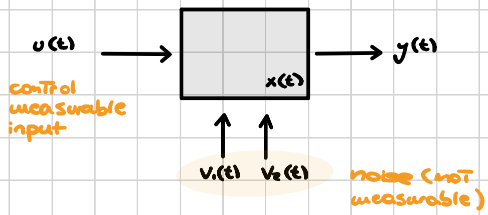
3 - Virtual model based Software Sensing in feed back using Kalman Filter

Kalman Filter is based on a state space representation of the system

$$\begin{cases} x(t+1) = Fx(t) + Gu(t) + v_1(t) \\ y(t) = Hx(t) + Du(t) + v_2(t) \end{cases}$$ noise



We assume that the system model (F, G, H) is given, typically built white-box.

Goal: given a model of the system, with KF we can solve these 3 (+1) problems

1) k-steps ahead prediction of the output

$\{u(1), \dots, u(N)\}$ $\{y(1), \dots, y(N)\}$ $\Rightarrow$ $\hat{y}(N+k \mid N)$

present time is N

estimation

future k-steps ahead

present time

Notice that this is not a new problem: we can solve this problem using the prediction theory on ARMA/ARMAX models (I/O models)

2) k-step ahead prediction of state

state is not measured $\{u(1), \dots, u(N)\}$ $\{y(1), \dots, y(N)\}$ $\Rightarrow$ $\hat{x}(N+k \mid N)$

time instant to predict

This is a new problem: with ARMAX models we cannot predict the state

3) Filter of the state

$\{u(1), \dots, u(N)\}$ $\{y(1), \dots, y(N)\}$ $\Rightarrow$ $\hat{x}(N \mid N)$

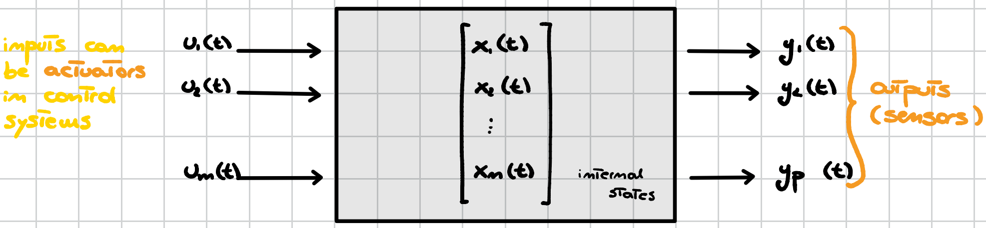
estimate the state at present time

we have data up to time N

This is the main motivation of the Kalman filter algorithm

4) Gray Box system identification (chapter 5)

Why software sensing is so useful?

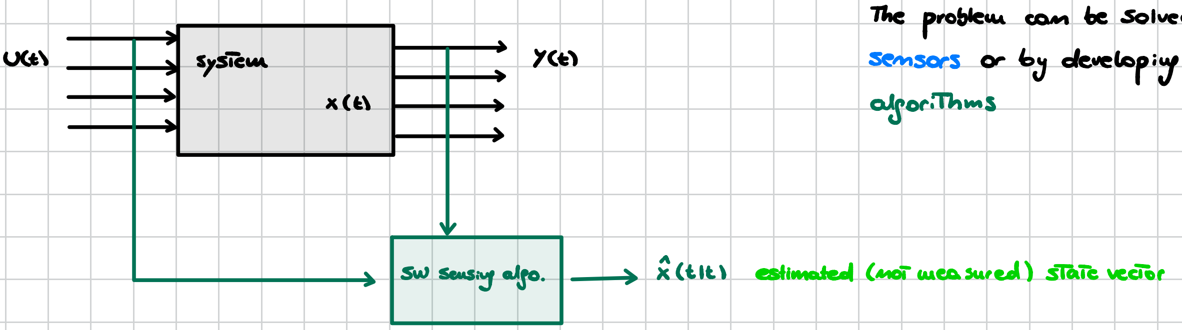


This is a MIMO system, with m inputs, p outputs and n states. Usually

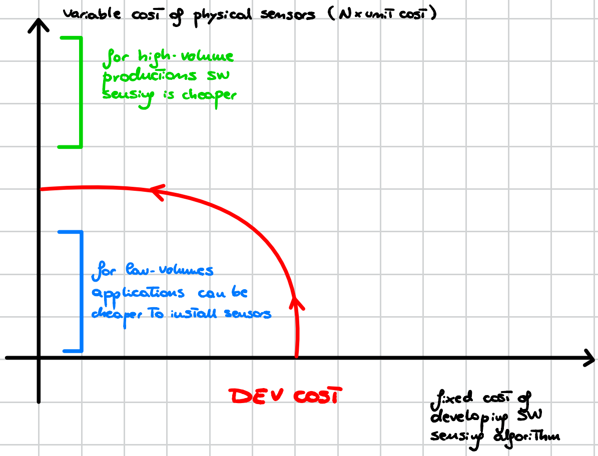
$$n \gg p$$

because of the cost of sensors

Usually only a few states are measured, but the availability of the full state measurement is very useful for the design of control algorithm and monitoring the system (eg. fault detection, predictive maintenance...)



The problem can be solved by adding physical sensors or by developing software sensing algorithms



Using physical or software sensors? In some cases is easy because no physical sensing is available; in many cases we have both option available. So we have to find a balance between development (fixed) cost and hardware (variable) cost

In some cases we might use both (physical and software) for redundancy (especially in safety-critical application).
We have 2 important issues of software sensing:

1) Is software sensing feasible? We have to check the observability
2) If it is feasible, we have to check the level of estimation error (is like the measurement error in physical sensors!)

