11. Adaptation algorithms with improved parametric convergence

1. Motivation

Consider the adaptation algorithm

$$\dot{\hat{\theta}} = \gamma\omega\varepsilon \tag{11.1}$$

where $\gamma > 0$ is the adaptation gain, $\omega \in \mathbb{R}^m$ is the regressor,

$$\varepsilon = y - \hat{\theta}^T\omega \tag{11.2}$$

is the signal error (e.g., error of identification or control),

$$y = \theta^T\omega + \sigma \tag{11.3}$$

is the output of the linear regression, $\hat{\theta}$ is the vector of adjustable parameters (or estimates), θ is the vector of unknown parameters, σ is an exponentially decaying term.

11. Adaptation algorithms with improved parametric convergence

1. Motivation

Consider the adaptation algorithm

$$\dot{\hat{\theta}} = \gamma \omega \varepsilon \tag{11.1}$$

where $\gamma > 0$ is the adaptation gain, $\omega \in \mathbb{R}^m$ is the regressor,

$$\varepsilon = y - \hat{\theta}^T \omega \tag{11.2}$$

is the signal error (e.g., error of identification or control),

$$y = \theta^T \omega + \sigma = \varepsilon + \hat{\theta}^T \omega \tag{11.3}$$

is the output of the linear regression, $\hat{\theta}$ is the vector of adjustable parameters (or estimates), θ is the vector of unknown parameters, σ is an exponentially decaying term.

σ **will be omitted.**

1. Motivation

Properties

1. The vector of parametric errors $\tilde{\theta}(t)$ is bounded.

   If $\omega(t), \dot{\omega}(t)$ are bounded, then $\varepsilon(t), \dot{\hat{\theta}}(t)$ are bounded;

2. If $\omega(t), \dot{\omega}(t)$ are bounded, then $\varepsilon(t)$ tends to zero asymptotically as $t \to 0$;

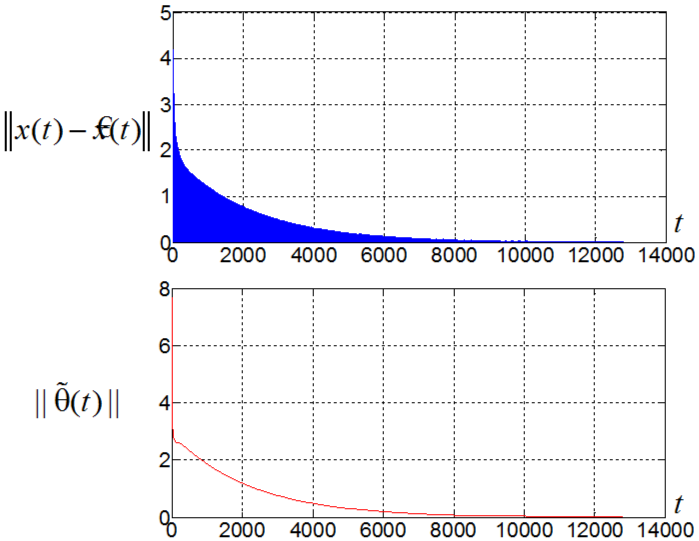
3. $\left\|\tilde{\theta}(t)\right\|$ approaches zero exponentially fast iff $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0 \qquad (11.4)$$

for some positive α, T;

4. If $\omega \in PE$, then there exists an optimal gain γ, for which the rate of parametric convergence is maximum.

The fastest rate of tuning may be arbitrarily slow!



See adaptive observer simulation (Lecture 8) 7000-9000sec

4. If $\omega \in PE$, then there exists an optimal gain γ, for which the rate of parametric convergence is maximum.

2. Kreisselmeier scheme

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Regressor extension:

$$y = \theta^T \omega \tag{11.5}$$

2. Kreisselmeier scheme

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Regressor extension:

$$y = \theta^T \omega \tag{11.5}$$

$$\Downarrow$$

$$\omega y = \omega\omega^T \theta$$

2. Kreisselmeier scheme

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Regressor extension:

$$y = \theta^T \omega \tag{11.5}$$

$$\Downarrow$$

$$\omega y = \omega \omega^T \theta$$

$$L(s)[\omega y] = L(s)\left[\omega \omega^T\right]\theta$$

$$\Downarrow$$

Select and apply a transfer function operator

$L(s)$ must be at least stable and minimum phase

2. Kreisselmeier scheme

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Regressor extension:

$$y = \theta^T \omega \tag{11.5}$$

$$\Downarrow$$

$$\omega y = \omega\omega^T \theta$$

$$\underbrace{L(s)\left[\omega y\right]}_{Y} = \underbrace{L(s)\left[\omega\omega^T\right]}_{\Omega}\theta$$

Select and apply a transfer function operator

$L(s)$ must be at least stable and minimum phase

2. Kreisselmeier scheme

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Regressor extension:

$$y = \theta^T \omega \tag{11.5}$$

$$\Downarrow$$

$$\omega y = \omega \omega^T \theta$$

Select and apply a transfer function operator

$$\underbrace{L(s)[\omega y]}_{Y} = \underbrace{L(s)\left[\omega \omega^T\right]}_{\Omega = \Omega^T} \theta$$

$L(s)$ must be at least stable and minimum phase

Result of extension:

$$Y = \Omega \theta \tag{11.6}$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Result of extension:

$$Y = \Omega\theta \tag{11.7}$$

Adaptation algorithm:

$$\dot{\hat{\theta}} = \gamma\Omega E, \tag{11.8}$$

where $E = Y - \Omega\hat{\theta}$ is the memory extended error, $\gamma > 0$ is the adaptation gain.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Result of extension:

$$Y = \Omega\theta \tag{11.7}$$

Adaptation algorithm:

$$\dot{\hat{\theta}} = \gamma\Omega E, \tag{11.8}$$

where $E = Y - \Omega\hat{\theta}$ is the memory extended error, $\gamma > 0$ is the adaptation gain.

Remark 11.1. If selected $L(s)$ ***is positive, i.e., for any time function*** $f(t) > 0$, $L(s)\left[f(t)\right] > 0 \;\; \forall t \geq T_0$ ***, then the algorithm (11.8) can be simplified as follows:***

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Result of extension:

$$Y = \Omega\theta \tag{11.7}$$

Adaptation algorithm:

$$\dot{\hat{\theta}} = \gamma\Omega E, \tag{11.8}$$

where $E = Y - \Omega\hat{\theta}$ is the memory extended error, $\gamma > 0$ is the adaptation gain.

Remark 11.1. If selected $L(s)$ ***is positive, i.e., for any time function*** $f(t) > 0$, $L(s)\left[f(t)\right] > 0 \ \forall t \geq T_0$ ***, then the algorithm (11.8) can be simplified as follows:***

