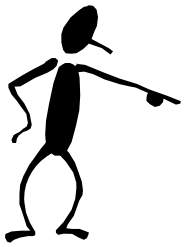
Computer Process Control Using MATLAB



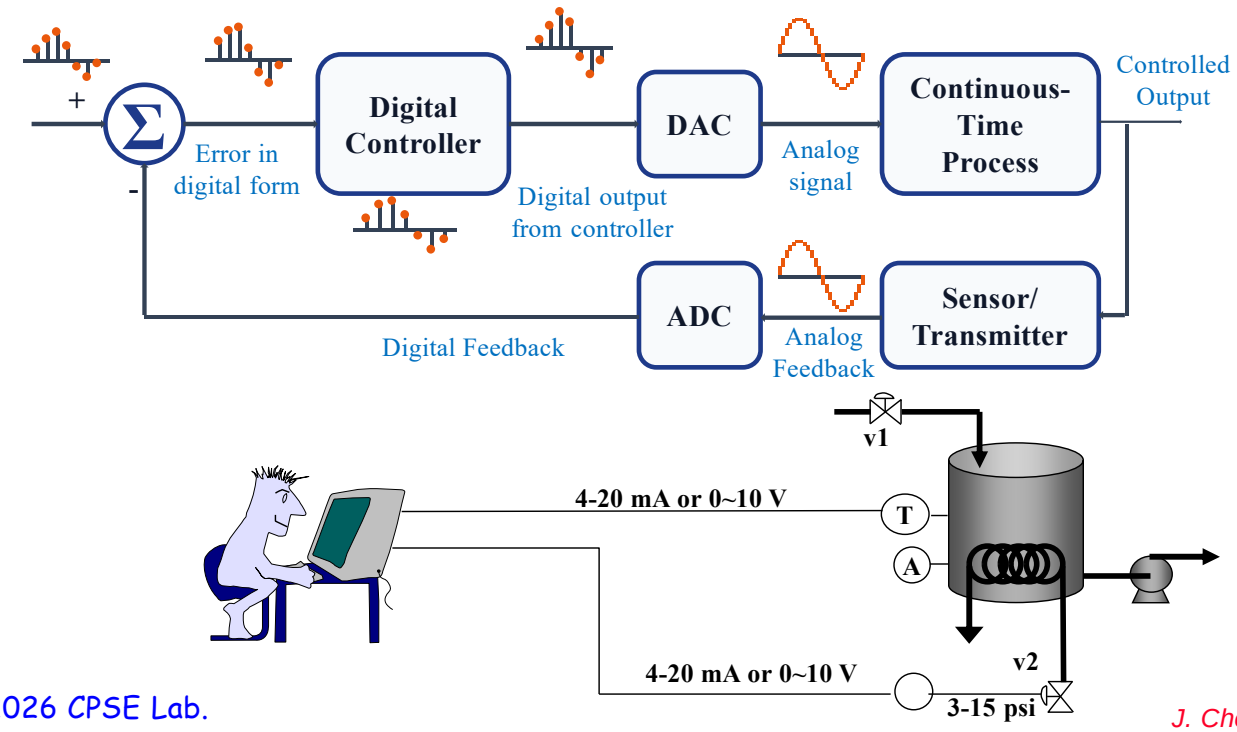


陳 榮 輝 (J. Jason Chen)
中原大學化工系
jason@wavenet.cycu.edu.tw

(C) 2026 CPSE Lab.

