Binary Star System

Zheyu Lu **Rui Liu** **Shuqi Xu**
University of California, Berkeley
{zheyulu, rui_liu, sqxu}@berkeley.edu

Abstract

We create a fictional binary star solar system which has two suns, one Earth and one moon and simulate the orbital motion of these celestial bodies. Based on the observation data from a local position on Earth, we use machine learning techniques and successfully learn the time evolution curve of the moon phase and the altitude and azimuth of the other three celestial bodies including the two suns and the moon. Furthermore, we demonstrate the underlying physics of these time evolution curves which can help us better understand the physical principles dominating the astronomical phenomena. Below is the link to our project directory:
https://github.com/Leo-godel/Project-S-final

1 Introduction

We create a fictional binary star solar system which has two suns, one Earth and one moon. In our early project, we generate our data by using SpaceEngine and OCR. This time, however, we come up with a new method to generate the altitude and azimuth data of the celestial bodies by analytically solving our model which generalized and simplified our previous method. Our current method can be used to quickly generate observation data from an arbitrary position on Earth. Combining the two methods can provide us with abundant data for learning the orbital motion and positions of the celestial bodies and the moon phase.

Our results can be used to further study and predict both solar and lunar eclipse phenomena since we can accurately predict the time evolution curve of the sun's and the moon's position on the sky. That will be an interesting work for us in the future.

The paper is organized as follows: In Section 2, we firstly introduce data generation in Section 2.1 which includes the two different methods we used to generate our data and we also verify the consistency between using the two methods. For the second method, we also introduce the theory and show the analytical derivation of the coordinate transformation. In Section 2.2, we discuss the physical intuition when we analyzed the data and chose the relevant features. This can be viewed as preprocessing the data with reasonable physical consideration and certain domain knowledge. The main learning results of the position of the celestial bodies and the moon phase are shown in Section 3. In Section 4, we conclude the results we learned and also discuss the underlying physics which is related to the observed astronomical phenomena.

2 Method

2.1 Data generation

2.1.1 First method: SpaceEngine screenshots & OCR

One way to generate data is to simulate the binary star system in SpaceEngine. But SpaceEngine doesn't provide users with API functions to directly export data, so we can only take screenshots and then use optical character recognition (OCR) to extract data from the raw pictures. We managed to generate 27,000 data points with 3-year span in 3 different locations: Berkeley, Shanghai and Sydney with very high recognition accuracy (at least 99%). The data includes time, spherical coordinates, *i.e.*, altitude and azimuth of two suns and one moon in observer's topocentric coordinate system, the size of the three celestial bodies, and the phase of the moon.

2.1.2 Second method: analytical derivation of the altitude and azimuth of celestial body

In addition to the data generated by using OCR on screenshots from SpaceEngine, we also simulated all celestial bodies' orbital motion by using Mathematica in the two-sun center of mass coordinate system, where the four celestial bodies, *i.e.*, the two suns, the Earth, and the moon, are arranged to be along the x-axis at $t = 0$ as shown in Figure 1.

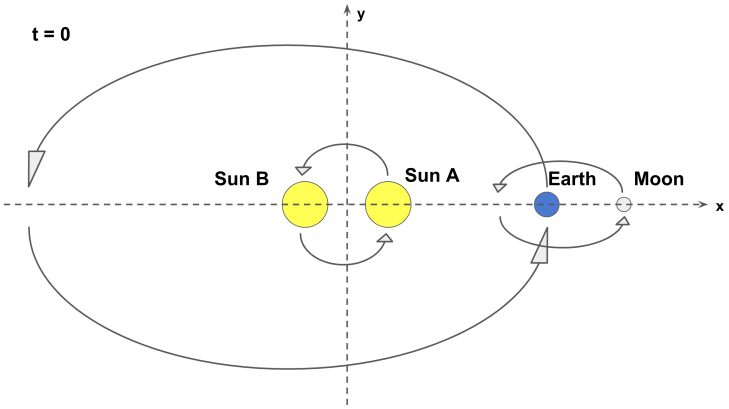


Figure 1: The initial condition of the binary star system

To begin with, we generate the orbital motion, *i.e.*, x, y, z coordinates for all celestial bodies in the coordinate system S shown in Figure 1. Our goal is to transform these coordinates into observations(altitude and azimuth) in observer's topocentric coordinate system S_3 on Earth(defined below). In order to derive such transformation, we need to introduce three coordinate systems:

