SIMULINK tutorial

Objectives

- Apply Matlab coding AND SIMULINK to model a physical system.

Useful Functions & Notation

Elevation angle, α – Angle between the sun and the horizon.

Azimuth Angle, β - Angle along the horizon between north and the sun.

Solar Zenith Angle, θ – Angle between the zenith (straight up) and the sun.

Hour angle, *h* – time after solar noon measured in degrees, ranges between -180^{o} and 180^{o}. Changes by one degree per four minutes or fifteen degrees per hour. Zero at noon (midday). Negative in the morning and positive in the afternoon.

Sunrise / sunset hour angle, ω_0 – hour angle at sunrise or sunset. Positive at sunset and negative at sunrise.

Daily daylight hours, *L* – total hours of daylight in a single day.

Declination, ∂ – Angle between the sun and the plane of the earth's equator. Varies between -23.45^{o} and 23.45^{o}, because of the earth's seasons.

Day Number, *d* – The day of the year.

If you are unsure of the Greek alphabet, this might help:
https://www.medcalc.org/manual/greek_alphabet.php

https://uk.mathworks.com/help/

Part 1 – Where is the sun?

This part we'll use SIMULINK to model the amount of solar radiation received by an area on the earth's surface over some time. We'll start by calculating the position of the sun in the sky over time.

1) Use equation (1) to calculate and plot the declination over the course of a year (d = 1:365).

$\delta = -23.45\cos\left(\frac{360}{365}(\mathrm{d}+10)\right)$	(1)

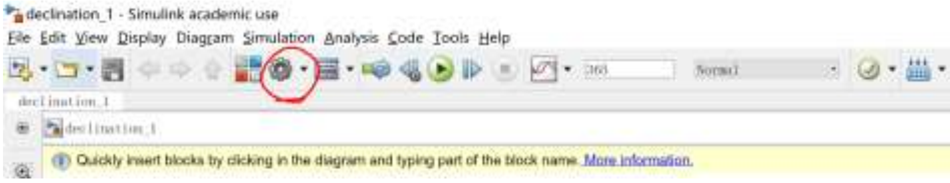
a. Write a function which has day number (d) as its input and declination (δ) as its output. Remember that we're working in degrees so you'll need the cosd function in Matlab. Check that your function gives -13.7018 degrees for d=45.

b. Write a script which uses a for loop to calculate and stores the value of δ for each day of the year, d=1:365.

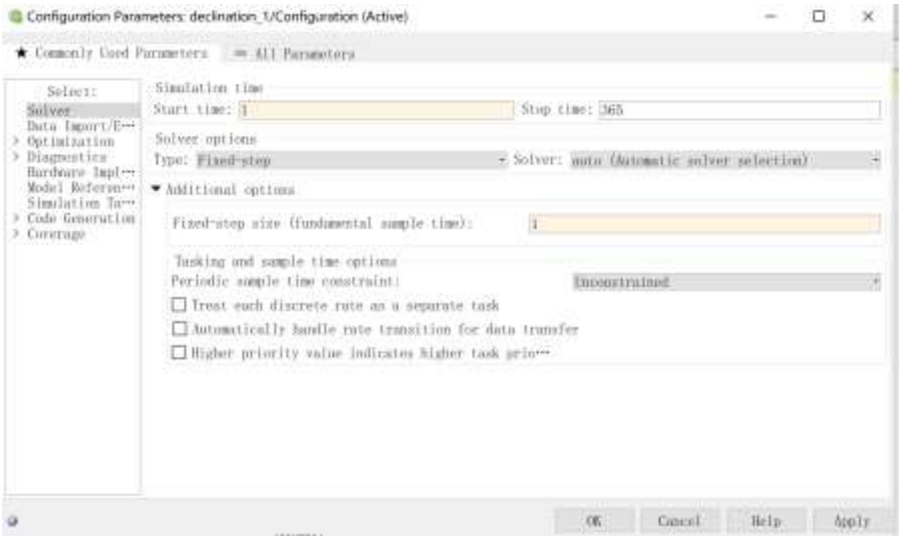
c. Plot your result. You should see a line which varies between -23.45° and +23.45° over the year.

