Simulations in MATLAB vs. Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

Dynamic Systems (1.order)

Dynamic system represented as a differential equation (1.order system):

$$\dot{x} = -ax + bu$$

$$\frac{dx}{dt}$$

Note!

Where:

x - Process variable, e.g., Level, Pressure, Temperature, etc.

u - Input variable, e.g., Control Signal from the Controller

a, b - Constants

Discretization

Given the following differential equation:

$$\dot{x} = -ax + bu$$

In order to simulate this system in C#, we need to find the discrete differential equation.

We can use e.g., the **Euler Approximation**:

$$\dot{x} \approx \frac{x_{k+1} - x_k}{T_s}$$

T_s - Sampling Interval

Then we get:

$$\frac{x_{k+1} - x_k}{T_s} = -ax_k + bu_k$$

This gives the following discrete differential equation:

$$x_{k+1} = (1 - T_s a)x_k + T_s b u_k$$

Simulation in MATLAB

Hans-Petter Halvorsen, M.Sc.

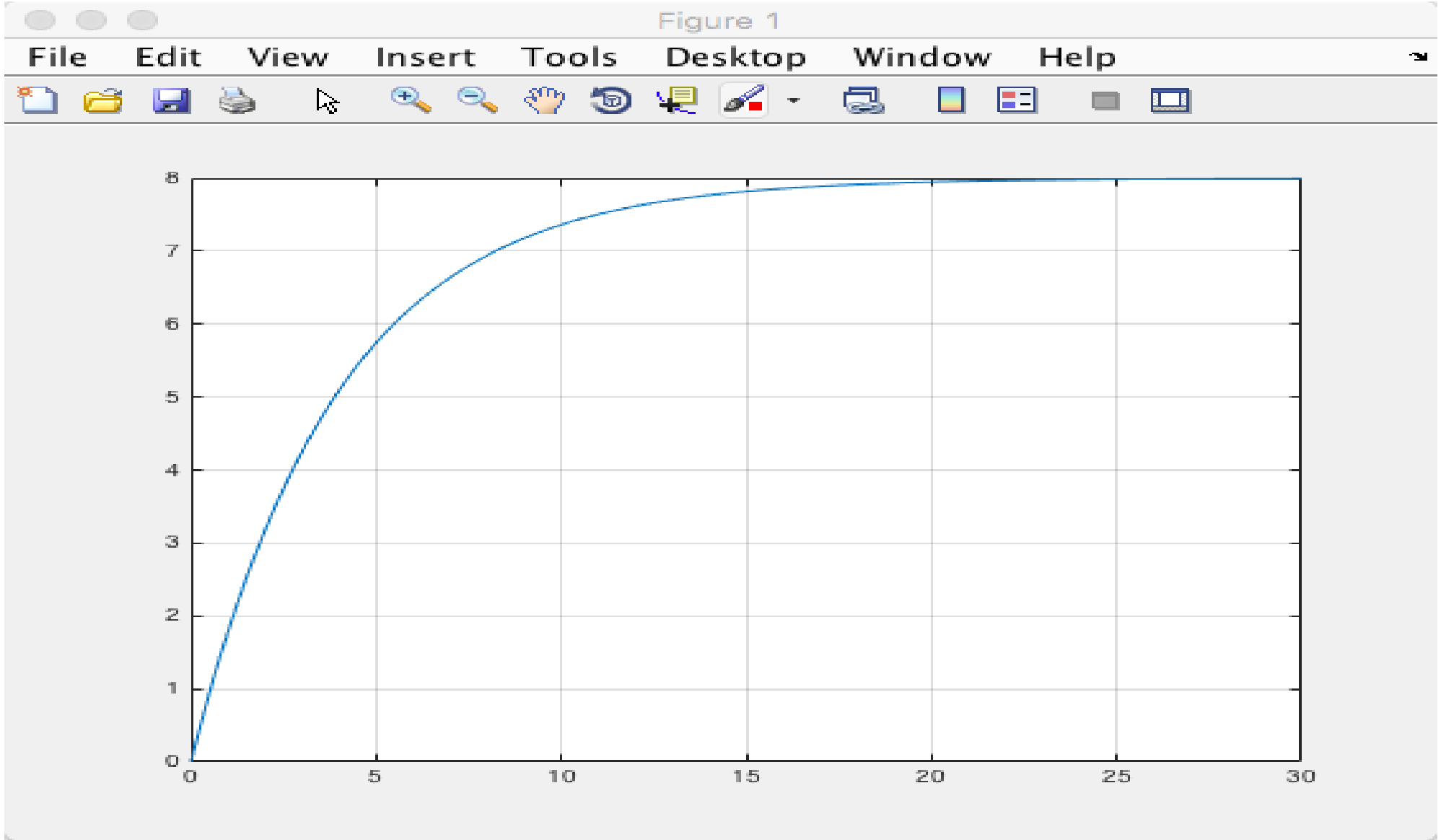
MATLAB Simulation

We set $a = 0.25\ and\ b = 2$

(You can explore with other values)

```
% Simulation of discrete model
clear, clc

% Model Parameters
a = 0.25;b = 2;

% Simulation Parameters
Ts = 0.1; %s
Tstop = 30; %s
uk = 1; % Step Response
x(1) = 0;

% Simulation
for k=1:(Tstop/Ts)
    x(k+1) = (1-a*Ts).*x(k) + Ts*b*uk;
end

% Plot the Simulation Results
k=0:Ts:Tstop;
plot(k,x)
grid on
```

Simulation Results



Simulation in Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

Plotting in Visual Studio

Visual Studio has a Chart control that you can use in Windows Forms or Web application (ASP.NET)
https://msdn.microsoft.com/en-us/library/dd489237.aspx
http://www.i-programmer.info/programming/uiux/2756-getting-started-with-net-charts.html

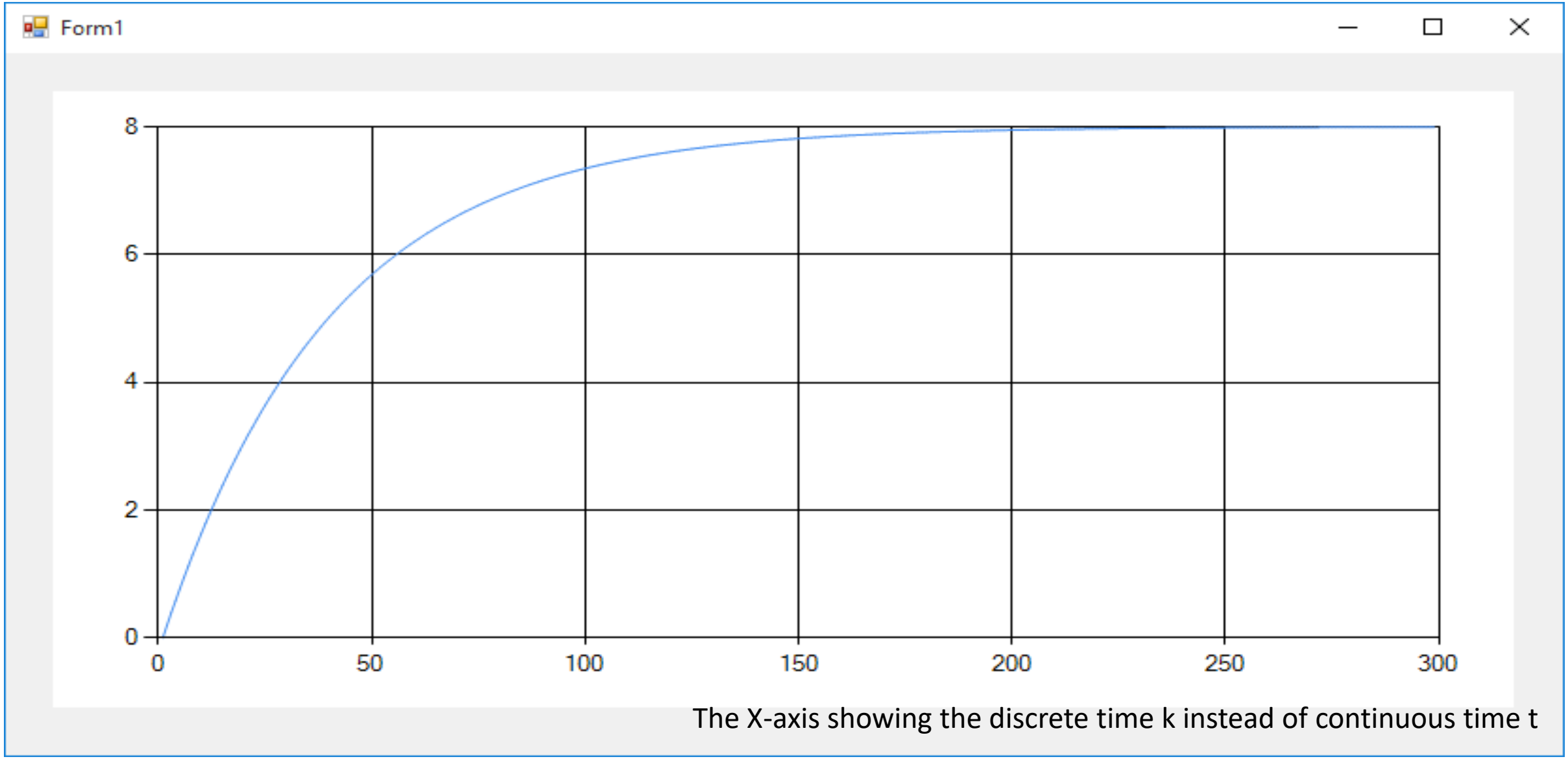
```
using System.Windows.Forms.DataVisualization.Charting;
...
chart1.Series.Clear();
chart1.Series.Add("My Data");
chart1.Series["My Data"].ChartType=SeriesChartType.Line;
...
int[] x = {1, 2, 3, 4, 5, 6, 7, 8};
int[] y = {20, 22, 25, 24, 28, 27, 24, 26};
for (int i = 0; i < x.Length; i++)
{
   chart1.Series["My Data"].Points.AddXY(x[i],y[i]);
}
```

Creating a Web App? Use the following Namespace instead:
System.Web.UI.DataVisualization.Charting

```
using System;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace SimulationExample
{
   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
      }

      private void Form1_Load(object sender, EventArgs e)
      {
         Simulation();
      }

      private void Simulation()
      {
         chart1.Series.Clear();
         chart1.Series.Add("SimulationData");
         chart1.Series["SimulationData"].ChartType = SeriesChartType.Line;

         double[] x;

         double a = 0.25;
         double b = 2;
         double Ts = 0.1;
         double Tstop = 30;

         int size;
         size = Convert.ToInt32(Tstop/Ts);
         x = new double[size];

         double uk = 1; // Step Response

         x[0] = 0;

         for (int k = 0; k < x.Length-1; k++)
         {
            x[k + 1] = (1 - a * Ts) * x[k] + Ts * b * uk;
            chart1.Series["SimulationData"].Points.AddXY(k+1, x[k]);
         }

      }
   }
}
```

```
private void Simulation()
{
        chart1.Series.Clear();
        chart1.Series.Add("SimulationData");
        chart1.Series["SimulationData"].ChartType = SeriesChartType.Line;

        double[] x;

        double a = 0.25;
        double b = 2;
        double Ts = 0.1;
        double Tstop = 30;

        int size;
        size = Convert.ToInt32(Tstop/Ts);
        x = new double[size];

        double uk = 1; // Step Response

        x[0] = 0;

        for (int k = 0; k < x.Length-1; k++)
        {
                x[k + 1] = (1 - a * Ts) * x[k] + Ts * b * uk;
                chart1.Series["SimulationData"].Points.AddXY(k+1, x[k]);
        }
}
```

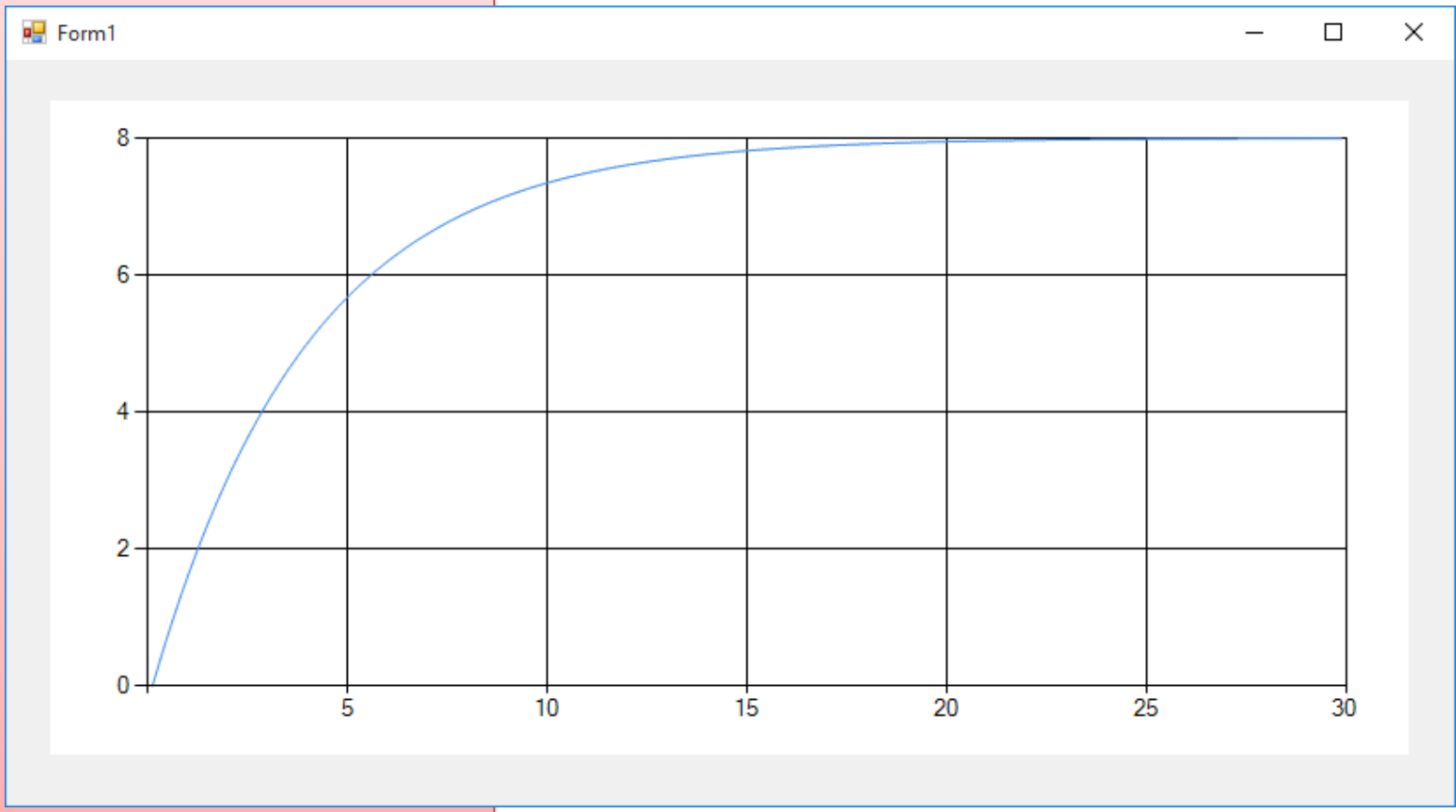
Simulation Results



The X-axis showing the discrete time k instead of continuous time t

Alternativ 2

```
private void Simulation()
{
        chart1.Series.Clear();
        chart1.Series.Add("SimulationData");
        chart1.Series["SimulationData"].ChartType = SeriesChartType.Line;

        double[] x;

        double a = 0.25;
        double b = 2;
        double Ts = 0.1;
        double t = 0;
        double Tstop = 30;

        int size;
        size = Convert.ToInt32(Tstop / Ts);
        x = new double[size];

        double uk = 1; // Step Response

        x[0] = 0;

        for (int k = 0; k < x.Length - 1; k++)
        {
           x[k + 1] = (1 - a * Ts) * x[k] + Ts * b * uk;
           t = t + Ts;
           chart1.Series["SimulationData"].Points.AddXY(t, x[k]);
        }

    }
```



In this case the x axis shows the continuous time t with some small adjustment in the code

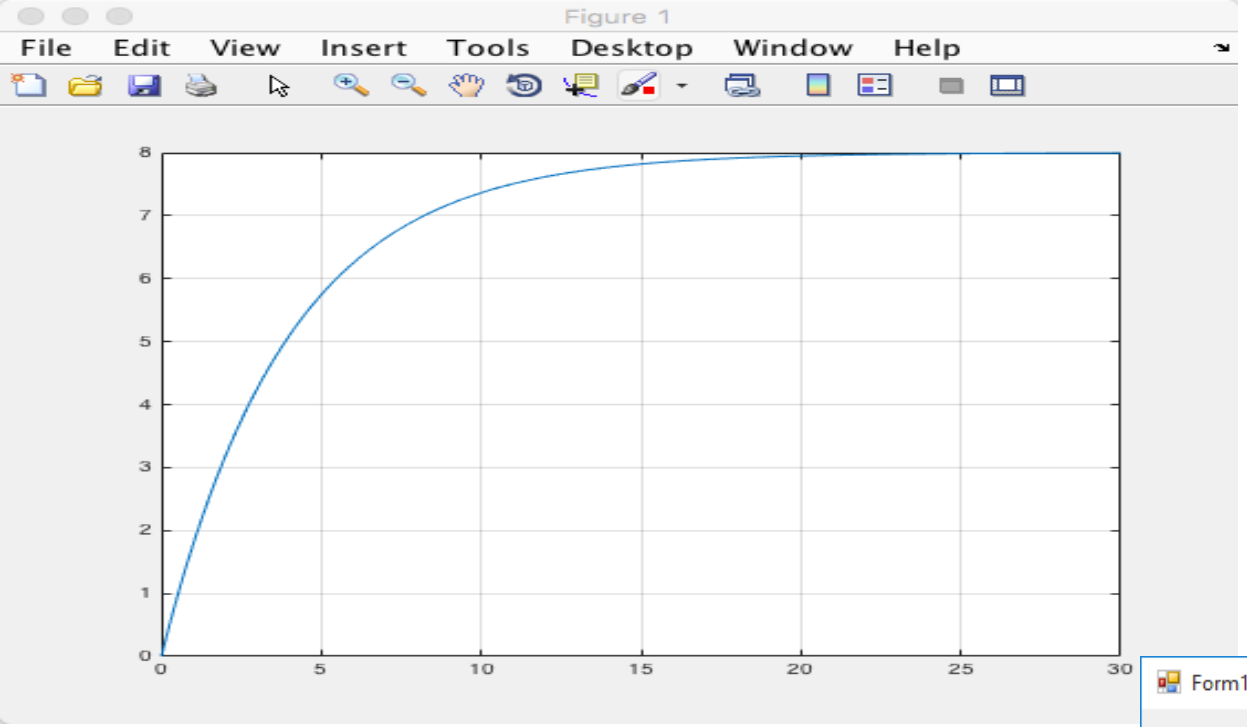
MATLAB

```
% Simulation of discrete model
clear, clc

% Model Parameters
a = 0.25;b = 2;

% Simulation Parameters
Ts = 0.1; %s
Tstop = 30; %s
uk = 1; % Step Response
x(1) = 0;

% Simulation
for k=1:(Tstop/Ts)
   x(k+1) = (1-a*Ts).*x(k) + Ts*b*uk;
end

% Plot the Simulation Results
k=0:Ts:Tstop;
plot(k,x)
grid on
```

The Code is very similar!

Visual Studio/C#

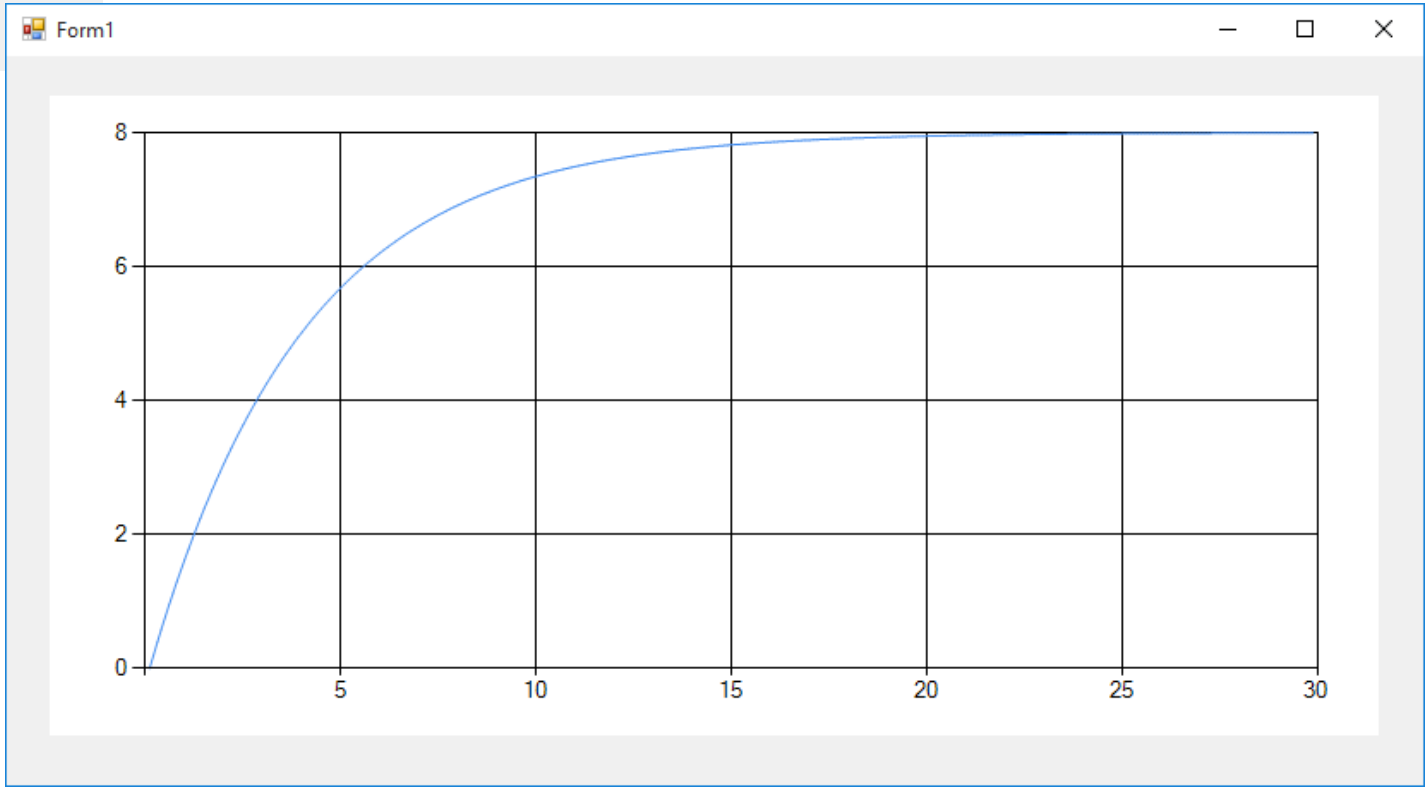
```
private void Simulation()
{
        chart1.Series.Clear();
        chart1.Series.Add("SimulationData");
        chart1.Series["SimulationData"].ChartType = SeriesChartType.Line;

        double[] x;

        double a = 0.25;
        double b = 2;
        double Ts = 0.1;
        double t = 0;
        double Tstop = 30;

        int size;
        size = Convert.ToInt32(Tstop / Ts);
        x = new double[size];

        double uk = 1; // Step Response

        x[0] = 0;

        for (int k = 0; k < x.Length - 1; k++)
        {
           x[k + 1] = (1 - a * Ts) * x[k] + Ts * b * uk;
           t = t + Ts;
           chart1.Series["SimulationData"].Points.AddXY(t, x[k]);
        }

    }
```



MATLAB

Identical Results!

Visual Studio/C#



Real-Time Simulations in Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

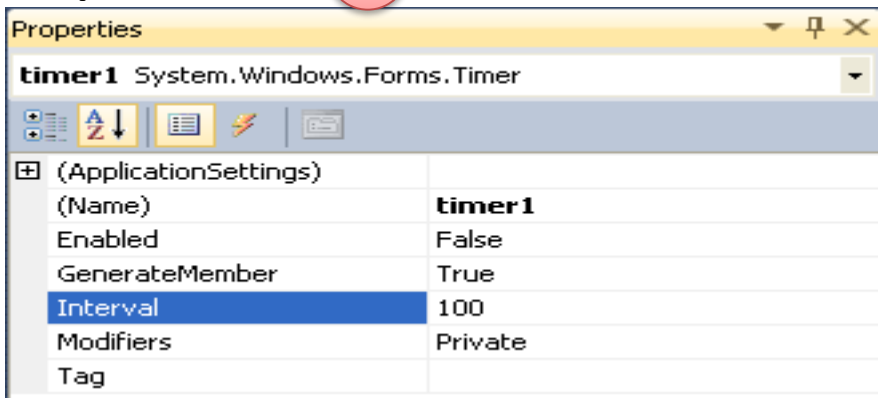
Timer

In Visual Studio you may want to use a Timer instead of a While Loop in order to read values at specific intervals.

1

Select the "Timer" component in the Toolbox

Properties: 3



You may specify the Timer Interval in the Properties Window

2 **Initialization:**

```
public Form1()
      {
         InitializeComponent();

         timer1.Start();
      }
```

Double-click on the Timer object in order to create the Event

4 **Timer Event:**

```
private void timer1_Tick(object sender, EventArgs e)
      {
        … //Read from DB
        … //Formatting
        … //Plot Data
      }
```

Structure your Code properly!!
Define Classes and Methods which you can use here

Hans-Petter Halvorsen, M.Sc.

University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: http://home.hit.no/~hansha/