# Documentation:
# Policy synthesis via formal abstraction

December 7, 2017

# Contents

# Chapter 1

# Do abstraction of LTI system

## 1.1 Computation of simulation relation

Define LTI system as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + Du_k + v_k \end{aligned} \tag{1.1}$$

with

- $x$ state of size $n$
- $u$ input of size $m$
- $A$ matrix of size $n \times n$
- $B$ matrix of size $n \times m$
- $y$ the output (used to compare accuracy)
- $C$ output matrix of size $q \times n$
- $D$ matrix currently assumed to be zero

$$x_+ - \tilde{x}_+ = (A + BK)(x - \tilde{x}) + \mathbf{r} \tag{1.2}$$

with

- $\mathbf{r}$ in a polytope, i.e., $\mathbf{r} \in \mathcal{V}(r_i)$, the polytope generated from vertices $r_i$.

Consider a set defined as

$$\mathcal{R} := \{(\tilde{x}, x) \mid (x - \tilde{x})^T M (x - \tilde{x}) \leq \epsilon^2\} \tag{1.3}$$

**Objective:** Design $M$, $K$ and $\epsilon$ such that if $(\tilde{x}, x) \in \mathcal{R}$ then also

$$\{(x_+ - \tilde{x}_+) | \text{ s.t. (1.2) } \forall \mathbf{r} \in \mathcal{V}(r_i)\} \subseteq \mathcal{R}.$$

More over for all $(\tilde{x}, x) \in \mathcal{R}$ it should hold that $d(\tilde{y}, y) \leq \epsilon$. The latter can be expressed as $C^T C \preceq M$. The former can be written with matrix inequalities as

$$
\begin{aligned}
(x_+ - \tilde{x}_+)^T M (x_+ - \tilde{x}_+) &\leq \epsilon^2 \\
((A+BK)(x-\tilde{x}) + \mathbf{r})^T M ((A+BK)(x-\tilde{x}) + \mathbf{r}) &\leq \epsilon^2
\end{aligned}
$$

Hence we get something of this form

$$
(x-\tilde{x})^T M (x-\tilde{x}) \leq \epsilon^2 \implies ((A+BK)(x-\tilde{x}) + \mathbf{r})^T M ((A+BK)(x-\tilde{x}) + \mathbf{r}) \leq \epsilon^2
$$

**S-procedure**[a]

The implications

$$
x^T F_1 x + 2 g_1^T x + h_1 \leq 0 \implies x^T F_2 x + 2 g_2^T x + h_2 \leq 0 \tag{1.4}
$$

holds if and only if there exists $\lambda \geq 0$ such that

$$
\lambda \begin{bmatrix} F_1 & g_1 \\ g_1^T & h_1 \end{bmatrix} - \begin{bmatrix} F_2 & g_2 \\ g_2^T & h_2 \end{bmatrix} \succeq 0 \tag{1.5}
$$

[a] https://en.wikipedia.org/wiki/S-procedure

Using the S-procedure we get

$$
(x-\tilde{x})^T (A+BK)^T M (A+BK)(x-\tilde{x}) + 2\mathbf{r}^T M (A+BK)(x-\tilde{x}) + \mathbf{r}^T M \mathbf{r} \leq \epsilon^2 \tag{1.6}
$$

$$
\lambda \begin{bmatrix} M & 0 \\ 0 & -\epsilon^2 \end{bmatrix} - \begin{bmatrix} (A+BK)^T M (A+BK) & (A+BK)^T M \mathbf{r} \\ \mathbf{r}^T M (A+BK) & \mathbf{r}^T M \mathbf{r} - \epsilon^2 \end{bmatrix} \succeq 0 \tag{1.7}
$$

$$
\begin{bmatrix} \lambda M - ((A+BK)^T M (A+BK)) & -(A+BK)^T M \mathbf{r} \\ -\mathbf{r}^T M (A+BK) & (1-\lambda)\epsilon^2 - \mathbf{r}^T M \mathbf{r} \end{bmatrix} \succeq 0 \tag{1.8}
$$

