## Contents

1 Software 3
  1.1 ROS ............................................ 3
  1.2 Gazebo ....................................... 14
  1.3 ROS and Gazebo ............................... 17
  1.4 Project Chrono ................................. 22
  1.5 Polysync ....................................... 25
  1.6 julia ............................................ 26
  1.7 git .............................................. 40
  1.8 Documentation ................................. 48
  1.9 LaTeX .......................................... 54
  1.10 C ............................................... 58
  1.11 Operating Systems ........................... 58
  1.12 misc .......................................... 60

2 Hardware 67
  2.1 LiDAR Devices ................................. 67

3 Huck's notes 71
  3.1 Navigation and Guidance of Aerospace Vehicles (AERO 584) ............................. 71
My PhD project involves using model predictive control to avoid moving obstacles.

Contents:
1.1 ROS

1.1.1 Installing ROS

From Pre-Built Debiants

For ROS kinetic:

```bash
sudo apt-get install ros-kinetic-desktop-full
```

From source

Additional steps/notes to the link above.

1. Remove any sourced files from the `bash.bashrc` files.
2. 2.1: after making the `catkin_ws`

   Run:

   ```bash
catkin init
```

   **Note:** make sure that there are no other external workspaces that you are building on

3. Resolve Dependencies

   Before you build the `catkin_ws` make sure that you have the required dependencies!

   ```bash
   rosdep install --from-paths src --ignore-src --rosdistro kinetic -y
   ```
Potential Issues:

Using the second option (Desktop Install) for kinetic, the following issues may occur:

Note: potential error:

```
/home/febbo/ros_catkin_ws/src/opencv3/opencv_contrib/rgbd/src/odometry.cpp:41:45:
  fatal error: unsupported/Eigen/MatrixFunctions: No such file or directory
compilation terminated.
```

The fix is to changed line 41 in odometry.cpp to (or where eigen3 is):

```
#include <eigen3/unsupported/Eigen/MatrixFunctions>
```

Note: potential error:

```
c++: error: /home/febbo/ros_catkin_ws/build_isolated/qt_gui_cpp/src/qt_gui_cpp
  -->cpp_shiboken/libqt_gui_cpp_shiboken/qt_gui_cpp_recursivepluginprovider.
  -->wrapper.cpp: No such file or directory
  fatal error: no input files
  compilation terminated.

<== Failed to process package 'qt_gui_cpp':
```

After posting an issue here. The fix was identified to be removing shiboken with:

```
sudo apt-get remove libshiboken*
```

An issue with the above fix is that if you try updating your ros dependencies and it installs shiboken. So, installing Qt 5 with:

```
sudo apt-get install qt5-default
```

Note: potential error <https://github.com/ros-planning/navigation/issues/579>:

```
-- Found Bullet: /usr/lib/x86_64-linux-gnu/libBulletDynamics.so
CMake Error at /usr/share/cmake-3.5/Modules/FindPackageHandleStandardArgs.cmake:148
  (message):
  Could NOT find SDL (missing: SDL_LIBRARY SDL_INCLUDE_DIR)
Call Stack (most recent call first):
```

Note: potential error

```
CMake Error at /usr/local/lib/cmake/gazebo/gazebo-config.cmake:172 (find_package):
  By not providing "Findignition-math3.cmake" in CMAKE_MODULE_PATH this
Installing missing Dependencies

**Note:** This is a truly magical command that works well (even when doing a source build)!!

Go to the `catkin_workspace` and type:

```
rosdep install --from-paths src --ignore-src -r -y
```

Uninstalling ROS

**From Pre-Built Debians:**

If you’re running Ubuntu, and installed ROS with `apt-get`, the first step would be:

```
sudo apt-get remove ros-`
```

Remove configuration files:

```
sudo apt-get purge ros-`
```

Remove packages that are no longer required:

```
sudo apt autoremove
```

**From Source**

TODO

1.1.2 Resources and Tutorials

**Tutorials and Resources**

- Node Tutorial Python
- Writing a Publisher Subscriber
- Publishing the State Info
- Navigation Tutorials
- Turn your package into a python module
- MIT stuff
- basic ROS concepts
- best practices
- Installing Packages
- Navigating the File System

Other packages:
**Great Commands**

To check for any missing dependencies:

```
sudo apt-get install libfcl-dev
```

To automatically install missing dependencies:

```
rosdep install --from-paths . --ignore-src --rosdistro kinetic -y
```

**Potential Issues**

*permission denied: .gvfs*

Fix unmount Gnome virtual file system and delete the mounting point:

```
sudo umount /home/febbo/.gvfs
sudo rm -rf .gvfs/
```

**Useful Questions**

- good path following code
- https://github.com/osrf/homebrew-simulation/issues/167

### 1.1.3 Useful ROS Packages

**tf2**

This package uses `geometry_msgs` for some of the messages.

**Useful functions**

**Warning:** use `tf2` not `tf`

**Commands**

Check tf tree:

```
roswtf
rosrun tf tf_monitor
rosrun tf view_frames
```

to print a particular transformation:
Joint States

Terminology

1) Quaternions: a way to describe rotations that does not have singularities. It has four numbers, the first three describe an axis and the third is the amount of rotation about that axis.

The first three components are imaginary:

\[ q = xi + yj + zk + w \]

The magnitude of a quaternion is:

\[ \text{magnitude} = \sqrt{x^2 + y^2 + z^2 + w^2} \]

where:

\[
\begin{align*}
    x &= x / \text{magnitude} \\
    y &= y / \text{magnitude} \\
    z &= z / \text{magnitude} \\
    w &= w / \text{magnitude}
\end{align*}
\]

http://www.euclideanspace.com/maths/geometry/rotations/conversions/quaternionToEuler/

- Almost never have to set these manually in ROS!
- Programming Quaternions
- ' a calculator <http://quat.zachbennett.com/>'
- ' a Gazebo example <http://gazebosim.org/tutorials?tut=drcsim_atlas_siminterface&cat=>'

Tutorials and Resources

- tutorials
- FAQ
- docs
- setting up the frames: for SLAM
- setting up your robot using tf

Example Resources

- python omniaxbot

Issues

- quaternions
- using tf
Potential Errors

Note: potential error:

```
fatal error: tf/transform_broadcaster.h: No such file or directory
compilation terminated.
```

Use tf2! The new version of tf.

RViz

Official visualization tool for ROS
http://wiki.ros.org/rviz/Troubleshooting

obstacle_detector

To use this package you must install Armadillo C++.

Installing Armadillo C++

Put these programs in the /opt folder

First you need OpenBlas

To get it type:

```
sudo git clone https://github.com/xianyi/OpenBLAS.git
```

Then go into the OpenBlas folder and type:

```
sudo make
```

Get devel files from Debian repos as per:

```
sudo apt update
apt search openblas
sudo apt install libopenblas-dev
sudo update-alternatives --config libblas.so.3
sudo apt install cmake liblapack-dev
apt search arpack
sudo apt install libarpack2-dev
```
Then you need LAPACK

To get it type:

```bash
sudo git clone https://github.com/Reference-LAPACK/lapack-release.git
```

Then go into the `lapack-release` folder and type:

```bash
sudo mv make.inc.example make.inc
sudo make
```

**Warning:** Then you may run into this issue:

```bash
./liblapack.a ../../librefblas.a
make[2]: Leaving directory '/opt/lapack-release/TESTING/LIN'
Testing COMPLEX16 LAPACK linear equation routines
./LIN/xlintstz < ztest.in > ztest.out 2>&1
SEP: Testing Symmetric Eigenvalue Problem routines
./EIG/xseigtstz < sep.in > zsep.out 2>&1
Makefile:467: recipe for target 'zsep.out' failed
make[1]: *** [zsep.out] Error 139
make[1]: Leaving directory '/opt/lapack-release/TESTING'
Makefile:42: recipe for target 'lapack_testing' failed
make: *** [lapack_testing] Error 2
```

As per this site the stack size can be increased with:

```bash
ulimit -s 100000
```

Then run:

```bash
sudo make
```

And it should compile fine!

---

**Note:** can also install SuperLU and ARPACK etc. ‘as per<https://github.com/conradsnicta/armadillo-code/tree/8.300.x>’

---

Now we can get Armadillo C++

To get it type:

```bash
sudo git clone https://github.com/conradsnicta/armadillo-code.git
```

Then go into the `armadillo-code` folder and type:

```bash
sudo cmake .
sudo make
```

**Eigen**

a C++ library that I use for quaternion transformations, to get started

1.1. ROS
Note: when going to use it with ROS there is no FindEigen.cmake on Ubuntu, so you need to include it in your package by default.

http://wiki.ros.org/jade/Migration#Eigen_CMake_Module_in_cmake_modules

**roscpp**

A C++ implementation of ROS that allows users to quickly interact with ROS.

**rospy**

**Common Messages**

a meta package for common messages

**message_filters**

Make the decision to use data based off of the time stamp. Basically sequence the data

**PCL**

Point cloud library

**Ackermann Group**

Group that supports Ackermann steering geometry.

**navigation**

install with:

```bash
sudo apt-get install ros-kinetic-navigation
```

**hector_localization**

install with:

```bash
sudo apt-get install ros-kinetic-hector-localization
```

**ros_control**

install with:

```bash
sudo apt-get install ros-kinetic-ros-control ros-kinetic-ros-controllers
```
pointcloud_to_laserscan

install with:

```
sudo apt-get install ros-kinetic-pointcloud-to-laserscan
```

teleop_twist_keyboard

Cartographer

A Google map builder

laser_pipeline

Meta-package for laser data processing

robot_localization

Provide state estimates given sensor data

robot_state_publisher

- tutorial
- load new parameters on the server
- Troubleshooting
- messages being blocked
- use_sim_time = false

Requires:

- joint_state_publisher

clock

For a ROS node to use the /clock topic set the /use_sim_time parameter before node is started

ControllIt!

MoveIT

Install

ROS Kinetic on Ubuntu:

```
sudo apt-get install ros-kinetic-moveit
```
Using With Kinetic

Go to this link and download the PDF for instructions.

URDF

Unified Robot Description Format (URDF), an Extensible Markup Language (XML) format (a markup language (can document revisions) that is both machine and human readable)

catkin-tools

Lots of useful info on setting up workspaces.

1.1.4 Useful Examples

ADAS

Great example for controlling autonomous vehicles.

ROS tutorial for Autonomous Cars

http://www.robotigniteacademy.com/en/

IPA320

Very useful resource for using ROS with Gazebo

- https://github.com/ipa320

Potential Vehicle Models

- robots.ros.org
- http://wiki.ros.org/Robots/Husky
- http://wiki.ros.org/Robots/Grizzly
- http://wiki.ros.org/Robots/Jackal

NVIDIA’s Driveworks

https://developer.nvidia.com/driveworks
1.1.5 My Personal Build

Packages required for MAVS

as of 8/22/2017 on top of ros-kinetic-full:

```bash
sudo apt-get install ros-kinetic-controller-manager
sudo apt-get install ros-kinetic-joint-state-controller
sudo apt-get install ros-kinetic-effort-controllers
sudo apt-get install ros-kinetic-teleop-twist-keyboard
sudo apt-get install ros-kinetic-pointcloud-to-laserscan
sudo apt-get install ros-kinetic-velodyne-description
```

Packages used in my catkin_ws

as of 8/22/2017 on top of ros-kinetic-full:

```bash
sudo apt-get install ros-kinetic-move-base
sudo apt-get install ros-kinetic-gmapping
sudo apt-get install ros-kinetic-amcl
sudo apt-get install ros-kinetic-ackermann-msgs
sudo apt-get install ros-kinetic-transmission-interface
sudo apt-get install ros-kinetic-moveit-ros-move-group
sudo apt-get install ros-kinetic-robotnik-sensors
sudo apt-get install ros-kinetic-robot-pose-ekf
sudo apt-get install ros-kinetic-moveit-kinematics
sudo apt-get install ros-kinetic-moveit-ros-visualization
sudo apt-get install ros-kinetic-moveit-planners-ompl
sudo apt-get install ros-kinetic-teb-local-planner
sudo apt-get install ros-kinetic-joint-limits-interface
sudo apt-get install ros-kinetic-map-server
sudo apt-get install ros-kinetic-robot-state-publisher
sudo apt install ros-kinetic-rqt
```

To use Autoware

added:

```bash
sudo apt-get install ros-kinetic-nmea-msgs ros-kinetic-nmea-navsat-driver ros-kinetic-
→sound-play ros-kinetic-jsk-visualization ros-kinetic-grid-map ros-kinetic-gps-common
sudo apt-get install ros-kinetic-controller-manager ros-kinetic-ros-control ros-
→kinetic-ros-controllers ros-kinetic-gazebo-ros-control ros-kinetic-joystick-drivers
sudo apt-get install libnlopt-dev freeglut3-dev qtbase5-dev libqt5opengl5-dev libssh2-
→1-dev libarmadillo-dev libpcap-dev gksu libgll1-mesa-dev libglew-dev python-wxgtk3.0
```

```bash
sudo apt install ros-kinetic-roslint
sudo apt-get install ros-kinetic-joint-state-publisher
sudo apt-get install ros-kinetic-controller-manager
```

**my .bashrc**

sourced:

```bash
```

1.1. ROS
source /opt/ros/kinetic/setup.bash
source /home/febbo/Documents/workspace/tutorials/catographer_ws/install_isolated/setup.bash
source /home/febbo/Documents/workspace/tutorials/velodyne_simulator/devel/setup.bash
source /home/febbo/Documents/workspace/tutorials/kobuki_desktop/devel/setup.bash
source /home/febbo/.julia/v0.6/MAVs/catkin_ws/devel/setup.bash
source /home/febbo/Documents/workspace/tutorials/mastering_ros/chapter_3_codes/devel/setup.bash
source /home/febbo/catkin_ws/devel/setup.bash
alias tf='cd /var/tmp && rosrun tf view_frames && evince frames.pdf &'
. ~/.bash_profile
source /home/febbo/ros_catkin_ws/install_isolated/setup.bash

1.1.6 Potential Errors

Warning: Potential Error:

```
fatal error: tf2/LinearMath/Quaternion.h: No such file or directory compilation terminated.
```

It is trying to find some file and it cannot. The problem may be that the header files where not specified! To fix:

```
include_directories(
    ${catkin_INCLUDE_DIRS}
)
```

1.2 Gazebo

1.2.1 Installation

Automatically installed after following the ROS installation instructions.

Make sure after you install it you source the setup.sh file in the .bashrc script:

```
source /usr/share/gazebo-7/setup.sh
```

Also source any setup.bash scripts that are in your projects:

```
source /home/febbo/.julia/v0.6/MAVs/workspace/devel/setup.bash
```

Also in the setup.sh file in the `usr/share/gazebo-7/` directory, add to the resource path if you need to as:

```
export GAZEBO_RESOURCE_PATH=/usr/share/gazebo-7:/usr/share/gazebo_models:/usr/share/gazebo-7/worlds:${GAZEBO_RESOURCE_PATH}
```

In the above example we also added the `/worlds` folder.

Note: Using the gui to create models is limited because the default full version of ros kinetic does not let you edit the models after you save them. So, then I started to learn how to make the models directly using XML. The only way to get around this is to build from source but then you won’t be able to use any ROS Ubuntu package related to Gazebo from the ROS deb repo.
From source (option A)

Follow these instructions

Additional notes:

- **Step 1:**
  To get the Gazebo-8 stable branch:
  ```bash
  /opt$ sudo hg clone https://bitbucket.org/osrf/gazebo /opt/gazebo -b gazebo8  
  ```
  To get the Gazebo-7 stable branch:
  ```bash
  /opt$ sudo hg clone https://bitbucket.org/osrf/gazebo /tmp/gazebo -b gazebo7  
  ```

- **Step 3:**
  Configure Gazebo:
  ```bash
  sudo mkdir build
  cd build
  sudo cmake ..
  ```
  Build Gazebo:
  ```bash
  /opt/gazebo/build$ sudo make -j4
  ```
  Install Gazebo:
  ```bash
  sudo make install
  ```

- **Step 4:**
  Set up environmental variables and

From source (option B) -preferred for Gazebo-7

http://gazebosim.org/tutorials?tut=install_dependencies_from_source

Potential Issue

**Note:** If not sourced properly:

```bash
febbo@febbo-HP-ZBook-17-G2:$ gazebo
gazebo: error while loading shared libraries: libgazebo_common.so.7: cannot open
  --shared object file: No such file or directory
```

Then try:

```bash
febbo@febbo-HP-ZBook-17-G2:$ find / -name 'libgazebo_common.so.7' 2>/dev/null
/usr/local/lib/libgazebo_common.so.7
/tmp/gazebo/build/gazebo/common/libgazebo_common.so.7
```
And updated the `/etc/bash.bashrc` file appropriately:

```bash
gksu gedit /etc/bash.bashrc
```

By adding (in this case):

```bash
export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

**Note:** when trying to start Gazebo from the command line:

```
gazebo: error while loading shared libraries: libsdformat.so.5: cannot open shared object file: No such file or directory
```

### 1.2.2 Terminology

1) Xacro:

(Xacro) (XML macros) an XML macro language useful for shorter robot descriptions.

- `xacro:include`: import content from another file
- `property`: define constants `$\{property\_name\}`
- `xacro:macro`: include the file where the macro is and call it using the macro’s name and filling in the required values

2) SDFormat

An XML format that describes environments for robot simulators visualizations and controls. Originally part of Gazebo but now is stable enough that it stands alone.

### 1.2.3 Basics

#### Model Structure

Make sure that the model is structured correctly. An issue that was ran into is described here where I tried to put a model in a lower level directory. Models need to go just below `catkin/src/`, then `setup.sh` must be modified as:

```bash
export GAZEBO_MODEL_PATH=/usr/share/gazebo-7/models:/home/febbo/catkin_ws/src/:
udy://{GAZEBO\_MODEL\_PATH}
```

#### Model State Info

Using `rospy` to get the model states.

#### World Files

**Note:** When saving a world file, put it into the folder with the other world files!
1.2.4 Tutorials:

• Add Sensor to A Robot
• Gazebo Plugins
• Getting SDF information for a model
• Control Plugin
• 3D models

1.2.5 Resources:

• 3D Warehouse: a place to grab CAD models
• Possible Model for HMMWV
• simplify a .dae file
• 3Delicious
• Free 3D

1.2.6 Useful Software

• Blender

1.2.7 Issues

• importing Collada files

1.3 ROS and Gazebo

1.3.1 Installation of gazebo_ros_pkgs

From Pre-Built Debians:

For ROS Kinetic

```
sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
```

Gazebo 8 and ROS kinetic (preferred)

remove all Gazebo and ros binaries:

```
sudo apt-get remove --purge '.*gazebo.*' '.*sdformat.*' '.*ignition-math.*' '.*ignition-msgs.*' '.*ignition-transport.*'
sudo apt-get remove ros-*
sudo apt-get purge ros-*
sudo apt autoremove
```
install dependencies:

```bash
sudo apt-get update
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool
    --build-essential
sudo apt-get install -y wget
sudo apt-get install -y lsb-release
sudo apt-get install -y sudo
sudo apt-get install -y mesa-utils
sudo apt-get clean
```

Get gazebo and ROS binaries:

```bash
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key
    421C365BD9FF1F717815A3895523BAEEB01FA116
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'
wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install ros-kinetic-ros-base
sudo apt-get install -y gazebo8
sudo apt-get install -y ros-kinetic-gazebo8-ros-pkgs
sudo apt-get install -y ros-kinetic-fake-localization
sudo apt-get install -y ros-kinetic-joy
sudo apt-get install -y ros-kinetic-robot-state-publisher
sudo apt-get install -y ros-kinetic-rviz
sudo apt-get clean
sudo apt autoremove
```

Other packages (I am currently using...updated):

```bash
sudo apt install rosbash
sudo apt-get install ros-kinetic-ros-core
sudo apt-get install ros-kinetic-turtlesim
sudo apt-get install ros-kinetic-xacro
sudo apt install ros-kinetic-teleop-twist-keyboard
sudo apt install ros-kinetic-velodyne-description
sudo apt install ros-kinetic-controller-manager
sudo apt install ros-kinetic-joint-state-controller
sudo apt install ros-kinetic-rqt-gui
sudo apt install ros-kinetic-effort-controllers
sudo apt install ros-kinetic-pcl-ros
sudo apt install ros-kinetic-pointcloud-to-laserscan
```

http://wiki.ros.org/velo2cam_gazebo

Initialize rosdep:

```bash
sudo rosdep init
rosdep update
```

**From Source**

After installing ROS from source with the Desktop Install you will also need to put the following packages into the `/ros_catkin_ws/src` folder:
git clone https://github.com/ros-simulation/gazebo_ros_pkgs -b kinetic-devel
git clone https://github.com/ros-perception/image_common -b hydro-devel
git clone https://github.com/ros-controls/control_toolbox -b kinetic-devel
git clone https://github.com/ros-controls/ros_control -b kinetic-devel
git clone https://github.com/ros-controls/realtime_tools -b kinetic-devel

Then rebuild the workspace:

```
./src/catkin/bin/catkin_make_isolated --install
```

To test installation, run:

```
 roscore &
 rosrun gazebo_ros gazebo
```

## 1.3.2 Resources

### Examples of using ROS and Gazebo

Get the position of a model:

```
gazebo::math::Pose pose = this->model->gazebo::physics::ModelState::GetPose();
```

Pose to initialize the model to:

```
msgs::Set(msg.mutable_pose(),
  math::Pose(math::Vector3(1, -2, 0), math::Quaternion(0, 0, 0)));
```

### Writing Publisher Subscriber

### Messages

### Example Resources

- [gazebo_and_ros_control](#): with a tutorial
- [srs](#): a very useful example of using python for control

### Tutorials

- [ROS and Gazebo: using roslaunch](#)
- [a blog post](#) to help get started
- [A helpful resource](#): lots of questions answered
- [Gazebo ROS API for C-Turtle](#)
- [ROS Plugins](#)
- [a blog post](#)
- [another blog post](#)
- [JointTrajectory](#): good for controlling a robot when you don’t care about the physics

### 1.3. ROS and Gazebo


Tools

- gazebo_ros_pkgs or the look at the source code
- pygazebo: a python interface to Gazebo
- gazebo_ros_p3d : Ground Truth Pose and Rates Interface

Potential Packages to Fork for Closing the Loop

- ADAS
- Georgia Tech
- https://github.com/CPFL/Autoware

**car_demo**

A recent package by Tully Foote with a Gazebo model of Mcity using ROS Kinetic and Gazebo 8.

Additional Install Notes

**Required Packaged (on top of Desktop Install)**

(for a source build) clone into the `src` directory:

```
git clone https://github.com/ros-planning/navigation.git
git clone https://github.com/ros-drivers/joystick_drivers.git
```

**Joystick**

Usually when you plug in the Joysick it automatically sends that can be viewed with either:

```
cat /dev/input/js0
```

Or:

```
cat /dev/input/js1
```

**Note:** potential error: libusb is not installed on the machine. I ran into this issue, where the ps3joy package failed with this error:

```
CMake Error at /usr/share/cmake-3.5/Modules/FindPkgConfig.cmake:578 (message):
  None of the required 'libusb' found
Call Stack (most recent call first):
  CMakelists.txt:6 (pkg_search_module)
```

The fix is to install libusb:

```
sudo apt-get install libusb-dev
```
Additional Info (not needed)

install a program to test and calibrate:

```bash
sudo apt-get install joystick
```

to test:

```bash
jscal /dev/input/js0
jstest /dev/input/js0
```

**NVIDIA-DOCKER**

To show the current architecture, type:

```bash
dpkg --print-architecture
```

type:

```bash
wget -P /tmp https://github.com/NVIDIA/nvidia-docker/releases/download/v1.0.1/nvidia-
˓→docker_1.0.1-1_amd64.deb
sudo dpkg -i /tmp/nvidia-docker*.deb && rm /tmp/nvidia-docker*.deb
```

install nvidia-modprobe:

```bash
sudo apt install nvidia-modprobe
```

**Note:** potential error:

Error response from daemon: create nvidia_driver_375.66: error looking up volume_
˓→plugin nvidia-docker: plugin "nvidia-docker" not found.
See 'docker run --help'.

the fix is:

```bash
journalctl -n -u nvidia-docker
```

**Warning:** Restart computer after installing GPU software

1.3.3 Issues

Helpful Questions

- Using python to set ModelState in Gazebo
- publisher and subscriber in C code
- set position in gazebo
- Using python as a node

1.3. ROS and Gazebo
My Questions


1.4 Project Chrono

1.4.1 Background

Physics simulator organized as a set of modules that has a high-fidelity HMWWV model validated using Adams.

1.4.2 Install

Get the IRRlicht engine

Make sure that you have the required files:

```
sudo apt-get install cmake-extras cmake-qt-gui build-essential xorg-dev freeglut3-dev
```

After ‘downloading <http://irrlicht.sourceforge.net/?page_id=10‘ unzip:

```
sudo unzip /home/febbo/Downloads/irrlicht-1.8.4.zip -d /opt/irrlicht
```

Compile the Irrlicht Engine 1.8.4 as a shared lib:

```
/opt/irrlicht/irrlicht-1.8.4/source/Irrlicht$ sudo make sharedlib
/opt/irrlicht/irrlicht-1.8.4/source/Irrlicht$ sudo make install
```

Chrono

Install instructions

Go into /opt folder:

```
sudo mkdir chrono
cd chrono
sudo mkdir chrono_source
sudo mkdir chrono_build
cd chrono_source
```

Clone the develop branch into the /opt/chrono/chrono_source directory:

```
sudo git clone -b develop https://github.com/projectchrono/chrono.git
cd chrono
```

To configure:

```
sudo cmake-gui
```
Then to build (assuming ninja was used) go into the `build` directory and type:
Then you can modify permissions if desired:

```
sudo chmod -R 777 chrono_build/
```

The rest is old....

VEHICLE Module

Connecting to Gazebo

A repo was developed to connect Gazebo to Chrono.