1. The center of mass reference system of Earth without rotation. We call it S_1 which is shown in Figure 2.
2. The fixed center of mass reference system of Earth. We call it S_2 which is shown in Figure 3 on the left. The x-axis is pointing to the intersection of equator and prime meridian and the z-axis is pointing toward north pole. ξ_0 here is the obliquity of Earth. S_2 and S_1 are related by the Euler angles $(\frac{\pi}{2}, \xi_0, \gamma_0)$.
3. The local observer's reference system which is fixed relative to the Earth. We call it S_3 which is shown in Figure 3 on the right. The x-axis is pointing toward south and the y-axis is pointing toward east. S_2 and S_3 are related by the Euler angles $(\phi, \theta, 0)$ and a translation with length R_e, *i.e.*, the radius of the Earth. Note that the latitude and longitude are given by $90 - \theta$ and ϕ respectively.

Now, for any celestial body, we want to find the transformation between (x, y, z) and (x_3, y_3, z_3).

First, we transform (x, y, z) into (x_1, y_1, z_1) which is simply given by

$$\begin{aligned} x_1 &= y_e - y \\ y_1 &= -x_e + x \\ z_1 &= 0 \end{aligned} \tag{1}$$

where (x_e, y_e, z_e) is the Earth's coordinates in S.

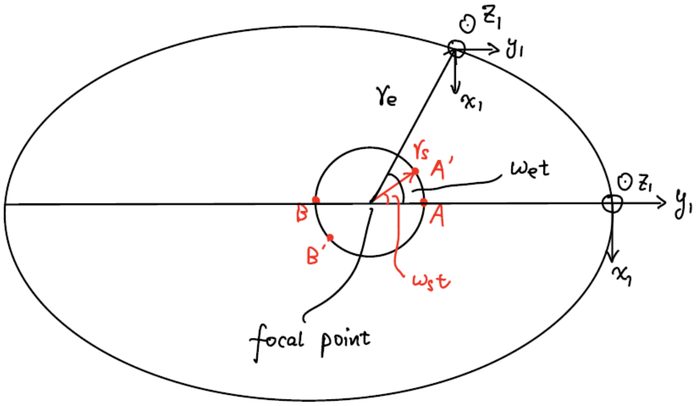


Figure 2: S_1 coordinate system.

Second, we transform (x_1, y_1, z_1) into (x_2, y_2, z_2). As we mentioned, S_2 and S_1 are related by the Euler angles $(\frac{\pi}{2}, \xi_0, \gamma_0)$ so that we have

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = R^{-1}(\frac{\pi}{2}, \xi_0, \gamma_0) \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \tag{2}$$

where the rotation matrix is given by

$$R^{-1}(\alpha, \beta, \gamma) = \begin{pmatrix} \cos(\alpha)\cos(\beta)\cos(\gamma) - \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\gamma) + \sin(\alpha)\cos(\beta)\cos(\gamma) & -\sin(\beta)\cos(\gamma) \\ -\sin(\alpha)\cos(\gamma) - \cos(\alpha)\cos(\beta)\sin(\gamma) & \cos(\alpha)\cos(\gamma) - \sin(\alpha)\cos(\beta)\sin(\gamma) & \sin(\beta)\sin(\gamma) \\ \cos(\alpha)\sin(\beta) & \sin(\alpha)\sin(\beta) & \cos(\beta) \end{pmatrix} \tag{3}$$

And the transformation is given by

$$\begin{aligned} x_2 &= \cos\xi_0 \cos(\gamma_0 + \Omega t)\, x_1 + \sin(\gamma_0 + \Omega t)\, y_1 \\ y_2 &= -\cos\beta_0 \sin(\gamma_0 + \Omega t)\, x_1 + \cos(\gamma_0 + \Omega t)\, y_1 \\ z_2 &= \sin\xi_0 x_1 \end{aligned} \tag{4}$$

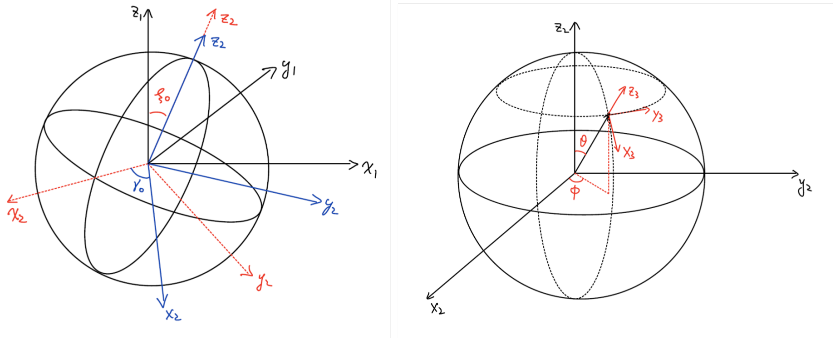


Figure 3: Left: S_2 coordinate system. Right: S_3 coordinate system.