SIMULINK Implementation

- Full answer: declination_1.slx
- To run this answer:



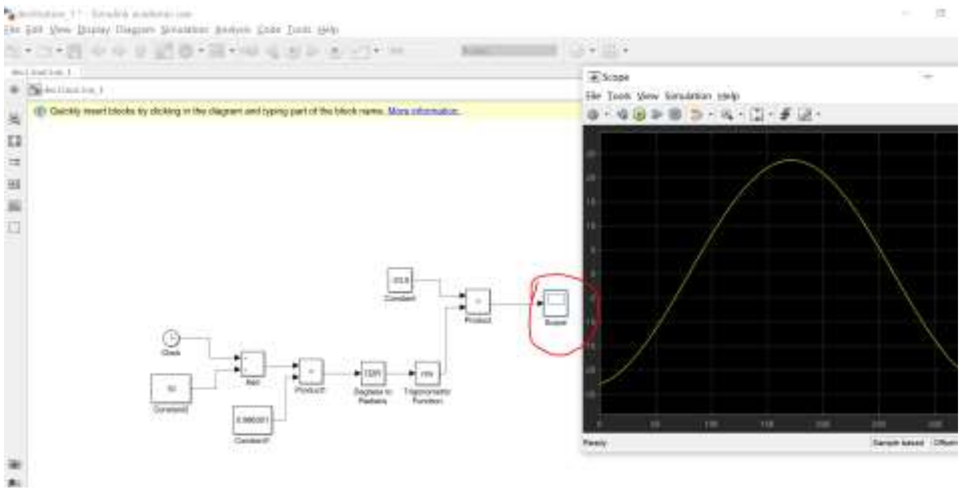
Click the button "Model Configuration Parameters"



Configure as the above, and click "OK". In this page,

- "Start time" 1 and "Stop time" 365 mean the simulation start time step and end time step.
- Solver type "Fixed-step" mean we will do fixed time step discrete simulation.
- "Fixed-step size" 1 means the simulation time step interval.
- Under this setting, its behaviour is like a for loop that we iterate for 365 times; for each iteration, the "Clock" object in our model has the values from 1 to 365.

Click the button "run" to run the model.



After the model finishes, double click the scope to view the answer.

2) Use equations (2) and (3) to calculate the hours of sunlight each year.

$\omega_0 = \cos^{-1}(-\tan\varphi\tan\delta)$	(2)
$L = 48\left(\frac{\omega_0}{360}\right)$	(3)

a. Write a function which has latitude (φ) and day number (d) as its inputs and total daily daylight hours (L) as its output. Hint: you'll need to use your function from step 1 as part of this new function.

   Use Google or Matlab help to find the function for inverse cos. Again remember that we're working in degrees.

   Check that your function gives L = 11.55hrs for $\varphi = 53$ (Nottingham) and d=75.

b. Write a script which uses a for loop to calculate and store the value of L for each day of the year, d=1:365.

c. What is the maximum number of daily daylight hours in Nottingham? (answer = 16.68hrs).

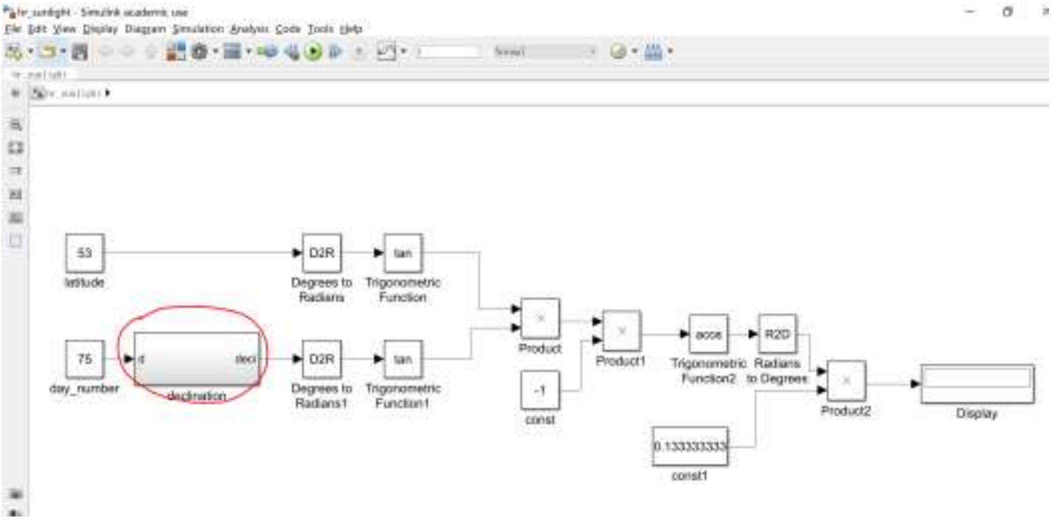
d. What is the total number of daylight hours in one year, in Nottingham? (answer = 4380 hours).

e. What is the total number of daylight hours at $\varphi = 64$ (Reykjavík) and $\varphi = 21$ (Honolulu)?

