SEEE1022 INTRODUCTION TO SCIENTIFIC PROGRAMMING

CH12
Polynomial

Dr. Mohd Saiful Azimi Mahmud (azimi@utm.my)
P19a-04-03-30, School of Electrical Engineering, UTM

OBJECTIVES

After studying this chapter you should be able to:

1. Represents polynomial as vector.
2. Find polynomial roots.
3. Use MATLAB function to multiply and divide polynomials
4. Use polynomial to model a data through curve fitting function polyfit and polyval

POLYNOMIAL

WHAT IS POLYNOMIAL?

- Polynomial is a mathematical expression consisting **the sum of many terms** (*'poly'=many, 'nomial'=terms*).
- Each of the term consist a **variable** raised to a positive integer power and multiplied by a **coefficient**.

$$a_N x^N + a_{N-1} x^{N-1} + \cdots + a_2 x^2 + a_1 x^1 + a_0 x^0$$

- Based on above, there are four terms with x is the variable. From the right, each term has the variable is raised to an increasing integer value starts at 0.
- $a_N, a_{N-1}, \dots, a_2, a_1, a_0$ are the coefficients for the terms.

POLYNOMIAL

APPLICATION

- The followings are common application of polynomial in electrical engineering.
  1) System representation.
  2) System stability evaluation.
  3) Modelling acquired data.
  4) Calibration.

POLYNOMIAL

VECTOR REPRESENTATION

- Polynomial coefficients are normally constants and can be group into a single vector.
- In MATLAB, the coefficient vector can be either in the form of row vector or column vector.
- With the coefficients represented as vector, MATLAB provide many functions to work with polynomial. Functions that will be covered in this chapter are as below:

Function	Description
conv	Polynomial multiplication
deconv	Polynomial division
roots	Polynomial roots
polyval	Polynomial evaluation
polyfit	Polynomial curve fitting

POLYNOMIAL

VECTOR REPRESENTATION

EXAMPLE 1

	Polynomial	Vector Representation
1	$3x^4 + x^3 - 2x^2 + 0.5x + 1$	$[3 \quad 1 \quad -2 \quad 0.5 \quad 1]$
2	$x - 1$	$[1 \quad -1]$
3	$-6y^2 + \frac{3}{2}y$	$\left[-6 \quad \frac{3}{2} \quad 0\right]$
4	8	$[8]$
5	a^3	$[1 \quad 0 \quad 0 \quad 0]$
6	$1 - 2x - 3x^2$	$[-3 \quad -2 \quad 1]$
7	$f(y) = 2y - 1 + 3y^2 + a$	$[3 \quad 2 \quad (a-1)]$
8	$f(y) = 3xy^3 + xy + 1$	$[3x \quad 0 \quad x \quad 1]$
9	$f(x) = 3xy^3 + xy + 1$	$[(3y^3 + y) \quad 1]$
10	$f(x) = a(x + x^2) + x - 3$	$[a \quad a+1 \quad -3]$

conv AND deconv

- **Syntax**

```
w = conv(u,v)                %Polynomial multiplication

[w,r] = deconv(u,v)          %Polynomial division
```

Description:

- `w = conv(u,v)` returns the polynomial multiplication of vector `u` and `v`.
- `w = deconv(u,v)` returns vector `w` as the polynomial division of vector `u` over vector `v` and `r` as the remainder of the polynomial long division such that `u = conv(v,w)+r`.

POLYNOMIAL

conv AND deconv

EXAMPLE 2

Find $f(x) = (x^3 + 3x^2 + 2)(x^2 - 5)$ and $g(x) = (x^3 + 3x^2 + 2)/(x^2 - 5)$

```
>> A = [1 3 0 2];
>> B = [1 0 -5];
>> f = conv(A,B)
f1 =
     1     3    -5   -13     0   -10

>> [g,r] = deconv(A,B)
g =
     1     3
r =
     0     0     5    17
```

Thus,

$$f(x) = x^5 + 3x^4 - 5x^3 - 13x^2 - 10$$

$$g(x) = x + 3 + \frac{5x + 17}{x^2 - 5}$$

POLYNOMIAL

SYSTEM REPRESENTATION

- Many engineering related system can be represented with mathematical model.
- Polynomial is one of the mathematical expression commonly used to represent the system, especially for electrical and electronic system.
- One case is when a system is represented in Laplace domain where s is use as the polynomial variable. This polynomial representation of the system is called 'System Function'.
- Other than that, polynomial can also be used to represent signal or data.