$$L(s) = \prod_{i=1}^{N} \frac{d_i}{s + d_i}$$

$$\dot{\hat{\theta}} = \gamma E$$

$$\cancel{L(s) = \frac{1}{T^2 s^2 + T\xi s + 1}}$$

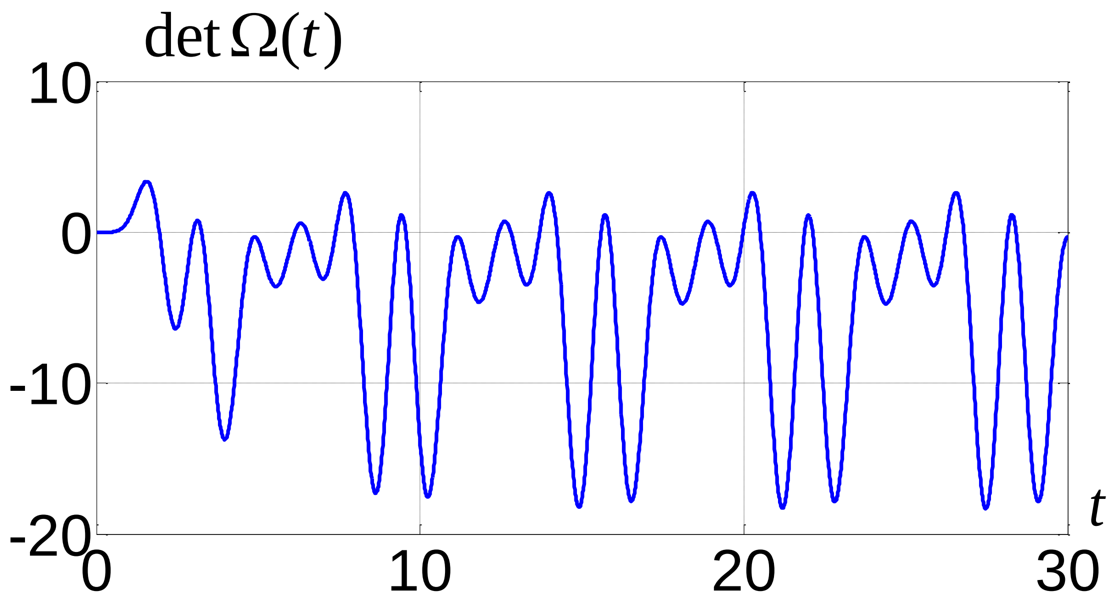
$$\cancel{\xi \in (0,1)}$$

6.3. Dynamic error model with measurable output

Example 11.1. Nonpositive property of an oscillatory block

$$\Omega = L(s)\left[\omega\omega^T\right]$$

$$L(s) = \frac{1}{0.25s^2 + 0.2s + 1}, \quad \omega = \left[1+\sin t, 1+\cos 2t\right]^T$$



Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Parametric error model

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}},\ E = Y - \Omega\hat{\theta},$$

$$L(s) = \prod_{i=1}^{N} \frac{d_i}{s + d_i} : \mathbb{R}^+ \to \mathbb{R}^+$$

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta} \tag{11.10}$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

1. Boundedness

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \qquad (11.11)$$

and evaluate its time derivative in view of (11.10):

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} \le 0.$$

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta}$$

Hence, $\left\|\tilde{\theta}(t)\right\|$ is bounded (for any ω).

If ω is bounded, then $\varepsilon, E,$ and $\dot{\hat{\theta}}$ are bounded.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

1. Boundedness

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \qquad (11.11)$$

and evaluate its time derivative in view of (11.10):

$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} \leq 0.$

Hence, $\left\|\tilde{\theta}(t)\right\|$ is bounded (for any ω).

If ω is bounded, then $\varepsilon, E,$ and $\dot{\hat{\theta}}$ are bounded.

$$V(t) = V(0) - \int_0^{\infty}\tilde{\theta}^T(\tau)\Omega(\tau)\tilde{\theta}(\tau)d\tau \leq c_1 < \infty.$$

Therefore, $\sqrt{\Omega\ (t)}\tilde{\theta}(t) \to 0$ as $t \to \infty$ (thanks to the Barbalat lemma).

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

2. Convergence of ε and E

Since ω is bounded, there exist such constants $c_2, c_3 > 0$ that

$$\int_0^\infty \tilde{\theta}^T \Omega^2 \tilde{\theta} d\tau \leq c_2 \int_0^\infty \tilde{\theta}^T \Omega \tilde{\theta} d\tau \leq c_3 < \infty.$$

As a result, $\Omega(t)\tilde{\theta}(t) = E(t) \to 0$ and $\dot{\hat{\theta}}(t) = -\dot{\tilde{\theta}}(t) = \gamma E(t) \to 0$ as $t \to \infty$.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

2. Convergence of ε and E

Since ω is bounded, there exist such constants $c_2, c_3 > 0$ that

$$\int_0^\infty \tilde{\theta}^T \Omega^2 \tilde{\theta} d\tau \le c_2 \int_0^\infty \tilde{\theta}^T \Omega \tilde{\theta} d\tau \le c_3 < \infty.$$

As a result, $\Omega(t)\tilde{\theta}(t) = E(t) \to 0$ and $\dot{\hat{\theta}}(t) = -\dot{\tilde{\theta}}(t) = \gamma E(t) \to 0$ as $t \to \infty$.

Proceeding, we have

$$E = Y - \Omega\hat{\theta} = L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\hat{\theta}\right] - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= [\textit{Swapping lemma}]$$

$$= L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\right]\hat{\theta} - Z - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

2. Convergence of ε and E

$$E = Y - \Omega\hat{\theta} = L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\hat{\theta}\right] - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= [\textit{Swapping lemma}]$$

$$= L(s)[\omega\varepsilon] + \cancel{L(s)\left[\omega\omega^T\right]}\hat{\theta} - Z - \cancel{L(s)\left[\omega\omega^T\right]}\hat{\theta}$$

where the matrix Z is generated by the swapping matrix filters

$$\dot{X} = \left(I_m \otimes A_L\right)X + \left(I_m \otimes b_L\right)R\dot{\hat{\theta}}, \qquad Z = \left(c_L^T \otimes I_m\right)X$$

$$\dot{R} = \left(I_m \otimes A_L\right)R + \left(I_m \otimes b_L\right)\omega\omega^T$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

The triple (A_L, b_L, c_L) is the minimal realization of the transfer function

$L(s) = c_L^T (sI_N - A_L)^{-1} b_L$ given by

$$A_L = \begin{bmatrix} -d_1 & 1 & 0 & \cdots & 0 \\ 0 & -d_2 & 1 & \ddots & \vdots \\ 0 & 0 & -d_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 0 & -d_N \end{bmatrix}, \; b_L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \prod_{i=1}^{N} d_i \end{bmatrix}, \; c_L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The Kronecker products are defined as

$$I_m \otimes A_L = \begin{bmatrix} A_L & O_N & \cdots & O_N \\ O_N & A_L & \ddots & O_N \\ \vdots & \ddots & \ddots & \vdots \\ O_N & O_N & \cdots & A_L \end{bmatrix}, \; I_m \otimes b_L = \begin{bmatrix} b_L & O_N & \cdots & O_N \\ O_N & b_L & \ddots & O_N \\ \vdots & \ddots & \ddots & \vdots \\ O_N & O_N & \cdots & b_L \end{bmatrix}, \; c_L^T \otimes I_m = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_m \end{bmatrix}^T.$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

2. Convergence of ε and E

$$E = Y - \Omega\hat{\theta} = L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\hat{\theta}\right] - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= [\textit{Swapping lemma}]$$

$$= L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\right]\hat{\theta} - Z - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= L(s)[\omega\varepsilon] - Z,$$

where the matrix Z is generated by the swapping matrix filters

$$\dot{X} = \left(I_m \otimes A_L\right)X + \left(I_m \otimes b_L\right)R\dot{\hat{\theta}}, \qquad Z = \left(c_L^T \otimes I_m\right)X$$

$$\dot{R} = \left(I_m \otimes A_L\right)R + \left(I_m \otimes b_L\right)\omega\omega^T$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

2. Convergence of ε and E

$$E = Y - \Omega\hat{\theta} = L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\hat{\theta}\right] - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= [\textit{Swapping lemma}]$$

$$= L(s)[\omega\varepsilon] + L(s)\left[\omega\omega^T\right]\hat{\theta} - Z - L(s)\left[\omega\omega^T\right]\hat{\theta}$$

$$= L(s)[\omega\varepsilon] - Z,$$

where the matrix Z is generated by the swapping matrix filters

$$\dot{X} = (I_m \otimes A_L)X + (I_m \otimes b_L)R\dot{\hat{\theta}}, \quad Z = (c_L^T \otimes I_m)X$$

$$\dot{R} = (I_m \otimes A_L)R + (I_m \otimes b_L)\omega\omega^T$$

$$E(t) \to 0$$

$$\dot{\hat{\theta}}(t) = \gamma\Omega(t)E(t) \to 0$$

$$Z(t) \to 0$$

if ω *is bounded*

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.11}$$

and evaluate its time derivative in view of (11.10):

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta}$$

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} = -\tilde{\theta}^T L(s)\left[\omega\omega^T\right]\tilde{\theta}$$

$$= -\tilde{\theta}^T c_L^T \int_0^t e^{A_L(t-\tau)} b_L \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta}$$

$$A_L = \begin{bmatrix} -d_1 & 1 & 0 & \cdots & 0 \\ 0 & -d_2 & 1 & \ddots & \vdots \\ 0 & 0 & -d_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 1 \\ 0 & \cdots & 0 & 0 & -d_N \end{bmatrix}, \quad b_L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \prod_{i=1}^{N} d_i \end{bmatrix}, \quad c_L = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.11}$$

and evaluate its time derivative in view of (11.10):

