Quadcopter_Project

June 28, 2019

1 Project: Train a Quadcopter How to Fly

Design an agent to fly a quadcopter, and then train it using a reinforcement learning algorithm of your choice!

Try to apply the techniques you have learnt, but also feel free to come up with innovative ideas and test them.

1.1 Instructions

Take a look at the files in the directory to better understand the structure of the project.

- `task.py`: Define your task (environment) in this file.
- `agents/`: Folder containing reinforcement learning agents.
  - `policy_search.py`: A sample agent has been provided here.
  - `agent.py`: Develop your agent here.
- `physics_sim.py`: This file contains the simulator for the quadcopter. **DO NOT MODIFY THIS FILE**.

For this project, you will define your own task in `task.py`. Although we have provided a example task to get you started, you are encouraged to change it. Later in this notebook, you will learn more about how to amend this file.

You will also design a reinforcement learning agent in `agent.py` to complete your chosen task.

You are welcome to create any additional files to help you to organize your code. For instance, you may find it useful to define a `model.py` file defining any needed neural network architectures.

1.2 Controlling the Quadcopter

We provide a sample agent in the code cell below to show you how to use the sim to control the quadcopter. This agent is even simpler than the sample agent that you'll examine (in `agents/policy_search.py`) later in this notebook!

The agent controls the quadcopter by setting the revolutions per second on each of its four rotors. The provided agent in the `Basic_Agent` class below always selects a random action for each of the four rotors. These four speeds are returned by the `act` method as a list of four floating-point numbers.

For this project, the agent that you will implement in `agents/agent.py` will have a far more intelligent method for selecting actions!

In [1]:
```
import random

class Basic_Agent():
    def __init__(self, task):
        self.task = task

    def act(self):
        new_thrust = random.gauss(450., 25.)
        return [new_thrust + random.gauss(0., 1.) for x in range(4)]
```

Run the code cell below to have the agent select actions to control the quadcopter.

Feel free to change the provided values of `runtime`, `init_pose`, `init_velocities`, and `init_angle_velocities` below to change the starting conditions of the quadcopter.

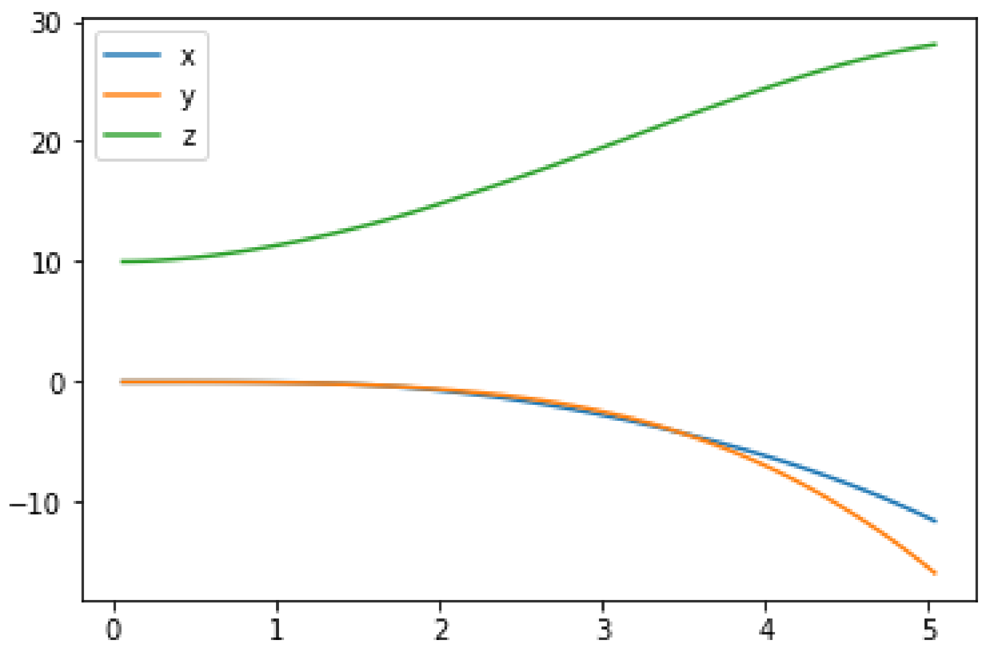
The `labels` list below annotates statistics that are saved while running the simulation. All of this information is saved in a text file `data.txt` and stored in the dictionary `results`.

In [2]:
```
#%load_ext autoreload
%reload_ext autoreload

#%autoreload 2

import csv
import numpy as np
from task import Task

# Modify the values below to give the quadcopter a different starting position.
runtime = 5.                                     # time limit of the episode
init_pose = np.array([0., 0., 10., 0., 0., 0.])  # initial pose
init_velocities = np.array([0., 0., 0.])         # initial velocities
init_angle_velocities = np.array([0., 0., 0.])   # initial angle velocities
file_output = 'data.txt'                         # file name for saved results

# Setup
task = Task(init_pose, init_velocities, init_angle_velocities, runtime)
agent = Basic_Agent(task)
done = False
labels = ['time', 'x', 'y', 'z', 'phi', 'theta', 'psi', 'x_velocity',
          'y_velocity', 'z_velocity', 'phi_velocity', 'theta_velocity',
          'psi_velocity', 'rotor_speed1', 'rotor_speed2', 'rotor_speed3', 'rotor_speed4
results = {x : [] for x in labels}

# Run the simulation, and save the results.
with open(file_output, 'w') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(labels)
    while True:
        rotor_speeds = agent.act()
        _, _, done = task.step(rotor_speeds)
```

```
    to_write = [task.sim.time] + list(task.sim.pose) + list(task.sim.v) + list(tas
    for ii in range(len(labels)):
        results[labels[ii]].append(to_write[ii])
    writer.writerow(to_write)
    if done:
        break
```

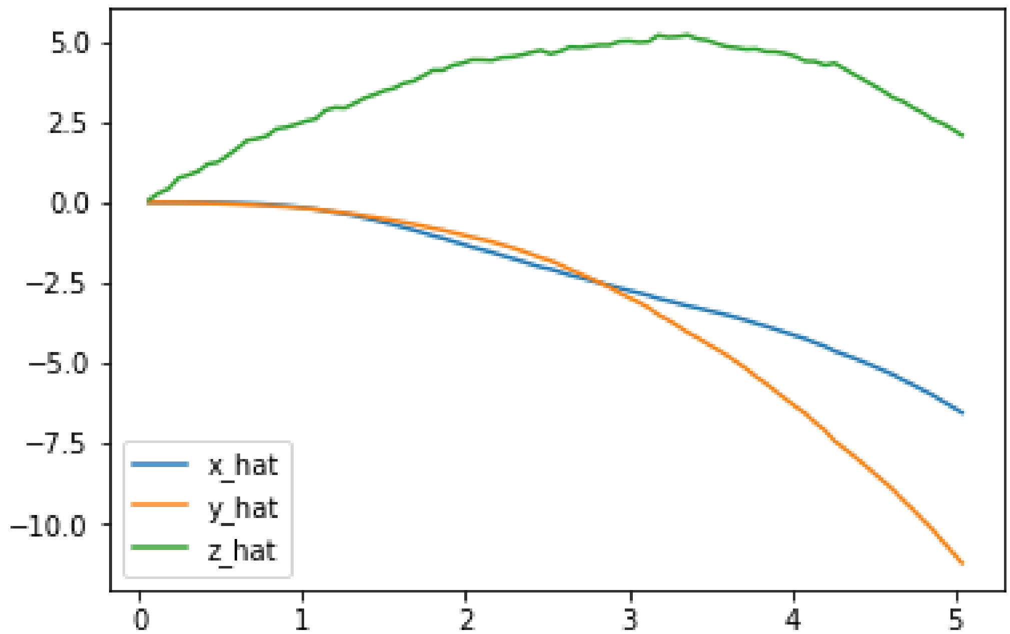
Run the code cell below to visualize how the position of the quadcopter evolved during the simulation.

In [3]:
```
import matplotlib.pyplot as plt
%matplotlib inline

plt.plot(results['time'], results['x'], label='x')
plt.plot(results['time'], results['y'], label='y')
plt.plot(results['time'], results['z'], label='z')
plt.legend()
_ = plt.ylim()
```



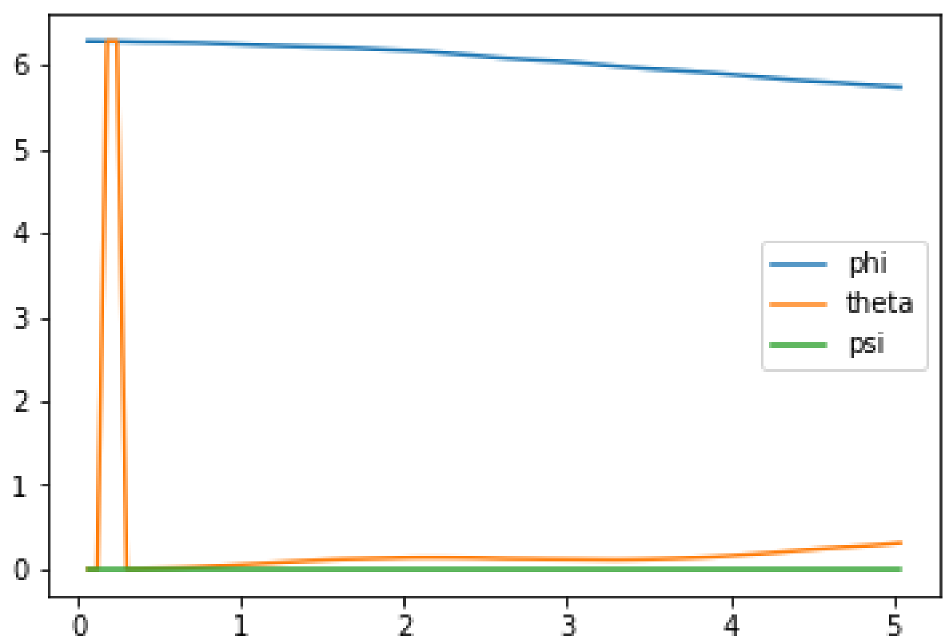
The next code cell visualizes the velocity of the quadcopter.

