MPC Workshop

- There exists a MPC Toolbox available through Mathworks
  - I do not use this!
- My students write all of their own MPC code

- This workshop is based on MATLAB code that I have written
  - It is definitely NOT commercially quality code
- Feel free to modify my code to fit your needes

PASI 2011 | Process Modeling and Optimization for Energy and Sustainability

July 19-29, 2011, Angra dos Reis, RJ, Brazil

B. Wayne Bequette

MPC Workshop

- Primary MPC files
  - QSSmpcNLPlant.m (QP, State Space MPC)
    - isim = 1 (DMC), isim = 2 (KF-based MPC)
    - iqp = 1 (unconstrained), iqp = 2 (QP solution)
  - KmatQSS.m (generates several matrices)
  - qSSmpccalc.m (calculates control moves)
  - planteqns.m (set of plant odes)
- Driver files
  - r_FOmpc.m (first-order example)
    - FOodedev.m (planteqns = ‘FOodedev’)
  - r_VDV.m (linear van de vuuse example)

Driving Programs: r_FOmpc.m

$$\dot{x} = -\frac{1}{2}x + \frac{1}{2}u + \frac{1}{2}d$$

$$y = x$$

```
% r_FOmpc.m
% 20 July 2011 - B.W. Bequette
%    First-order problem
% Illustrates the use of State Space MPC
% The plant equations are continuous ODES provided in FOodedev.m
% The plant equations are in deviation variable form
% The controller model is discretized
%
% ---------- model parameters ---------------------------------
%       (continuous time, first-order problem)
  am = [-1/2]         % single state (time constant = 2)
  bm = [1/2 1/2]      % first column manip, 2nd column disturbance
  cm = [1]
  dm = [0 0]
%
  ninputs = size(bm,2);     % includes disturbance input
  noutputs = size(cm,1);
  nstates = size(am,1);     % number of model states
  sysc_mod = ss(am,bm,cm,dm);
%
```

$$x_{k+1} = \Phi x_k + \Gamma u_k + \Gamma^d d_k$$

$$y_k = C x_k$$

```
% discretize the model
  delt = 0.2;               % sample time = 0.2
  sysd_mod = c2d(sysc_mod,delt);
  [phi_mod,gamma_stuff,cd_mod,dd_stuff] = ssdata(sysd_mod)
%
  gamma_mod  = gamma_stuff(:,1);   % first input is manipulated
  gammad_mod = gamma_stuff(:,2);   % second input is a disturbance

% ------------ plant parameters -----------------------------
  planteqns = 'FOodedev'
  cplant = [1];   % the only state is also the output
  nstates_p = 1; % one plant state
  parvec(1) = 2; % the parameter is a time constant with value = 2
```

planteqns = 'FOodedev' → Continuous plant

Controller Parameters

```
% ---------------- controller parameters -------------------
  p = 10;        % prediction horizon
  m = 1;         % control horizon
  ny = 1;        % number of measured outputs
  nu = 1;        % number of manipulated inputs
  nd = 1;        % number of actual disturbances
  nd_est = 1;    % number of estimated disturbances (used by KF)
  weightu = [0]; % weighting matrix for control action
  weighty = [1]; % weighting matrix for outputs
% -----------------constraints -----------------------------
  umin = [-1000];
  umax = [1000];
  dumin = [-1000];
  dumax = [1000];
% ------ Kalman Filter matrices ----------------------------
  Q  = 100*eye(nd_est,nd_est);  % state covariance
  R  = eye(ny);             % measurement noise covariance
%
```