POLYNOMIAL

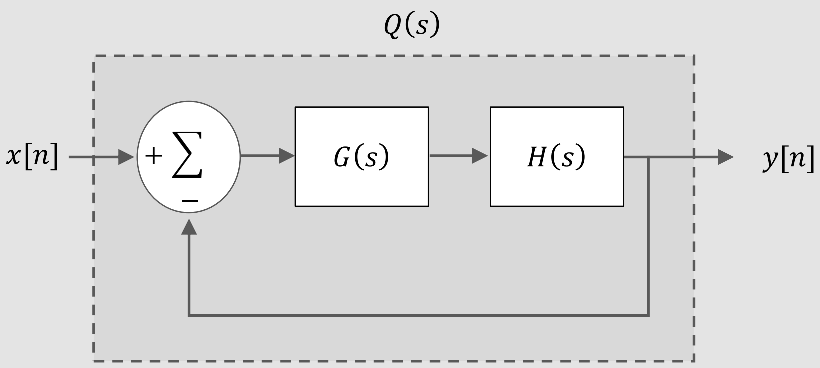
SYSTEM FUNCTION

EXAMPLE 3

Figure below shows a system with PID controller $G(s)$. This controller is used to stabilize the plant $H(s)$ (e.g.: motor, car acceleration pedal, compressor). The overall system function of the system $Q(s)$ is shown by the polynomial equation at the right side of the figure.

Q_n is denoted as *numerator* polynomial and Q_d is the *denumerator* polynomial.

D, P and I are the controller constant.



$$G(s) = \frac{G_n}{G_d} = \frac{Ds^2 + Ps + I}{s}$$

$$H(s) = \frac{H_n}{H_d}$$

$$Q(s) = \frac{Q_n}{Q_d} = \frac{G_n H_n}{G_D H_d + G_n H_n}$$

SYSTEM FUNCTION

EXAMPLE 3

- Now let say we want to control a motor at a specific speed where the motor can be represented using polynomial $H(s) = \frac{1}{s-2}$. Below is the MATLAB code to generate the two polynomial Q_n and Q_d.
- In this example we set $P = 4$, $D = 2$ and $I = 1$.

```
P=4; D=2; I=1;

Gn = [D P I];    Gd = [1 0];
Hn = 1;          Hd = [1 -2];

Qn = conv(Gn,Hn)
Qd = addpoly(conv(Gd,Hd),Qn)
```

```
Qn =
     2     4     1

Qd =
     3     2     1
```

`addpoly` is a user defined function to add two polynomial.

$$G(s) = \frac{G_n}{G_d} = \frac{2s^2 + 4s + 1}{s}$$

$$H(s) = \frac{H_n}{H_d} = \frac{1}{s - 2}$$

$$Q(s) = \frac{Q_n}{Q_d} = \frac{2s^2 + 4s + 1}{3s^2 + 2s + 1}$$

POLYNOMIAL

polyval

- Syntax

```
y = polyval(p,x)          %Polynomial evaluation
```

Description

p – Polynomial coefficients, specified as a vector

x – evaluated points

y – evaluated values

EXAMPLE 4

Find $f(x) = x^3 + 4x^2 - 2x + 1$ for $x = 3.5$

```
>> p = [1 4 -2 1];
>> f = polyval(p,3.5)

f =
   85.8750
```

POLYNOMIAL

FREQUENCY RESPONSE

EXAMPLE 5

- Back to Example 3 where numerator and denumerator polynomial is use to represent a system where $Q(s) = \frac{Q_n}{Q_d}$. To understand the system, the polynomial is evaluated over a range of s values.
- In this case, $s = j2\pi F$ where F is frequency in Hz. Thus, the values for F must be first determine to have the s values. Here we set F from $0Hz$ to $10kHz$.
- Lastly, a graph is plotted to visualize the behaviour of the system.

```
Qn = [2 4 1]; Qd = [3 2 1];
F = 0:10e3;
s = 1i*2*pi*F;
Qns = polyval(Qn,s);
Qds = polyval(Qd,s);
Qs = Qns./Qds;

figure, semilogx(F,abs(Qs))
xlabel('Frequency (Hz)')
ylabel('Magnitude |H(s)|')
title('Frequency Response')
```

- H_n and H_d were created from Example 3.
- semilogx plot is used to plot the data because the range of the F is very wide.