In [4]:
```
plt.plot(results['time'], results['x_velocity'], label='x_hat')
plt.plot(results['time'], results['y_velocity'], label='y_hat')
plt.plot(results['time'], results['z_velocity'], label='z_hat')
plt.legend()
_ = plt.ylim()
```



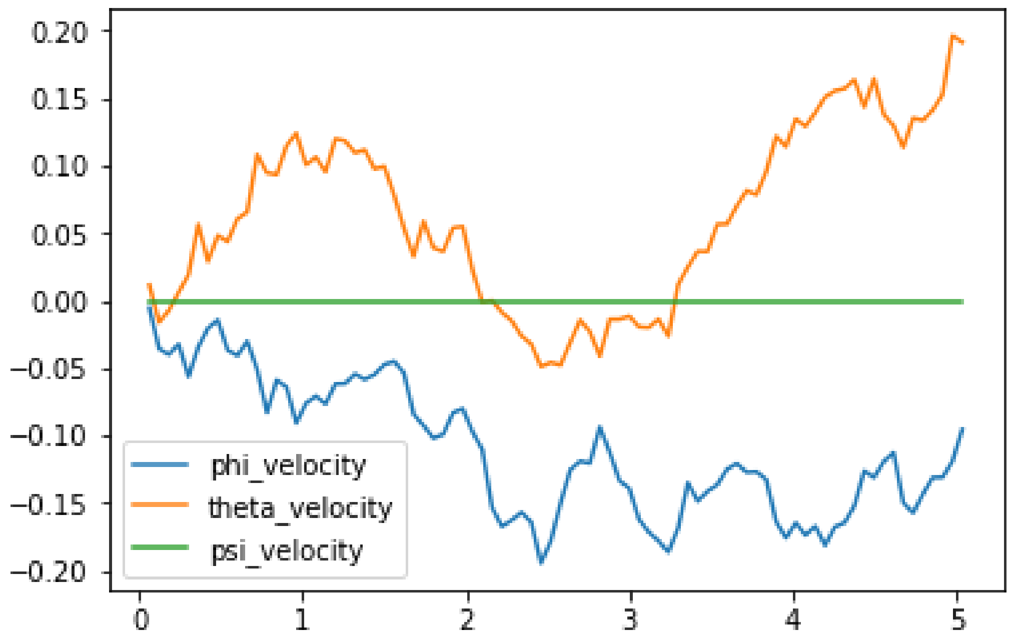
Next, you can plot the Euler angles (the rotation of the quadcopter over the *x*-, *y*-, and *z*-axes),

```
In [5]: plt.plot(results['time'], results['phi'], label='phi')
        plt.plot(results['time'], results['theta'], label='theta')
        plt.plot(results['time'], results['psi'], label='psi')
        plt.legend()
        _ = plt.ylim()
```



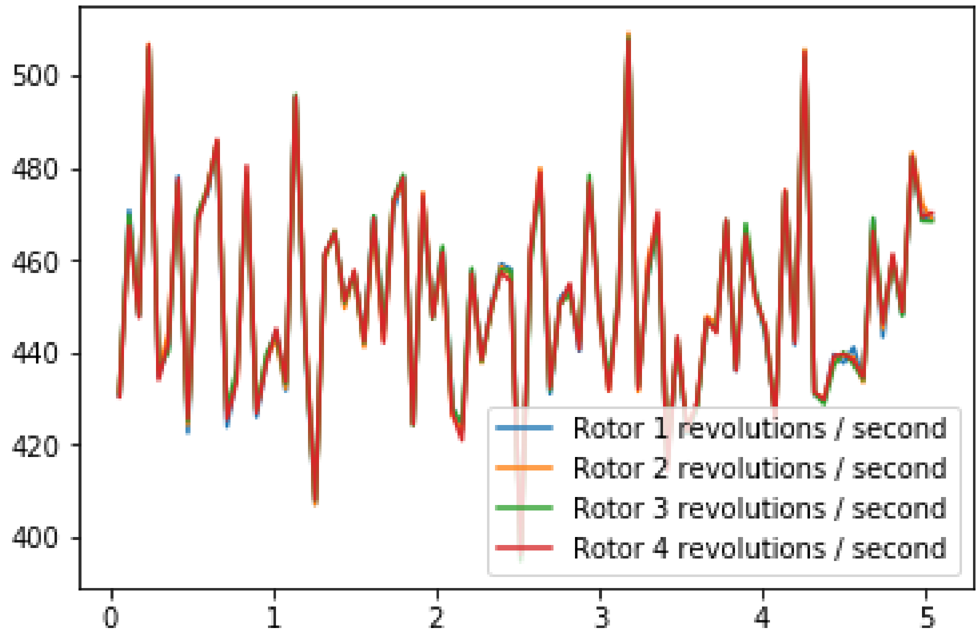
before plotting the velocities (in radians per second) corresponding to each of the Euler angles.

In [6]:
```
plt.plot(results['time'], results['phi_velocity'], label='phi_velocity')
plt.plot(results['time'], results['theta_velocity'], label='theta_velocity')
plt.plot(results['time'], results['psi_velocity'], label='psi_velocity')
plt.legend()
_ = plt.ylim()
```



Finally, you can use the code cell below to print the agent's choice of actions.

In [7]:
```
plt.plot(results['time'], results['rotor_speed1'], label='Rotor 1 revolutions / second
plt.plot(results['time'], results['rotor_speed2'], label='Rotor 2 revolutions / second
plt.plot(results['time'], results['rotor_speed3'], label='Rotor 3 revolutions / second
plt.plot(results['time'], results['rotor_speed4'], label='Rotor 4 revolutions / second
plt.legend()
_ = plt.ylim()
```



When specifying a task, you will derive the environment state from the simulator. Run the code cell below to print the values of the following variables at the end of the simulation: - `task.sim.pose` (the position of the quadcopter in (x, y, z) dimensions and the Euler angles), - `task.sim.v` (the velocity of the quadcopter in (x, y, z) dimensions), and - `task.sim.angular_v` (radians/second for each of the three Euler angles).

```
In [8]: # the pose, velocity, and angular velocity of the quadcopter at the end of the episode
        print(task.sim.pose)
        print(task.sim.v)
        print(task.sim.angular_v)

        a=np.concatenate((task.sim.pose, task.sim.v, task.sim.angular_v),axis=0)
        print(a)
```

```
[-11.56110441 -15.89105842  28.05815646   5.73470276   0.31012669
   0.        ]
[ -6.53786978 -11.21809548   2.08464382]
[-0.09565907  0.19124073  0.        ]
[-11.56110441 -15.89105842  28.05815646   5.73470276   0.31012669
   0.          -6.53786978 -11.21809548   2.08464382  -0.09565907
   0.19124073   0.        ]
```

In the sample task in `task.py`, we use the 6-dimensional pose of the quadcopter to construct the state of the environment at each timestep. However, when amending the task for your purposes, you are welcome to expand the size of the state vector by including the velocity information. You can use any combination of the pose, velocity, and angular velocity - feel free to tinker here, and construct the state to suit your task.

1.3 The Task

A sample task has been provided for you in `task.py`. Open this file in a new window now.

The `__init__()` method is used to initialize several variables that are needed to specify the task.
- The simulator is initialized as an instance of the `PhysicsSim` class (from `physics_sim.py`).
- Inspired by the methodology in the original DDPG paper, we make use of action repeats. For each timestep of the agent, we step the simulation `action_repeats` timesteps. If you are not familiar with action repeats, please read the **Results** section in the DDPG paper. - We set the number of elements in the state vector. For the sample task, we only work with the 6-dimensional pose information. To set the size of the state (`state_size`), we must take action repeats into account.
- The environment will always have a 4-dimensional action space, with one entry for each rotor (`action_size=4`). You can set the minimum (`action_low`) and maximum (`action_high`) values of each entry here. - The sample task in this provided file is for the agent to reach a target position. We specify that target position as a variable.

The `reset()` method resets the simulator. The agent should call this method every time the episode ends. You can see an example of this in the code cell below.

The `step()` method is perhaps the most important. It accepts the agent's choice of action `rotor_speeds`, which is used to prepare the next state to pass on to the agent. Then, the reward is computed from `get_reward()`. The episode is considered done if the time limit has been exceeded, or the quadcopter has travelled outside of the bounds of the simulation.

In the next section, you will learn how to test the performance of an agent on this task.

1.4 The Agent

The sample agent given in `agents/policy_search.py` uses a very simplistic linear policy to directly compute the action vector as a dot product of the state vector and a matrix of weights. Then, it randomly perturbs the parameters by adding some Gaussian noise, to produce a different policy. Based on the average reward obtained in each episode (`score`), it keeps track of the best set of parameters found so far, how the score is changing, and accordingly tweaks a scaling factor to widen or tighten the noise.

Run the code cell below to see how the agent performs on the sample task.

