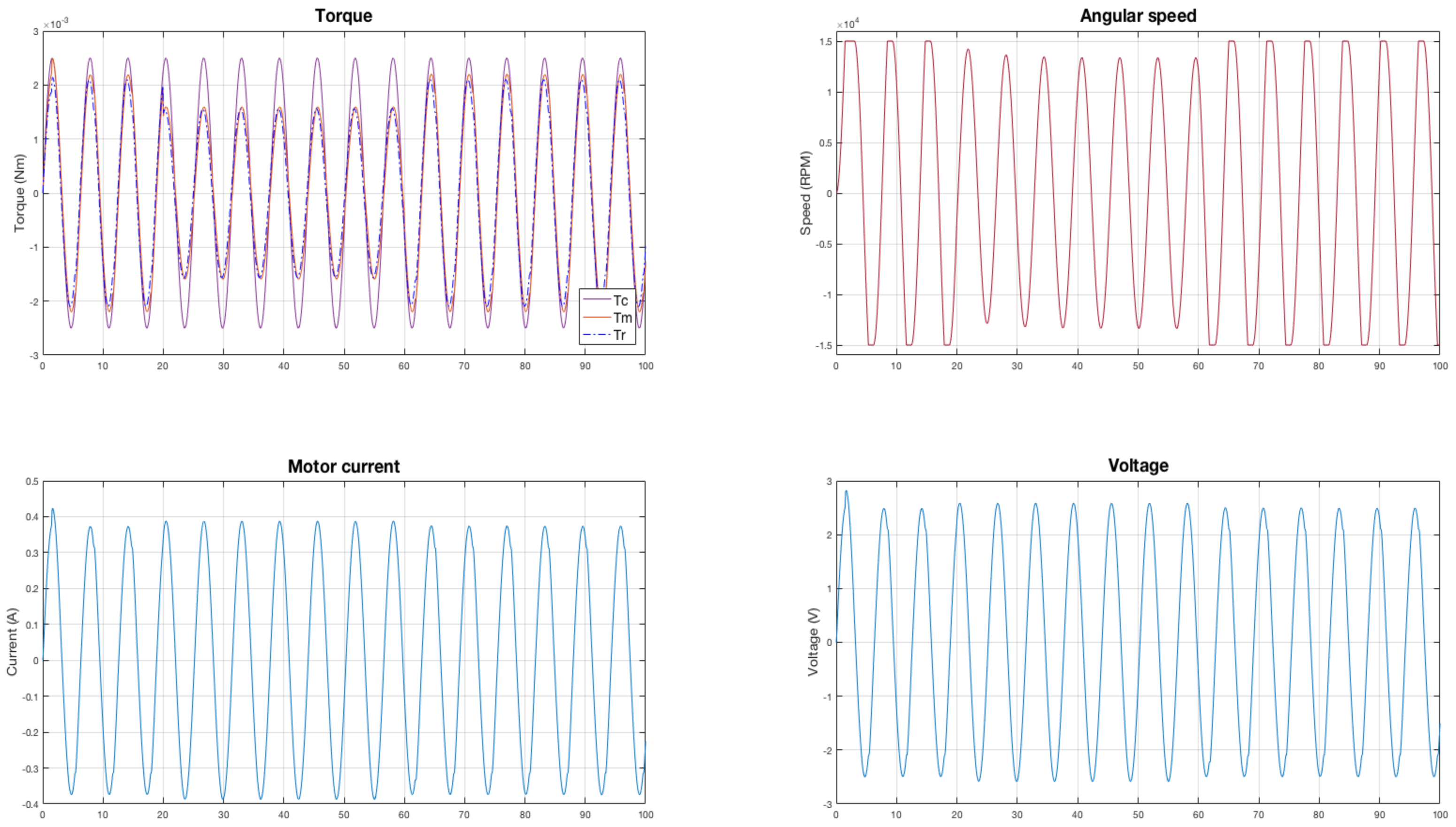
Fault 2:
Voltage
Disturbance

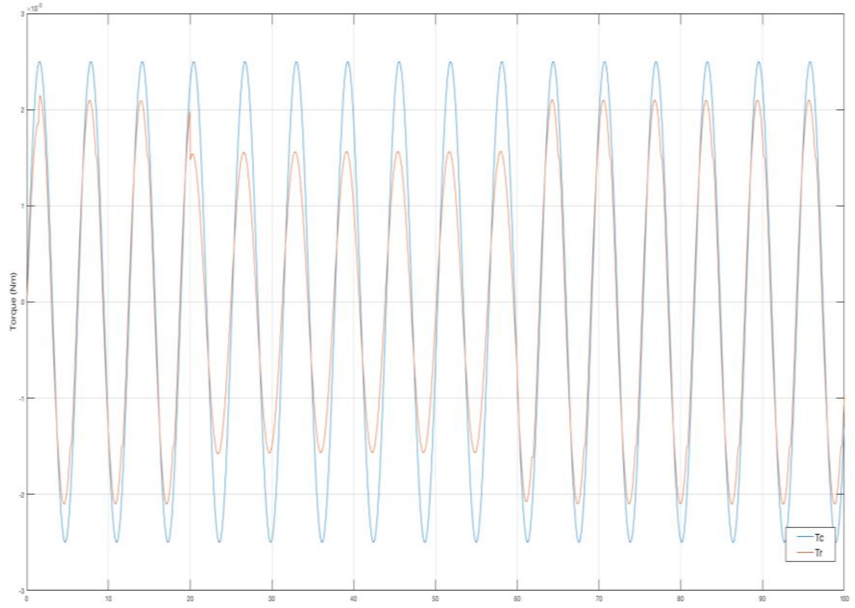


Fault 1:
Efficiency
loss

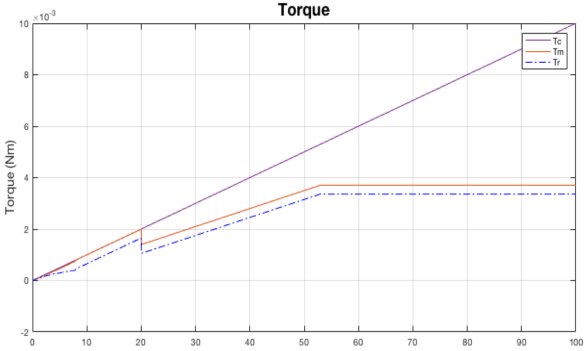
Fault 4:
Increase in
current with
increase in
friction

