Lecture 6

12. Projection Operator

Definition A.1: Subset $\Omega \subset R^n$ is convex if

$$\left[\forall x, y \in \Omega \subset R^n\right] \Rightarrow \left[\lambda x + (1-\lambda) y = z \in \Omega\right], \quad \forall 0 \le \lambda \le 1 \tag{12.1}$$

Relation (12.1) states that if two points belong to the convex subset Ω then all the points on the connecting line also belong to Ω.

Definition A.2: Function $f : R^n \to R$ is convex if

$$f\left(\lambda x + (1-\lambda) y\right) \le \lambda f(x) + (1-\lambda) f(y), \quad \forall 0 \le \lambda \le 1 \tag{12.2}$$

Inequality (12.2) is illustrated on Figure 12.1, and it states that graph of a convex function must be located below the straight line, which connects any two corresponding function values.

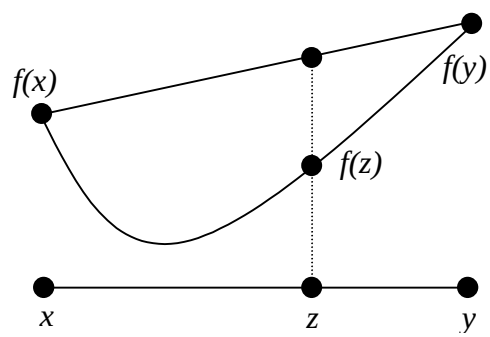

Figure 12.1: A Convex Function

Statement A.1: Let $f(x): R^n \to R$ be convex function. Then for any constant $\delta > 0$ the subset $\Omega_\delta = \left\{\theta \in R^n \,\middle|\, f(\theta) \le \delta\right\}$ is convex.

Proof: Let $\theta_1, \theta_2 \in \Omega_\delta$. Then $f(\theta_1) \le \delta$ and $f(\theta_2) \le \delta$. Since $f(x)$ is convex then for any $0 \le \lambda \le 1$

$$f\left(\underbrace{\lambda \theta_1 + (1-\lambda)\theta_2}_{\theta}\right) \le \lambda \underbrace{f(\theta_1)}_{\le\delta} + (1-\lambda) \underbrace{f(\theta_2)}_{\le\delta} \le \lambda \delta + (1-\lambda)\delta = \delta$$

Therefore $f(\theta) \le \delta$ and, consequently, $\theta \in \Omega_\delta$ which completes the proof. □

Statement A.2: Let $f(x): R^n \to R$ be continuously differentiable convex function. Choose a constant $\delta > 0$ and consider the subset $\Omega_\delta = \{\theta \in R^n \mid f(\theta) \le \delta\} \subset R^n$. Let $\theta^* \in \Omega_\delta$ and assume that $f(\theta^*) < \delta$, (i.e., θ^* is <u>*not*</u> on the boundary of Ω_δ). Also, let $\theta \in \Omega_\delta$ and assume that $f(\theta) = \delta$, (i.e., (θ <u>*is*</u> on the boundary of Ω_δ). Then the following inequality takes place:

$$(\theta^* - \theta)^T \nabla f(\theta) \le 0 \tag{12.3}$$

where $\nabla f(\theta) = \begin{pmatrix} \frac{\partial f(\theta)}{\partial \theta_1} & \dots & \frac{\partial f(\theta)}{\partial \theta_n} \end{pmatrix}^T \in R^n$ is the gradient vector of f evaluated at θ.

Relation (12.3) is illustrated on Figure A.2. It shows that the gradient vector evaluated at the boundary of a convex set always points away from the set.

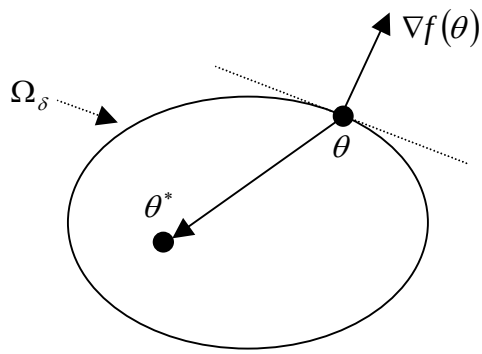

Figure A.2: Gradient and Convex Set

<u>*Proof*</u>: Since $f(x)$ is convex then

$$f(\lambda \theta^* + (1-\lambda)\theta) \le \lambda f(\theta^*) + (1-\lambda) f(\theta)$$

or equivalently:

$$f(\theta + \lambda(\theta^* - \theta)) \le f(\theta) + \lambda(f(\theta^*) - f(\theta))$$