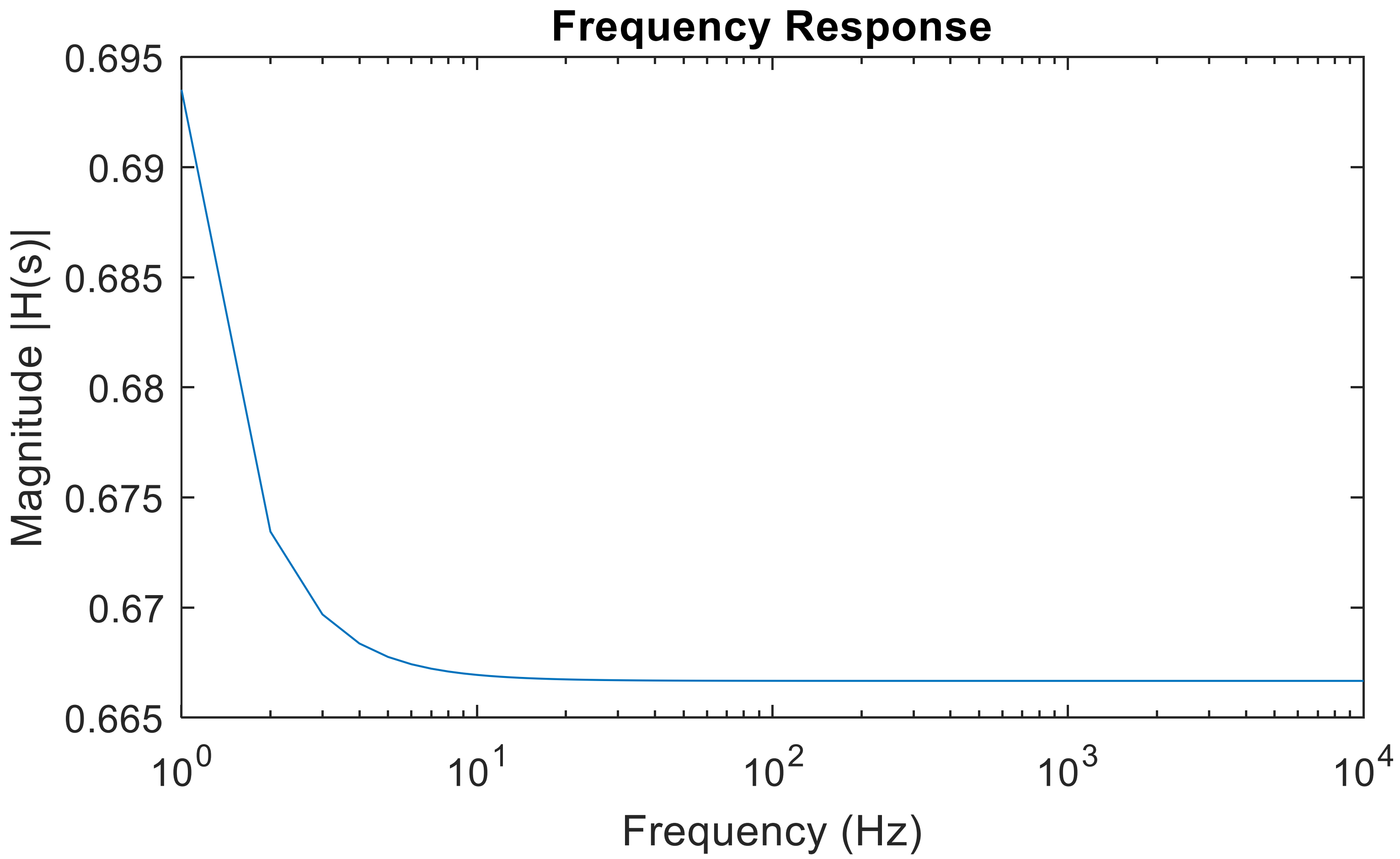
$$G(s) = \frac{G_n}{G_d} = \frac{2s^2 + 4s + 1}{s}$$

$$H(s) = \frac{H_n}{H_d} = \frac{1}{s + 1}$$

$$Q(s) = \frac{Q_n}{Q_d} = \frac{2s^2 + 4s + 1}{3s^2 + 2s + 1}$$

FREQUENCY RESPONSE

EXAMPLE 5



POLYNOMIAL

roots

- Roots of polynomial $P(x)$ is all x values that return $P(x)$=0.
- Polynomial with its highest power equals to N will have N roots.
- **Syntax**

```
y = roots(p)        %p is polynomial specified in vector
```

EXAMPLE 6

Find roots for polynomial $f(x) = x^3 + x^2 - 4x - 4$

```
>> p = [1 1 -4 -4];
>> r = roots(p)
r =
    2.0000
   -2.0000
   -1.0000
```

VERTICAL DISPLACEMENT

EXAMPLE 7

- Launch an object from ground vertically and find out how long will it take to hit back the ground. The formula for vertical displacement is as below where v is the initial velocity and g is the acceleration due to gravity. Lets set the $v = 30ms^{-1}$ and $g = -9.8ms^{-2}$.