Finally, we transform (x_2, y_2, z_2) into (x_3, y_3, z_3). As we mentioned, S_3 and S_2 are related by the Euler angles $(\phi, \theta, 0)$ so that we have

$$\begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} = R^{-1}(\phi, \theta, 0) \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} \tag{5}$$

so that the transformation is given by

$$
\begin{aligned}
x_3 = &\cos\phi\cos\theta\Big[\cos\xi_0\cos\left(\gamma_0+\Omega t\right)x_1+\sin\left(\gamma_0+\Omega t\right)y_1\Big]\\
&+\sin\phi\cos\theta\Big[-\cos\beta_0\sin\left(\gamma_0+\Omega t\right)x_1+\cos\left(\gamma_0+\Omega t\right)y_1\Big]\\
&-\sin\theta\sin\xi_0 x_1\\
y_3 = &-\sin\phi\Big[\cos\xi_0\cos\left(\gamma_0+\Omega t\right)x_1+\sin\left(\gamma_0+\Omega t\right)y_1\Big]\\
&+\cos\phi\Big[-\cos\beta_0\sin\left[\gamma_0+\Omega t\right)x_1+\cos\left(\gamma_0+\Omega t\right)y_1\Big]\\
z_3 = &\cos\phi\sin\theta\Big[\cos\xi_0\cos\left(\gamma_0+\Omega t\right)x_1+\sin\left(\gamma_0+\Omega t\right)y_1\Big]\\
&+\sin\phi\sin\theta\Big[-\cos\beta_0\sin\left(\gamma_0+\Omega t\right)x_1+\cos\left(\gamma_0+\Omega t\right)y_1\Big]\\
&+\cos\theta\sin\beta_0 x_1-R_e
\end{aligned}
\tag{6}
$$

and the altitude and azimuth are simply given by

$$
\begin{aligned}
\text{altitude} &= \arctan\Big(\frac{z_3}{\sqrt{x_3^2+y_3^2}}\Big)\\
\text{azimuth} &= 180^\circ-\arctan\Big(\frac{y}{x}\Big)
\end{aligned}
\tag{7}
$$

Here $arctan$ should consider quadrant properly. Note that in theory, one can generate the azimuth and the altitude data in any arbitrary location on Earth which paves the way for further predicting when and where on Earth can we observe both solar and lunar eclipses.

2.1.3 Data verification

The above mentioned two completely different methods for generating data yield almost identical results. Figure 4 and Figure 5 show the comparison between the azimuth and the altitude data of a sun measured in Berkeley generated by the two different methods. Notice that the number of orange points is much smaller than that of the blue points since the consumption of generating data by using SpaceEngine is much higher than that by using analytical model.

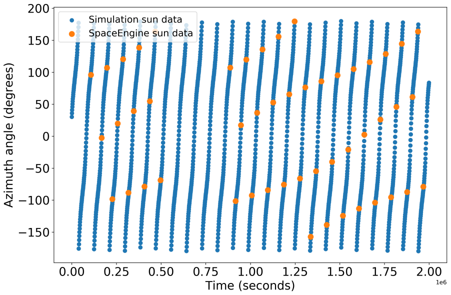


Figure 4: Azimuth data comparison

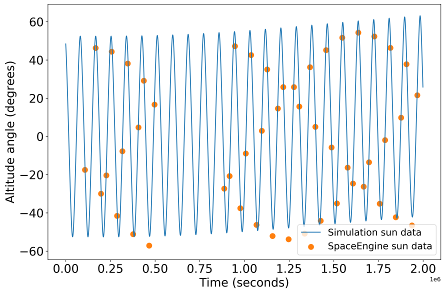


Figure 5: Altitude data comparison

The definition of the altitude in SpaceEngine is different from the traditional topocentric coordinate system. The altitude in SpaceEngine ranges from -180° to 180° and 0° is defined to point to the South pole, while the traditional topocentric coordinate system ranges from 0° to 360° and it starts from the North pole. We decided to follow the traditional topocentric coordinate system definition in the following training part.