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} = -\tilde{\theta}^T L(s)\left[\omega\omega^T\right]\tilde{\theta}$$

$$= -\tilde{\theta}^T c_L^T \int_0^t e^{A_L(t-\tau)} b_L \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} = -\tilde{\theta}^T\left(\int_0^t \sum_{i=1}^N c_i e^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau\right)\tilde{\theta}$$

where c_i are constants depending on the elements of A_L, b_L, c_L.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.11}$$

and evaluate its time derivative in view of (11.10):

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} = -\tilde{\theta}^T L(s)\left[\omega\omega^T\right]\tilde{\theta}$$

$$= -\tilde{\theta}^T c_L^T \int_0^t \mathrm{e}^{A_L(t-\tau)} b_L \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} = -\tilde{\theta}^T\left(\int_0^t \sum_{i=1}^N c_i \mathrm{e}^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau\right)\tilde{\theta}$$

$$\leq -\tilde{\theta}^T\left(\sum_{i=1}^N \int_{t-T}^t c_i \mathrm{e}^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau\right)\tilde{\theta}$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.11}$$

and evaluate its time derivative in view of (11.10):

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} = -\tilde{\theta}^T L(s)\left[\omega\omega^T\right]\tilde{\theta}$$

$$= -\tilde{\theta}^T c_L^T \int_0^t e^{A_L(t-\tau)} b_L \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} = -\tilde{\theta}^T\left(\int_0^t \sum_{i=1}^N c_i e^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau\right)\tilde{\theta}$$

$$\le -\tilde{\theta}^T\left(\sum_{i=1}^N \int_{t-T}^t c_i e^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau\right)\tilde{\theta}$$

$$\le -\tilde{\theta}^T\left(\sum_{i=1}^N c_i e^{-d_i T}\right)\int_{t-T}^t \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} \qquad c_i > 0$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.11}$$

and evaluate its time derivative in view of (11.10):

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} = -\tilde{\theta}^T L(s)\left[\omega\omega^T\right]\tilde{\theta}$$

$$= -\tilde{\theta}^T c_L^T \int_0^t e^{A_L(t-\tau)} b_L \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} = -\tilde{\theta}^T \left(\int_0^t \sum_{i=1}^N c_i e^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau \right)\tilde{\theta}$$

$$\le -\tilde{\theta}^T \left(\sum_{i=1}^N \int_{t-T}^t c_i e^{-d_i(t-\tau)}\omega(\tau)\omega^T(\tau)d\tau \right)\tilde{\theta}$$

if $\omega \in PE$, *i.e.*,

$$\int_{t-T}^{t} \omega(\tau)\omega^T(\tau)d\tau \ge \alpha I > 0$$

$$\le -\tilde{\theta}^T \left(\sum_{i=1}^N c_i e^{-d_i T} \right) \int_{t-T}^t \omega(\tau)\omega^T(\tau)d\tau\,\tilde{\theta} \le -2\alpha\gamma \left(\sum_{i=1}^N c_i e^{-d_i T} \right) V$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

3. Convergence of $\tilde{\theta}(t)$

$$\dot{V} \le -2\alpha\gamma\left(\sum_{i=1}^{N} c_i e^{-d_i T}\right)V$$

$$\Downarrow$$

$$V(t) \le e^{-2\alpha\gamma\left(\sum_{i=1}^{N} c_i e^{-d_i T}\right)t} V(0)$$

$$\left\|\tilde{\theta}(t)\right\|^2 \le e^{-2\alpha\gamma\left(\sum_{i=1}^{N} c_i e^{-d_i T}\right)t} \left\|\tilde{\theta}(0)\right\|^2$$

if $\omega \in PE$, *i.e.*,

$$\int_{t-T}^{t} \omega(\tau)\omega^T(\tau)d\tau \ge \alpha I > 0$$

Hence, $\tilde{\theta}(t)$ tends to zero exponentially iff $\omega(t) \in PE$.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

1. The vector of parametric errors $\tilde{\theta}(t)$ is bounded.

   If $\omega(t)$ are bounded, then $\varepsilon(t), E(t), \dot{\hat{\theta}}(t)$ are bounded;

2. If $\omega(t)$ are bounded, then $E(t)$ and $\varepsilon(t)$ tend to zero asymptotically as $t \to 0$;

3. $\left\|\tilde{\theta}(t)\right\|$ approaches zero exponentially fast iff $\omega \in PE$, i.e.,

$$\int_{t}^{t+T} \omega(\tau)\omega^{T}(\tau)d\tau \geq \alpha I > 0$$

   for some positive α, T;

4. If $\omega \in PE$, then the rate of parametric convergence can be increased arbitrarily (in theory) by increasing the gain γ .

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

1. The vector of parametric errors $\tilde{\theta}(t)$ is bounded.

   If $\omega(t)$ are bounded, then $\varepsilon(t), E(t), \dot{\hat{\theta}}(t)$ are bounded;

2. If $\omega(t)$ are bounded, then $E(t)$ and $\varepsilon(t)$ tend to zero asymptotically as $t \to 0$;

3. $\left\|\tilde{\theta}(t)\right\|$ approaches zero exponentially fast iff $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0$$

is not provided

by the gradient algorithm

$\dot{\hat{\theta}} = \gamma\omega\varepsilon$

for some positive α, T;

4. If $\omega \in PE$, then the rate of parametric convergence can be increased arbitrarily (in theory) by increasing the gain γ.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

5. Denote by $\lambda_\Omega(t)$ the minimum eigenvalue of the matrix $\Omega(t)$. Then it follows from the inequality

$$\dot{V} = -\tilde{\theta}^T \Omega(t) \tilde{\theta} \leq -\lambda_\Omega(t) \tilde{\theta}^T \tilde{\theta} = -2\gamma\lambda_\Omega(t) V,$$

that

$$V(t) \leq \mathrm{e}^{-2\gamma \int_0^t \lambda_\Omega(\tau) d\tau} V(0), \quad \text{or} \quad \left\|\tilde{\theta}(t)\right\|^2 \leq \mathrm{e}^{-2\gamma \int_0^t \lambda_\Omega(\tau) d\tau} \left\|\tilde{\theta}(0)\right\|^2,$$

and, hence, even if $\omega \notin PE$ but $\lambda_\Omega(t) \notin L_1$, i.e.,

$$\int_0^\infty \lambda_\Omega(\tau) d\tau = \infty,$$

$\left\|\tilde{\theta}(t)\right\|$ approaches zero **asymptotically** as $t \to 0.$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

5. Denote by $\lambda_\Omega(t)$ the minimum eigenvalue of the matrix $\Omega(t)$. Then it follows from the inequality

$$\dot{V} = -\tilde{\theta}^T\Omega(t)\tilde{\theta} \le -\lambda_\Omega(t)\tilde{\theta}^T\tilde{\theta} = -2\gamma\lambda_\Omega(t)V,$$

that

$$V(t) \le e^{-2\gamma\int_0^t \lambda_\Omega(\tau)d\tau} V(0), \text{ or } \left\|\tilde{\theta}(t)\right\|^2 \le e^{-2\gamma\int_0^t \lambda_\Omega(\tau)d\tau} \left\|\tilde{\theta}(0)\right\|^2,$$

and, hence, even if $\omega \notin PE$ but $\lambda_\Omega(t) \notin L_1$, i.e.,

$$\int_0^\infty \lambda_\Omega(\tau)d\tau = \infty,$$

$\left\|\tilde{\theta}(t)\right\|$ approaches zero **asymptotically** as $t \to 0$.

is not provided by the gradient algorithm $\dot{\hat{\theta}} = \gamma\omega\varepsilon$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma \Omega \tilde{\theta} \tag{11.12}$$

a) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 \left(\rho I + \Omega(t) \right)^{-1},$$

where $\gamma_0 > 0$ is a constant, $\rho > 0$ is a small constant.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta} \qquad (11.12)$$

a) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0\left(\rho I + \Omega(t)\right)^{-1},$$

where $\gamma_0 > 0$ is a constant, $\rho > 0$ is a small constant.