Physical sensors are preferred but redundancy is made more and more with software sensors. In some cases, physical sensors could be cheaper, but software sensor is used when there are installation or space/weight problems.
Tendency today is to put more more software sensors than physical one.

I'll present Kalman Filter algorithm starting from a basic system

- No exogenous input ($Gu(t)=0$), so only time series
- Linear system
- Time invariant system

and solve the "basic problem" of the 1-step predictor. Then we'll make many extensions:

1) from 1 to k-step ahead predictors
2) Filter problem
3) system with exogenous input ($Gu(t) \neq 0$)
4) Time-varying systems
5) Non-linear systems (extended Kalman Filter)

Detailed description of the system (must be a given white-box model)

time series signal

$$S: \begin{cases} x(t+1) = Fx(t) + \cancel{Gu(t)} + v_1(t) \\ y(t) = Hx(t) + \cancel{Du(t)} + v_2(t) \end{cases}$$

noise models

Strictly proper system ($Du(t)=0$ $\forall t$)

with $Gu(t)=0$ and $Du(t)=0$ and two noise models $v_1(t)$ and $v_2(t)$; also

$x(t)=[x_1(t) \cdots x_n(t)]^T$ $\quad$ $y(t)=[y_1(t) \cdots y_p(t)]^T$ $\quad$ $u(t)=[u_1(t) \cdots u_m(t)]$

A frequent special case is the one of SISO system, where $p=m=1$. If we remove $Gu(t)=0$ we obtain a time series signal.

$v_1(t)$ is the White Noise vector $v_1 \sim WN(0, V_1^2)$ defined as $v_1(t)=\begin{bmatrix} v_{11}(t) \\ \vdots \\ v_{1n}(t)\end{bmatrix}$. $v_1(t)$ is called State Noise (or Model Noise) and is used to model internal disturbance or (small) model errors.

Property:

1) $E[v_1(t)] = \vec{0}$ $\quad$ expected value
2) $E[v_1(t)v_1(t)^T] = V_1^2$ $\quad$ covariance matrix of $v_1(t)$ is square, symmetric and semi-definite-positive
 so $V_1^2 \geq 0$ (semi-definite positive)
3) $E[v_1(t)v_1(t-\tau)^T] = 0$ $\quad \forall t, \forall \tau \neq 0$ $\quad$ (whiteness property)

$v_2(t)$ is the White Noise vector $v_2 \sim WN(0, V_2^2)$ defined as $v_2(t) = \begin{bmatrix} v_{21}(t) \\ \vdots \\ v_{2p}(t) \end{bmatrix}$. $v_2(t)$ is called output or sensor error and is used to model the noise of physical sensors.

Property:

1) $E[v_2(t)] = \vec{0}$ expected value
2) $E[v_2(t) v_2(t)^T] = V_2^2$ covariance matrix of $v_1(t)$ is square, symmetric and semi-definite-positive so $V_2^2 \geq 0$, but we make the additional request of being definite positive (so $V_2^2 > 0$). This will be usefull to guarantee the existence of kalman filter
3) $E[v_2(t) v_2(t-\tau)^T] = 0 \quad \forall t, \forall \tau \neq 0$ (whiteness property)

Consideration between V_1 and V_2: we assume that

$$E[v_1(t) v_2(t-\tau)^T] = \begin{cases} 0 & \text{if } \tau \neq 0 \\ V_{12} & \text{if } \tau = 0 \end{cases}$$ cross correlation matrix ($n \times p$ size)

where we assume that $v_1(t)$ and $v_2(t)$ can be correlated only at the same time. In practice usually we can assume $V_{12} = 0$

Since the system is a dynamical system, we must say something on the initial conditions of the state equation. Being $x(1)$ the initial state:

- $E[x(1)] = x_0$ ($n \times 1$ vector)
- $E[(x(1) - x_0)(x(1) - x_0)^T] = VAR[(x_1 - x_0)] = P_0$ ($n \times n$ matrix)

This is a problematic description of the initial state. If $P_0 = 0$ (special case) we know exactly the initial state.

We finally assume that:

- $v_1(t)$ is un-correlated with $x(1)$ ($v_1(t) \perp x(1)$)
- $v_2(t)$ is un-correlated with $x(1)$ ($v_2(t) \perp x(1)$)

Technical assumptions for the proof of KF's optimality

Presentation of the solution of Kalman Filter - basic system / basic problem (1-step prediction)

state equation + "correction"	:	$\hat{x}(t+1\|t) = F \hat{x}(t\|t-1) + K(t) \cdot e(t)$
output equation	:	$\hat{y}(t+1\|t) = H \hat{x}(t\|t-1)$
error equation	:	$e(t) = y(t) - \hat{y}(t\|t-1)$
gain equation	:	$K(t) = (F P(t) H^T + V_{12})(H P(t) H^T + V_2)^{-1}$
difference Riccati equation (DRE)	:	$P(t+1) = (F P(t) F^T + V_1) - (F P(t) H^T + V_{12})(H P(t) H^T + V_2)^{-1}(F P(t) H^T + V_{12})^T$

1° and 5° dynamical equations $\Rightarrow$ we need two initial condition for the two dynamical equations