2.2 Physical reasoning of the characteristic frequency

From the time evolution plot of the altitude, we can notice that this quantity is periodic. In order to get a sense of what features are relevant in learning such 1-d curve, we firstly use matlab to do a rough fitting of the time evolution curve using a physically reasonable function given by:

$$f(t) = A_1 \sin(\omega_1 t + \phi_1) + A_2 \sin(\omega_2 t + \phi_2) \tag{8}$$

The fitting result for moon is shown in Figure 6. It is obvious from the fitting result that there are indeed two main characteristic frequencies underlying the data and the simple formula in Equation 8 captures most of the behaviors of the altitude data.

Physically speaking, there are three physically important frequencies in our system related to the sun's and the moon's position on the sky, that is the self rotation and revolution of the Earth and the revolution of the moon which are given by

$$f_s = \frac{1}{T_s}, \quad f_r = \frac{1}{T_r}, \quad f_m = \frac{1}{T_m} \tag{9}$$

Actually, the two frequencies appearing in our simple fitting for sun's altitude are exactly corresponding to the self-rotation and the revolution of the Earth in our binary star solar system which is physically reasonable (we did not show the plot for sun's altitude here due to 8 pages limit). The two angular velocities in Equation 8 are thus given by

$$\omega_1 = 2\pi f_s$$

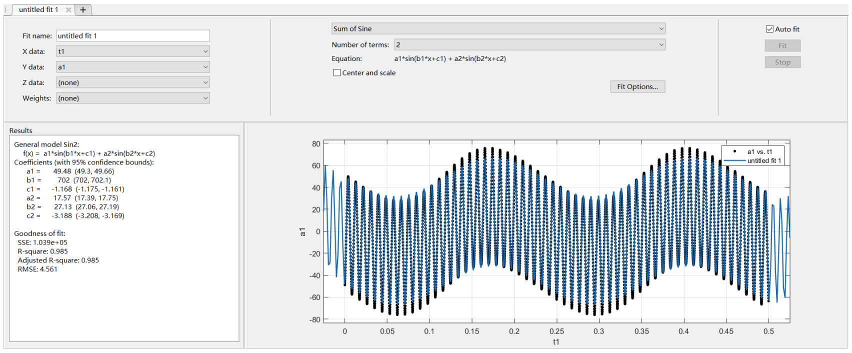
$$\omega_2 = 2\pi f_r$$

As for moon's altitude, one of the two frequencies appearing in the fitting is exactly corresponding to the revolution of the moon and the other one is roughly corresponding to the difference between self-rotation of the Earth and the revolution of the moon. We consider the latter frequency as a further beat phenomenon where the self-rotation frequency is further modulated by the evolution frequency of the moon since f_m is not negligible comparing to f_s. The two angular velocities are thus given by

$$\omega_1 \approx 2\pi(f_s - f_m)$$

$$\omega_2 = 2\pi f_m$$

Therefore, it is reasonable to use the combination of the Fourier features $\sin \omega_1 t$, $\cos \omega_1 t$, $\sin \omega_2 t$, and $\cos \omega_2 t$ to learn such 1-d function. And the more accurate values for those frequencies can be further tuned by k-fold cross validation.



Fit name: untitled fit 1
X data: t1
Y data: a1
Z data: (none)
Weights: (none)

Sum of Sine
Number of terms: 2
Equation: a1*sin(b1*x+c1) + a2*sin(b2*x+c2)
□ Center and scale
☑ Auto fit

Results

```
General model Sin2:
   f(x) =  a1*sin(b1*x+c1) + a2*sin(b2*x+c2)
Coefficients (with 95% confidence bounds):
     a1 =      49.48  (49.3, 49.66)
     b1 =        702  (702, 702.1)
     c1 =     -1.168  (-1.175, -1.161)
     a2 =      17.57  (17.39, 17.75)
     b2 =      27.13  (27.06, 27.19)
     c2 =     -3.188  (-3.208, -3.169)

Goodness of fit:
  SSE: 1.039e+05
  R-square: 0.985
  Adjusted R-square: 0.985
  RMSE: 4.561
```

Figure 6: Matlab fitting result of the altitude time evolution curve of the moon.

