ECE 5551 Passino Part 3

$$\underline{\underline{\Phi^T W Y}}$$

$$\Phi(N+1)\, W(N+1)\, Y(N+1) = \left[\phi(n) \;\cdots\; \phi(N) \;\; \phi(N+1) \right] \cdot \begin{bmatrix} a\gamma^{N+1-n} & & & 0 \\ & \ddots & & \\ & & a\gamma & \\ 0 & & & a \end{bmatrix} \begin{bmatrix} y(n) \\ \vdots \\ y(N+1) \end{bmatrix}$$

and splitting into <u>two</u> <u>terms</u>

$\odot$

$$\Phi^T(N+1)W(N+1)Y(N+1) =$$

$$\gamma\,\Phi^T(N)\,W(N)Y(N) + \phi(N+1)a\,y(N+1)$$

Use $\odot$'s in $\star$ (Let $P(N)=P$, $\phi(N+1)=\phi$, $y(N+1)=y$)

$$\hat{\theta}_{WLS} = \left[\frac{P}{\gamma} - \frac{P}{\gamma}\phi\left(\frac{1}{a} + \phi^T\frac{P}{\gamma}\phi\right)^{-1}\phi^T\frac{P}{\gamma}\right]\left[\,\cdot\,\right]$$

$$\boxed{\gamma\,\Phi^T W Y(N) + \phi a y}$$

Here, $P\Phi^T W Y(N) = \hat{\theta}_{WLS}(N)$ (estimate using N data)

So,

$$\hat{\theta}_{WLS}(N+1) = \hat{\theta}_{WLS}(N) + \frac{P}{\gamma}\phi a y -$$

$$\frac{P}{\gamma}\phi\left(\frac{1}{a} + \phi^T\frac{P}{\gamma}\phi\right)\phi^T\frac{P}{\gamma}\phi\, a y$$

insert

$$\underbrace{\left(\frac{1}{a} + \phi^T\frac{P}{\gamma}\phi\right)}_{1\times 1}^{-1}\left(\frac{1}{a} + \phi^T\frac{P}{\gamma}\phi\right) = \underset{1\times 1}{1}$$

Get, with algebra,

$$(1)\quad \hat{\theta}_{WLS}(N+1) = \hat{\theta}_{WLS}(N) + \underbrace{L(N+1)\left(y(N+1) - \phi^T\hat{\theta}_{WLS}(N)\right)}_{\text{corrector}}$$

* MIL makes this possible → | Trivial to compute → Scalar

Here,

* ② $$L(N+1) = \frac{P}{\gamma}\phi\left(\frac{1}{a} + \frac{\phi^T P \phi}{\gamma}\right)^{-1}$$

From earlier,

③ * $$P(N+1) = \frac{P(N)}{\gamma} - \frac{P(N)}{\gamma}\phi\left(\frac{1}{a} + \phi^T\frac{P(N)}{\gamma}\phi\right)^{-1}\phi^T\frac{P(N)}{\gamma}$$

so that

$$P(N+1) = \frac{P(N)}{\gamma} - L(N+1)\phi^T\frac{P(N)}{\gamma}$$

①, ②, ③ allow for computation of $\hat{\theta}_{WLS}$ (in real-time, "recursively", have N data, get one more, above shows $\boxed{\hat{\theta}_{WLS}(N+1)}$ ← same as for RLS *

* Initial condition options

- Get $N > 2n$ data, use batch to solve for $P(N)$, $L(N+1)$, $\hat{\theta}(N)$

- What I try first,

  let $\hat{\theta}_{wls}(N) = 0$, pick

  $P(N) = \alpha I$,

  $\alpha > 0$ is a large scalar

  (e.g. $\alpha = 10$, 1000, or 10000.)

Development as before, but with

$$J = \sum e^T w e$$

$$e(k), \quad p \times 1$$

$$w(k) \quad p \times p, \text{ nonsingular matrices}$$

Consider,

$$L(N+1) = \frac{P}{\gamma}\Phi\left(a^{-1} + \Phi^T \frac{P}{\gamma}\Phi\right)^{-1}$$

$$P(N+1) = \ldots$$

$$\hat{\theta}_{WLS}(N+1) = \ldots$$

Getting back to the Kalman Filter

- Let $\gamma = 1$

$$a = R_v^{-1} \quad \text{(assumed invertible)}$$

- Change notation

$$\hat{x}(k) = \bar{x}(k) + L(k)\left(y(k) - H\bar{x}\right)$$

and from above,

$$L(k) = M(k)H^T\left(HM(k)H^T + R_v\right)^{-1}$$
$$= P(k)H^T R_v^{-1} \quad \text{(with algebra)}$$

