BME 5742 Bio-Systems Modeling and Control

Lecture 9

Simulink Tutorial 3: Running Simulink from Matlab; Multiple Simulink runs controlled from Matlab

Reference for this part of the tutorial:

- James B. Dabney and Thomas L. Harman, “Mastering SIMULINK”, Prentice Hall 2004.

Recall: Predator-Prey Model

$$\frac{dx}{dt} = rx - axy$$

$$\frac{dy}{dt} = -my + bxy$$

- Prey population = x
- Predator population = y
- Physical parameters: r=prey's birth rate, m=predator's death rate, a and b are interaction parameters.

We shall simulate of the Predator-Prey normalized model:

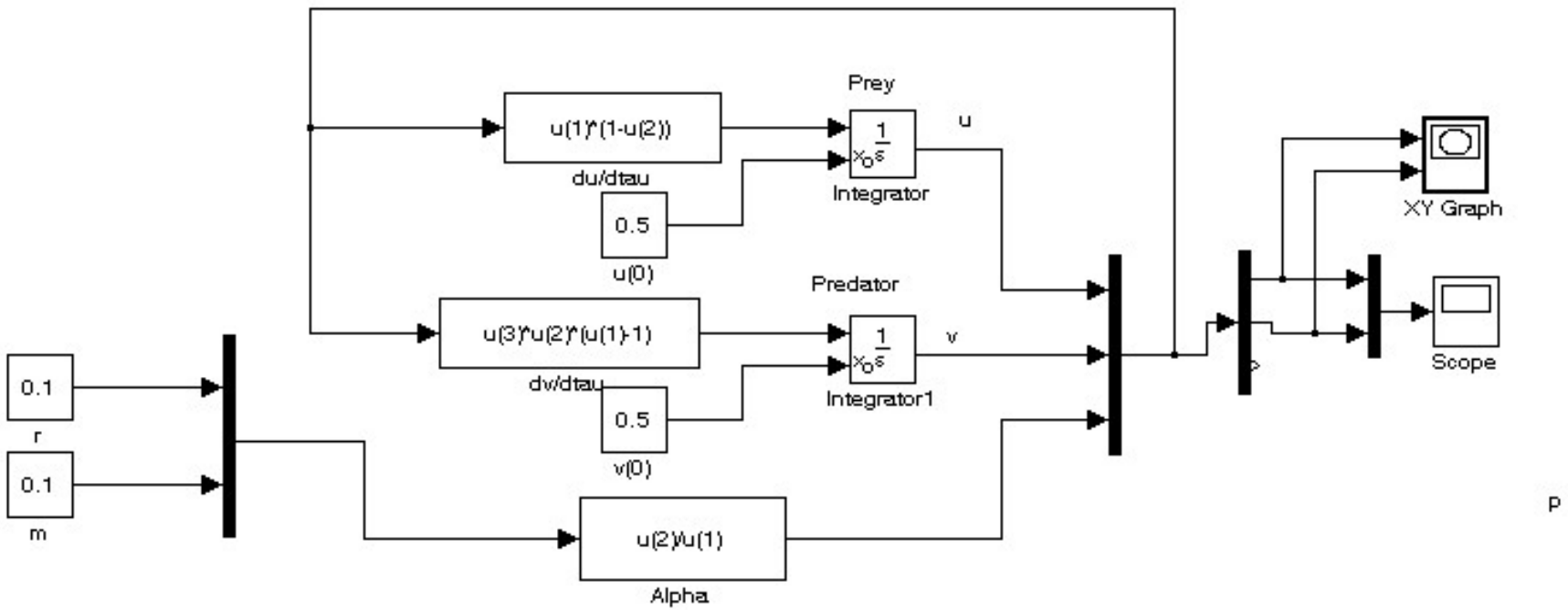
$$u = \frac{b}{m} x \qquad v = \frac{a}{r} y \qquad \alpha = \frac{m}{r} \qquad \tau = r \cdot t$$

