

# Satellite project, AST 2000

## 5 Part 5: Launching the satellite mission

Having now written software making it possible for the satellite to orient itself, we will launch the satellite. You have (hopefully) simulated the planned trajectory and prepared in detail for every eventuality. Now your task is to ensure the actual satellite mission goes according to the plan. Before proceeding, you should now have calculated the total fuel consume expected to reach your destination and reported this number to your group teacher.

### 5.1 Launching the actual satellite

You now have a satisfactory simulated satellite trajectory. Launching the satellite “for real” is done using the `AST2000SolarSystem.send_satellite` method, which takes as input a text file containing instructions.

Due to various factors (such as imprecise knowledge of planet positions, masses and even unknown objects in the solar system) and because of imprecise numerical methods used for the satellite calculation, the actual trajectory will probably deviate from your prediction. However, since we have fitted cameras to the satellite, we can use these mid-mission to orient it and compute its position relative to the planets in the system. The approach you should use, which is illustrated in Fig. 1, is the following:

- (i) Launch the satellite with the same initial conditions as in your simulation.
- (iii) At some point later (e.g. just before your first planned boost), orient the satellite using the software you developed in part 4 (the first time you use it, you will need the instructions given in section 5.2 at the end here). In this way you obtain the actual position and velocity of the satellite before the boost is performed.
- (iv) You will see that there is a (hopefully) small deviation from your calculated satellite position and velocity at this point. You therefore need to go back to your calculations of the satellite trajectory and change the position and velocity just before the boost with the actual position and velocity. Rerun your calculation from this point on. Determine based on this whether changes are needed in the upcoming boost in order to achieve the planned close encounter with the target planet.
- (v) Repeat steps (ii) to (iv) until you, hopefully, have a satellite–planet rendezvous.

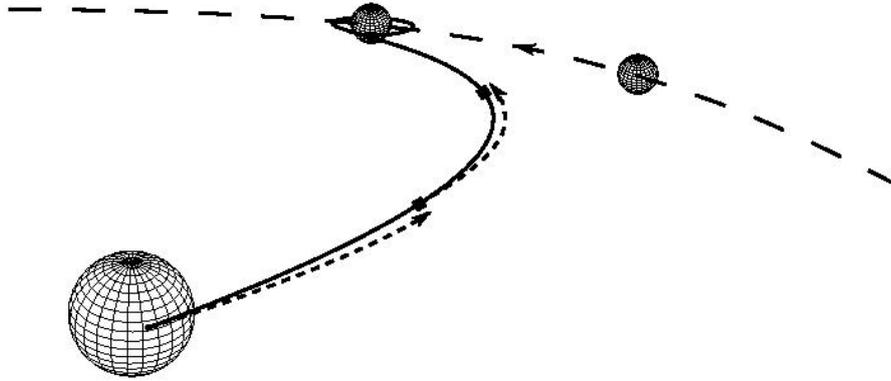


Figure 1: Simulated (dotted) and actual trajectory of the satellite. At the points in the real trajectory marked by black squares, pictures are taken and analyzed in an effort to orientate the satellite in space. Since this gives us the exact position of the satellite, we may next perform a simulation from this point on to the next time we perform the orientation operation. Please note that your trajectory and resulting orbit may look (a lot) more complicated.

Along the trajectory you should take videos in the direction of the home planet or target planet at appropriate intervals. For the launch, note that the position you specify as the launch position needs to be a point at the surface of your home planet. If you have made a mistake in calculating the position of the surface of the planet and try to launch away from the surface, the program will terminate and tell you that you are too far away, as well as the maximum allowed distance. If you are sufficiently close to the surface, the program will move the rocket to the nearest point exactly on the surface, and let you launch from there.

**Exercise 5.1.1:** We denote the resolution of the satellite camera by  $P$  pixels  $\times$   $P$  pixels, and its field of view by  $F^\circ \times F^\circ$ . We let  $R$  denote the radius of the target planet. Show that the distance from the planet,  $L$ , must be

$$L \lesssim \frac{RP}{F}$$

in order for it to show up as more than a single pixel in the image. For our satellite camera,  $P \approx 1000$ , while  $F = 70^\circ$ . (Hint: Compute first the angular size of a single pixel in the image, then use the small-angle approximation for  $\tan \theta$  to express  $\theta$  in terms of  $R$  and  $L$ .)

**Hints :**

- Before you are allowed to call `send_satellite`, you must in the same instance (i.e. in the same script) already have a successful test-launch. You can use the one you calculated in part 1.
- Furthermore, the `engine_settings` method must be called, possibly again if you want another launch time or initial satellite position than what your `mass_needed_launch` test used.