- $M(k)$ is the covariance of $\bar{x}(k)$, before measurement

- State estimate, after measure, $\hat{x}(k)$, has a error covariance P,

$$P(k) = M(k) - M(k) H^T \left(H M(k) H^T + R \right)^{-1} H M(k)$$

Compared to RLS,

$$\hat{\theta}_{WLS}(N+1) \longleftarrow \hat{x}(k)$$

$$\hat{\theta}_{WLS}(N) \longleftarrow \bar{x}(k)$$

$$\Phi^{T} \longleftarrow H$$

$$P(N) \longleftarrow M(k)$$

$$P(N+1) \longleftarrow P(k)$$

$$a^{-1} \longleftarrow R_v$$

- Convert as parameter estimation problem into a <u>state</u> estimation problem
- This is an optimal estimator in the same sense that WLS is optimal

Notes

① Choose as a predictor

$$\bar{x}(\underset{=\,k+1}{k}) = \Phi \hat{x}(\underset{=\,k}{k-1}) + \Gamma u(\underset{=\,k}{k-1})$$

where let $w(k) = 0$, $k \geq 0$

as $E[w(k-1)] = 0$ (zero mean)

② $$\left(x(k+1) - \bar{x}(k+1)\right) = \Phi\left(x(k) - \hat{x}(k)\right) + \Gamma_1 w(k)$$

Covariance of the state at $k+1$, <u>before</u> taking into account $y(k+1)$

$$M(k+1) = E\left[\left(x(k+1) - \bar{x}(k+1)\right)\left(x(k+1) - \bar{x}(k+1)\right)^T\right]$$

If $v(k)$ and $w(k)$ are uncorrelated, cross-terms in are $= 0$

$$\Rightarrow \quad M(k+1) = E\left[\left(\Phi\left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T \Phi^T + \Gamma_1 w(k) w^T(k) \Gamma_1^T\right)\right]$$

But $\quad P(k) = E\left[\left(x(k) - \hat{x}(k)\right)\left(x(k) - \hat{x}(k)\right)^T\right]$

$$R_w = E\left[w(k) w^T(k)\right]$$

so $\quad M(k+1) = \Phi P(k) \Phi^T + \Gamma_1 R_w \Gamma_1^T$

Kalman Filter

Measurement update

$$\hat{x}(k) = \bar{x}(k) + \overbrace{P(k) H^T R_v^{-1}}^{\text{"Kalman gain" (time-varying gain)}} \left(y(k) - H \bar{x}(k) \right)$$

$$P(k) = M(k) - M(k) H^T \left(H M(k) H^T + R_v \right)^{-1} H M(k)$$

Like the "current estimator" — but KF takes into account w, v (noise); minimizes the estimation error $(x - \hat{x})$

* Needed in coding KF in Matlab

Between measurements ("time update")

$$\bar{x}(k+1) = \Phi \hat{x}(k) + \Gamma u(k)$$

(need $\bar{x}(0)$ to get $\hat{x}(0)$)

$$M(k+1) = \Phi P(k) \Phi^T + \Gamma_1 R_w \Gamma_1^T$$

(need $M(0)$)

(a) $\bar{x}(0)$ – guess at the state estimate (could use $\bar{x}(0) = 0$, if know nothing)

* Also, need to simulate the process

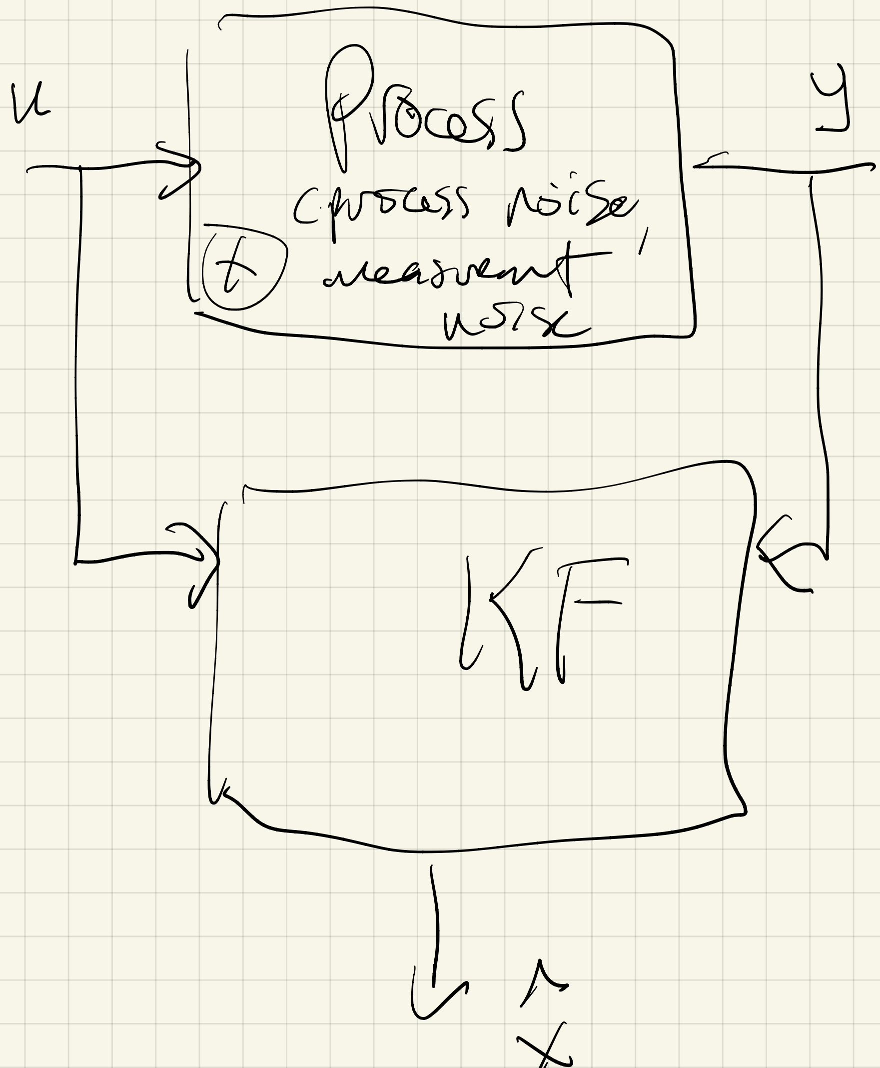
$x(k+1) = \Phi x(k) + \Gamma u(k)$, $y = Hx(k)$