$$\boxed{\frac{du}{d\tau} = u(1-v) \qquad \frac{dv}{d\tau} = \alpha v(u-1)}$$

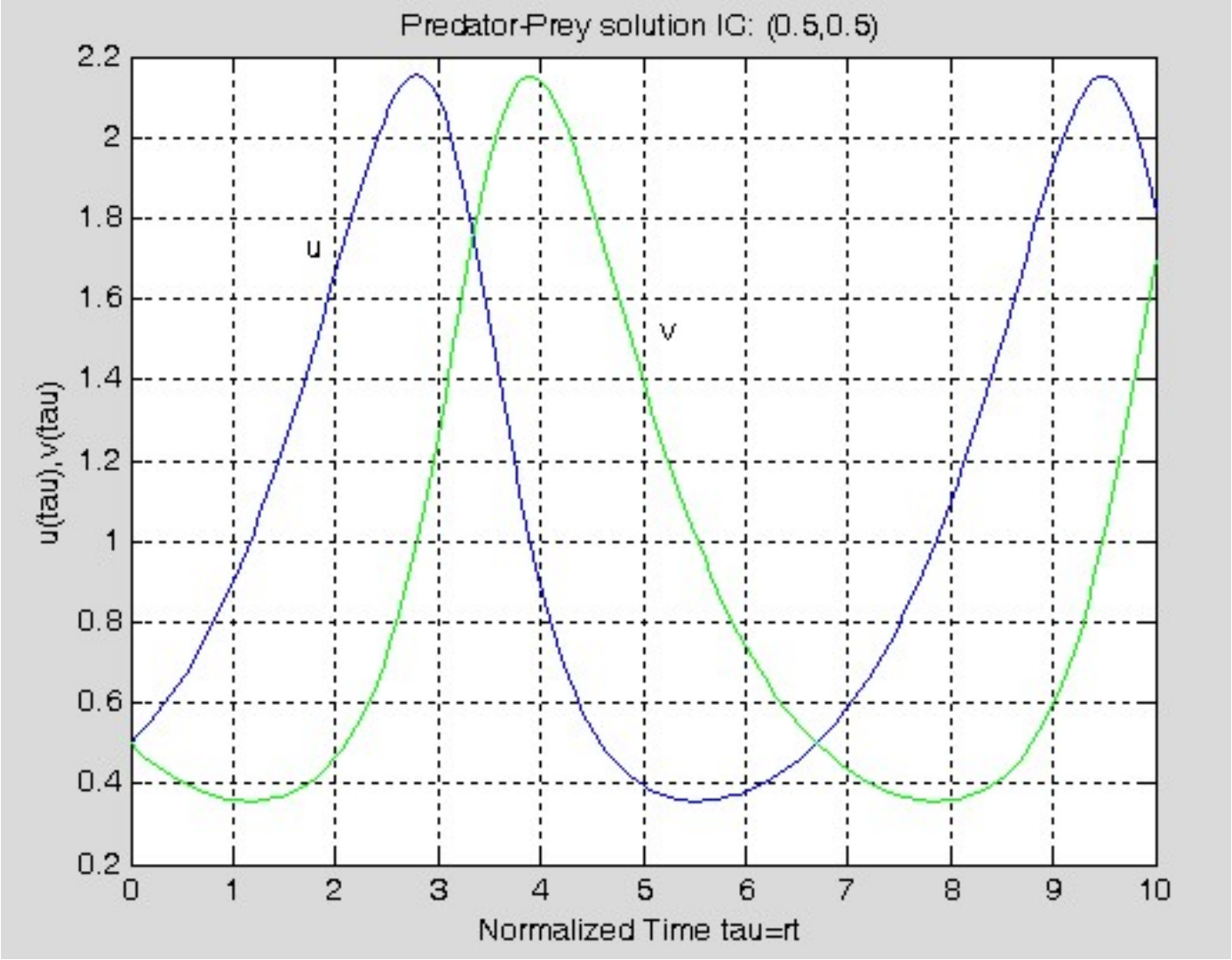
Prey model

Predator model