In [9]:
```
import sys
import pandas as pd
from agents.policy_search import PolicySearch_Agent
from task import Task
import numpy as np

num_episodes = 1000
target_pos = np.array([0., 0., 10.])
task = Task(target_pos=target_pos)
agent = PolicySearch_Agent(task)

for i_episode in range(1, num_episodes+1):
    state = agent.reset_episode() # start a new episode
    while True:
        action = agent.act(state)
        next_state, reward, done = task.step(action)
```

```
            agent.step(reward, done)
            state = next_state
            if done:
                print("\rEpisode = {:4d}, score = {:7.3f} (best = {:7.3f}), noise_scale = 
                    i_episode, agent.score, agent.best_score, agent.noise_scale), end="")
                break

        sys.stdout.flush()
```

```
Episode = 1000, score = 294.059 (best = 300.655), noise_scale = 3.25
```

In [10]:
```
reward
```

Out[10]:
```
32.93990952130259
```

This agent should perform very poorly on this task. And that's where you come in!

1.5 Define the Task, Design the Agent, and Train Your Agent!

Amend `task.py` to specify a task of your choosing. If you're unsure what kind of task to specify, you may like to teach your quadcopter to takeoff, hover in place, land softly, or reach a target pose.

After specifying your task, use the sample agent in `agents/policy_search.py` as a template to define your own agent in `agents/agent.py`. You can borrow whatever you need from the sample agent, including ideas on how you might modularize your code (using helper methods like `act()`, `learn()`, `reset_episode()`, etc.).

Note that it is **highly unlikely** that the first agent and task that you specify will learn well. You will likely have to tweak various hyperparameters and the reward function for your task until you arrive at reasonably good behavior.

As you develop your agent, it's important to keep an eye on how it's performing. Use the code above as inspiration to build in a mechanism to log/save the total rewards obtained in each episode to file. If the episode rewards are gradually increasing, this is an indication that your agent is learning.

