1. Mechanics
2. Advanced Python
3. Ubuntu
4. ROS

1.1 Python Library http://www.pydy.org/documentation.html

bivariate_normal:

Simple Pendulum: https://projects.skill-lync.com/projects/Simulation-of-a-Simple-Pendulum-on-Python-95518

https://stackoverflow.com/questions/44945111/how-to-efficiently-compute-the-heat-map-of-two-gaussian-distribution-in-python

https://jakevdp.github.io/PythonDataScienceHandbook/04.12-three-dimensional-plotting.html

Drawing Sphere:
https://stackoverflow.com/questions/32424670/python-matplotlib-drawing-3d-sphere-with-circumferences

Pendulum in Python: https://projects.skill-lync.com/projects/Simulation-of-a-Simple-Pendulum-on-Python-95518

SymEngine Installation: add.h not found?

https://www.boost.org/doc/libs/1_61_0/more/getting_started/unix-variants.html

ROS & Python GYM

https://robohub.org/machine-learning-with-openai-gym-on-ros-development-studio/

Python Gym Tutorial: https://gym.openai.com/docs/

https://stackoverflow.com/questions/43397162/show-matplotlib-plots-in-ubuntu-windows-subsystem-for-linux

Way forward:

1. DST/Unimelb Papers
2. MR's Paper
3. MR's Code
4. Robotics
5. Python
6. CVXOPT Samples
7. Linux
8. Git Hub tutorial (Pull, Push)

Robotics: http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/

Achieved:

1. Dilinium Installation
2. Boost
3. MR's Paper
4. Space Flight/Robotics

Ubuntu

TAB: Auto-complete

Right Click: Paste

Dual Boot: http://dailylinuxuser.com/2015/11/how-to-install-ubuntu-linux-alongside.html

Desktop: https://www.zdnet.com/article/how-to-run-run-the-native-ubuntu-desktop-on-windows-10/

Common Functions

http://www.sfu.ca/~ssurjano/optimization.html

Way forward:
1. RP to Git Hub-Done
2. Field D* Code
3. MR Code Installation

Engineering
Vehicle Dynamics, Communication & Control

Scientific
Proof

System Point of View
Input, Process & Output
Point Vs Policy

Achieved: Input, Process & Output
Combined Research Paper, References from Champ version and Behzad's.

Vehicle Simulation
https://projectchrono.org/

Robot Path Planning

https://atsushisakai.github.io/PythonRobotics/

https://github.com/yangmingustb/planning_books_1

Simulated Annealing:
http://apmonitor.com/me575/index.php/Main/LinearProgramming

Optimization Knowledge Upgrade

1. Robotics Lectures
2. QEcon Lectures
3. John Hedengren

Steepest Decent Animation - Gordon College

www.math-cs.gordon.edu/courses/ma342/python/steepest_decent-animation.py

This may be either a scalar value or an n-element vector. sf - **Gradient** step size ... AUTHOR: **Jonathan R**. **Senning** <jonathan.senning@gordon.edu

16-April-2019 DST

(most recent call last):
File "C:\ProgramData\Anaconda3\lib\site-packages\psutil_pswindows.py", line 635, in wrapper
return fun(self, *args, **kwargs)
File "C:\ProgramData\Anaconda3\lib\site-packages\psutil_pswindows.py", line 705, in cmdline
ret = cext.proc_cmdline(self.pid)
PermissionError: [WinError 5] Access is denied

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
File "C:\ProgramData\Anaconda3\lib\site-packages\anaconda_navigator\exceptions.py", line 75, in exception_handler
return_value = func(*args, **kwargs)
File "C:\ProgramData\Anaconda3\lib\site-packages\anaconda_navigator\app\start.py", line 113, in start_app
if misc.load_pid() is None: # A stale lock might be around
File "C:\ProgramData\Anaconda3\lib\site-packages\anaconda_navigator\utils\misc.py", line 384, in load_pid
cmds = process.cmdline()
File "C:\ProgramData\Anaconda3\lib\site-packages\psutil__init__.py", line 666, in cmdline
return self._proc.cmdline()
File "C:\ProgramData\Anaconda3\lib\site-packages\psutil_pswindows.py", line 638, in wrapper
raise AccessDenied(self.pid, self._name)
psutil._exceptions.AccessDenied: psutil.AccessDenied (pid=3264)

Solution:

https://github.com/ContinuumIO/anaconda-issues/issues/8765

23-April-2019 DST

Python to Executable: https://stackoverflow.com/questions/5458048/how-to-make-a-python-script-standalone-executable-to-run-without-any-dependency

3D Python: https://vpython.org/contents/docs/VisualIntro.html

Jupyter & Anaconda: Update Index from CLI Environment

Research Paper Review

Significance, Novelty, Technically Sound, Clarity

1. Ordering
2. Sections Linked
3. Flow, Important Elements, Motivation & Introduced
4. Edited
5. Improvements
6. Measurements & Baseline

IEEE

1. Body of knowledge?
2. Technically Sound?
3. Presentation: Structure & Expression?
4. References: Applicable & Sufficient?