2.3 Features for learning time evolution curves

In our learning process for the altitude data, we choose "polynomial-type" combination of the Fourier features mentioned above as our features in order to capture any additional "nonlinear correlation effect" in the sense of Fourier features. That is, for example, we use the following kind of features for learning in the degree $= 2$ case:

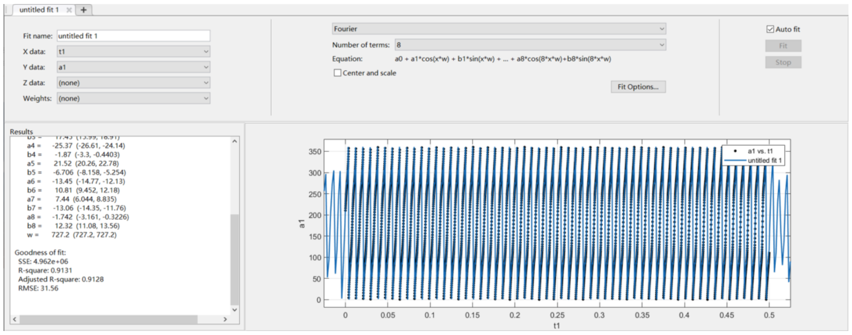
$$1, \sin(\omega_1 t), \cos(\omega_1 t), \sin(\omega_2 t), \cos(\omega_2 t), \sin^2(\omega_1 t), \cos^2(\omega_1 t), \sin^2(\omega_2 t), \cos^2(\omega_2 t)$$

$$\sin(\omega_1 t)\cos(\omega_1 t), \sin(\omega_1 t)\sin(\omega_2 t), \sin(\omega_1 t)\cos(\omega_2 t), \cos(\omega_1 t)\sin(\omega_2 t)$$

$$\sin(\omega_1 t)\cos(\omega_2 t), \sin(\omega_2 t)\cos(\omega_2 t)$$

It is clearly that the degree parameter D plays a role as a hyperparameter in our training process which can take values larger than 1. Considering the number of polynomial features scales roughly as D^3 which increases very fast, we focus on using D in range 2 to 5. Through our training, we pick up the optimal hyperparameter $D = 3$ considering both performance and the time consumption.

In our learning process for the azimuth data, we simply choose Fourier features. In this case, there are two hyperparameters: the base frequency ω and the number of Fourier features N to use. Therefore, we did k-fold cross validation for hyperparameter tuning. But before that, it is useful to firstly give a rough estimation of the range of the hyperparameter. So, we firstly use matlab to estimate the value of the base frequency and the fitting result is shown in Figure 7. After this, we use a much narrower and precise range for the hyperparameter tuning and the tuning result for the sun is shown in Figure 8.



Fit name: untitled fit 1
X data: t1
Y data: a1
Z data: (none)
Weights: (none)

Fourier
Number of terms: 8
Equation: a0 + a1*cos(x*w) + b1*sin(x*w) + ... + a8*cos(8*x*w)+b8*sin(8*x*w)
☐ Center and scale
☑ Auto fit

```
  b3 =    17.45  (15.99, 18.91)
  a4 =   -25.37  (-26.61, -24.14)
  b4 =    -1.87  (-3.3, -0.4403)
  a5 =    21.52  (20.26, 22.78)
  b5 =   -6.706  (-8.158, -5.254)
  a6 =   -13.45  (-14.77, -12.13)
  b6 =    10.81  (9.452, 12.18)
  a7 =     7.44  (6.044, 8.835)
  b7 =   -13.06  (-14.35, -11.76)
  a8 =   -1.742  (-3.161, -0.3226)
  b8 =    12.32  (11.08, 13.56)
  w =     727.2  (727.2, 727.2)

Goodness of fit:
  SSE: 4.962e+06
  R-square: 0.9131
  Adjusted R-square: 0.9128
  RMSE: 31.56
```

Figure 7: Matlab fitting result of the azimuth time evolution curve of the sun.

