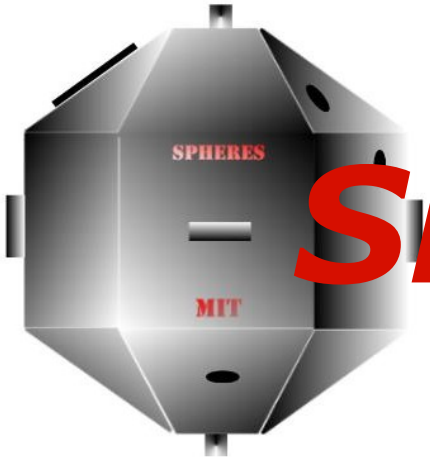


SPHERES

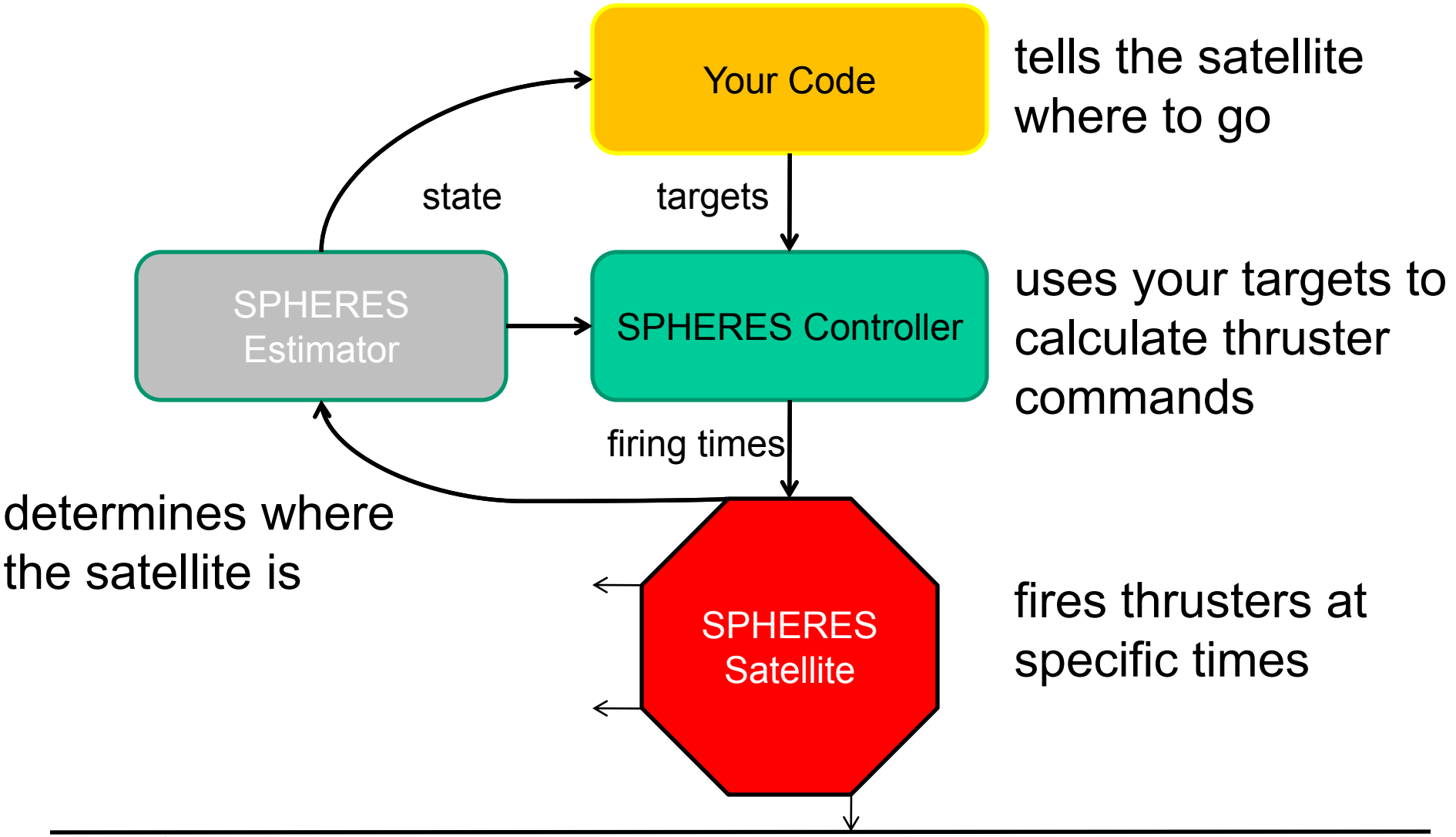
ZR API Overview

API Functions

This document will explain the Zero Robotics Application Programming Interface. The following is a list of functions that are automatically included by the simulation, and can be called by any user function.