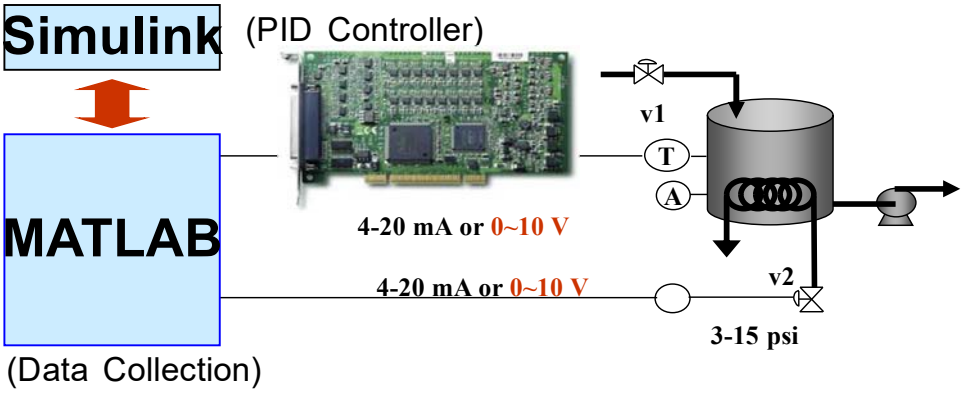
COMPUTER PROCESS CONTROL

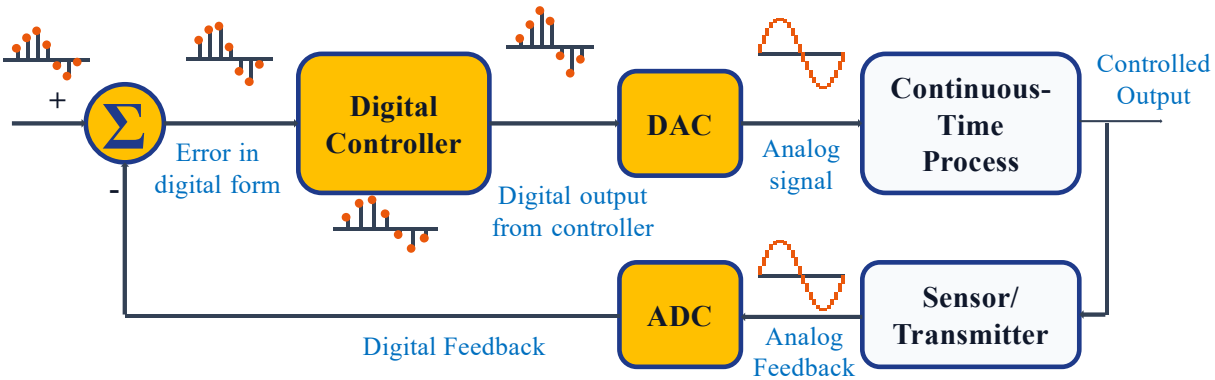
In computer process control, a **digital computer** is used to direct the operations of a manufacturing process



(C) 2026 CPSE Lab.

COMPUTER PROCESS CONTROL





(C) 2026 CPSE Lab.

J. Chen

COMPUTER CONTROL: MATLAB & SIMULINK CODES

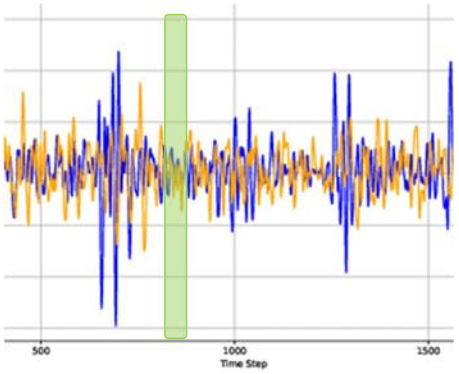


```
clc;
Clear;
Close all;
%% DAQ setup
BDaq = NET.addAssembly('Automation.BDaq4');
aoCtrl = Automation.BDaq.InstantAoCtrl();
aiCtrl.SelectedDevice = Automation.BDaq.DeviceInformation('PCI-
  1716');

%% ===== Channels =====
aoChannel = int32(1);    % Valve 1 / MV1
aiChannel = int32(1);    % Tank 3 / P3

%% ===== Timing =====
Ts = 0.5;
SP1_N = round(200/Ts);     % 200 s
SP2_N = round(200/Ts);     % 200 s
FBC_N = SP1_N + SP2_N;

% 10 samples x 0.5 s = about 5-s moving average
MovingWind = 10;
```



(C) 2026 CPSE Lab.