Then for any nonzero $0 < \lambda \le 1$:

$$\frac{f(\theta + \lambda(\theta^* - \theta)) - f(\theta)}{\lambda} \le f(\theta^*) - f(\theta) < \delta - \delta = 0$$

Taking the limit as $\lambda \to 0$ yields relation (12.3) and completes the proof. □

Suppose that θ, the "true" parameter vector, belongs to a <u>*convex*</u> set Ω_0

$$\Omega_0 = \{\theta \in R^n \mid f(\theta) \le 0\} \tag{12.4}$$

Introduce another convex set:

$$\Omega_1 = \{\theta \in R^n \mid f(\theta) \le 1\} \tag{12.5}$$

It is obvious that $\Omega_0 \in \Omega_1$. We may now define the <u>*projection operator*</u>, which we shall use in the adaptive laws.

$$\mathrm{Proj}(\theta, y) = \begin{cases} y, & \text{if } f(\theta) \le 0 \\ y, & \text{if } f(\theta) \ge 0 \text{ and } y^T \nabla f(\theta) \le 0 \\ y - \dfrac{\nabla f(\theta)(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2} y\, f(\theta), & \text{if not.} \end{cases}$$

or equivalently:

$$\mathrm{Proj}(\theta, y) = \begin{cases} y - \dfrac{\nabla f(\theta)(\nabla f(\theta))^T}{\|\nabla f(\theta)\|^2} y\, f(\theta), & \text{if } f(\theta) > 0 \text{ and } y^T \nabla f(\theta) > 0 \\ y, & \text{if not} \end{cases} \tag{12.6}$$

Namely, $\mathrm{Proj}(\theta, y)$ does not alter the vector y if θ belongs to the convex set Ω_0 defined in (12.4). In the set $\{0 \le f(\theta) \le 1\}$, the projection operator subtracts a vector normal to the boundary $\{f(\theta) = \lambda\}$ from y so that we get a smooth transformation from the original vector field y for $\lambda = 0$ to a tangent to the boundary vector field for $\lambda = 1$. The projection operator concept is illustrated on Figure 12.3.

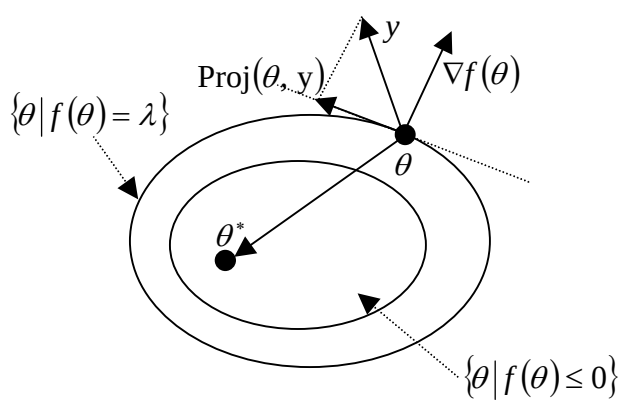

Figure 12.3: Projection Operator

Using Statement A.2 and inequality (12.3), we get the following important property of the projection operator:

$$\left(\theta^* - \theta\right)^T \left(y - \mathrm{Proj}\left(\theta, y\right)\right) = \left\{ \begin{array}{l} 0, \quad \text{if } f\left(\theta\right) \le 0 \\ 0, \quad \text{if } f\left(\theta\right) \ge 0 \text{ and } y^T \nabla f\left(\theta\right) \le 0 \\ \dfrac{\underbrace{\left(\theta^* - \theta\right)^T \nabla f\left(\theta\right)}_{\le 0} \underbrace{\left(\nabla f\left(\theta\right)\right)^T y}_{\ge 0}}{\left\|\nabla f\left(\theta\right)\right\|^2} \underbrace{f\left(\theta\right)}_{=\lambda \ge 0}, \quad \text{if not.} \end{array} \right\} \le 0 \qquad (12.7)$$

or, equivalently

$$\left(\theta - \theta^*\right)^T \left(\mathrm{Proj}\left(\theta, y\right) - y\right) \le 0 \qquad (12.8)$$

