1. Identification of FIR Process

Generate input $u(n)$, white noise $\xi(n)$ and the corresponding output $y(n)$ of:

$$y(n) = \frac{0.6 - 0.2z^{-1}}{1 - 0.5z^{-1}} u(n-1) + \xi(n).$$

From the input-output data, determine the underlying FIR model.
Expanding it in power series,

$$\begin{aligned} y(n) &= (0.6 - 0.2z^{-1})(1 + 0.5z^{-1} + 0.25z^{-2} + \cdots)u(n-1) + \xi(n) \\ &= (0.6 + 0.1z^{-1} + 0.05z^{-2} + 0.025z^{-3} + \cdots)u(n-1) + \xi(n) \end{aligned}$$

As this is a fast decaying sequence, it can be approximated well by a FIR model.
Recall a old fact:
Consider an LTI system, $\xi(k)$ is noise, u and ξ uncorrelated, *i.e.*, $r_{u\xi}(k) = 0$:

$$\begin{bmatrix} r_{uu}(0) & \cdots & r_{uu}(N) \\ r_{uu}(-1) & \cdots & r_{uu}(N-1) \\ \vdots & & \\ r_{uu}(-N) & \cdots & r_{uu}(0) \end{bmatrix} \begin{bmatrix} g(0) \\ g(1) \\ \vdots \\ g(N) \end{bmatrix} = \begin{bmatrix} r_{yu}(0) \\ r_{yu}(1) \\ \vdots \\ r_{yu}(N) \end{bmatrix}.$$