$$d = vt + \frac{g}{2}t^2$$

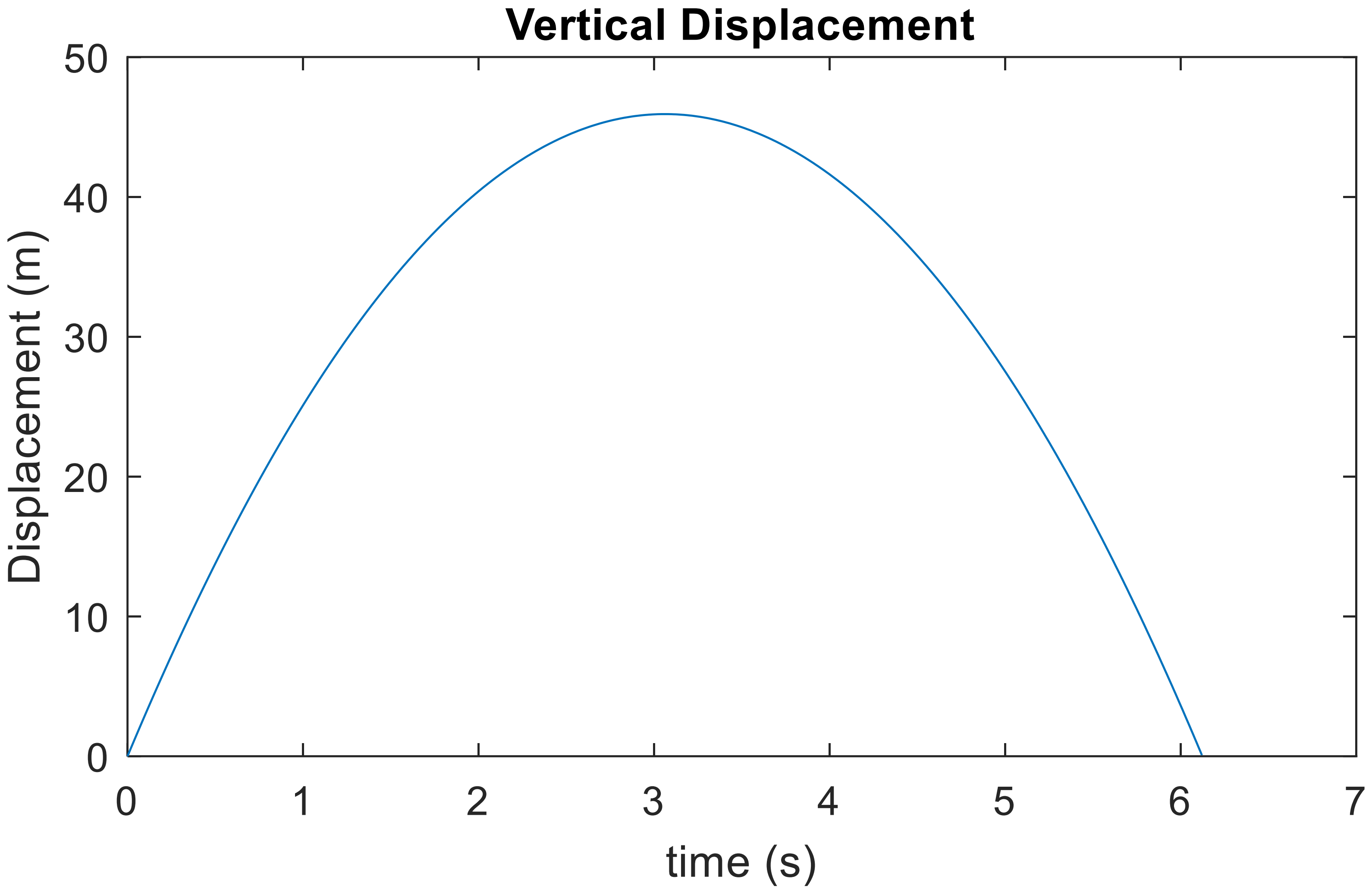
- This problem can be solve by finding roots of polynomial since ground is where $d = 0$.

```
p = [-9.8/2 30 0];
r = roots(p)
tground = r(r~=0);
t = 0:0.01:tground;
d = polyval(p,t);
plot(t,d)
xlabel('time (s)'), ylabel('Displacement (m)')
title('Vertical Displacement')
```

```
r =
          0
     6.1224
```

POLYNOMIAL

VERTICAL DISPLACEMENT



SYSTEM STABILITY

EXAMPLE 8

- Back to Example 3, stability of the system represented by polynomial can be checked by the roots value of the denumerator polynomial Q_d. A system is stable if all of the denumerator roots are less than 0.

- Now, create a MATLAB function that check the stability of the PID control system described in Example 3. The input variables to the function are as below while the output variable is none.

  1) Constant P, I and D specified as vector $C = [P \quad I \quad D]$.
  2) Numerator polynomial of the plant, specified in vector Pn.
  3) Denumerator polynomial of the plant, specified in vector Pd.

  Apart from the above, the function should display at command window the roots and string message on whether the system is stable or not stable.

POLYNOMIAL

SYSTEM STABILITY

EXAMPLE 8

Below is the MATLAB function for Example 8.

```
function chkStability(C,Pn,Pd)
Gn = [C(3) C(1) C(2)];
Gd = [1 0];
Hn = conv(Gn,Pn);
Hd = addpoly(conv(Gd,Pd),Hn);

Root = roots(Hd);
length(Root);
fprintf('root%d = %.2f%+.2fi\n',...
        [1:length(Root);real(Root)';imag(Root)'])

R = Root(Root>=0);
if isempty(R)
    disp('The system is stable')
else
    disp('The system is not stable')
end
```

POLYNOMIAL

SYSTEM STABILITY

EXAMPLE 8

Below is the MATLAB code on using the `checkStability` function for several values of P, I and D and different plant $P(s)$.

```
>> checkStability([4,1,0],1,[1 -2])
root1 = -1.00+0.00i
root2 = -1.00+0.00i
The system is stable

>> checkStability([1,1,2],1,[1 -2])
root1 = 0.17+0.55i
root2 = 0.17-0.55i
The system is not stable

>> checkStability([3,3,3],[1 0],[1 1])
root1 = 0.00+0.00i
root2 = -0.67+0.94i
root3 = -0.67-0.94i
The system is stable
```

POLYNOMIAL

polyfit

- **Syntax**

```
p = polyfit(x,y,n)        %Polynomial curve fitting
```

Description:

x – x-axis data, specified as a vector

y – y-axis data, specified as a vector

n – Degree of the polynomial fit

p – Least-squares fit polynomial coefficients, returned as a vector

CURVE FITTING

- Curve fitting is the process of constructing a curve to a series of collected data points.
- The curve is represented by mathematical function such as polynomial.
- The idea of having the curve represented by mathematical function is a way to model a process or activity that varies over some variables.
- **Example:**
    - Modelling yearly climate data
    - Modelling internet usage
- **Why modelling?**
    - Automation
    - Analysing relationship between variables
    - Data Trend

SENSOR CALIBRATION

EXAMPLE 9

Spreadsheet ‘chp12ex12_3.xlsx’ contain a temperature sensor reading in volt (V) taken over temperature between 20℃ to 50℃. Find a mathematical function that will convert the sensor reading from volt (V) to temperature (℃).

Solution

- Finding mathematical function to a sensor reading is normally done to calibrate the sensor. Sensor calibration is important to ensure accurate reading.
- In this example, the calibration can be done by applying polynomial curve fitting to the collected data.
- In Matlab, function `polyfit()` can be used to obtain the mathematical function. Then use function `polyval()` to test the output.

