Problem Statement of Summer 2021 Project: Bounding the Residual Error for Static Luenberger Observers for Polytopic Systems

Jonas Wagner

2021 July 02

Note: See the (OneNote problem statement page) for additional info. (hopefully that link works... idk how well OneNote will integrate as pdf references or GitHub)

1 Polytopic Systems Background

(A detailed walkthrough is in Appendix A)

1.1 Discrete Time Polytopic Model

A standard DT-Polytopic system will be used in this project, as given as:

$$\begin{cases} x_{k+1} &= \sum_{i=1}^{m} \alpha^i (A_i x_k + B_i u_k) \\ y &= C x_k \end{cases} \tag{1}$$

with state variable $x \in \Re^n$, control input $u \in \Re^p$, and output $y \in \Re^q$ common to all of the m submodels. Each submodel is also associated with state matricies A_i and B_i while the output is calculated from the actual state by matrix C.

The scheduling parameter, $\alpha \in \mathcal{A}$ is unknown and time-varying, with $\mathbf{A}$ defined as:

$$\mathcal{A} = \{\alpha \in \Re^m \mid \sum_{i=1}^{m} \alpha^i = 1,\ \alpha^i \geq 0\ \ \forall\ i \in \{1, 2, \dots, m\}\} \tag{2}$$

1.2 Assumptions

The following assumptions will also be made:

1. A_i is stable $\forall\ i = 1, \dots, m$
2. (A_i, B_i) is a controllable pair $\forall\ i = 1, \dots, m$
3. (A_i, C) is an observable pair $\forall\ i = 1, \dots, m$
4. $\alpha \in \mathcal{A}$ is constant (or at least slowly time-varying)

2 State Observer and Residual Definition

The polytopic system described in (1) for assumed scheduling parameters α^i, a State Observer can be designed to estimate the state of the system from the known inputs and outputs.

2.1 Simple Luenberger Observer

A simple Luenberger Observer for system matrices A, B, and C is defined as

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(C\hat{x}_k - y_k) \tag{3}$$

where $L \in \Re^{n \times q}$ is the Luemburger gain.

2.2 Polytopic System Luenburger Observer

For a Polytopic System given with (1) with known (or estimated) scheduling parameters $\bar{\alpha} \in \mathcal{A}$[1], a Luenberger Observer can be defined by:

$$\hat{x}_{k+1} = \sum_{i=1}^{m} \bar{\alpha}_k^i (A_i \hat{x}_k + Bu_k + L_i(C\hat{x}_k - y_k)) \tag{4}$$

with L_i designed so that $(A_i - L_i C)$ is stable $\forall i = 1 \ldots m$.[2]

2.3 State Estimation Error

In a deterministic system with a selected scheduling parameters $(\hat{\alpha})$ equivalent to the actual scheduling parameters of the system $(\tilde{\alpha})$. The state-estimation error is defined by

$$e_k = x_k - \hat{x}_k \tag{5}$$

where x_k is the actual state and $\hat{x}_k$ is the estimated state.

The estimation error update equation can then be calculated to be:

$$e_{k+1} = \sum_{i=1}^{m} \hat{\alpha}_k^i (A_i + L_i C) e_k + v_k^i \tag{6}$$

where the disturbance term v_k^i is defined by

$$v_k^i = \left(\tilde{\alpha}_k^i - \bar{\alpha}_k^i\right)(A_i x_k + B_i u_k) \tag{7}$$

Question: Since $\bar{\alpha}_k$ will be constant, $\tilde{\alpha}_k - \bar{\alpha}_k$ but does that mean v_k will never decay to zero? and if so, will it at least remain bounded (under certain conditions for A_i and B_i)? (I assume this is a corrallary when finding the residual bounds)

2.4 Output Residual Definition

The measured output $y_k = Cx_k$ and estimated output $\hat{y}_k = C\hat{x}_k$ are used to define the residual, r_k as:

$$r_k = y_k - \hat{y}_k = C(x_k - \hat{x}_k) = Ce_k \tag{8}$$

The output residual update equation can be calculated from (4) and (8) to be:

$$r_{k+1} = \sum_{i=1}^{k} \bar{\alpha}_k^i (A_i + L_i C) r_k + Cv_k \tag{9}$$

[1]which technically may not need to be restricted to be within $\mathcal{A}$

[2]might be useful to also specify L_i specifically based on the LMI from the paper... $L_i = G_i^{-1} F_i$

3 Problem Objectives

0. Simulate using a toy system to gain intuition for bounds on the residual using the simple SISO system w/ a static system scheduling parameter ($\tilde{\alpha}$) and no noise (deterministic).