(b) $M(0) = \mathbb{E}\left[(x(0) - \bar{x}(0))(x(0) - \bar{x}(0))^T \right]$

- a priori estimate of the accuracy of $\bar{x}(0)$

→ just guess at it (e.g. diagonal βI, $\beta \gg 0$)

Where are we at?



(tries to minimize $(x - \hat{x})$)

What if the process model is nonlinear?

If nonlinear process $\rightarrow$ use the "Extended Kalman Filter"

- Linearizes the process model at each time
- Uses the linear KF

Steady-state optimal estimation

- Get a constant-gain estimator
- Compare ☆ (P and M) to result from optimal control

$$M(k) = S(k) - S(k)\Gamma\left(Q_2 + \Gamma^T S(k)\Gamma\right)^{-1}\Gamma^T S(k)$$

$$S(k) = \Phi^T M(k+1)\Phi + Q_1$$

Same form! (M compared to P and S compared to M)

Control – estimation duality

Control	Estimation
Φ	Φ^T
M	P
S	M
Q_1	$\Gamma_1 R_W \Gamma_1^T$
Γ	H^T
Q_2	R_V

Gather equations for optimal control case, and using the Hamiltonian, can show that the optimal Kalman gain is

Controller that takes into account noise $w(k)$ and $v(k)$

$$\underbrace{L_\infty}_{} = M_\infty H^T (H M_\infty H^T + R_J)^{-1}$$

called "linear quadratic Gaussian" (LQG)

① - easily computed in Matlab

② - can create an "optimal regulator" (based on opt controller + opt. est.)

③ - could add ref. input or integrator

Nonlinear Stability Analysis Using Lyapunov Functions

NODE:

$V(x)$ is a Lyapunov function for

$$x(k+1) = f(x), \quad f(0) = 0 \quad ⊛$$

equilibrium

Where does this come from?

if the following conditions hold:

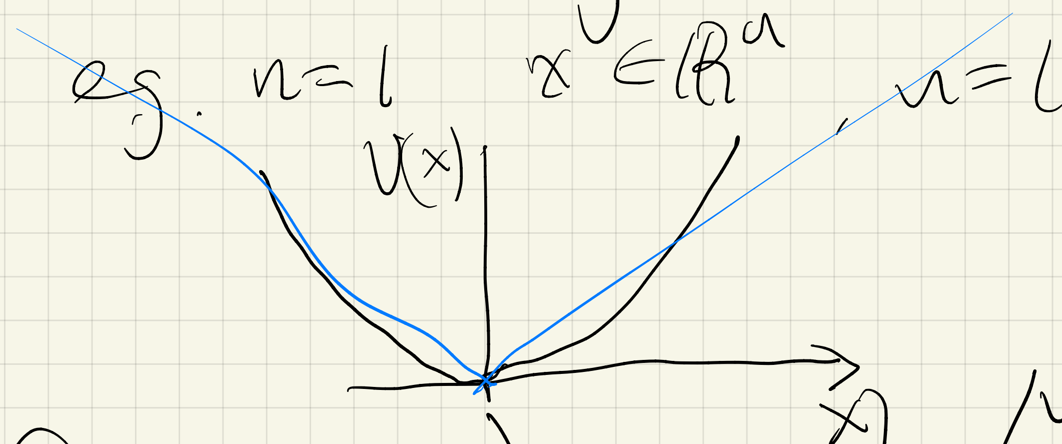
1. $V(0) = 0$, $V(x)$ continuous in x

2. $V(x)$ is "positive definite" — that is,

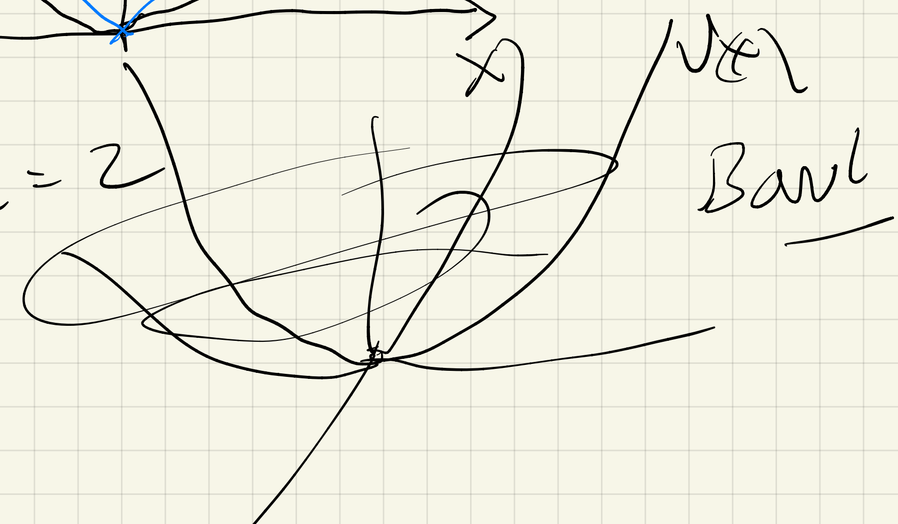
$$V(x) \geq 0 \text{ with } V(x) = 0 \text{ only if } x = 0$$

$x \in \mathbb{R}^n$

① e.g. $n=1$



② e.g. $n=2$



3. $\Delta V(x) = \underbrace{V(f(x^{(k)})) - V(x^{(k)})}_{x(k+1) = f(x(k))}$ is negative definite,

$= V(x(k+1)) - V(x(k))$

That is, $\boxed{\underbrace{V(f(x)) - V(x)}_{\Delta V(x)} < 0 \qquad \Delta V < 0}$

with $\Delta V(x) = 0$ only if $x = 0$

$V(f(x)) < V(x) \quad \forall x,\ x \neq 0$

- Solution $x(t) = 0$ (equilibrium) for ⊛ is

asymptotically stable

if there exists a Lyapunov function in x ($V(x)$)