Install

1) Install Gazebo 7 from source

2) run:

```
sudo apt-get install libogre-1.9-dev
```

1.4.3 Potential Fixes

Try updating glib with:

```
sudo apt-get update && sudo apt-get upgrade
```

1.4.4 Issues:

- https://groups.google.com/forum/#!topic/projectchrono/ETBnyS18MNM
- https://groups.google.com/forum/#!topic/projectchrono/1GI3RiZI-2A
- https://stackoverflow.com/questions/13978692/strcpy-sse2-unaligned-s-not-found
- https://sourceforge.net/p/irrlicht/bugs/375/

Issue might look something like:

```
/build$ ninja
[1/2] Building CXX object demo_Steerin....dir/demo_VEH_SteeringController.cpp.o
FAILED: /usr/bin/c++ -MMD -MT demo_SteeringController/CMakeFiles/demo_
  → SteeringController.dir/demo_VEH_SteeringController.cpp.o -MF demo_
  → SteeringController/CMakeFiles/demo_SteeringController.dir/demo_VEH_
  → SteeringController.cpp.o.d -o demo_SteeringController/CMakeFiles/demo_
  → SteeringController.dir/demo_VEH_SteeringController.cpp.o -c /home/febbo/Documents/
  → workspace/tutorials/chrono_tutorials/source/demo_SteeringController/demo_VEH_
  → SteeringController.cpp
/home/febbo/Documents/workspace/tutorials/chrono_tutorials/source/demo_
  → SteeringController/demo_VEH_SteeringController.cpp:22:37: fatal error: chrono/core/
  → ChFileutils.h: No such file or directory
```

(continues on next page)
https://github.com/projectchrono/chrono/issues/79
In this case, copy the Resources =========
  • https://vimeo.com/uwsbel
  • http://sbel.wisc.edu/Animations/

1.4.5 OLD
NOT NEEDED:
make a build directory:
sudo mkdir build
set up build environment:
sudo cmake CMakeLists.txt
make:
git clone git://git.code.sf.net/p/libosmscout/code libosmscout-code

1.4.6 Shared Pointers
http://api.projectchrono.org/shared_pointers.html

1.4.7 TEMP
gazonoVehicle.cc
//gazebo_ros includes NOTE because ros is not in/usr/include/ there are sourcing issues
#include <ros/callback_queue.h> #include <ros/subscribe_options.h> #include <ros/ros.h> #include "boost/thread/mutex.hpp"
#include <std_msgs/Float64.h>
Issue:
/usr/bin/ld: cannot find -lpthreads
https://stackoverflow.com/questions/30600978/cpp-usr-bin-ld-cannot-find-lnameofthelibrary

1.5 Polysync
Polysync does play with ROS
http://docs.polysync.io/flows/getting-started/
It talks to MATLAB http://docs.polysync.io/articles/development/ecosystem-tools/matlab-and-simulink-toolbox/
Also, it looks like it supports the Puck (along with many other sensors)
http://docs.polysync.io/sensors/velodyne-vlp-16-puck/
All in all it looks like the documentation was well made and it is yet another very useful tool and they claim if we follow their tutorial we can build a self-driving Kia Soul for under $10,000 (I think this excludes LiDAR and GPS etc.)! Looking through the wiki a bit it looks like the vehicle needs to only slightly be modified.

### 1.6 julia

#### 1.6.1 Getting Started with julia

Extensive julia documentation is available [here](#).

**Building julia release 0.5 - current**

Recently a new version of julia was released and it will be tested. The following directions also assume that julia was previously installed on the machine. If it was not, skip to step 3.

**Remove current version of julia**

1. get a graphical file manager with **ROOT ACCESS**:

   ```
gksu nautilus
   ```

   Also, may need to:

   ```
sudo apt install gksu
   ```

   • **NOTE**: be **very** careful in here!!

2. Type `ctrl+l` to type into location bar

   • now navigate to where julia is and delete the binaries (or bin), i.e. it might be in /usr/local/begin

**Add new version current version of julia**

3. unzip the .gz into the \opt folder (the place where by convention you'd put “optional” packages)

   for instance:

   ```
sudo tar -xvf /home/febbo/Downloads/julia-0.5.0-linux-x86_64.tar.gz -C /opt
   ```

4. Check it:

   ```
febbo@febbo-HP-ZBook-17-G2:/opt/julia-3c9d75391c/bin$ ./julia
   OR
febbo@febbo-HP-ZBook-17-G2:/opt/julia-3c9d75391c/bin$ /opt/julia-3c9d75391c/bin/julia
   _(_)_(_)_ | A fresh approach to technical computing
   (_____) (___) | Documentation: http://docs.julialang.org
   ___|___|___|___| Type "?help" for help.
   | | | | | | | | | | | Version 0.5.0 (2016-09-19 18:14 UTC)
   /|___|\_'_\_\_\_'_ | Official http://julialang.org/ release
   \___/ x86_64-pc-linux-gnu
   julia>
   ```

5. Now we need to create a symbolic link so that we can just type `julia` in the command line. There are three ways to do this (**just do the third option; it is permanent**):
1. create an “alias”
   • create an alias using the following command.
     type:
     ```
     alias julia='/opt/julia-3c9d75391c/bin/julia'
     ```

2. add `/opt/julia-3c9d75391c/bin` to your PATH
   • all directories contained in the PATH are accessible from ANYWHERE
   • you can see the current PATH variable by typing in your terminal.
     type:
     ```
     echo $PATH
     ```
   • you can redefine a variable in bash e.g. by doing..
     type:
     ```
     PATH=new definition here
     ```
     * (no spaces----> this is important)
   • add your julia path to the existing path, you can do that by saying...
     type:
     ```
     PATH=/opt/julia-3c9d75391c/bin:$PATH
     ```
     * the $PATH contains the old value, so you're basically adding your folder and a colon and then all the rest, into a new PATH variable

The Problem with The Above Two Options is that:

With both adding to the path or creating an alias, is that these changes are TEMPORARY the minute you close your terminal and open it again, you'll see that these changes have disappeared try it and see!
   • so to make it permanent, you actually have to do option 3

3. change a file called bashrc which is run every time a terminal starts
   • the file you need to edit should be /etc/bash.bashrc
     so type:
     ```
     gksu gedit /etc/bash.bashrc
     ```
     If it exists and it's the right file, you should see its contents inside gedit
   • go to the very end of that file
     – add the alias line...
     type:
     ```
     alias julia='/opt/julia-3c9d75391c/bin/julia'
     ```
     and press enter!!
6. Now close the terminal and type:

```
febbo@febbo-HP-ZBook-17-G2:~$ julia
```

So, it should be running!

7. If you are using Atom, make sure that you change the path in the config folder in settings:

```
juliaPath: "/opt/julia-3c9d75391c/bin/julia"
```

8. next time you download a new version of julia:

   • simply extract it under /opt in the same way and either replace the old one

   OR

   • if you want to keep both, you can just update your alias in /etc/bash.bashrc to point to the new one

**Building julia release 0.5 - old**

1. Type:

```
sudo apt -rm julia
```

2. Also, julia can be completely removed by deleting the ~/.julia folder.

Note: Can also remove PPA (according to this):

```
sudo apt install ppa-purge
```

   • although this does not seem to be useful.