J. Chen

COMPUTER CONTROL: MATLAB & SIMULINK CODES

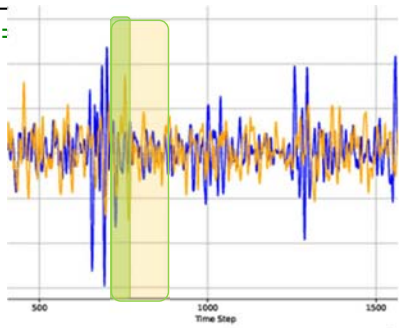
```
%% ===== Operating point for PRECONDITION only =
MV0 = 1.80;
PV0_ref = 0.400;

% Recent 60-s stability window
stableWindowSec = 60;
stableWindowN = round(stableWindowSec/Ts);

% These values are intentional tolerance levels
PV_mean_tolerance = 0.010;   % mean must be within 0.390~0.410 V
PV_std_limit      = 0.015;   % recent 60-s std
PV_drift_limit    = 0.015;   % total fitted drift over 60 s

% Require several consecutive successful checks.
readyHoldChecks = 10;

% 600 s maximum precondition wait
maxPreconditionSec = 600;
maxPreconditionN = round(maxPreconditionSec/Ts);
```



(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
%% ============================================================
% PRESSURE DISCRETE ONE-STEP MODEL at Ts = 0.5 s
%% ============================================================
a_model = 0.9877495516;
b_model = -0.0012364569;
d_model = 0.0072965877;
Gamma = 0;

%% ===== Storage =====
maxN = maxPreconditionN + FBC_N + 20;
T_data         = zeros(maxN,1);
Phase_data     = zeros(maxN,1);
SP_data        = NaN(maxN,1);
MV1_data       = zeros(maxN,1);
PV_data        = zeros(maxN,1);
Avg_PV_data    = zeros(maxN,1);
PV_model_data  = NaN(maxN,1);
dMV_data       = zeros(maxN,1);
ReadyMean_data  = NaN(maxN,1);
ReadyStd_data   = NaN(maxN,1);
ReadyDrift_data = NaN(maxN,1);

% Actual MV history for u(k-Gamma)
MV_history = MV0*ones(maxN + Gamma + 20,1);
```

$$y_k = e^{-\Delta t/\tau} y_{k-1} + K_p(1-e^{-\Delta t/\tau})u_{k-\Gamma-1} + c(1-e^{-\Delta t/\tau})$$

(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

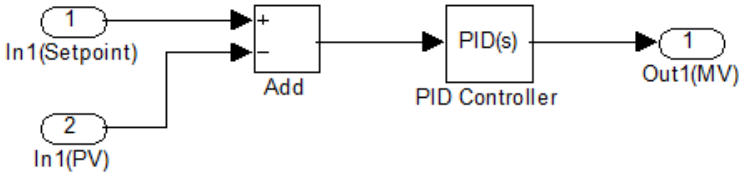
```
%% ===== States =====
readyCheckCount = 0;
fbcStarted = 0;
fbcIndex = 0;

d_MV = 0;
d_MV_old = 0;

% Prediction made on previous loop for current sample
PV_pred_next = NaN;
predictionValid = 0;

%% ===== Initial hardware output =====
aoCtrl.Write(aoChannel,MV0);

%% ===== Load Simulink =====
sim_model = 'PID_Cont';
load_system(sim_model);
```



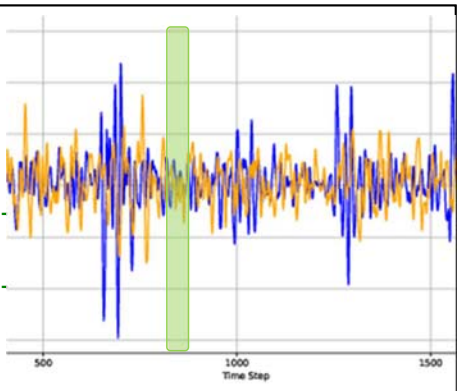
(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
experiment_start = tic;
i = 0;
while 1
    i = i + 1;
    loop_start = tic;
    %% ------------------------------------
    % Read ACTUAL P3
    %% ------------------------------------
    [status,data] = aiCtrl.Read(aiChannel);
    PV = data;
    PV_data(i) = PV;

    %% ------------------------------------------------------
    % Moving-average ACTUAL feedback value
    %% ------------------------------------------------------
    if i >= MovingWind
        Avg_PV = mean(PV_data(i-MovingWind+1:i));
    else
        Avg_PV = mean(PV_data(1:i));
    end

    Avg_PV_data(i) = Avg_PV;
```