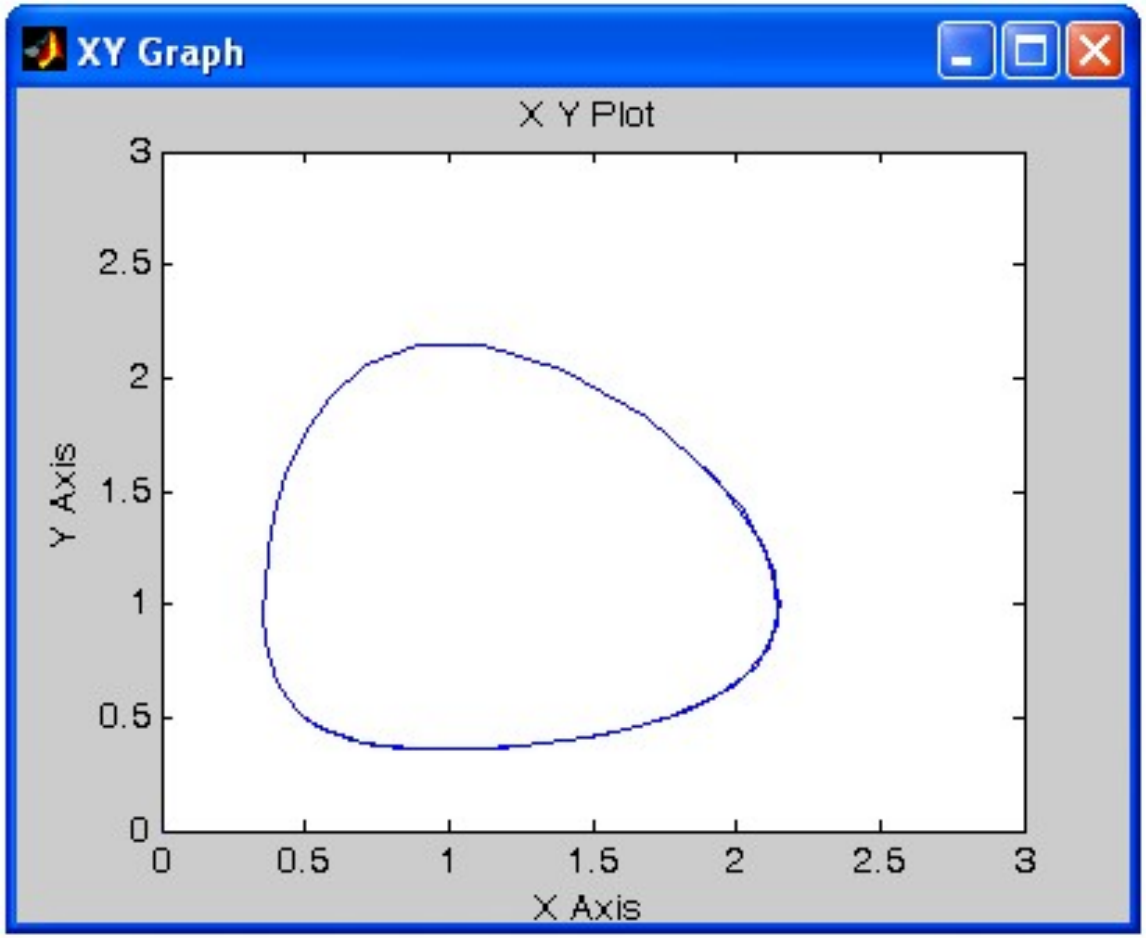
Simple simulation model



Typical scope output (transferred to Matlab for better annotation)



