So,

$$\dot{V} = -k_1 x_1^2 - k_2 x_2^2 \leq 0$$

By LaSalle - Yoshizawa,

$$x_1, \Delta x_2 \to 0 \quad \text{as} \quad t \to \infty$$

$$\Delta\alpha \text{ bounded}$$

[Notice that the system dynamics are autonomous since just stabilizing to the origin. By LaSalle, system stabilizes to largest invariant set, which is the origin. So, $x_1, \Delta x_2, \Delta\alpha \to 0$ and $x_1, x_2, \Delta\alpha \to 0$.

$\boxed{5}$ Control laws:

$$u = -x_1 - k_2 [x_2 - k_1 x_1 - \hat{\alpha}\varphi(x_1)] - [k_1 + \alpha \frac{\partial \varphi(x_1)}{\partial x_1}](x_2 + \hat{\alpha}\varphi(x_1)) - \varphi(x_1)\dot{\alpha}$$

$$\dot{\hat{\alpha}} = \gamma \left[x_1 \varphi(x_1) + (k_1 + \hat{\alpha}\frac{\partial \varphi(x_1)}{\partial x_1})(x_2 + \hat{\alpha}\varphi(x_1)) \varphi(x_1) \right]$$

Concurrent Learning

Observation: Adaptive control involves

1) Controlling / stabilizing a plant

2) Estimating the plant model

Notes: ① · Usually a "stabilizing" controller exists [at least from "unstable" to almost stable, if unforced system is unstable].

↳ time constant high relative to adaptation.

How to bridge the gap?

② Done online but inspired from sys. id. literature, which is an offline process.

The enable to bridge the gap is the difference in the time scales. Let's exploit the fast/slow processing and the long/medium time-scale by incorporating offline methods.

Simplest offline method is least squares.

a) Suppose you can measure $\dot{x}$ & x, and we know u.

$$\dot{x} = Ax + Bu \quad , \quad A, B \text{ unknown.}$$

$$\dot{x} = [A \,\vdots\, B]\begin{bmatrix} x \\ u \end{bmatrix} \Big\} \text{ linear in unknowns. } \quad \text{linear in } \begin{bmatrix} x \\ u \end{bmatrix}, \text{ and linear in}$$

So we can really write it as:

$$\dot{x} = M(x,u) \cdot \begin{bmatrix} \vec{A} \\ B \end{bmatrix} \leftarrow \text{vectorized } \vec{A}$$

Example:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \bar{a}_1 & \bar{a}_2 & \bar{a}_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix}$$

$$= \bar{a}_1 x_1 + \bar{a}_2 x_2 + \bar{a}_3 u$$

$$= \begin{bmatrix} x_1 & 0 \\ 0 & x_1 \end{bmatrix} \bar{a}_1 + \begin{bmatrix} x_2 & 0 \\ 0 & x_2 \end{bmatrix} \bar{a}_2 + \begin{bmatrix} u & 0 \\ 0 & u \end{bmatrix} \bar{a}_3$$

$$= \begin{bmatrix} x_1 & 0 & x_2 & 0 & u & 0 \\ 0 & x_1 & 0 & x_2 & 0 & u \end{bmatrix} \begin{bmatrix} \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \end{bmatrix}$$

Now, if x is $\mathbb{R}^n$ and b is $\mathbb{R}^n$, then have n^2+n unknowns. A single observation of $x, \dot{x}$ and u provides n equations. So we need $(n+1)$ observations (at least).

So we set it up as:

$$\left.\begin{aligned} \dot{x}(1) &= M(x(1), u(1))\, \vec{d} \\ \dot{x}(2) &= M(x(2), u(2))\, \vec{d} \\ &\vdots \\ \dot{x}(n+1) &= M(x(n+1), u(n+1))\, \vec{d} \end{aligned}\right\} \Rightarrow \begin{bmatrix} \dot{x}(1) \\ \dot{x}(2) \\ \vdots \\ \dot{x}(n+1) \end{bmatrix} = \begin{bmatrix} M(x(1), u(1)) \\ M(x(2), u(2)) \\ \vdots \\ M(x(n+1), u(n+1)) \end{bmatrix} \vec{d} \Rightarrow \vec{\dot{X}} = M(X, U)\, \vec{d}$$

Since there is uncertainty in the measurement, best to collect more data:

$$\underset{\substack{\uparrow \\ n\cdot N_s \times 1}}{\vec{\dot{X}}} = \underset{\substack{\uparrow \\ n\cdot N_s \times (n^2+n)}}{M(X,U)}\,\vec{d} \leftarrow (n^2+n)\times 1$$