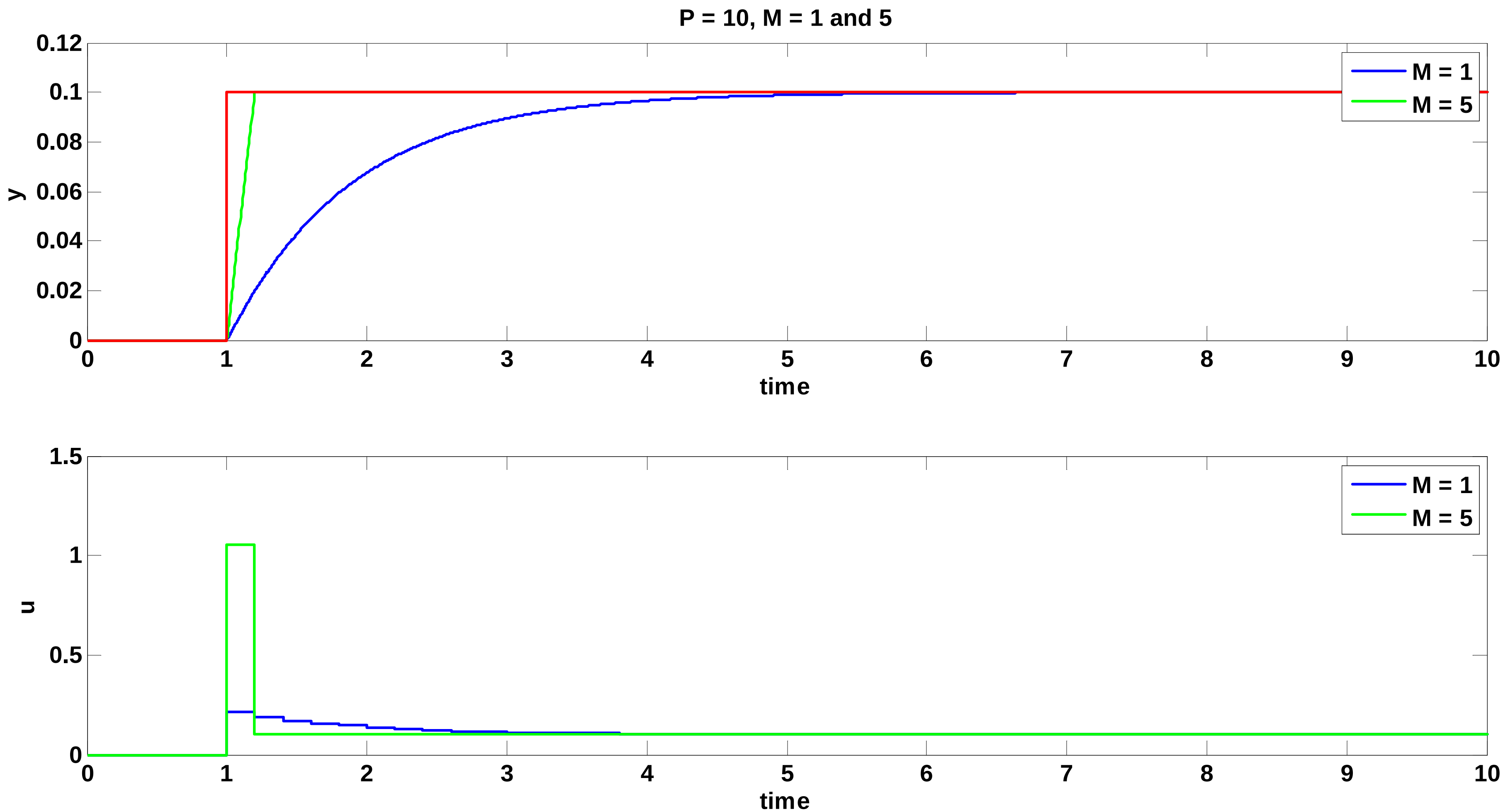
Simulation Parameters

```
% -------- simulation parameters ----------------------------
% first, setpoint change only (no disturbance)
  ysp = [0.1];    % setpoint change (from 0 vector); dimension ny
  timesp = 1;   % time of setpoint change
  dist = [0];   % magnitude of input disturbance; dimension nd
  timedis = 6;  % time of input disturbance
  tfinal = 10;
%
  isim = 1; % additive output disturbance
  iqp = 1;  % unconstrained solution
% noisemag = zeros(ny,1); % no noise
  noisemag = 0.0;      % no measurement noise
```