Image Print

- Research Paper
- U Value from Potential Fields
- AI Integration
- Literature Review

3.1.2 Motion Model

The robot model is

$$\dot{x} = vcos(\phi)$$

$$\dot{y} = vsin((\phi)$$

$$\dot{\phi} = \omega$$

So, the motion model is

$$\mathbf{x}_{t+1} = F\mathbf{x}_t + B\mathbf{u}_t$$

where

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} cos(\phi)dt & 0 \\ sin(\phi)dt & 0 \\ 0 & dt \\ 1 & 0 \end{bmatrix}$$

dt is a time interval.

This is implemented at code

Its Jacobian matrix is

$$J_F = \begin{bmatrix} \frac{dx}{dx} & \frac{dx}{dy} & \frac{dx}{d\phi} & \frac{dx}{dv} \\ \frac{dy}{dx} & \frac{dy}{dy} & \frac{dy}{d\phi} & \frac{dy}{dv} \\ \frac{d\phi}{dx} & \frac{d\phi}{dy} & \frac{d\phi}{d\phi} & \frac{d\phi}{dv} \\ \frac{dv}{dx} & \frac{dv}{dy} & \frac{dv}{d\phi} & \frac{dv}{dv} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & -vsin(\phi)dt & cos(\phi)dt \\ 0 & 1 & vcos(\phi)dt & sin(\phi)dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Vehicle Dynamics Analysis

Unicycle Model, Sideway travel (BZ)
Achieved: References (DST), MR Model
Converting to Linear Model

04-May-2019 DSI

```
Spyder not launching
pip install PyQtWebEngine
```

Other's code???
Bootstrap, Numpy, Scipy, CVXpx all others libraries after initial implementiations
Move forward, Where is Comparison???

Luke Marsh Meeting

1. Ant: Drones
2. BZ: C…
3. Obstacles
4. Shepherding
5. Heuristics, Metrics (Ammunition)
6. Collaborative Environment

Flocking

Python
Target???, Source??? Known, Unknown (To be)
No Specific Path Planning (To be)
Agent Knowledge (Gossip to be)
Lost???, Threshold for connection
Exchange of Information
Obstacles, detected from Range
Communication Range & Detection Range (Seek Mode already)
Vehicle Profile (To be)
Gossip Algorithm
Genetic
Auction (Most recent)
Energy (To be)
MESA (2D only)

WF

Target detection
Follow Leader Keeps Global
Multi Paths
Multi Flocks
Graph
Starting Points
Joining Conditions (Cohesive coefficient)
De-Centralize Processing
Increasing communication Power
Encrypting Intercepting
ML, Vehicle Dynamics
Obstacles
Friendly forces
Drones (Information Sources)
Repulsion when Colliding
Coding & Agent Size Limitation

SPP-1

GA to find paths
ANN to find SA
Stage I: Single Vehicle, 10 Obstacles, Circular, 2D, Vehicle Profile, X, U
Stage II: Dynamic Obstacles (Range detection)
Stage III: Multi-Vehicle (Flocking, Starting Points, Different Routes)
Detection & Action with ANN

CM Observations:

1. Path
2. Vehicle Dynamics
3. Vehicle Profile and Environment
4. Boundary, Map Size
5. Processing Power, Battery, Number of Agents

Schedule

1. Shepherding
2. SPP Notebook
3. LR
4. Clone MR Environment for Comparisons
5. Flocking

SPP Comparisons:
Best Path, Lowest cost, safest Index?

Shepherding

Ref: Solving the Shepherding Problem
https://parasol.tamu.edu/groups/amatogroup/research/shepherding/

Flocking
Drone
Sweden
Dog, Cohesive Cluster
Drive
Attracted to Centre of Mass (COM)
65m
Target known
Small number harder COM movement
Safety range
Funnel effect, Ramp, Pin
V Shape Control (Misread)
3R (3.R) Japanese
Pc, Pd
Activation Distance (Stall factor)
Closer, Far (Better)
Steps
Path length, Collection, Driving, Shed
Line of Sight?? ← No Sound, NO
Dog knows (GCM), Sheep (GCM), Line of sight
Obstacles

Sheep (Collection of Agents), Shepherding (Control)
Force Vector
Total Travel (Minimizing distance travel)
Python 2 to 3 Issue, Area as Trapezoidal
Python version dependency
Reinforcement Learning, Rewarding
Obstacle Types: Funnel (I/O), Pin, Narrow Gap