POLYNOMIAL

SENSOR CALIBRATION

EXAMPLE 9

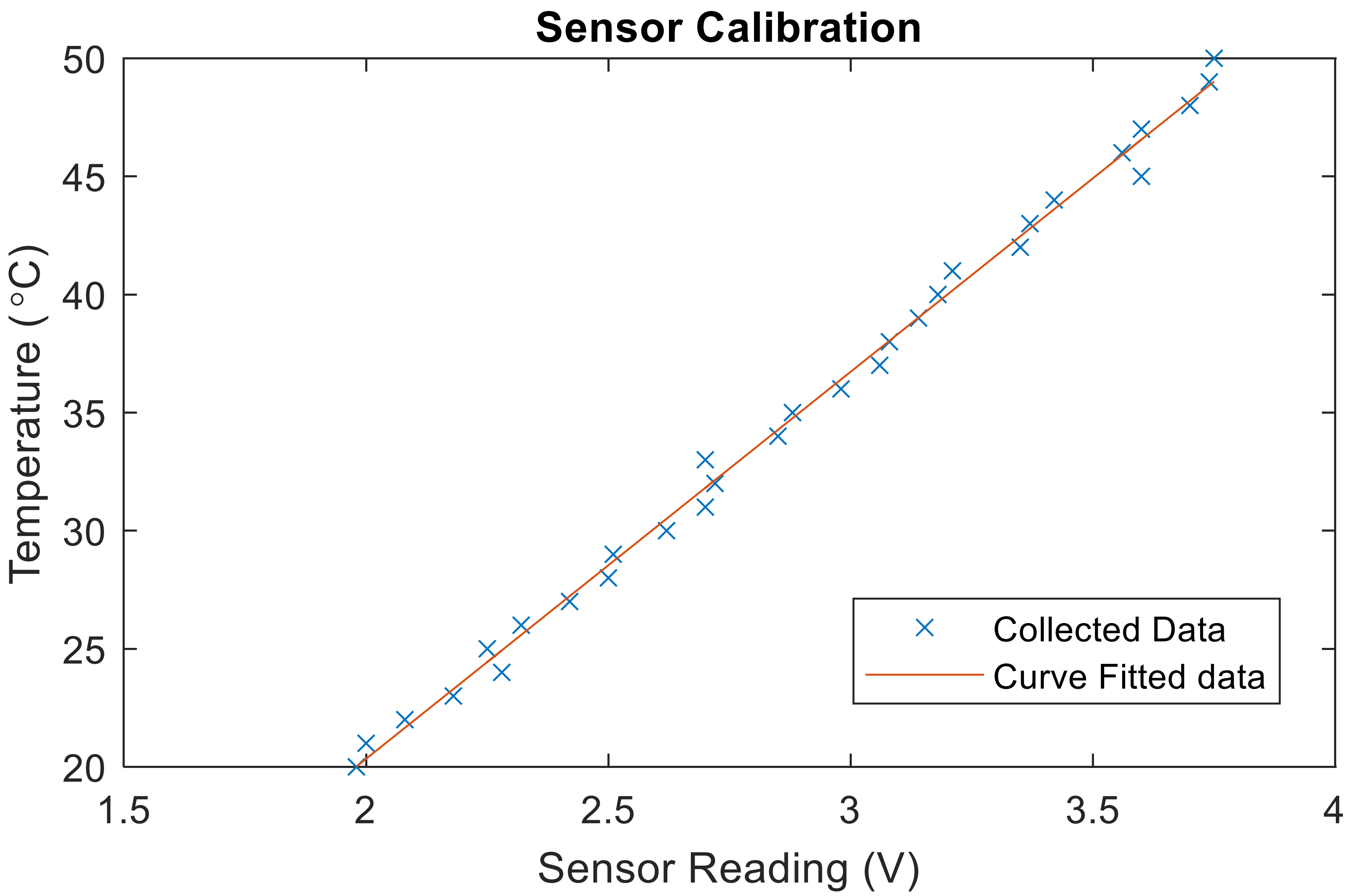
Below is the MATLAB code to obtain the mathematical function using polynomial curve fitting function. Based on the collected data, the polynomial order is chosen equals to 1.

```
x = xlsread('calibration.xlsx','B3:C33');
V = x(:,1);
T = x(:,2);

p = polyfit(V,T,1);
fprintf('T(V) = %.2fV%+.2f\n',p)

sT = polyval(p,V);

figure, plot(V,T,' x',V,sT)
title('Sensor Calibration')
xlabel('Sensor Reading (V)')
ylabel('Temperature (\circC)')
legend('Collected Data','Curve Fitted data')
```

```
T(V) = 16.38V-12.42
```

POLYNOMIAL

SENSOR CALIBRATION



POLYNOMIAL

DATA TREND

EXAMPLE 10

Below is a table of six months internet usage for five persons of age above 30 that previously never use an internet. Find the usage trend and estimate the internet usage at month 7, 8 and 9.

This data is available in file internetusage.xlsx.

Person	Internet Usage (hour)					
	1^{st} month	2^{nd} month	3^{rd} month	4^{th} month	5^{th} month	6^{th} month
1	7	11	23	59	120	180
2	3	8	20	48	88	140
3	4	11	21	58	128	195
4	5	9	16	45	111	156
5	2	5	12	38	78	145