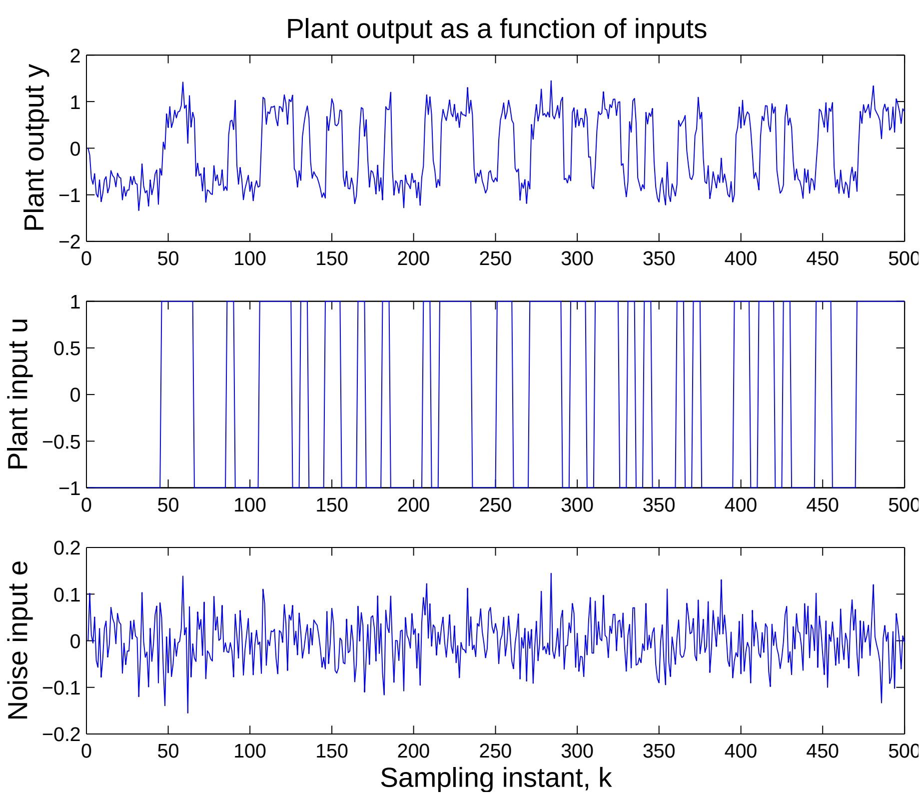
Done by `cra` command of Matlab

2. Matlab Code to Demonstrate cra

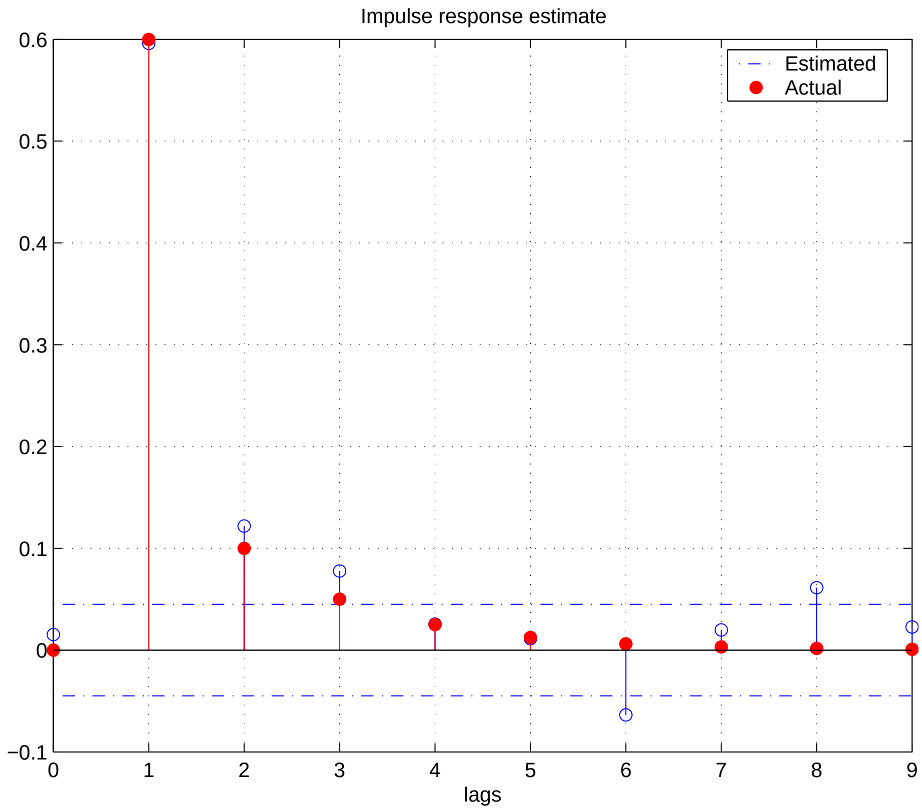
```
% Create the plant and noise model objects
  var = 0.05;
  process_mod = idpoly(1,[0 0.6 -0.2], 1, 1,...
         [1 -0.5],'Noisevariance',var,'Ts',1);
% Create input sequence
  u = idinput(2555,'prbs',[0 0.2],[-1 1]);
  e = randn(2555,1);
% Simulate the process
  y = idsim([u e],process_mod);
% Plot y as a function of u and e
  subplot(3,1,1), plot(y(1:500)),
  title('Plant output as a function of inputs',...
         'FontSize',14)
  ylabel('Plant output y','FontSize',14)
  subplot(3,1,2), plot(u(1:500))
  ylabel('Plant input u','FontSize',14)
  subplot(3,1,3), plot(var*e(1:500))
  ylabel('Noise input e','FontSize',14)
  xlabel('Sampling instant, k','FontSize',14)
% Build iddata objects
```

```
z = iddata(y,u,1);
% Compute impulse response using
% CRA after removal of means
figure; [ir,r,cl] = cra(detrend(z,'constant'));
hold on
% Compare the first 10 impulse response
% computed from G(q)
ir_act = filter([0 0.6 -0.2],[1 -0.5],...
                [1 zeros(1,9)]);
% Plot the actual IR
set(gca,'XLim',[0 9]); grid on;
h_act = stem((0:9),ir_act,'ro','filled');
% Add legend
ch_f = get(gcf,'Children');
ch_f2 = get(ch_f,'Children');
legend([ch_f2(5) h_act(1)],...
            {'Estimated'; 'Actual'});
```