In [11]:
```
## TODO: Train your agent here.
import sys
import numpy as np
from agents.agent import DDPGAgent
import pandas as pd
from task import Task

num_episodes = 500
target_pos = np.array([0., 0., 100.])
runtime = 10
task = Task(target_pos=target_pos, runtime = runtime)
agent = DDPGAgent(task)
score = 0.0
best_score = -100

labels = ['Episode', 'Score', 'Best Score']
```

```python
    results = {x : [] for x in labels}

    x_pos = []
    y_pos = []
    z_pos = []
    rewards = []

    for i_episode in range(1, num_episodes+1):
        state = agent.reset_episode() # start a new episode
        x_pos_temp = []
        y_pos_temp = []
        z_pos_temp = []

        while True:
            x_pos_temp.append(task.sim.pose[0])
            y_pos_temp.append(task.sim.pose[1])
            z_pos_temp.append(task.sim.pose[2])

            action = agent.act(state)
            next_state, reward, done = task.step(action)

            score += reward
            best_score = max(best_score, score)

            agent.step(action, reward, next_state, done)
            state = next_state

            if done:
                rewards.append(score)

                print("\rEpisode = {:4d}, score = {:7.3f} (best = {:7.3f}) position = {} 
                    i_episode, score, best_score,
                      round(task.sim.pose[:3][0],1),
                      round(task.sim.pose[:3][1],1),
                      round(task.sim.pose[:3][2],1), end=""))  # [debug]
                x_pos.append(x_pos_temp)
                y_pos.append(y_pos_temp)
                z_pos.append(z_pos_temp)
                to_write = [i_episode] + [score] + [best_score]
                for ii in range(len(labels)):
                    results[labels[ii]].append(to_write[ii])
                break
        sys.stdout.flush()
```

```
Using TensorFlow backend.
```

```
WARNING:tensorflow:From /Users/adrianlievano/anaconda3/envs/quadcop/lib/python3.6/site-package
```

```
Instructions for updating:
Colocations handled automatically by placer.
Episode =    1, score = -25.790 (best =  -1.859) position = -2.3 0.3 0.0
Episode =    2, score = -51.553 (best =  -1.859) position = -2.3 0.3 0.0
WARNING:tensorflow:From /Users/adrianlievano/anaconda3/envs/quadcop/lib/python3.6/site-packages
Instructions for updating:
Use tf.cast instead.
Episode =    3, score = -77.314 (best =  -1.859) position = -2.3 0.2 0.0
Episode =    4, score = -83.395 (best =  -1.859) position = -0.5 0.1 0.0
Episode =    5, score = -112.155 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =    6, score = -140.919 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =    7, score = -169.685 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =    8, score = -198.450 (best =  -1.859) position = 0.0 0.0 0.0
Episode =    9, score = -227.215 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   10, score = -255.980 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   11, score = -284.744 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   12, score = -313.509 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   13, score = -342.274 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   14, score = -371.039 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   15, score = -399.805 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   16, score = -428.570 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   17, score = -457.335 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   18, score = -486.101 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   19, score = -514.866 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   20, score = -543.632 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   21, score = -572.397 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   22, score = -601.162 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   23, score = -629.928 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   24, score = -656.114 (best =  -1.859) position = 0.0 0.6 0.0
Episode =   25, score = -667.627 (best =  -1.859) position = -0.0 0.4 0.0
Episode =   26, score = -696.393 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   27, score = -725.158 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   28, score = -753.923 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   29, score = -782.689 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   30, score = -811.454 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   31, score = -840.219 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   32, score = -868.985 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   33, score = -897.750 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   34, score = -926.516 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   35, score = -955.281 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   36, score = -984.046 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   37, score = -1012.812 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   38, score = -1041.577 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   39, score = -1070.342 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   40, score = -1099.108 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   41, score = -1127.873 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   42, score = -1156.638 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   43, score = -1185.404 (best =  -1.859) position = -0.0 0.0 0.0
```

```
Episode =   44, score = -1214.169 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   45, score = -1242.934 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   46, score = -1271.700 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   47, score = -1300.465 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   48, score = -1329.231 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   49, score = -1357.996 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   50, score = -1386.761 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   51, score = -1415.527 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   52, score = -1444.292 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   53, score = -1473.057 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   54, score = -1501.823 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   55, score = -1530.588 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   56, score = -1559.353 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   57, score = -1588.119 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   58, score = -1616.884 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   59, score = -1645.650 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   60, score = -1674.415 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   61, score = -1703.180 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   62, score = -1731.946 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   63, score = -1760.711 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   64, score = -1789.476 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   65, score = -1818.242 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   66, score = -1847.007 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   67, score = -1875.772 (best =  -1.859) position = 0.0 0.0 0.0
Episode =   68, score = -1904.537 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   69, score = -1916.273 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   70, score = -1945.038 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   71, score = -1973.803 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   72, score = -1975.568 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   73, score = -1977.332 (best =  -1.859) position = -0.0 0.0 0.0
Episode =   74, score = -1979.082 (best =  -1.859) position = -0.0 -0.0 0.0
Episode =   75, score = -2007.846 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   76, score = -2036.611 (best =  -1.859) position = 0.0 -0.0 0.0
Episode =   77, score = -2050.836 (best =  -1.859) position = 0.0 0.3 0.0
Episode =   78, score = -2067.159 (best =  -1.859) position = 0.0 1.7 0.0
Episode =   79, score = -2095.114 (best =  -1.859) position = -0.0 1.0 0.0
Episode =   80, score = -2113.550 (best =  -1.859) position = 0.1 0.6 0.0
Episode =   81, score = -2116.807 (best =  -1.859) position = -1.5 2.2 0.0
Episode =   82, score = -2160.171 (best =  -1.859) position = -1.2 -9.8 0.0
Episode =   83, score = -2176.156 (best =  -1.859) position = -0.9 6.9 0.0
Episode =   84, score = -2192.153 (best =  -1.859) position = 2.8 6.9 0.0
Episode =   85, score = -2225.234 (best =  -1.859) position = 1.7 6.7 0.0
Episode =   86, score = -2258.603 (best =  -1.859) position = -0.8 6.8 0.0
Episode =   87, score = -2291.995 (best =  -1.859) position = -0.2 6.8 0.0
Episode =   88, score = -2325.226 (best =  -1.859) position = -1.0 6.7 0.0
Episode =   89, score = -2358.314 (best =  -1.859) position = -0.5 6.7 0.0
Episode =   90, score = -2391.639 (best =  -1.859) position = -1.0 6.7 0.0
Episode =   91, score = -2425.000 (best =  -1.859) position = -1.8 6.8 0.0
```

```
Episode =   92, score = -2440.912 (best =  -1.859) position = -0.3 6.9 0.0
Episode =   93, score = -2483.920 (best =  -1.859) position = -0.3 6.7 0.0
Episode =   94, score = -2516.960 (best =  -1.859) position = -0.7 6.7 0.0
Episode =   95, score = -2550.183 (best =  -1.859) position = -1.7 6.7 0.0
Episode =   96, score = -2583.339 (best =  -1.859) position = -3.6 6.7 0.0
Episode =   97, score = -2616.535 (best =  -1.859) position = -2.9 6.7 0.0
Episode =   98, score = -2632.432 (best =  -1.859) position = 1.4 6.9 0.0
Episode =   99, score = -2665.686 (best =  -1.859) position = 0.6 6.8 0.0
Episode =  100, score = -2698.815 (best =  -1.859) position = -0.8 6.8 0.0
Episode =  101, score = -2732.101 (best =  -1.859) position = -1.7 6.8 0.0
Episode =  102, score = -2748.022 (best =  -1.859) position = 1.0 6.9 0.0
Episode =  103, score = -2764.153 (best =  -1.859) position = -4.5 6.9 0.0
Episode =  104, score = -2807.174 (best =  -1.859) position = 0.2 6.7 0.0