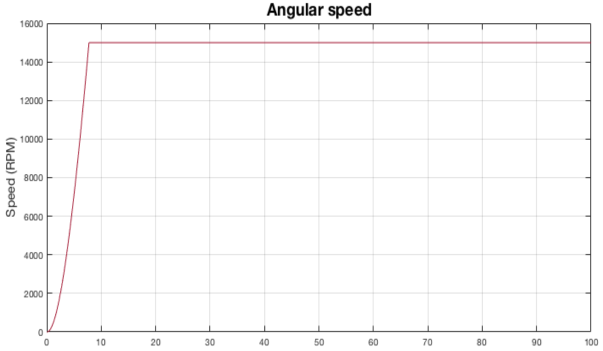
Fault 1: $k_m = 0.7k_m$

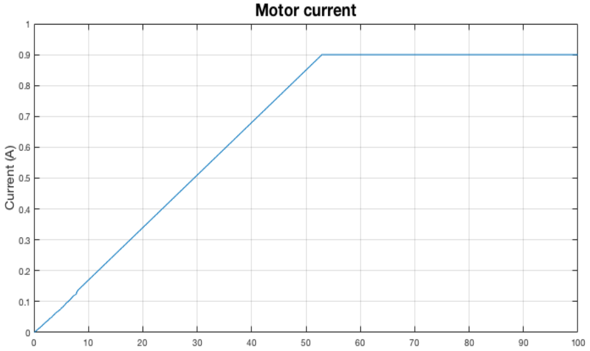


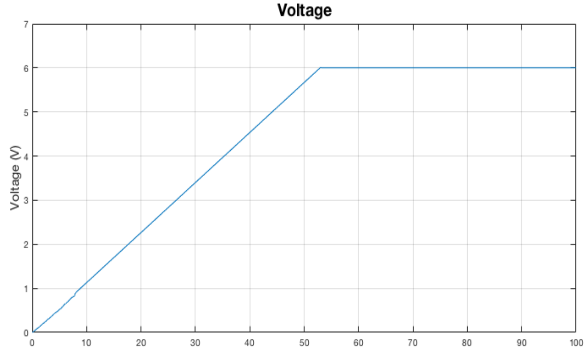


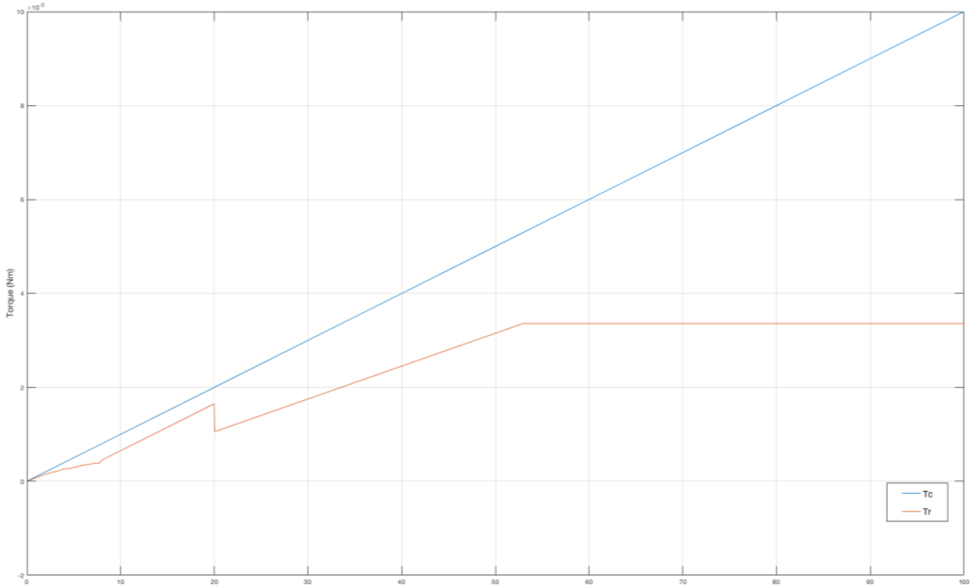
Fault 1: $k_m = 0.7 k_m$



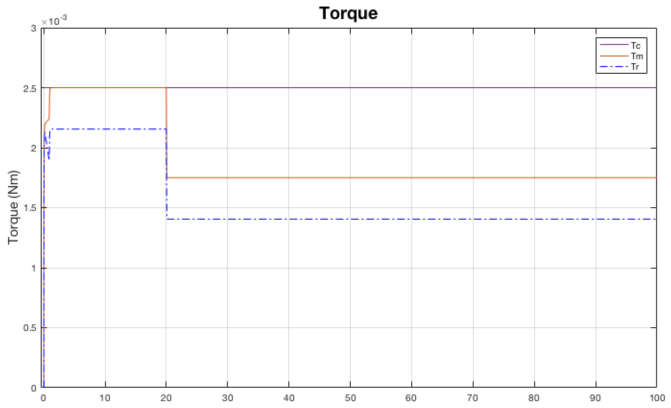


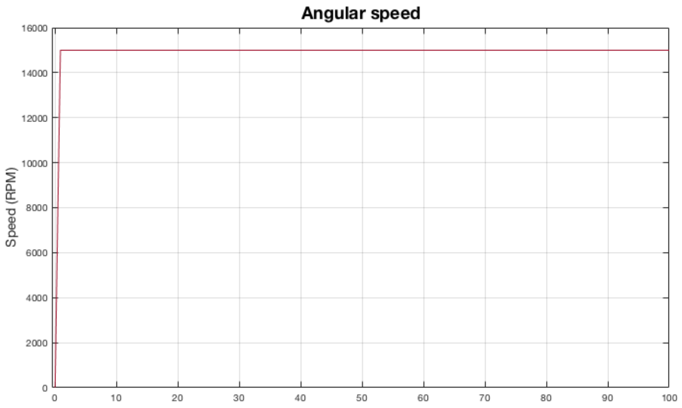


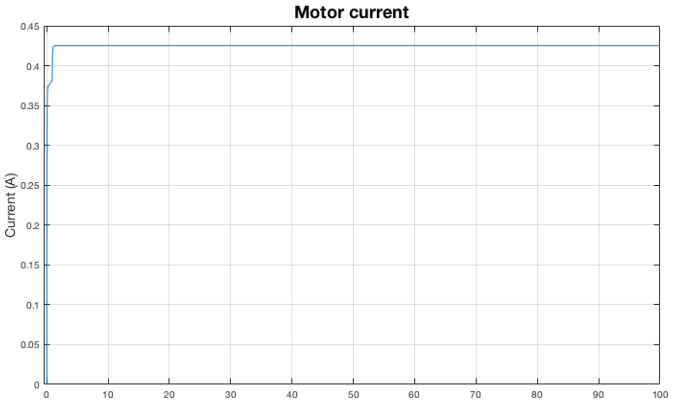