- Solution $x(t) = 0$ is

globally asymptotically stable

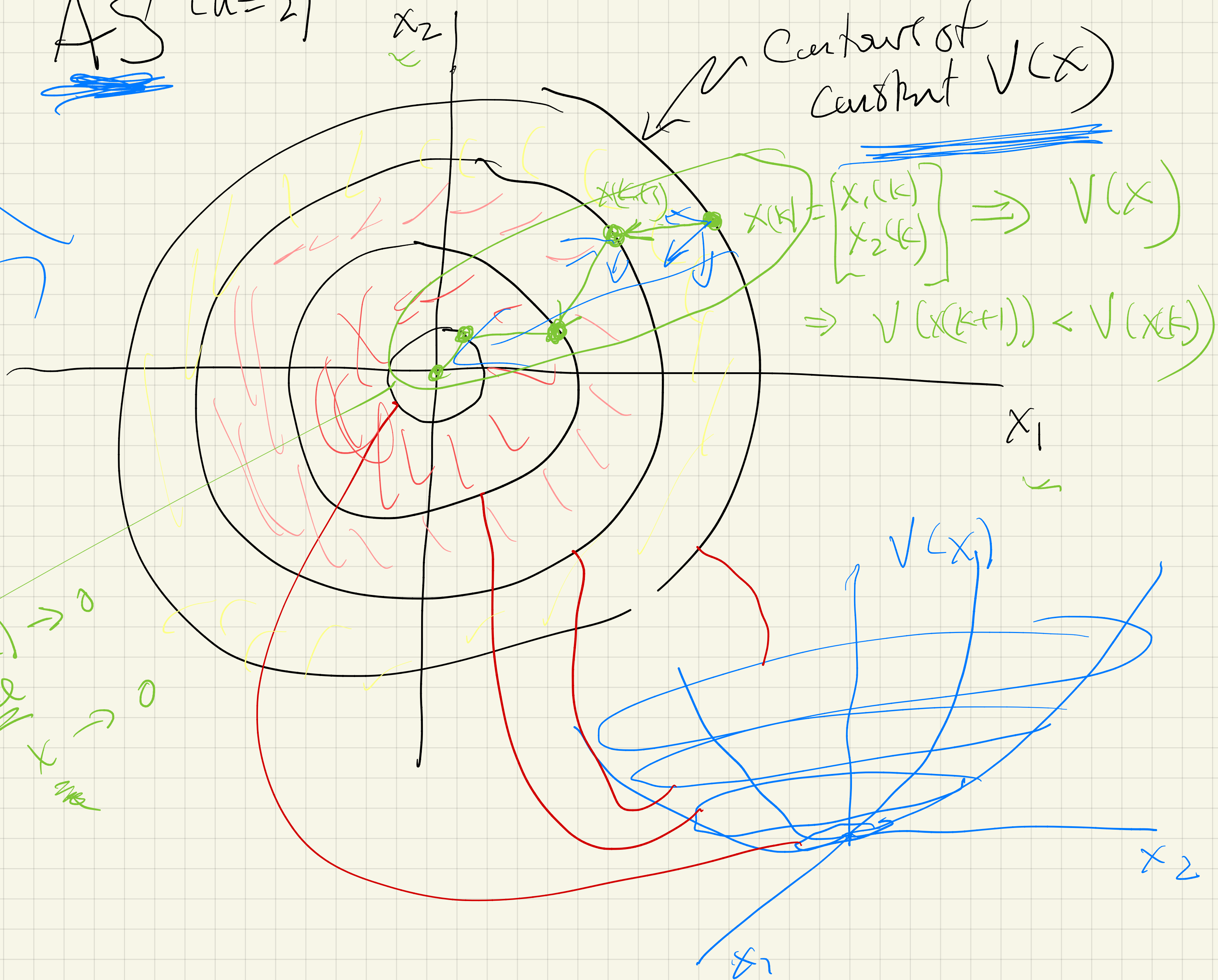
if, in addition,

$$0 < \phi(\|x\|) < V(x) \qquad \|x\| = \sqrt{x^T x}$$

where $\lim_{\|x\| \to \infty} \phi(\|x\|) = \infty$ (and above cond. hold)

Another way to view how $V(x)$ has a role in AS $(n=2)$

In Matlab (contour)



$\Rightarrow V(x) \to 0$ and $x \to 0$

Example of Application of Lyapunov theory:

- Apply to LQR
- Steady state solution of Ricatti eq. gives

$$\underline{u = -Kx} = \underline{-(Q_2 + \Gamma^T S_\infty \Gamma)^{-1} \Gamma^T S_\infty \Phi x}$$

<u>Question:</u> Does K give us a (globally) asymptotically stable (AS) system

$$x(k+1) = \underbrace{(\Phi - \Gamma K)}\, x(k)$$

Let cost for control be

$$J = x^T S_\infty x > 0$$

and consider the Lyapunov function

$$V(x(k)) = x^T(k) S_\infty x(k)$$

Here $\underset{n \times n}{S_\infty} > 0$, i.e. it is "positive definite"

$\Rightarrow$ eig (S_∞) are all positive

Need to show that $\Delta V(x)$ is negative definite,

i.e. $\Delta V(x) < 0 \quad \forall x$

$$\Delta V(x) = x^T(k+1)\, S_\infty\, x(k+1) - x^T(k)\, S_\infty\, x(k)$$

$$= \underbrace{x^T(k)\,(\Phi - \Gamma K)^T}\, S_\infty\, \underbrace{(\Phi - \Gamma K)\, x(k)} - x^T(k)\, S_\infty\, x(k)$$

Doing some algebra...

$$= -x^T(k)\,(Q_1 + K^T Q_2 K)\, x(k)$$

to the last step substitute in S_∞,

$$S_\infty = (\Phi - \Gamma K)^T S_\infty (\Phi - \Gamma K) + Q_1 + K^T Q_2 K$$

Since $(Q_1 + K^T Q_2 K) > 0$ (p.d.) $\nabla V(x)$ is negative definite $\Rightarrow$

The closed-loop LQR is (globally) AS

$\Rightarrow$ The choice of $u = -Kx$
where K is chosen in the LQR
formulation, results in

$$x(k+1) = \underline{(\Phi - \Gamma K)}\, x(k)$$

being AS

Know that $\text{eig}(\Phi - \Gamma K)$ will all be in the unit disk

Here: for all initial condition $x(0)$

- $x(0)$ is stable
- $x(k) \to 0$ as $k \to \infty$ for all $x(k) \in \mathbb{R}^n$

Summary of where we are at ...

