Contents

- AERO 626 Homework #2
- Part A: Least-Squares Estimate of State
- Part B: Weighted Least-Squares Estimate of State
- Part C: Weighted Least-Squares Estimate with prior information
- Part D: LUMVE of State

AERO 626 Homework #2

Spring 2023 David van Wijk

```
data = load('data_HW02.mat');
format long

% DATA PROVIDED ARE:
% T = (m x 1) array of measurement times [s]
% Z = (m x 1) array of position measurements [m]
% W = (m x 1) array of measurement weights [nd]
% R = (m x 1) array of measurement noise covariances [m^2]
```

Part A: Least-Squares Estimate of State

```
F = [0 1; -1 0];
H_tilde = [1 0];
H = [];
for i = 1:length(data.T)
    Phi_i = expm(F*(data.T(i) - data.T(1)));
    H_i = H_tilde*Phi_i;
    H = [H; H_i];
end
disp('Least-Squares Estimate of Initial State using matrix exponential for Phi:')
x_hat_0 = (H'*H)\(H'*data.Z)

H = [];
for i = 1:length(data.T)
    t_i = data.T(i);
    Phi_i = [cos(t_i) sin(t_i); -sin(t_i) cos(t_i)];
    H_i = H_tilde*Phi_i;
    H = [H; H_i];
end
disp('Least-Squares Estimate of Initial State using analytical solution for Phi:')
x_hat_0 = (H'*H)\(H'*data.Z)
```

```
Least-Squares Estimate of Initial State using matrix exponential for Phi:

x_hat_0 =

   1.001143323295085
  -0.005513909940539

Least-Squares Estimate of Initial State using analytical solution for Phi:
```

```
x_hat_0 =

   1.001143323295085
  -0.005513909940539
```

Part B: Weighted Least-Squares Estimate of State

```
W = diag(data.W);
disp('Weighted Least-Squares Estimate of Initial State:')
x_hat_0 = (H'*W*H)\(H'*W*data.Z)
```

```
Weighted Least-Squares Estimate of Initial State:

x_hat_0 =

   1.004569504535051
  -0.007651308799164
```

Part C: Weighted Least-Squares Estimate with prior information

```
x_bar = [1; 0];
W_bar = [3 0; 0 3];
disp('Weighted Least-Squares Estimate of Initial State using prior info:')
x_hat_0 = (H'*W*H + W_bar)\(H'*W*data.Z + W_bar*x_bar)
```

```
Weighted Least-Squares Estimate of Initial State using prior info:

x_hat_0 =

   1.004375010528215
  -0.007340452622603
```

Part D: LUMVE of State

```
disp('LUMVE of Initial State:')
P_vv = diag(data.R);
x_hat_0 = (H'*P_vv^-1*H)\(H'*P_vv^-1*data.Z)
disp('The uncertainty in our measurement can be evaluated using the covariance matrix of the estimate:')
cov_matrix = (H'*P_vv^-1*H)^-1
```

```
LUMVE of Initial State:

x_hat_0 =

   1.001143323295085
  -0.005513909940539
```

```
The uncertainty in our measurement can be evaluated using the covariance matrix of the estimate:

cov_matrix =

   1.0e-03 *

   0.188443928965983  -0.006703912786495
  -0.006703912786495   0.209123502754448
```

Published with MATLAB® R2022b

Final Project Proposal: Fault-Tolerant Run Time Assurance
AERO626, Spring 2023

Name: David van Wijk UIN: 932001896

1 Problem Statement

For my final project, I intend to use estimation theory to estimate complex, nonlinear system dynamics in the development of a Fault-Tolerant Run Time Assurance (FT-RTA) structure. For full transparency, this is a concept I developed with my collaborators at the Air Force Research Laboratory's Safe Autonomy Team, but as of right now no work has been done on this project other than the ideation phase. Run Time Assurance (RTA) techniques are a relatively recent development in control theory, acting as a wrapper which monitors the actions of a high performance but possibly more dangerous controller, and intervenes in the case of safety constraint violations. The main appeal of RTA architectures is that they allow for the deployment of controllers such as Machine Learning (ML) based controllers. Currently, no guarantees can be made on the performance of ML based controllers, unlike most traditional controllers. RTA techniques allow this issue to be mitigated by using a wrapper to monitor the actions of the black box controller, and intervene only when absolutely necessary. The proposed problem is a large undertaking, and encompasses much more than estimation theory. The full scope of the problem will be described for clarity, but the remainder of the proposal will attempt to focus on the estimation theory portion of the proposed research.