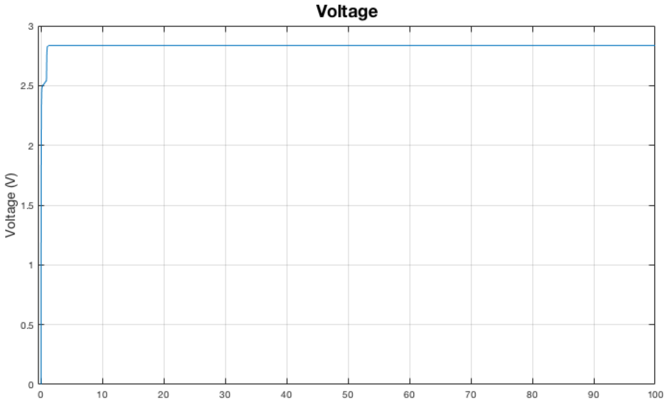


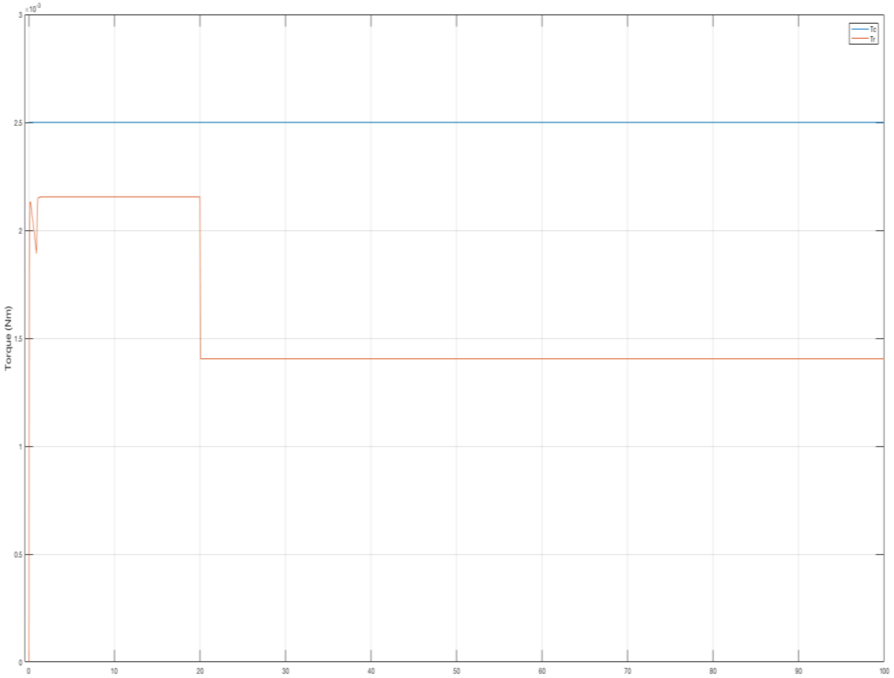
Fault 1: $k_m=0.7k_m$



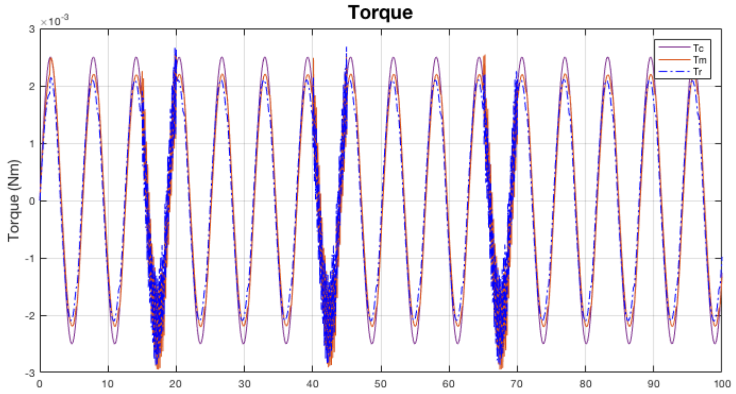


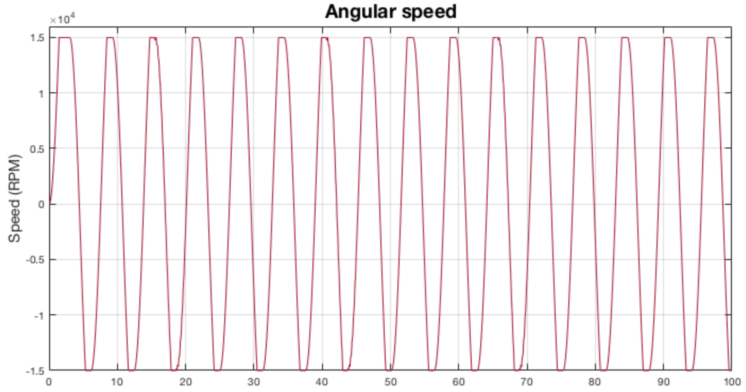


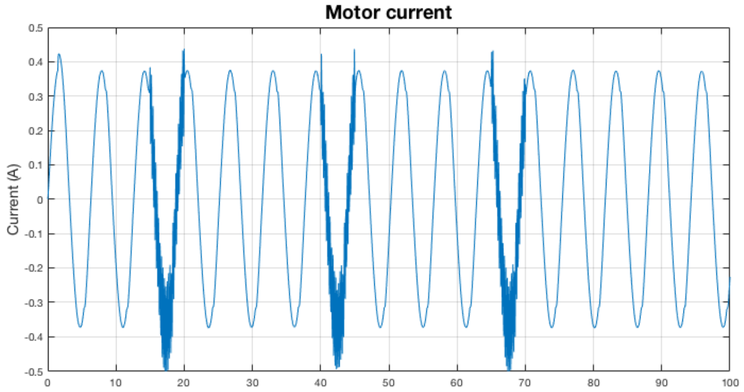


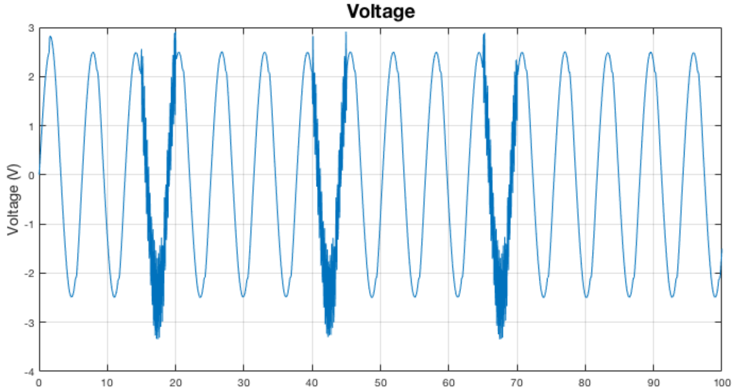


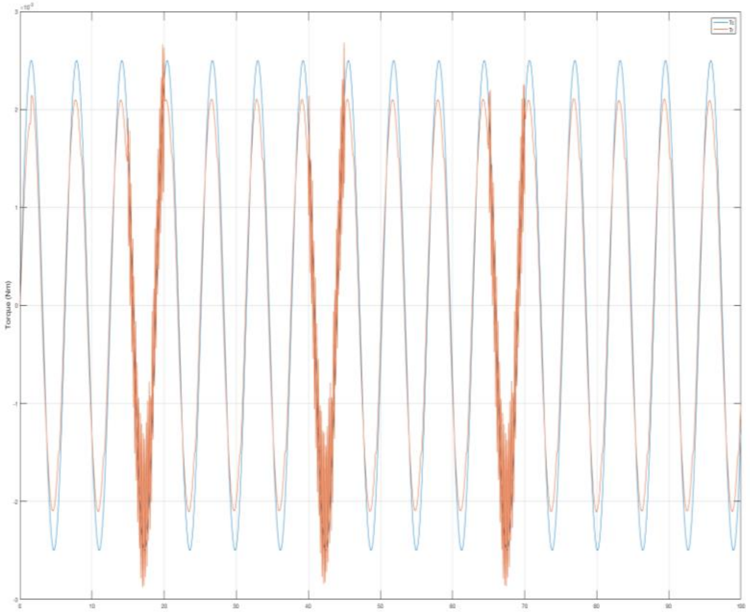
Fault 2: V_f =1.5*sin(30*t)