Episode =  105, score = -2823.098 (best =  -1.859) position = -4.0 6.8 0.0
Episode =  106, score = -2839.057 (best =  -1.859) position = 3.8 6.9 0.0
Episode =  107, score = -2872.124 (best =  -1.859) position = -0.2 6.7 0.0
Episode =  108, score = -2905.271 (best =  -1.859) position = -0.8 6.8 0.0
Episode =  109, score = -2921.173 (best =  -1.859) position = 1.6 6.9 0.0
Episode =  110, score = -2954.485 (best =  -1.859) position = -0.1 6.7 0.0
Episode =  111, score = -2987.652 (best =  -1.859) position = 0.1 6.8 0.0
Episode =  112, score = -3020.879 (best =  -1.859) position = 1.0 6.8 0.0
Episode =  113, score = -3063.888 (best =  -1.859) position = -0.3 6.7 0.0
Episode =  114, score = -3096.989 (best =  -1.859) position = -0.5 6.7 0.0
Episode =  115, score = -3122.553 (best =  -1.859) position = -0.1 6.8 0.0
Episode =  116, score = -3138.506 (best =  -1.859) position = 1.5 6.9 0.0
Episode =  117, score = -3171.805 (best =  -1.859) position = 0.8 6.8 0.0
Episode =  118, score = -3205.059 (best =  -1.859) position = -2.7 6.8 0.0
Episode =  119, score = -3238.229 (best =  -1.859) position = 0.6 6.7 0.0
Episode =  120, score = -3254.117 (best =  -1.859) position = 1.7 6.9 0.0
Episode =  121, score = -3287.232 (best =  -1.859) position = -3.2 6.7 0.0
Episode =  122, score = -3320.578 (best =  -1.859) position = -1.3 6.8 0.0
Episode =  123, score = -3336.599 (best =  -1.859) position = -1.1 6.9 0.0
Episode =  124, score = -3379.624 (best =  -1.859) position = -1.0 6.7 0.0
Episode =  125, score = -3395.573 (best =  -1.859) position = 0.2 6.9 0.0
Episode =  126, score = -3428.873 (best =  -1.859) position = -3.5 6.7 0.0
Episode =  127, score = -3461.990 (best =  -1.859) position = 0.1 6.8 0.0
Episode =  128, score = -3477.885 (best =  -1.859) position = -1.6 6.9 0.0
Episode =  129, score = -3510.963 (best =  -1.859) position = -0.8 6.7 0.0
Episode =  130, score = -3544.027 (best =  -1.859) position = -1.0 6.7 0.0
Episode =  131, score = -3577.427 (best =  -1.859) position = 0.9 6.8 0.0
Episode =  132, score = -3610.607 (best =  -1.859) position = 0.7 6.8 0.0
Episode =  133, score = -3643.694 (best =  -1.859) position = 1.2 6.8 0.0
Episode =  134, score = -3659.564 (best =  -1.859) position = -1.7 6.9 0.0
Episode =  135, score = -3692.697 (best =  -1.859) position = -1.4 6.7 0.0
Episode =  136, score = -3708.697 (best =  -1.859) position = 1.3 6.9 0.0
Episode =  137, score = -3741.747 (best =  -1.859) position = 0.0 6.8 0.0
Episode =  138, score = -3757.688 (best =  -1.859) position = -1.3 6.9 0.0
Episode =  139, score = -3783.292 (best =  -1.859) position = -0.7 6.8 0.0
```

```
Episode =  140, score = -3816.485 (best =  -1.859) position = 0.3 6.7 0.0
Episode =  141, score = -3849.611 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  142, score = -3892.642 (best =  -1.859) position = 1.2 6.7 0.0
Episode =  143, score = -3925.990 (best =  -1.859) position = 3.1 6.7 0.0
Episode =  144, score = -3959.109 (best =  -1.859) position = -0.1 6.7 0.0
Episode =  145, score = -3982.606 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  146, score = -3998.461 (best =  -1.859) position = -0.3 6.9 0.0
Episode =  147, score = -4024.066 (best =  -1.859) position = -0.0 6.8 0.0
Episode =  148, score = -4039.966 (best =  -1.859) position = -0.9 6.9 0.0
Episode =  149, score = -4073.107 (best =  -1.859) position = 2.1 6.7 0.0
Episode =  150, score = -4089.058 (best =  -1.859) position = -0.2 6.9 0.0
Episode =  151, score = -4122.391 (best =  -1.859) position = -1.0 6.8 0.0
Episode =  152, score = -4165.409 (best =  -1.859) position = 0.1 6.7 0.0
Episode =  153, score = -4181.451 (best =  -1.859) position = 2.7 6.9 0.0
Episode =  154, score = -4197.390 (best =  -1.859) position = 3.5 6.9 0.0
Episode =  155, score = -4220.888 (best =  -1.859) position = -2.2 6.8 0.0
Episode =  156, score = -4263.945 (best =  -1.859) position = 2.2 6.7 0.0
Episode =  157, score = -4279.811 (best =  -1.859) position = 0.2 6.9 0.0
Episode =  158, score = -4313.035 (best =  -1.859) position = -2.0 6.7 0.0
Episode =  159, score = -4346.125 (best =  -1.859) position = 0.3 6.8 0.0
Episode =  160, score = -4362.081 (best =  -1.859) position = -1.4 6.9 0.0
Episode =  161, score = -4377.998 (best =  -1.859) position = 1.0 7.0 0.0
Episode =  162, score = -4394.010 (best =  -1.859) position = 0.8 7.0 0.0
Episode =  163, score = -4409.757 (best =  -1.859) position = 2.1 6.9 0.0
Episode =  164, score = -4425.775 (best =  -1.859) position = -1.6 7.0 0.0
Episode =  165, score = -4458.998 (best =  -1.859) position = 2.0 6.8 0.0
Episode =  166, score = -4474.905 (best =  -1.859) position = -3.5 6.8 0.0
Episode =  167, score = -4490.745 (best =  -1.859) position = -2.1 6.9 0.0
Episode =  168, score = -4506.742 (best =  -1.859) position = -3.1 6.9 0.0
Episode =  169, score = -4530.506 (best =  -1.859) position = -0.4 6.9 0.0
Episode =  170, score = -4556.144 (best =  -1.859) position = -0.6 6.9 0.0
Episode =  171, score = -4589.518 (best =  -1.859) position = -2.3 6.8 0.0
Episode =  172, score = -4622.716 (best =  -1.859) position = 1.6 6.8 0.0
Episode =  173, score = -4655.966 (best =  -1.859) position = -0.4 6.8 0.0
Episode =  174, score = -4689.103 (best =  -1.859) position = 0.6 6.7 0.0
Episode =  175, score = -4722.325 (best =  -1.859) position = -0.2 6.8 0.0
Episode =  176, score = -4745.835 (best =  -1.859) position = 0.3 6.8 0.0
Episode =  177, score = -4779.054 (best =  -1.859) position = 2.1 6.8 0.0
Episode =  178, score = -4812.206 (best =  -1.859) position = 0.0 6.8 0.0
Episode =  179, score = -4827.971 (best =  -1.859) position = 0.7 6.9 0.0
Episode =  180, score = -4843.993 (best =  -1.859) position = -1.5 7.0 0.0
Episode =  181, score = -4887.015 (best =  -1.859) position = -1.1 6.7 0.0
Episode =  182, score = -4930.068 (best =  -1.859) position = 1.6 6.7 0.0
Episode =  183, score = -4945.985 (best =  -1.859) position = -0.2 6.9 0.0
Episode =  184, score = -4979.087 (best =  -1.859) position = -0.0 6.7 0.0
Episode =  185, score = -5012.227 (best =  -1.859) position = -2.0 6.8 0.0
Episode =  186, score = -5028.284 (best =  -1.859) position = 2.3 6.9 0.0
Episode =  187, score = -5061.605 (best =  -1.859) position = -2.1 6.8 0.0
```

```
Episode =  188, score = -5077.340 (best =  -1.859) position = -1.8 6.9 0.0
Episode =  189, score = -5093.303 (best =  -1.859) position = 1.6 6.9 0.0
Episode =  190, score = -5126.578 (best =  -1.859) position = -2.5 6.7 0.0
Episode =  191, score = -5159.780 (best =  -1.859) position = 1.0 6.7 0.0
Episode =  192, score = -5192.985 (best =  -1.859) position = 0.5 6.8 0.0
Episode =  193, score = -5208.864 (best =  -1.859) position = 2.2 6.9 0.0
Episode =  194, score = -5224.732 (best =  -1.859) position = -1.5 6.9 0.0
Episode =  195, score = -5240.638 (best =  -1.859) position = 3.4 6.9 0.0
Episode =  196, score = -5256.488 (best =  -1.859) position = 3.4 6.9 0.0
Episode =  197, score = -5289.707 (best =  -1.859) position = 1.2 6.8 0.0
Episode =  198, score = -5313.144 (best =  -1.859) position = 0.4 6.8 0.0
Episode =  199, score = -5329.087 (best =  -1.859) position = -1.1 6.9 0.0
Episode =  200, score = -5362.259 (best =  -1.859) position = -0.3 6.8 0.0
Episode =  201, score = -5387.939 (best =  -1.859) position = 1.8 6.9 0.0
Episode =  202, score = -5421.235 (best =  -1.859) position = -0.0 6.8 0.0
Episode =  203, score = -5454.274 (best =  -1.859) position = -0.2 6.7 0.0
Episode =  204, score = -5497.334 (best =  -1.859) position = -2.3 6.7 0.0
Episode =  205, score = -5530.519 (best =  -1.859) position = 0.8 6.8 0.0
Episode =  206, score = -5563.835 (best =  -1.859) position = 1.4 6.8 0.0
Episode =  207, score = -5597.087 (best =  -1.859) position = 0.9 6.8 0.0
Episode =  208, score = -5630.136 (best =  -1.859) position = 1.3 6.7 0.0
Episode =  209, score = -5663.416 (best =  -1.859) position = -0.4 6.8 0.0
Episode =  210, score = -5696.823 (best =  -1.859) position = 1.2 6.8 0.0
Episode =  211, score = -5739.819 (best =  -1.859) position = 0.6 6.7 0.0
Episode =  212, score = -5763.533 (best =  -1.859) position = 0.6 6.9 0.0
Episode =  213, score = -5796.596 (best =  -1.859) position = 0.4 6.7 0.0
Episode =  214, score = -5812.521 (best =  -1.859) position = -0.9 6.9 0.0
Episode =  215, score = -5845.584 (best =  -1.859) position = -1.1 6.7 0.0
Episode =  216, score = -5861.366 (best =  -1.859) position = -1.0 6.9 0.0
Episode =  217, score = -5877.346 (best =  -1.859) position = 0.4 7.0 0.0
Episode =  218, score = -5910.732 (best =  -1.859) position = -1.0 6.8 0.0
Episode =  219, score = -5936.364 (best =  -1.859) position = 2.3 6.8 0.0
Episode =  220, score = -5952.241 (best =  -1.859) position = -1.8 6.9 0.0
Episode =  221, score = -5985.617 (best =  -1.859) position = 2.0 6.8 0.0
Episode =  222, score = -6011.149 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  223, score = -6054.224 (best =  -1.859) position = 2.4 6.7 0.0
Episode =  224, score = -6070.156 (best =  -1.859) position = 0.9 6.9 0.0
Episode =  225, score = -6103.473 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  226, score = -6119.452 (best =  -1.859) position = 1.1 6.9 0.0
Episode =  227, score = -6135.453 (best =  -1.859) position = -4.4 6.9 0.0
Episode =  228, score = -6178.524 (best =  -1.859) position = -2.7 6.7 0.0
Episode =  229, score = -6211.776 (best =  -1.859) position = 0.4 6.7 0.0
Episode =  230, score = -6254.830 (best =  -1.859) position = 1.3 6.7 0.0
Episode =  231, score = -6270.799 (best =  -1.859) position = 0.6 6.9 0.0
Episode =  232, score = -6304.085 (best =  -1.859) position = 0.9 6.8 0.0
Episode =  233, score = -6320.051 (best =  -1.859) position = -1.5 6.9 0.0
Episode =  234, score = -6353.335 (best =  -1.859) position = 0.9 6.8 0.0
Episode =  235, score = -6378.922 (best =  -1.859) position = 4.5 6.8 0.0
```

```
Episode =  236, score = -6394.727 (best =  -1.859) position = -0.5 6.9 0.0
Episode =  237, score = -6420.075 (best =  -1.859) position = 1.6 6.8 0.0
Episode =  238, score = -6463.100 (best =  -1.859) position = 0.7 6.7 0.0
Episode =  239, score = -6479.139 (best =  -1.859) position = -3.2 6.9 0.0