(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
%% ====================================================
% PRECONDITION
%% ====================================================
if fbcStarted == 0
    Phase = 0;
    SP = NaN;
    MV1 = MV0;
    MV1_old = MV0;

    aoCtrl.Write(aoChannel,MV1);
    MV_history(i) = MV1;

    d_MV = 0;
    d_MV_old = 0;

    PV_model = NaN;

    readyMean  = NaN;
    readyStd   = NaN;
    readyDrift = NaN;
```

(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

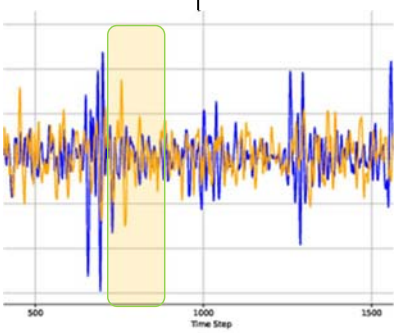
```
    if i >= stableWindowN
       % Use recent 60 s of MOVING-AVERAGED ACTUAL P3
       w = Avg_PV_data(i-stableWindowN+1:i);
       readyMean = mean(w);
       readyStd  = std(w);

       x = (0:stableWindowN-1)';
       p = polyfit(x,w,1);

       % Total fitted change across the 60-s window
       readyDrift = p(1)*(stableWindowN-1);
       meanOK  = abs(readyMean-PV0_ref) <= PV_mean_tolerance;
       stdOK   = readyStd <= PV_std_limit;
       driftOK = abs(readyDrift) <= PV_drift_limit;

       if meanOK && stdOK && driftOK
           readyCheckCount = readyCheckCount + 1;
       else
           readyCheckCount = 0;
       end

   else
       readyCheckCount = 0;
   end
```



(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
if i >= maxPreconditionN && fbcStarted == 0

    aoCtrl.Write(aoChannel,MV0);

    error(['Preconditioning timeout. ' ...
           'Plant did not satisfy READY conditions.']);

end
```

(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
%% ---------- Start FBC ----------
if readyCheckCount >= readyHoldChecks

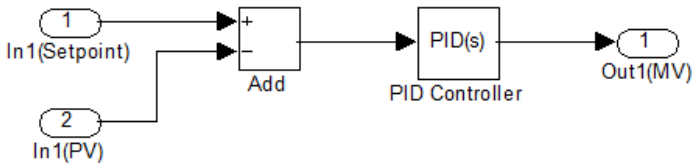
    % Initialize Simulink PI states ONLY
    [sizes,x0] = feval(sim_model,[],[],[],0);

    fbcStarted = 1;
    fbcIndex = 0;

    MV1_old = MV0;
    d_MV_old = 0;

    % Past MV values are nominal MV0
    MV_history(1:i) = MV0;

    % No recursive model state
    PV_pred_next = NaN;
    predictionValid = 0;
```



(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
    else
        %% ================================================
        % CLOSED-LOOP FBC
        %% ================================================
        fbcIndex = fbcIndex + 1;
        if fbcIndex <= SP1_N
            Phase = 1;
            SP = SP1;
        else
            Phase = 2;
            SP = SP2;
        end

        %% ------------------------------------------------
        % Prediction for CURRENT sample
        % made on previous loop using ACTUAL Avg_PV
        %% ------------------------------------------------
        if predictionValid == 1
            PV_model = PV_pred_next;
        else
            PV_model = NaN;
        end
```

(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

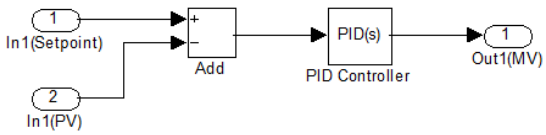
```
        %% ------------------------------------------------
        % Simulink PI CONTROLLER ONLY
        % Out1 = dMV
        %% ------------------------------------------------
        Utmp = [0,  SP, Avg_PV; ...
                Ts, SP, Avg_PV];

        set_param(sim_model,'InitInArrayFormatMsg','None');
        simopts = simset('InitialState',x0);

        [h,x0,sim_out] = ...
            sim(sim_model,[0 Ts],simopts,Utmp);
        x0 = x0(size(x0,1),:)';
        d_MV = sim_out(end,1);

        %% ------------------------
        % MV update
        % EXACT SAME successful liquid-level logic
        %% ------------------------------------------------

        MV1 = MV1_old + (d_MV-d_MV_old);
```