**Fresh install instructions for UBUNTU:**

A. Follow these instructions.

*Or*

B. Type this terminal:

```
sudo add-apt-repository ppa:staticfloat/juliareleases
sudo add-apt-repository ppa:staticfloat/julia-deps
sudo apt-get update
sudo apt-get install julia
```
Bleeding Edge Version - previously used

I found that Julia is constantly being developed and that most of the developers do not make sure that every version is maintained, this can create issues when using particular packages. So, if you want to use the latest and greatest features, you might consider the bleeding edge version of Julia.

Fresh install instructions for UBUNTU:

A. Follow these instructions.

Or

B. Type this terminal:

```
sudo apt-add-repository ppa:staticfloat/julianightlies
sudo apt-add-repository ppa:staticfloat/julia-deps
sudo apt-get update
sudo apt-get install julia
```

Warnings

If you are getting warnings like this:

```
WARNING: Deserialization checks failed while attempting to load cache from /home/‐
˓→febbo/.julia/lib/v0.6/JuMP.ji.
WARNING: Module Lazy uuid did not match cache file.
INFO: Recompiling stale cache file /home/febbo/.julia/lib/v0.6/JuMP.ji for module ‐
˓→JuMP.
WARNING: Deserialization checks failed while attempting to load cache from /home/‐
˓→febbo/.julia/lib/v0.6/ReverseDiffSparse.ji.
WARNING: Module Lazy uuid did not match cache file.
INFO: Recompiling stale cache file /home/febbo/.julia/lib/v0.6/PyPlot.ji for module ‐
˓→PyPlot.
WARNING: Deserialization checks failed while attempting to load cache from /home/‐
˓→febbo/.julia/lib/v0.6/PyCall.ji.
WARNING: Module Conda uuid did not match cache file.
```

It is a precompilation failure; restart Julia

1.6.2 Useful Packages and Programs

This page includes details of the packages and programs that I am using in OCP.

Adding and Removing Packages in Julia

A useful description of the syntax for adding and remove packages in Julia can be found here.

All packages that I have on Julia

I have many packages and to configure them all can be tricky, so I include a list of commands that you can copy and past into Julia to get started. Make sure that you are restarting the both Julia and the terminal after things are installed.
Basics

```julia
Pkg.add("DataFrames")
Pkg.add("IJulia")
Pkg.add("Parameters")
Pkg.add("PkgDev")
Pkg.add("AmplNLWriter")
#Pkg.clone("https://github.com/JunoLab/Juno.jl")
Pkg.add("HDF5")
Pkg.build("HDF5")
Pkg.add("SymPy")
Pkg.add("Jacobi")
```

Math

I use:

```julia
Pkg.add("DiffBase")
Pkg.clone("https://github.com/Keno/Polynomials.jl")
Pkg.add("DifferentialEquations")
Pkg.add("Dierckx")
Pkg.add("ImplicitEquations")
Pkg.add("Interpolations")
```

Optimization

I use:

```julia
Pkg.add("JuMP")  # adds MathProgBase automatically
Pkg.add("Ipopt")
Pkg.add("CoinOptServices")
Pkg.add("NLopt")
Pkg.build("NLopt")
```

Plotting

Basically I use Plots.jl to interface with some of these:

```julia
Pkg.add("Conda")
ENV["PYTHON"]="
Pkg.add("PyPlot")
Pkg.add("Plots")
Pkg.build("PyPlot")
Pkg.add("ImageMagick")
Pkg.add("GR")
Pkg.add("Plotly")
Pkg.add("StatPlots")
Pkg.add("PlotRecipes")
Pkg.add("UnicodePlots")
Pkg.add("Gadfly")
Pkg.add("RDatasets")
Pkg.add("Winston")
Pkg.add("PGFPlots")
```

(continues on next page)
Pkg.build("PGFPlots")
Blink.AtomShell.install()
import Conda # to fix pyplot!!!
Conda.add("qt=4.8.5")

My Packages

I started these:

Pkg.clone("https://github.com/huckl3b3rry87/VehicleModels.jl")
Pkg.clone("https://github.com/huckl3b3rry87/NLOptControl.jl")

Miscellaneous

probably do not need:

Pkg.clone("https://github.com/pwl/MovcolN.jl")
Pkg.add("Devectorize")
Pkg.add("FactCheck")

Useful Command After Installing Packages

Type:

Pkg.update()

Customizing Keybindings and Tab Completion

Info here

type:

import Base: LineEdit, REPL

const mykeys = Dict{Any,Any}(
    # Up Arrow
    "\e[A" => (s,o...)->(LineEdit.edit_move_up(s) || LineEdit.history_prev(s, LineEdit.
    →mode(s).hist)),
    # Down Arrow
    "\e[B" => (s,o...)->(LineEdit.edit_move_up(s) || LineEdit.history_next(s, LineEdit.
    →mode(s).hist))
)

function customize_keys(repl)
    repl.interface = REPL.setup_interface(repl; extra_repl_keymap = mykeys)
end

at REPLinit(customize_keys)
1.6.3 Basic Pkg. Usage Notes

Optimization

Currently the only optimization tool that is being tested is IPOPT.

1. IPOPT

It is very easy to get going using IPOPT in julia using IPOPT.jl.

Derivatives

JuMP.jl is one of the most useful packages for solving the OCP because it takes very fast automatic derivatives and it allows the user to easily set up optimization problems. So, with this tool there is no need to write out the complicated Jacobian and Hessian functions.

The documentation for this package can be found JuMP docs.

Some useful Methods are found by clicking.

Some useful commands Query upper and lower bounds of all constraints in the model:

```julia
JuMP.constraintbound(m::Model)
```

MathProgBase.jl

MathProgBase.jl.

Polynomial Division

There was an issue when JuMP was sent a term with polynomial division. This section deals with the attempt to use the Polynomials.jl to take care of the polynomial division on the front end, before the expressions are sent to JuMP.

Basic Functionality:

```julia
using Polynomials
Poly([1,0,3,4])
Poly(1 + 3x^2 + 4x^3)
```

Division Functionality:

```julia
P1 = Poly([1,2,3,5,7])
P2 = Poly([1,0,3])
P3 = div(P1,P2)
Poly(0.22222222222222218 + 1.6666666666666667x + 2.3333333333333335x^2)
```

Plots.jl

A very powerful plotting tool is Plots.jl. It took some time to get everything working because I was not using the same versions of the packages as the developers, but in the end it was work the time. It you run the code that I have listed below you should not have to deal with the setup issues that I had.

With PGFPlots:
Problem 1

I wasted a bunch if my time, could not recreate in a simpler example, but basically, when `plot()` should have worked it failed and when I changed the order of the terms in `plot()` it works.