Based on (12.6), we can now define the projection operator when both Y and Θ are matrices of the same dimensions:

$$Y = \left(y_1 \quad \ldots \quad y_N\right) \in R^{n \times N} \text{ and } \Theta = \left(\theta_1 \quad \ldots \quad \theta_N\right) \in R^{n \times N} \qquad (12.9)$$

$$\mathrm{Proj}\left(\Theta, Y\right) = \left(\mathrm{Proj}\left(\theta_1, y_1\right) \quad \ldots \quad \mathrm{Proj}\left(\theta_N, y_N\right)\right) \qquad (12.10)$$

Relation (12.10) implies that for matrices the projection operator is defined column-wise. Finally, we show that the trace terms in the Lyapunov function derivative (11.16) become semi-negative due to the usage of the Projection Operator in forming the adaptive laws (11.17). Since all the trace terms in (11.16) are similar, we only consider the first one.

$$\mathrm{tr}\left(\underbrace{\Delta K_x^T}_{\left(\hat{K}_x - \hat{K}_x^*\right)^T} \left[\underbrace{\Gamma_x^{-1} \dot{\hat{K}}_x}_{\mathrm{Proj}\left(\hat{K}_x, Y\right)} + \underbrace{x e^T P B}_{-Y} \right] \Lambda \right) = \sum_{j=1}^{m} \underbrace{\left(\hat{K}_x - \hat{K}_x^*\right)_j^T \left(\mathrm{Proj}\left(\hat{K}_x, Y_j\right) - Y_j\right)}_{\le 0} \underbrace{\lambda_j}_{\ge 0} \le 0 \qquad (12.11)$$

Using (12.11) yields the same adaptation law for $\hat{K}_x$ as in (11.17), namely:

$$\dot{\hat{K}}_x = \Gamma_x \, \mathrm{Proj}\left(\hat{K}_x, -x e^T P B\right) \qquad (12.12)$$

Basically, Projection Operator ensures that the columns $\left(\hat{K}_x\right)_j$ of the adaptive parameter matrix $\hat{K}_x$ do not exceed their pre-specified bounds $\left(\hat{K}_x^{\max}\right)_j$, and at the same time the operator contributes to the negative semi-definiteness of the Lyapunov function (11.16).

Next we show how to define convex function $f(\bullet) = \begin{pmatrix} f_1 & \dots & f_m \end{pmatrix}^T$ and m – convex sets $\left\{\Omega_\delta^j\right\}_{j=1,\dots,m}$. Both the function and the set definitions are based on the desired upper bounds $\left\|\left(\hat{K}_x\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j$ that are imposed column-wise. For a jth column $\left(\hat{K}_x\right)_j$ of the adaptive parameter matrix $\hat{K}_x \in R^{n\times m}$, choose <u>*projection tolerance*</u> $\varepsilon_j^x > 0$ and define $f_j(\bullet)$ as in (11.20):

$$f_j = f\left(\left(\hat{K}_x\right)_j\right) = \frac{\left\|\left(\hat{K}_x\right)_j\right\|^2 - \left(\hat{K}_x^{\max}\right)_j^2}{\varepsilon_j^x \left(\hat{K}_x^{\max}\right)_j^2} \tag{12.13}$$

Using (12.13), the sets Ω_δ^j are defined as:

$$\Omega_\delta^j = \left\{\left(\hat{K}_x\right)_j \in R^{n\times m} \,\middle|\, f_j \le \delta_j\right\} \tag{12.14}$$

From (12.14) it follows that for each $j = 1,\dots,m$:

$$\begin{aligned} \Omega_0^j &= \left\{\left(\hat{K}_x\right)_j \in R^{n\times m} \,\middle|\, \left\|\left(\hat{K}_x\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j\right\} \\ \Omega_1^j &= \left\{\left(\hat{K}_x\right)_j \in R^{n\times m} \,\middle|\, \left\|\left(\hat{K}_x\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j \sqrt{1+\varepsilon_j^x}\right\} \end{aligned} \tag{12.15}$$

We need to compute the gradient of the convex function (12.13).

$$\nabla f_j = \frac{1}{\varepsilon_j^x \left(\hat{K}_x^{\max}\right)_j^2} \nabla\left[\left\|\left(\hat{K}_x\right)_j\right\|^2\right] = \frac{2}{\varepsilon_j^x \left(\hat{K}_x^{\max}\right)_j^2}\left(\hat{K}_x\right)_j \tag{12.16}$$

