Problem Statement:
Detector Designed from Vertices of Polytopic System

Jonas Wagner

2021 September 21

1 System Definition

1.1 Plant Definition

The system to be controlled (the plant) will be defined with a standard LTI system:

$$\mathcal{S}_{plant} := \begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k + Du_k \end{cases} \tag{1}$$

where actual state $x_k \in \Re^n$, control input $u_k \in \Re^p$, and output $y_k \in \Re^q$. The state matrices A, B, C, and D fully define the dynamics of the system.

1.2 System Uncertainty

It is known that A is within a polyotopic set of state matrices, $\{A_i : \forall i = 1, \dots, m\}$, calculated as $A = A(\alpha)$.

$$A(\alpha) := \sum_i^m \alpha^{(i)} A_i \tag{2}$$

where $\alpha \in \mathcal{A}$

$$\mathcal{A} = \left\{ \alpha \in \Re^m \ : \ \sum_{i=1}^m \alpha^{(i)} = 1,\ \alpha^{(i)} \geq 0 \ \forall \ i \in \{1, 2, \dots, m\} \right\} \tag{3}$$

The following assumptions are also known about the system:

Assumption 1. (A_i, B) *is controllable* $\forall i = \{1, \dots, m\}$

Assumption 2. (A_i, C) *is observable* $\forall i = \{1, \dots, m\}$

1.3 Individual Subsystems

Each individual subsystem can be considered individually with

$$\mathcal{S}^{(i)} := \begin{cases} x_{k+1}^{(i)} = A_i x_k^{(i)} + Bu_k \\ y_k^{(i)} = Cx_k^{(i)} + Du_k \end{cases} \tag{4}$$

where subsystem state $x_k^{(i)} \in \Re^n$ and estimated output $y_k^{(i)} \in \Re^q$.

1.4 State Observer

A state observer is designed using a simple Luemburger observer:

$$\mathcal{S}_{obsv} := \begin{cases} \hat{x}_{k+1} = \hat{A}\hat{x}_k + Bu_k + L(y_k - \hat{y}_k) \\ \hat{y}_k = C\hat{x}_k + Du_k \end{cases} \tag{5}$$

where estimated $\hat{x}_k \in \Re^n$ and estimated output $\hat{y}_k \in \Re^q$.
The estimated state matrix $\hat{A}$ is calculated as $\hat{A} = A(\hat{\alpha})$ using the estimated parameter $\hat{\alpha} \in \mathcal{A}$.

1.5 Observer Gain Calcualtion

The observer gain matrix L is designed so that regardless of the actual system matrix the observer is stable:

$$L \in \left\{ L \in \Re^{n \times q} \mid (A(\alpha) - LC) \text{ stable } \forall \alpha \in \mathcal{A} \right\} \tag{6}$$

Alternativly, using polytopic methods, L can be defined to satisfy each of the polytopic vertices and therefore satisfy it for all potential matrices in the polytope.

Theorem 1. *The feasable region of L is when the following LMIs as satisfied:*

$$\begin{bmatrix} Q & (QA_i + XC)^T \\ (QA_i + XC) & Q \end{bmatrix} \succeq 0, \ \forall i = 1, \dots, m \tag{7}$$

where $Q \in \{Q \in \Re^{n \times n} \mid Q \succeq 0\}$, $X \in \Re^{n \times p}$, and L is calculated as $L = Q^{-1}X$.

Proof. **Include this proof... nah... mabye just reference boyd (or include anyway)** □

1.6 Controller?

Design a controller into it? Is it actually needed for this or can the $A = A + BK$, $A_i = A_i + Bk$, and $\hat{A} \approx (A + BK)$ with just 'no loss of generality'?
My guess is yes becouse although the controller may be affected, the input and measured output are knonw so (when proved for $A, A_i, \hat{A}$) it imnplies generality

2 Residual Bounding

2.1 Residual Definition

The residual, r_k, is defined by

$$r_k := y_k - \hat{y}_k \tag{8}$$

Additionally, we define associated residuals comparing the observer to each of the individual subsystems, $r_k^{(i)}$, as

$$r_k^{(i)} := y_k^{(i)} - \hat{y}_k \tag{9}$$

2.2 Test Statistic

A test statistic, z_k, is then defined as

$$z_k := r_k^T \Sigma^{-1} r_k \tag{10}$$

where Σ is designed to be the covariance matrix of the expected residual. Similarly, the test statistic for each subsystem residuals is defined as

$$z_k^{(i)} = (r_k^{(i)})^T \Sigma^{-1} r_k^{(i)} \tag{11}$$

A maximum test statisic, z_k^*, can then be calculated as

$$z_k^* := \max_i z_k^{(i)} \tag{12}$$

2.3 Statistic Bounding

Theorem 2. *The actual system test statistic, z_k, is bounded from above by the maximum test statistic for each of the other subsystem, z_k^*.*

$$z_k \leq z_k^*, \ \forall k \geq 0 \tag{13}$$

Proof. **Put the proof of this thing....**
Should be based on the explicit def of it... □

3 Detector Design

A detector will be designed to compare the residual potentially caused from the observer to simulated systems at each vertice of the polytope.

3.1 Detector Alarm

The detector then sounds an alarm acording to the following rule:

$$\begin{cases} z_k < z_k^* & \text{no alarm} \\ z_k \geq z_k^* & \text{alarm} \end{cases} \tag{14}$$

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:[1]

A.1 General Continuous Time Polytopic Model

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

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{r} \mu_i(\xi(t))\{A_i x(t) + B_i u(t)\} \\ y(t) = \sum_{i=1}^{r} \mu_i(\xi(t)) C_i x(t) \end{cases} \tag{15}$$

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:

$$0 \leq \mu_i(\xi), \ \forall i = 1, \ldots, r \quad \text{and} \quad \sum_{i=1} \mu_i(\xi) = 1 \tag{16}$$

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, (15), is extended into the discrete time equivalence (either through sampling and zero-order holds or by definition) by the following parameter-varying system:

$$\begin{cases} x_{k+1} &= \sum_{i=1}^{m} \alpha^i (A_i x_k + B_i u_k) \\ y &= C x_k \end{cases} \tag{17}$$

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:

$$\mathcal{A} = \{\alpha \in \Re^m \mid \sum_{i=1}^{m} \alpha^i = 1, \ \alpha^i \geq 0 \ \ \forall \ i \in \{1, 2, \ldots, m\}\} \tag{18}$$

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.

[1]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/polytopic-system-security