Then

$$\dot{\tilde{\theta}} \approx -\gamma_0\tilde{\theta},$$

and we obtain "almost" monotonic element-wise exponential convergence of the parametric error if $\omega \in PE$:

$$\tilde{\theta}_i(t) \approx e^{-\gamma_0 t}\tilde{\theta}_i(0).$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta} \tag{11.12}$$

b) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 adj\{\Omega\},$$

where $\gamma_0 > 0$ is a constant, $adj\{\Omega\}$ is the adjugate of Ω such that

$$adj\{\Omega\} = \Omega^{-1} det\{\Omega\}.$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma\Omega\tilde{\theta} \tag{11.12}$$

b) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 adj\{\Omega\},$$

where $\gamma_0 > 0$ is a constant, $adj\{\Omega\}$ is the adjugate of Ω such that

$$adj\{\Omega\} = \Omega^{-1} det\{\Omega\},$$

then

$$\dot{\tilde{\theta}} = -\gamma_0 det\{\Omega\}\tilde{\theta},$$

and if $\omega \in PE$, we obtain monotonic element-wise exponential convergence of the parameters.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

7. Let the positive operator

$$L(s) = d_0 \prod_{i=1}^{\rho} \frac{1}{s + p_i} = \frac{d_0}{s^\rho + d_{\rho-1}s^{\rho-1} + d_{\rho-2}s^{\rho-2} + \cdots d_0}, \tag{11.13}$$

where p_i are positive real numbers, d_i $(i = 1, 2, ..., \rho)$ are the coefficients of the Hurwitz polynomial, ρ is a sufficiently large order.

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

7. Let the positive operator

$$L(s) = d_0 \prod_{i=1}^{\rho} \frac{1}{s + p_i} = \frac{d_0}{s^{\rho} + d_{\rho-1}s^{\rho-1} + d_{\rho-2}s^{\rho-2} + \cdots d_0}, \qquad (11.13)$$

where p_i are positive real numbers, d_i $(i = 1, 2, ..., \rho)$ are the coefficients of the Hurwitz polynomial, ρ is a sufficiently large order.

Then the algorithm

$$\dot{\hat{\theta}} = \gamma E = \gamma\left(L(s)[\omega y] - L(s)\left[\omega\omega^T\right]\hat{\theta}\right)$$

can be represented in the closed-loop form generating the high-order time derivatives of the adjustable parameters $\hat{\theta}^{(j)}, j = 1, 2, ..., \rho + 1.$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

$$\dot{\hat{\theta}} = \gamma\left(L(s)\left[\omega y\right] - L(s)\left[\omega\omega^T\right]\hat{\theta}\right)$$

$$\Downarrow$$

$$\hat{\theta}^{(\rho+1)} + \left(d_{\rho-1}I_m + \gamma d_\rho \Omega\right)\hat{\theta}^{(\rho)} + \left(d_{\rho-2}I_m + \gamma d_{\rho-1}\Omega + \gamma d_\rho C^{\rho}_{\rho-1}\dot{\Omega}\right)\hat{\theta}^{(\rho-1)} + \cdots$$

$$+\left(d_1 I_m + \gamma\sum_{j=2}^{\rho} d_j C_2^j \Omega^{(j-2)}\right)\ddot{\hat{\theta}} + \left(d_0 I_m + \gamma\sum_{j=1}^{\rho} d_j C_1^j \Omega^{(j-1)}\right)\dot{\hat{\theta}} = \gamma\omega\varepsilon,$$

where $d_\rho = 1,\ \dot{\hat{\theta}}(0) = \ddot{\hat{\theta}}(0) = \ldots = \hat{\theta}^{(\rho)}(0) = 0,$

$$\Omega^{(j)} = \frac{d_0 s^j}{s^\rho + d_{\rho-1}s^{\rho-1} + d_{\rho-2}s^{\rho-2} + \cdots d_0},\quad C_i^j = \frac{j!}{i!\left(j-i\right)!}.$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

$$\dot{\hat{\theta}} = \gamma\Big(L(s)\big[\omega y\big] - L(s)\big[\omega\omega^T\big]\hat{\theta}\Big)$$

$$\Downarrow$$

$$\hat{\theta}^{(\rho+1)} + \big(d_{\rho-1}I_m + \gamma d_\rho \Omega\big)\hat{\theta}^{(\rho)} + \big(d_{\rho-2}I_m + \gamma d_{\rho-1}\Omega + \gamma d_\rho C_{\rho-1}^{\rho}\dot{\Omega}\big)\hat{\theta}^{(\rho-1)} + \cdots$$

$$+\left(d_1 I_m + \gamma\sum_{j=2}^{\rho} d_j C_2^j \Omega^{(j-2)}\right)\ddot{\hat{\theta}} + \left(d_0 I_m + \gamma\sum_{j=1}^{\rho} d_j C_1^j \Omega^{(j-1)}\right)\dot{\hat{\theta}} = \gamma\omega\varepsilon,$$

where $d_\rho = 1,\ \dot{\hat{\theta}}(0) = \ddot{\hat{\theta}}(0) = \ldots = \hat{\theta}^{(\rho)}(0) = 0,$

$$\Omega^{(j)} = \frac{d_0 s^j}{s^\rho + d_{\rho-1}s^{\rho-1} + d_{\rho-2}s^{\rho-2} + \cdots d_0},\quad C_i^j = \frac{j!}{i!(j-i)!}.$$

is not provided

by the gradient algorithm

$\dot{\hat{\theta}} = \gamma\omega\varepsilon$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Properties

Example 11.2. The second and the third order high-order tuners

$$L(s) = \frac{d_0}{s + d_0}:$$

$$\ddot{\hat{\theta}} + \left(d_0 I_m + \gamma\Omega\right)\dot{\hat{\theta}} = \gamma\omega\varepsilon, \quad \dot{\hat{\theta}}(0) = 0$$

$$L(s) = \frac{d_0}{s^2 + d_1 s + d_0}:$$

$$\dddot{\hat{\theta}} + \left(d_1 I_m + \gamma\Omega\right)\ddot{\hat{\theta}} + \left(d_0 I_m + \gamma d_1\Omega + 2\gamma\dot{\Omega}\right)\dot{\hat{\theta}} = \gamma\omega\varepsilon, \quad \dot{\hat{\theta}}(0) = \ddot{\hat{\theta}}(0) = 0$$

Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)

Example 11.3. Adaptive parameters identifier based on the adaptation algorithm with regressor memory

Linear regressor

$$y = \theta^T \omega, \tag{11.14}$$

$$\theta = [2,\ 3,\ 4,\ 5]^T,\ \ \omega(t) = [\sin t,\ \cos t,\ \sin 2t,\ \cos 2t]^T \in PE$$

Gradient adaptation algorithm

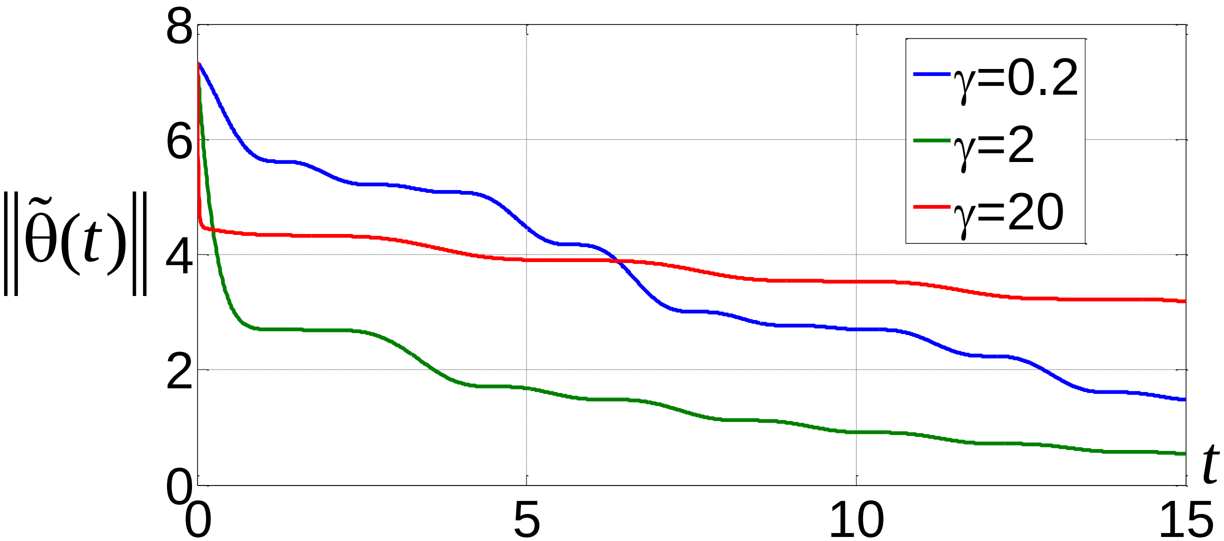
$$\dot{\hat{\theta}} = \gamma\omega\left(y - \hat{\theta}^T\omega\right) \tag{11.15}$$