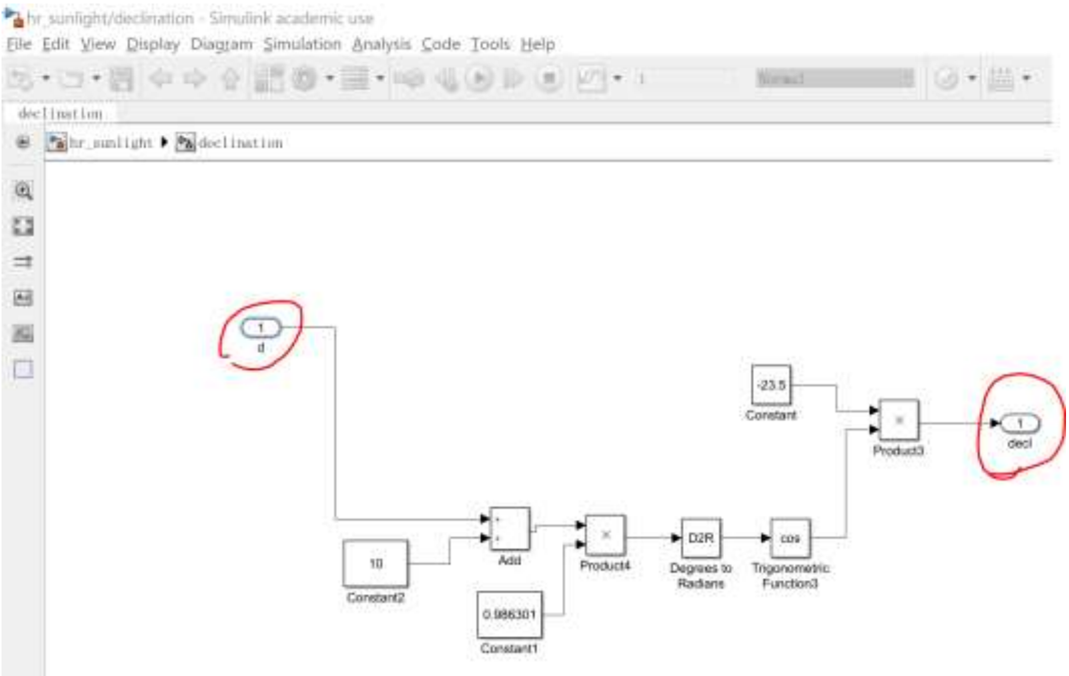
You should find that the total annual daylight hours is the same regardless of where we are on earth. This is true when we make the assumption that the earth is a sphere and has no atmosphere. In fact the total hours of daylight will vary very slightly from place to place but we've chosen to keep our model simple. The longest and shortest day does vary a lot however. The longest day in Reykjavík is over 20hrs and the shortest under 4hrs. In Honolulu, near the equator, the longest day is 13hours long but the shortest is just under 11hrs.

SIMULINK Implementation

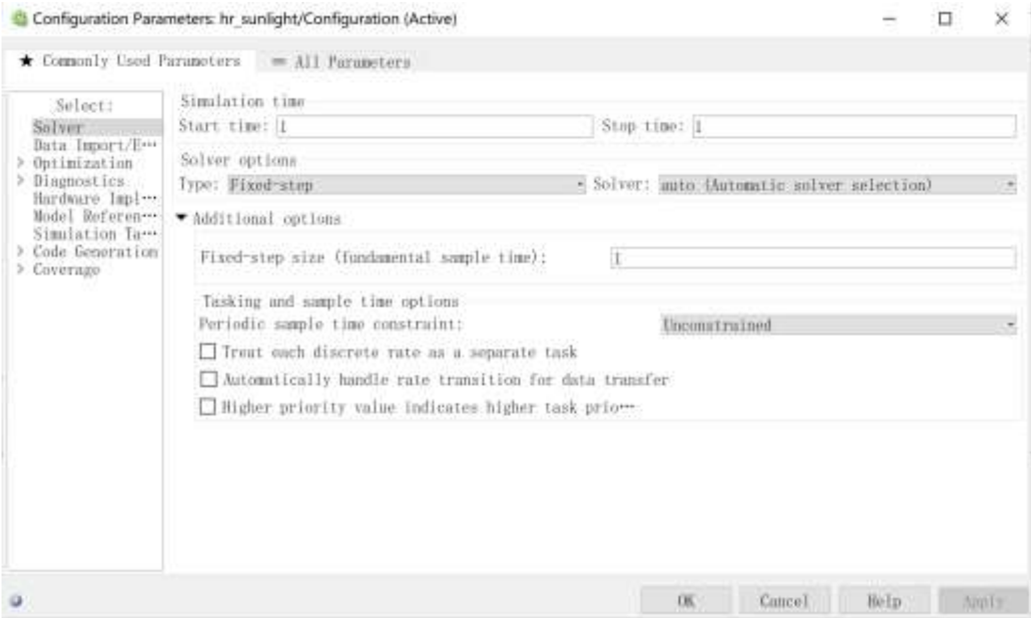
- Full answer (for question a): hr_sunlight.slx
- Note the use of subsystem for the declination calculation. You can double click to see what is inside.



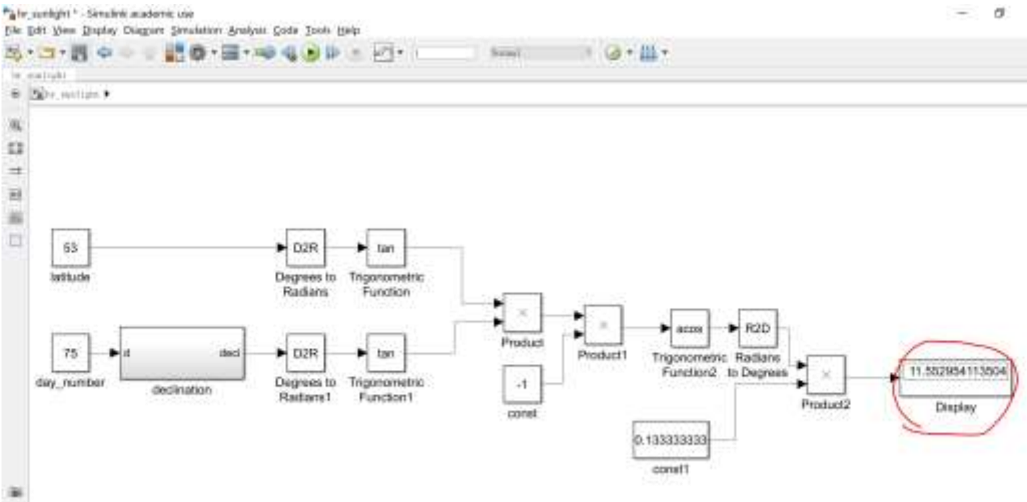
- Subsystem in SIMULINK is a little like function in MATLAB. It exposes some “in” ports and “out” ports, and hide the complicated logics. If you get inside the subsystem “declination”, you can find an “in” port named “d”, and an “out” port named “decl”, they the subsystem’s input and output. We can have multiple inputs and outputs for a subsystem, just like a MATLAB function.



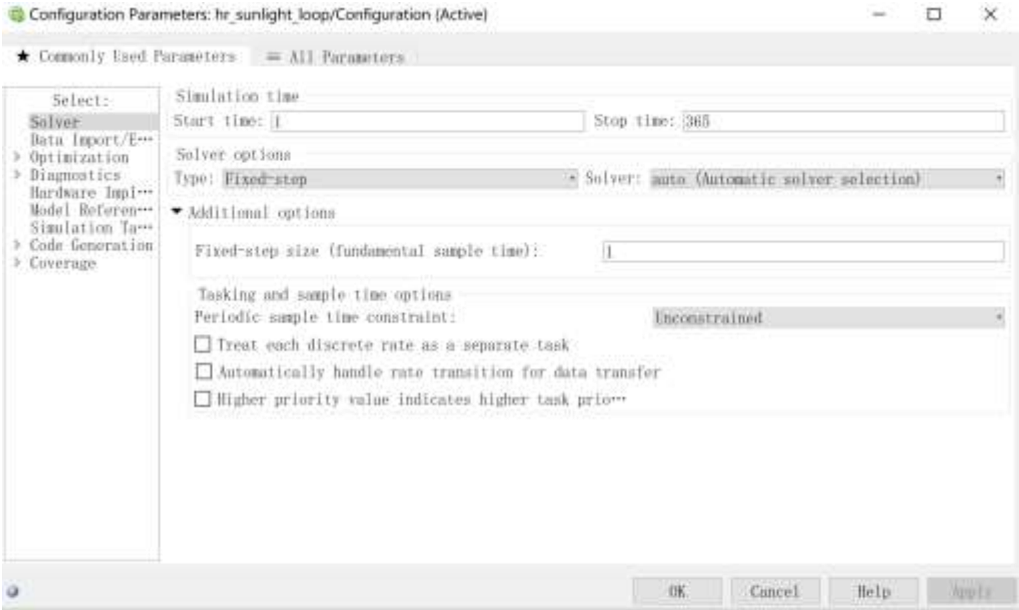
- Configure the model parameters as the following, and run the model.