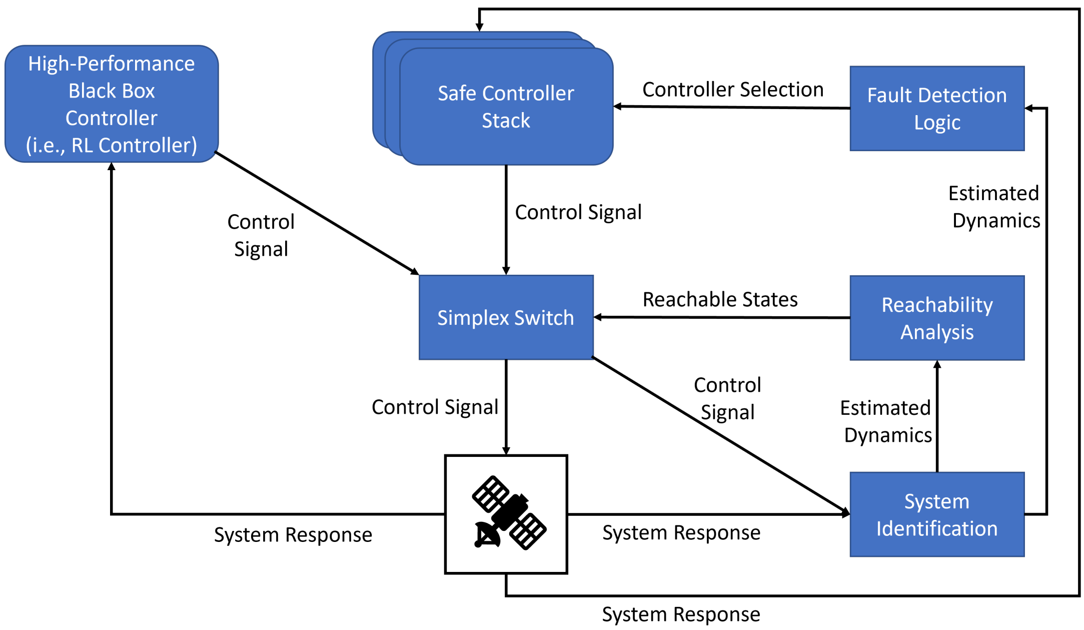

Figure 1: Proposed Fault-Tolerant Run Time Assurance Framework.

FT-RTA is a proposed framework to deal with possible failures that may occur during normal operation of complex systems, such as a spacecraft, fixed-wing aircraft or multi-rotor vehicles. An outline of the proposed framework is illustrated in Figure 1 and the example of a spacecraft is considered throughout the proposal. In the event that a failure such as a thruster failure or change in inertial properties occurs, the dynamics of the spacecraft would be modified, sometimes very starkly. Using the system response from the spacecraft, as well as the previous control signal, the modified dynamics can be estimated using system identification techniques, making use of Baysian methods and a type of Kalman filter. It will be clear when a fault occurs due to the drastic change in the estimated system dynamics. Using the estimated dynamics, a safe backup controller can be selected from a stack of possible controllers designed to deal with different faults. Additionally, reachability analysis can be performed using the estimated system dynamics to compute the reachable states of the system. Using the reachability analysis, the simplex switch identifies when it is safe to use the black box controller, and when it must switch to the safe controller.

2 Literature Review

There have been many successful attempts to use RTA for complex aerospace systems in the literature. Dunlap et al. demonstrated the effectiveness of a RL controller for spacecraft docking in tandem with RTA methods to ensure safety [2]. The authors also developed several translational motion safety constraints incorporated in the RTA framework to ensure safety for multi-agent spacecraft inspection [3]. Schierman et al. developed a multi-level RTA framework for a fleet of unmanned aircraft systems, demonstrating the effectiveness of these methods for complex, safety critical missions [7].

System identification techniques for estimating dynamics of complex systems is an area of engineering with a rich history and extensive literature. Green et al. were able to estimate the dynamics of a system using data with the presence of noise [4]. In this paper they use Bayesian inference, and demonstrate its effectiveness using a spring mass damper. Cortiella et al. show that an ℓ_1-regularized least squares strategy is able to identify nonlinear system dynamics in the presence of state measurement noise robustly and accurately [1]. Wan et al. utilize an unscented Kalman filter (UKF) for nonlinear system identification, and contrast its performance with the extended Kalman filter (EKF) [8]. Juang et al. developed theory for Markov parameter identification using an observer or using a Kalman filter from input and output data, and use this theory for a novel algorithm to compute these Markov parameters [5]. The Markov parameters can inform a state-space representation of the system.

3 Expected Results

The minimum viable product for this project will be the completion of the bottom right section of the proposed FT-RTA framework (Figure 1). This will include accurate identification of system dynamics using a type of Bayesian filter and Kalman filter, and accurate fault detection from the set of possible faults. In order to do this, a framework will be built for modeling the system dynamics, propogating them, sampling the system data, and injecting faults. Ideally, I will also build out the rest of the proposed framework to successfully synthesize the estimation block with the remainder of the components. I expect to use a Monte Carlo analysis to characterize the effectiveness of the estimation techniques and fault detection logic combined.

4 Work Plan and Timeline

The main technical objectives are enumerated below in chronological order, accompanied by a short description and an estimate for completion time.