Solution 1

EX:

```julia
for i in 1:k
    # plot!(dfs[i][:t],dfs[i][:SA]*180/pi,w=w2,label=label_string[i])
    plot!(dfs[i][:t],dfs[i][:SA]*180/pi,label=label_string[i],w=w2)
end
```

Problem 2

Segfault when attempting to plot after Conda update to Segfault with qt >=4.8.6 on ubuntu

I was running into an issue with Plots.jl after doing a `Pkg.update()`:

```julia
julia> using Plots
julia> plot(rand(4,100))
signal (11): Segmentation fault
while loading no file, in expression starting on line 0
unknown function (ip: 0x32735)
Allocations: 14606607 (Pool: 14605017; Big: 1590); GC: 25
Segmentation fault (core dumped)
```

Solution 2

Change back to old Conda:

```julia
import Conda
Conda.add("qt=4.8.5")
using PyPlot
plot(rand(10))
```

Check out this link.

**DifferentialEquations.jl**

**Useful Commands**

all of the timepoints:

```julia
sol[:]
```

Many times you might want to use a good interpolation. For example, the plots use something like:
To get 100 points from time $t_0$ to $t_{\text{end}}$ for the first component:

```
[sol(t)[1] for t in linspace(t0,tend,100)]
```

The array of timepoints for component $j$:

```
[sol[i][j] for i in 1:length(sol)]
```

**Parameters.jl**

Great package for working with parameters. More info can be found at [this link](#).

### 1.6.4 Making Your Own Packages

To create packages and modules click [this](#) to get started.

Or type:

```
using PkgDev
PkgDev.generate("VehicleModles","MIT")
```

Possible Next Steps:

- Make a github repository with this name (plus a .jl at the end of the name) [github.com](#)
  - Don’t add a README.MD automatically using github, there will be a conflict if you will make one from the using [sphinx-quickstart](#)
    - Or just pick one, don’t do both!
- Make some documentation getting_started_with_docs
  - Or, if everything is setup, type:
    ```
sphinx-quickstart
```
  - Then:
    ```
git remote add origin git@github.com:huckl3b3rry87/new_repo.jl
```

To tag:

```
using PkgDev
PkgDev.tag("VehicleModles")
```

To view the package:

```
$ cd ~/.julia/v0.6/VehicleModles && git show --stat
```

To recompile a package:

```
Pkg.test()
```

To go into the package directory:
In Juno, the workflow for building a package, you can do a Ctrl+J Ctrl+k to quit the current process and then use `Pkg` again.

it'll be in your lib folder and all setup to be used with using:

```
Pkg.test
Pkg.dir
etc
```

Then in order to run `Pkg.update()` without getting this error:

```
ERROR: Update finished with errors.
=> Package NLOptControl cannot be updated.
GitError(Code:ERROR, Class:Merge, There is no tracking information for the current__branch.)
etc....
```

Which is talked about here, you have to:

```
 julia> Pkg.checkout("NLOptControl")
INFO: Checking out NLOptControl master...
INFO: Pulling NLOptControl latest master...
INFO: No packages to install, update or remove
```

Making Modules

look here

Directories

Try:

```
@__FILE__
```

link

1.6.5 Macros

Detailed information on macros in julia is found here.

@def

Given some parameters:

```
using Parameters

@with_kw immutable Vpara @deftype Float64
  m = 2.6887e+03
end
pa = Vpara(); # initialize parameter set
```
Instead of unpackaging the same parameters each time in a nested function like this:

```julia
function outer_f(pa::Vpara)
    num = zeros(Float64, (10,1))
    for i in 1:10
        num[i] = inner_f(pa::Vpara,i)
    end
    return num
end
```

with:

```julia
function inner_f(pa::Vpara,i)
    @unpack_Vpara pa
    m + i + 0.1
end
```

We define a macro as:

```julia
macro def(name, definition)
    return quote
    macro $name()
        esc($($(Expr(:quote, definition))))
    end
    end
end
```

then redefine `inner_f` as:

```julia
@def inner_f2 begin
    m + i + 0.1
end
```

We also need to modify the `outer_f()` as:

```julia
function outer_f(pa::Vpara)
    @unpack_Vpara pa
    num = zeros(Float64, (10,1))
    for i in 1:10
        num[i] = @inner_f2
    end
    return num
end
```

The `@def` macro is functionally equivalent to copying and pasting the contents of `inner_f()` into `outer_f()`.

**Parallel Computing**


container:

```julia
rr=RemoteChannel()
# stores a value -> like a container that you can put something in and take something out
```
to add some processors:

```
addprocs()
```

calling a processor:

```
remotecall(function, proc, data)
# don't use this all of the time -> sort of low level
```

Example:

```
addprocs(1)
rc=remotecall(rand,2,100,100)
# then later...
f=fetch(rc)
```

An easier way to do this example is using `@spawn`:

```
r=@spawn rand(100,100)
# it picks a processor and sets this up for us
```

When you are typing you are typing to processor #1.

Number of processors:

```
nprocs()
```

Be able to run something everywhere:

```
@everywhere println(myid())
```

Also try:

```
fetch(@spawn myid())
```

Another useful macro:

```
@parallel
# works with `for` loops
#a=zeros(20) # wrong --> returns all 0's
a = SharedArray(Float64,20)
@parallel for i=1:20
    a[i]=i;
end
```

### 1.6.6 JuliaBox and Jupyter Notebooks

A way to run julia online without installing anything is to use JuliaBox with Jupyter notebooks. The examples will be demonstrated using this tool.

**Jupyter Notebooks**

If you have IJulia installed you can run the examples using the following commands in julia:
using IJulia

# a few examples of changing the path
notebook(dir = Pkg.dir("VehicleModels")*"/examples")
notebook(dir="~/home/febbo/Documents/workspace/OCP/examples")

More information can be found `here <https://github.com/JuliaLang/IJulia.jl>`_.

## 1.6.7 Julia notes

### Using kwargs...

```julia
function
    kw = Dict(kwargs)
    # if there was nothing passed -> set default
    if !haskey(kw,:mode); kw = Dict(:mode => :default) end
    mode = get(kw, :mode, 0);
    if mode == :default
        B = 10
    elseif mode == :LGRM
        B = A
    else
        print("pick a mode\n")
    end
    return B
end
```

B=test(2)
B=test(2;(:mode=>:LGRM))

### Making Variables Programatically


code:

```julia
ex=Array(Expr,4,1);
for i in 1:4
    ex[i]=Expr(:(=),Symbol("x",i),i)
    eval(ex[i])
end
```

vector case:

```julia
A=[1 2 3 4]
ex=Array(Expr,4,1);
for i in 1:4
    ex[i]=Expr(:(=),Symbol("x",i),A)
    eval(ex[i])
```

