Un Esempio di Applicazione del Filtro di Kalman alle reti WiFi

ilenia.tinnirello@tti.unipa.it

Sommario

- Che cosa e' il filtro di Kalman?

  [Riepilogo intuitivo]

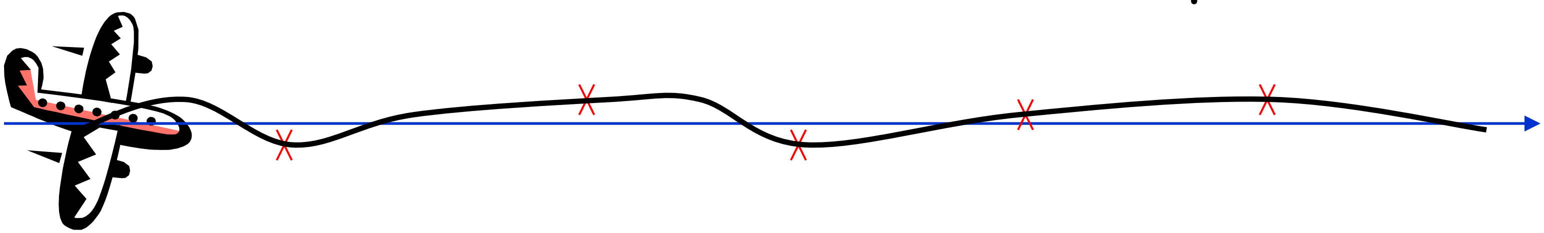
- Come funzione l'accesso al mezzo nelle reti WiFi?

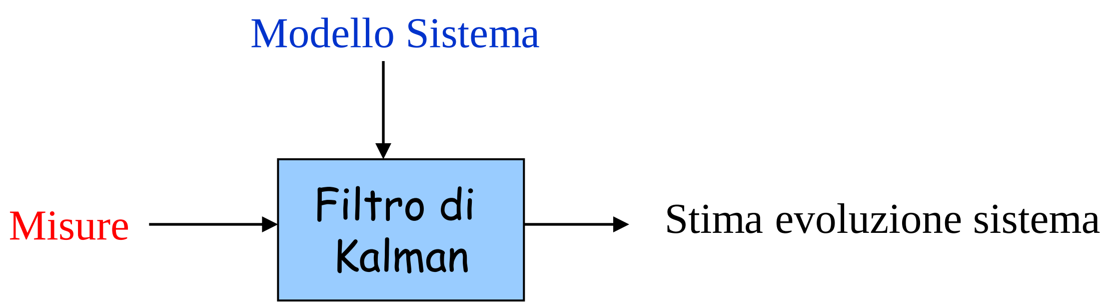
- Come si puo' usare il filtro di Kalman per migliorare le prestazione delle reti WiFi?

Introduzione
al filtro di Kalman

Che cosa e'il filtro di Kalman?

Teoricamente: e' uno strumento per ***stimare*** lo stato di un sistema dinamico lineare perturbato da rumore, sulla base di misure (o osservazioni) linearmente dipendenti dallo stato e corrotte da rumore. Lo stimatore e' **ottimo** rispetto a qualunque funzione quadratica dell'errore di stima: si basa su tutte le informazioni disponibili.



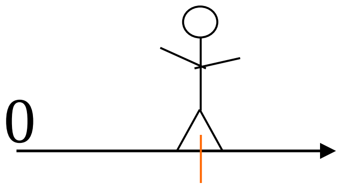


Che cosa e'il filtro di Kalman?

Praticamente: una delle piu' grandi invenzioni della teoria della stima (e forse del secolo!). Migliaia di applicazioni soprattutto nel campo meccanico e aerospaziale. Non un filtro in senso tradizionale (caratterizzato da poli e zeri), ma piuttosto un "programma". E' ricorsivo: nuove misure possono essere processate via via che arrivano.

Ma e' stimatore o un filtro?
Solitamente il filtro e' un dispositivo che elimina o seleziona alcune componenti frequenziali di segnale ->
Si chiama filtro nel senso che elimina il rumore delle misure per costruire le stime.

Stima della posizione di un oggetto fermo



Un osservatore si trova in posizione *y;*

Misurando la distanza da un punto di riferimento trova z1; ripetendo la misura ottiene z2, z3, z4..

Come si puo' stimare *y* a partire dalle misure Z ?

Prima misura: $\hat{y}$ = z1;
Seconda misura: $\hat{y}$ = (z1 + z2)/2; ...

Ogni volta che si dispone di una nuova misura, occorre riutilizzare tutte le precedenti per produrre la nuova stima. Oppure:

$$\hat{y}_{new} = \hat{y}_{old} + \alpha(z - \hat{y}_{old})$$

Dove α puo' essere costante o dipendere dal numero di misure gia' pervenute. Piu' in dettaglio..

Media di misure in modo ricorsivo

$\hat{y}(1) = z1$; $\hat{y}(2) = (z1 + z2)/2$; $\hat{y}(3) = (z1 + z2 + z3)/3$; …

In alternativa:

$\hat{y}(1) = z1;$

$\hat{y}(2) = (\hat{y}(1) + z2)/2 = \hat{y}(1) + 1/2 * (z2 - \hat{y}(1));$

$\hat{y}(3) = (\hat{y}(2)*2 + z3)/3 = \hat{y}(2) + 1/3 * (z3 - \hat{y}(2));$

…

$\hat{y}(n) = (\hat{y}(n-1)*(n-1) + zn)/n = \hat{y}(n-1) + 1/n * (zn - \hat{y}(n-1));$

A partire dalla vecchia stima, ad ogni passo aggiungo un termine pari alla differenza tra nuova misura e vecchia stima. Questo termine (chiamato innovazione) rappresenta l'informazione aggiuntiva rappresentata dalla nuova misura e pesa sempre meno nell'aggiornamento della stima.

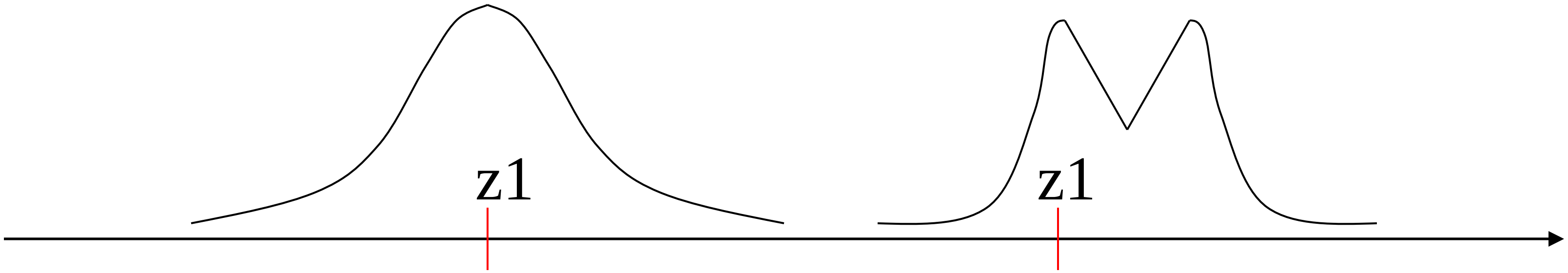
Giustificazioni (1)

Le misure Z saranno variabili aleatorie ottenute dalla somma di *y* e di un rumore v: Z = *y* + v;

Assumiamo *v* gaussiana a media nulla e varianza σ_v^2.

Z sara' pure gaussiana a media *y* e varianza σ_v^2. Ma non sappiamo quanto vale *y*! Posto che *y* sia noto:

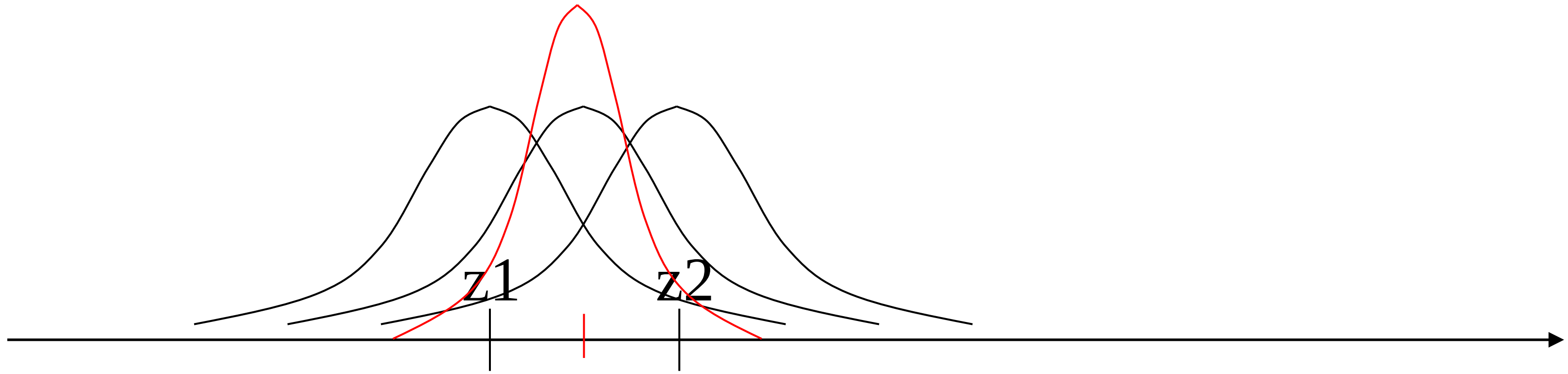
$$f(z/y) = \frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{(z-y)^2}{2\sigma^2})$$



Se disponiamo di una sola misura, assumiamo di aver beccato il valore piu' probabile; Poiche' per distribuzione guassiana coincide con il valore medio, assumiamo la stima pari a z1. (non e' valido in generale!)

Giustificazioni (2)

E se disponiamo di due misure? Dove sta la vera curva di dispersione delle misure? Quando si ottiene il massimo della pr che su due misure becchiamo giusto i valori z1 e z2?

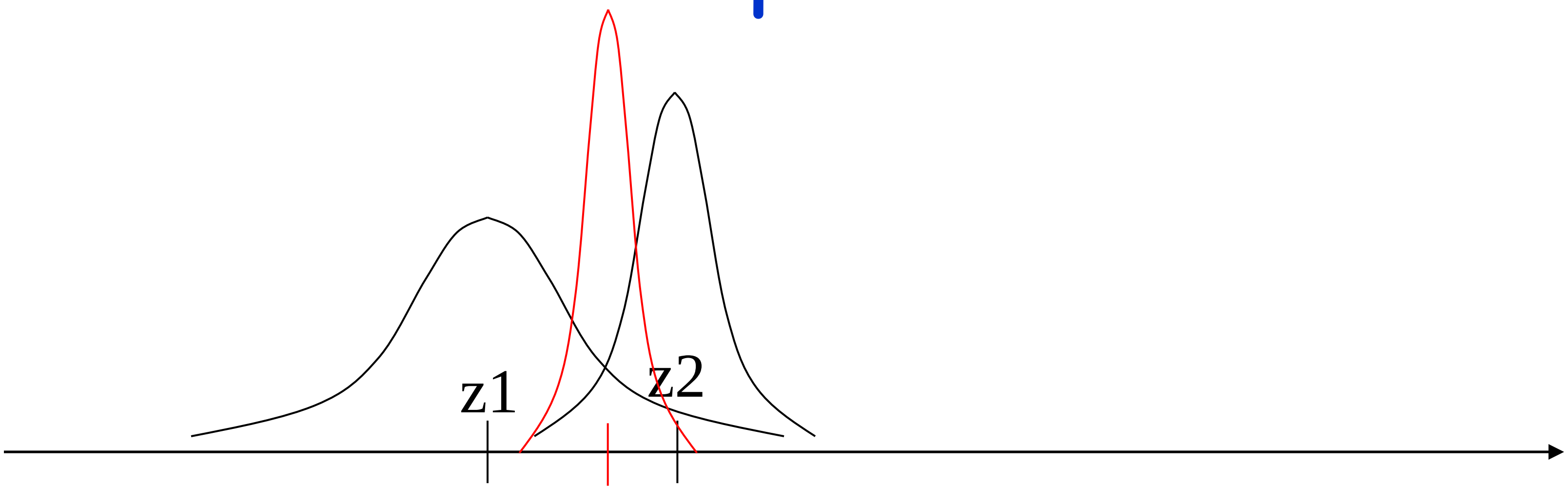


Otteniamo il massimo quando il centro della gaussiana (cioe' *y*) e' collocato a meta' tra z1 e z2!

Piu' formalmente, dobbiamo annullare la derivata di f(z1,z2/y) rispetto a y.

Con una misura, l'errore dipende da σ_v^2; combinando N misure di varianza σ_v^2, la varianza della stima si riduce a σ_v^2/N -> all'aumentare delle misure disponibili la stima migliora.

Combinazione di misure a diversa precisione

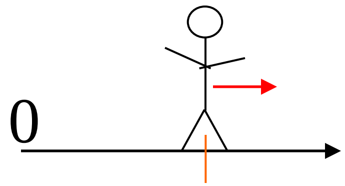


Conviene fare la media? O dobbiamo pesare di piu' la misura piu' accurata? E quanto di piu'?

Si ricava che secondo il criterio di massima verosimiglianza la migliore stima e':

$$\hat{y} = \frac{\sigma_2^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} z1 + \frac{\sigma_1^{\ 2}}{\sigma_1^{\ 2} + \sigma_2^{\ 2}} z2$$

Stima della posizione di un oggetto in moto



Un osservatore si trova in posizione *y0* e si muove di moto uniforme con velocita' v (che supponiamo nota);

Misurando la distanza da un punto di riferimento trova z1; ripetendo la misura ottiene z2, z3, z4..

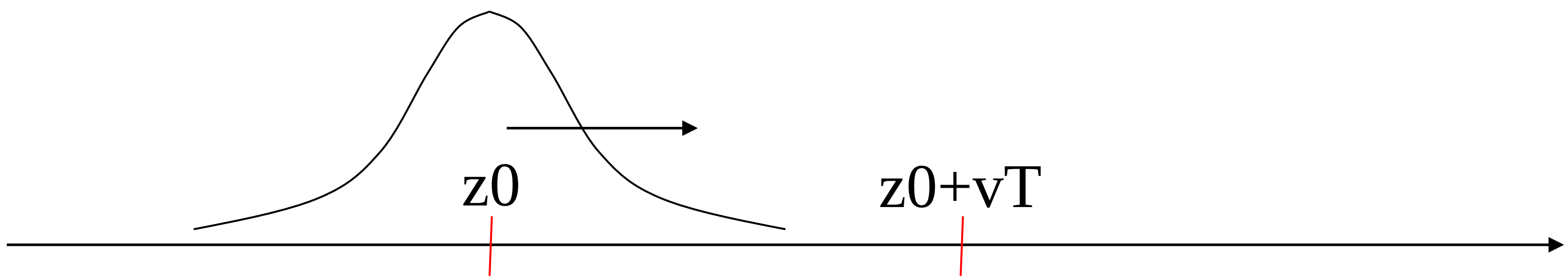
Supponiamo che una nuova misura sia disponibile ogni T secondi. Come si puo' stimare *y(t)* a partire dalle misure Z ? Posso ancora usare le vecchie misure per fare la stima, visto che la posizione cambia?

Stima inziale: $\hat{y}(0) = z0;$

Dopo T sec. potro' assumere: $\hat{y}(T^-) = z0 + vT;$

Appena arriva la nuova misura: $\hat{y}(T^+) = (z0 + vT + z1)/2\ ;$

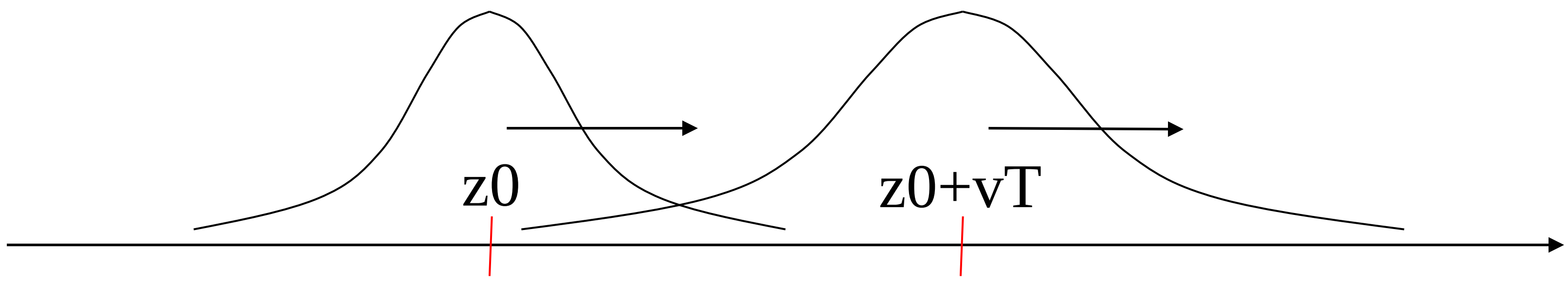
Stime per sistemi dinamici (1)



Legge di evoluzione del moto: $y(n) = y0 + v*nT = y(n-1) + vT$;
Sulla base di questa legge posso aggiornare le stime nel tempo, anche se non dovessero arrivare altre misure.
$\hat{y}(0) = z0$; $\hat{y}(1) = z0+vT$; $\hat{y}(2) = z0+2vT$; …

N.B. Se v e' nota e costante, la stima non degrada con il tempo (cioe' l' errore sulla posizione vera non peggiora):
$e(0) = y(0) - \hat{y}(0) = y(0) - z0$; $e(1) = y(0) + vT - z0 - vT = e(0)$; ..

Stime per sistemi dinamici (2)



Legge di evoluzione del moto: y(n) = y0 + v(n)*nT; v(n) = v + u(n);

-> y(n) = y0 + v(n)*nT + w

E se la velocita' non e' costante, ma ci sono piccole accelerazioni/decelerazioni casuali attorno a v?

Disponendo della sola misura z0, intuitivamente mi aspetto che nel tempo l'incertezza sulla stima peggiori, poiche' io non ho modo di rilevare le fluttuazioni di v!

(Se anche z0 fosse y(0) non posso aspettarmi di continuare a tracciare con esattezza il moto)

Come funziona il filtro di Kalman?

Tiene in considerazione tutto quello che abbiamo detto !

Si basa su un modello di stato cosi' fatto:

$$y(n) = A\, y(n-1) + w;$$

$$z(n) = H\, y(n) + v;$$

1) Dinamica della grandezza da stimare (legge di aggiornamento dello stato)
2) Relazione tra osservazioni (misure) e stato

Algoritmo

Parametri in ingresso: statistiche rumori sullo stato e sulle misure

Supponiamo che al tempo n si dispone della stima $\hat{y}(n)$ con varianza $P(n)$; Al tempo n+1:

1) **Stima a priori**: prevediamo come si e' evoluto lo stato sulla base della legge del modello

   $\hat{y}(n+1)^- = A\,\hat{y}(n)^+$;

2) **Stima della misura**: prevediamo quale sara' il nuovo valore della misura sulla base del modello

   $\check{z}(n+1) = H\,\hat{y}(n+1)^-$;

3) **Stima a posteriori**: aggiorniamo la stima con la nuova misura pervenuta (il coefficiente K e' tempo variante e si chiama guadagno di Kalman)

   $\hat{y}(n+1)^+ = \hat{y}(n+1)^- + K\,(\,z(n+1) - \check{z}(n+1)\,)$;

4) **Aggiorniamo la varianza della stima** tenendo in conto che ogni nuova misura migliora la stima e quindi ne riduce la varianza

MATLAB DEMO

[Estimatation of a constant parameter x]

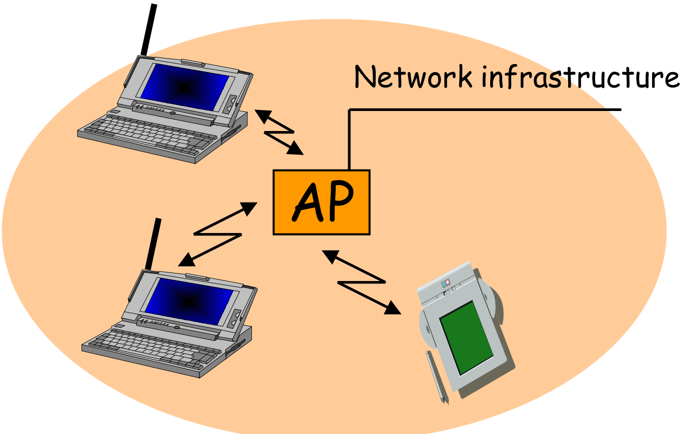
```
function [xhat, P, K] = es_kalman(a, h, sw, sv, x0, xhat0, P0)
…
for i=2:1:N
    x(i)          = a*x(i-1)  + sqrt(sw)*randn;
    z(i)          = h*x(i) + sqrt(sv)*randn;

    x_a_priori   = a*xhat(i-1);
    z_attesa     = h*x_a_priori;
    P_a_priori   = a^2*P(i-1) + sw;

    K(i)          = h*P_a_priori/(h^2*P_a_priori + sv);
    inn(i)        = z(i) - z_attesa;
    xhat(i)       = x_a_priori + K(i)*inn(i);
    P(i)          = (1 - h*K(i))*P_a_priori;
end
```

Meccanismo di accesso al mezzo in reti WIFI

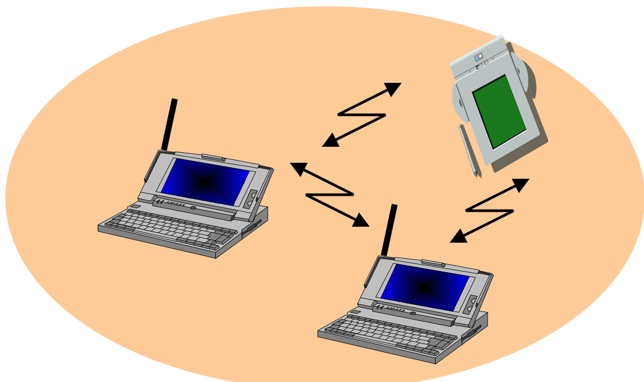
Basic Service Set (BSS)

group of stations that can communicate with each other

- Infrastructure BSS
  - or, simply, BSS
  - Stations connected through AP

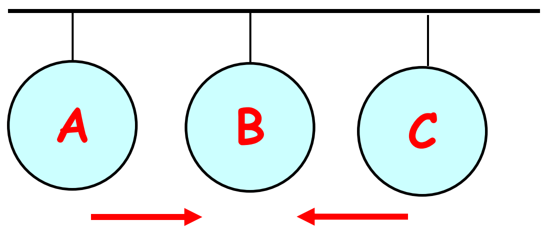


- Independent BSS
  - or IBSS
  - Stations connected in ad-hoc mode



Wireless Ethernet

- 802.3 (Ethernet)
  - CSMA/CD
    - Carrier Sense Multiple Access
    - Collision Detect
- 802.11(wireless LAN)
  - CSMA/CA
  - (Distributed Coordination Function)
    - Carrier Sense Multiple Access
    - Collision Avoidance



➔ **Both A and C sense the channel idle at the same time → they send at the same time.**

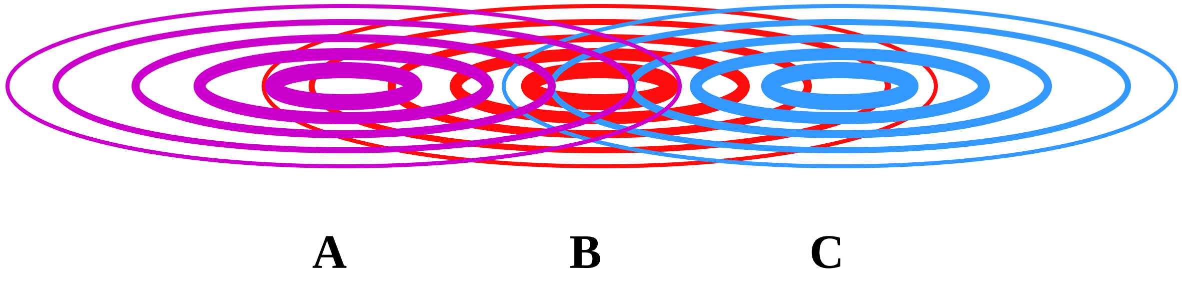
➔ **Collision can be detected at sender in Ethernet.**

➔ **Why this is not possible in 802.11?**

1. *Either TX or RX (no simultaneous RX/TX)*
2. *Large amount of power difference in Tx and Rx (even if simultaneous tx-rx, no possibility in rx while tx-ing)*
3. *Wireless medium = additional problems vs broadcast cable!!*

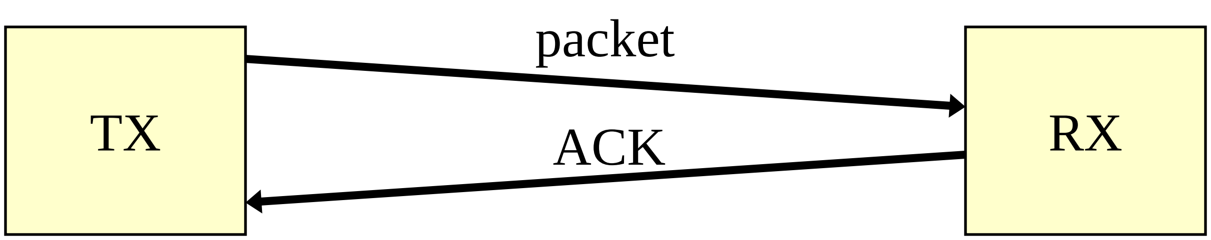
Hidden Terminal Problem

- Large difference in signal strength; physical channel impairments (shadowing)
  - **It may result that two stations in the same area cannot communicate**
- Hidden terminals
  - **A and C cannot hear each other**
  - **A transmits to B**
  - **C wants to send to B; C cannot receive A;C senses "idle" medium** (Carrier Sense fails)
  - **Collision occurs at B.**
  - **A cannot detect the collision** (Collision Detection fails).
  - **A is "hidden" to C.**



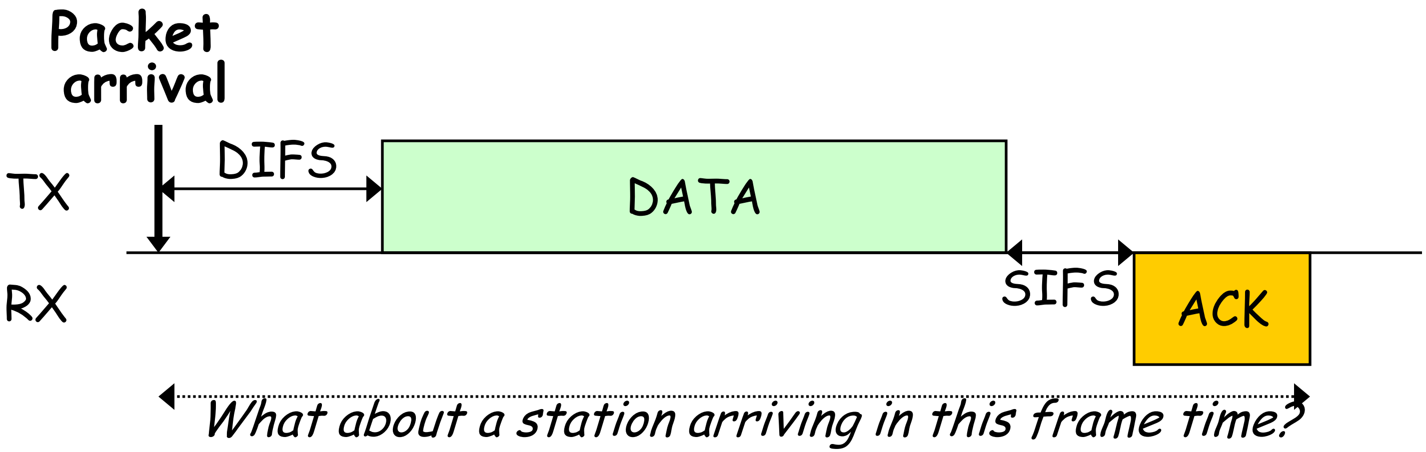
802.11 MAC approach

- Still based on Carrier Sense:
  - Listen before talking
- But collisions can only be inferred afterwards, at the receiver
  - Receivers see corrupted data through a CRC error
  - Transmitters fail to get a response
- Two-way handshaking mechanism to infer collisions
  - DATA-ACK packets



Channel Access details

- A station can transmit only if it senses the channel IDLE for a DIFS time
  - DIFS = Distributed Inter Frame Space

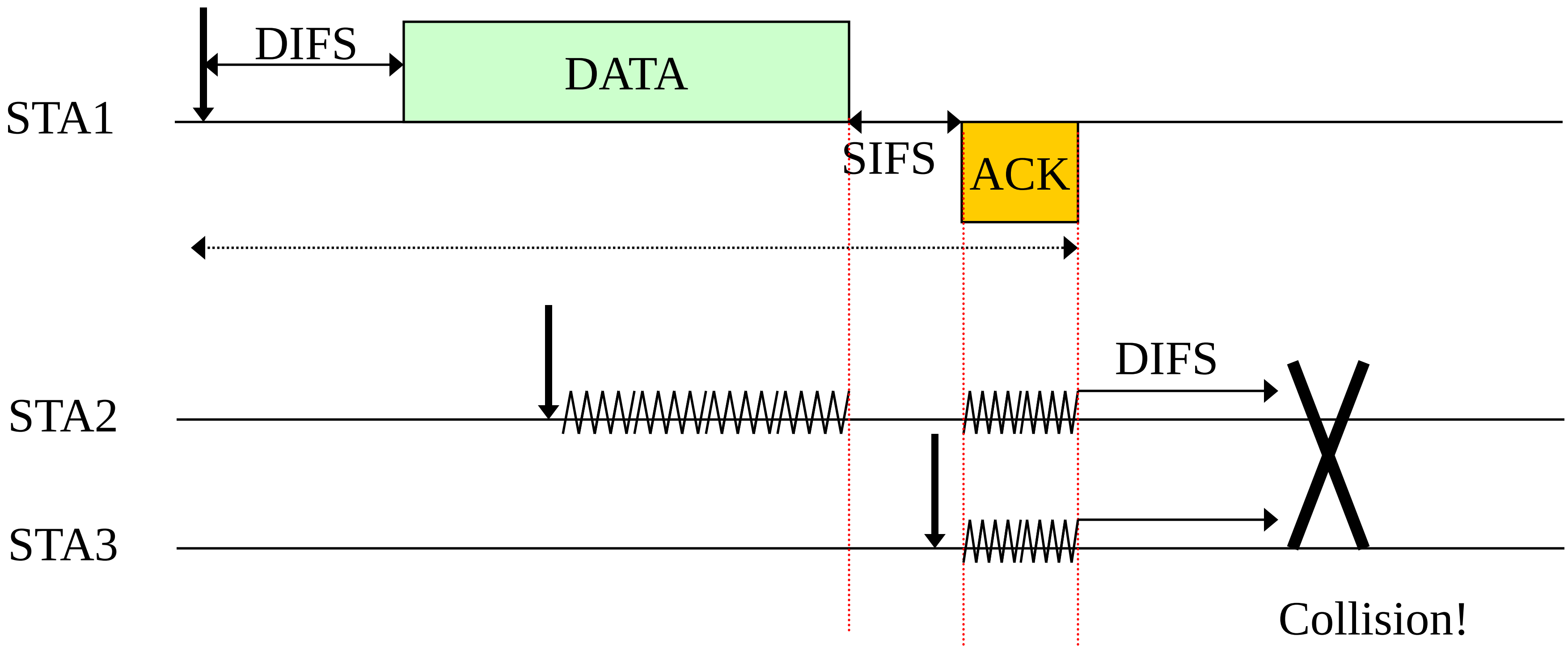


➔ **Key idea: DATA and ACK separated by a different Inter Frame Space**

  ⇨ SIFS = Short Inter Frame Space

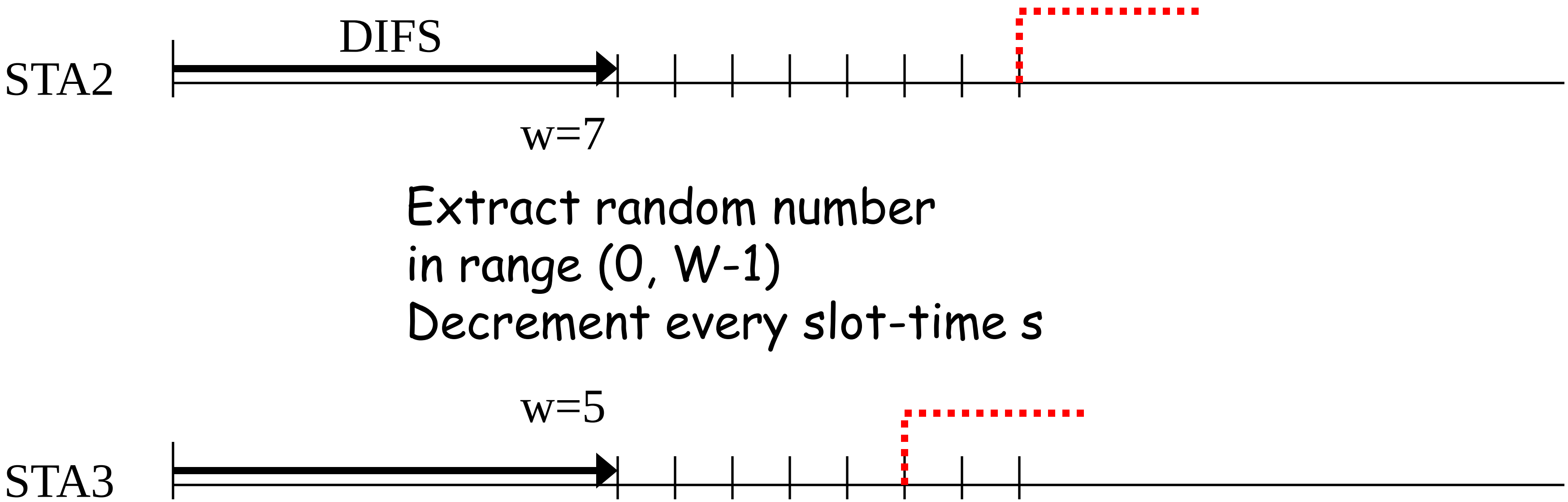
⇨ **Second station cannot hear a whole DIFS, as SIFS<DIFS**

Why backoff?



RULE: *when the channel is initially sensed BUSY, station defers transmission;*
But when it is sensed IDLE for a DIFS, defer transmission of a further random time (BACKOFF TIME)

Slotted Backoff

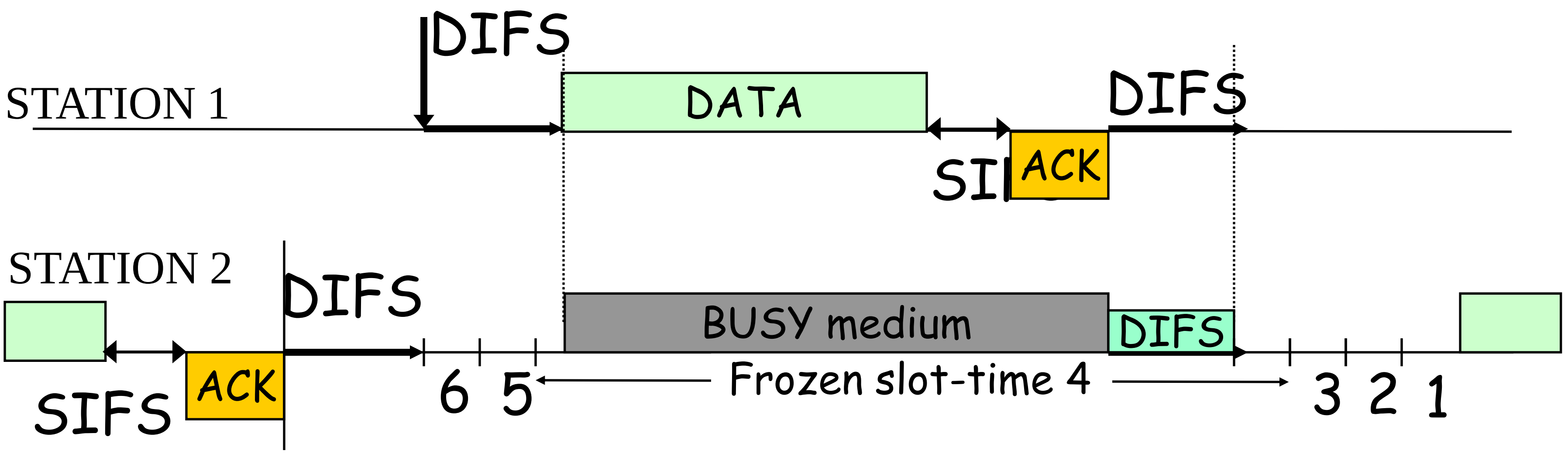


Note: slot times are not physically delimited on the channel!
Rather, they are logically identified by every STA

Slot-time values: 20ms for DSSS (wi-fi)
Accounts for: 1) RX_TX turnaround time
2) busy detect time
3) propagation delay

Backoff freezing

- When STA is in backoff stage:
  - It freezes the backoff counter as long as the channel is sensed BUSY
  - It restarts decrementing the backoff as the channel is sensed IDLE for a DIFS period

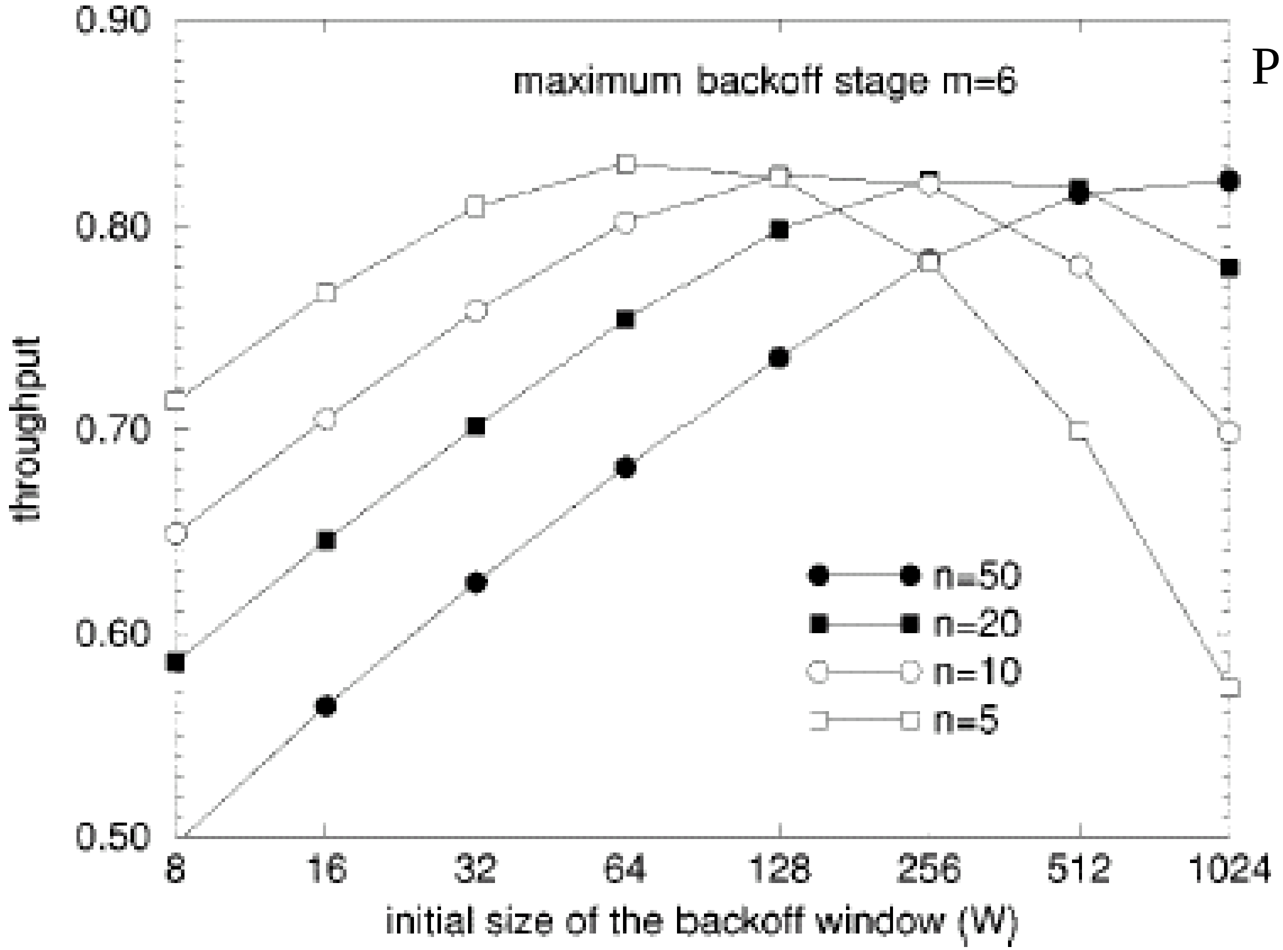


Backoff rules

- First backoff value:
  - Extract a uniform random number in range $(0, CW_{min})$
- If unsuccessful TX:
  - Extract a uniform random number in range $(0, 2 \times (CW_{min}+1)-1)$
- If unsuccessful TX:
  - Extract a uniform random number in range $(0, 2^2 \times (CW_{min}+1)-1)$
- Etc up to $2^m \times (CW_{min}+1)-1$

Exponential Backoff!
CWmin = 31
CWmax = 1023 (m=5)

Throughput vs CWmin



P=1000 bytes

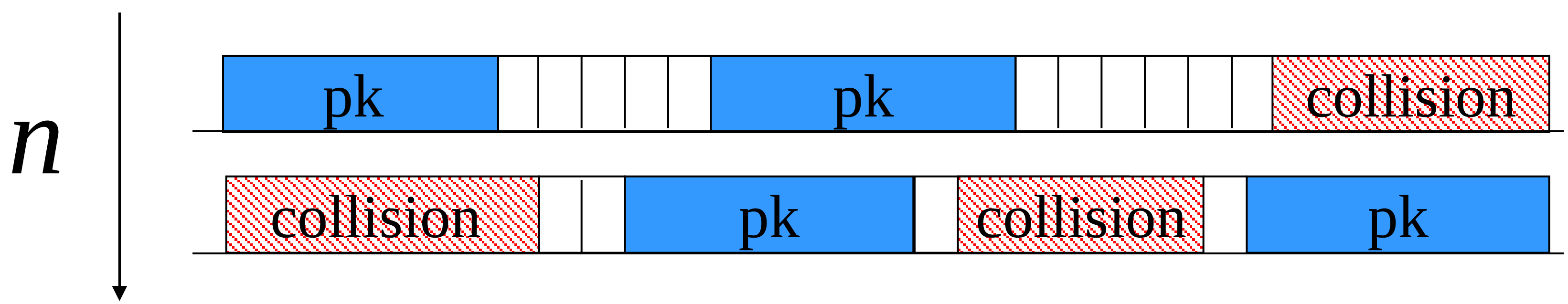
CWopt depends on n!!

Stima del numero di stazioni in reti WiFi

Who knows *n* in DCF?

In Distributed Systems:

- Lack of direct knowledge of traffic load
  - Even AP is not able to provide such an information: # of "associated" stations can be very different from the # of active stations
- **Possible indirect estimation** of traffic load via run-time monitoring of channel status



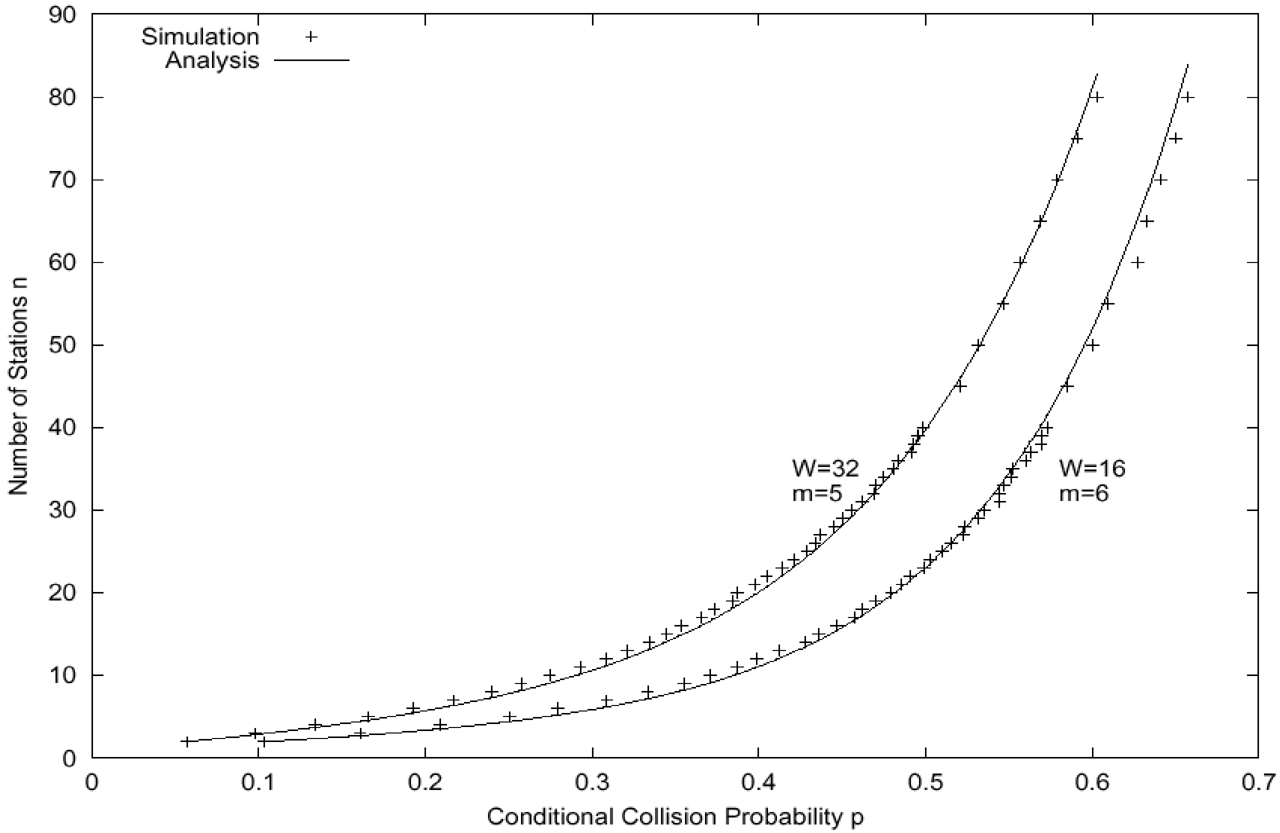
Theoretical finding

Assumption: active stations have always queues not empty
Based on previously developed time-discrte Markov model for CSMA/CA:

$$n = 1 + \frac{Log(1-p)}{Log\left(1 + \dfrac{2(1-2p)}{p(w+2) + pw(2p)^m - (w+1)}\right)}$$

n = number of active users
p = collision probability seen by a competing MS
w = min. contention window (32 for FHSS. 16 for DSSS)
m = backoff stages (5 for FHSS, 6 for DSSS)

Analysis vs. Simulation (accurate model < 3%)



n = f(p) : Strongly not linear relation!

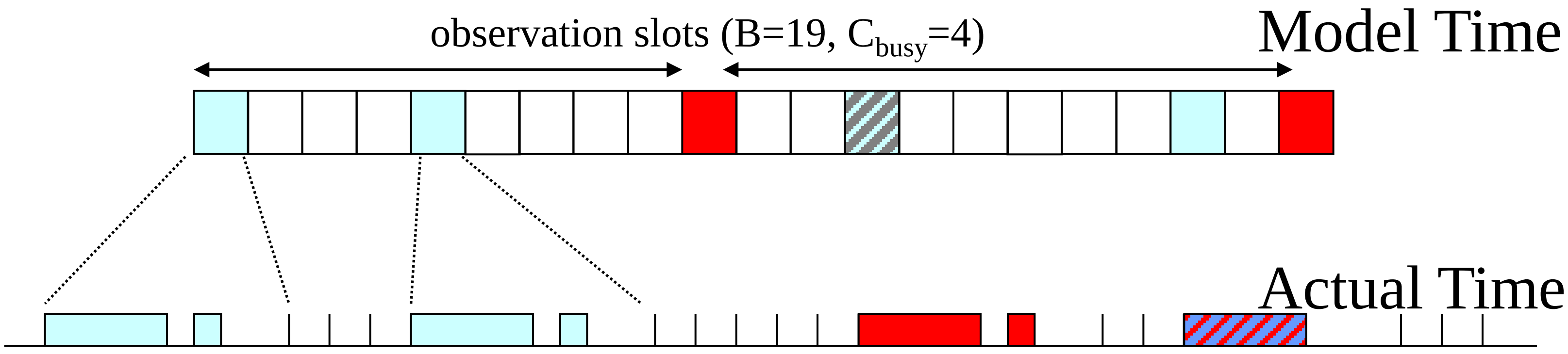
Key idea: monitoring the channel all the time!

..instead of considering only our ACK_TIMEOUTS.

- Direct **pc** measurement: by looking at all the slots in which the target station (e.g. station red) does not transmit

1) Busy slots = potential collisions
2) Idle slots = potential success

$\Rightarrow$ pc = Cbusy / B

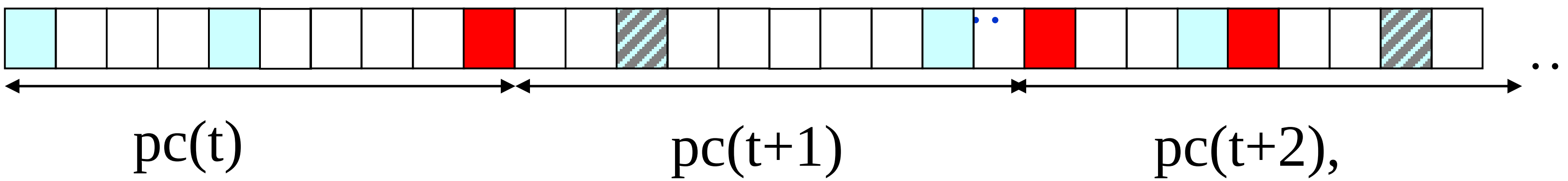


Run-Time Measurements

If stations activate/deactivate dynamically, congestion levels and interference conditions are time-varying.

Each station can perform its *pc* and *pr* measurements at different time-windows

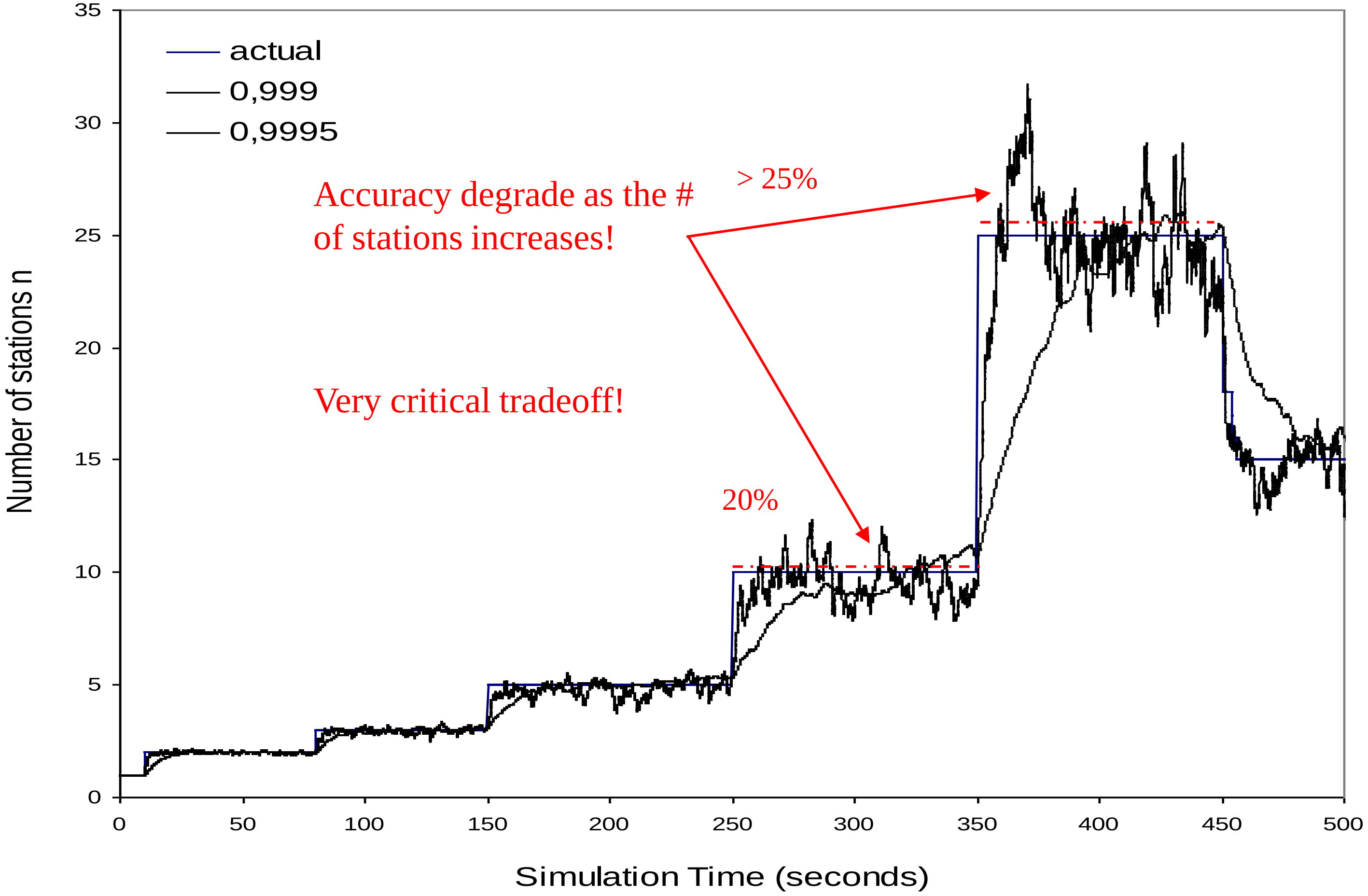
Measurement smoothing obtained by filtering



ARMA filters: usual accuracy/tracking tradeoff..

$$\begin{cases} \hat{p}(t+1) = \alpha \hat{p}(t) + \frac{(1-\alpha)}{q} \sum_{i=0}^{q-1} C_{t-i} \\ \\ \hat{n}(t+1) = f\left(\hat{p}(t+1)\right) \end{cases}$$

ARMA Filtering



Kalman Filtering

- Allows to obtain fast tracking in the transient phase and error minimizing in stationary conditions
  - It works as an AR filter with time-varying α coefficient
- Include available information (e.g. measurement statistics, non linearity) in the filtering process
- Model not available information (e.g. MS activation/deactivation) as a noise

$$n_k = n_{k-1} + w_k$$

$$p_k = f^{-1}(n_k) + v_k$$

Model

Kalman Filter Design (1)

Not coefficient settings, but model definition

a) system state = number of competing stations

b) observable parameter (measure) = collision probability

c) relation between state and measure = $f()$ or $f^{-1}()$

d) measure variance R_k = ?

e) state dynamic = state noise variance Q_k = ?

$$\begin{cases} n_k = n_{k-1} + w_k \\ p_k = f^{-1}(n_k) + v_k \end{cases}$$

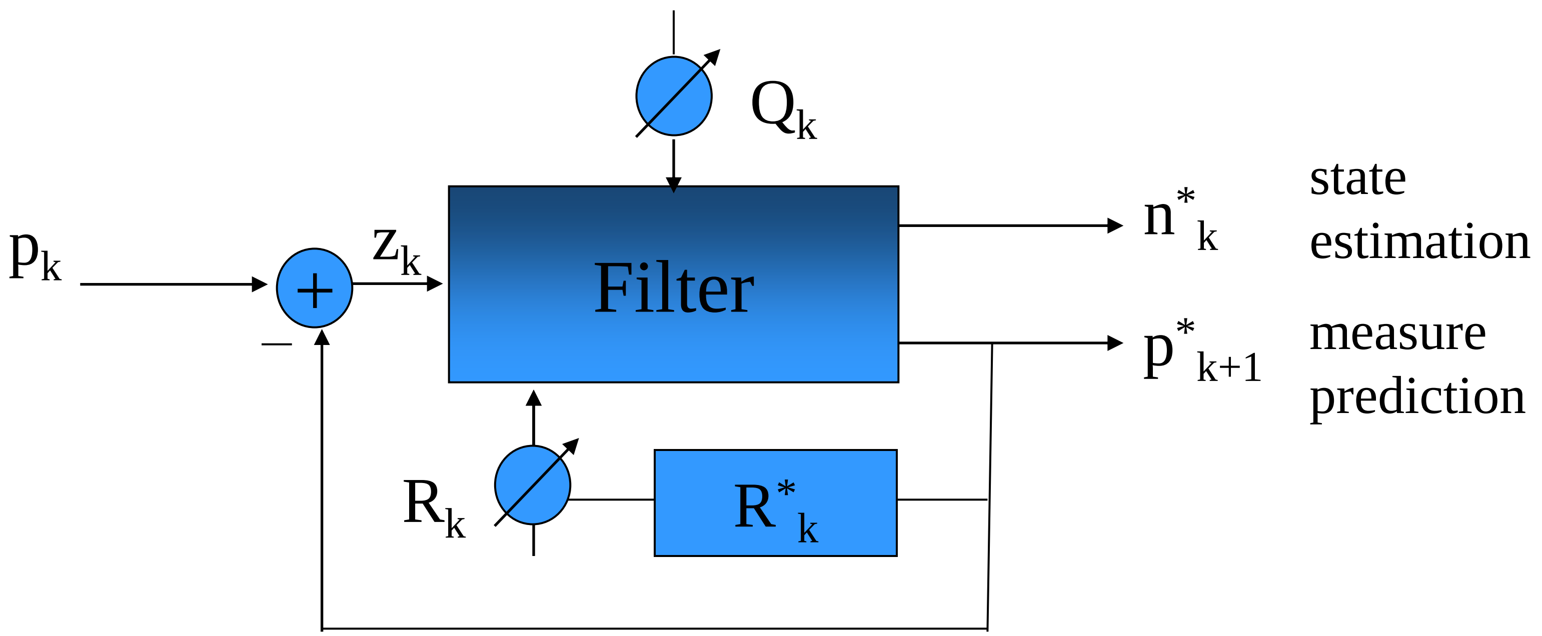
Kalman Filter Design (2)

From model to filter:

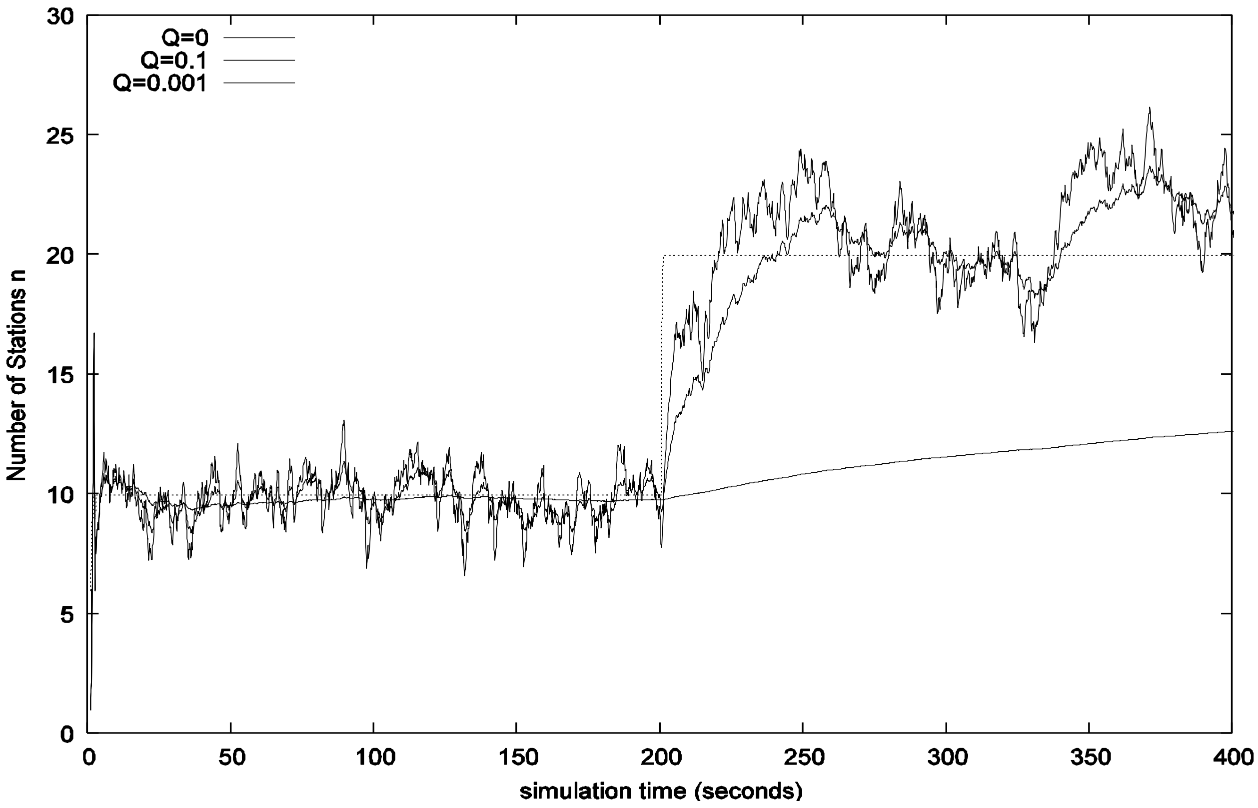
$z_k = p_k - p^*_k$ $\qquad n_k = n_{k-1} + K_k z_k$ $\qquad p^*_{k+1} = f^{-1}(n^*_k)$

Two design parameters: Q_k, R_k

Measure statistics can be estimated!



What value for Q_k?



Again accuracy/tracking tradeoff problem!

What new with respect to ARMA filter?

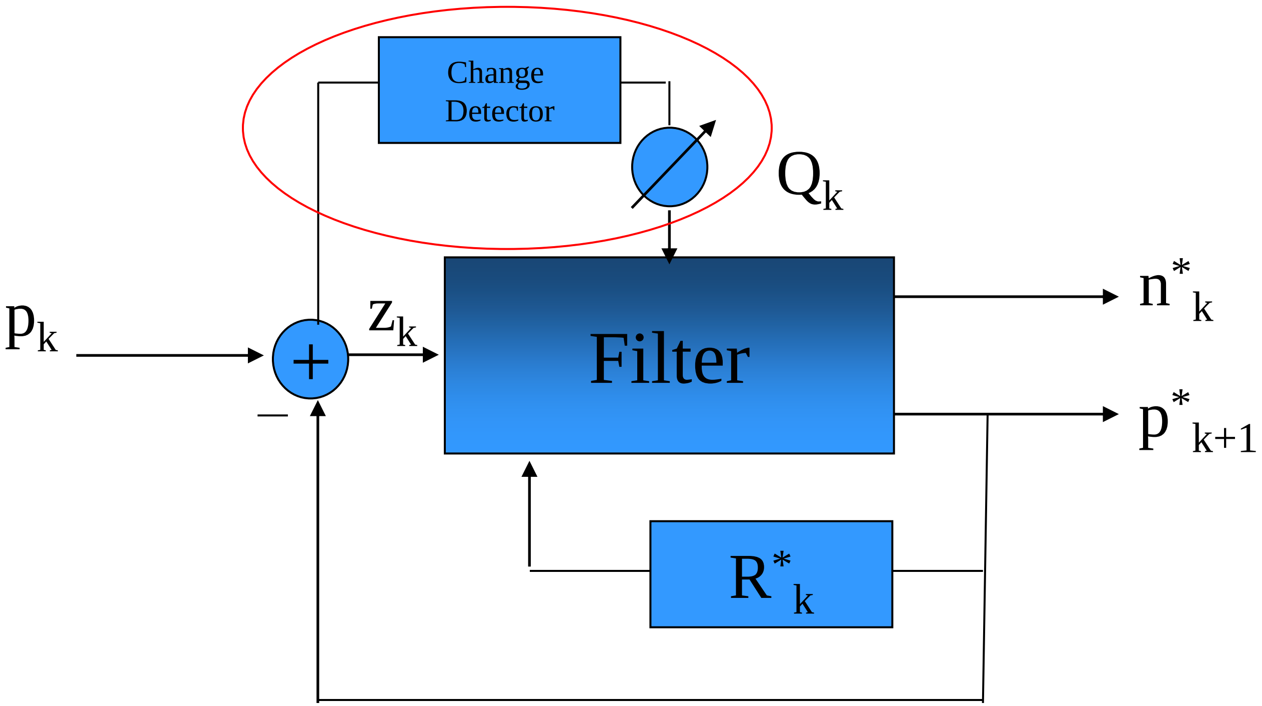
MATLAB DEMO

[Estimatation of a parameter x with step-wise dynamics]

```
function [xhat, P, K] = es_kalman(a, h, sw, sv, x0, xhat0, P0)
…
for i=2:1:N
    x(i)         = a*x(i-1)  + u(i) + sqrt(sw)*randn;
    z(i)         = h*x(i) + sqrt(sv)*randn;

    x_a_priori   = a*xhat(i-1);
    z_attesa     = h*x_a_priori;
    P_a_priori   = a^2*P(i-1) + sw;

    K(i)         = h*P_a_priori/(h^2*P_a_priori + sv);
    inn(i)       = z(i) - z_attesa;
    xhat(i)      = x_a_priori + K(i)*inn(i);
    P(i)         = (1 - h*K(i))*P_a_priori;
end
```

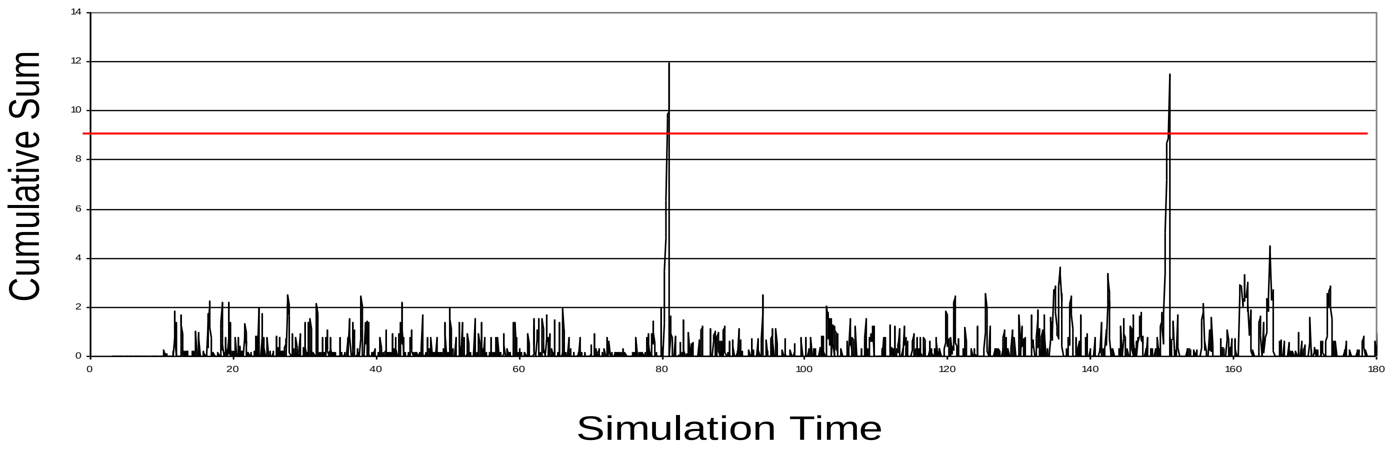
Automatic Q_k setting

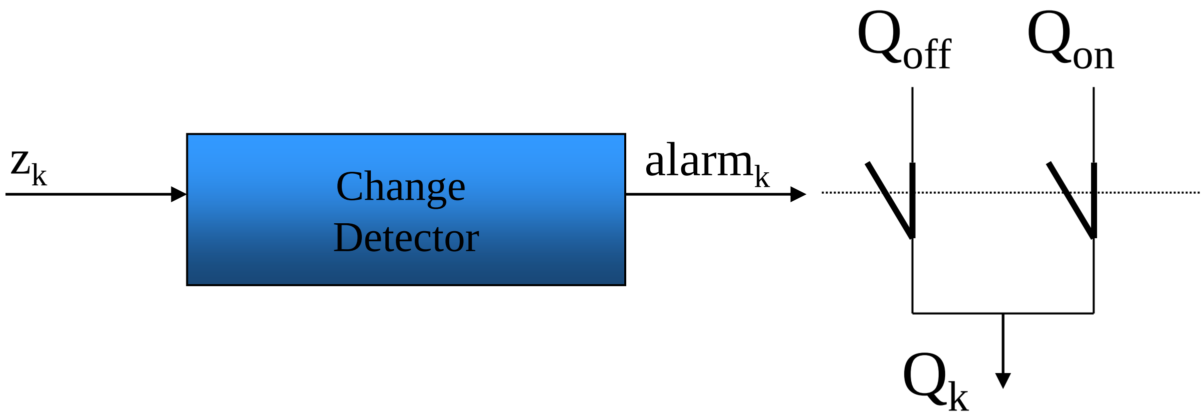
- Further filter (change detector) for Q_k driving based on innovations
  - Innovations process is a white process with zero mean in stationary conditions: the filtering can allow to distinguish between transient and stationary phases



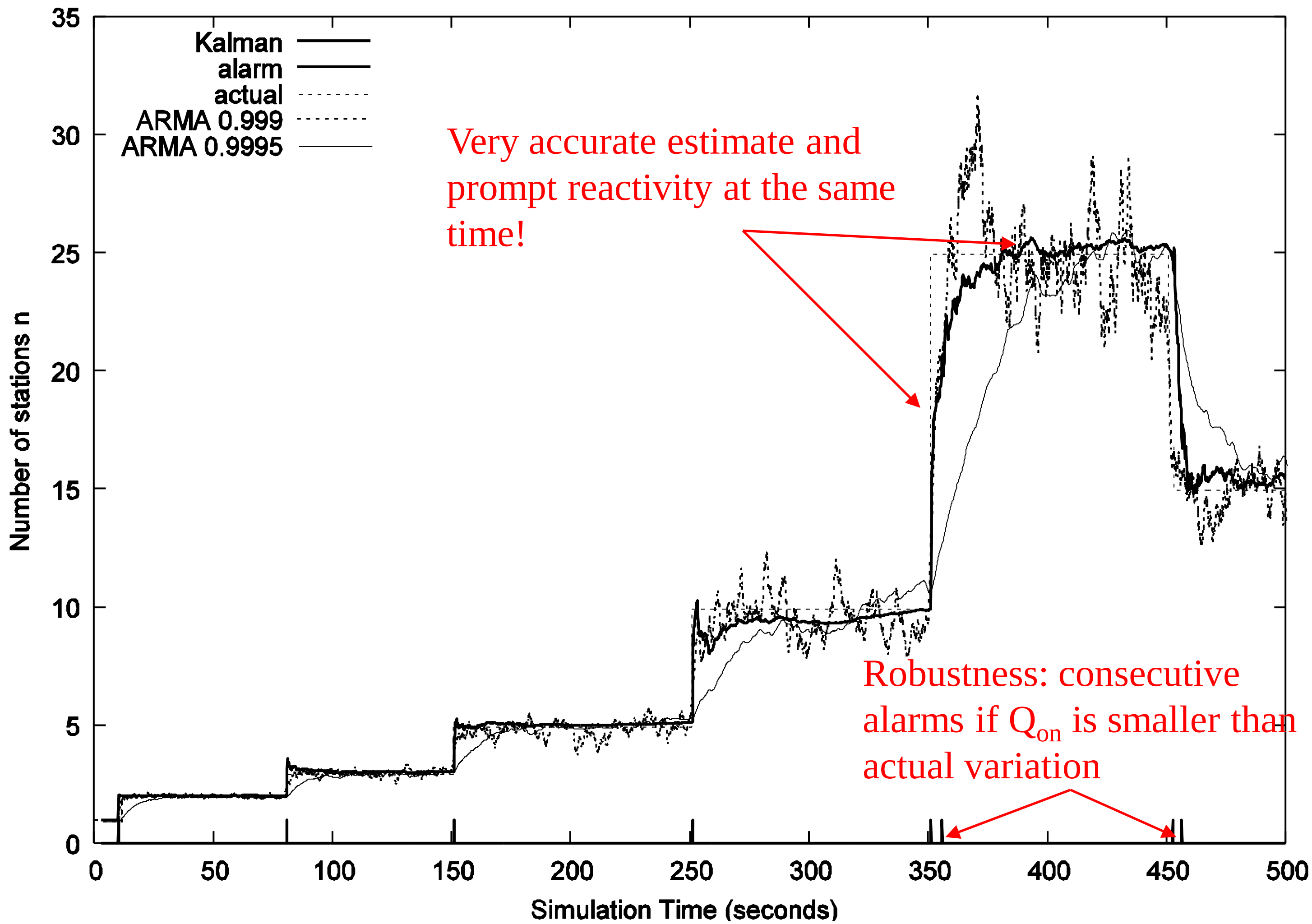
CUSUM Test for Q_k switching

- Basically: cumulative sum of innovations; alarm when such a sum overcomes a given threshold.





Performance Evaluation



MATLAB DEMO

[Estimatation of a parameter x with step-wise dynamics;
CUSUM test on innovation process]

```
function [xhat, P, K] = es_kalman(a, h, sw, sv, x0, xhat0, P0)
…
for i=2:1:N
    x(i)        = a*x(i-1)  + u(i) + sqrt(sw)*randn;
    z(i)        = h*x(i) + sqrt(sv)*randn;

    x_a_priori   = a*xhat(i-1);
    z_attesa     = h*x_a_priori;
    P_a_priori   = a^2*P(i-1) + sw;
    …
    cusum1(i) = max (cusum1(i)+inn(i)-0.5, 0);
    cusum2(i) = min (cusum2(i)-+nn(i)+0.5, 0);
end
```