N_s = # samples collected $> n+1$

Least squares problem

$$\hat{d} = \underset{\vec{d}}{\operatorname{argmin}} \| \vec{\dot{X}} - M(X,U)\vec{d}\,\|^2$$

$$\hat{d} = \left[M^T(X,U)\,M(X,U)\right]^{-1} M^T(X,U)\,\vec{\dot{X}}$$

Damped Least Squares

$$\hat{d} = \underset{\vec{d}}{\operatorname{argmin}} \| \dot{X} - M(X,U)\vec{d}\,\|^2 + \rho^2 \|\vec{d}\,\|^2$$

$$\hat{d} = \left(M^T(X,U)\,M(X,U) + \rho^2 I\right)^{-1} M^T(X,U)\,\dot{X}$$

b) Can do the same for the non-linear terms.

$$\dot{x} = Ax + b(u + \underset{\uparrow\ \alpha^T \Phi(x)}{f(x)})$$

With $\dot{x}$, x, u are measured, A, b known. Then,

$$b f(x) = \dot{x} - Ax - bu$$
$$b \alpha^T \Phi(x) = \dot{x} - Ax - bu$$
$$\underset{\substack{\uparrow \\ N \text{ unknowns}}}{b \Phi(x)^T \alpha} = \dot{x} - Ax - bu \quad , \quad x \in \mathbb{R}^n, \ u \in \mathbb{R}$$

N unknowns : in practice 1 equation, it's a singe rank problem by virtue of being singe input.

So N observations are needed to solve the system.

$$\vec{\alpha} = [b \Phi^T(x)]^+ (\dot{x} - Ax - bu) \qquad \text{for one observation.}$$

Or with N observations:

$$b \Phi^T(x(1)) \alpha = \dot{x}(1) - Ax(1) - bu(1)$$
$$\vdots$$
$$b \Phi^T(x(N)) \alpha = \dot{x}(N) - Ax(N) - bu(N)$$
$$\Downarrow$$
$$\bar{b} [\Phi(X)]^T \alpha = \dot{X} - \bar{A} X - \bar{b} U$$

with $\bar{A} = \operatorname{diag}(A, A, \ldots, A)$
$\bar{b} = \operatorname{diag}(b, b, \ldots, b)$

The matrix inversions required to solve parameter estimation problem prevents real-time use.

Idea: Merge adaptive control with least squares estimation.

Plant:

$$\dot{x} = Ax + b\left[u + (\alpha^*)^T \Phi(x)\right]$$

Adaptive:

$$\dot{x}_m = A_m X + b_m r$$

$$u = K_x^T x + K_r^T r - \hat{\alpha}\,\Phi(x)$$

Least Squares Plant

Collected Input: $X = \{x_j\}_{j=1}^{N_s}$

Collected Output $\mathcal{Y} = \{y_j\}_{j=1}^{N_s}$

$V(y,\beta)$

Presume existence of α^*:

$$y_j = (\alpha^*)^T \Phi(x_j)$$

$$\downarrow$$

$$\alpha\,\Phi(x) - y = \alpha^T \Phi(x) - (\alpha^*)^T \Phi(x)$$

$$= (\alpha - \alpha^*)^T \Phi(x)$$

$\widehat{V(y,\beta)}$

$$= \Delta\alpha^T \Phi(x)$$

Collect at least N samples (unique). Now, we want to merge them.

$$V_{CL}(e, \Delta k_x, \Delta k_r, \Delta\alpha) = V_{AD} + V_p$$

So,

$$V_{CL} = \frac{1}{2}e^TPe + \frac{1}{2}\Delta k_x^T \Gamma_x \Delta k_x + \frac{1}{2}\Delta k_r^T \Gamma_r^{-1}\Delta k_r + \frac{1}{2}\Delta\alpha^T\Gamma_\alpha \Delta\alpha$$

We have as before:

$$\begin{aligned}\dot{V}_{CL} = -e^TQe &+ |b|\Delta k_x^T\left[xe^TPb + \Gamma_x^{-1}\Delta\dot{k}_x\right] \\ &+ |b|\Delta k_r^T\left[re^TPb + \Gamma_r^{-1}\Delta\dot{k}_r\right] \\ &+ |b|\Delta\alpha^T\left[-\Phi(x)e^TPb + \Gamma_\alpha^{-1}\Delta\dot{\alpha}\right]\end{aligned}$$