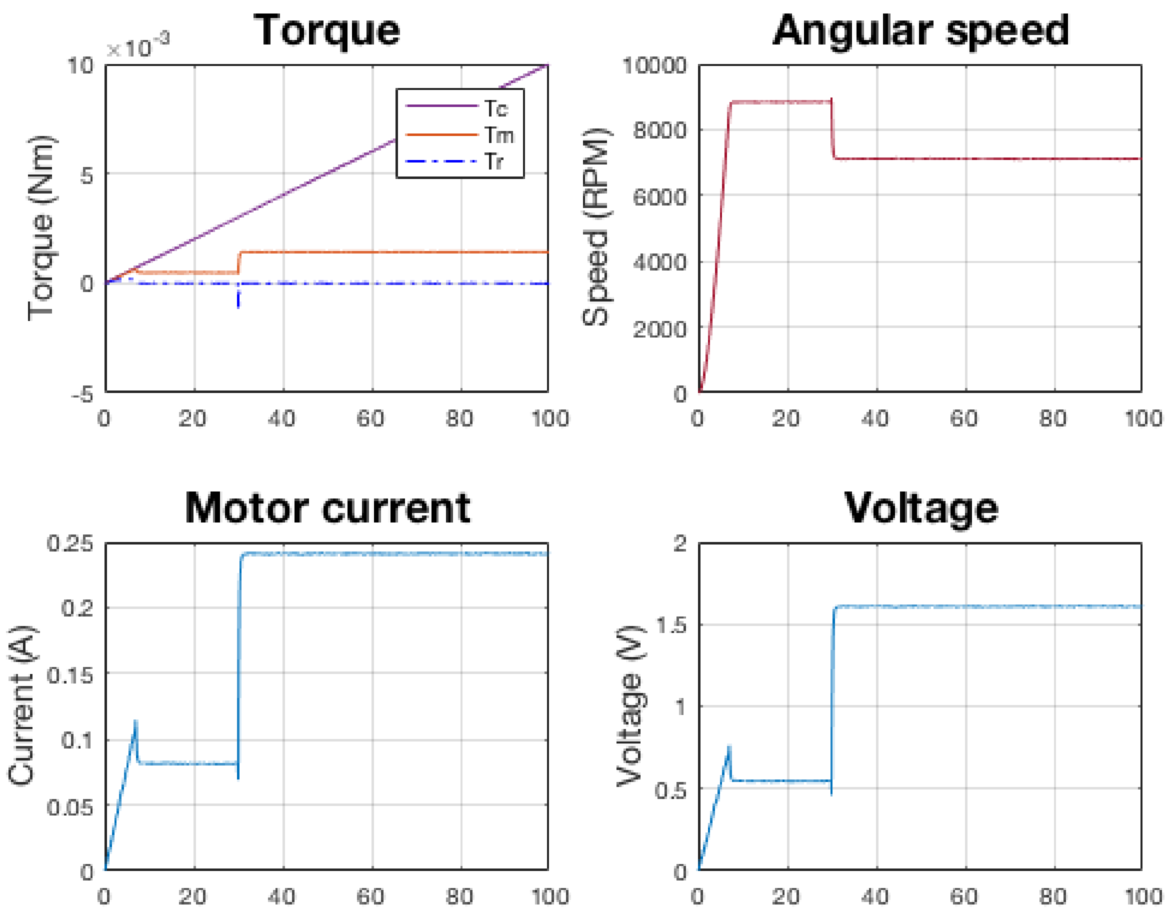


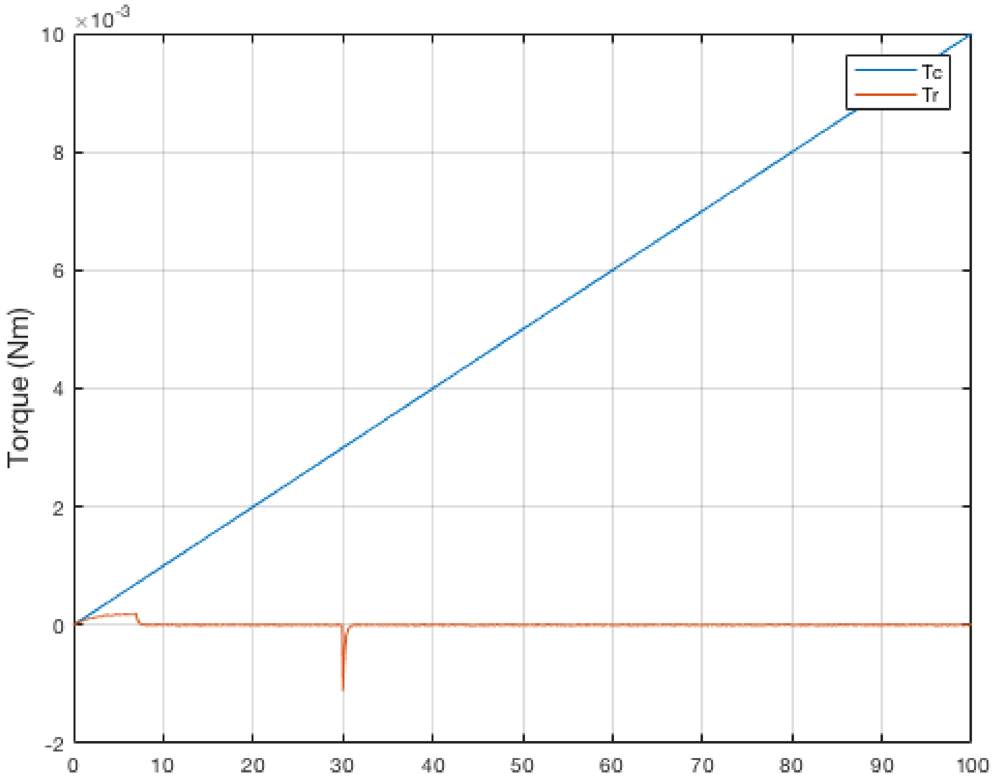




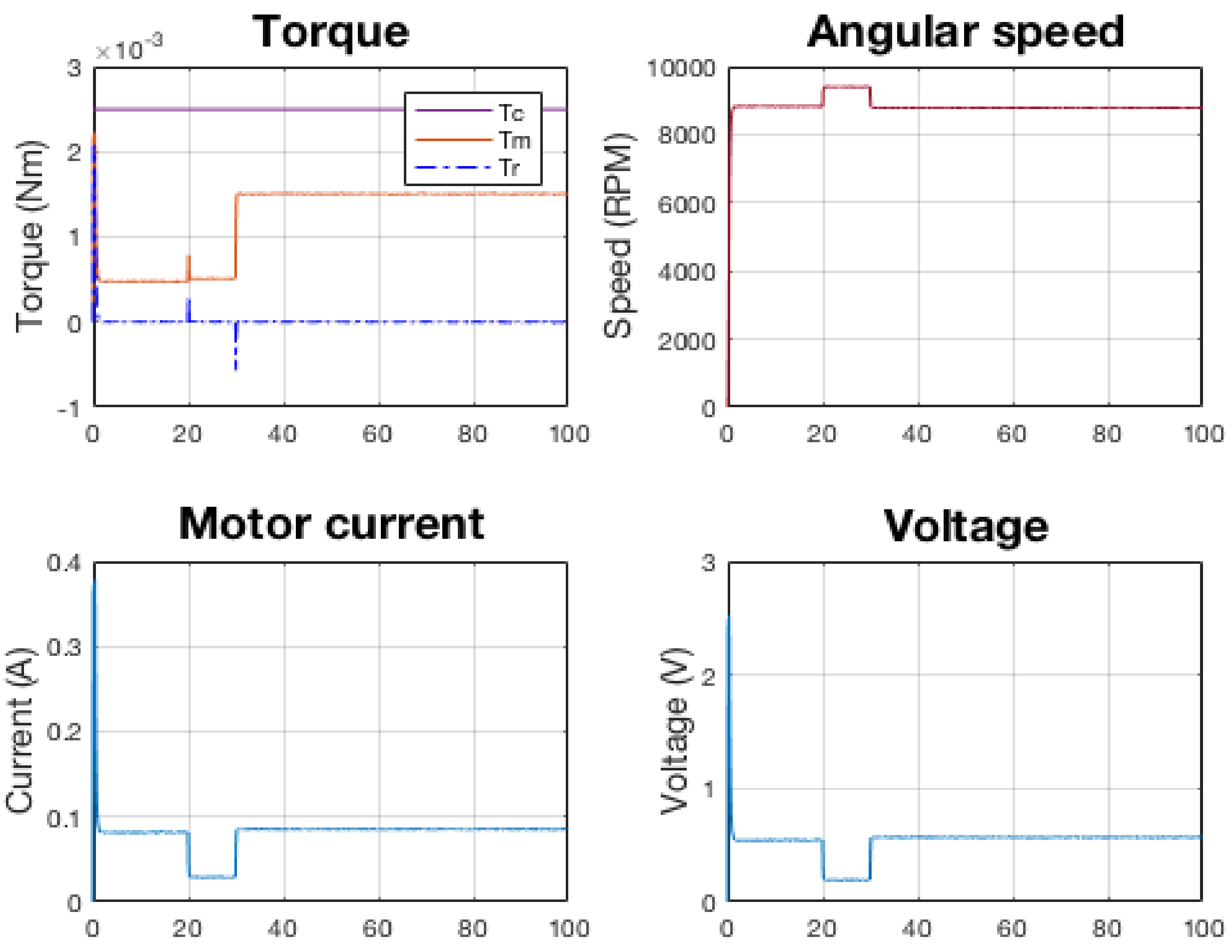


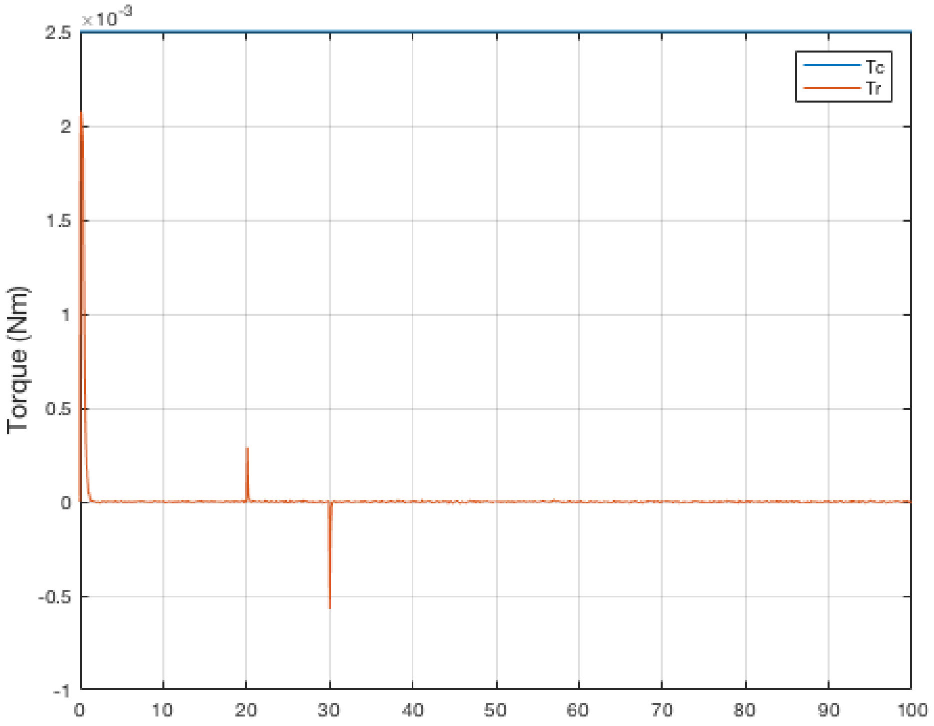
Fault 3: Friction Increase



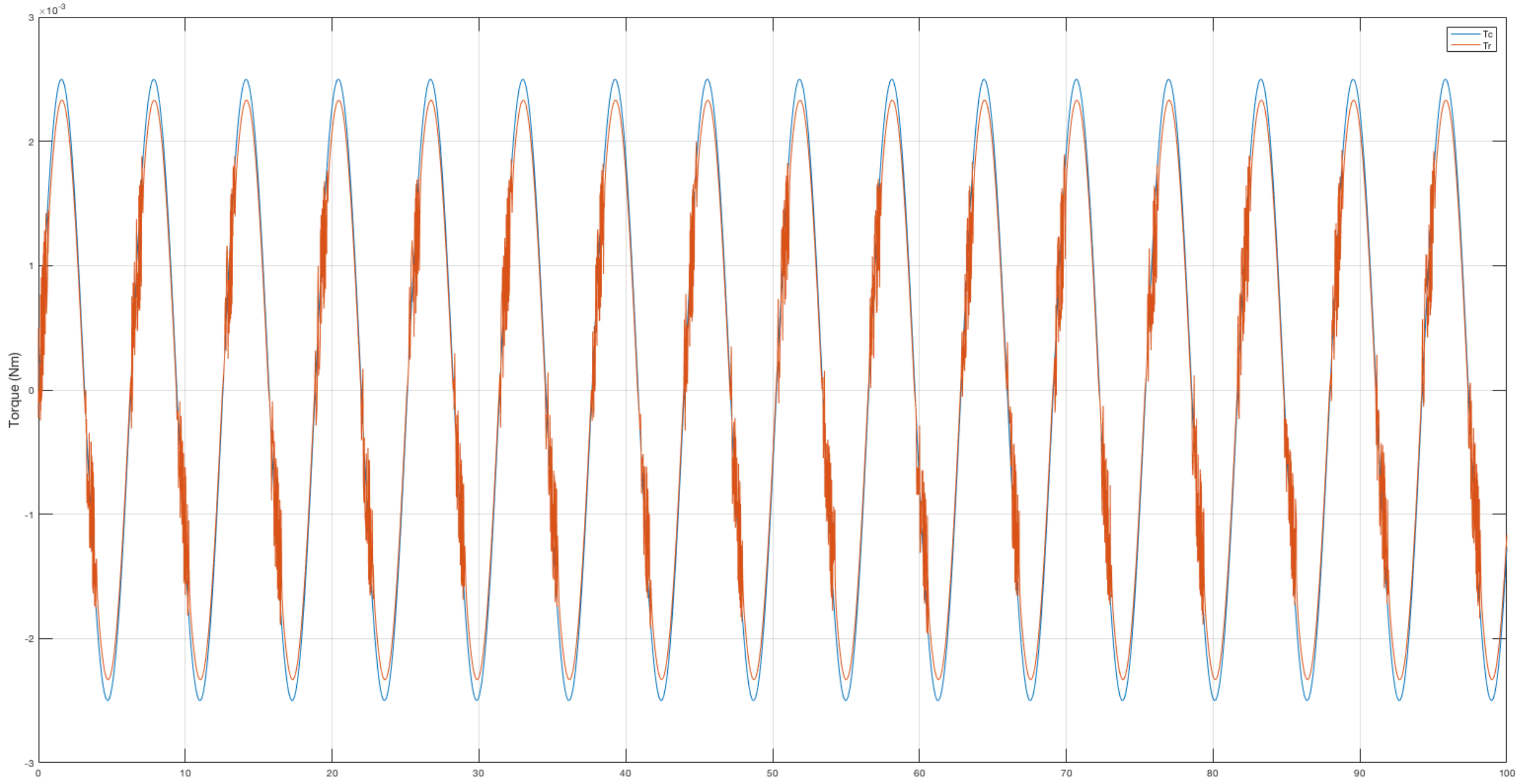


Fault 4: Increased current due to friction





Ripple and Cogging



Future Work on Satellite SHM

- Validate with experimental data (testbed design)
- Include spacecraft dynamics and momentum dumping devices to observe further faults
- Augmenting model with motor drive electronics
- Develop diagnostics and prognostics approach (i.e. wheel speed condition, energy consumption, etc.)

SHM Questions for Integration with Mission Design

- What is the asset? What is the system of interest within the asset.
  - The satellite formation? The satellite? The AOCS?
- What are the failure modes? FMECA (Failure Modes Effects & Criticality) analysis.
  - May give overview of subcomponents. Drill down to what and how the AOCS can fail in multiple ways (e.g. friction build-up, sensor failure, communication faults etc). Extensive literature review and past failure trend. E.g. Kepler mission had issues with AOCS (reactor wheels) in the past.
- Is anomaly detection, diagnosis, or remaining useful life the goal? Is anomaly detection enough? Is diagnosis enough? Why and why not?
- Do you really need remaining useful life (RUL)?
  - If so, why? Justify. E.g there are multiple missions and RUL can provide feasibility assessment of completion of current or alternate mission.
- What is the business model for this project? Are you selling Prognostics & Health Management (PHM) as an add-on service to current science mission satellites for their projects? Can you use it to incentivize sales of a particular satellite which will have this PHM capability or will you add it on to current satellites as a service?
- Are you using PHM for satellite swarms internally? Who is the customer/stakeholder?
  - What may you need to do to convince them that this is critical to mission success?