① Modeling

- Physics - based
- Data - based (sysid)

$$\hookrightarrow \begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) + \Gamma_1 w(k) \\ y(k) = H x(k) \end{cases}$$

② Pole - placement design

- Control, $u = -Kx$
- Estimators

③ Optimal control and estimation (MIMO)

- LQR, design (Q_1, Q_2), AS
- Estimation ~ Kalman filter

④ Topics in feedback control

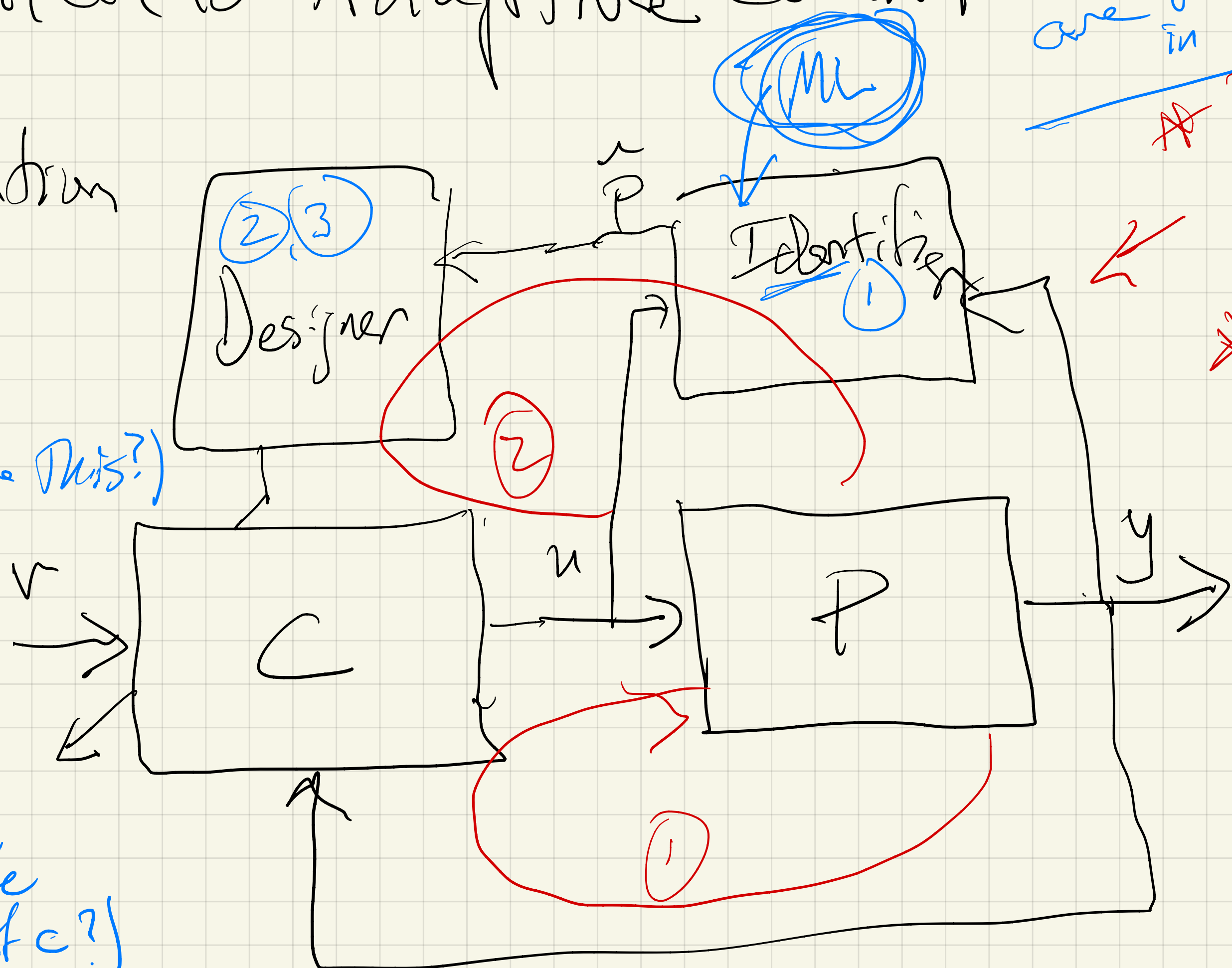
More theory, more application can be considered

- ECE 6754 Stability Theory (stab. defs., Lyapunov for LQR)
- ECE 7854 Adaptive Control *
- ECE 7858 Intelligent control

Introduction to Adaptive Control:

① - ② - ③ are major divisions in class

- Motivation



- Can model P using u, y info.
(can we automate this?)
- Can systematically design C if given P
(can we automate the design of C?)