$$
\begin{bmatrix} \lambda M & 0 \\ 0 & (1-\lambda)\epsilon^2 \end{bmatrix} - \begin{bmatrix} ((A+BK)^T M (A+BK)) & (A+BK)^T M \mathbf{r} \\ \mathbf{r}^T M (A+BK) & \mathbf{r}^T M \mathbf{r} \end{bmatrix} \succeq 0 \tag{1.9}
$$

$$
\begin{bmatrix} \lambda M & 0 \\ 0 & (1-\lambda)\epsilon^2 \end{bmatrix} - \begin{bmatrix} (A+BK)^T M \\ \mathbf{r}^T M \end{bmatrix} M^{-1} \begin{bmatrix} (A+BK)^T M \\ \mathbf{r}^T M \end{bmatrix}^T \succeq 0 \tag{1.10}
$$

$$
\begin{bmatrix} \lambda M & 0 & (A+BK)^T M \\ 0 & (1-\lambda)\epsilon^2 & \mathbf{r}^T M \\ M(A+BK) & M\mathbf{r} & M \end{bmatrix} \succeq 0 \tag{1.11}
$$

$$
\begin{bmatrix} \lambda M^{-1} & 0 & M^{-1}(A+BK)^T \\ 0 & (1-\lambda)\epsilon^2 & \mathbf{r}^T \\ (A+BK)M^{-1} & \mathbf{r} & M^{-1} \end{bmatrix} \succeq 0 \tag{1.12}
$$

$$
\begin{bmatrix} \lambda M^{-1} & 0 & M^{-1}(A+BK)^T \\ 0 & (1-\lambda)\epsilon^2 & r_i^T \\ (A+BK)M^{-1} & r_i & M^{-1} \end{bmatrix} \succeq 0, \ \forall r_i \tag{1.13}
$$

$$
\tag{1.14}
$$

Remark that this implies that $1-\lambda \geq 0$ hence $1 \geq \lambda \geq 0$. And remark that

The objective to find a minimal $\epsilon$ can be expressed as follows

$$
\textbf{Objective}: \min_{M_{inv}, L} \epsilon^2 \tag{1.15}
$$

$$
\begin{bmatrix} \lambda M_{inv} & 0 & M_{inv}A^T + L^T B^T \\ 0 & (1-\lambda)\epsilon^2 & r_i^T \\ AM_{inv} + BL & r_i & M_{inv} \end{bmatrix} \succeq 0 \tag{1.16}
$$

$$
\begin{bmatrix} M_{inv} & M_{inv}C^T \\ CM_{inv} & I \end{bmatrix} \succeq 0 \tag{1.17}
$$

with $LM = K$ and $M^{-1} = M_{inv}$. This has been implemented as function $eps\_err()$ in python.

### 1.1.1 Verify that Polytope $\mathcal{V}(r_i)$ is in relation.

$\mathcal{V}(r_i)$ is in relation $\mathcal{R} := \{(\tilde{x}, x) \mid (x - \tilde{x})^T M (x - \tilde{x}) \leq \epsilon^2\}$ if for all $r_i$ it holds that

$$r_i^T M r_i \leq \epsilon^2$$

.

### 1.1.2 Plot simulation relation

Input

$$\mathcal{R} := \{x \mid x^T M_\epsilon x \leq 1\}$$

**Algorithm:**

1. Compute $M_\epsilon^{1/2} = U\Sigma^{1/2}$ with singular value decomposition $M_\epsilon = U\Sigma V^T$
2. Switch variable
$$\mathcal{R} := \{(\tilde{x}, x) \mid z^T z \leq 1 \text{ with } z = M_\epsilon^{1/2} x\}$$
3. compute outline given angle $\alpha$
$$z(\alpha) = \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}$$
remark $z^T z = 1$. then $x(\alpha) = \Sigma^{-1/2} U^T z(\alpha)$.