Cancel + Add in beneficial terms

Data fitting terms:

$$-\sum \Phi(x_j)\Phi(x_j)^T\Delta\alpha$$

So, let:

$$\dot{\hat{\alpha}} = \Delta\dot{\alpha} = \Gamma_\alpha\Phi(x)e^TPb - \sum_{j=1}^{N_s}\Gamma_\alpha\Phi(x_j)\Phi(x_j)^T\Delta\alpha$$

Then,

$$\dot{V} = -e^TQe - |b|\Delta\alpha^T\left[\sum_{j=1}^{N_s}\Phi(x_j)\Phi^T(x_j)\right]\Delta\alpha$$

By Barbalat,

$$e \to 0, \quad \Delta\alpha \to 0.$$

So

$\alpha(t)$ is current estimate

$$\alpha^T(t)\,\Phi(x(j)) = y(j) + \varepsilon(j)$$

$$\parallel$$

$$(\alpha^*)^T\,\Phi(x(j))$$

$$\Delta\alpha^T\,\Phi(x(j)) = \varepsilon(j)$$

$$\Phi^T(x(j))\Delta\alpha = \varepsilon(j)$$

$$\Sigma\,\Phi(x(j))\left(\Phi^T(x(j))\,\Delta\alpha\right)$$

$$\Sigma\,\Phi(x(j))\,\varepsilon(j)$$

$$\Rightarrow \quad \dot{\hat{\alpha}} = \Gamma_\alpha\,\Phi(x)\,e^TPb - \Gamma_\alpha \sum_{j=1}^{N_s} \Phi(x(j))\,\varepsilon(j)$$

So the control law is:

$$\dot{\hat{\alpha}} = \Gamma_\alpha\,\Phi(x)e^TPb - \Gamma_\alpha \sum_{j=1}^{N_s}\left[\Phi(x(j))\left(\hat{\alpha}^T(t)\,\Phi(x(j))\right) - y(j)\right]$$

$$\dot{\hat{\alpha}} = \Gamma_\alpha\,\Phi(x)e^TPb - \Gamma_\alpha \sum_{j=1}^{N_s}\left[\Phi(x(j))\left(\Phi^T(x(j))\,\hat{\alpha} - y(j)\right)\right]$$

N_p: picked

N_s: to select.

① Where does data come from?

Approach 1) Evolve real system for a little bit and take measurements

[requires stable controlled baseline system]

Construct best $\Omega(\mathcal{X}) = \sum_{j=1}^{N_s} \Phi(x(j))\, \Phi^T(x(j))$, where

$\mathcal{X} = \{x(j)\}_{j=1}^{N_s}$

What does best mean?

$$\Omega = U \Sigma V^T \qquad \text{(S.V.D.)}$$

Remember we have $\mathcal{X}_c, \mathcal{Y}_c$ ← complete data set of size N_c.

Goal: Craft $\Omega(\mathcal{X})$ from $N_s < N_c$ data points.

$\binom{N_c}{N_s}$ → factorial is horrible op.

Greedy approach:

Look at $\Phi(x(j))\Phi^T(x(j))$, pick best

Pick 1) Biggest norm. $\mathcal{X} = \{x(j^*)\}$

Pick 2) Given $\mathcal{X}$ has N_p elements look at

$$\Omega(\mathcal{X}) + \Phi(x(j))\Phi^T(x(j))$$

with remaining j. Notice $N_p < N_s$.

You only have to check (N_p+1)th s.v. Select the j that gives highest σ_{N_p+1}

Pick 3) Repeat until $N_p = N$

Approach 2) Online

① Collect data and accept it if $N_p < N_s$, or in other words:

$$|\mathcal{X}| < N_s$$

Cardinality of $\mathcal{X}$ is # elements

② Look at

$$[\underbrace{\Omega(\mathcal{X}_{-1})} + \Phi(\mathcal{X}(\underset{\text{current measurement}}{\tau}))\Phi(x(\tau))^T]$$

$\mathcal{X}_{-1}$ is $\mathcal{X}$ with one element removed.

and throw out the measurement that does not contribute as much. It must satisfy $1 \le j^* \le N_s$. Replace $x(\tau)$ into $x(j^*)$. Might be none.

③ Repeat 2 each measurement.

Only problem to this approach is in:

$$\dot{x} = Ax + b(u + \alpha\Phi(x))$$

$$b\alpha\Phi(x) = \dot{x} - Ax - bu$$ ⇒ we assume have access to RHS

$\downarrow$

Introduce a smoother.