$\hat{x}(1|0) = x_0$ $\qquad P(1) = P_0$

Structure of $K(t)$ and DRE: notice that they are a simple block structure, built using 3 blocks:

$$P(t+1) = (\underbrace{F P(t) F^T + V_1}_{\text{STATE}}) - (\underbrace{F P(t) H^T + V_{12}}_{\text{MIX}})(\underbrace{H P(t) H^T + V_2}_{\text{OUTPUT}})^{-1}(\underbrace{F P(t) H^T + V_{12}}_{\text{MIX}})^T$$

1) State : $F P(t) F^T + V_1$ $\Rightarrow$ F and V_1 are linked to the state eq
2) Output: $H P(t) H^T + V_2$ $\Rightarrow$ H and V_2 are linked to the output eq
3) Mix : $F P(t) H^T + V_{12}$ $\Rightarrow$ Mix between state and output equation

Also we can see that:

$K(t) = (\text{MIX}) \cdot (\text{OUTPUT})^{-1}$ $\qquad P(t+1) = (\text{STATE}) - (\text{MIX})(\text{OUTPUT})^{-1}(\text{MIX})^T$

DRE is a special type of non-linear, matrix, difference equation

$$\text{DRE}: \begin{cases} P(t+1) = f(P(t)) \;\cancel{+\text{ input}} \\ P(1) = P_0 \end{cases}$$ autonomous difference equation

Existence of DRE $\forall t$: notice that, to guarantee the existence of DRE $\forall t$, the matrix

$$HP(t)H^T + V_2$$

This is a strictly definite positive matrix by construction (≥ 0) - its invertibility is not guaranteed

we assumed $V_2 > 0$ and not only SDP

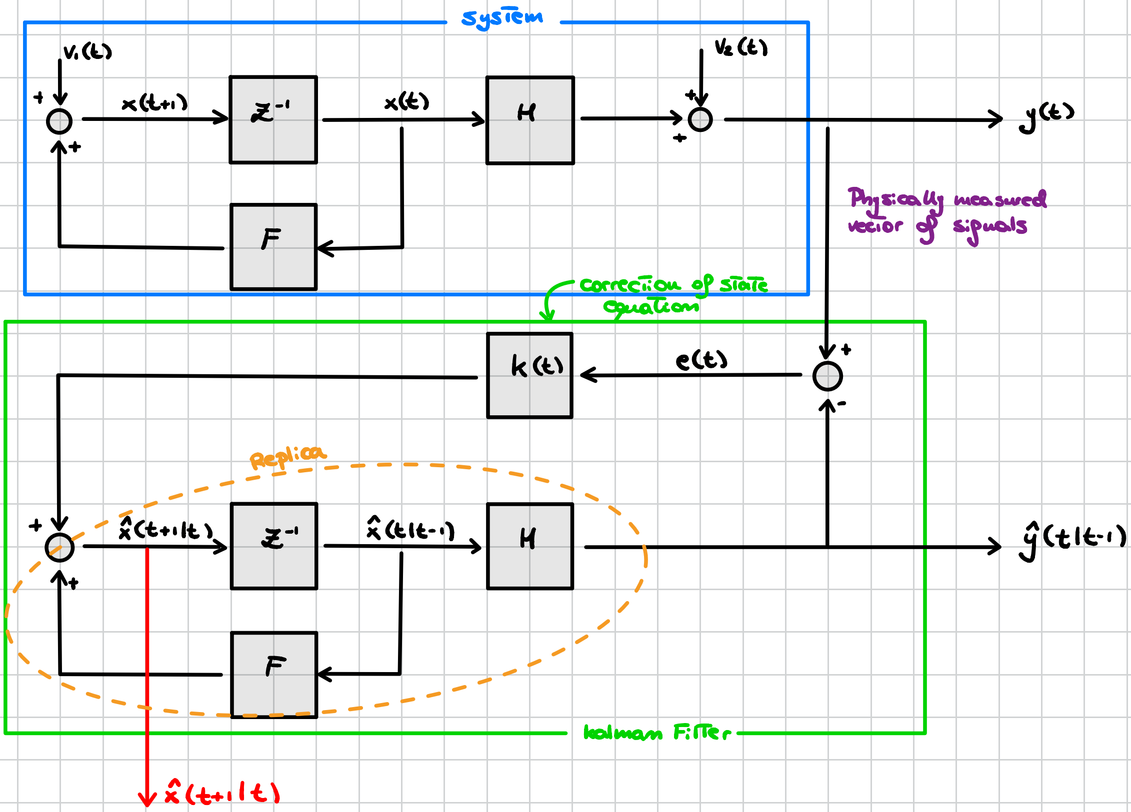
must be invertible; overall, this matrix is guaranteed to be (>0) definite positive and so always invertible.

Meaning of $P(t)$: $P(t)$ is a symmetric, strictly definite positive, $m \times m$ matrix; its meaning

$$P(t) = E\left[(x(t) - \hat{x}(t|t-1))(x(t) - \hat{x}(t|t-1))^T\right] = VAR\left[x(t) - \hat{x}(t|t-1)\right]$$

So it's the covariance matrix of the 1-step prediction error on the state

Block scheme of a system and kalman filter



The kF structure is very intuitive:

- it makes a replica (digital twin) of the system, except for the two unmeasurable noises $v_1(t)$ and $v_2(t)$
- it compares the real measured output with the estimated output and computes the output error
- corrects in feedback the state equation using a gain $k(t)$ applied to the output error: $e(t) = y(t) - \hat{y}(t|t-1)$

kF tries to follow the real system with a math model (simulator) with a feedback correction from output error. Feedback is used for estimation and not for control.