Using (12.12), the adaptive law for $\hat{K}_x$ can be written column-wise as:

$$\left(\dot{\hat{K}}_x\right)_j = \Gamma_x \begin{cases} -\left(x e^T P B\right)_j + \dfrac{\nabla f_j \nabla f_j^T}{\left\|\nabla f_j\right\|^2}\left(x e^T P B\right)_j f_j, & \text{if } \begin{cases} f_j > 0 \\ \left(x e^T P B\right)_j^T \nabla f_j < 0 \end{cases} \\ -\left(x e^T P B\right)_j, & \text{if not} \end{cases} \tag{12.17}$$

The adaptation process in (12.17) ensures uniform boundedness of each column of the adaptive parameter matrix $\hat{K}_x$ forward in time, that is:

$$\left\{\left\|\left(\hat{K}_x(0)\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j \Rightarrow \left\{\left\|\left(\hat{K}_x(t)\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j \sqrt{\left(1+\varepsilon_j^x\right)}, \quad \forall t \ge 0, \quad 1 \le j \le m\right\}\right. \tag{12.18}$$

The rest of the adaptive parameters in (11.17) are defined in a similar fashion.

13. Projection Operator based MRAC Design for MIMO Systems with Unstructured Uncertainties

Using the adaptive laws (11.17), it is easy to see that the derivative of the Lyapunov function (11.11) satisfies the following inequality:

$$\begin{aligned}\dot{V} &= -e^T Q e + 2 e^T P B \Lambda \varepsilon(x) \le -\lambda_{\min}(Q)\|e\|^2 + 2\|P B \Lambda\| \varepsilon_{\max} \|e\| \\ &= -\|e\|\left(\lambda_{\min}(Q)\|e\| - 2\|P B \Lambda\| \varepsilon_{\max}\right)\end{aligned} \tag{13.1}$$

At the same time, due to Projection Operator properties, all the adaptive parameters are UUB. Consequently, using Theorem 5.1, we conclude that the trajectories of the closed-loop system (11.1)-(11.5)-(11.17) are UUB. Moreover, the tracking error $e = x - x_{ref}$ enters a neighborhood of the origin, within a finite time. The radius of the neighborhood (i.e., the ultimate bound) is determined by the minimum level set of the Lyapunov function V which contains the set E where $\dot{V} \le 0$:

$$\begin{aligned}E = &\left\{e \in R^n : \|e\| \le \frac{2\|P B \Lambda\| \varepsilon_{\max}}{\lambda_{\min}(Q)}\right\} \\ &\times \left\{\hat{K}_x \in R^{n\times m} : \left\|\left(\hat{K}_x\right)_j\right\| \le \left(\hat{K}_x^{\max}\right)_j \sqrt{\left(1+\varepsilon_j^x\right)}, \quad 1 \le j \le m\right\} \\ &\times \left\{\hat{K}_r \in R^{m\times m} : \left\|\left(\hat{K}_m\right)_j\right\| \le \left(\hat{K}_r^{\max}\right)_j \sqrt{\left(1+\varepsilon_j^r\right)}, \quad 1 \le j \le m\right\} \\ &\times \left\{\hat{\Theta} \in R^{N\times m} : \left\|\left(\hat{\theta}\right)_j\right\| \le \left(\hat{\theta}^{\max}\right)_j \sqrt{\left(1+\varepsilon_j^{\Theta}\right)}, \quad 1 \le j \le m\right\}\end{aligned} \tag{13.2}$$

14. Adaptive Augmentation of a Baseline Controller

Consider MIMO plant dynamics:

$$\dot{x}_p = A_p x_p + B_p \Lambda\left(u + f\left(x_p\right)\right) \tag{14.1}$$

where $x_p \in R^n$ is the *n*-dimensional system state vector, $u \in R^m$ is the *m*-dimensional vector of *independent* (virtual) control inputs, A_p is the (*n*-by-*n*) – <u>*known*</u> constant matrix, B_p is the (*n*-by-*m*) – <u>*known*</u> constant matrix, Λ is the (*m*-by-*m*) – <u>*unknown*</u> constant <u>*diagonal*</u> matrix with positive diagonal elements, and $f\left(x_p\right)$ is the (*m*-by-1) – <u>*unknown*</u> possibly nonlinear state-dependent vector.