Episode =  240, score = -6512.177 (best =  -1.859) position = -0.5 6.7 0.0
Episode =  241, score = -6528.201 (best =  -1.859) position = 2.1 6.9 0.0
Episode =  242, score = -6561.263 (best =  -1.859) position = 0.7 6.7 0.0
Episode =  243, score = -6576.999 (best =  -1.859) position = 0.5 6.8 0.0
Episode =  244, score = -6592.850 (best =  -1.859) position = 2.1 6.9 0.0
Episode =  245, score = -6608.926 (best =  -1.859) position = -4.1 6.9 0.0
Episode =  246, score = -6624.903 (best =  -1.859) position = -1.0 6.9 0.0
Episode =  247, score = -6648.336 (best =  -1.859) position = -0.0 6.8 0.0
Episode =  248, score = -6671.770 (best =  -1.859) position = -1.4 6.8 0.0
Episode =  249, score = -6704.954 (best =  -1.859) position = -1.0 6.8 0.0
Episode =  250, score = -6738.049 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  251, score = -6771.307 (best =  -1.859) position = 0.9 6.8 0.0
Episode =  252, score = -6804.569 (best =  -1.859) position = -1.2 6.8 0.0
Episode =  253, score = -6837.614 (best =  -1.859) position = -0.8 6.7 0.0
Episode =  254, score = -6853.601 (best =  -1.859) position = -0.2 6.9 0.0
Episode =  255, score = -6869.462 (best =  -1.859) position = 1.7 6.9 0.0
Episode =  256, score = -6902.626 (best =  -1.859) position = -2.2 6.7 0.0
Episode =  257, score = -6935.902 (best =  -1.859) position = 1.2 6.8 0.0
Episode =  258, score = -6978.927 (best =  -1.859) position = 0.3 6.7 0.0
Episode =  259, score = -6994.856 (best =  -1.859) position = -1.6 6.9 0.0
Episode =  260, score = -7010.829 (best =  -1.859) position = -1.9 6.9 0.0
Episode =  261, score = -7043.988 (best =  -1.859) position = -0.5 6.8 0.0
Episode =  262, score = -7086.998 (best =  -1.859) position = -0.2 6.7 0.0
Episode =  263, score = -7102.812 (best =  -1.859) position = -1.4 6.9 0.0
Episode =  264, score = -7135.918 (best =  -1.859) position = 0.5 6.8 0.0
Episode =  265, score = -7178.940 (best =  -1.859) position = -0.3 6.7 0.0
Episode =  266, score = -7212.256 (best =  -1.859) position = 1.5 6.8 0.0
Episode =  267, score = -7245.317 (best =  -1.859) position = -1.2 6.7 0.0
Episode =  268, score = -7261.209 (best =  -1.859) position = 2.6 6.9 0.0
Episode =  269, score = -7277.550 (best =  -1.859) position = 20.3 5.2 0.0
Episode =  270, score = -7307.279 (best =  -1.859) position = -11.8 1.2 0.0
Episode =  271, score = -7331.803 (best =  -1.859) position = 2.2 6.0 0.0
Episode =  272, score = -7356.744 (best =  -1.859) position = 1.7 3.4 0.0
Episode =  273, score = -7403.568 (best =  -1.859) position = -0.2 -1.0 0.0
Episode =  274, score = -7967.493 (best =  -1.859) position = 18.2 128.5 40.4
Episode =  275, score = -7909.658 (best =  -1.859) position = -23.5 -10.7 90.8
Episode =  276, score = -7848.773 (best =  -1.859) position = -22.9 -3.3 90.3
Episode =  277, score = -7787.712 (best =  -1.859) position = -22.9 -0.3 90.3
Episode =  278, score = -7728.163 (best =  -1.859) position = -22.4 9.8 89.8
Episode =  279, score = -7678.497 (best =  -1.859) position = -23.7 -22.1 90.4
Episode =  280, score = -7619.096 (best =  -1.859) position = -22.8 -6.6 90.2
Episode =  281, score = -7557.331 (best =  -1.859) position = -22.4 4.1 90.1
Episode =  282, score = -7502.100 (best =  -1.859) position = -23.2 -16.1 90.4
Episode =  283, score = -7444.406 (best =  -1.859) position = -22.2 9.4 89.5
```

```
Episode =  284, score = -7397.295 (best =  -1.859) position = -21.8 22.2 88.1
Episode =  285, score = -7336.847 (best =  -1.859) position = -22.5 0.9 89.9
Episode =  286, score = -7278.791 (best =  -1.859) position = -22.3 7.1 89.6
Episode =  287, score = -7218.853 (best =  -1.859) position = -22.6 7.5 89.9
Episode =  288, score = -7157.412 (best =  -1.859) position = -23.1 -4.7 90.6
Episode =  289, score = -7111.316 (best =  -1.859) position = -23.8 -23.8 90.8
Episode =  290, score = -7069.156 (best =  -1.859) position = -23.9 -26.2 90.8
Episode =  291, score = -7020.526 (best =  -1.859) position = -22.1 18.1 88.4
Episode =  292, score = -7014.018 (best =  -1.859) position = -24.5 -47.3 89.0
Episode =  293, score = -6954.419 (best =  -1.859) position = -22.7 6.0 89.7
Episode =  294, score = -6894.424 (best =  -1.859) position = -22.7 5.5 89.8
Episode =  295, score = -6889.516 (best =  -1.859) position = -24.4 -49.1 88.3
Episode =  296, score = -6874.703 (best =  -1.859) position = -24.4 -46.3 88.8
Episode =  297, score = -6828.332 (best =  -1.859) position = -23.6 -27.9 90.2
Episode =  298, score = -6785.441 (best =  -1.859) position = -21.7 24.8 87.7
Episode =  299, score = -6723.787 (best =  -1.859) position = -23.0 0.6 90.4
Episode =  300, score = -6670.016 (best =  -1.859) position = -22.1 16.2 88.9
Episode =  301, score = -6614.432 (best =  -1.859) position = -23.4 -16.2 90.8
Episode =  302, score = -6553.967 (best =  -1.859) position = -22.9 -1.5 90.3
Episode =  303, score = -6501.788 (best =  -1.859) position = -23.5 -21.2 90.6
Episode =  304, score = -6440.431 (best =  -1.859) position = -22.9 -5.6 90.5
Episode =  305, score = -6381.343 (best =  -1.859) position = -22.4 6.4 89.7
Episode =  306, score = -6324.803 (best =  -1.859) position = -23.5 -12.5 90.8
Episode =  307, score = -6268.581 (best =  -1.859) position = -22.2 9.3 89.2
Episode =  308, score = -6213.684 (best =  -1.859) position = -22.2 16.0 89.0
Episode =  309, score = -6158.237 (best =  -1.859) position = -23.6 -14.2 90.7
Episode =  310, score = -6097.704 (best =  -1.859) position = -23.2 -5.6 90.6
Episode =  311, score = -6038.124 (best =  -1.859) position = -23.2 -4.1 90.4
Episode =  312, score = -5977.952 (best =  -1.859) position = -23.3 -6.5 90.6
Episode =  313, score = -5925.547 (best =  -1.859) position = -23.6 -21.1 90.3
Episode =  314, score = -5864.985 (best =  -1.859) position = -22.8 5.2 90.0
Episode =  315, score = -5805.111 (best =  -1.859) position = -23.2 -11.5 90.4
Episode =  316, score = -5744.046 (best =  -1.859) position = -22.6 6.1 90.0
Episode =  317, score = -5682.753 (best =  -1.859) position = -22.8 5.2 90.0
Episode =  318, score = -5622.419 (best =  -1.859) position = -22.9 -6.3 90.5
Episode =  319, score = -5564.071 (best =  -1.859) position = -23.0 -12.2 90.1
Episode =  320, score = -5503.999 (best =  -1.859) position = -22.7 3.3 89.9
Episode =  321, score = -5450.820 (best =  -1.859) position = -23.5 -18.0 90.7
Episode =  322, score = -5396.038 (best =  -1.859) position = -22.2 14.3 89.1
Episode =  323, score = -5335.179 (best =  -1.859) position = -22.8 -2.9 90.3
Episode =  324, score = -5289.570 (best =  -1.859) position = -23.8 -22.3 90.5
Episode =  325, score = -5242.962 (best =  -1.859) position = -21.9 20.9 88.4
Episode =  326, score = -5183.513 (best =  -1.859) position = -23.0 -5.1 90.2
Episode =  327, score = -5122.683 (best =  -1.859) position = -22.8 1.4 90.1
Episode =  328, score = -5090.126 (best =  -1.859) position = -24.0 -37.2 89.7
Episode =  329, score = -5035.043 (best =  -1.859) position = -23.2 -20.0 90.3
Episode =  330, score = -4973.224 (best =  -1.859) position = -23.0 0.8 90.4
Episode =  331, score = -4912.128 (best =  -1.859) position = -22.7 0.4 90.1
```

```
Episode =  332, score = -4853.418 (best =  -1.859) position = -23.1 -9.0 90.6
Episode =  333, score = -4793.297 (best =  -1.859) position = -23.1 -7.4 90.7
Episode =  334, score = -4736.414 (best =  -1.859) position = -22.9 -13.5 90.1
Episode =  335, score = -4681.078 (best =  -1.859) position = -22.4 14.4 89.1
Episode =  336, score = -4621.612 (best =  -1.859) position = -22.6 1.4 89.9
Episode =  337, score = -4572.866 (best =  -1.859) position = -23.7 -20.4 90.5
Episode =  338, score = -4511.604 (best =  -1.859) position = -22.6 0.8 90.1
Episode =  339, score = -4451.158 (best =  -1.859) position = -22.8 -0.0 90.1
Episode =  340, score = -4405.199 (best =  -1.859) position = -23.6 -25.8 90.3
Episode =  341, score = -4345.055 (best =  -1.859) position = -23.0 -8.2 90.7
Episode =  342, score = -4286.426 (best =  -1.859) position = -23.1 -10.7 90.6
Episode =  343, score = -4237.783 (best =  -1.859) position = -23.5 -25.1 90.2
Episode =  344, score = -4176.186 (best =  -1.859) position = -22.9 -1.4 90.3
Episode =  345, score = -4119.525 (best =  -1.859) position = -23.3 -15.4 90.6
Episode =  346, score = -4058.284 (best =  -1.859) position = -22.6 3.2 90.1
Episode =  347, score = -4013.842 (best =  -1.859) position = -23.6 -27.8 89.8
Episode =  348, score = -3975.754 (best =  -1.859) position = -23.8 -29.7 90.2
Episode =  349, score = -3918.358 (best =  -1.859) position = -23.3 -13.1 90.9
Episode =  350, score = -3858.957 (best =  -1.859) position = -22.9 -4.6 90.2
Episode =  351, score = -3799.234 (best =  -1.859) position = -22.4 0.5 89.8
Episode =  352, score = -3740.735 (best =  -1.859) position = -22.4 9.9 89.5
Episode =  353, score = -3679.423 (best =  -1.859) position = -22.7 2.9 90.2
Episode =  354, score = -3622.649 (best =  -1.859) position = -23.6 -12.1 91.1
Episode =  355, score = -3571.076 (best =  -1.859) position = -23.4 -22.1 90.2
Episode =  356, score = -3511.442 (best =  -1.859) position = -22.4 1.0 89.8
Episode =  357, score = -3463.902 (best =  -1.859) position = -23.2 -24.6 90.1
Episode =  358, score = -3407.643 (best =  -1.859) position = -23.1 -17.1 90.2
Episode =  359, score = -3346.136 (best =  -1.859) position = -22.9 -3.0 90.4
Episode =  360, score = -3309.968 (best =  -1.859) position = -23.7 -33.4 89.4
Episode =  361, score = -3266.804 (best =  -1.859) position = -23.8 -29.7 89.8
Episode =  362, score = -3210.604 (best =  -1.859) position = -23.4 -17.2 90.6
Episode =  363, score = -3165.910 (best =  -1.859) position = -22.0 26.3 87.6
Episode =  364, score = -3112.429 (best =  -1.859) position = -23.3 -17.1 90.3
Episode =  365, score = -3052.037 (best =  -1.859) position = -23.1 4.6 90.0
Episode =  366, score = -3004.824 (best =  -1.859) position = -23.7 -23.1 90.6
Episode =  367, score = -2948.532 (best =  -1.859) position = -22.1 10.4 89.3
Episode =  368, score = -2927.328 (best =  -1.859) position = -24.2 -41.5 89.4
Episode =  369, score = -2866.474 (best =  -1.859) position = -22.6 5.4 90.0
Episode =  370, score = -2804.560 (best =  -1.859) position = -22.9 -0.3 90.4