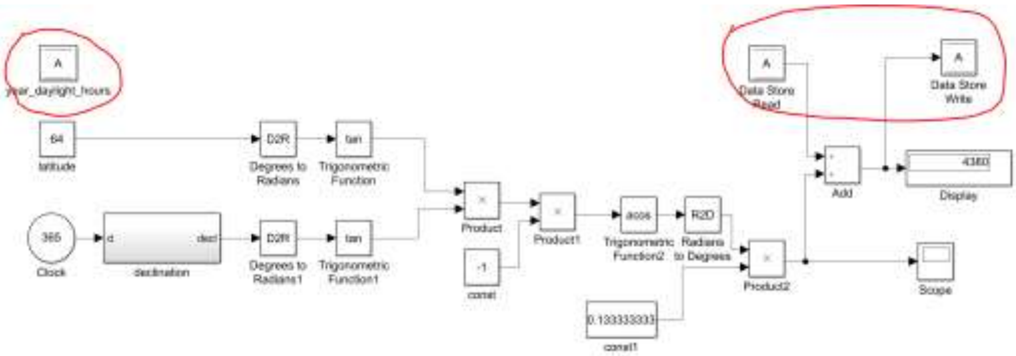


- And you will get the answer to question a.

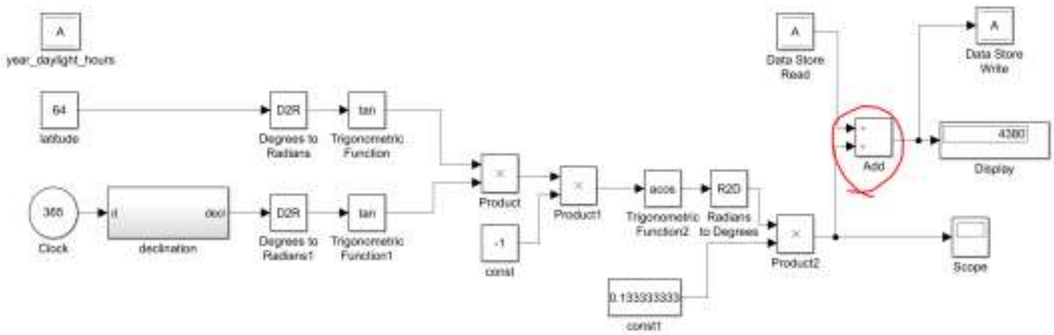


- Full answer (for the remaining questions): hr_sunlight_loop.slx
- This model is similar as the last one, but with some changes to realize a for loop in SIMULINK.
  1) Different configurations for the model, as the following

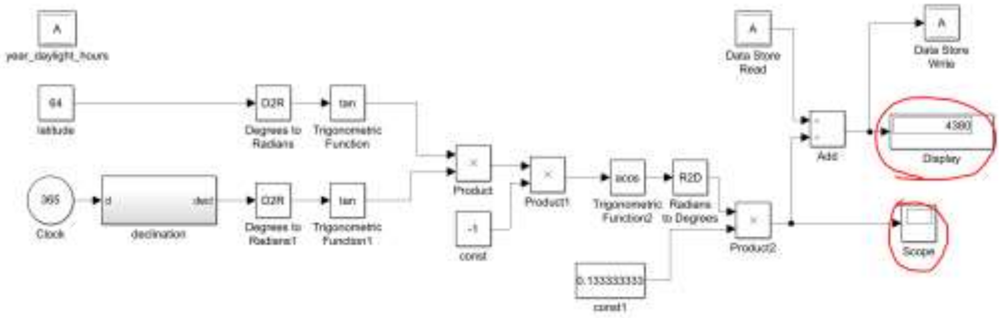




2) Add “Data Store” to store the accumulative daylight hours.



3) Add an “Add” block to accumulate daylight hours.



4) Add a “Scope” and a “Display” block to display the results.

- Double click “Display” to view the results of daylight hours at each day, and the “Display” object shows the result of the total daylight hours in a year.

3) Use equations (4) to (7) to calculate the path of the sun across the sky each day.

$h = 360(d - [d])$	(4)
$\alpha = \sin^{-1}(\sin\varphi\sin\delta + \cos\varphi\cos\delta\cos h)$	(5)
$\theta = 90 - \alpha$	(6)
$\beta = \begin{cases} -\cos^{-1}\left(\dfrac{\sin\delta\cos\varphi - \cos h\cos\delta\sin\varphi}{sin\theta}\right) & if\ h \le 0 \\ \cos^{-1}\left(\dfrac{\sin\delta\cos\varphi - \cos h\cos\delta\sin\varphi}{sin\theta}\right) & otherwise \end{cases}$	(7)

Where [d] denotes a rounding of d to the nearest integer. Note that equation seven has two cases – before midday and after, there is only a small difference between them.

a) Write a function to describe position of the sun and any time in the day. We can use elevation angle (α) and azimuth (β) to describe the path of the sun from any place on earth so these will be the output of the function. The inputs will be everything else we need: d and φ. You will need to use the function you wrote in part 1.