Assuming no uncertainties in the model, that is $\Lambda = I_{m\times m}$, $f\left(x_p\right) = 0_{m\times 1}$, a <u>*baseline (nominal) linear dynamic controller*</u> can be designed to provide command tracking. The controller dynamics can be written in the form:

$$\dot{x}_c = A_c\, x_c + B_{1c}\, x_p + B_{2c}\, r \tag{14.2}$$

where $x_c \in R^{n_c}$ is the n_c – dimensional controller state vector, $r \in R^m$ is the m – dimensional reference (desired) command signal. The corresponding augmented plant-controller system becomes:

$$\underbrace{\begin{pmatrix} \dot{x}_p \\ \dot{x}_c \end{pmatrix}}_{\tilde{x}} = \underbrace{\begin{pmatrix} A_p & 0 \\ B_{1c} & A_c \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} x_p \\ x_c \end{pmatrix}}_{x} + \underbrace{\begin{pmatrix} B_p \\ 0 \end{pmatrix}}_{B_1} \Lambda\left(u + f\left(x_p\right)\right) + \underbrace{\begin{pmatrix} 0 \\ B_{2c} \end{pmatrix}}_{B_2} r \tag{14.3}$$

The <u>*baseline controller*</u> is defined as:

$$u_{bl} = K_x^T\, x + K_r^T\, r \tag{14.4}$$

where $K_x \in R^{n\times m}$ and $K_r \in R^{m\times m}$ are the feedback and feedforward nominal gain matrices, correspondingly, with $n = n_p + n_c$.

The <u>*reference model dynamics*</u> is chosen as:

$$\dot{x}_{ref} = A\, x_{ref} + B_2\, r + B_1\, u_{bl} \tag{14.5}$$

or equivalently

$$\dot{x}_{ref} = \underbrace{\left(A + B_1\, K_x^T\right)}_{A_{ref}} x_{ref} + \underbrace{\left(B_2 + B_1\, K_r^T\right)}_{B_{ref}} r \tag{14.6}$$

where

$$A_{ref} = \begin{pmatrix} A_p & 0 \\ B_{1c} & A_c \end{pmatrix} \tag{14.7}$$

It is assumed that the baseline controller is designed such that A_{ref} is Hurwitz.

Assumption 14.1: If $\Lambda = I_{m\times m}$ and $f\left(x_p\right) = 0_{m\times 1}$ then the baseline control feedback u^{bl} in (14.4) yields asymptotic tracking, i.e., $\lim_{t\to\infty}\left\| e(t)\right\| = 0$.

<u>*The control objective*</u> is to find u in (14.1) such that the state x of the augmented system (14.3) tracks the state x_{ref} of the reference model (14.6) in the presence of the system uncertainties Λ and $f\left(x_p\right)$, while all signals in the closed-loop system remain bounded.

To this end, we introduce the <u>*tracking error vector*</u>:

$$e = x - x_{ref} \tag{14.8}$$

<u>*Total*</u> control input is defined as adaptive augmentation of the baseline controller:

$$u = \underbrace{K_x^T\, x + K_r^T\, r}_{u_{bl}} + \underbrace{\hat{K}_x^T\, x + \hat{K}_r^T\, r - \hat{f}\left(x_p\right)}_{u_{ad}} = u_{bl} + u_{ad} \tag{14.9}$$

where $\hat{K}_x \in R^{n\times m}$ and $\hat{K}_r \in R^{n\times m}$ are the *incremental* adaptive gains and $\hat{f}\left(x_p\right) \in R^{m\times 1}$ is the estimated uncertainty. Substituting (14.9) into (14.3), yields:

$$\dot{x} = \underbrace{\left(A + B_1\,\Lambda\left(K_x + \hat{K}_x\right)^T\right)}_{\hat{A}}\tilde{x} + \underbrace{\left(B_2 + B_1\,\Lambda\left(K_r + \hat{K}_r\right)^T\right)}_{\hat{B}} r - B_1\,\Lambda\left(\underbrace{\hat{f}\left(x_p\right) - f\left(x_p\right)}_{\Delta f\left(x_p\right)}\right) \tag{14.10}$$

Assumption 14.2: Given a constant diagonal matrix Λ, there exist "ideal" matrix gains $K_x^* \in R^{n\times m}$ and $K_r^* \in R^{m\times m}$ such that the *uncertainty matching conditions* take place:

$$\begin{cases} A_{ref} = A + B_1\,\Lambda\left(K_x^*\right)^T \\ B_{ref} = B_2 + B_1\,\Lambda\left(K_r^*\right)^T \end{cases} \tag{14.11}$$

Note that the knowledge of the true gains will not be required. From (14.10) - (14.11), it immediately follows that: