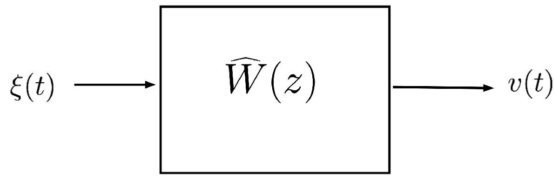
# Optimal Prediction from Data: an AR(1) Process

## The AR(1) Process

Consider an $AR(1)$ stationary stochastic process, described by the spectral canonical model




where

$$\widehat{W}(z) = \frac{1}{1 - a\,z^{-1}}\ , \quad |a| < 1\ , \qquad \xi(\cdot) \sim \mathrm{WN}\left(0, \lambda_\xi^2\right)$$

Hence the difference equation describing the process in the time domain is

$$v(t) = a\,v(t-1) + \xi(t)$$

### The $k$-Steps Ahead Predictor

The long division algorithm, performed for one, two, and three steps leads to the following results:

One step:

$$\widehat{W}(z) = 1 + \frac{a\,z^{-1}}{1 - a\,z^{-1}} = 1 + z^{-1}\left[\frac{a}{1 - a\,z^{-1}}\right] \quad \Longrightarrow \quad \widehat{W}_1(z) = \frac{a}{1 - a\,z^{-1}}$$

Two steps:

$$\widehat{W}(z) = 1 + a\,z^{-1} + \frac{a^2\,z^{-2}}{1 - a\,z^{-1}} = 1 + a\,z^{-1} + z^{-2}\left[\frac{a^2}{1 - a\,z^{-1}}\right] \quad \Longrightarrow \quad \widehat{W}_2(z) = \frac{a^2}{1 - a\,z^{-1}}$$

Three steps:

$$\widehat{W}(z) = 1 + a\,z^{-1} + a^2\,z^{-2} + \frac{a^3\,z^{-3}}{1 - a\,z^{-1}} = 1 + a\,z^{-1} + a^2\,z^{-2} + z^{-3}\left[\frac{a^3}{1 - a\,z^{-1}}\right] \quad \Longrightarrow \quad \widehat{W}_3(z) = \frac{a^3}{1 - a\,z^{-1}}$$

$r$-steps ahead:

$$
\begin{aligned}
\widehat{W}(z) &= 1 + az^{-1} + a^2 z^{-2} + \cdots + z^{-r}\,\frac{a^r}{1 - a\,z^{-1}} \\
&= 1 + az^{-1} + a^2 z^{-2} + \cdots + a^{r-1}\,z^{-r+1} + z^{-r}\left[\frac{a^r}{1 - a\,z^{-1}}\right]
\end{aligned}
\quad \Longrightarrow \quad \widehat{W}_r(z) = \frac{a^r}{1 - a\,z^{-1}}
$$

## The Variance of the Prediction Error

The variance of the prediction error is given by

One step:

$$\text{var}[\epsilon] = \lambda_\xi^2$$

Two steps:

$$\text{var}[\epsilon] = (1 + a^2)\,\lambda_\xi^2$$

Three steps:

$$\text{var}[\epsilon] = (1 + a^2 + a^4)\,\lambda_\xi^2$$

$r$ steps ahead:

$$\text{var}[\epsilon] = (1 + a^2 + a^4 + \cdots + a^{2(r-1)})\,\lambda_\xi^2$$

## Optimal Predictor from Data

The *whitening filter* is given by

$$\widetilde{W}(z) = \left[\widehat{W}(z)\right]^{-1} = \frac{A(z)}{1} = 1 - a\,z^{-1}$$

Thus, the optimal predictor from data is $\quad W_r(z) = \widetilde{W}(z) \cdot \widehat{W}_r(z)$

One step: $\quad W_1(z) = a \quad \Longrightarrow \quad \hat{v}(t+1|t) = a\,v(t)$

Two steps: $\quad W_2(z) = a^2 \quad \Longrightarrow \quad \hat{v}(t+2|t) = a^2\,v(t)$

Three steps: $\quad W_3(z) = a^3 \quad \Longrightarrow \quad \hat{v}(t+3|t) = a^3\,v(t)$

$r$ steps ahead: $$W_r(z) = a^r \implies \hat{v}(t+r|t) = a^r\, v(t)$$

## Hands-On: The $k$-Steps Ahead Predictor

```
clear
close all
clc
```

Consider an $AR(1)$ stationary process

```
a = 0.711; % select the AR parameter
```

Moreover, configure the stationary white noise process feeding in the $AR$ filter

```
lambda2_xi = 0.1% the noise variance
```

```
lambda2_xi = 0.1000
```

Choose how much AR process data you want to simulate and collect:

```
Ndata = 100;
```

Let's estimate how long is the initial transient output of the filter:

```
zero_threshold = 1e-6;
% let's assume that the value is practically zero if less than or equal to zero_thresh
Ntransient = ceil(log10(zero_threshold)/log10(abs(a)))+1;

N_TOT_data = Ndata + Ntransient;
```

Now simulate and collect the data

```
xi_data = sqrt(lambda2_xi) * randn(N_TOT_data,1);

v = zeros(N_TOT_data,1);
v(1) = xi_data(1); % initial condition ofr the AR(1) r.v.
```

```
for t=2:N_TOT_data
    v(t) = a*v(t-1)+xi_data(t);
end % for t

v = v(Ntransient+1:end); % blowing away the transient part
```

Evaluating the predictions of the one-step ahead and of the two-steps ahead predictors

```
v1 = zeros(size(v)); % the array containing the 1-step ahead predictor values
v2 = zeros(size(v)); % the array containing the 2-steps ahead predictor values

for t=2:Ndata
    v1(t) = a*v(t-1); % 1-step ahead predictor
end % for t

for t=3:Ndata
    v2(t) = a*a*v(t-2); % 2-steps ahead predictor
end % for t

var_v1 = lambda2_xi;
var_v2 = lambda2_xi*(1+a^2);
```

Choose the order of a $k$-steps ahead predictor $(3 \leq k \leq 20)$ to compare the performance with those of the others predictors

```
k_steps = 8; % the order of the k-steps ahead predictor
```

and evaluate the prediction

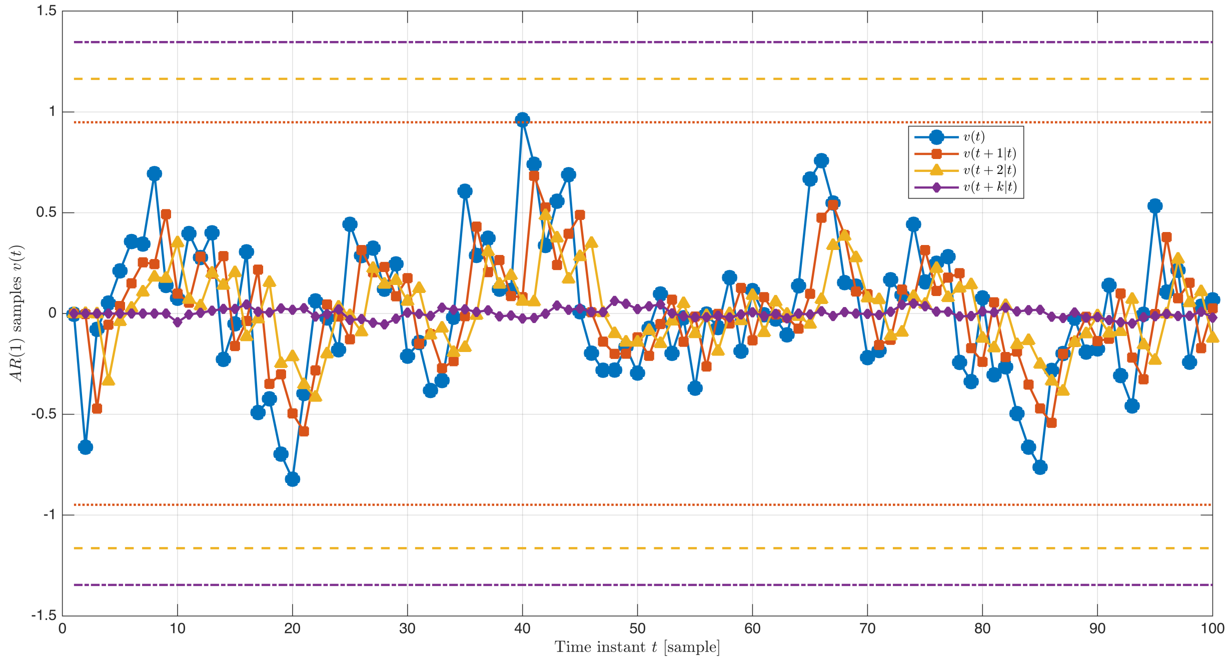
```
vk = zeros(size(v)); % the array containing the k-steps ahead predictor values
ak = a^(k_steps);
for t=(k_steps+1):Ndata
    vk(t) = ak*v(t-k_steps); % k-steps ahead predictor
end % for t

var_vk = 0;
for kk=1:k_steps
    var_vk = var_vk+a.^(2*(kk-1));
end % for kk
var_vk = var_vk*lambda2_xi;
```

Plotting the $AR(1)$ process realization toghether with the one-step, two-steps and three-steps predictor outputs.

Let's take into considerations also the expected value $\mathrm{E}[v(t)] = \bar{v} = 0$

```
bar_v = 0; % the expected value of the r.v. v

figure('Units','normalized','Position',[0.1, 0.1, 0.9, 0.8]);
plot(v, 'LineStyle','-', 'LineWidth', 1.5, 'Marker','o', 'MarkerSize', 9, ...
         'MarkerEdgeColor',[0, 0.4470, 0.7410],...
         'MarkerFaceColor',[0, 0.4470, 0.7410], 'Color',[0, 0.4470, 0.7410]);
grid on
hold on
xlabel('Time instant $t$ [sample]', 'Interpreter', 'latex');
ylabel('$AR(1)$ samples $v(t)$', 'Interpreter','latex');

plot(v1, 'LineStyle','-', 'LineWidth', 1.5, 'Marker','square', 'MarkerSize', 7, ...
          'MarkerEdgeColor',[0.8500 0.3250 0.0980],...
    'MarkerFaceColor',[0.8500 0.3250 0.0980], 'Color',[0.8500 0.3250 0.0980]);

high_threshold_v1 = (bar_v+3*sqrt(var_v1)).*ones(size(v1));
low_threshold_v1 = (bar_v-3*sqrt(var_v1)).*ones(size(v1));
plot(high_threshold_v1, 'LineStyle',':', 'LineWidth', 1.5, 'Color',[0.8500 0.3250 0.098
plot(low_threshold_v1, 'LineStyle',':', 'LineWidth', 1.5, 'Color',[0.8500 0.3250 0.098

%p = patch('XData', [1:Ndata,  Ndata:-1:1], 'YData', [v1-3*sqrt(var_v1) fliplr(v1+3*sq

plot(v2, 'LineStyle','-', 'LineWidth', 1.5, 'Marker','^', 'MarkerSize', 7,...
           'MarkerEdgeColor',[0.9290 0.6940 0.1250],...
            'MarkerFaceColor',[0.9290 0.6940 0.1250], 'Color',[0.9290 0.6940 0.1250]);

high_threshold_v2 = (bar_v+3*sqrt(var_v2)).*ones(size(v2));
low_threshold_v2 = (bar_v-3*sqrt(var_v2)).*ones(size(v2));
plot(high_threshold_v2, 'LineStyle','--', 'LineWidth', 1.5, 'Color',[0.9290 0.6940 0.1
plot(low_threshold_v2, 'LineStyle','--', 'LineWidth', 1.5, 'Color',[0.9290 0.6940 0.12
```

```
plot(vk, 'LineStyle','-', 'LineWidth', 1.5, 'Marker','diamond', 'MarkerSize', 5,...
           'MarkerEdgeColor',[0.4940 0.1840 0.5560],...
            'MarkerFaceColor',[0.4940 0.1840 0.5560], 'Color',[0.4940 0.1840 0.5560]);

high_threshold_vk = (bar_v+3*sqrt(var_vk)).*ones(size(vk));
low_threshold_vk = (bar_v-3*sqrt(var_vk)).*ones(size(vk));
plot(high_threshold_vk , 'LineStyle','-.', 'LineWidth', 1.5, 'Color',[0.4940 0.1840 0.
plot(low_threshold_vk, 'LineStyle','-.', 'LineWidth', 1.5, 'Color',[0.4940 0.1840 0.55

legend('$v(t)$', '$v(t+1|t)$', '','','$v(t+2|t)$', '','','$v(t+k|t)$', '','', 'Interpr
    'Location', 'best')
```




## Using Several Predictions at Once

Given the AR(1) process under consideration, let us consider the N samples collected. How will the process evolve? We use the samples to provide one-step forward, two-step forward etc. estimates.

Simply, we can estimate the future evolution (with respect to the time instant $N$) of the $AR(1)$ process by using the one-step predictor, the two-steps predictor and so on, all applied to the observation obtained at time instant $N$

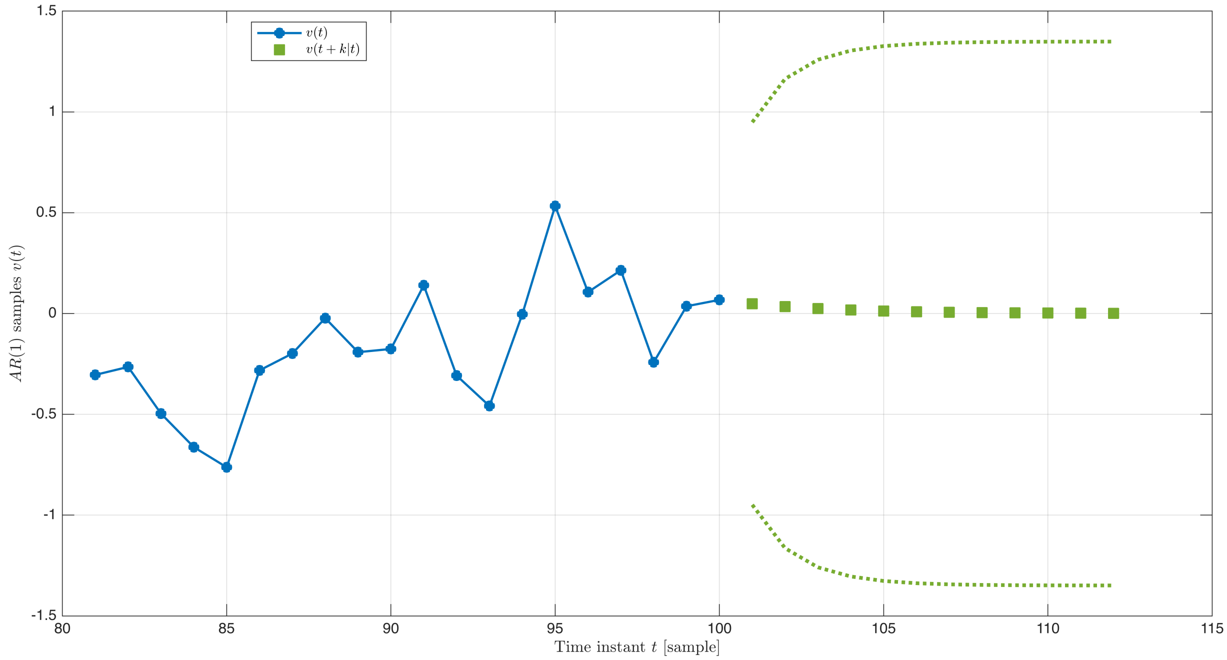
$$\hat{v}(N+1|t) = a\,v(N) \quad \hat{v}(N+2|N) = a^2\,v(N) \quad \cdots \quad \hat{v}(N+r|N) = a^r\,v(N) \quad \cdots$$

Consider the r.v. $v(t)$ at the time instant $t = N$ and then evaluate the predictors

```
vN = v(end); % pick the last collected value of the r.v.

pred_H = 12; % select the prediction horizon

v_pred = zeros(size(pred_H,1));    % preallocate the array devoted to store the predict
var_vpred = zeros(size(pred_H,1));% preallocate the array devoted to store the varianc
```

```
dummy_v = 0; % temporary storage, useful when evaluating recursively the prediction va
for k=1:pred_H
    v_pred(k) = (a.^k)*vN; % the k-steps ahead predicted value
    var_vpred(k) = dummy_v + +a.^(2*(k-1));
    dummy_v = var_vpred(k);
end % for k
var_vpred = var_vpred * lambda2_xi;
```

```
figure('Units','normalized','Position',[0.1, 0.1, 0.9, 0.8]);
% consider the last M data only
M = 20;
Ninit = Ndata - M+1;
timeInstants = (Ninit:Ndata);
AR_vM = v(Ninit:end);
plot(timeInstants, AR_vM, 'LineStyle','-', 'LineWidth', 1.5, 'Marker','o', 'MarkerSize
         'MarkerEdgeColor',[0, 0.4470, 0.7410],...
         'MarkerFaceColor',[0, 0.4470, 0.7410], 'Color',[0, 0.4470, 0.7410]);
grid on
hold on
xlabel('Time instant $t$ [sample]', 'Interpreter', 'latex');
ylabel('$AR(1)$ samples $v(t)$', 'Interpreter','latex');

future_timeSteps = (Ndata+1:Ndata+pred_H);
plot(future_timeSteps, v_pred, 'LineStyle','none','Color',[0.4660 0.6740 0.1880], 'Mar
    'MarkerSize', 9, 'MarkerFaceColor', [0.4660 0.6740 0.1880], 'MarkerEdgeColor', [0.
high_threshold_vk = (bar_v+3*sqrt(var_vpred));
low_threshold_vk = (bar_v-3*sqrt(var_vpred));
plot(future_timeSteps,high_threshold_vk , 'LineStyle',':', 'LineWidth', 2.5, 'Color',[
plot(future_timeSteps,low_threshold_vk, 'LineStyle',':', 'LineWidth', 2.5, 'Color',[0.
legend('$v(t)$', '$v(t+k|t)$', '','', 'Interpreter', 'latex', ...
    'Location', 'best')
```



1.5
1
0.5
0
-0.5
-1
-1.5
$AR(1)$ samples $v(t)$
80
85
90
95
100
105
110
115
Time instant $t$ [sample]
$v(t)$
$v(t+k|t)$