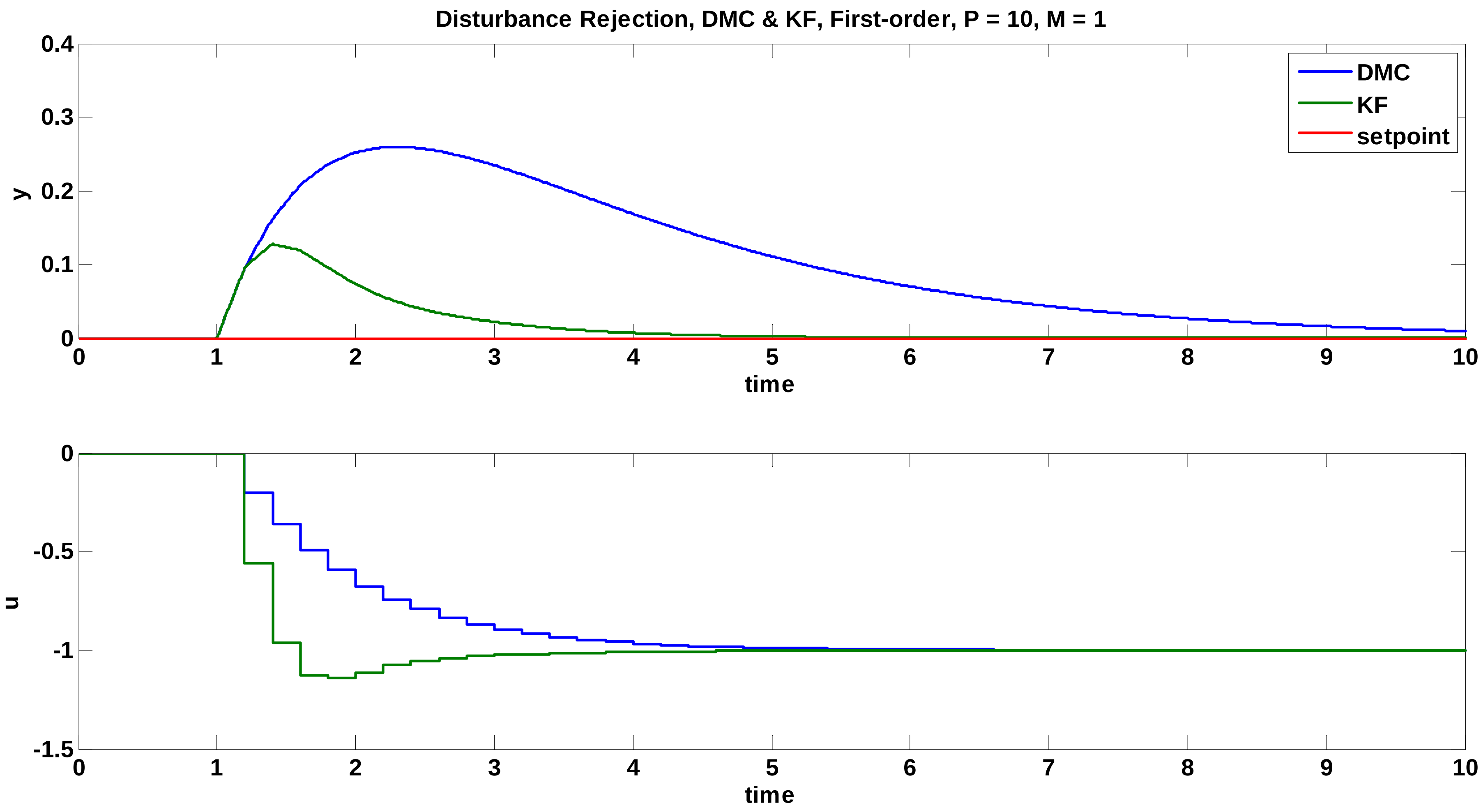
Plant Equations file

```
function xdot = FOodedev(t,x,flag,parvec,u,d)
%
% b.w. bequette - 20 July 2011
% First-order problem
% revised for explicit disturbance and deviation variable form
%
 time_constant = parvec(1);
%
 dxdt(1) = -(1/time_constant)*x(1) + u(1)/time_constant + d(1)/time_constant;
 xdot = dxdt(1);
```

Simulation Results, P = 10
Comparison of M = 1 and M = 5



Disturbance Results, P = 10, M = 1
Comparison of KF and DMC

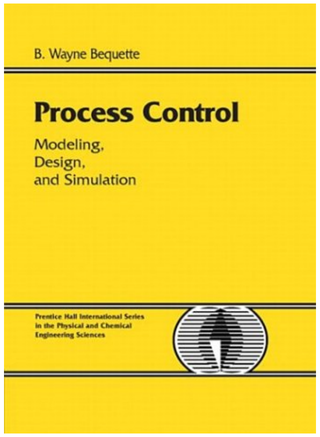


Linear Van de Vuuse Reactor (inverse response)

