In [1]:
```
from pathhack import pkg_path
import sys
from os.path import isdir, join
import time
import pickle
import numpy as np
import argparse

import torch
import matplotlib.pyplot as plt
from tensorboardX import SummaryWriter

from src.utils import average_offset_error, max_offset_error, final_offset_error
    padding, unpadding, padding_mask, batch_iter_no_shuffle, \
    load_preprocessed_train_test_dataset, ilm
from src.wlstm.models import ReBiL
from src.wlstm.utils import load_rebil_model
```

In [2]:
```
"""
- dataset
    # a dict that has two keys: train and test.
    - ['train']
        # training dataset.
        # a dict that has three keys: base, true, loss_mask.
        # Note the entries with the same index in these three lists correspond t
        - ['base']
            # baseline prediction with the observation.
            # a list with torch tensors with varying time steps.
            - traj
                # tensor with shape (t, 2)
        - ['true']
            # the observation and the ground truth.
            # a list with torch tensors with varying time steps.
            - traj
                # tensor with shape (t, 2)
        - ['loss_mask']
            # mask with one on prediction time steps, with zero on observation t
            # a list with torch tensors with varying time steps.
            - mask
                # tensor with shape (t)

    - ['test']
        # test dataset.
        # a dict that has three keys: base, true, loss_mask.
        # Structure is the same as ['train'].
        - ['base']
            - traj
        - ['true']
            - traj
                # tensor with shape (t, 2)
        - ['loss_mask']
            - mask
"""
print()
```

In [3]:
```
```

```
dataset_ver=25 # 0, 25, 50, 75
dataset_filename = 'full_'+str(dataset_ver)+'.p'
dataset_filepath = join(pkg_path, 'datasets', dataset_filename)
with open(join(dataset_filepath), 'rb') as f:
    x_dict_list_tensor = pickle.load(f)
    print()
    print('LOAD DATASET')
    print(dataset_filename+' is loaded.')
    print()
traj_base_train, traj_true_train, traj_loss_mask_train, \
    traj_base_test, traj_true_test, traj_loss_mask_test = \
    x_dict_list_tensor['train']['base'], x_dict_list_tensor['train']['true'], x_
    x_dict_list_tensor['test']['base'], x_dict_list_tensor['test']['true'], x_di
```

```
LOAD DATASET
full_25.p is loaded.
```

In [4]:

```
traj_loss_mask_train[0].shape
```

Out[4]:

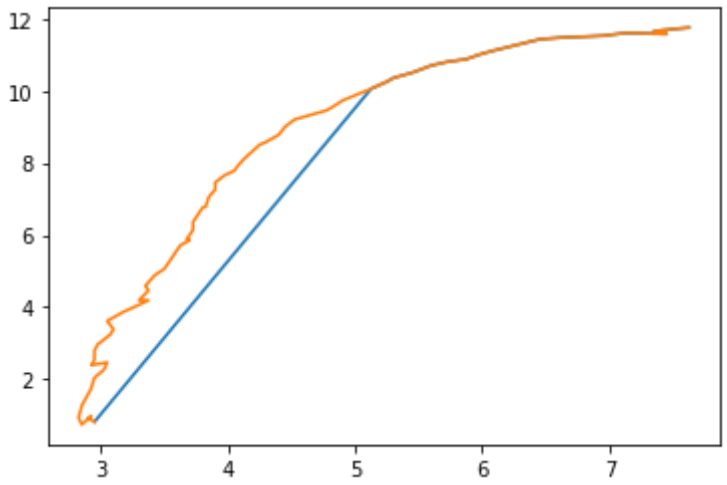
```
torch.Size([78])
```

In [5]:

```
# orange is ground truth, and blue is prediction.
plt.figure()
plt.plot(traj_base_train[0][:,0], traj_base_train[0][:,1])
plt.plot(traj_true_train[0][:,0], traj_true_train[0][:,1])
```

Out[5]:

```
[<matplotlib.lines.Line2D at 0x7ff46045db00>]
```



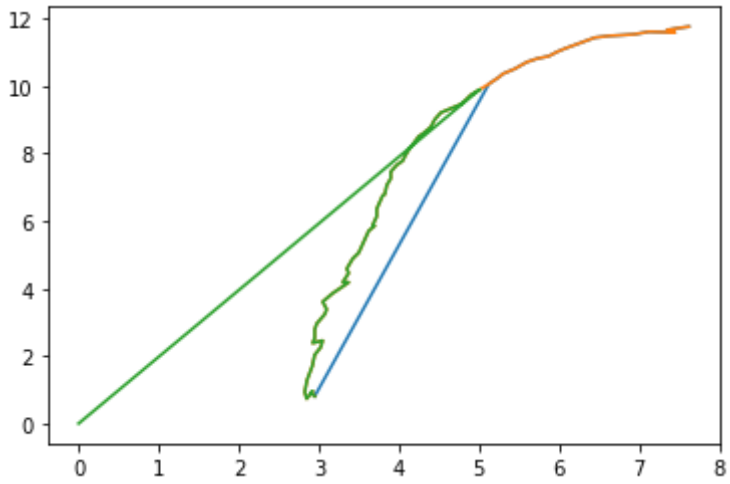
In [6]:

```
# orange is ground truth, blue is prediction, and green masks the ground truth i
# The straight line connecting the last observed position and (0,0) exists
# because the zero mask masks out the observation and set them as zeros.
plt.figure()
plt.plot(traj_base_train[0][:,0], traj_base_train[0][:,1])
plt.plot(traj_true_train[0][:,0], traj_true_train[0][:,1])

traj_true_pred_train = traj_loss_mask_train[0].unsqueeze(1)*traj_true_train[0]
plt.plot(traj_true_pred_train[:,0], traj_true_pred_train[:,1])
```

Out[6]:

```
[<matplotlib.lines.Line2D at 0x7ff45e999c18>]
```


12
10
8
6
4
2
0
0
1
2
3
4
5
6
7
8