The idea is simple and effective, was known before kF (was called state observer in feedback). The fundamental contribution of Kalman was to provide a formula for the optimal choice of correction gain $k(t)$

- if $k(t)$ too small: under-exploit our information on $e(t)$
- if $k(t)$ is too high: over-amplifying the noise or even create instability

$k(t)$ is not just a scalar number (that can be tuned empirically); is a matrix of size $m \times p$ (eg. $m=10, p=3 \Rightarrow$ size of $k \Rightarrow 30$: impossible to tune empirically)

Weak point is the knowledge

V_1 — how can we design?

V_2 — covariance error of the sensor

~~V_{12}~~ — not a problem

The new trend is the optimization of V_1 and V_2 or directly $k(t)$ using ML (require a dataset)

Extension #1 : Multi-step prediction

Assume that $\hat{x}(t+1|t)$ is known. It can be proven that future predictions can be found as

$$\hat{x}(t+2|t) = F\,\hat{x}(t+1|t) \qquad \hat{x}(t+3|t) = F\,\hat{x}(t+2|t) = F^2\,\hat{x}(t+1|t)$$

So, in general we have

$$\boxed{\hat{x}(t+k|t) = F^{k-1}\,\hat{x}(t+1|t)} \quad \Rightarrow \quad \hat{y}(t+k|t) = H\,\hat{x}(t+k|t)$$

Extension #2 : Filter ($\hat{x}(t|t)$)

- if F is invertible we can go back and forth between predicted values

$$\hat{x}(t|t) \xleftarrow{\cdot F^{-1}} \hat{x}(t+1|t) \xrightarrow{\cdot F} \hat{x}(t+2|t)$$

So $\hat{x}(t|t) = F^{-1}\,\hat{x}(t+1|t)$ is known as filter.

- if F is not invertible there is a special version of kF formula (filter version). This formula is valid if $V_{12} = 0$ (in practice always). The filter version of the kF formula is

$$\hat{x}(t|t) = F\hat{x}(t-1|t-1) + k_0(t)\cdot e(t)$$

refer to prediction formulation

$$\hat{y}(t|t-1) = H\hat{x}(t|t-1)$$
$$e(t) = y(t) - \hat{y}(t|t-1)$$
$$k_0(t) = (P(t)H^T)(HP(t)H^T + V_2)^{-1}$$
$$P(t+1) = (FP(t)F^T + V_1) - (FP(t)H^T + V_{12})(HP(t)H^T + V_2)^{-1}(FP(t)H^T + V_{12})^T$$

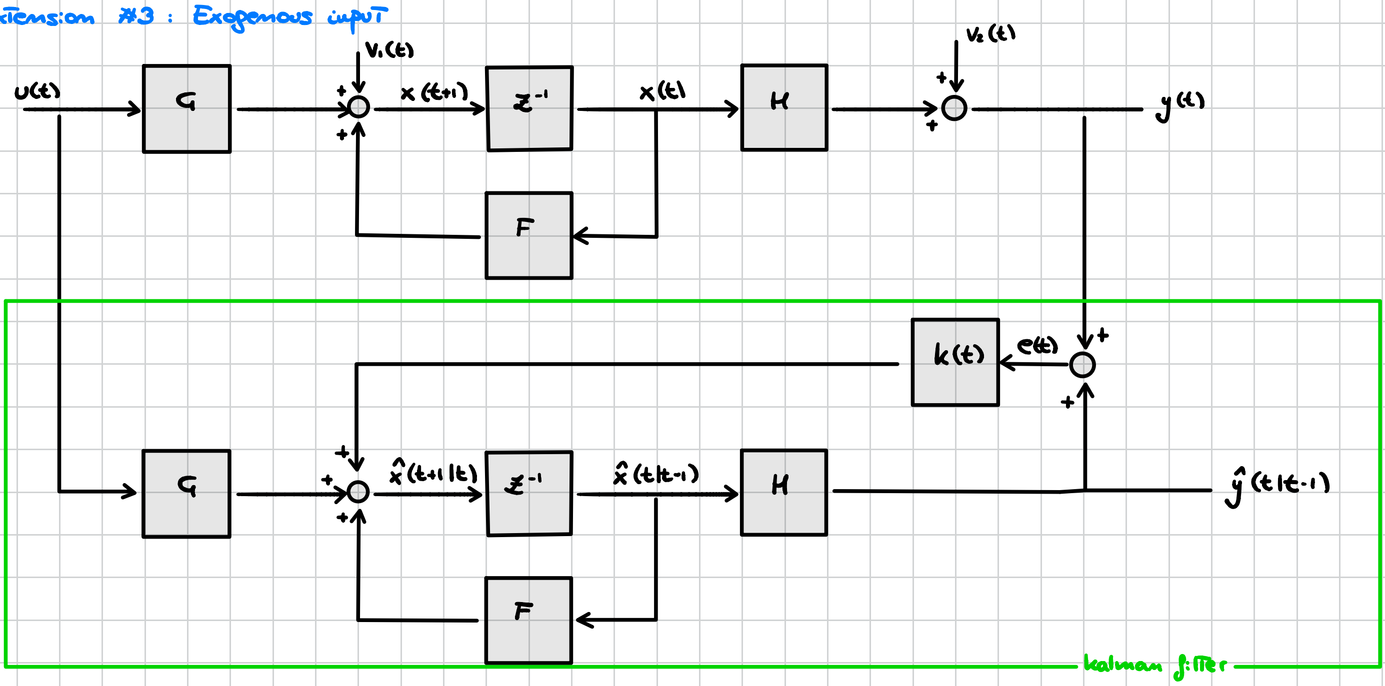
Invertibility of F is not requested in this version of the kF formula