Way Forward

- Changing posture
- PF
- Abstraction (Dynamics, Agents control Agents)
- Communication
- UNSW,(Ontology, Semantics, Symbolic, Non-Symbolic)
- Clustering
- General Prior Knowledge, Reinforcement Learning
- Three Sheep, Motor Bike Issue, Herd Optimal Number
- Horse Shoe Issue
- GOAL (Rob)

CM Observations:

- Sheep, Smart Agents, Dog Knows all Herd Details
- SPP, Vehicle non-uniform vehicle profile, Sheep all similar, different path, own control
- Sheep→Shephard???, Multi Flocks and Shephard
- Obstacles (Tangible or non-tangible)
- **Python**, Octave or Matlab
- Vector Method
- Cohesive Flock Definition
- Guiding Systems (GPS, Differential?, Drones, Friendly agents)
- **Time to Reach the Goal**
- PSO for Flocking

Overall

SPP: Best and Safest Path
Flocking: Best Flocking movement
Shepherding: Best Sherpherding

CM:
KERs
SPP LR
Flocking: LR
Shepherding: LR

Safe Path Planning (SPP):

- Vehicle Profile
- Stage I, II, III
- Safe, Shortest, Fastest Path
- LR (Paths & GPS, Autonomous, Our approach)
- PF with DST-Cost Function, Horse Shoe effect, Ant's Obstacles

Shepherding-Strombom Paper:

- 3Ra???
- Shep
  - Start
  - Collect Agents (Side to Side, V Shape better)
  - Cohesive Flock
  - Drive Flock
  - Driving/Collecting
  - Reach Goal
- Centre of Mass (Local & Global)
- Blind Zone

PF-JP-Latex

- Latex Formatting: https://tex.stackexchange.com/questions/74353/what-commands-are-there-for-horizontal-spacing

- https://www.latex-tutorial.com/tutorials/amsmath/

Writing – Paragraphs

https://libguides.newcastle.edu.au/writing-paragraphs/types

https://aso-resources.une.edu.au/academic-writing/paragraphs/

Vehicle Models

1. https://github.com/MPC-Berkeley/barc/wiki/Car-Model
2. iLQG/DDP Tassa Model
3. Atsushi Sakai Model
4. PyDy Model
5. Charles Tytler Model

DRH Presentation:

1. **Github**
2. **Octave PF Algorithm**
3. Anaconda/Ubuntu
   - Spyder
   - JP Notebook
4. Overleaf

Tasks
1. SPP_JP Notebook
2. Performance Analysis

Python Script Index

1. ML & AI (Tensor Flow, SciKit)
2. GA
3. SA
4. PSO
5. Gym
6. Simple Pendulum with Movie Scripting
7. Dynamic Programming
8. Modular with Environment (FaSTrack???)
9. Quad 6 DOF
10. Gradient Descent Check
11. Multi-Processing
12. Miquel's Code
13. Python Robotics by Atsushi Akai
14. Rosenbrock Demo
15. Environment, Surface, Heatmap Plots
16. Extended Kalman Filter
17. Convex Optimization
18. Numerical Methods

Octave Scripts

1. PF
2. iLQG Car Demo (Iterative Linear Quadratic Gaussian)
3. Environments

Features

1. Conversion to papers with JP Notebook
2. Debug Capability
3. Other Language Integration
4. Possible DST/AVS Python Library

MS Surface Keyboard Issue:
https://answers.microsoft.com/en-us/surface/forum/surfbook-surfperf/surface-book-cursor-keeps-jumping-around-and/57a42636-f1ac-4cf0-8eff-e21ddf9c3267

Literature Review Evaluation:
http://www.indiana.edu/~educy520/sec5982/week_2/pyrczak99.pdf

PF Path Planning & Vehicle Profile

- Local Minima
- Gradient
- Symbolic Form
- Integration with Main Paper on DDF
- AI techniques
- Comparison Methods
- Metrics of comparison, with what, how & why, where difference

WF:

1. Matlab Installation
2. Paper review
3. Method comparison
4. Vehicle Dynamics Modelling
5. Testing PF & Vehicle Profile

SSP & VP

Option 1: Unicycle
Option 2: Theta Change Rate
Option 3: Overall Optimized route

Another Change:
Gradient Descent vs PSO approach

1.1: Weekend
2.1: Chat with DMR
3.1: Needed? Or Prove that it works in all states
4.1: JP Notebook
5.1: Error Bugs Correction

DMR & DCM

Relative Angle (Option II)

High Level (DDP), Low Level Processing (PF)

Local Minima

Complexity

Stage I: Static, Stage II: Dynamic

Stage III: Multi-Body

Scaling

Street Maps, Waypoints, Earth & Maps, Obstacles

Pre-defined

Sands & Actuator Failure

Control Signal Frequency

Backing

What are threats & Where in DCM?