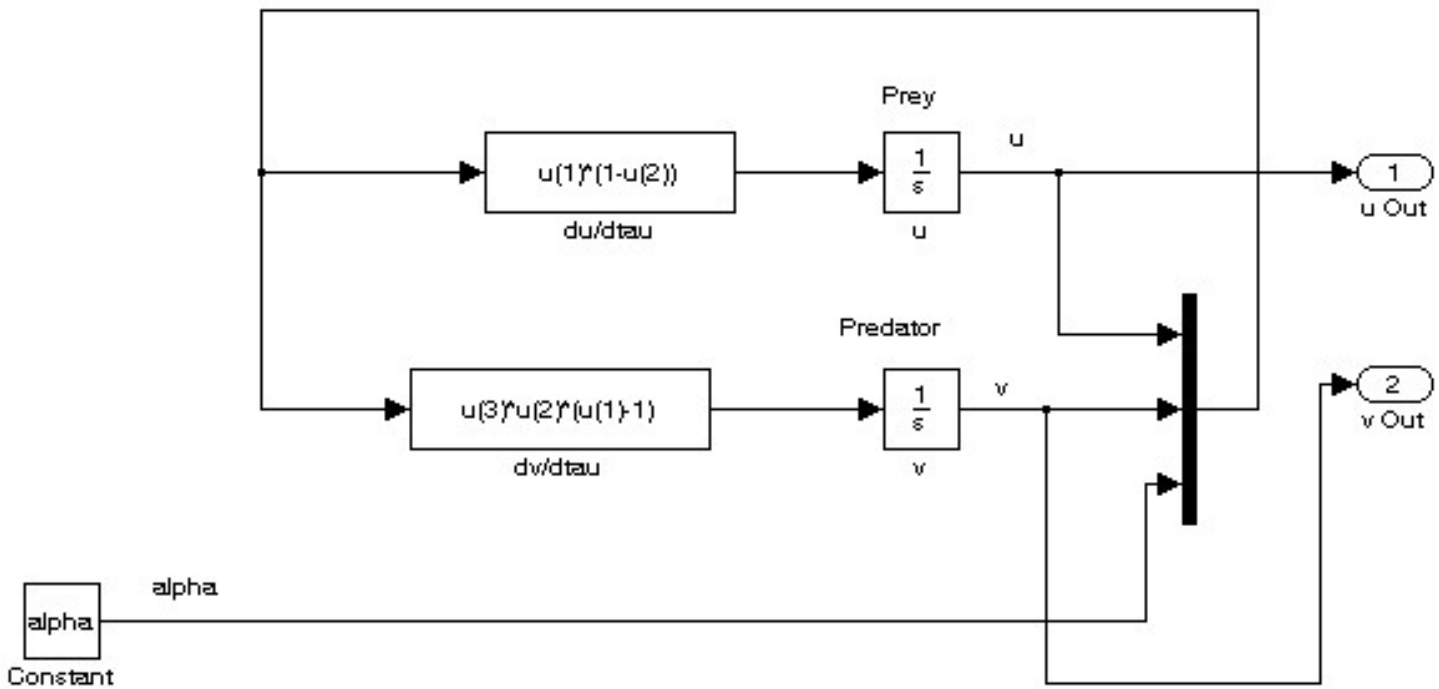
Simulink XY-Graph output cannot be edited, and cannot be sent to Matlab from within the block. Need other means.



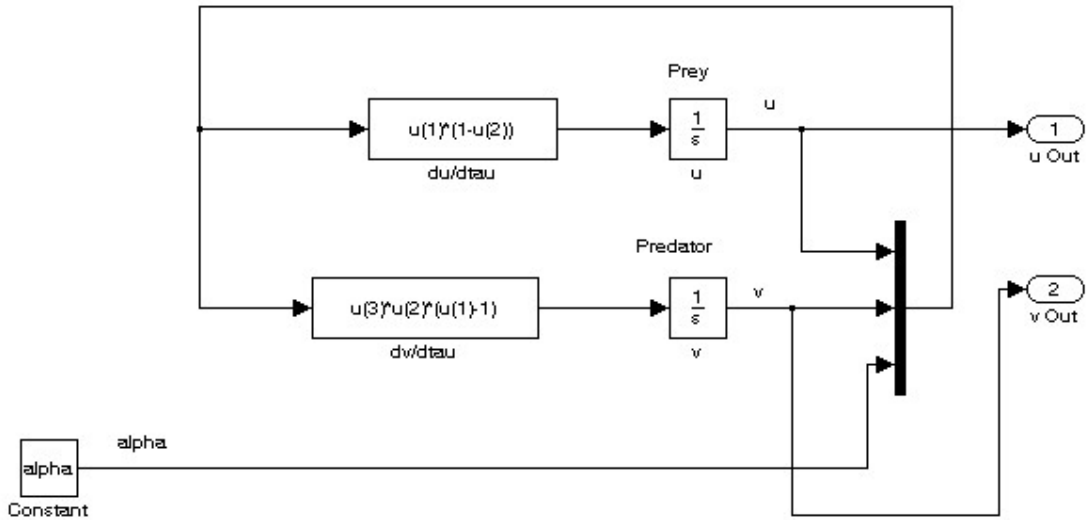
Multiple runs via Matlab

- Let us create a Matlab m file to perform multiple runs of the Simulink program, varying parameters in each run.
- Task 1: Keep α fixed; Vary initial conditions u(0),v(0) →Create an XY graph featuring a portrait of state-plane curves.
- Task 2: Keep α fixed; Vary initial conditions u(0),v(0) →Create multiple u(t) and v(t) plots.
- Task 3: Vary α ; Keep initial conditions u(0),v(0) fixed →Create an XY graph featuring a portrait of state-plane curves.

Modified Simulink model (to allow multiple runs, controlled by Matlab)

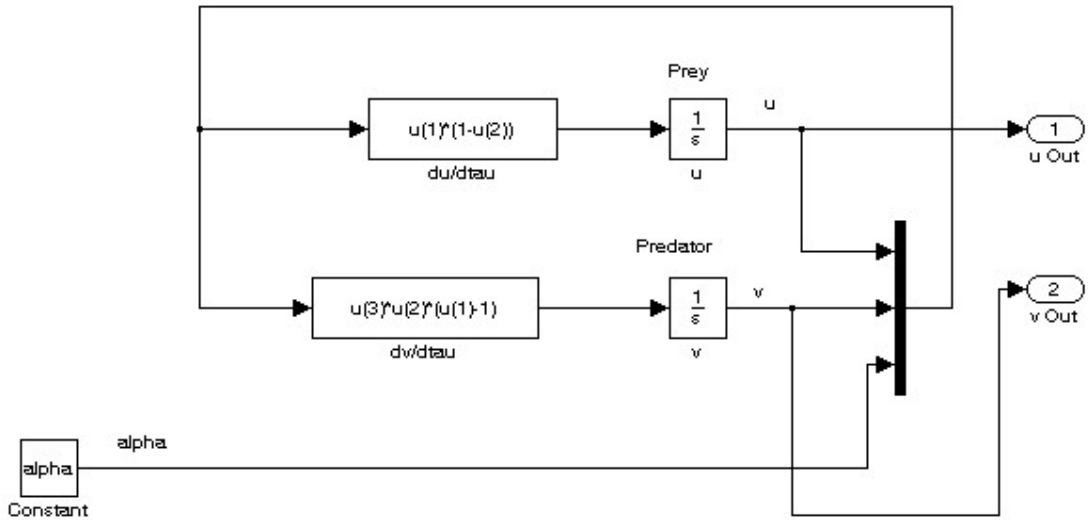


Model Changes:



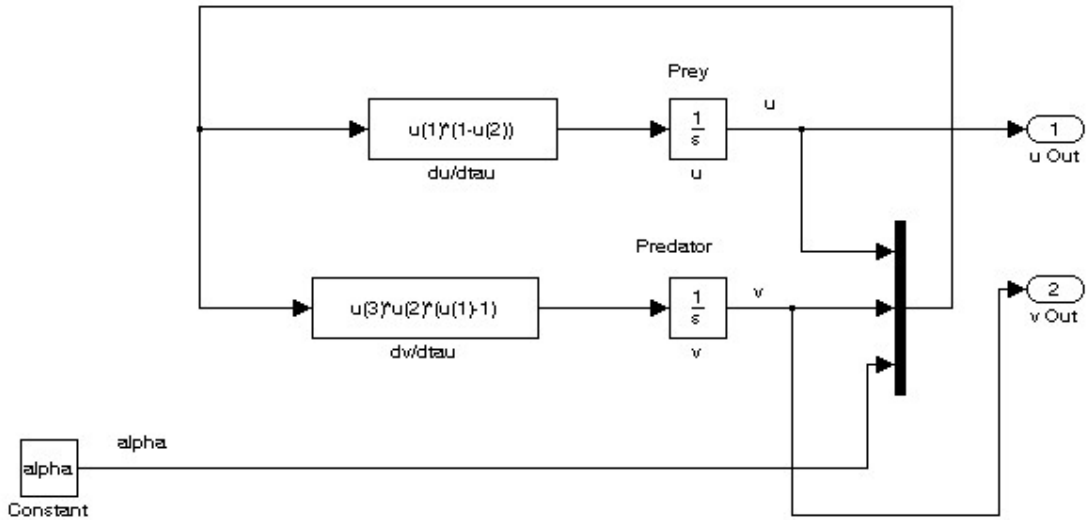
From Ports & Subsystems library add Outport components connected to the output of each integrator. Modify the original Out1,Out2 names – these are the “states” of the model.

Model Changes: (cont'd)



Get the parameter α from a Constant block. However, instead of assigning a numerical constant value, put in a constant name, say "alpha". This variable will be used in the Matlab program.

Model Changes: (cont'd)



Integrator blocks can now go back to "Initial Conditions Source: Internal" option. Initial conditions can now be assigned from the Matlab program.

```
% Produce a state-plane portrait
%   for SIMULINK model
% Bio_predator2_011904
% Alpha parameter is fixed
% Initial conditions vary
%
alpha=2;
for u0=0.5:0.5:3
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        hold on;
        plot(x(:,1),x(:,2));
    end
end
axis([0,5,0,5]);
grid on;
xlabel('u');
ylabel('v');
```

Comments. Include if possible in every line!

Simulink alpha constant

Can edit and run m files from within the debugger/editor. No need to return to the command line.

Multiple runs can be obtained using “for loops” (see below) or “while loops”

```
for u0=0.5:0.5:3
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        hold on;
        plot(x(:,1),x(:,2));
    end
end
```

The main command is “sim”: Its output include the time vector t, the state vector x (you may name these differently), and output vector y.

Sim command operates on the Simulink model name (that you assigned)

```
for u0=0.5:0.5:3
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        hold on;
        plot(x(:,1),x(:,2));
    end
end
```

The “10” argument is optional. This is each simulation’s final time. It overrides the final time specified in Simulation Parameters (in Simulink)

Simset is used to modify any parameter of Simulation Parameters such as RelTol, Solver etc.

```
for u0=0.5:0.5:3
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        hold on;
        plot(x(:,1),x(:,2));
    end
end
```

Here we use simset to modify the integrators' initial conditions' names.

Note: Plot command relates to x, defined in [t,x,y] above.

Plotting of the state elements: The ":" in x(:,1) means, "all rows of x(1)", which is u(t). Similarly, x(2) is v(t)

```
for u0=0.5:0.5:3
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        hold on;
        plot(x(:,1),x(:,2));
    end
end
```

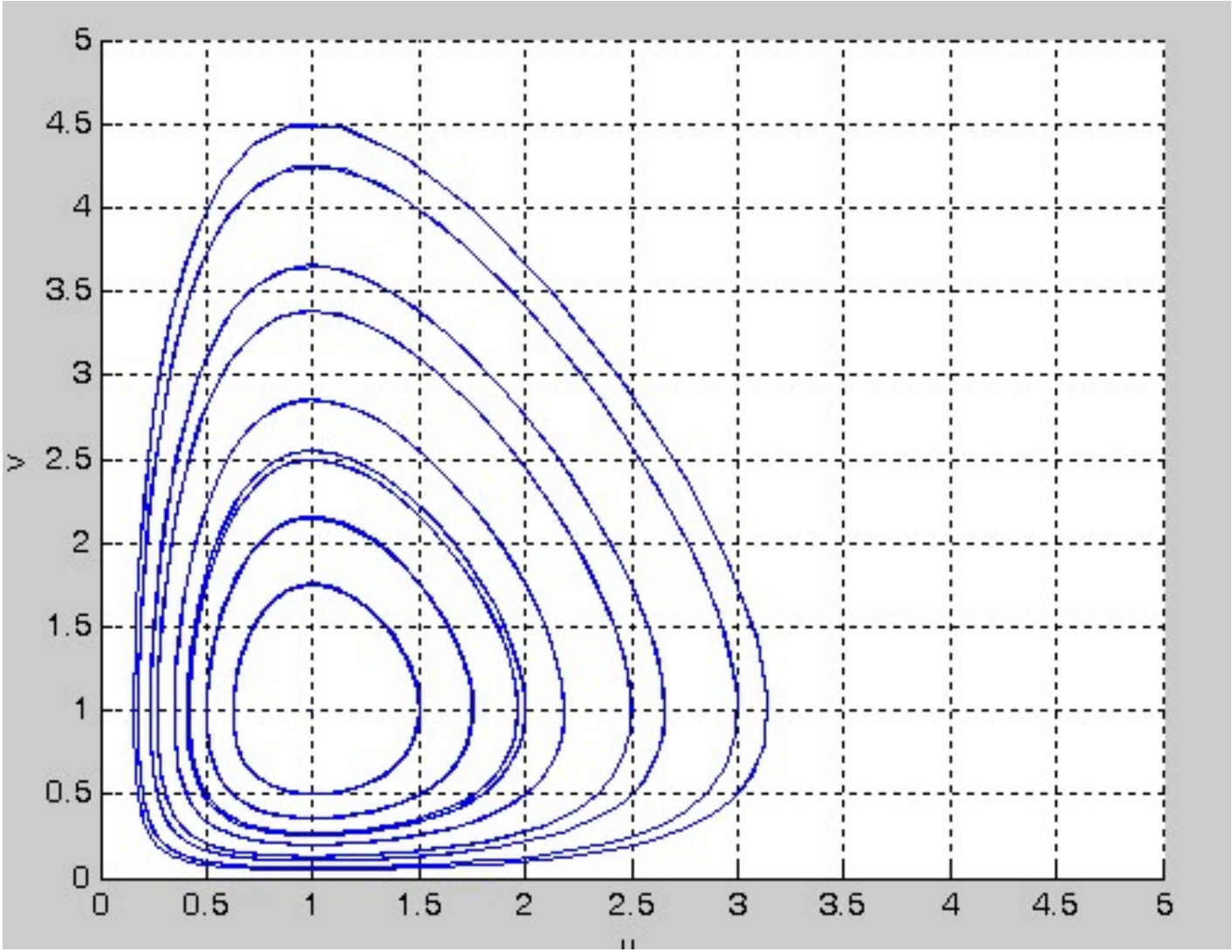
"Hold on" command overlays each simulation result <u>on the same plot</u>.

Plot editing features – axes scale, labels, etc.

```
axis([0,5,0,5]);
grid on;
xlabel('u');
ylabel('v');
```

- Here we include all these commands at the end of the program.
- If we have more than one plot, we may need to move these commands to where each plot command is.

Resulting multiple state-plane plots: Fixed alpha parameter and multi IC (task 1)



```
% Produce time signals portrait
%   for SIMULINK model
% Bio_predator2_011904
% Alpha parameter is fixed
% Initial conditions vary
%
alpha=2;
colorcode=['b','g','r','c','m','y','k'];
colorindex=1;
for u0=0.5:0.5:1.5
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        colorindex=colorindex+1;
        subplot(2,1,1)
        hold on;
        plot(t,x(:,1),colorcode(colorindex));
        ylabel('u(t)');
        subplot(2,1,2)
        hold on;
        plot(t,x(:,2),colorcode(colorindex));
        ylabel('v(t)');
    end
end
grid on;
xlabel('t');
```

Task 2

Here is a trick designed to assign color code to the multiple plots: Define a colorcode string variable, as long as you forsee that is needed. In each run index to a new element of the string vector.

```
colorcode=['b','g','r','c','m','y','k'];
colorindex=1;
for u0=0.5:0.5:1.5
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        colorindex=colorindex+1;
        subplot(2,1,1)
        hold on;
        plot(t,x(:,1),colorcode(colorindex));
        ylabel('u(t)');
```

Here we want a family of u(t) plots and v(t) plots, on two separate frames, on the same page, each plot with its own color code, and each has a different y-axis label.

```
colorcode=['b','g','r','c','m','y','k'];
colorindex=1;
for u0=0.5:0.5:1.5
    for v0=0.5:0.5:1
        opts=simset('InitialState',[u0,v0]);
        [t,x,y]=sim('Bio_predator2_011904',10,opts);
        colorindex=colorindex+1;
        subplot(2,1,1)
        hold on;
        plot(t,x(:,1),colorcode(colorindex));
        ylabel('u(t)');
```

Need to use "Hold on" twice. Subplot (2,1,…) arranges for the two plots to be one above the other.

```
subplot(2,1,2)
hold on;
plot(t,x(:,2),colorcode(colorindex));
ylabel('v(t)');
```