Adaptation algorithm with memory regression

$$\dot{\hat{\theta}} = \gamma\left(L(s)[\omega y] - L(s)\left[\omega\omega^T\right]\hat{\theta}\right), \tag{11.16}$$

$$L(s) = \frac{1}{s+1}$$

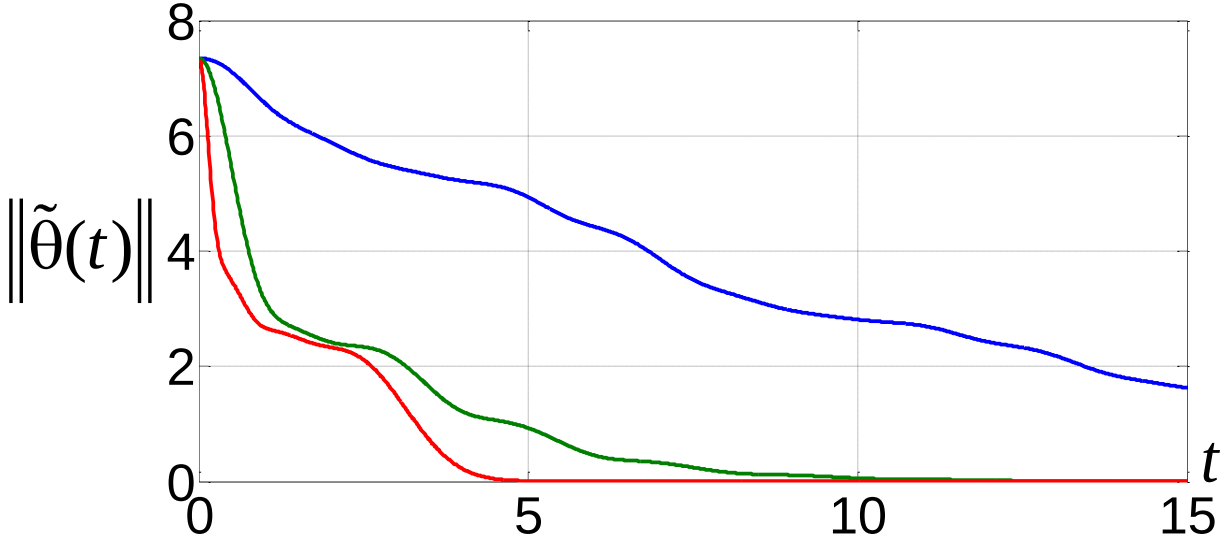
Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)



Gradient AA

$$\dot{\tilde\theta} = -\gamma\omega\omega^T\tilde\theta$$

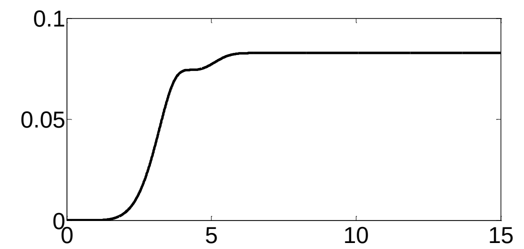
$$\lambda_{\min}\left\{\omega\omega^T\right\} \equiv 0$$



AA with RM

$$\dot{\tilde\theta} = -\gamma\Omega\tilde\theta$$

$$\lambda_\Omega(t) \geq 0$$



3. Lion scheme

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Regressor extension:

$$y = \theta^T \omega = \omega^T \theta \tag{11.17}$$

$$\Downarrow$$

3. Lion scheme

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Regressor extension:

$$y = \theta^T \omega = \omega^T \theta \tag{11.17}$$

$$\Downarrow$$

Select and apply different proper asymptotically stable and minimum phase (PASMP) transfer functions $H_i(s),\ i = 1, 2, \ldots, p-1$

3. Lion scheme

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Regressor extension:

$$y = \theta^T \omega = \omega^T \theta \qquad (11.17)$$

$$\Downarrow$$

Select and apply different proper asymptotically stable and minimum phase (PASMP) transfer functions

$$H_i(s),\ i = 1, 2, \ldots, p-1$$

$$\Downarrow$$

$$\underbrace{H_i(s)[y]}_{Y_i} = \underbrace{H_i(s)\left[\omega^T\right]}_{W_i^T}\theta$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Regressor extension:

$$\underbrace{\begin{bmatrix} y \\ H_1(s)[y] \\ \vdots \\ H_{p-1}(s)[y] \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} \omega^T \\ H_1(s)\left[\omega^T\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\right] \end{bmatrix}}_{W^T} \theta$$

Result of extension:

$$Y = W^T \theta \qquad (11.18)$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Result of extension:

$$Y = W^T \theta \tag{11.19}$$

Adaptation algorithm:

$$\dot{\hat{\theta}} = \gamma W E, \tag{11.20}$$

where $E = Y - W^T \hat{\theta}$ is the dynamically extended error, $\gamma > 0$ is the adaptation gain.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Parametric error model:

$$\dot{\tilde{\theta}} = -\dot{\hat{\theta}}, \quad E = Y - W^T\hat{\theta}$$

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta} \tag{11.21}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

1. Boundedness

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \qquad (11.22)$$

and evaluate its time derivative in view of (11.21):

$$\dot{V} = -\tilde{\theta}^T W(t) W^T(t)\tilde{\theta} \le 0.$$

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta},$$

Hence, $\left\|\tilde{\theta}(t)\right\|$ is bounded (for any ω).

If ω, $\dot{\omega}$ are bounded, then $\varepsilon, E,$ and $\dot{\hat{\theta}}$ are bounded.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

1. Boundedness

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.22}$$

and evaluate its time derivative in view of (11.21):

$\dot{V} = -\tilde{\theta}^T W(t) W^T(t)\tilde{\theta} \le 0.$

Hence, $\left\|\tilde{\theta}(t)\right\|$ is bounded (for any ω).

If ω, $\dot{\omega}$ are bounded, then $\varepsilon, E,$ and $\dot{\hat{\theta}}$ are bounded.

$$V(t) = V(0) - \int_0^\infty \tilde{\theta}^T(\tau) W(\tau) W^T(\tau)\tilde{\theta}(\tau) d\tau \le c_1 < \infty.$$

Therefore, $E(t) = W^T(t)\tilde{\theta}(t) \to 0$ as $t \to \infty$ (thanks to the Barbalat lemma).