POLYNOMIAL

DATA TREND

EXAMPLE 10

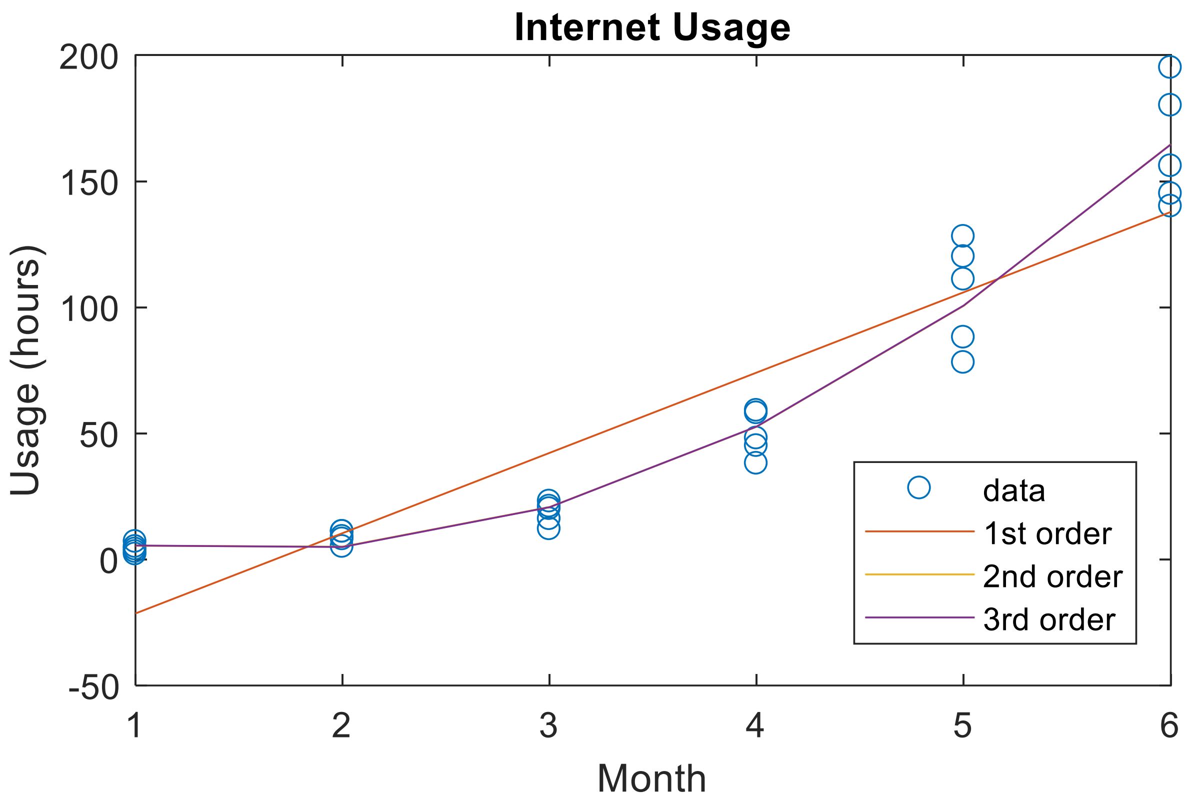
The data trend can be obtained by fitting the data using polynomial. The data can be plotted with month as the x-axis variable and usage as the y-axis variable.

```
x = xlsread('internetusage.xlsx','C5:H9');
data = [x(1,:) x(2,:) x(3,:) x(4,:) x(5,:)];
month = [1:6 1:6 1:6 1:6 1:6];

p1 = polyfit(month,data,1);
p2 = polyfit(month,data,2);
p3 = polyfit(month,data,3);

monthFit =[1:6 6*ones(1,24)];
dataFit1 = polyval(p1,monthFit);
dataFit2 = polyval(p2,monthFit);
dataFit3 = polyval(p3,monthFit);

plot(month,data,' o',monthFit,[dataFit1;dataFit2;dataFit3])
title('Internet Usage')
xlabel('Month')
ylabel('Usage (hours)')
legend('data','1st order','2nd order','3rd order')
```

POLYNOMIAL

DATA TREND

EXAMPLE 10



- From the above figure, 2nd order and 3rd order polynomial fitting are more accurate to represent the data compared to the 1st order polynomial. Thus, the 2nd order polynomial is chosen as the mathematical model of the data trend.

POLYNOMIAL

DATA TREND

- To find the internet usage for month 7, 8 and 9, `polyval()` function is used with the evaluated point set equals to a vector of [7 8 9].

```
>> usage = polyval(p2,[7 8 9]);
>> fprintf('Month %d = %.2fhours\n',[7:9;usage])
Month 7 = 244.88hours
Month 8 = 341.19hours
Month 9 = 453.61hours
```

DATA MODELLING

EXAMPLE 11

A farmer want to automate the irrigation system for his crops so that he will never have to change the amount of water setting to the irrigation system everyday. Based on the data he collected on the water usage of the previous crops cycle, find a mathematical formulation for the water usage. This formula will be used as the basis to the automatic irrigation system.

The water usage data can be found in the file waterusage.xlsx.

Solution

In this example we will do the curve fitting using several degree of polynomials and pick the best for the water usage data based on a measure called R-squared shown by below equation. d_i is the measured data and f_i is the fitted data.

$$R^2 = 1 - \frac{\sum_i (d_i - f_i)^2}{\sum_i d_i^2}$$

POLYNOMIAL

DATA MODELLING

EXAMPLE 11

Matlab code for Example 11

```
x = xlsread('waterusage.xlsx','B3:C47');
day = x(:,1);  water = x(:,2);
N=5;
p = zeros(N,N+1);
waterFit = zeros(N,45);
label = {'Measured Data'};
for n = 1:N
    p(n,N+1-n:N+1) = polyfit(day,water,n);
    waterFit(n,:) = polyval(p(n,N+1-n:N+1),day);
    label{n+1} = sprintf('Polinomial Order = %d',n);
end

R2 = 1 - sum((waterFit'-water).^2)/sum(water.^2);
disp('R-SQUARED VALUES:')
fprintf('Polynomial Order %d = %.4f\n',[1:N;R2])
plot(day,water,' x',day,waterFit)
xlabel('day')
ylabel('Water Usage (Litre)')
title('Water Usage for 1 Harvesting Cycle')
legend(label)
```