Notice the difference between the form of the prediction and filter form

$$k(t) = (FP(t)H^T)(HP(t)H^T + V_2)^{-1} \qquad k_0(t) = (P(t)H^T)(HP(t)H^T + V_2)^{-1}$$

we miss F

Extension #3 : Exogenous input



Notice that both $k(t)$ and $P(t)$ are exactly the same of the formula of kF without $Gu(t)$: intuitive because $Gu(t)$ doesn't add any additional uncertainty to the system; $Gu(t)$ is perfectly known (deterministic signal).
Obviously the corresponding prediction error variance $P(t)$ does not change

Extension #4: Time-varying system

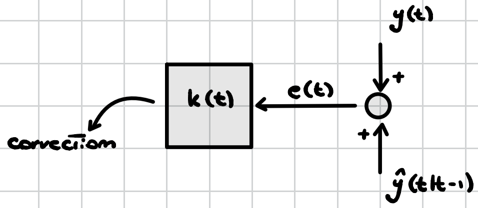
$$\begin{cases} x(t+1) = F(t)x(t) + G(t)u(t) + V_1 \\ y(t) = H(t)x(t) + V_2(t) \end{cases}$$

So we move from linear time invariant (LTI) system to a linear time varying (LTV) system

$F \to F(t)$ $\qquad G \to G(t)$ $\qquad H \to H(t)$

The kF equation are exactly the same. Aging, slow variation of system dynamics in time is a typical example of LTV.

Asymptotic (or steady state) solution of the kalman Filter



Notice that kalman Filter is an LTV system, even when the original system $\mathcal{f}$ is LTI (because of the gain $k(t)$)

The gain is time-varying, so the KF is an LTV system.

This fact creates 2 main problems:

1) checking (and guaranteeing) the asymptotic stability of the kF algorithm is very difficult

LTI: $\quad x(t+1) = Fx(t) + Gu(t) \quad \Rightarrow$ stability check is easy $\Rightarrow$ EIG(F)

LTV: $\quad x(t+1) = F(t)x(t) + G(t)u(t) \quad \Rightarrow$ even if all EIG(F(t)) are strictly inside the unit cycle $\forall t$, stability is not guaranteed! Checking the stability for an LTV system (especially if n is large) can be extremely complicated.

2) At every sampling time, we must update the computation of $k(t)$ and $P(t)$; $k(t)$ and DRE computation requires the inversion of a $p \times p$ matrix ("output block")

Because of problem 1 and 2, in real application almost always the kF is used in its "asymptotic version"

DRE is an autonomous dynamic system: $P(t+1) = f(P(t))$. If $P(t)$ does converge to $\bar{P}$ then also $k(t)$ does converge to $\bar{k}$, which can replace $k(t)$ (constant correction gain) that turn kF into an LTI system

Before addressing the question on the existence and the volume of $\bar{k}$, let's check the asymptotical stability property of kF, assuming that $\bar{k}$ does exist.

$$\begin{aligned} \hat{x}(t+1|t) &= F\hat{x}(t|t-1) + \bar{k}e(t) \\ &= F\hat{x}(t|t-1) + \bar{k}[y(t) - \hat{y}(t|t-1)] \\ &= F\hat{x}(t|t-1) + \bar{k}[y(t) - H\hat{x}(t|t-1)] = \underbrace{(F-\bar{k}H)}_{\text{stable if EIG}(F-\bar{k}H)\text{ in unit cycle}}\hat{x}(t|t-1) + \bar{k}y(t) \end{aligned}$$

where $F-\bar{k}H$ is the state matrix of the kF, so the kF is asymptotically stable if and only if all the EIG$(F-\bar{k}H)$ are strictly inside the unit cycle.

This means that

- The stability of the original system $\mathcal{f}$ depends on F
- The stability of kF depends on $F-\bar{k}H$

so, kF can be asymptotically stable even if the system $\mathcal{f}$ is not asymptotically stable. Since

$$k(t) = (FP(t)H^T + V_{12})(HP(t)H^T + V_2)^{-1}$$

we have a constant gain $\bar{k}$ if we have a constant matrix $\bar{P}$.

How can we find the equilibrium points of DRE? Remember:

- Continuous time system $\quad \dot{x} = f(x) \quad$ Equilibrium pairs is the solution of $\quad 0 = f(\bar{x})$
- Discrete time system $\quad x(t+1) = f(x(t)) \quad$ Equilibrium pairs is the solution of $\quad \bar{x} = f(\bar{x})$ (DRE in our case)

DRE is a discrete time, dynamic, autonomous system and its equilibrium points are the solution of:

$$\bar{P} = (F\bar{P}F^T + V_1) - (F\bar{P}H^T + V_{12})(H\bar{P}H^T + V_2)^{-1}(F\bar{P}H^T + V_{12})^T$$

This is a non-linear algebraic vector equation named Algebraic Riccati Equation (ARE).
$\bar{P}$ is the equilibrium points of the DRE. If a steady state solution of DRE does exist, it must be a solution of the ARE. So we have 3 key questions:

1) Existence : does DRE have SDP solutions?
2) Convergence: if $\bar{P}$ does exist, does DRE converge to $\bar{P}$?
3) Stability of KF: if we have existence and convergence, is the corrisponding KF asymptotically stable? Are all the EIG$(F-\bar{K}H)$ strictly inside the unit cycle?