3. Input-Output Plots in FIR Example



4. Predicted and estimated impulse response coefficients



5. ARX Model: Example of Equation Error Model

$$A(z)y(k) = B(z)u(k) + \xi(k)$$

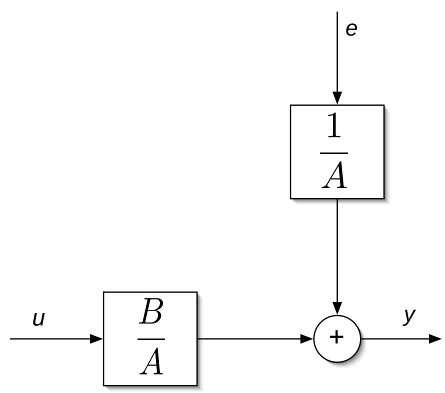
where,

$$A(z) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots$$
$$B(z) = b_1 z^{-1} + b_2 z^{-2} + \cdots$$

Substituting,

$$y(k) = -a_1 y(k-1) - a_2 y(k-2) - \cdots + b_1 u(k-1) + b_2 u(k-2) + \cdots + \xi(k)$$

$\xi(k)$ appears in equation, as opposed to appearing in the output directly. So known as Equation Error Model.



6. ARX Model as a Regression Equation

$$y(k) = a_1 y(k-1) + \sum_{l=0}^{N} g(l)u(k-l) + \xi(k)$$

$$\begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \end{bmatrix} = \begin{bmatrix} y(k-1) & u(k) & \cdots & u(k-N) \\ y(k-2) & u(k-1) & \cdots & u(k-N-1) \\ \vdots & & & \end{bmatrix} \begin{bmatrix} a_1 \\ g(0) \\ g(1) \\ \vdots \\ g(N) \end{bmatrix} + \begin{bmatrix} \xi(k) \\ \xi(k-1) \\ \vdots \end{bmatrix}$$

This is in the form of $Z(k) = \Phi(k)\theta + \Xi(k)$ with

$$Z(k) = \begin{bmatrix} y(k) \\ y(k-1) \\ \vdots \end{bmatrix}, \ \Phi(k) = \begin{bmatrix} y(k-1) & u(k) & \cdots & u(k-N) \\ y(k-2) & u(k-1) & \cdots & u(k-N-1) \\ \vdots & & & \end{bmatrix}, \ \theta = \begin{bmatrix} a_1 \\ g(0) \\ g(1) \\ \vdots \\ g(N) \end{bmatrix}$$

Note that θ consists of a_1 and the impulse response coefficients $g(0)$, $\ldots$, $g(N)$.

7. ARX Model: Example of Equation Error Model

Generate input, $u(n)$, white noise, $\xi(n)$ and the corresponding output $y(n)$ of the ARX model

$$y(n) - 0.5y(n-1) = 0.6u(n-2) - 0.2u(n-3) + \xi(n)$$

From the input-output data, determine the underlying ARX model.