   Check that your function gives α =17.83 and azimuth β = 75.98 for d = 156.25 and φ = 53.

b) Use the code below in a new script to evaluate your function over one day. You may need to replace my function name (getSunPos) with whatever you called your function in step (a). If you don't understand what this code is doing then please ask!

```
clear
phi = 53;
k = 1;
for d = 55.5:0.01:56.5
    [alpha(k),Beta(k)] = getSunPos(d,phi);
    k=k+1;
end
```

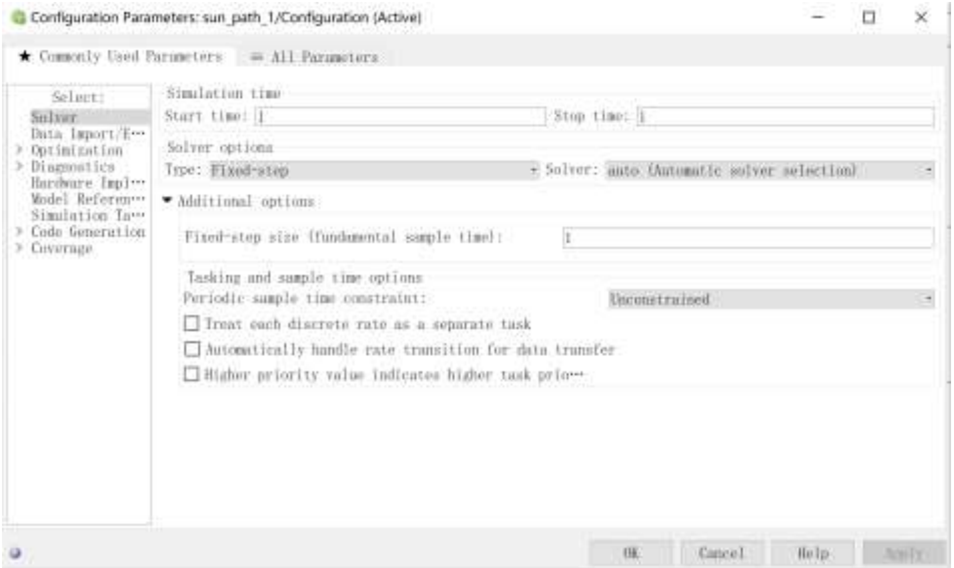
c) Plot α (y axis) against β. What does a negative value for α mean?

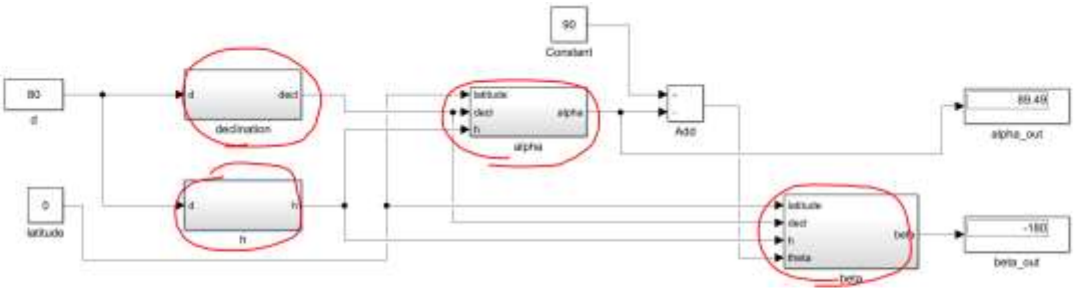
d) What is the maximum value for α on this day? (answer = 27.1 degrees).
e) What is the maximum value for α for the day if φ = 0 and d=79.5:0.01:80.5?

The maximum value of α in part 3e should be 89.5 degrees. On the solar equinoxes the day length is the same as the night length everywhere on earth (12hrs). On these days the sun appears directly overhead at the equator (φ = 0) at midday (d = 80 or d = 264).

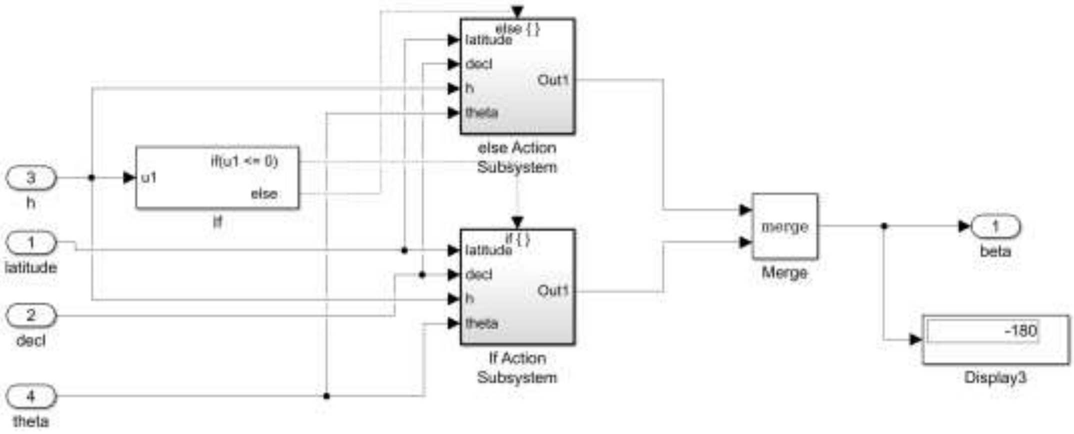
SIMULINK Implementation

- Full answer (for question a): sun_path_1.slx. To run this model, use the following settings:

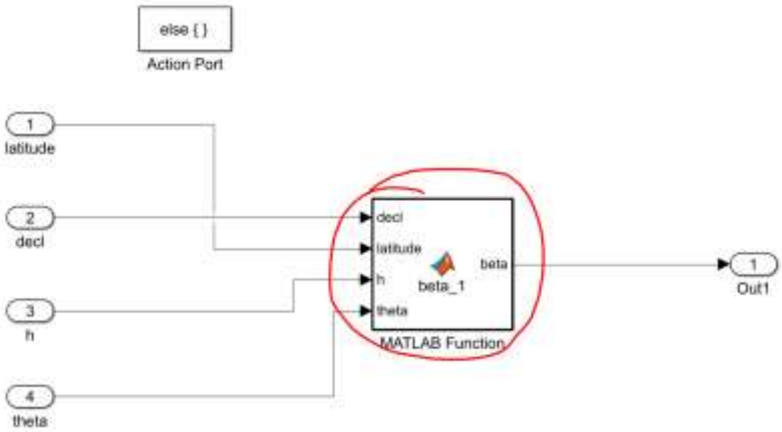




- Basically, each subsystem corresponds to each equation.
- If you double click the subsystem “beta” to open it, you will see this subsystem contains if-else branches.



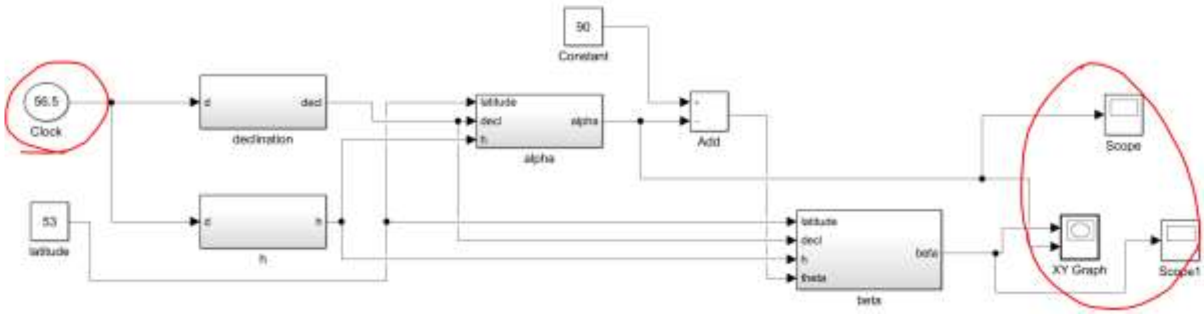
- To be honest, it is painful to use SIMULINK to write mathematical equations, so we can actually use MATLAB function in SIMULINK. Double click “else Action Subsystem” (or “if Action Subsystem”), you will see a MATLAB function block.



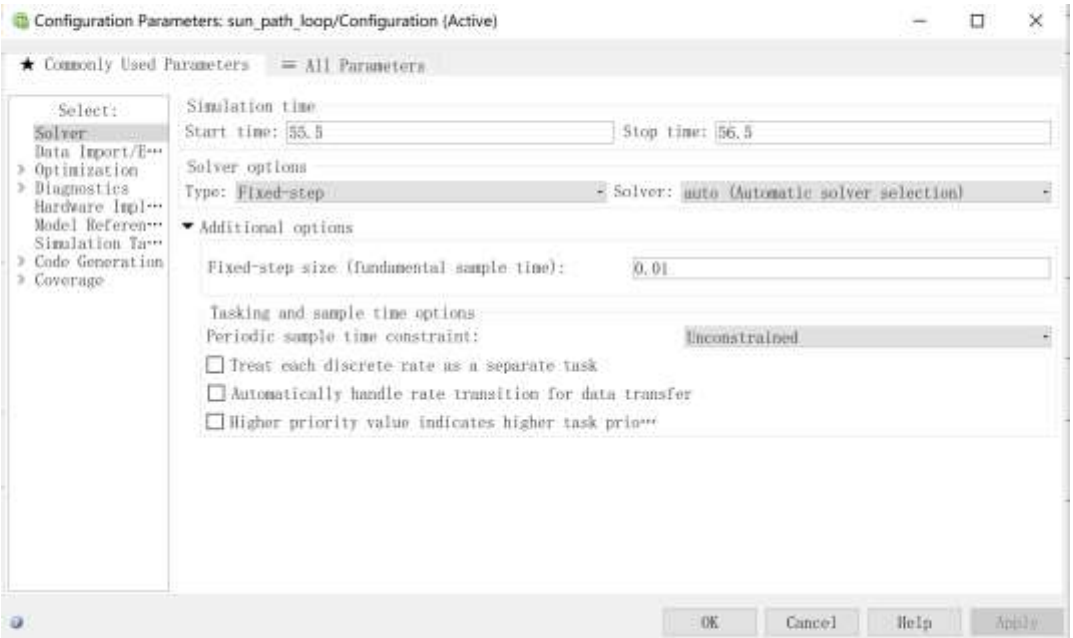
The usage of MATLAB function block in SIMULINK is actually the same as a normal MATLAB function. Your function must have function inputs and outputs (as shown below), and the names of the inputs and outputs will appear on the MATLAB function block.

```
function beta = beta_1(decl, latitude, h, theta)
decl = deg2rad(decl);
latitude = deg2rad(latitude);
h = deg2rad(h);
theta = deg2rad(theta);
trian_val = (sin(decl)*cos(latitude) - cos(h)*cos(decl)...
            *sin(latitude))/sin(theta);
trian_val = min(max(trian_val, -1), 1);
beta = acos(trian_val);
beta = rad2deg(beta);
end
```