Answering to these questions is very difficult, but we can use two famous theorems (Asymptotic KF Theorem). They provide only sufficient conditions for the questions.

1st Asymptotic Kalman Filter Theorem

Assumptions: $V_{12} = 0$ and f is asymptotically stable

Then:

- ARE has one and only one strictly definite positive solution $\bar{P} \geq 0$
- DRE converges to $\bar{P}$ $\forall P_0 \geq 0$
- The corresponding $\bar{K}$ is such that KF is asymptotically stable.

For the statement of the second theorem, we must recall some concepts of observability and controllability: The state $x(t)$ is fully observable from the output $y(t)$ if and only if

$$O = \begin{bmatrix} H \\ HF \\ \vdots \end{bmatrix}$$ is full rank.

For the second theorem we need a special type of controllability from the noise $v_1(t)$

$$x(t+1) = Fx(t) + Gu(t) + v_1(t) \qquad v_1 \sim WN(0, V_1)$$

$v_1(t)$ is an input for the state equation. We can define the controllability of the state from input $v_1(t)$.
We can manipulate the system state equation in this way

$$x(t+1) = Fx(t) + Gu(t) + \Gamma\omega(t) \qquad \omega(t) \sim WN(0, I)$$

where Γ is the factorization of $V_1 = \Gamma\Gamma^T$ ↑ identity matrix

Example when $n = 1$

$$\left.\begin{aligned} x(t+1) &= \tfrac{1}{2}x(t) + 1\,v_1(t) \qquad & v_1 \sim WN(0,4) \\ x(t+1) &= \tfrac{1}{2}x(t) + 2\,\omega(t) \qquad & \omega(t) \sim WN(0,1) \end{aligned}\right\} \quad V_1 = 4 \qquad \Gamma = 2 \Rightarrow V_1 = \Gamma^2$$

We can say that the system is fully controllable from noise $v_1(t)$ if and only if

$$R = [\Gamma \quad F\Gamma \quad F^2\Gamma \quad \cdots \quad F^{n-1}\Gamma]$$

is full rank. The meaning of this special controllability is the fact that noise $v_1(t)$ must affect all the states (we should not have "clean" or noise-free state equations)

2nd asymptotical KF Theorem

Assumptions:

- $V_{12} = 0$
- (F, H) is fully observable
- (F, Γ) is fully controllable ($V_1 = \Gamma\Gamma^T$)

Then

- ARE has 1 solution $\bar{P} > 0$ ($\bar{P}$ is guaranteed to be definite positive)
- DRE converges to $\bar{P}$ $\forall P_0 \geq 0$
- The corresponding $\bar{K}$ is such that the KF is asymptotically stable

These Theorems are very useful in practice since we can skip the very complicated direct convergence analysis of DRE

The KF formulas can be applied under the assumption that both $v_1(t)$ and $v_2(t)$ are white noises. This assumption can be quite restrictive in many practical cases (eg. The sensor/output noise $v_2(t)$ is not white but a colored noise).

How can we deal with This problem? The workaround is a "Trick" and is called "Model / System extension"

Example

$$\mathcal{S}: \begin{cases} x(t+1) = a x(t) + \eta(t) \\ y(t) = b x(t) + v_2(t) \end{cases} \qquad v_2(t) \sim WN(0,1) \quad \text{but } \eta(t) \text{ is not white}$$

$\eta(t)$ can be modeled as an AR(1) colored noise: $\eta(t) = \frac{1}{1-cz^{-1}} e(t)$ $\quad e(t) \sim WN(0,1)$ and $v_2(t) \perp e(t)$. We cannot apply KF Theory!

We expand the noise model

$$\eta(t+1) = c\eta(t) + e(t+1)$$

and we define $v(t) = e(t+1)$ so we obtain

$$\eta(t+1) = c\eta(t) + v(t) \qquad v(t) \sim WN(0,1) \qquad v \perp v_2 \text{ (uncorrelated)}$$

State extension:

$$x(t) = x_1(t) \qquad \eta(t) = x_2(t)$$

We are including the noise dynamics into the system dynamics. The new system is of order $n=2$, $x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$

$$\begin{cases} x_1(t+1) = a x_1(t) + x_2(t) \\ x_2(t+1) = c x_2(t) + v(t) \\ y(t) = b x_1(t) + v_2(t) \end{cases} \Rightarrow \quad F = \begin{bmatrix} a & 1 \\ 0 & c \end{bmatrix} \quad H = \begin{bmatrix} b & 0 \end{bmatrix}$$

We can apply The KF formulas To obtain

$$v_1(t) = \begin{bmatrix} 0 \\ v(t) \end{bmatrix} = v_1(t) \sim WN(0, V_1) \qquad \Rightarrow \quad V_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$v_2(t) \sim WN(0,1) \qquad \Rightarrow \quad V_2 = 1$$

$$\Rightarrow \quad V_{12} = 0$$

Extension #5: Non-linear systems (Extended Kalman Filter - EKF)