1) **List of Faults:** A set of possible faults will be generated ordered by severity and likelihood. This will be done by mid February.

2) **Base Dynamics Simulation:** The dynamics of the chosen system without the presence of faults will be developed and simulated. Simulations should allow for system response data to be sampled, and will also allow sensor and measurement noise to be injected and by tuned by the user. This will be finished by the end of February.

3) **Fault Injection:** A method for artificially introducing faults will be developed, which will simulate faults proposed in the first stage. This will be finished by the end of February.

4) **System Identification Framework:** Since the purpose of this final project is to gain a deep understanding of estimation theory, this portion of the project should be the most detailed. A system identification framework will be developed, to take a control signal and system response data, and estimate the dynamics of the system. This framework will allow the user to toggle which estimation techniques to utilize. This sub-task will be finished by the end of March.

5) **Fault Detection Logic:** Once the system identification block has been constructed, it will be necessary to use the estimated system dynamics to identify faults. This will be finished by mid-April.

6) **(Optional) Reachability Analysis:** For the complete FT-RTA it will be important to incorporate a method for propagating the estimated dynamics forward in time to determine the reachable states. If I find myself ahead of schedule for this project, I will incorporate a reachability tool in the framework, and would like to do this by mid-April.

7) **(Optional) Safety Controller Development:** Developing the safe controller stack based on possible faults could be a final project on its own, but time permitting, it would be excellent to develop one safe controller for one of the faults within the fault set. Time permitting, it would be excellent to have this finished by the end of April.

8) **(Optional) Reinforcement Learning Controller Development:** Again, for the full framework, it will be crucial to have an appropriate ML based controller to test on. For the scope of this project, I may use a Reinforcement Learning (RL) controller that I have already built and gauge its performance in the presence of faults. However, this remains a stretch task that would be great to complete before the conclusion of the semester.

9) **Monte Carlo Analysis:** To evaluate the effectiveness of the system identification and fault detection block, I will conduct a series of tests using the set of faults determined in step 1, varying the noise, initial conditions, and the estimation technique utilized. This will be the final step of the final project and will be completed by the end of April.

10) **Final Report:** Since the project will culminate in the final report, I will be writing intermittently throughout the semester such that the majority of the writing has been done by the end of April. I will use the first week of May to add any final results and polish the final report.

5 Challenges

The proposed framework is quite a large undertaking, and there are many sub-components which could pose issues. Firstly, the fault detection logic may be difficult, because while it is important to detect and classify failures, it is also important that false positives are minimized. Detecting a fault and resorting to a backup controller due to a mis-classified failure could negatively impact the mission. To overcome possible issues in this area, I could try to emulate the work done in [6], and model the faults as external forces or torques acting on the original system.

It is also possible that I will face challenges in accurately estimating the dynamics of the system. In this case, I will try various system identification algorithms and techniques, since this field is quite rich with options. Additionally, my advisor Dr. Majji will likely be able to provide useful feedback and advice in this area.

As mentioned in the previous sections, the full FT-RTA framework is a large undertaking that may take more than a semester to complete, which poses an obvious challenge. However, to overcome this challenge, I have outlined that the estimation block will be the priority, and the other blocks will be developed only in the case where I have satisfactorily completed the estimation block.

References

[1] Alexandre Cortiella, Kwang-Chun Park, and Alireza Doostan. Sparse identification of nonlinear dynamical systems via reweighted l1-regularized least squares. *Computer Methods in Applied Mechanics and Engineering*, 376:113620, April 2021.

[2] Kyle Dunlap, Mark Mote, Kaiden Delsing, and Kerianne L. Hobbs. Run time assured reinforcement learning for safe satellite docking. *Journal of Aerospace Information Systems*, 20(1):25–36, 2023.

[3] Kyle Dunlap, David van Wijk, and Kerianne L. Hobbs. Run time assurance for autonomous spacecraft inspection. `https://arxiv.org/abs/2302.02885`, 2023.

[4] P.L. Green, E.J. Cross, and K. Worden. Bayesian system identification of dynamical systems using highly informative training data. *Mechanical Systems and Signal Processing*, 56-57:109–122, 2015.

[5] Jer-Nan Juang, Minh Phan, Lucas G. Horta, and Richard W. Longman. Identification of observer/kalman filter markov parameters - theory and experiments. *Journal of Guidance, Control, and Dynamics*, 16(2):320–329, 1993.

[6] Christopher D. McKinnon and Angela P. Schoellig. Estimating and reacting to forces and torques resulting from common aerodynamic disturbances acting on quadrotors. *Robotics and Autonomous Systems*, 123:103314, 2020.

[7] John D. Schierman, Michael D. DeVore, Nathan D. Richards, and Matthew A. Clark. Runtime assurance for autonomous aerospace systems. *Journal of Guidance, Control, and Dynamics*, 43(12):2205–2217, 2020.

[8] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, pages 153–158, 2000.