* Done in real time
* It learns how to control P
* Indirect adaptive control
* Learning control

Two loops - ① & ②

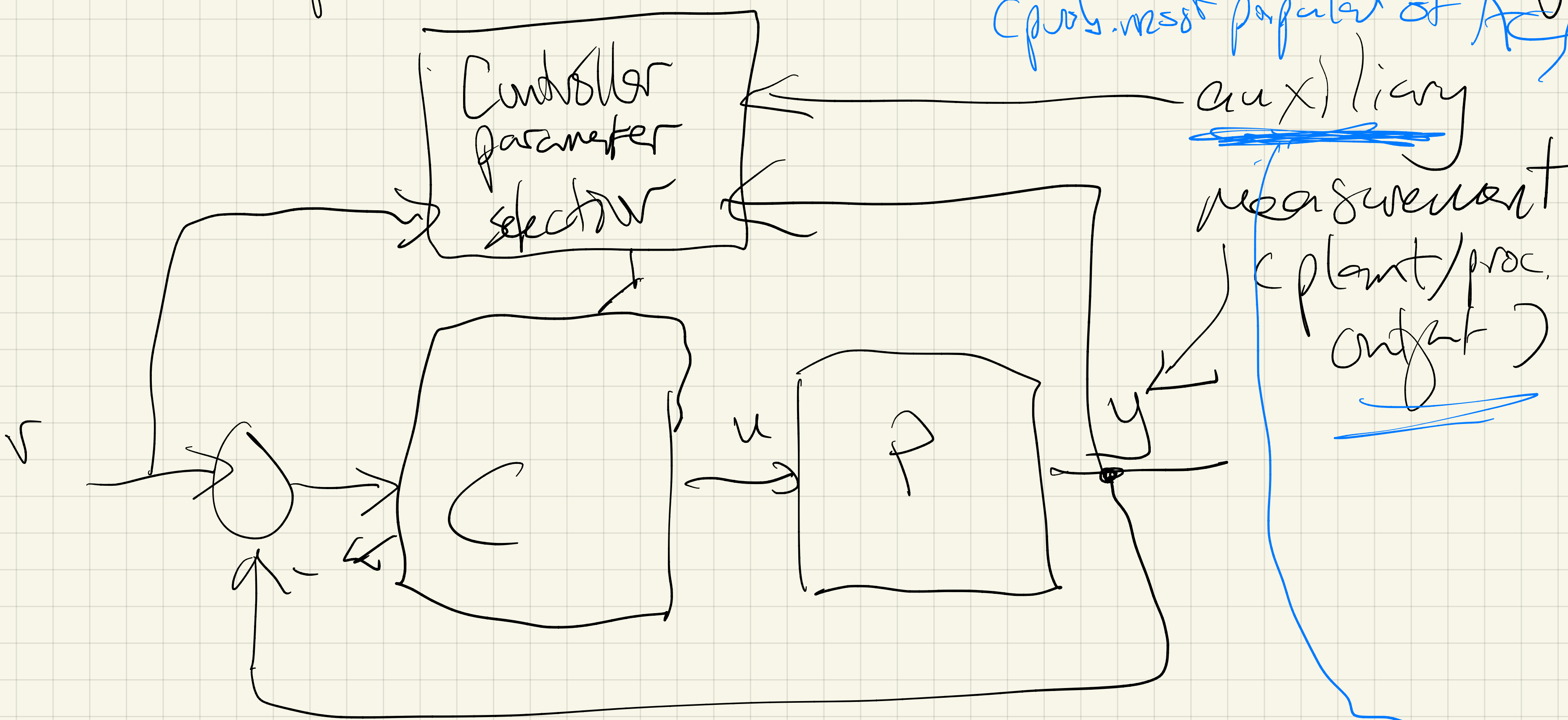
↳ u, y → P → C

It's just an automation of what you learned in class

⇒ On-line machine learning [of process]:

Another adaptive control method: "Gain Scheduling"

(prob. most popular of AC)



Ex Aircraft flight control

- pick an "operating point"