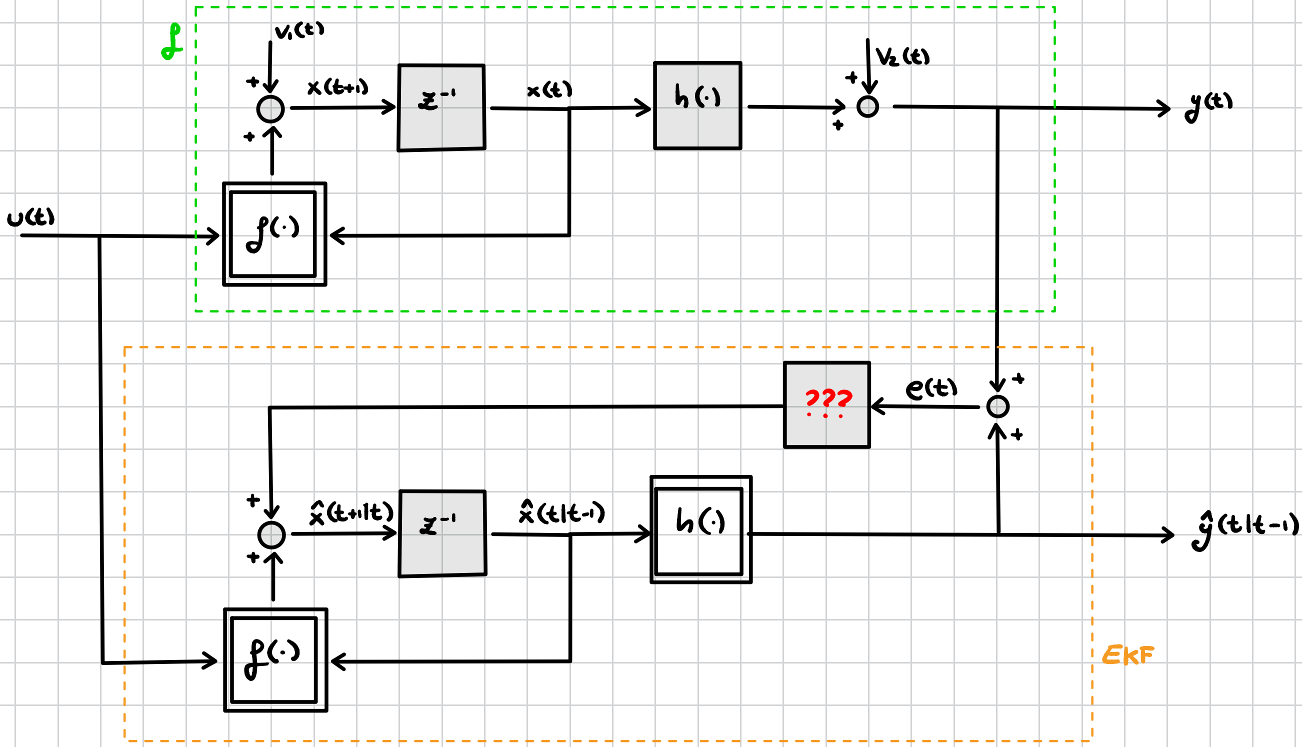
Consider a State Space model system with non-linear dynamics

$$\begin{cases} x(t+1) = f(x_1(t), u(t)) + v_1(t) \\ y(t) = h(x(t)) + v_2(t) \end{cases}$$

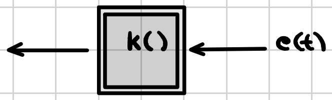
where $f(\cdot)$ and $h(\cdot)$ are non linear function of class C^1 or more. For example, a system of its kind could be:

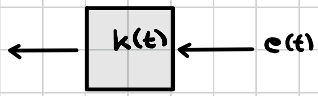
$$\begin{cases} x(t+1) = \frac{1}{2} x(t)^5 + u(t)^3 + v_1(t) \\ y(t) = e^{x(t)} + v_2(t) \end{cases}$$

How can we design KF? Follow The main idea: model replica and feedback correction



What is the EKF gain? To design the feedback correction gain we can have 2 different options:





1) Most intuitive and natural: the gain itself is a non-linear function $k(e(t))$, where $k(\cdot)$ is a non-linear function.

2) The gain is again a linear time varying block of the KF: $k(t)e(t)$

The second option is the adopted by EKF; less intuitive but very effective since we can reuse most of KF theory.

$k(t)$ of EKF can be computed using the traditional KF formula

$$k(t) = \left(F(t)\,P(t)\,H(t)^T + V_{12} \right)\left(H(t)\,P(t)\,H(t)^T + V_2 \right)^{-1}$$

$$P(t+1) = \left(F(t)P(t)F(t)^T + V_1 \right) - \left(F(t)P(t)H(t)^T + V_{12} \right)\left(H(t)P(t)H(t)^T + V_2 \right)^{-1}\left(F(t)P(t)H(t)^T + V_{12} \right)^T$$

we need to compute $F(t)$ and $H(t)$ at every sampling time:

$$F(t) = \frac{\partial}{\partial x(t)} f(x(t), u(t)) \Big|_{x(t) = \hat{x}(t|t-1)} \qquad H(t) = \frac{\partial}{\partial x(t)} h(x(t)) \Big|_{x(t) = \hat{x}(t|t-1)}$$

$F(t)$ and $H(t)$ are time-varying matrices obtained by local linearization of the non-linear functions around the predicted state recomputed at each time step.

The idea of EKF is to transform a non-linear time-invariant system into an LTV system; we can use KF on LTV systems.