- Full answer (question b c d e): sun_path_loop.slx
- To be honest, it is a little bit difficult to implement these questions in SIMULINK, because the plotting capability of SIMULINK is weak. But anyway, since we just want to learn SIMULINK, let me show you how to roughly do these questions in SIMULINK.



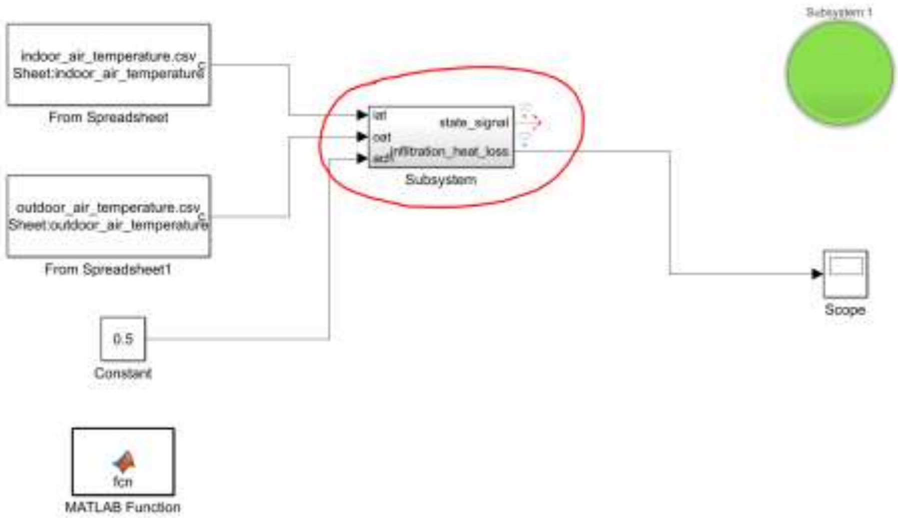
- Compared to the previous model, this model adds a “Clock” object on the left and some plotting objects on the right. In addition, we need to change the run settings as the following. This setting will ensure us to loop through d from 55.5 to 56.5 with 0.01 interval (for question b c d).



- After running the model, double click the “Scope” objects and “XY Graph” object to view the results. Note that, since SIMULINK has weak plotting capability, the results may not be that intuitive.

Part 2 – Practices

1. **Infiltration heat loss and indoor environment indicator.**
   Open practice.slx file. In this SIMULINK model, you can find a subsystem block, as shown below.



This subsystem has three inputs and two outputs. Its inputs are:

1) iat: indoor air temperature in degree C.
2) oat: outdoor air temperature in degree C.
3) ach: air change per hour

Its outputs are:

1) state_signal: state_signal is 1 if oat > 18; state_signal is 2 if oat <= 18 and iat > 20; state_signal is 3 for other conditions.
2) infiltration_heat_loss: infiltration heat loss by using the following equation:

$$infiltration\_heat\_loss = C_p \times \rho \times ACH \times \frac{V}{3600}(iat - oat)$$

where $C_p = 1$ kJ/kg/K, $\rho = 1.2$ kg/m3, $V = 300$ m3

When you run this SIMULINK model, you will can double click the "Scope" object to open the scope window. You will see the value of infiltration_heat_loss is plotted on the opened window, and the big round lamp above changes its colour based on different state_signal.

Part 2 – MATLAB implementation

1) Use MATLAB, rather than SIMULINK, to finish the questions in part 1.