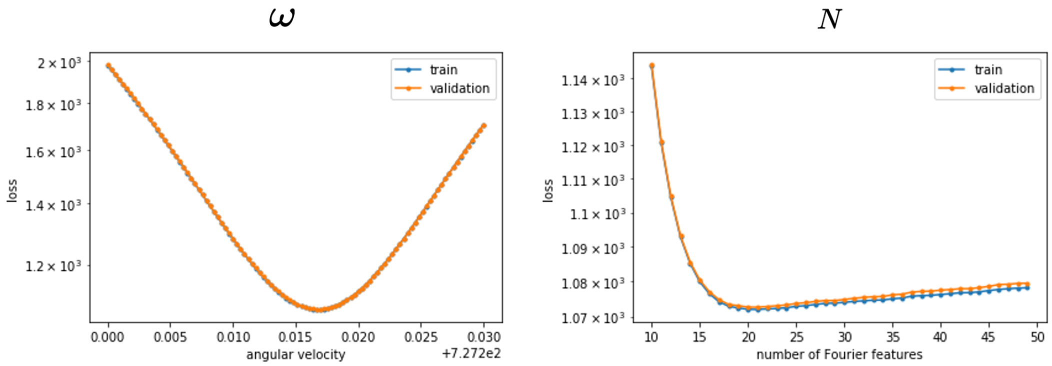


Figure 8: Left: cross validation loss versus ω. Right: cross validation loss versus N.

3 Results

Our learning results for the altitude are shown in Figure 9. The mean squared error(MSE) of the test data for the sun's altitude is 9.4 which means that our prediction only deviates from the true value by around $\sqrt{9.4} \approx 3$ degrees on average which is very accurate. The MSE of the test data for the moon's altitude is 27.6 which means that our prediction deviates from the true value by around $\sqrt{27.6} \approx 5$ degrees on average which is also very accurate.

Our learning results for the azimuth are shown in Figure 10. The MSE of the test data is in the order of 10^3 due to the rooted discontinuous behavior of the data at the turning point of $0°$ and $360°$ which is clearly shown in Figure 10. Apart from that, the fitting in the other region is quite accurate.

Our learning result for the moon phase is shown in Figure 11. The MSE of the test data for the moon phase is 0.002 which is clearly very accurate.

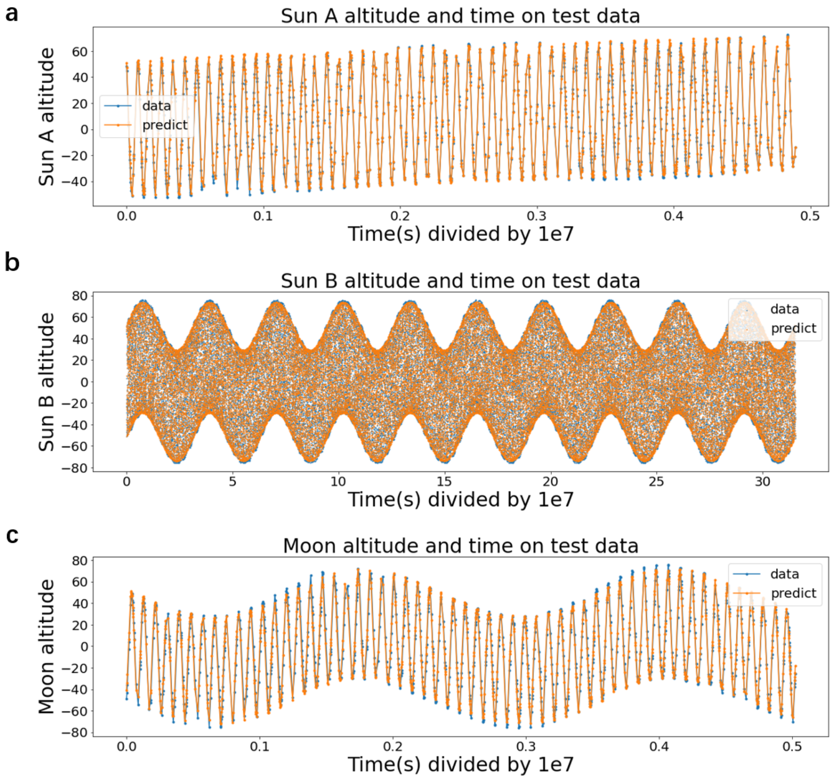


Figure 9: (a) Learning result of the altitude time evolution curve of Sun A. (b) Learning result of the altitude time evolution curve of Sun B. (c) Learning result of the altitude time evolution curve of the moon. Orange dots are the prediction and blue dots are the original data. Notice that plot in (a) and (c) is in a shorter time period than that in (b) in order for clarity.

