Application
Adaptive ANN Inverse Control
Control of DC motor with nonlinear pump load

Prof. Karim Hassan

Application
Adaptive ANN Inverse Control
Control of DC motor with nonlinear pump load

- ***Model of DC Motor with nonlinear load***

$$L\frac{di(t)}{dt} = v(t) - R\,i(t) - K_m\omega(t)$$

$$J\frac{d\omega(t)}{dt} = K_m i(t) - B\,\omega(t) - \mu\omega^2(t) * sign(\omega(t))$$

- x_1 Motor armature current $i(t)$ in (Ampere)
- x_2 Motor rotor speed $\omega(t)$ in (rad/sec).
- u Motor armature input voltage $v(t)$ in (Volt).

Application
Adaptive ANN Inverse Control
Control of DC motor with nonlinear pump load

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{R}{L} & -\frac{K_m}{L}\\ \frac{K_m}{J} & \frac{-B}{J}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}u + \begin{bmatrix}0\\ -\frac{\mu x_2^2 * \mathrm{sign}(x_2)}{J}\end{bmatrix}$$

$$y = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

$$\boxed{\dot{X} = AX + BU + F(X)}$$

- x_1 Motor armature current $i(t)$ in (Ampere)
- x_2 Motor rotor speed $\omega(t)$ in (rad/sec).
- u Motor armature input voltage $v(t)$ in (Volt).

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-\frac{R}{L} & -\frac{K_m}{L}\\ \frac{K_m}{J} & \frac{-B}{J}\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}\frac{1}{L}\\ 0\end{bmatrix}u + \begin{bmatrix}0\\ -\frac{\mu x_2^2 * \text{sign}(x_2)}{J}\end{bmatrix}$$

$$y = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}$$

Symbol		Value	Unit
R	Motor armature resistance	7.56	Ω
L	Motor armature inductance	0.055	H
K_m	Motor torque constant	3.475	N. m/(A)
J	Motor inertia	0.068	$Kg.m^2$
B	Motor friction Constant	0.03475	(N. m) /(rad/sec)
μ	Motor fan load torque constant	0.0039	(N. m) /$(rad/sec)^2$

After discretization at sampling instants $k,\ k+1,\ k+2, \dots \dots$

The forward Model

$$y(k+1) = f_1(y(k), y(k-1), u(k))$$

The inverse Model

$$u(k) = f_2(y(k+1), y(k), y(k-1))$$

We use ANN to obtain an approximation of f_2 as $\hat{f}_2$

In the identification (Training) phase:

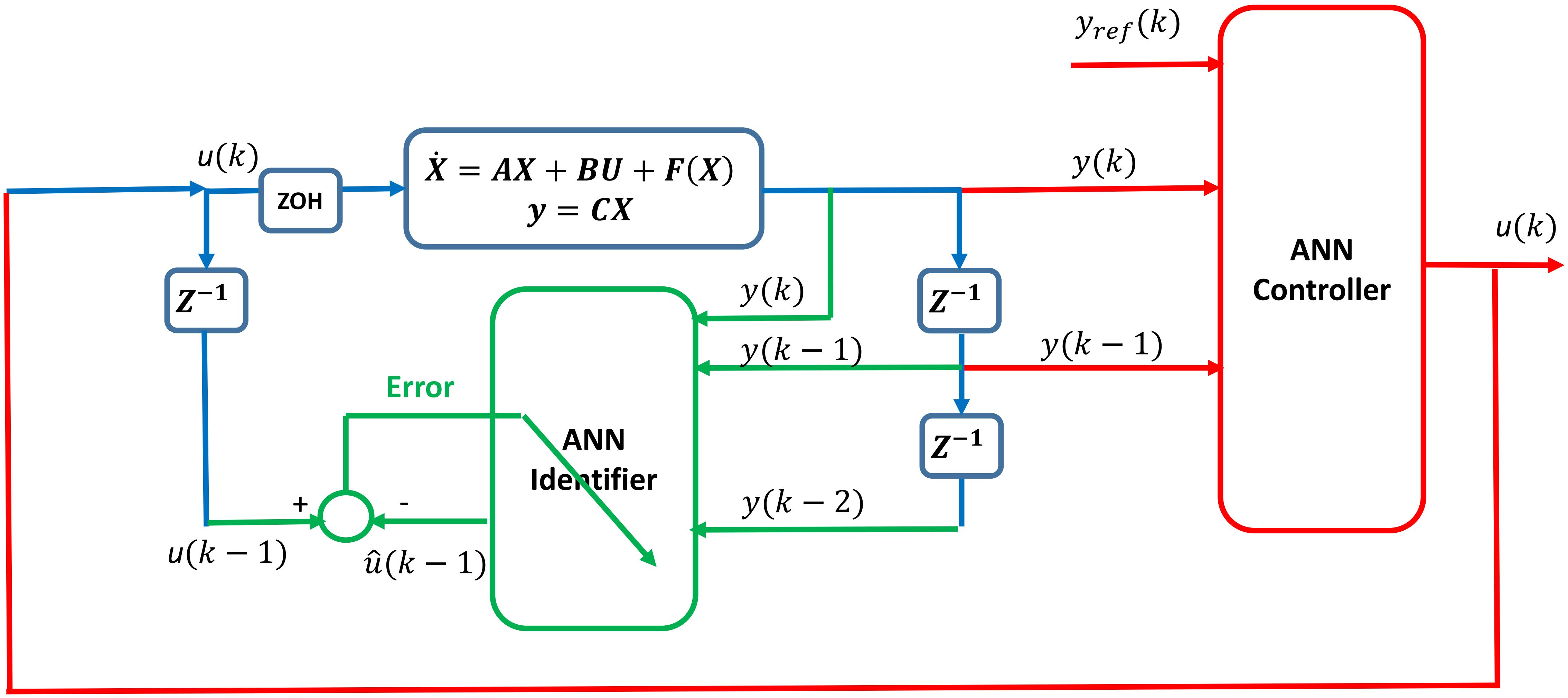
Because we don't know $y(k+1)$ in advance, we take the previous sample