- ZRUser – the main user code loop
- ZRSetForces – apply a force to the satellite
- ZRSetPositionTarget – set a position for the satellite to move to
- ZRSetVelocityTarget – set the velocity for the satellite to move by
- ZRSetAttitudeTarget – set the unit vector for the satellite to point along

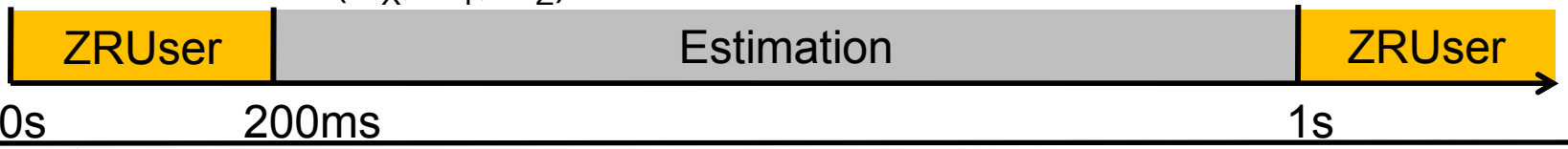
ZERO Robotics Control System



ZRUser

```
void ZRUser(float *myState,
            float *otherState,
            float time);
```

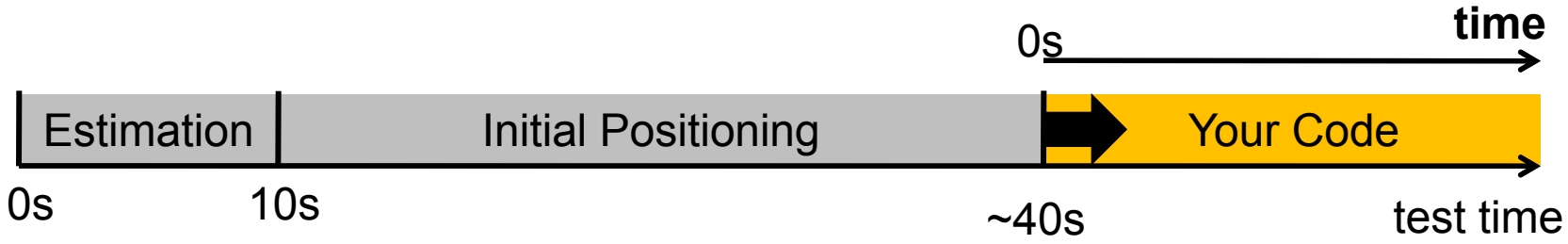
- This is the outer function where all user code is implemented and calls to other API functions are made. It is called **once per second** by the SPHERES control system. The thrusters are active for 200ms.
- 3 Inputs:
  - **myState[12]** the State of your satellite
  - **otherState[12]** the State of the other satellite
  - **time** the elapsed time since your code was activated
- The State arrays contain, in order:
  - Position (x, y, z)
  - Velocity (v_X, v_Y, v_Z)
  - Attitude Vector (n_X, n_Y, n_Z)
  - Attitude rate (ω_X, ω_Y, ω_Z)



float time

- Use **time** to keep track of the time elapsed since your code was activated
- In the Zero Robotics simulation, your code is activated immediately

- On the ISS, your code will be activated after a few brief maneuvers to position the satellites

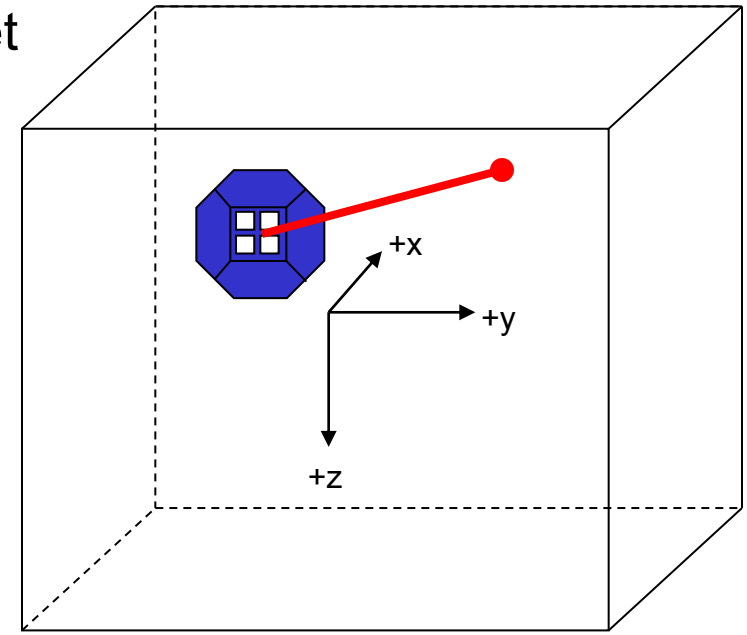


ZRSetPositionTarget

```
void ZRSetPositionTarget(float posTarget[3]);
```

- Input:
  - **posTarget[3]** {x, y, z} Allows you to set the x, y, and z position targets for the satellite
  - Units are in meters
- Cannot be combined with velocity target