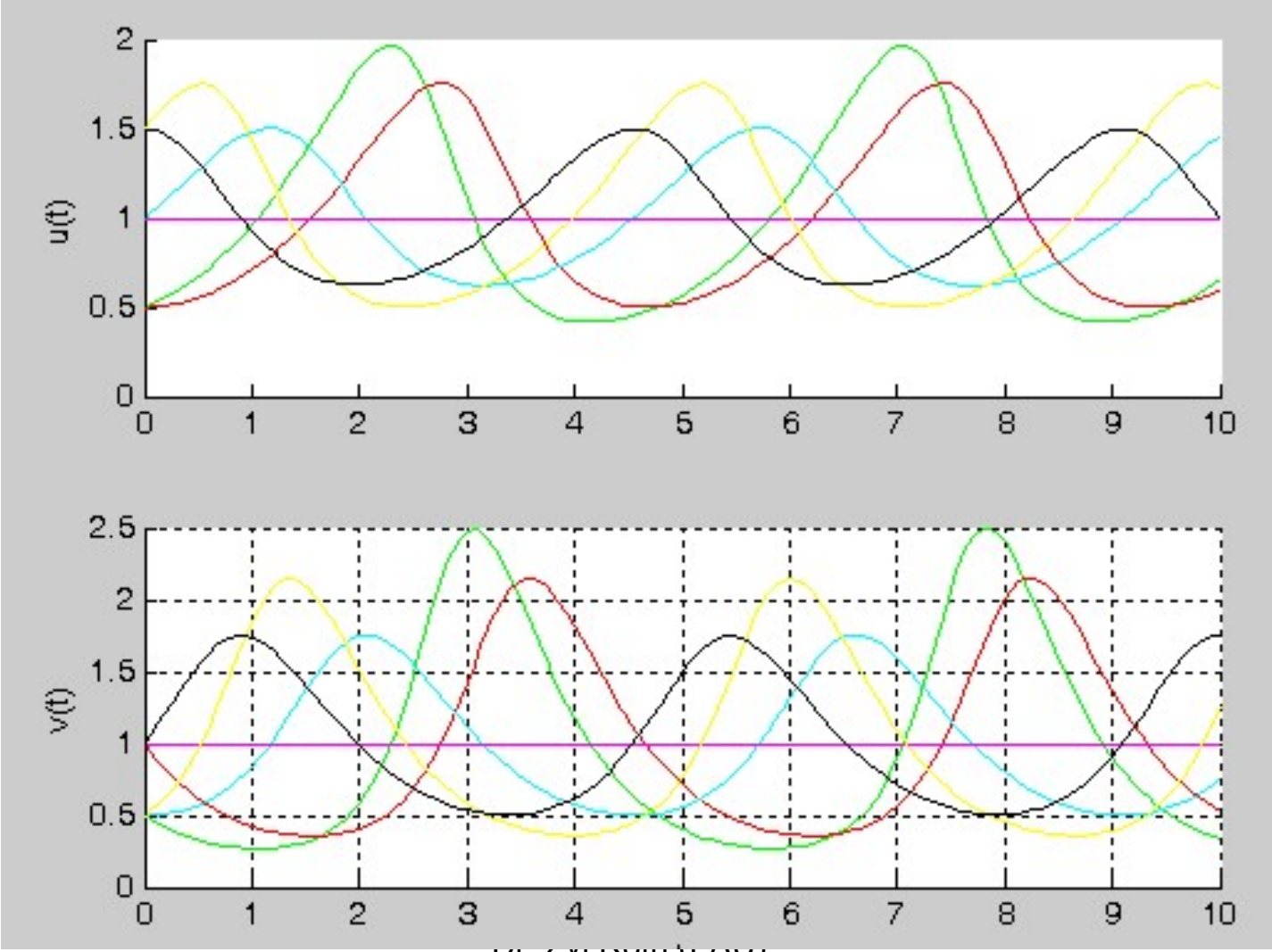
If, for instance, we want the two plots to be side-by-side, we use instead subplot(1,2,…).

In Matlab, you may type "help subplot" to get a full description of the syntax.

Plots commands common to both plots:

```
grid on;
xlabel('t');
```

Resulting plots (Task 2)



Task 3 – varying the alpha parameter

- You may now use similar methods to vary alpha (in a for loop), keeping the initial conditions fixed (outside the for loop)
- Here, using a color code is almost a must. Without it, the plots become unreadable.