- Will the asset owners have to use the system? Will it be running in the background? If they don't use the PHM system, what could happen to their mission(s)?

SHM Questions for Integration with Mission Design

- What is mission success? How is it quantified? How successful is abandoning one mission for another? Is there a mission hierarchy? Can current mission still be completed if one satellite is down? How about two? What kind of missions are needed/common now with satellite formations?
- How far in advance does the PHM system need to work to be of practical value? How late is too late? How early is too early? Consider RUL vs reconfiguration time, current mission time remaining, future mission length of time?
- What is the value of a true detection? What is the cost of a missed detection? What is the cost of a false alarm?
- List three issues and gaps you see.
- Data-driven model? Physics-based model? Hybrid approach?
  - Is there data available? What kind? Is there environmental (rad hit and other environmental conditions), control system, operational (typical science missions for example), and maintenance data? Who owns the data? Will they give it to you?
- How much of the data is missing or incomplete? Is the data “real time” (usually there is a delay) or does it come as a batch after a run? Is the data readily available or will you have to work hard to assemble and organize it? Is the data all in the same database, or spread across several databases? How difficult will it be to merge the data?
- How many actual failed cases are there for each fault type?
- Is there a lot of variation between assets? Homogenous vs heterogenous component/system types? Is the performance of the asset strongly affected by the operating environment? Does variation in operational and maintenance programs between users strongly affect the lifetime of the asset (satellite or satellite component) or the manifestation of faults?