Episode =  371, score = -2744.260 (best =  -1.859) position = -23.0 -4.5 90.5
Episode =  372, score = -2682.925 (best =  -1.859) position = -23.0 -6.3 90.6
Episode =  373, score = -2667.073 (best =  -1.859) position = -24.3 -45.2 88.9
Episode =  374, score = -2622.600 (best =  -1.859) position = -23.4 -26.0 89.7
Episode =  375, score = -2582.298 (best =  -1.859) position = -23.8 -31.6 90.0
Episode =  376, score = -2523.133 (best =  -1.859) position = -22.6 -4.4 89.9
Episode =  377, score = -2461.782 (best =  -1.859) position = -22.8 3.2 90.1
Episode =  378, score = -2401.700 (best =  -1.859) position = -22.6 5.2 89.9
Episode =  379, score = -2354.922 (best =  -1.859) position = -23.5 -24.0 90.1
```

```
Episode =  380, score = -2293.380 (best =  -1.859) position = -22.8 6.8 90.0
Episode =  381, score = -2240.727 (best =  -1.859) position = -23.3 -22.9 90.4
Episode =  382, score = -2195.368 (best =  -1.859) position = -23.6 -27.8 90.0
Episode =  383, score = -2134.268 (best =  -1.859) position = -23.1 -2.2 90.4
Episode =  384, score = -2148.097 (best =  -1.859) position = -24.9 -56.6 87.6
Episode =  385, score = -2088.604 (best =  -1.859) position = -23.3 -9.4 90.5
Episode =  386, score = -2050.318 (best =  -1.859) position = -23.7 -32.8 89.5
Episode =  387, score = -2023.346 (best =  -1.859) position = -23.8 -37.9 89.6
Episode =  388, score = -1974.737 (best =  -1.859) position = -23.2 -24.2 89.7
Episode =  389, score = -1916.863 (best =  -1.859) position = -23.3 -7.1 90.5
Episode =  390, score = -1855.383 (best =  -1.859) position = -22.8 -2.6 90.2
Episode =  391, score = -1796.538 (best =  -1.859) position = -23.1 -9.1 90.4
Episode =  392, score = -1736.567 (best =  -1.859) position = -23.0 -5.8 90.5
Episode =  393, score = -1712.967 (best =  -1.859) position = -21.5 37.6 85.7
Episode =  394, score = -1651.861 (best =  -1.859) position = -22.6 7.0 90.0
Episode =  395, score = -1594.663 (best =  -1.859) position = -22.4 12.5 89.3
Episode =  396, score = -1534.885 (best =  -1.859) position = -23.2 -11.9 90.7
Episode =  397, score = -1475.814 (best =  -1.859) position = -22.5 0.3 89.7
Episode =  398, score = -1416.358 (best =  -1.859) position = -22.6 4.3 89.8
Episode =  399, score = -1361.077 (best =  -1.859) position = -23.4 -9.8 90.5
Episode =  400, score = -1298.979 (best =  -1.859) position = -22.7 2.1 90.2
Episode =  401, score = -1238.864 (best =  -1.859) position = -22.5 7.0 89.8
Episode =  402, score = -1181.359 (best =  -1.859) position = -23.5 -5.6 90.5
Episode =  403, score = -1120.234 (best =  -1.859) position = -22.9 -2.0 90.3
Episode =  404, score = -1059.147 (best =  -1.859) position = -22.7 -1.7 90.2
Episode =  405, score = -1015.510 (best =  -1.859) position = -23.7 -28.1 90.1
Episode =  406, score = -985.007 (best =  -1.859) position = -23.9 -39.0 88.6
Episode =  407, score = -993.833 (best =  -1.859) position = -24.5 -57.4 86.7
Episode =  408, score = -936.157 (best =  -1.859) position = -22.5 11.1 89.5
Episode =  409, score = -886.725 (best =  -1.859) position = -21.7 16.6 88.5
Episode =  410, score = -827.723 (best =  -1.859) position = -22.7 10.1 89.5
Episode =  411, score = -769.047 (best =  -1.859) position = -23.1 -12.3 90.5
Episode =  412, score = -707.393 (best =  -1.859) position = -22.7 5.4 90.2
Episode =  413, score = -652.236 (best =  -1.859) position = -23.5 -17.2 90.9
Episode =  414, score = -589.927 (best =  -1.859) position = -22.7 -0.4 90.3
Episode =  415, score = -551.887 (best =  -1.859) position = -23.6 -32.3 89.5
Episode =  416, score = -491.157 (best =  -1.859) position = -22.9 0.9 90.2
Episode =  417, score = -431.462 (best =  -1.859) position = -22.6 -1.3 89.9
Episode =  418, score = -442.490 (best =  -1.859) position = -25.0 -53.0 89.0
Episode =  419, score = -422.748 (best =  -1.859) position = -24.1 -44.8 88.4
Episode =  420, score = -366.718 (best =  -1.859) position = -23.5 -14.8 91.0
Episode =  421, score = -362.993 (best =  -1.859) position = -24.6 -53.1 87.6
Episode =  422, score = -309.597 (best =  -1.859) position = -23.3 -15.5 90.5
Episode =  423, score = -247.021 (best =  -1.859) position = -23.0 0.9 90.5
Episode =  424, score = -206.856 (best =  -1.859) position = -23.8 -31.1 90.2
Episode =  425, score = -145.774 (best =  -1.859) position = -22.8 -0.9 90.2
Episode =  426, score = -94.363 (best =  -1.859) position = -23.7 -20.1 91.1
Episode =  427, score = -34.858 (best =  -1.859) position = -22.5 0.8 89.9
```

```
Episode =  428, score =  25.041 (best =  25.041) position = -22.6 3.0 89.9
Episode =  429, score =  72.992 (best =  72.992) position = -23.4 -24.8 90.2
Episode =  430, score = 132.428 (best = 132.428) position = -23.1 -8.6 90.4
Episode =  431, score = 190.713 (best = 190.713) position = -22.3 5.0 89.6
Episode =  432, score = 250.997 (best = 250.997) position = -23.1 -1.5 90.2
Episode =  433, score = 307.717 (best = 307.717) position = -23.1 -13.7 90.1
Episode =  434, score = 368.249 (best = 368.249) position = -23.1 -10.0 90.6
Episode =  435, score = 384.164 (best = 384.164) position = -24.1 -47.6 87.7
Episode =  436, score = 441.141 (best = 441.141) position = -23.2 -11.4 90.7
Episode =  437, score = 493.275 (best = 493.275) position = -21.9 18.0 88.6
Episode =  438, score = 550.864 (best = 550.864) position = -22.6 14.4 89.1
Episode =  439, score = 613.106 (best = 613.106) position = -23.0 -1.6 90.5
Episode =  440, score = 657.061 (best = 657.061) position = -21.9 24.5 87.8
Episode =  441, score = 690.879 (best = 690.879) position = -23.9 -35.1 89.5
Episode =  442, score = 748.428 (best = 748.428) position = -23.3 -15.4 90.9
Episode =  443, score = 800.827 (best = 800.827) position = -23.2 -20.1 90.3
Episode =  444, score = 852.125 (best = 852.125) position = -23.4 -22.3 90.6
Episode =  445, score = 911.999 (best = 911.999) position = -22.8 4.1 89.9
Episode =  446, score = 962.685 (best = 962.685) position = -22.1 19.8 88.3
Episode =  447, score = 1010.358 (best = 1010.358) position = -23.5 -23.4 90.3
Episode =  448, score = 1065.181 (best = 1065.181) position = -23.3 -16.9 90.3
Episode =  449, score = 1122.224 (best = 1122.224) position = -23.4 -12.9 90.7
Episode =  450, score = 1172.496 (best = 1172.496) position = -23.2 -24.1 89.7
Episode =  451, score = 1234.574 (best = 1234.574) position = -22.7 2.6 90.2
Episode =  452, score = 1281.555 (best = 1281.555) position = -23.8 -22.4 90.7
Episode =  453, score = 1340.822 (best = 1340.822) position = -22.9 -7.4 90.3
Episode =  454, score = 1399.403 (best = 1399.403) position = -23.4 -6.1 90.5
Episode =  455, score = 1458.119 (best = 1458.119) position = -22.5 7.6 89.5
Episode =  456, score = 1509.455 (best = 1509.455) position = -23.4 -22.1 90.6
Episode =  457, score = 1549.875 (best = 1549.875) position = -21.7 27.2 87.5
Episode =  458, score = 1608.905 (best = 1608.905) position = -22.5 1.8 89.7
Episode =  459, score = 1661.280 (best = 1661.280) position = -23.4 -18.3 90.7
Episode =  460, score = 1720.381 (best = 1720.381) position = -22.7 -1.0 89.8
Episode =  461, score = 1781.365 (best = 1781.365) position = -22.9 -2.7 90.3
Episode =  462, score = 1843.632 (best = 1843.632) position = -22.8 2.0 90.4
Episode =  463, score = 1901.089 (best = 1901.089) position = -23.0 12.2 89.0
Episode =  464, score = 1962.974 (best = 1962.974) position = -23.0 -9.6 90.7
Episode =  465, score = 2004.131 (best = 2004.131) position = -23.6 -29.8 90.2
Episode =  466, score = 2066.241 (best = 2066.241) position = -22.9 2.1 90.4
Episode =  467, score = 2106.893 (best = 2106.893) position = -24.0 -26.3 90.7
Episode =  468, score = 2166.563 (best = 2166.563) position = -22.3 -3.0 89.7
Episode =  469, score = 2207.609 (best = 2207.609) position = -23.5 -28.6 90.2
Episode =  470, score = 2266.641 (best = 2266.641) position = -22.6 8.1 89.7
Episode =  471, score = 2318.366 (best = 2318.366) position = -23.3 -20.7 89.8
Episode =  472, score = 2357.352 (best = 2357.352) position = -23.9 -30.9 90.2
Episode =  473, score = 2412.143 (best = 2412.143) position = -22.1 11.1 89.2
Episode =  474, score = 2472.676 (best = 2472.676) position = -22.7 -3.0 90.2
Episode =  475, score = 2530.896 (best = 2530.896) position = -23.0 -10.9 90.4
```

```
Episode =  476, score = 2591.224 (best = 2591.224) position = -23.0 -5.2 90.5
Episode =  477, score = 2641.278 (best = 2641.278) position = -23.7 -18.1 90.7
Episode =  478, score = 2696.842 (best = 2696.842) position = -23.2 -15.4 90.5
Episode =  479, score = 2719.085 (best = 2719.085) position = -24.0 -43.6 88.7
Episode =  480, score = 2765.783 (best = 2765.783) position = -23.9 -23.2 90.8
Episode =  481, score = 2826.930 (best = 2826.930) position = -22.8 -2.5 90.2
Episode =  482, score = 2876.843 (best = 2876.843) position = -23.5 -20.7 90.3
Episode =  483, score = 2926.152 (best = 2926.152) position = -23.5 -24.5 90.3
Episode =  484, score = 2975.274 (best = 2975.274) position = -23.7 -21.0 90.7
Episode =  485, score = 2981.247 (best = 2981.247) position = -24.6 -47.1 89.1
Episode =  486, score = 3029.673 (best = 3029.673) position = -21.8 19.0 88.3
Episode =  487, score = 3088.924 (best = 3088.924) position = -22.4 2.1 89.6
Episode =  488, score = 3148.421 (best = 3148.421) position = -22.8 -6.0 90.3
Episode =  489, score = 3182.805 (best = 3182.805) position = -23.9 -35.4 89.8
Episode =  490, score = 3239.271 (best = 3239.271) position = -22.3 13.7 89.3
Episode =  491, score = 3286.879 (best = 3286.879) position = -23.7 -22.8 90.5
Episode =  492, score = 3342.244 (best = 3342.244) position = -22.5 16.9 88.9
Episode =  493, score = 3393.037 (best = 3393.037) position = -23.4 -20.0 90.5
Episode =  494, score = 3437.618 (best = 3437.618) position = -23.8 -28.6 90.0
Episode =  495, score = 3492.268 (best = 3492.268) position = -23.4 -19.0 90.7
Episode =  496, score = 3515.384 (best = 3515.384) position = -24.0 -42.6 88.4
Episode =  497, score = 3572.787 (best = 3572.787) position = -22.3 7.5 89.5
Episode =  498, score = 3631.676 (best = 3631.676) position = -22.5 5.0 89.7
Episode =  499, score = 3683.939 (best = 3683.939) position = -23.6 -19.7 90.6
Episode =  500, score = 3726.344 (best = 3726.344) position = -23.6 -28.6 90.0
```