Procedure of EKF at time t:

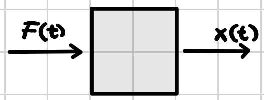
1) Take the last available prediction of the state $\hat{x}(t|t-1)$
2) Using $\hat{x}(t|t-1)$, compute the local linearization matrices $F(t)$ and $H(t)$
3) Compute $k(t)$ and update DRE up to $P(t+1)$
4) Compute $\hat{x}(t+1|t)$ ready for the next iteration

EKF is very powerful since it can be applied to non linear systems. It does not have a steady state (asymptotic) solution: so EKF is a non linear and time-varying system. As a consequence:

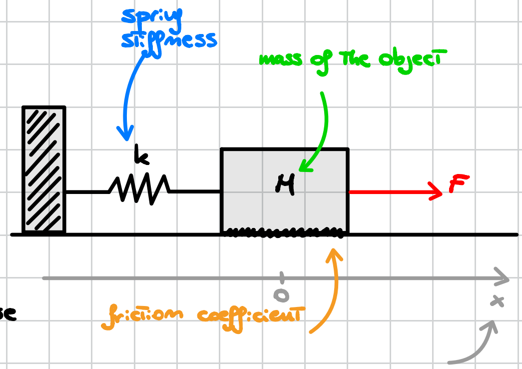
1) Impossible (or very difficult) to guarantee the asymptotic stability (only extensive empirical simulation can be done)
2) Computation load is high (at each time we must compute $F(t)$, $H(t)$, $P(t)$, $K(t)$)

EKF today is largely used but with limitations for
- safety - critical applications : sleep estimation on ABS
- mission critical application : rover on Mars.

Example: end-to-end application on a toy real-world problem

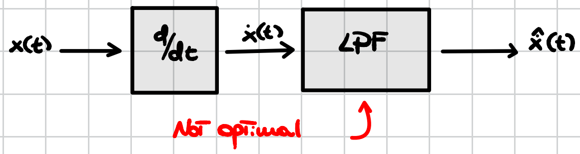


Input: control force F
Output: position x(t)



Estimate the unmeasurable speed $\dot{x}(t)$

Easy way: numerical derivative of position signal $x(t)$. Not optimal because creates a phase shift if we try to clean the s/p noise generated by numerical differentiation



Let's use the optimal solution with KF

1) Build the system model (Typically is white-box physical model)

II° Newton Law: $\ddot{x}(t)M = \underbrace{-k\,x(t)}_{\text{elastic force of the spring (conservative)}} \underbrace{-\,c\,\dot{x}(t)}_{\text{friction force (dissipative)}} + F(t)$

It is a Linear, 2° order dynamical system (2° ordinary differential equation) continuous time

We can transform it into standard state space form: we need 2 first order equations, 2 state variables (in mechanical systems: position $x(t)$ and speed $\dot{x}(t)$)

$$x(t) = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

In continuous time domain, the state space representation become:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -\frac{k}{M}x_1(t) - \frac{c}{M}x_2(t) + \frac{1}{M}F(t) \\ y(t) = x_1(t) \end{cases} \Rightarrow \quad A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{c}{M} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1/M \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad D = 0$$

2) Move from continuous time to discrete time

One of the simplest way of discretizing a continuous time system is to use the Eulero-forward method:

$$\dot{x}(t) \approx \frac{x(t+1) - x(t)}{\Delta}$$

where Δ is the sampling time (eg. $\Delta = 10$ ms). Applying in our system:

$$\begin{cases} \frac{x_1(t+1) - x_1(t)}{\Delta} = x_2(t) \\ \frac{x_2(t+1) - x_2(t)}{\Delta} = -k/M\, x_1(t) - C/M\, x_2(t) + 1/M\, F(t) \\ y(t) = x_1(t) \end{cases}$$

So in standard SS form in discrete time:

$$\begin{cases} x_1(t+1) = x_1(t) + \Delta x_2(t) \\ x_2(t+1) = -\frac{k\Delta}{M} x_1(t) - \frac{C\Delta}{M} x_2(t) + x_2(t) + \frac{\Delta}{M} F(t) \\ y(t) = x_1(t) \end{cases} \Rightarrow F = \begin{bmatrix} 1 & \Delta \\ -\frac{k\Delta}{M} & \frac{1-C\Delta}{M} \end{bmatrix} \quad G = \begin{bmatrix} 0 \\ \Delta/M \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

3) Add the noises

$$\begin{cases} x_1(t+1) = x_1(t) + \Delta x_2(t) + v_1(t) \\ x_2(t+1) = -\frac{k\Delta}{M} x_1(t) - \frac{C\Delta}{M} x_2(t) + x_2(t) + \frac{\Delta}{M} F(t) + v_{12}(t) \\ y(t) = x_1(t) + v_2(t) \end{cases}$$

Considering the standard state space for a discrete time system we can obtain

$$v_1(t) = \begin{bmatrix} v_{11}(t) \\ v_{12}(t) \end{bmatrix} \sim WN(0, V_1) \qquad v_2(t) \sim WN(0, V_2)$$

The problem is how to define V_1, V_2 (and V_{12}). Empirically solved:

- $V_{12} = 0$ No reason of correlation
- V_2 can be designed using the sensor data sheet or analyzing the signal recorded for the sensor
- V_1 is much more complicated: usually we simplify and we assume diagonal matrix $V_1 = \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix}$ with λ_1^2, λ_2^2 empirically tuned from data

New stream of research of KF is to estimate from data (learning) metaparameters: in principle, KF is not an ML technique; in practice today a lot of learning from data is used to tune at best the metaparameter

4) Apply KF (linear and time invariant): we can try to use Theorem 1 / Theorem 2