$$\boxed{u(k-1) = \hat{f}_2(y(k), y(k-1), y(k-2))}$$

In the control phase

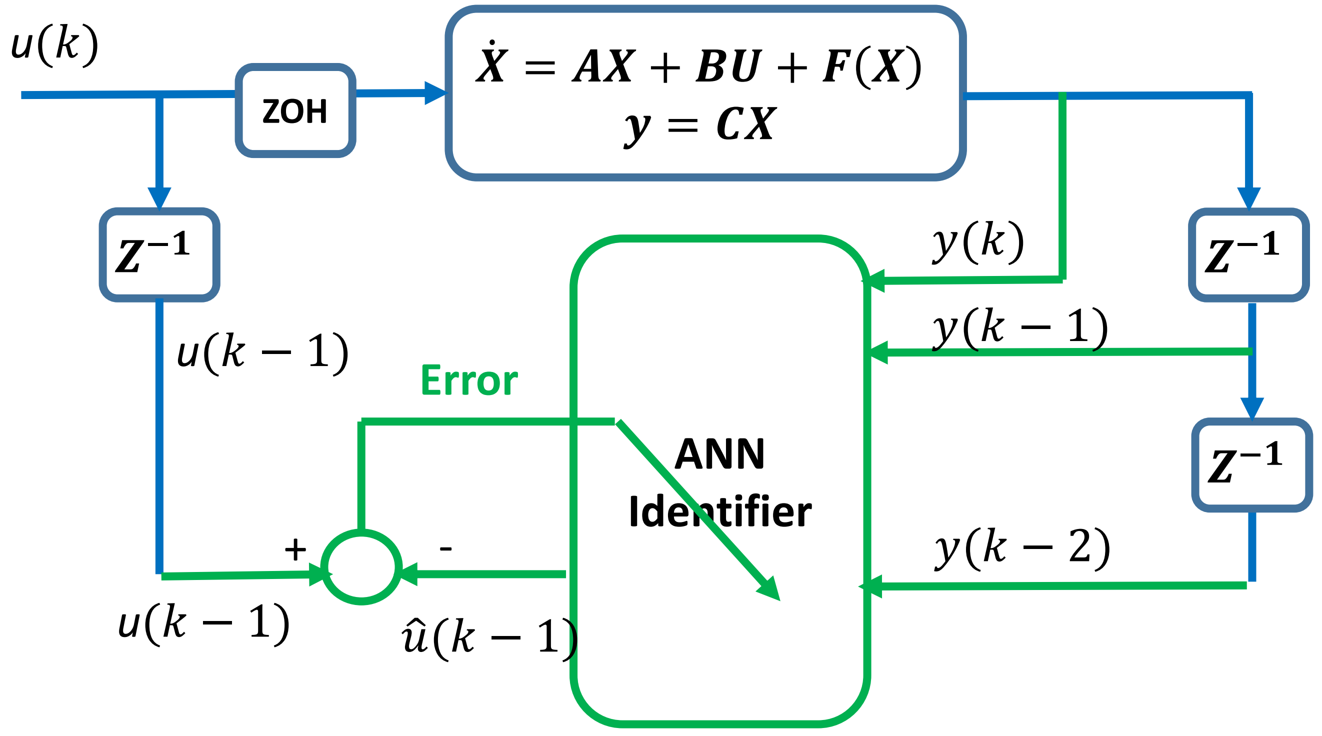
$$\boxed{u(k) = \hat{f}_2(y_{ref}(k), y(k), y(k-1))}$$