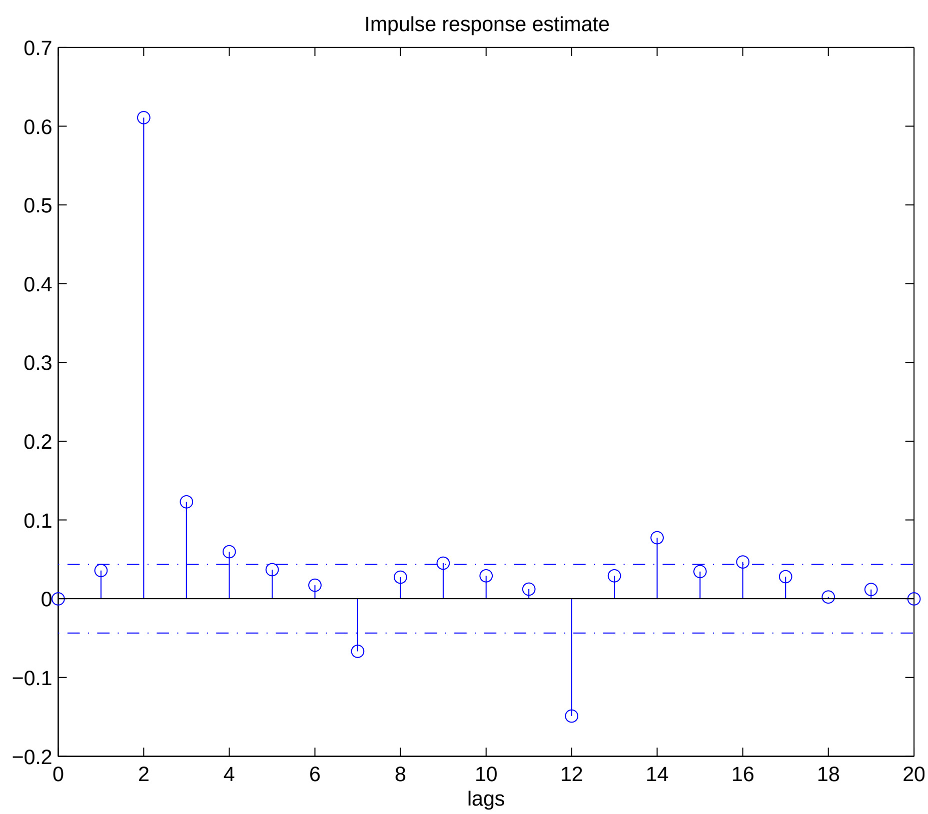
Code:

```
% Create the plant and noise model objects
 process_arx = idpoly([1 -0.5],[0 0 0.6 -0.2],...
                  1,1,1,'Noisevariance',0.05,'Ts',1);

% Create input sequence and simulate
 u = idinput(2555,'prbs',[0 0.2],[-1 1]);
 e = randn(2555,1);
 y = idsim([u e],process_arx);

% Build iddata objects and remove means
 z = iddata(y,u,1); zd = detrend(z,'constant');
```

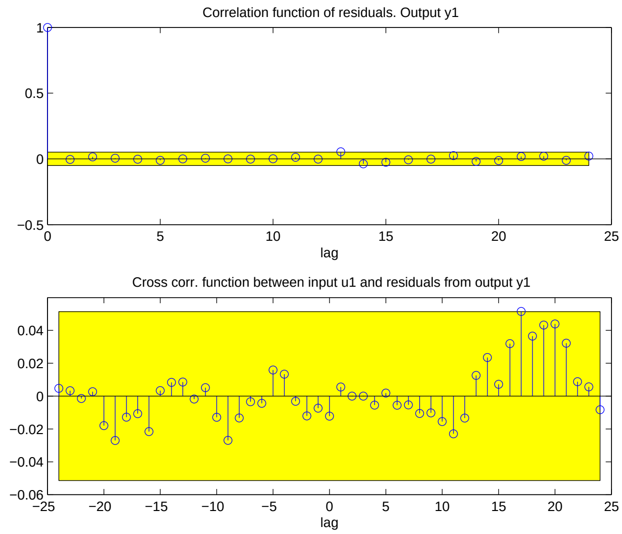
```
% Compute IR for time-delay estimation
figure; [ir,r,cl] = cra(zd);

% Time-delay = 2 samples
% Estimate ARX model (assume known orders)
na = 1; nb = 2; nk = 2;
theta_arx = arx(zd,[na nb nk])

% Present the model
present(theta_arx)

% Check the residual plot
figure; resid(theta_arx,zd);
```

- `resid` plots $\gamma_{(y-\hat{y})(y-\hat{y})}(n)$, $\gamma_{(y-\hat{y})u}(n)$

8. Impulse response coefficients



9. Impulse response coefficients




```
Discrete-time IDPOLY model: A(q)y(t) = B(q)u(t) + e(t)
A(q) = 1 - 0.4878 (+-0.01567) q^-1
B(q) = 0.604 (+-0.00763) q^-2 - 0.1887 (+-0.01416) q^-3
```

$$y(n) - 0.5y(n-1) = 0.6u(n-2) - 0.2u(n-3) + \xi(n)$$

10. Case Study (case1.m)

Determine the model from the data generated using

$$y(n) = \frac{1.2z^{-1} + 0.1z^{-2}}{1 - z^{-1} + 0.2275z^{-2}} u(n) + \frac{1}{1 - 0.94z^{-1}} e(n)$$

without using the knowledge of the order of the model.
code: case1.m