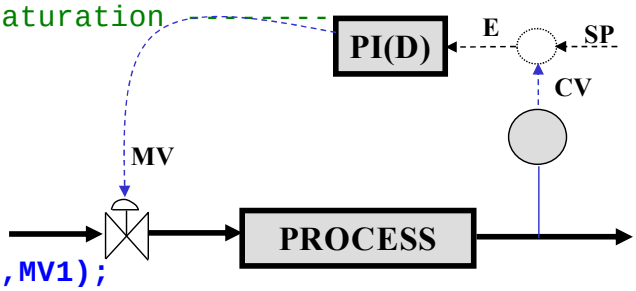
$$mv_{k-1} = mv_s + \Delta mv_{k-1}$$
$$mv_k = mv_s + \Delta mv_k$$

(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
        %% ---------- Safety saturation ----------
        if MV1 < MV_min
            MV1 = MV_min;
        end
        if MV1 > MV_max
            MV1 = MV_max;
        end
        aoCtrl.Write(aoChannel,MV1);

        %% ================================================
        % ONE-STEP prediction for NEXT sample
        % Use ACTUAL Avg_PV(k), never previous PV_model.
        %% ================================================
        delayedIndex = i-Gamma;
        if delayedIndex < 1
            MV_delay = MV0;
        else
            MV_delay = MV_history(delayedIndex);
        end
        PV_pred_next = a_model*Avg_PV + b_model*MV_delay + ...
            d_model;
        predictionValid = 1;
```



$$y_k = e^{-\Delta t/\tau} y_{k-1} + K_p(1-e^{-\Delta t/\tau})u_{k-\Gamma-1} + c(1-e^{-\Delta t/\tau})$$

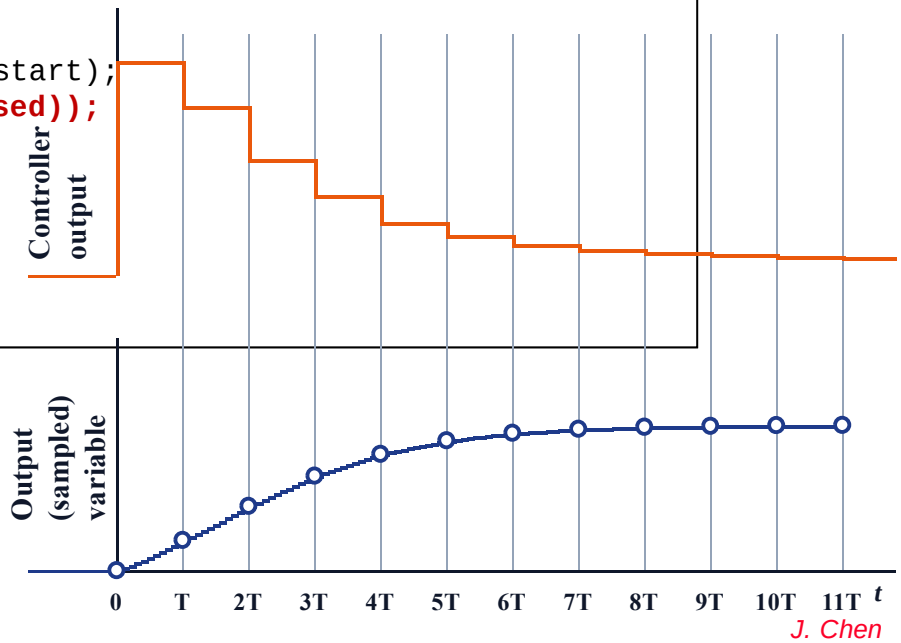
(C) 2026 CPSE Lab.

COMPUTER CONTROL: MATLAB & SIMULINK CODES

```
        %% ---------- Update controller histories ----------
        MV1_old = MV1;
        d_MV_old = d_MV;
    %% ===== Finish =====
    if fbcStarted == 1 && fbcIndex >= FBC_N
        break;
    end

    elapsed = toc(loop_start);
    pause(max(0,Ts-elapsed));

end
```



(C) 2026 CPSE Lab.