1. For a simple SISO DT-polytopic system, calculate/place an ellipsoid bound on the residual, assuming $r_k \sim \mathcal{N}(0, \Sigma)$, meaning a test statistic is defined by

$$z_k = r_k^T \Sigma^{-1} r_k \leq z_{threshold}$$

   so that the threshold $z_{threshold}$ (for a deterministic system) can be defined as the reachable residual given $\bar{\alpha} \in \mathcal{A} \neq \tilde{\alpha} \in \mathcal{A}$.

2. Extend to MIMO systems (hopefully this is pretty simple/already incorporated in the previous steps) and define a method to calculate Σ and $z_{threshold}$ (presumably with LMIs and associated definitions to take it to and from the general LMI form).

3. Extend the bound to find the bound for any $\tilde{\alpha} \in \mathcal{A}$ (worst case scenarios for a given $\bar{\alpha}$).

4. Find a way to calculate the minimum bounded region for all $\tilde{\alpha} \in \mathcal{A}$ (idk if it already is or not just w/ it being a feasibility problem) as well as defining the method to select the best $\bar{\alpha}$ to minimize the bounded region.

5. Confirm the analysis with simulations with the toy model, as well as, more interesting higher-order and MIMO systems.

   (a) Test with noise to ensure robustness of the estimates (and potentially robustness to stealthy/unstealthy attacks)

   (b) Mabye: Run a lot of simulations to experimentally find regions where it is vulnerable (i.e. find what is contained within the ellipsoidal bound but not actually reachable)

A In-Depth Polytopic System Backround

Polytopic LPV system models are essentially a smooth interpolation of a set of LTI submodels constructed using a specified weighting function. This can be looked at as decomposing a system into multiple operating spaces that operate as linear submodels. It is possibile for a Polytopic model to take a complex nonlinear model and redefine it as a time-varying interpolation of multiple linear submodels.

Section references:[3] [?] [?] [?]

A.1 General Continuous Time Polytopic Model

The simple polyotopic LPV structure can be described by the following weighted linear combination of LTI submodels:

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} \mu_i(\xi(t))\{A_i x(t) + B_i u(t)\} \\ y(t) = \sum_{i=1}^{r} \mu_i(\xi(t)) C_i x(t) \end{cases} \tag{10}$$

with state variable $x \in \Re^n$ common to all r submodels, control input $u \in \Re^p$, output $y \in \Re^q$, weighting function $\mu_i(\cdot)$ and premise variable $\xi(t) \in \Re^w$.

Additionally, the weighting functions $\mu_i(\cdot)$ for each subsystem must satisfy the convex sum constraints:

$$0 \leq \mu_i(\xi),\ \forall i = 1, \ldots, r \ \text{ and } \sum_{i=1} \mu_i(\xi) = 1 \tag{11}$$

One notable downside, for our application, is the requirement for $\xi(t)$ to be explicitly known in real-time for the model to function. This requirement is the primary driving factor in investigating this system as when $\xi(t)$ is not explicitly known additional uncertainties now exist in a system that are open for exploitation by an attacker.

A.2 Discrete Time Polytopic Model

In the DT-Polytopic Model the CT-Polytopic Model, (10), is extended into the discrete time equivalence (either through sampling and zero-order holds or by definition) by the following parameter-varying system:

$$\begin{cases} x_{k+1} & = \sum_{i=1}^{m} \alpha^i (A_i x_k + B_i u_k) \\ y & = C x_k \end{cases} \tag{12}$$

with state variable $x \in \Re^n$, control input $u \in \Re^p$, and output $y \in \Re^q$ common to all of the m submodels. Each submodel is also associated with state matricies A_i and B_i while the output is calculated from the actual state by matrix C.

The scheduling parameter, $\alpha \in \mathcal{A}$ is unknown and time-varying, with $\mathbf{A}$ defined as:

$$\mathcal{A} = \{\alpha \in \Re^m \mid \sum_{i=1}^{m} \alpha^i = 1,\ \alpha^i \geq 0 \ \ \forall\ i \in \{1, 2, \ldots, m\}\} \tag{13}$$

In the discrete time case, the unknown scheduling parameter, α, is problematic for when developing a state-estimator, thus a Joint State-Parameter estimator must be used. The discrete nature of the measurements may also prove to be even more problematic if an attack is injected in any discrete measurement.

[3] Each subsection is mostly a summary of sections from these sources but with elaboration and consistent notation.

A.3 MATLAB

All code I wrote for this project can be found on my GitHub repository:
https://github.com/jonaswagner2826/DT_LPV_attack_analysis