SHM Questions for Integration with Mission Design

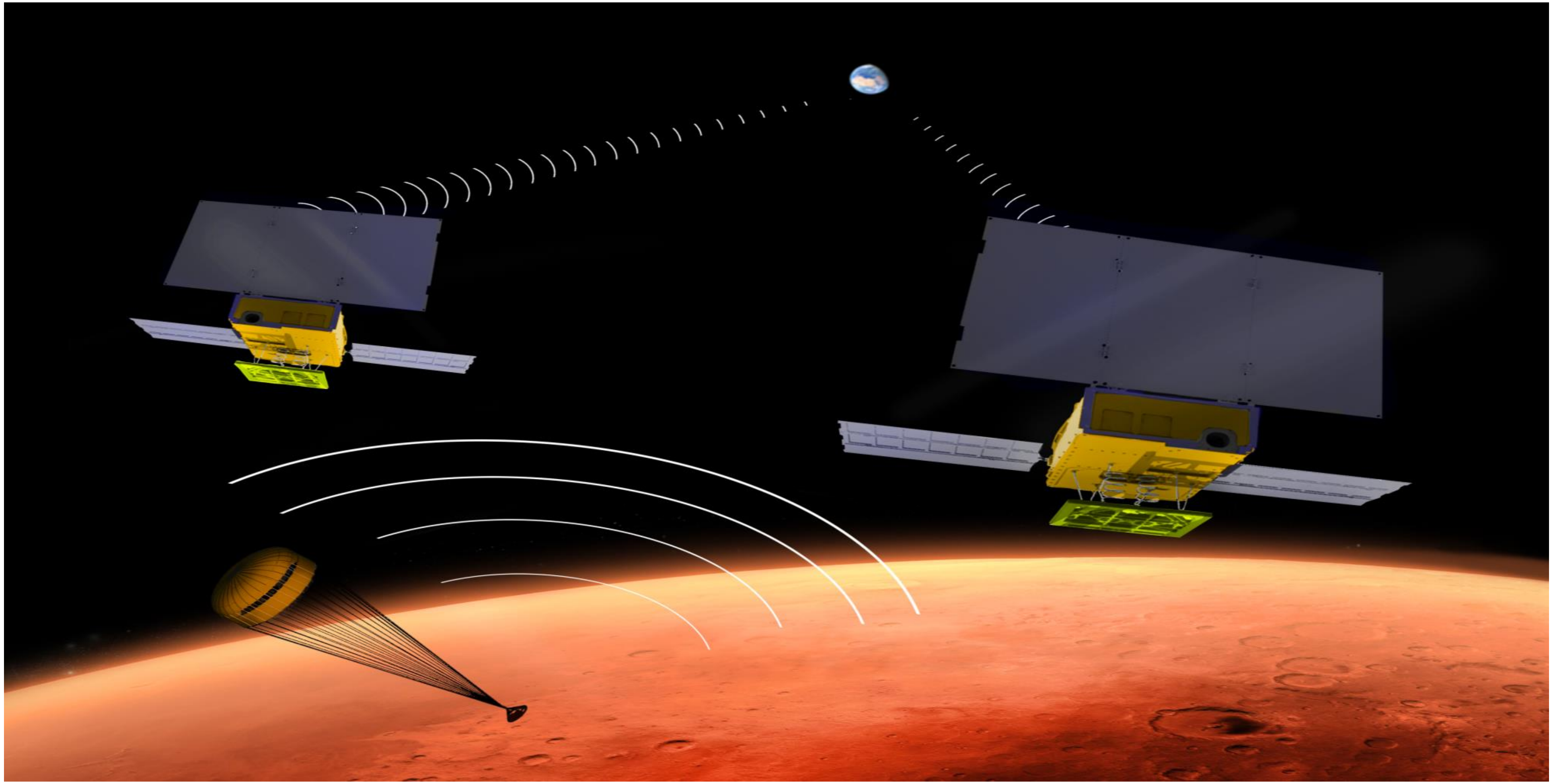
- ❑ What is the sampling rate of the data? Is the volume of data prohibitive? Do you have data from all of the combinations of environmental and operational conditions you want to model?
- ❑ Can a healthy asset (no fault) be simulated? Can a faulty asset be simulated (degradation mechanism etc)? Are all of the fault degradation mechanisms well understood? If you ask three engineers in the satellite/AOCS area, will they all agree on how to model the faults and normal wear? If not, how will you resolve the discrepancy?
- ❑ Is there already a model? Does it run quickly? Where will you do the simulations? Does the model account for environmental effects, wear, and variation between assets? Are there a ton of parameters to set in the model?
- ❑ Do experiments need to be run to model failure in the system or can the equations depict what failure looks like? Who will perform the experiments if so and how much will this cost?
- ❑ Are you willing to sacrifice some assets to validate models? Is there a good proxy for the failure mechanism that can be modeled in isolation? E.g. modeling reaction wheels failure independently of the satellite or satellite swarm.
- ❑ If you build a model and then switch vendors for a component, will the model still work?
- ❑ If you build a PHM system, how will it be used? Who will staff its use? Will there be a human in the loop? Frequency?
- ❑ How will decisions be made based on information from the PHM system? Reconfiguration strategy? Dependent on the mission(s)? Who will monitor the performance of the PHM system and keep it up to date over time?
- ❑ If you make the PHM system part of the asset, does it become another point of failure (i.e., if the PHM system breaks, can you operate the asset anyway)?
- ❑ Are there any potential regulatory issues? Are there any potential liability issues?
- ❑ List potential issues and gaps. E.g. validation data access etc

Acknowledgements

- Discovery & Systems Health Group (DaSH) Group
  - Lilly Spirkovska, Edward Balaban, Chetan Kulkarni, Chris Teubert, Anupa Bajwa
- Former Diagnostics & Prognostic Team Members
  - Matthew Daigle, Indranil Roychoudhury, Kai Goebel, & Shankar Sankararaman
- Visiting Researcher
  - Zainab Saleem
- Student Intern
  - Elizabeth Torres de Jesus
- Tracie Conn
- Prognostics & Health Management Society


thank you
danke
рахмат
謝謝
spas
ngiyabonga
شكراً جزيلاً
teşekkür ederim
Баярлалаа
mersi
kia ora
barka
welalin
tack
спасибо
faafetai lava
vinaka
спасибі
blagodaram
dank je
misaotra
matondo
paldies
grazzi
mahalo
tapadh leat
gracias
хвала
kiitos
dankie
nanni
nandri
dhanyavad
maururu
hvala
köszönöm
bedankt
bayarlalaa
gracje
aciu
dankon
akun
asante
manana
obrigada
tenki
murakoze
mochchakkeram
tau
djiere dieuf
enkosi
dziękuję
дякую
mamnun
chnorakaloutioun
grafias ago
gràcies
sulpáy
go raibh maith agat
chokrane
sobodi
dėkuji
sagolun
sukriya
kop khun krap
taiku
arigatô
takk
dakujem
trugarez
obrigado
mèsi
didi madloba
ありがとう
tanemirt
grazie
najis tuke
terima kasih
rahmet
kam sah hamnida
rahmat
diolch
dhanyavadagalu
shukriya
merce
мерси
ευχαριστώ
তোমাকে ধন্যবাদ
감사합니다
xièxie
merci

Questions?



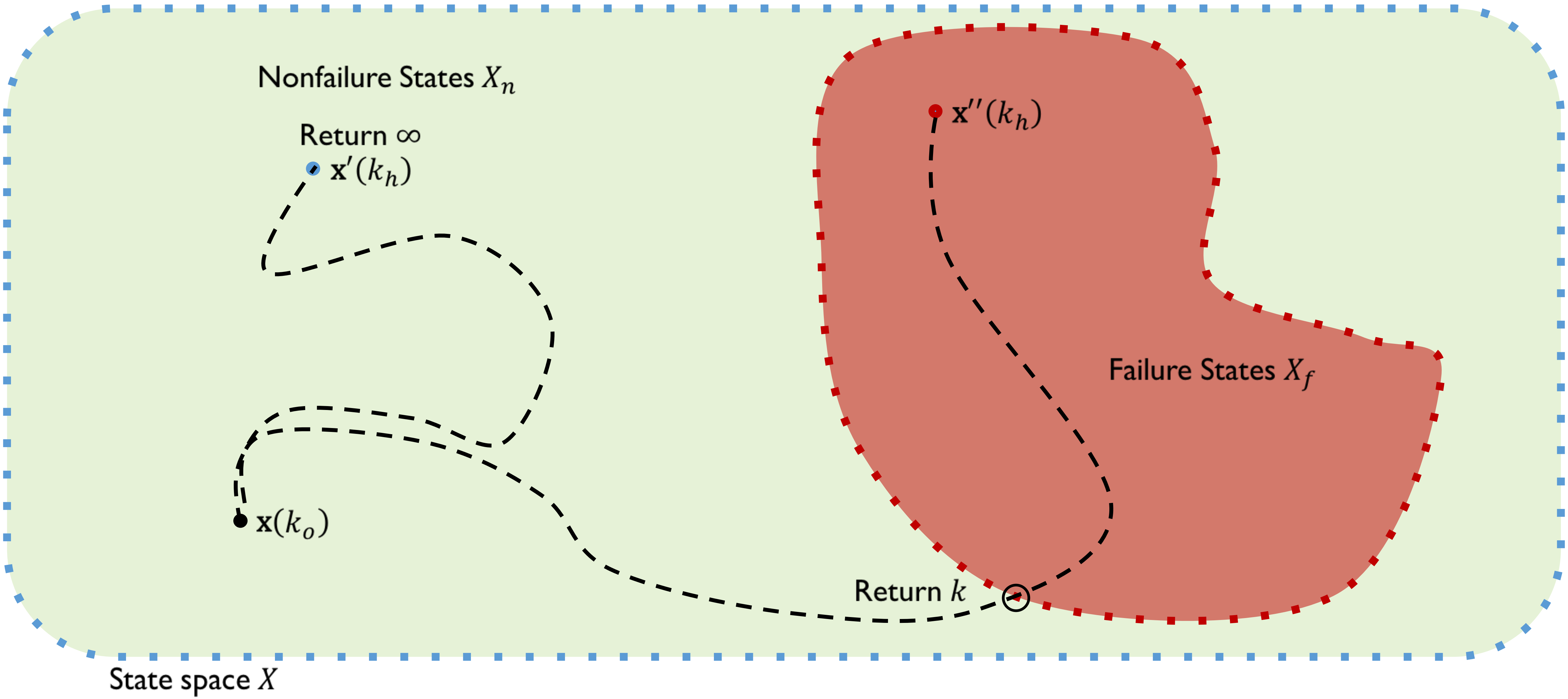
Source: NASA/JPL-Caltech

Back up slides

Initial Problem Formulation

- Assume we know
  - Initial state, $\mathrm{x}(k_o)$
  - Future input trajectory, $\mathbf{U}_{k_o,k_h} = [\mathrm{u}(k_o), \mathrm{u}(k_o + 1), \dots, \mathrm{u}(k_h)]$
  - Process noise trajectory, $\mathbf{V}_{k_o,k_h} = [\mathrm{v}(k_o), \mathrm{v}(k_o + 1), \dots, \mathrm{v}(k_h)]$
- Problem definition
  - Given $k_o, k_h, \mathrm{x}(k_o), \mathrm{U}_{k_o,k_h}, \mathrm{V}_{k_o,k_h}$
  - Compute EOL
    - $\mathrm{EOL}(k) = \inf\{k' : k' \geq k \text{ and } T_f(\mathbf{x}(k))\}$

Concept: ComputeEOL

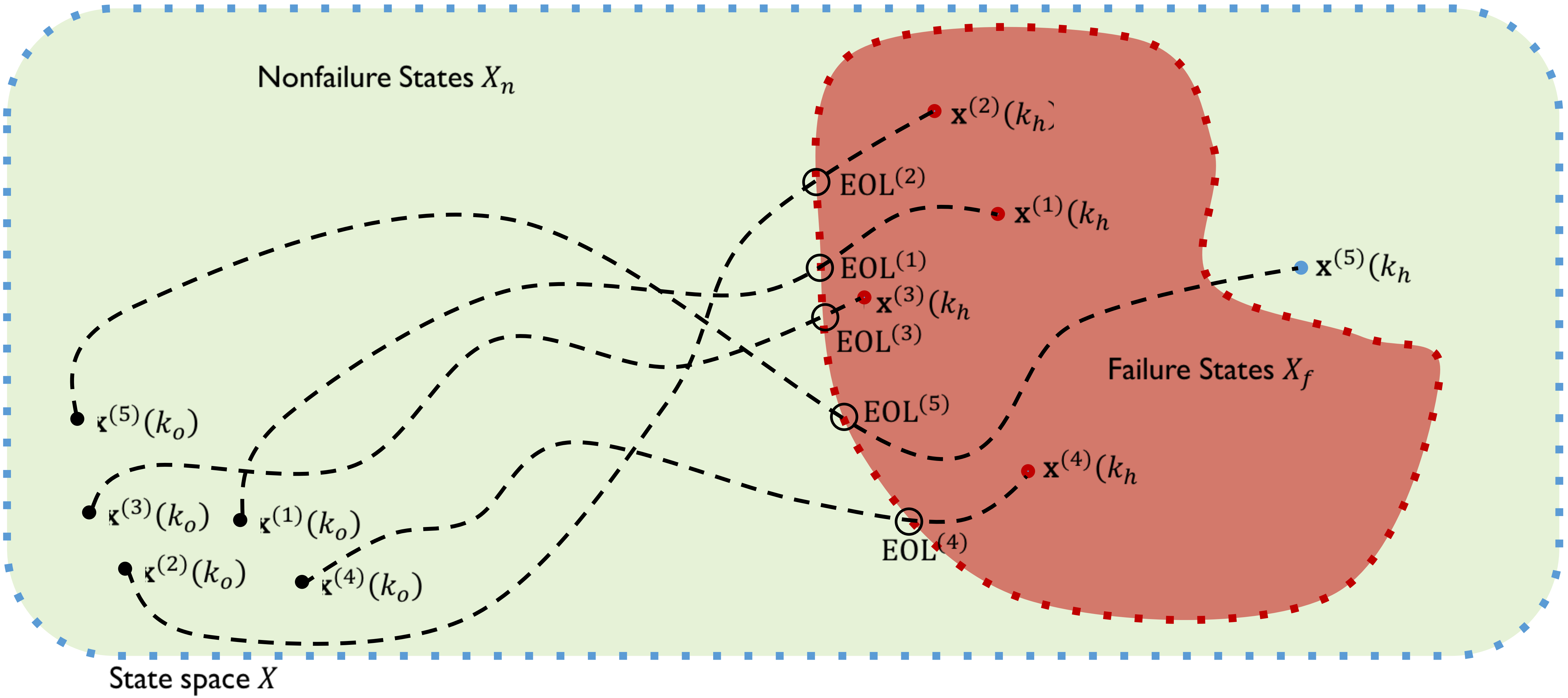


Computational Algorithm

- ComputeEOL$\left(k_o, k_h, \mathbf{x}(k_o), \mathbf{U}_{k_o,k_h}, \mathbf{V}_{k_o,k_h}\right)$