POLYNOMIAL DATA MODELLING

EXAMPLE 11

- Below are the displayed information at command window

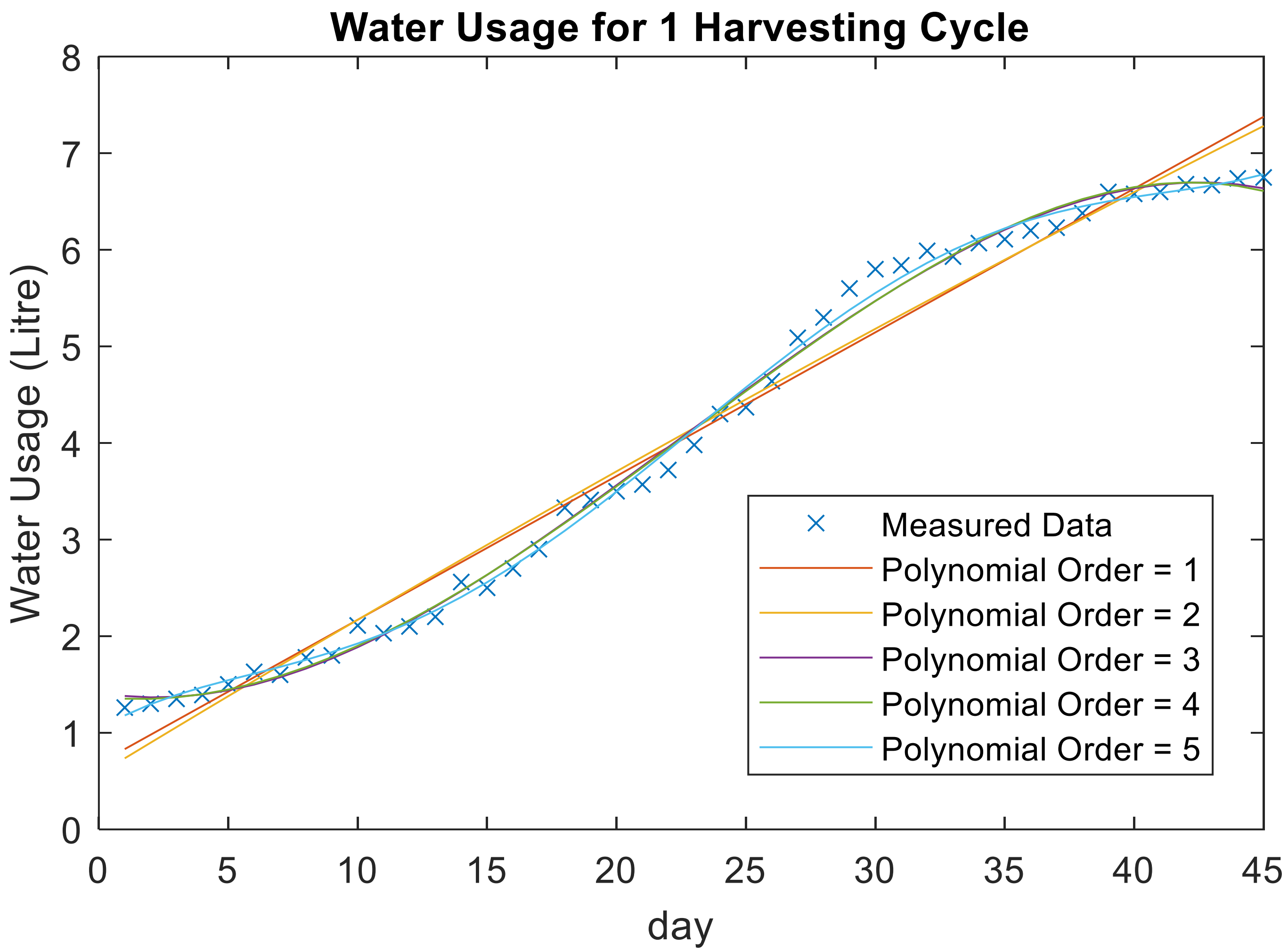
```
R-SQUARED VALUES:
Polynomial Order 1 = 0.9953
Polynomial Order 2 = 0.9954
Polynomial Order 3 = 0.9991
Polynomial Order 4 = 0.9991
Polynomial Order 5 = 0.9994
```

- Since the r-squared value for the 3rd order polynomial, which is equals to 0.9991 can be considered as good enough, thus the best polynomial order is 3 in order to keep the complexity of the polynomial low. Thus, the chosen polynomial is as below:

```
p(3,3:end)
ans =
  -0.00016249      0.010909      -0.04582        1.4151
```

$$w = -0.00016249d^3 + 0.010909d^2 - 0.04582d + 1.4151$$

DATA MODELLING



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

univteknologimalaysia

utm_my

utmofficial

Thank You

www.utm.my

innovative ● entrepreneurial ● global