Position Target in x, y, and z

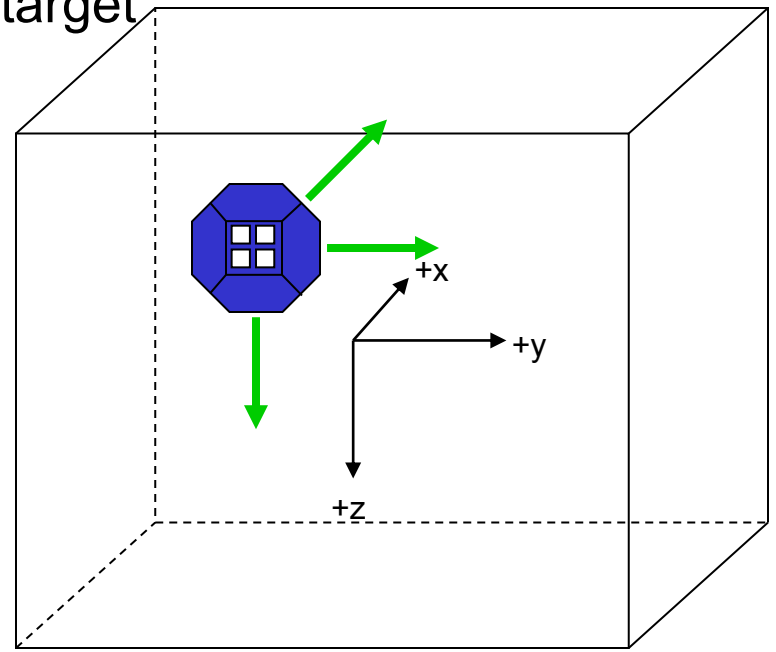


ZRSetVelocityTarget

```
void ZRSetVelocityTarget(float velTarget[3]);
```

- Input:
  - **velTarget[3]** {v_X, v_Y, v_Z} sets the x, y, and z target velocities
  - Units are in meters per second
- Cannot be combined with Position target

Target velocities in v_X, v_Y, and v_Z

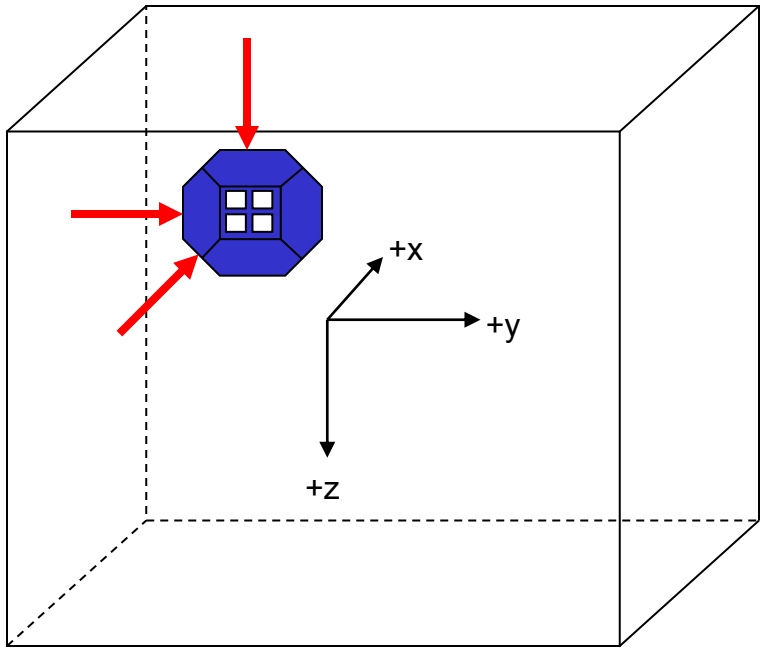


ZRSetForces

void ZRSetForces(float forces[3]);

- Input:
  - **forces[3]** $\{f_x, f_y, f_z\}$ Applies global frame x, y, and z forces to the satellite
  - Units are in Newtons
- May be combined with ZRSetPositionTarget or ZRSetVelocity Target

Forces applied in x, y, and z



ZRSetAttitiudeTarget

```
void ZRSetAttitudeTarget(float attTarget[3]);
```

- 1 Input

  attTarget[3] $\{n_X, n_Y, n_Z\}$ sets the unit vector for the satellite to point its Velcro side towards

- This allows you to point the satellite along a specific vector.

- For more information on setting Attitude, see the Rotate Tutorial.