1. $\mathbf{X}_{k_o,k_h}(k_o) \leftarrow \mathbf{x}(k_o)$ *// Set initial state*
2. **for** $k = k_o$ **to** $k_h - 1$ **do**
3.   **if** $T_f\left(\mathbf{X}_{k_o,k_h}\right)(k)$ *// Check if failure state*
4.   **return** k *// Return current time as EOL*
5.   **end if**
6. $\mathrm{X}_{k_o,k_h}(k+1) \leftarrow \mathrm{f}\left(\mathrm{X}_{k_o,k_h}(k), \mathrm{U}_{k_o,k_h}(k), \mathrm{V}_{k_o,k_h}(k)\right)$ *// Update state*
7. **end for**
8. **if** $T_f\left(\mathbf{X}_{k_o,k_h}\right)(k)$ *// Check if failure state*
9.   **return** k *// Return current time (k_h) as EOL*
10. **else**
11.   **return** ∞ *// Return infinity*
12. **end if**

Concept: Uncertainty



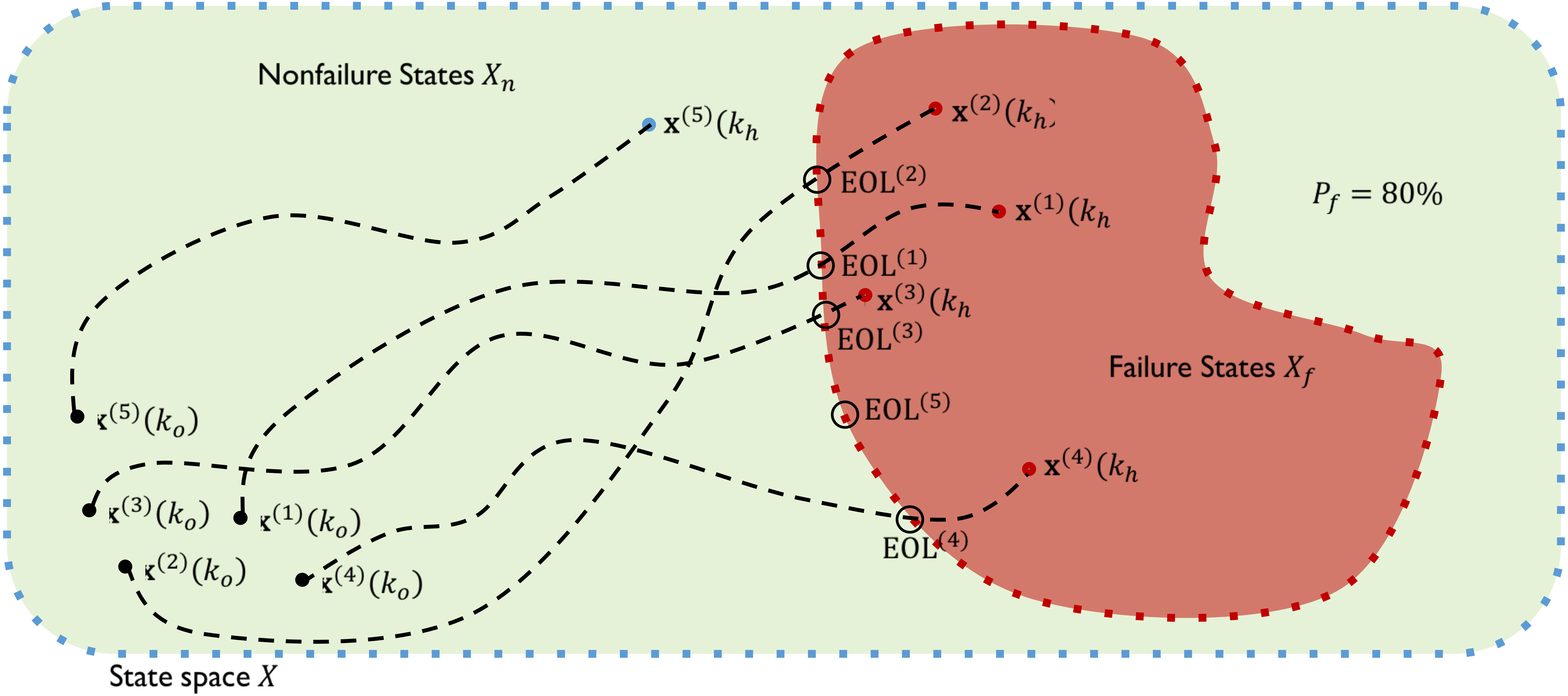
Handling Uncertainty

- Sources of uncertainty
  - Initial state, $\mathrm{x}(k_o)$
  - Future input trajectory, U_{k_o,k_h}
  - Process noise trajectory, V_{k_o,k_h}
- Requirements
  - Must define $p\left(\mathrm{x}(k_o)\right)$
  - Must define $p(\mathrm{U}_{k_o,k_h})$
  - Must define $p(\mathrm{V}_{k_o,k_h})$

Updated Problem Formulation

- Assume we know
  - Initial state distribution, $p\left(\mathbf{x}(k_o)\right)$
  - Future input trajectory distribution, $p(\mathbf{U}_{k_o,k_h})$
  - Process noise trajectory distribution, $p(\mathbf{V}_{k_o,k_h})$
- Problem definition
  - Given $k_o, k_h, p\left(\mathrm{x}(k_o)\right), p(\mathbf{U}_{k_o,k_h}), p(\mathbf{V}_{k_o,k_h})$
  - Compute $p(\mathrm{EOL}(k_o))$

Concept: Probability of Failure



Probability of Failure

- Can compute probability of reaching failure within the given time horizon within this framework
  - $P_f = P(\mathrm{x}(k_h) \in X_f)$, assuming that: $\mathrm{x}(k) \in X_f \vdash \mathrm{x}(k+1) \in X_f$

ComputePf$\left(k_o, k_h, p(\mathbf{x}(k_o)), p(\mathbf{U}_{k_o,k_h}), p(\mathbf{V}_{k_o,k_h})\right)$

1. *// Sample prediction inputs*
2. $\{(\mathbf{x}(k_o)^{(i)}, \mathbf{U}_{k_o,k_h}{}^{(i)}, \mathbf{V}_{k_o,k_h}{}^{(i)})\}_{i=1}^{N} \leftarrow$ GenerateSamples$(N, p(\mathbf{x}(k_o)), p(\mathbf{U}_{k_o,k_h}), p(\mathbf{V}_{k_o,k_h}))$
3. **for** $i = 1$ **to** N **do**
4. *// Compute EOL*
5. $\mathrm{EOL}^{(i)} \leftarrow$ ComputeEOL$(k_o, k_h, \mathbf{x}(k_o)^{(i)}, \mathbf{U}_{k_o,k_h}{}^{(i)}, \mathbf{V}_{k_o,k_h}{}^{(i)}$
6. **end for**
7. *// Return EOL realizations*
8. return $\left|\{\mathrm{EOL}^{(i)} : \mathrm{EOL}^{(i)} < \infty\}\right| \;\; /N$

Online Prognostics

- Up to now, described the problem of making a prediction at a single time point
- In online prognostics, predictions are made at several time points
- At each new time step k
  - Update state estimate, $p\left(\mathbf{x}(k)\right)$
    - Use state estimation algorithm (e.g., particle filter)
    - Requires output equation: $\mathbf{y}(k) = \mathbf{h}\left(\mathbf{x}(k), \mathbf{u}(k), \mathbf{n}(k)\right)$, where $\mathbf{n}(k)$ is sensor noise
  - Update future input trajectory distribution, $p(\mathbf{U}_{k,k+h})$
    - May update based on new load schedule
    - May be a function of the state
  - Update process noise trajectory distribution, $p(\mathbf{V}_{k,k+h})$
    - Usually, we assume this is independent of k