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

Since $\Omega(t)\tilde{\theta}(t) = E(t) \to 0$

As a result, if $W(t)$ is bounded then

$$\dot{\hat{\theta}}(t) = -\dot{\tilde{\theta}}(t) = \gamma W(t) E(t) \to 0.$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

Proceeding, we have

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] + H(s)\left[\omega^T\hat{\theta}\right] - H(s)\left[\omega^T\right]\hat{\theta}$$

$$= H(s)[\varepsilon] + \begin{bmatrix} \omega^T\hat{\theta} \\ H_1(s)\left[\omega^T\hat{\theta}\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\hat{\theta}\right] \end{bmatrix} - \begin{bmatrix} \omega^T \\ H_1(s)\left[\omega^T\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\right] \end{bmatrix}\hat{\theta}$$

$$H(s) = \left[1, H_1(s), H_2(s), \ldots, H_{p-1}(s)\right]^T$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] + \begin{bmatrix} \omega^T\hat{\theta} \\ H_1(s)\left[\omega^T\hat{\theta}\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\hat{\theta}\right] \end{bmatrix} - \begin{bmatrix} \omega^T \\ H_1(s)\left[\omega^T\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\right] \end{bmatrix}\hat{\theta}$$

$$= H(s)[\varepsilon] + \begin{bmatrix} 0 \\ H_1(s)\left[\omega^T\hat{\theta}\right] - H_1(s)\left[\omega^T\right]\hat{\theta} \\ \vdots \\ H_{p-1}(s)\left[\omega^T\hat{\theta}\right] - H_{p-1}(s)\left[\omega^T\right]\hat{\theta} \end{bmatrix}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] + \begin{bmatrix} 0 \\ H_1(s)\left[\omega^T\hat{\theta}\right] - H_1(s)\left[\omega^T\right]\hat{\theta} \\ \vdots \\ H_{p-1}(s)\left[\omega^T\hat{\theta}\right] - H_{p-1}(s)\left[\omega^T\right]\hat{\theta} \end{bmatrix}$$

$$= H(s)[\varepsilon] + \begin{bmatrix} 0 \\ H_1(s)\left[\omega^T\right]\hat{\theta} - H_{C1}(s)\left[H_{B1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] - H_1(s)\left[\omega^T\right]\hat{\theta} \\ \vdots \\ H_{p-1}(s)\left[\omega^T\right]\hat{\theta} - H_{C\,p-1}(s)\left[H_{B\,p-1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] - H_{p-1}(s)\left[\omega^T\right]\hat{\theta} \end{bmatrix}$$

$$H_i(s) = c_i^T\left(Is - A_i\right)^{-1} b_i,\; H_{Bi}(s) = \left(Is - A_i\right)^{-1} b_i,\; H_{Ci}(s) = c_i^T\left(Is - A_i\right)^{-1}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] + \begin{bmatrix} 0 \\ H_1(s)\left[\omega^T\hat{\theta}\right] - H_1(s)\left[\omega^T\right]\hat{\theta} \\ \vdots \\ H_{p-1}(s)\left[\omega^T\hat{\theta}\right] - H_{p-1}(s)\left[\omega^T\right]\hat{\theta} \end{bmatrix}$$

$$= H(s)[\varepsilon] + \begin{bmatrix} 0 \\ \cancel{H_1(s)\left[\omega^T\right]\hat{\theta}} - H_{C1}(s)\left[H_{B1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] - \cancel{H_1(s)\left[\omega^T\right]\hat{\theta}} \\ \vdots \\ \cancel{H_{p-1}(s)\left[\omega^T\right]\hat{\theta}} - H_{C\,p-1}(s)\left[H_{B\,p-1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] - \cancel{H_{p-1}(s)\left[\omega^T\right]\hat{\theta}} \end{bmatrix}$$

$$H_i(s) = c_i^T\left(Is - A_i\right)^{-1} b_i,\ H_{Bi}(s) = \left(Is - A_i\right)^{-1} b_i,\ H_{Ci}(s) = c_i^T\left(Is - A_i\right)^{-1}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] - \begin{bmatrix} 0 \\ H_{C1}(s)\left[H_{B1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] \\ \vdots \\ H_{C\,p-1}(s)\left[H_{B\,p-1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] \end{bmatrix}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

2. Convergence of ε and E

$$E = Y - W^T\hat{\theta} = H(s)[\varepsilon] - \begin{bmatrix} 0 \\ H_{C1}(s)\left[H_{B1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] \\ \vdots \\ H_{C\,p-1}(s)\left[H_{B\,p-1}(s)\left[\omega^T\right]\dot{\hat{\theta}}\right] \end{bmatrix}$$

$$E(t) \to 0$$

$$\dot{\hat{\theta}}(t) = \gamma W(t) E(t) \to 0$$

Hence, $\varepsilon(t) \to 0$ as $t \to \infty$.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

3. Convergence of $\tilde{\theta}$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.23}$$

and evaluate its time derivative in view of (11.21):

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta},$$

$$\dot{V} = -\tilde{\theta}^T W(t) W^T(t)\tilde{\theta} =$$

$$= -\tilde{\theta}^T\left(\omega\omega^T + \sum_{i=1}^{p-1} H_i(s)[\omega] H_i(s)\left[\omega^T\right]\right)\tilde{\theta}$$

$$W = \left[\omega \vdots H_1(s)[\omega] \vdots \ldots \vdots H_{p-1}(s)[\omega]\right], \quad W^T = \begin{bmatrix} \omega^T \\ H_1(s)\left[\omega^T\right] \\ \vdots \\ H_{p-1}(s)\left[\omega^T\right] \end{bmatrix}$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

3. Convergence of $\tilde{\theta}$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \tag{11.23}$$

and evaluate its time derivative in view of (11.21):

$$\dot{V} = -\tilde{\theta}^T W(t) W^T(t)\tilde{\theta} =$$

$$= -\tilde{\theta}^T\left(\omega\omega^T + \sum_{i=1}^{p-1} H_i(s)[\omega] H_i(s)\left[\omega^T\right]\right)\tilde{\theta}$$

If $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0,$$

then $H_i(s)[\omega] \in PE$, i.e.,

$$\int_t^{t+T} H_i(s)[\omega(\tau)] H_i(s)\left[\omega^T(\tau)\right] d\tau \geq \alpha_i I > 0$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

3. Convergence of $\tilde{\theta}$

Select the Lyapunov function candidate

$$V = \frac{1}{2\gamma}\tilde{\theta}^T\tilde{\theta} \qquad (11.23)$$

and evaluate its time derivative in view of (11.21):

$$\dot{V} = -\tilde{\theta}^T W(t) W^T(t)\tilde{\theta} =$$

$$= -\tilde{\theta}^T\left(\omega\omega^T + \sum_{i=1}^{p-1} H_i(s)[\omega] H_i(s)\left[\omega^T\right]\right)\tilde{\theta} \leq -\beta\tilde{\theta}^T\tilde{\theta} = -2\gamma\beta V$$

If $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0,$$

$\Rightarrow$

then $H_i(s)[\omega] \in PE$, i.e.,

$$\int_t^{t+T} H_i(s)[\omega(\tau)] H_i(s)\left[\omega^T(\tau)\right] d\tau \geq \alpha_i I > 0$$

Elements of ω *are linearly independent*

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

3. Convergence of $\tilde{\theta}$

$$\dot{V} \leq -2\gamma\beta V$$

$$\Downarrow$$

$$V(t) \leq e^{-2\beta\gamma t} V(0)$$

$$\left\|\tilde{\theta}(t)\right\|^2 \leq e^{-2\beta\gamma t} \left\|\tilde{\theta}(0)\right\|^2$$

if $\omega \in PE$, *i.e.*,

$$\int_{t-T}^{t} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0$$

Hence, $\tilde{\theta}(t)$ tends to zero exponentially iff $\omega(t) \in PE$.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

1. The vector of parametric errors $\tilde{\theta}(t)$ is bounded.

   If $\omega(t), \dot{\omega}(t)$ are bounded, then $\varepsilon(t), E(t), \dot{\hat{\theta}}(t)$ are bounded;

2. If $\omega(t), \dot{\omega}(t)$ are bounded, then $E(t)$ and $\varepsilon(t)$ tend to zero asymptotically

   as $t \to 0$;

3. $\left\|\tilde{\theta}(t)\right\|$ approaches zero exponentially fast iff $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0$$

   for some positive $\alpha,\ T$;

4. If $\omega \in PE$, then the rate of parametric convergence can be increased arbitrarily (in theory) by increasing the gain γ .