Where $y(k+1)$ has been replaced by $y_{ref}(k)$ so that in the next sample $y(k) = y_{ref}(k)$

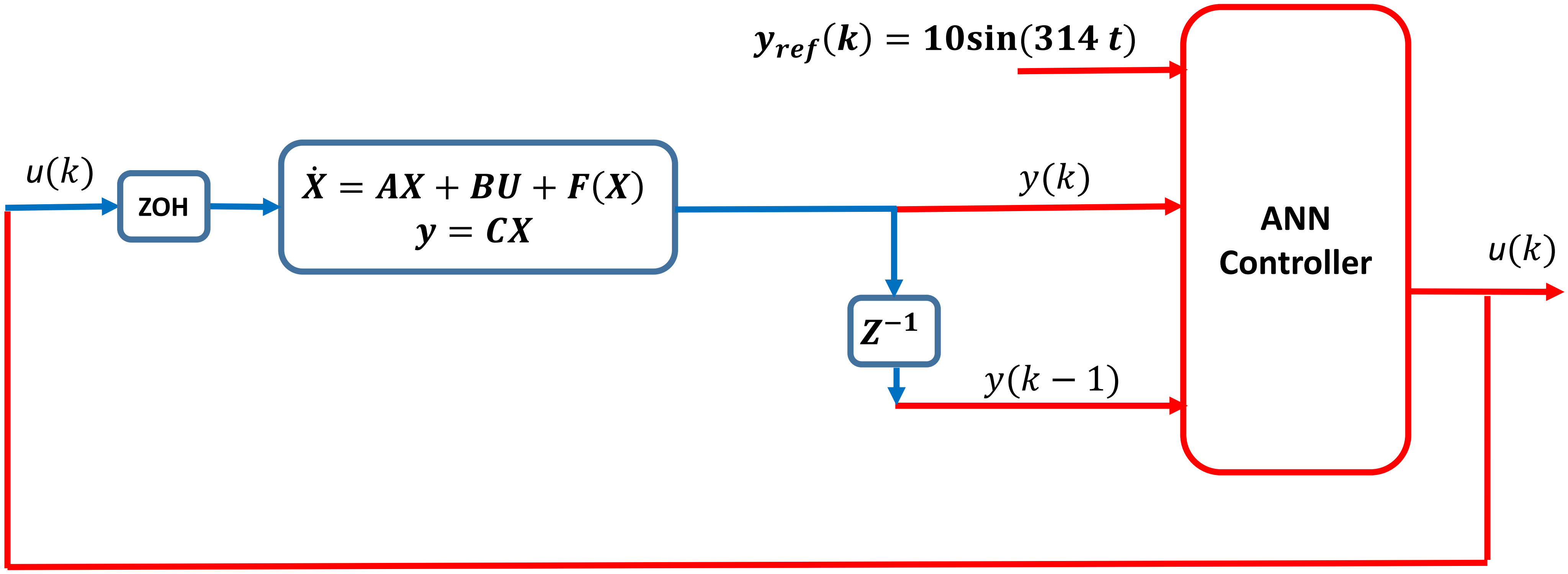


$y_{ref}(k)$
ANN Controller
$u(k)$
ZOH
$\dot{X} = AX + BU + F(X)$
$y = CX$
$y(k)$
Z^{-1}
Z^{-1}
Z^{-1}
$y(k)$
$y(k-1)$
$y(k-1)$
$y(k-2)$
$u(k)$
Error
ANN Identifier
+
-
$u(k-1)$
$\hat{u}(k-1)$

$$u(k) = 2\sin(314\,t) + rand(0,1) - rand(0,1)$$



Identification Phase



Control Phase

Application
Adaptive ANN Inverse Control
Control of DC motor with nonlinear pump load

- **Steps:**
- Simulate the system using: Integrator, gain, constant, function and square blocks in Simulink.
- Apply sinusoidal input plus random signal.
- Record delayed samples of input and outputs **for 10 seconds.**
  (sampling period = 0.04 Seconds).
- Export input and output data to Excel.
- Build an ANN model of input, output and hidden layer. Use **sigmoidal** activation function.
- Train the ANN using **python** by **backpropagation**.
- You may use **MATLAB Neural Network Toolbox** instead.
- You may also use **Python embedded in MATLAB** (new version is needed).
- You may also use **Neural Network Toolbox in MATLAB**.
- Apply the trained ANN as a controller.
- Test the system response for **sinusoidal reference** speed