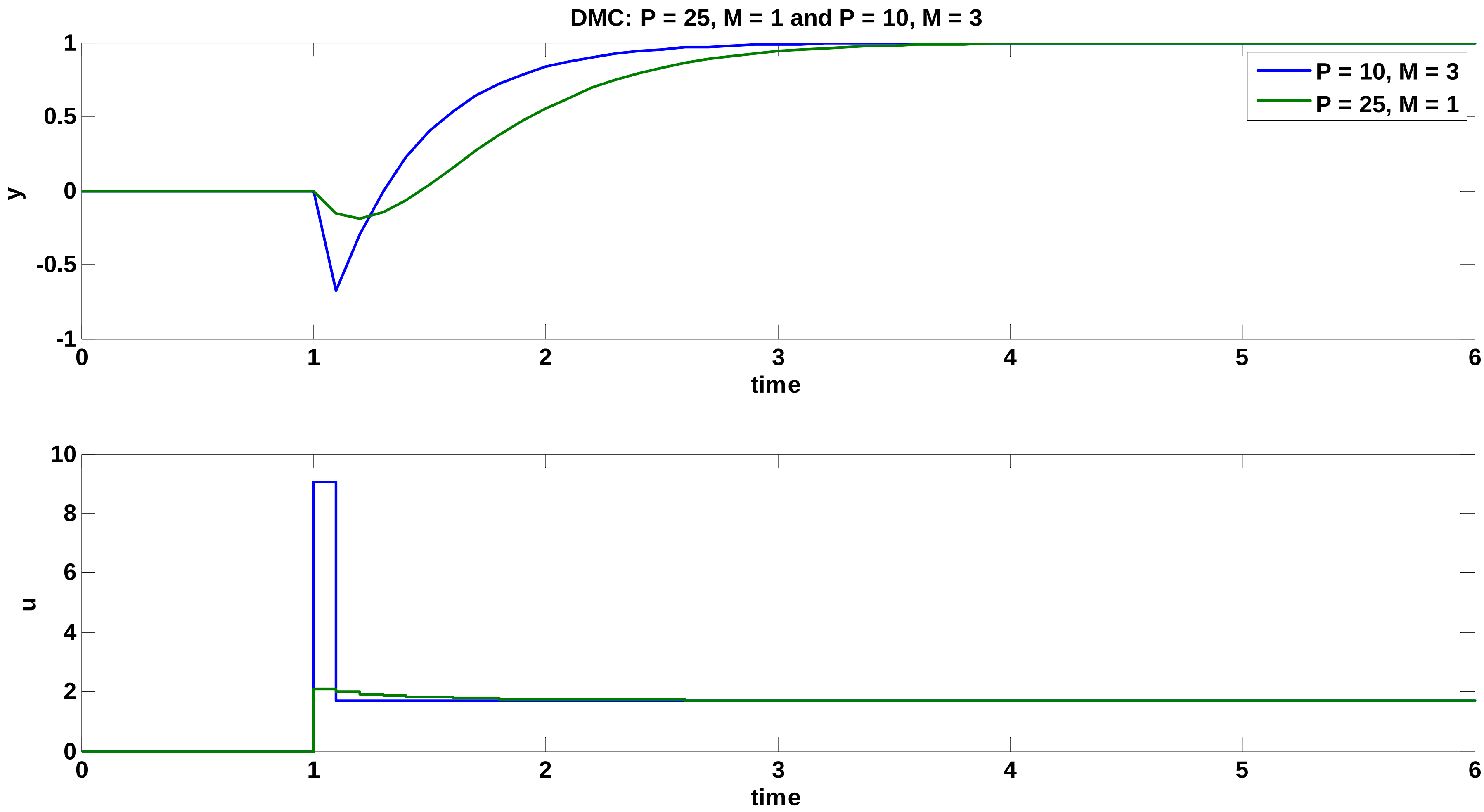
$$\dot{x} = \begin{bmatrix} -2.4048 & 0 \\ 0.833 & -2.2381 \end{bmatrix} x + \begin{bmatrix} 7 \\ -1.117 \end{bmatrix} u + \begin{bmatrix} 7 \\ -1.117 \end{bmatrix} d$$

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} x$$

Module 5



r_VDV
Comparison of (P=10,M=3) with (P=25,M=1)



r_VDV
Comparison of (P=8,M=1) with (P=7,M=1)