Concurrent Learning

Consists of two "parallel" processes:

1) Standard adaptive control system + smoother

2) The data selection process
   i) Does data fit model

$$\frac{\| \Phi(x(t)) - \Phi(x(\tau_{N_s}))\|}{\| \Phi(x(t))\|} < p,$$

if not, move on...

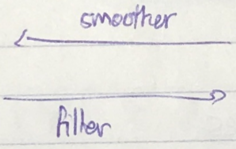
   ii) If not, then see if the data should be incorporated. Use SVD criterion.

Now, what about getting $\dot{x}$? Entire approach conditioned on being able to measure x and $\dot{x}$

In some cases a part of $\dot{x}$ may be available. For example,

IMU → acceleration (linear) $\quad \dot{\begin{Bmatrix} pos \\ vel \end{Bmatrix}} = \begin{Bmatrix} vel \\ acc \end{Bmatrix}$

A smoother is able to generate estimates of $\dot{x}$ given measurements x.



Kalman Filter

$$\xi = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}$$

$$\dot{\xi} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \xi$$

(In discrete time: $\hat{\xi}_k = F_k \hat{\xi}_{k-1}$, where

$F_k = e^{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \Delta t}$ is constant).

We will have:

Prediction • $\hat{\xi}_k^- = F_k \hat{\xi}_k^+$

$$\hat{Z}_k = [1\ \ 0]\,\xi_k = H_k \xi_k$$

$$P_k^- = F_k P_{k-1}^+ F_k^T + \underset{\uparrow\text{covariance of process}}{Q_k}$$

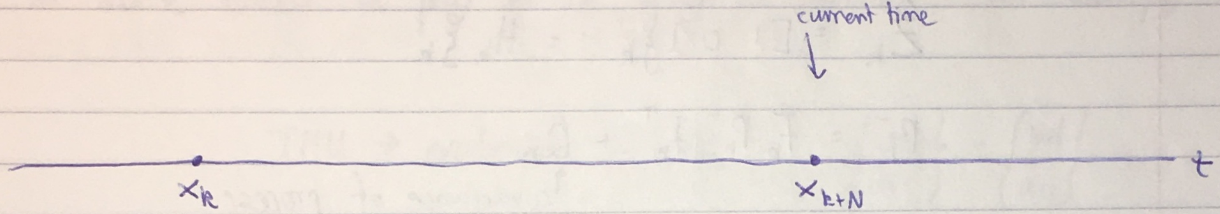
Correction: • $K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + \underset{\uparrow\text{covariance of measurement}}{R_k}\right]^{-1}$

$$\hat{\xi}_k^+ = \hat{\xi}_k^- + K_k[Z_k - \hat{Z}_k]$$

$$P_k^+ = [I - K_k H_k]\,P_k^-$$

Kalman Smoother

Fixed lag smoothing:



Takes the forward filter equations & adds the smoother: it is a bit like an estimator for the N states of the N steps. The state is:

$$\xi_{k|k+N}$$

Initialization:

$$\xi^{s}_{k|k+N} = \xi^{+}_{N}$$

$$P^{s}_{k|k+N} = P^{+}_{N}$$

$$M_{k|k+N} = I$$

Update:

$$M_{k+1|k+N+1} = M_{k|k+N} K^{s}_{k+N}$$

$$K^{s}_{k+N} = P^{+}_{k+N} A^{T}_{k+N} \left[P^{-}_{k+N+1} \right]^{-1}$$

$$P^{s}_{k+1|k+N+1} = P^{-}_{k+1} - \left[K^{s}_{k} \right]^{-1} \left(P^{+}_{k} - P^{s}_{k|k+N} \right) \left(K^{s}_{k} \right)^{-T}$$

$$- M_{k+1|k+N} K_{k+N+1} H_{k+N+1} P^{-}_{k+N+1} M_{k+1|k+N+1}$$

$$\xi^{s}_{k+1|k+N+1} = F_{k} \xi^{s}_{k|k+N} + B_{k} u_{k} + Q_{k} F_{k}^{-T} \left(P^{+}_{k} \right)^{-1} \left[\xi^{s}_{k|k+N} - \xi^{+}_{k} \right]$$

$$+ M_{k+1|k+N+1} K_{k+N+1} \left[z_{k+N+1} - H_{k+N+1} \xi^{-}_{k+N+1} \right]$$