4 Conclusion

1. There are three main physically important characteristic frequencies in our system which are corresponding to the self-rotation of the Earth, the revolution of the Earth, and the revolution of the moon.
2. The time evolution of the altitude and azimuth of the two suns and the moon are periodic and the periodicity is captured by the three characteristic frequencies mentioned above.
3. The azimuth curve of the sun basically has the period of one day since it will always change from 0 to 360 within one day.
4. Similarly, the azimuth curve of the moon also has the period of one day since it will also change from 0 to 360 within one day.
5. The altitude curve of the sun is basically a combination of two sinusoidal curves with two frequencies corresponding to the self-rotation of the Earth and the revolution of the Earth. This is physically reasonable.
6. The altitude curve of the moon is basically a combination of two sinusoidal curves with two frequencies corresponding to the self-rotation of the Earth and the revolution of the moon. This is physically reasonable.
7. The phase of the moon has the periodicity of around one month which is around 29.5 days.

References

[1] https://www.wikipedia.org/

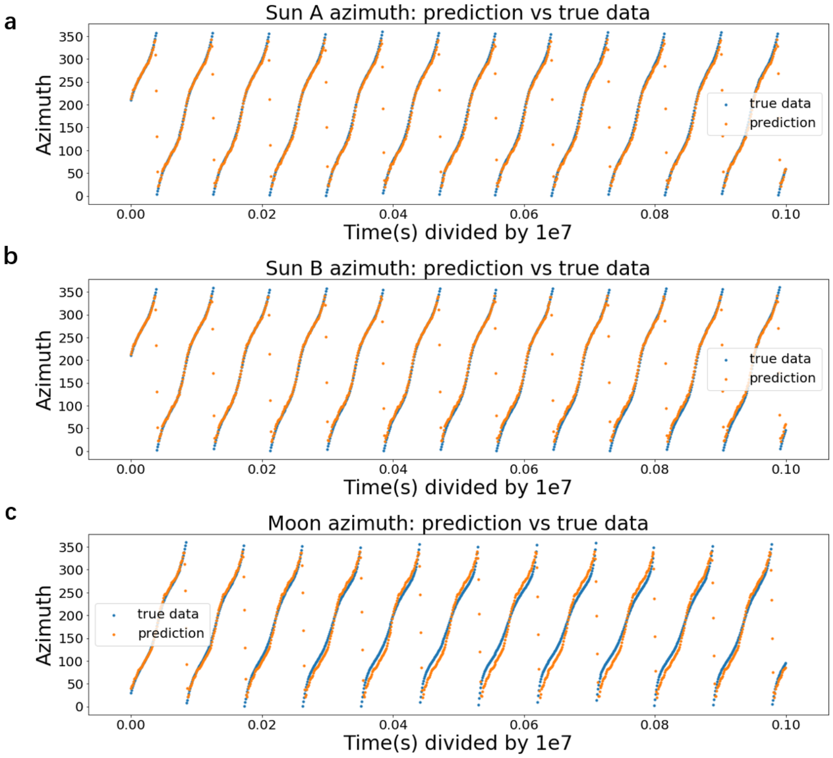


Figure 10: (a) Learning result of the azimuth time evolution curve of Sun A. (b) Learning result of the azimuth time evolution curve of Sun B. (c) Learning result of the azimuth time evolution curve of the moon. Orange dots are the prediction and blue dots are the original data. Notice that all plots are in a shorter time period in order for clarity.

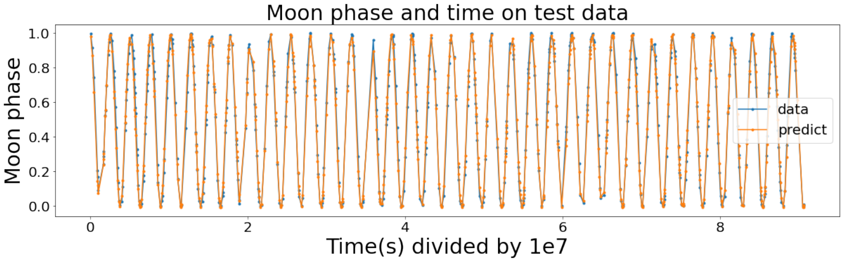


Figure 11: Learning result of the phase time evolution curve of the moon. The curve shows performance on test dataset. Orange curve is the prediction on test data and blue curve is the original test data. Notice that the plot is shown in a shorter time period in order for clarity.