In [12]:
```
target_pos
```

Out[12]:
```
array([  0.,   0., 100.])
```

In [13]:
```
task.sim.pose
```

Out[13]:
```
array([-23.61814166, -28.64944376,  89.96273257,   2.99232884,
         3.05002455,   0.        ])
```

In [14]:
```
reward
```

Out[14]:
```
2.1140035041897662
```

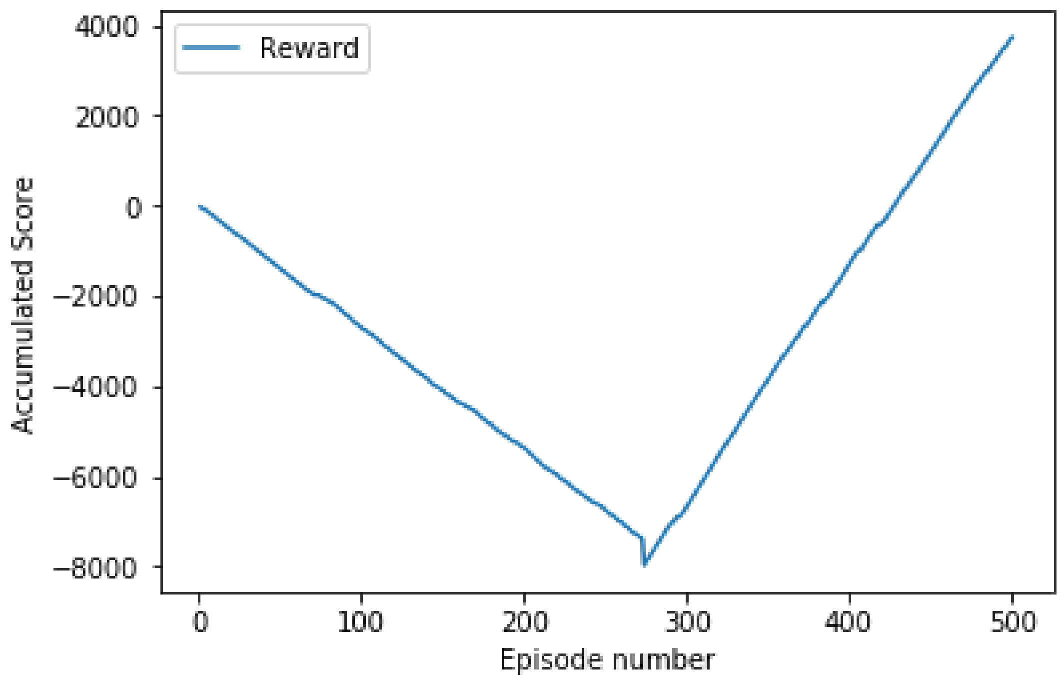
1.6 Plot the Rewards

Once you are satisfied with your performance, plot the episode rewards, either from a single run, or averaged over multiple runs.

In [15]:
```
## TODO: Plot the rewards.
import matplotlib.pyplot as plt
%matplotlib inline
```

```
plt.plot(results['Episode'], results['Score'], label='Reward')
#plt.plot(results['Episode'], rewards, label = 'Reward magnitude over episode')
#plt.plot(results['Episode'], results['Best Score'], label='Best total reward')
plt.xlabel('Episode number')
plt.ylabel('Accumulated Score')
plt.legend()
```

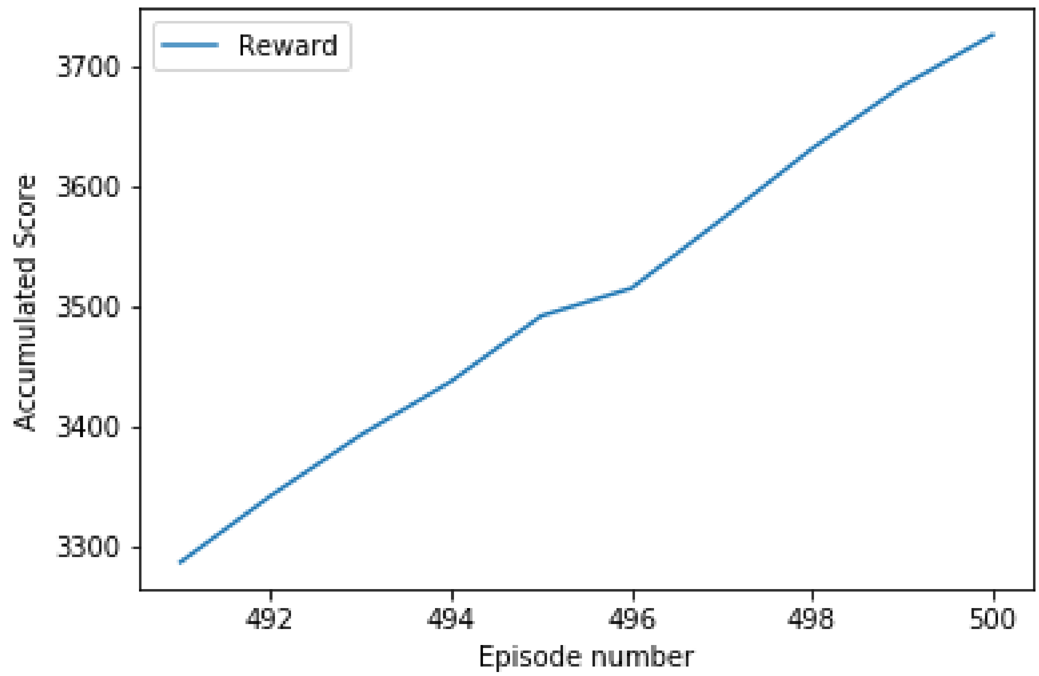
```
Out[15]: <matplotlib.legend.Legend at 0x10bde0a90>
```



1.7 Rewards over the Last 10 Episodes

```
In [20]: plt.plot(results['Episode'][-10:], results['Score'][-10:], label = 'Reward')
         plt.xlabel('Episode number')
         plt.ylabel('Accumulated Score')
         plt.legend()
```

```
Out[20]: <matplotlib.legend.Legend at 0x136a488d0>
```




```
In [26]: end_avg_reward = np.sum(results['Score'][-10:])/10
         end_avg_reward

Out[26]: 3508.2175232806394
```

1.8 Reflections

Question 1: Describe the task that you specified in `task.py`. How did you design the reward function?

Answer:

I set a target location for the quadroter to be 100 units off the ground. As a result, I separated the reward function into three distinct phases:

1.) I defined an error function to be proportional to the magnitude of the error, otherwise known as the distance between the current position and the target position, in each cartesian coordinate, x, y, and z. From there, I multipled a scaling factor to this error term to prevent too large of an error being added over the course of the agent's episode.

2.) If the agent was closer to the set target destination, the function gives the agent a large reward. This incentives the agent to get closer to the target destination, and once it does, it receives a large value so that it can continue to learn to move in that direction.

3.) My agent, over the course of a few episodes, tended to stay at z=0 height, so I built an additional condition that gave the quadroter a slight increase in velocity if it wasn't moving. This helped increase the rate of convergence on the target position.

Question 2: Discuss your agent briefly, using the following questions as a guide:

- What learning algorithm(s) did you try? What worked best for you?

- What was your final choice of hyperparameters (such as α, γ, ϵ, etc.)?
- What neural network architecture did you use (if any)? Specify layers, sizes, activation functions, etc.

Answer:
(1.) I used a deep deterministric policy gradient algorithm because the agent operates in a continuous action and state spaces. After tweaking the depth of the neural network for both the actor and the critic model, I discovered that most of the gains were coming from defining an appropriate reinforcement learning algorithm. The final choice of hyperparameters for alpha, gamma, and episilon were 0.99, 0.99, 0.01.
(2.) I used a three layer neural network for the actor and critic part of the implementation of the DDPG algorithm; each had 32, 64, and 32 nodes in each respective layer.
Question 3: Using the episode rewards plot, discuss how the agent learned over time.

- Was it an easy task to learn or hard?
- Was there a gradual learning curve, or an aha moment?
- How good was the final performance of the agent? (e.g. mean rewards over the last 10 episodes)

Answer:
(1.) I define easy depending on the number of degrees of freedom the agent needs to obtain and maintain for position, euler angels, velocity, and angular velocity. An easy task, therefore, would require the agent to hover at a set position; a difficult task would require the agent to hover at a specifc angle while moving in a direction with constant velocity. For our task, which required the agent to approach z at location 100, it is considered easy.
(2.) There seems to be an aha moment. Throughout the learning process, the agent is losing much of the reward. At around episode 250, the agent begins to gain gradually gain more reward for achieving the target position. It did, however, tend to converge at a z-distance of 90 instead of 100.
(3.) The final performance of the agent was satisfactory. The average reward for the last 10 episodes is 3508, which is significantly higher than the average reward at the start and middle of the learning process for the agent.
Question 4: Briefly summarize your experience working on this project. You can use the following prompts for ideas.

- What was the hardest part of the project? (e.g. getting started, plotting, specifying the task, etc.)
- Did you find anything interesting in how the quadcopter or your agent behaved?

Answer:
(1.) The hardest part of the project, besides defining an appropriate reward function, is understanding why the agent, given a set tasks, varies so drastically based on how we define the function. I learned that is pretty hard to achieve perfect convergence on a task, and that tweaking the function and network hyperparameters can drastically improve or ruin your results. As a result, the hardest part is just iterating through the different potential configurations and being happy with the outcome!
(2.) Most of the code is provided in the project description so it makes me think that this project would be more helpful if we had designated mentors to help us work through key concepts that we need to learn. I found it incredibly interesting to see the DDPG algorithm applied to such a

complex agent, but I felt like I didn't deeply understand it because no one answered any of my questions. The difference between Udacity and a top-tier university robotics program, and least in this case, is the ability to ask the hard questions and have someone more experienced walk us through the concepts so that we can implement the program.