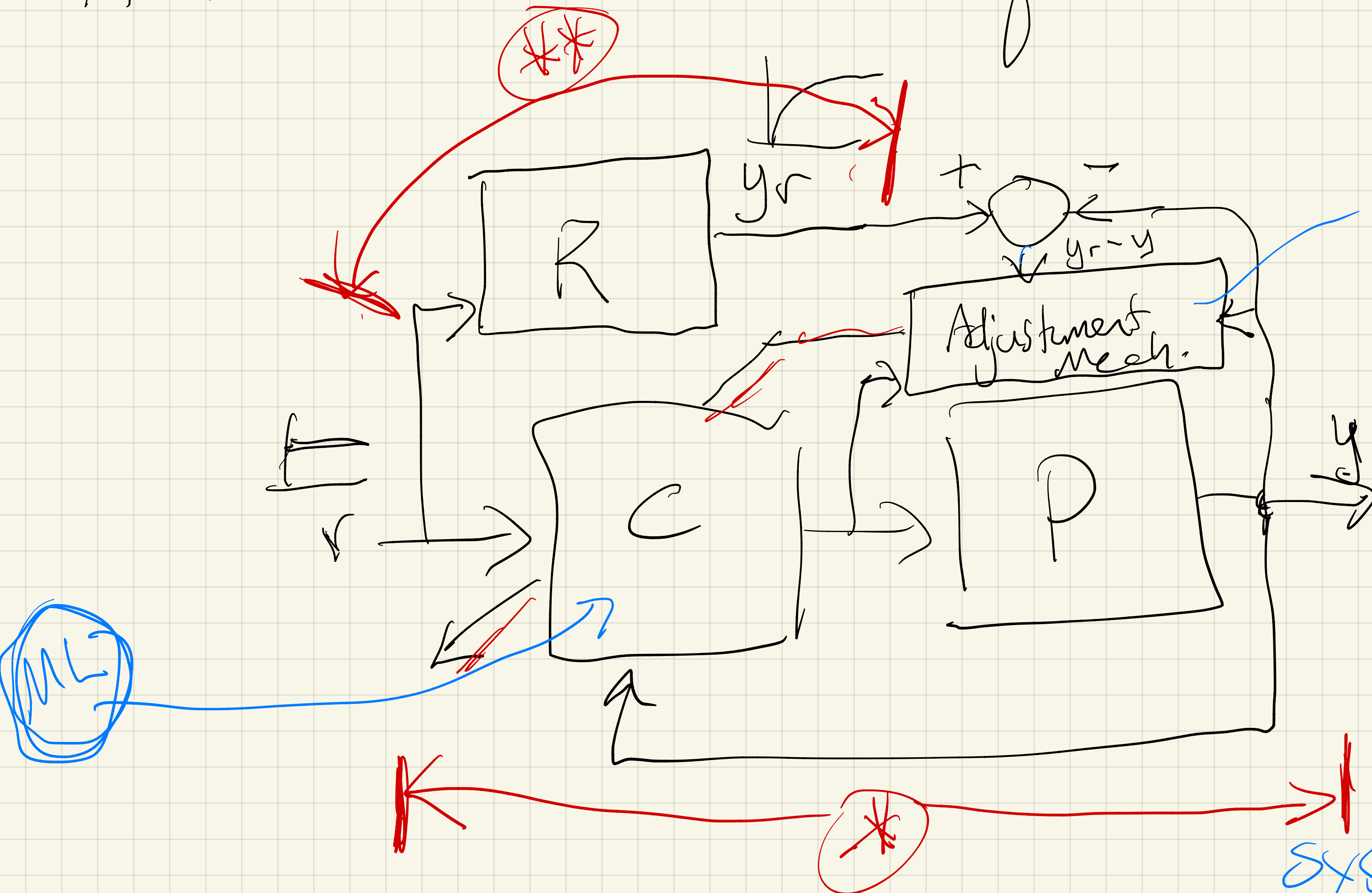
(altitude, temperature, ...)

* Design a linear controller for each such point

- During flight, interpolate between points

Also, many other systems

Answer: Model reference adaptive control (MRAC)



AM

⇒ make $y_r - y \to 0$

⇒ try to get the closed-loop system to behave like R

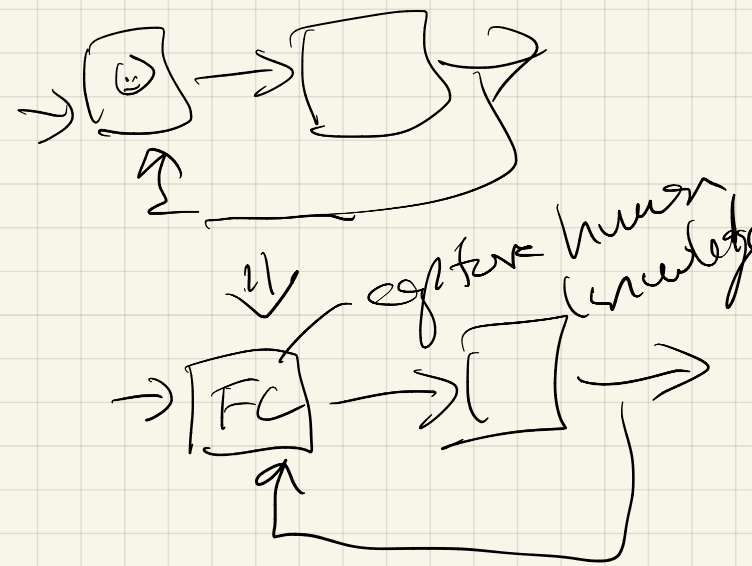
(no process ID needed!)

Want to adjust C so that (*) is the same as (**)

On-line machine learning (of controller)

Intelligent Control:

Biomimicry book... — 2006



- Fuzzy control
- Planning ("MPC")
- Attentional sys. 7858
- Learning
  - LS, Grad
  - NN, FS
  - AC

  ML
  RL
- Evolution
  - GA
  - design

- Foraging
  - Swarms
  - Multiagent sys.
  - Game Theory

  ⋮

Discussion