- The `engine_settings` method takes the following arguments: (1)  $dp/dt$  per box [N s], (2) number of boxes, (3) particles per second, (4) total fuel you want to bring [kg], (5) an estimate of the launching time (add a couple of seconds compared to your simulation!) [s], (6) initial satellite position on surface [shape 2 array of x-y positions in AU], (7) starting time of launch [years].
- The `send_satellite` method takes as input the file name of a text-file containing instructions. The possible instructions in the text file are
  - (i) `launch` is needed at the start of the command file. It takes either zero arguments or two arguments. With two arguments, there is given an extra boost directly after reaching escape velocity. (1) and (2) are the x- and y-velocity [AU/yr]
  - (ii) `boost` takes the following arguments: (1) The time for the boost [years]; (2, 3) change of velocity vector,  $\Delta v_x$  [AU/Yr] and  $\Delta v_y$  [AU/Yr].
  - (iii) `orient` : A time (1) is needed [years]. The satellite will tell you it's exact position and velocity. The first time this is used, you will need to orient it manually.
  - (iv) `video` : There are 2 ways of creating a video. You can either (A) make the camera focus on a planet of your choice (it's important to choose the right planet since a planet very far away will not be visible). For this option 2 lines are needed. First [video  $time_{start}$  planetToWatch] and on another line [video  $time_{end}$  planetToWatch] Or you can (B) choose starting angles and end angles and make the camera turn as you please. The command for this option is [video  $time_1$   $\theta_1$   $\phi_1$ ] and then on another line [video  $time_2$   $\theta_1$   $\phi_2$ ]. Only one video can be created for one command file. The video will be output as an xml file to be uploaded by MCAst.

The commands in the input file must be ordered chronologically. Remember that the launch takes some time, so don't give the satellite new instructions before the launch is complete. Furthermore, at most two events can be at the exact same time (but of course you can separate two events by e.g.  $10^{-7}$  years). An example input file might look like this (please note that this is an *example* and you will most likely have to use more precision (more significant digits) in an *actual* command file):

1	<code>launch</code>			
2	<code>orient</code>	<code>0.6</code>		
3	<code>boost</code>	<code>0.7</code>	<code>0.5</code>	<code>0.3</code>
4	<code>video</code>	<code>3.0</code>	<code>1</code>	
5	<code>video</code>	<code>3.1</code>	<code>1</code>	
6	<code>boost</code>	<code>3.3</code>	<code>-0.1</code>	<code>-0.1</code>

where

2	<code>orient</code>	<code>0.6</code>
---	---------------------	------------------

means we are performing an orientation at  $t = 0.6$  Yr. The orientation is either a manual one, or an automatic orientation depending on whether or

not you have already tested and verified your manual orientation software. The next line,

```
3 boost      0.7      0.5      0.3
```

means at  $t = 0.7$  yr, change the velocity by  $\Delta\mathbf{v} = (0.5, 0.3)$  AU/Yr. Using the same engine as during launch, defined by engine settings. The next lines after this,

```
4 video      3.0      1
5 video      3.1      1
```

is the first video option where your camera is focused on planet 1. The video commands for starting and ending the video do not need to be consecutive.

```
6 video      2.6      3.14     1.57
7 video      2.7      2.14     1.57
```

is the second video option where the camera rotates between  $\theta_1 = 3.14 \rightarrow \theta_2 = 2.14$  and from  $\phi_1 = 1.57 \rightarrow \phi_2 = 1.57$ .

## 5.2 Testing the orientation software

The first time you use the `orient` command you have to manually do the orientation in order to make sure that your code and methods work. Following the first orientation command, the manual testing procedure goes as follows:

- You will receive a .png file, which is a picture taken in the forward direction of the satellite's orientation. You will then be prompted to input the corresponding angle [degrees]. Use your orientation software to find this angle. Input and push enter.
- If you successfully found the angle, you will receive the measured  $\Delta\lambda$  of the two reference stars. Use this data to estimate the velocity of the satellite. Provide these velocities when you are prompted (x and y separately, in [AU/year]).
- If you have found correct velocities you will be given a file with the distances to the star and all the planets. Use this information to compute the position of the satellite and provide this when prompted. (x and y positions separately, in [AU])
- If all of these steps are completed successfully you have proved that your orientation software works, thus you can safely just ask the satellite for the position and velocity when you need it for simulation.

## 5.3 Goals of this part

Your first task is to launch the rocket successfully. Feel free to make a take-off video (direct the camera towards planet 0) starting at the end of the launch time and ending some hours later. Unfortunately, the cameras don't work until escape velocity is reached and the engines are turned off.

Your main goal is to create a command file which takes your satellite all the way to the last boost which is the orbital injection maneuver at your destination planet. Compare the position found from orientating the satellite with the calculated position of your target planet at the time of the last boost. Does the distance seem reasonable? Use the orient command to check if the satellite is still in orbit some years later. If not, you will need to send a new satellite... That will be expensive!

#### 5.4 Additional exercise compulsory for those working in groups

Just after performing the last boost which takes you into orbit around the planet, you can do orient to find your distance  $r$  from the center of the planet as well as to find your velocity components  $v_r$  and  $v_\theta$  with respect to the center of the planet. Assuming 2-body system, use these three numbers to find the semimajor axis, semiminor axis, eccentricity, period, aphelion and perihelion of your orbit around the planet. **Hint: In part1B, you can find analytical expressions for  $r$  as well as for  $v$ ,  $v_\theta$ ,  $v_r$  and also for the total energy  $E$ , remember that  $\vec{r} = r\vec{e}_r$  and  $\vec{v} = \dot{\vec{r}}$ .** Having found these properties of your orbit, let your satellite do a full orbit around the planet, do orient a few times and use these observations to confirm that you calculated the correct semimajor axis and orbital period.