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

1. The vector of parametric errors $\tilde{\theta}(t)$ is bounded.

   If $\omega(t), \dot{\omega}(t)$ are bounded, then $\varepsilon(t), E(t), \dot{\hat{\theta}}(t)$ are bounded;

2. If $\omega(t), \dot{\omega}(t)$ are bounded, then $E(t)$ and $\varepsilon(t)$ tend to zero asymptotically as $t \to 0$;

3. $\left\|\tilde{\theta}(t)\right\|$ approaches zero exponentially fast iff $\omega \in PE$, i.e.,

$$\int_t^{t+T} \omega(\tau)\omega^T(\tau)d\tau \geq \alpha I > 0$$

   for some positive α, T;

is not provided

by the gradient algorithm

$\dot{\hat{\theta}} = \gamma\omega\varepsilon$

4. If $\omega \in PE$, then the rate of parametric convergence can be increased arbitrarily (in theory) by increasing the gain γ .

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

5. Denote by $\lambda_W(t)$ the minimum eigenvalue of the matrix $W(t)W^T(t)$.

Then it follows from the inequality

$$\dot{V} = -\tilde{\theta}^T W(t)W^T(t)\tilde{\theta} \le -\lambda_W(t)\tilde{\theta}^T\tilde{\theta} = -2\gamma\lambda_W(t)V,$$

that

$$V(t) \le \mathrm{e}^{-2\gamma\int_0^t \lambda_W(\tau)d\tau} V(0), \quad \text{or} \quad \left\|\tilde{\theta}(t)\right\|^2 \le \mathrm{e}^{-2\gamma\int_0^t \lambda_W(\tau)d\tau} \left\|\tilde{\theta}(0)\right\|^2,$$

and, hence, even if $\omega \notin PE$ but $\lambda_W(t) \notin L_1$, i.e.,

$$\int_0^\infty \lambda_W(\tau)d\tau = \infty,$$

$\left\|\tilde{\theta}(t)\right\|$ approaches zero **asymptotically** as $t \to 0$.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

5. Denote by $\lambda_W(t)$ the minimum eigenvalue of the matrix $W(t)W^T(t)$.

Then it follows from the inequality

$$\dot{V} = -\tilde{\theta}^T W(t)W^T(t)\tilde{\theta} \le -\lambda_W(t)\tilde{\theta}^T\tilde{\theta} = -2\gamma\lambda_W(t)V,$$

that

$$V(t) \le \mathrm{e}^{-2\gamma\int_0^t \lambda_W(\tau)d\tau} V(0), \text{ or } \left\|\tilde{\theta}(t)\right\|^2 \le \mathrm{e}^{-2\gamma\int_0^t \lambda_W(\tau)d\tau} \left\|\tilde{\theta}(0)\right\|^2,$$

and, hence, even if $\omega \notin PE$ but $\lambda_W(t) \notin L_1$, i.e.,

$$\int_0^\infty \lambda_W(\tau)d\tau = \infty,$$

$\left\|\tilde{\theta}(t)\right\|$ approaches zero **asymptotically** as $t \to 0$.

is not provided
by the gradient algorithm
$\dot{\hat{\theta}} = \gamma\omega\varepsilon$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta} \qquad (11.24)$$

a) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 \left(\rho I + W(t) W^T(t) \right)^{-1},$$

where $\gamma_0 > 0$ is a constant, $\rho > 0$ is a small constant.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta} \tag{11.24}$$

a) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 \left(\rho I + W(t) W^T(t) \right)^{-1},$$

where $\gamma_0 > 0$ is a constant, $\rho > 0$ is a small constant.

Then

$$\dot{\tilde{\theta}} \approx -\gamma_0 \tilde{\theta},$$

and we obtain "almost" monotonic element-wise exponential convergence of the parametric error if $\omega \in PE$:

$$\tilde{\theta}_i(t) \approx e^{-\gamma_0 t} \tilde{\theta}_i(0).$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta} \tag{11.24}$$

b) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 adj\left\{W W^T\right\},$$

where $\gamma_0 > 0$ is a constant, $adj\left\{W W^T\right\}$ is the adjugate of $W W^T$ such that

$$adj\left\{W W^T\right\} = \left(W W^T\right)^{-1} det\left\{W W^T\right\}.$$

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Properties

6. Consider the model of parametric error

$$\dot{\tilde{\theta}} = -\gamma W W^T \tilde{\theta} \tag{11.24}$$

b) If we select the adaptation gain as

$$\gamma = \gamma(t) = \gamma_0 adj\left\{WW^T\right\},$$

where $\gamma_0 > 0$ is a constant, $adj\left\{WW^T\right\}$ is the adjugate of WW^T such that

$$adj\left\{WW^T\right\} = \left(WW^T\right)^{-1} det\left\{WW^T\right\}.$$

Then

$$\dot{\tilde{\theta}} = -\gamma_0 det\left\{WW^T\right\}\tilde{\theta},$$

and if $\omega \in PE$, we obtain monotonic element-wise exponential convergence of the parameters.

Algorithm with dynamic regressor extension (P. Lion, AIAA, 1967)

Example 11.4. Adaptive parameters identifier based on the adaptation algorithm with regressor memory

Linear regressor

$$y = \theta^T \omega, \tag{11.25}$$

$$\theta = [2,\ 3,\ 4,\ 5]^T,\ \ \omega(t) = [\sin t,\ \cos t,\ \sin 2t,\ \cos 2t]^T \in PE$$

Gradient adaptation algorithm

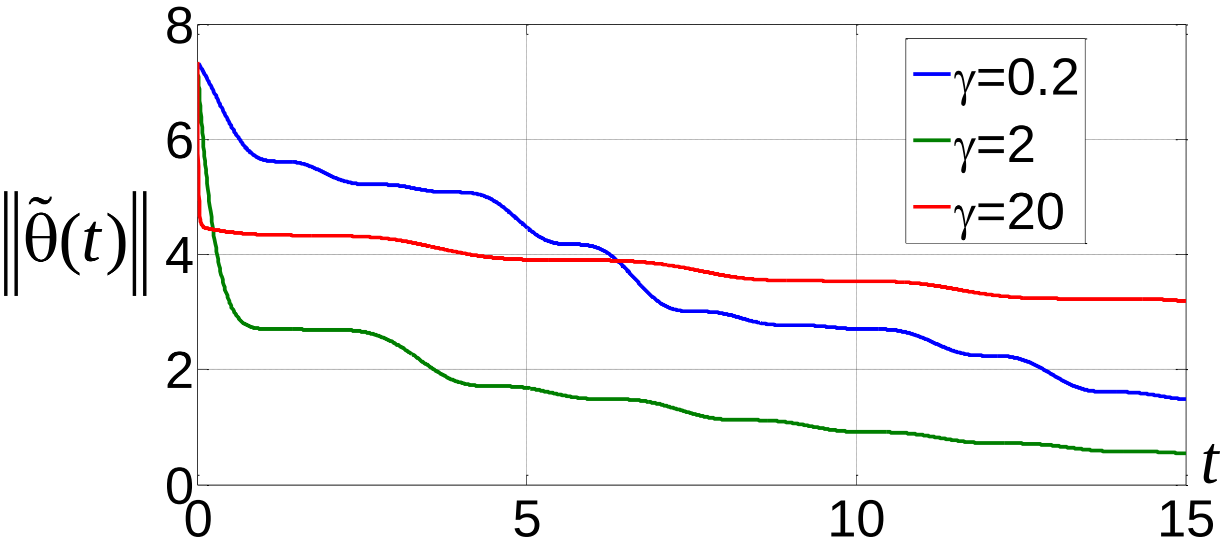
$$\dot{\hat{\theta}} = \gamma\omega\left(y - \hat{\theta}^T\omega\right) \tag{11.26}$$

Adaptation algorithm with dynamic regressor extension

$$\dot{\hat{\theta}} = \gamma\left(H(s)\left[\omega^T\right]\right)^T\left(H(s)[y] - H(s)\left[\omega^T\right]\hat{\theta}\right), \tag{11.27}$$

$$H(s) = [1, H_1(s), H_2(s), H_3(s)]^T,\ H_1(s) = \frac{1}{2s+1},\ H_2(s) = \frac{1}{s+1},\ H_3(s) = \frac{2}{s+2}.$$

