System Identification of Rover Dynamics

Bret Witt

Project Sponsor: Dr. Frances Zhu, Hawaii Institute of Geophysics and Planetology

ICS 496

Spring 2023

Overview

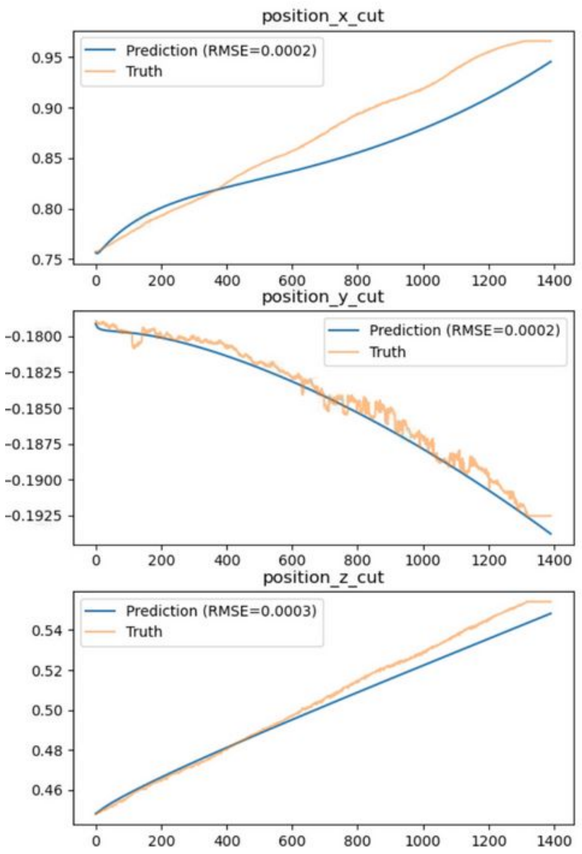
A proposed solution for improving onboard rover trajectory prediction involves finding ways to incorporate motor and IMU signals in place of expensive vision-bound methods. **We developed software that trains and evaluates ML models that attempt to accomplish this task**

Team Methodology

We use an Agile-like iterative methodology. The team is one CS graduate, one PhD student, and one faculty professor. We met once a week and individually as required

Dataset and Training

We use Python (NumPy, Pandas, SKLearn, and PyTorch). A dataset provided to us by the NASA JPL/Caltech is used to train these models. Standardized testing is used to evaluate the models against eachother




```python
from common import CADREDataset
from common import data
from torch.utils.data import random_split
from model import SysIDModel
from sklearn.linear_model import LinearRegression

# See SysIDModel to understand more vvv
class LinearRegressionModel(SysIDModel):
    def __init__(self, files, train_ratio):
        self.model = LinearRegression()
        super().__init__(files,train_ratio)

    def forward(self,X):
        return self.model.predict(X)

    def train(self):
        X = self.train_set.X
        y = self.train_set.y
        self.model.fit(X,y)

# Setup model with all pickle files, shuffled
import glob
files = glob.glob("data/*_refined.p")


model = LinearRegressionModel(files,0.75)

#model = NeuralNetworkModel(files,0.8)

# Use implemented Linear Regression train
model.train()

# Use universal (across models) test
model.test()
```

Solution

Tasks Accomplished

- Built a **rapidly iterable** SysID model training and testing suite
- The **testing suite** analyzes the models ability to predict a trajectory and continues until it reaches 10% error per distance traveled
- Built a **data pipeline** to load the dataset and manipulate its features
- **Evaluated 20 candidate methods** of ML models
- Built a stronger theoretical ML foundation for the project

Challenges

- Model inputs & outputs constantly change, requiring a lot of rewrites
- Python is prone to silent errors and NumPy is generally unreadable/unmaintainable
- Data processing is a huge part of ML but is very time consuming

Learnings

- Working with **cross-domain teams** and applying knowledge has unique challenges
- **Unit tests** are great tools to guard against silent errors and accidently changing behavior
- Dedicate a lot of time to **processing data correctly**, the first time

INFORMATION & COMPUTER SCIENCES

UNIVERSITY *of* HAWAIʻI *at* MĀNOA