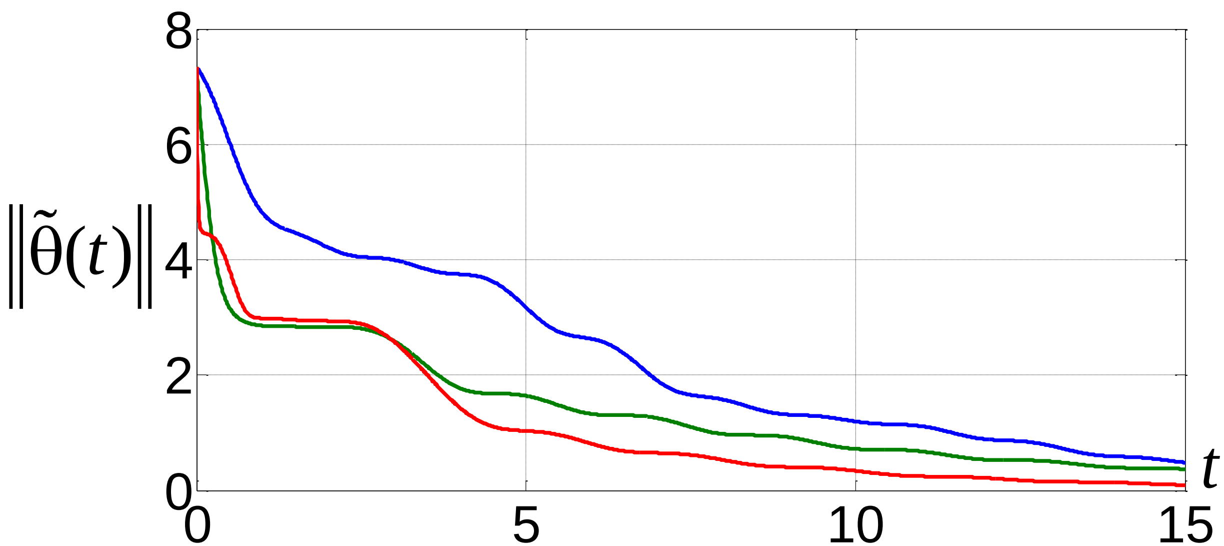
Algorithm with regressor memory (G. Kreisselmeier, TAC, 1977)



Gradient AA

$$\dot{\tilde{\theta}} = -\gamma\omega\omega^T\tilde{\theta}$$

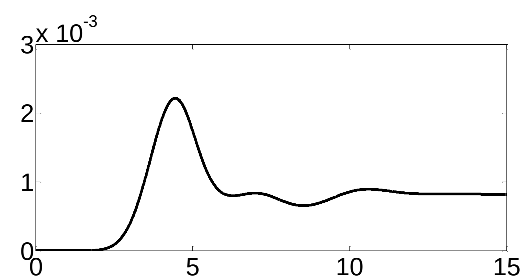
$$\lambda_{\min}\left\{\omega\omega^T\right\} \equiv 0$$



AA with DRE

$$\dot{\tilde{\theta}} = -\gamma WW^T\tilde{\theta}$$

$$\lambda_W(t) \geq 0$$



Example 11.5. Nonexponential convergence of the algorithms with RM and DRE

Linear regressor

$$y=\theta^T\omega,\quad \theta=\begin{bmatrix}-3\\3\end{bmatrix},\quad \omega(t)=\begin{bmatrix}1\\ \dfrac{\sin t+\cos t}{\sqrt{1+t}}-\dfrac{\sin t}{2\sqrt{(1+t)^3}}\end{bmatrix}\notin PE$$

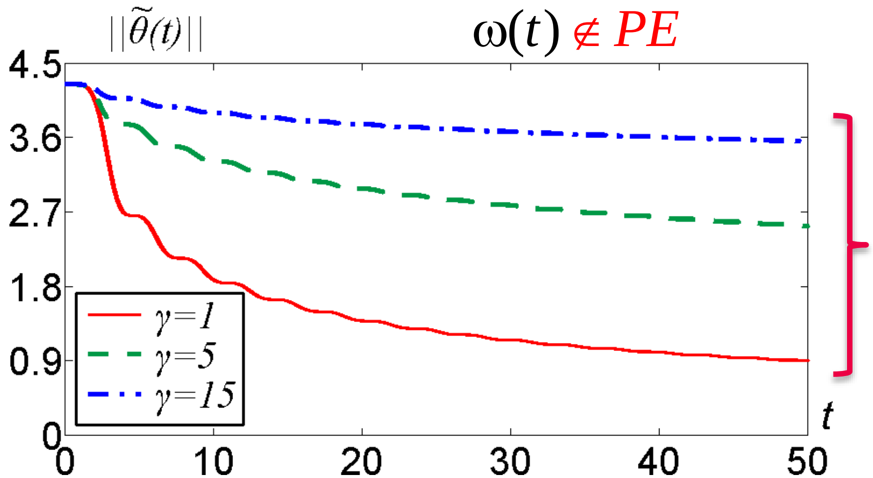
Adaptation algorithm with memory regression

$$\dot{\hat{\theta}}=\gamma\Big(L(s)[\omega y]-L(s)\big[\omega\omega^T\big]\hat{\theta}\Big),\qquad L(s)=\frac{1}{s+1}$$

Adaptation algorithm with dynamic regressor extension

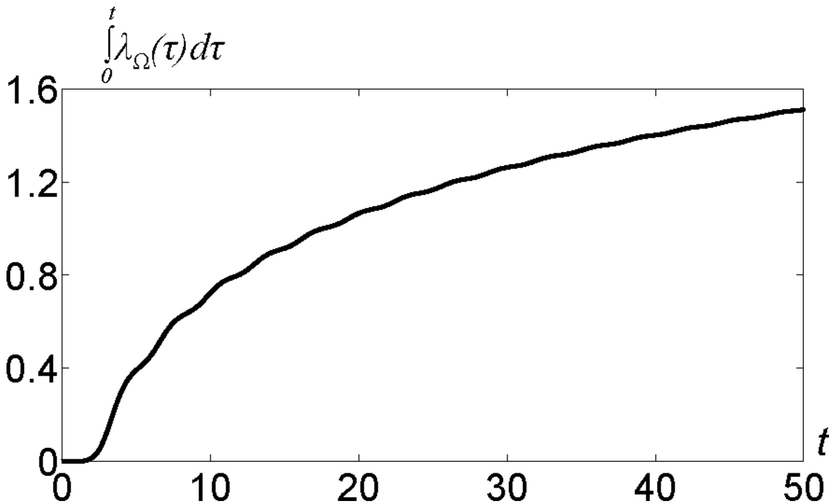
$$\dot{\hat{\theta}}=\gamma\Big(H(s)\big[\omega^T\big]\Big)^T\Big(H(s)[y]-H(s)\big[\omega^T\big]\hat{\theta}\Big),\quad H(s)=\begin{bmatrix}1\\ \dfrac{1}{s+1}\end{bmatrix}$$

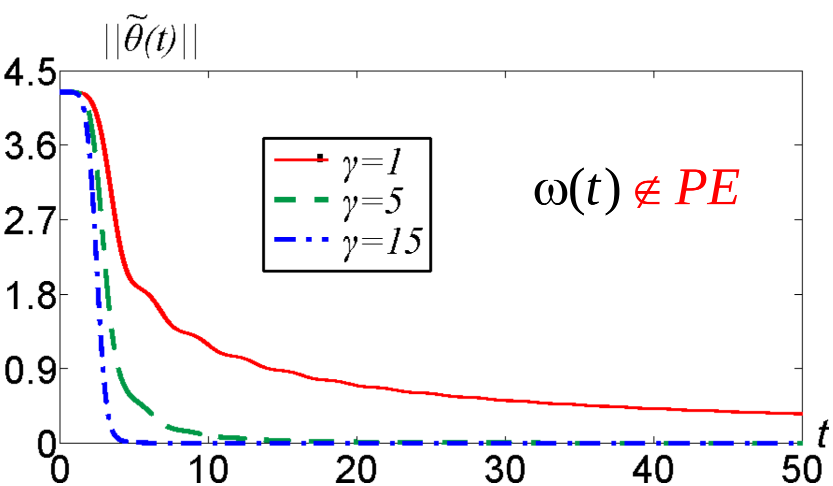
Basic gradient algorithm



Do not converge to zero

Adaptation algorithm with memory regression





Adaptation algorithm with dynamic regressor extension