(continues on next page)
Using JuMP to make a @NLexpression

making nlexps:

```julia
function test(n,exp,x)
    @eval begin
        x1=$x[1,1]
        x1=eval(x1);
        eq=@NLexpression($n,$exp)
    end
    return eq
end
to test:
using JuMP, Ipopt
n = Model(solver=IpoptSolver(print_level=0))
@variable
(n,x[1:4,1:4])
exp=test(n,:sin(x1)),x)
@NLconstraint
(n,exp==0)
@NLobjective
(n,Min,x[1,2])
solve(n)
```

making nlexps withough @eval to work with exp_arrs:

```julia
function test(n,exp_arr,x)
    code=quote
        x1=$x[1,1:4]
        x1=eval(x1);
        @NLexpression($n,$exp_arr[1])
    end
    return eval(code)
end
to test:
using JuMP, Ipopt
n = Model(solver=IpoptSolver(print_level=0))
@variable
(n,x[1,1])
exp_arr=[:(sin(x1)),:(sin(x1))]
exp=test(n,exp_arr,x)
@NLconstraint
(n,exp==0)
@NLobjective
(n,Min,x[1,2])
solve(n)
```

to test:
using JuMP, Ipopt
n = Model(solver=IpoptSolver(print_level=0))
@variable
(n,x[1,1])
exp_arr=[:(sin(x1)),:(sin(x1))]
exp=test(n,exp_arr,x)
@NLconstraint
(n,exp==0)
@NLobjective
(n,Min,x[1,2])
solve(n)
```

example in REPL

```julia
function EX(A)
    code=quote
        temp=$A[1]
        solution=temp
    end
    EX(A)
end
```
return eval(code)
end

A=[:(2+4);:(9-8)]
EX(A)

1.6.8 Problems and Solutions

python ENV with PyCall

Fix:

julia> ENV["PYTHON"]="/usr/bin/python"
"/usr/bin/python"

julia> Pkg.build("PyCall")
INFO: Building Conda
INFO: Building PyCall
INFO: PyCall is using /usr/bin/python (Python 2.7.12) at /usr/bin/python, libpython = libpython2.7
INFO: /home/febbo/.julia/v0.6/PyCall/deps/deps.jl has been updated
INFO: /home/febbo/.julia/v0.6/PyCall/deps/PYTHON has been updated

julia> using PyCall
INFO: Recompiling stale cache file /home/febbo/.julia/lib/v0.6/PyCall.ji for module

julia> @PyCall.pyimport yaml

1.7 git

1.7.1 Website links

- https://github.com/
- The Pro Git book is available here!

1.7.2 Useful Commands

add everything to the commit (including new file and files that were deleted):
git add -A

commit all of the changes:

```bash
git commit -m "some message about what you did"
```

push to remote account:

```bash
git push origin master
```

view current tags:

```bash
git tag
```

making a new tag:

```bash
git tag -a V0.0.1 -m " new version 0.0.1"
```

committing a tag:

```bash
git push origin master --tags
```

checkout a tag:

```bash
git checkout -b [branchname] [tagname]
```

see which branch you are on:

```bash
git branch
```

to make a new branch:

```bash
git checkout -b [name_of_your_new_branch]
```

to change the working branch:

```bash
git checkout [name_of_your_new_branch]
```

to push the branch to github:

```bash
git push origin [name_of_your_new_branch]
```

delete a local branch:

```bash
git branch -d the_local_branch
```

delete a remote branch:

```bash
git push origin --delete the_remote_branch
```

remove a large file from a commit that has not been pushed to master yet:

```bash
git filter-branch --index-filter 'git rm -r --cached --ignore-unmatch papers/MO3/results/laptop/demoZ/D/videos/Final.mp4' HEAD
```

or remove a large folder from a commit that has not been pushed to master yet:
git filter-branch --index-filter 'git rm -r --cached --ignore-unmatch papers/MO3/results/laptop/demoZ/D/videos/*' -f HEAD

creating to github:

```bash
git remote add origin git@github.com:username/new_repo
```

- making a branch look here

Then make a new repository using the interweb

- Check out this link for more info.

Caching your github password:

```bash
git config --global credential.helper 'cache --timeout=3600'
# Set the cache to timeout after 1 hour (setting is in seconds)
```

## Working with Remote Repositories

To clone a repo:

```bash
git clone https://github.com/CPFL/Autoware
```

To view the remote:

```bash
febbo@febbo-HP-ZBook-17-G2:~/Documents/workspace/Autoware$ git remote -v
origin       https://github.com/huckl3b3rry87/Autoware (fetch)
origin       https://github.com/huckl3b3rry87/Autoware (push)
upstream     https://github.com/CPFL/Autoware.git (fetch)
upstream     https://github.com/CPFL/Autoware.git (push)
```

Removing a remote origin:

```bash
git remote rm origin
```

Setting an origin:

```bash
git remote set-url origin "https://..."
```

- Source is this

Revert to an old commit:

```bash
git push -f origin $old_commit_id:master
```

Make sure that you commit changes before moving from one branch to another the changes that you make do not belong to any particular branch!

remove files that where previously cached that are now in .gitignore:

```bash
git rm -r --cached .
git add .
git commit -am "Removed ignored files"
```

Update your fork from the from the upstream remo:
More info here

To automatically fetch and merge from a remote:

```bash
git pull upstream master
```

Remove submodule:

```bash
git rm --cached the_submodule_path
```

<<<<<<< HEAD Merging development branch with master ————————————————————————————————————————————
Merge master into development to see if there are any conflicts, so master remains clean:

```bash
(on branch development)$ git merge master
(resolve any merge conflicts if there are any)
git checkout master
git merge development (there won't be any conflicts now)
```

<https://stackoverflow.com/questions/27828404/why-does-git-status-show-branch-is-up-to-date-when-changes-exist-upstream>'_

Resolving mere conflicts ———————————————————————————————————————-

====== Basics of Collaborating on git ============================================================= This is a
beginners guide to collaborating on git. Several examples will be provided assuming TulgaErsal is collaborating
with huckl3b3rry87 on the PhD repo.

Getting started

This tutorial assumes that you are using a command line interface to git, for Windows consider using cmd
and make sure that you download the full version that has git for Windows. Additionally, Atom is a useful tool
for resolving merge issues visually.

Fork a Repository

1) Go to github.com and login. If it is a private repo that you will be collaborating on, then accept any invitations
to collaborate.

2) Navigate to the repo that you will be colaborating on i.e.:

```bash
https://github.com/huckl3b3rry87/PhD
```

3) In the top right conner of the page click Fork

4) open terminal and navigate to a folder where you will be working i.e.

```bash
cd Documents\workspace\`
```

5) in the terminal, clone the forked repo that you will be collaborating on:

```bash
git clone https://github.com/TulgaErsal/PhD
```

6) To see the current remote repo, type:

```
1.7. git
```
7) To add the original repo as the upstream type:

```
git remote add upstream https://github.com/huckl3b3rry87/PhD.git
```

8) make sure that the upstream was added:

```
git remote -v
```

Which should say:

```
origin https://github.com/TulgaErsal/PhD.git (fetch)
origin https://github.com/TulgaErsal/PhD.git (push)
upstream https://github.com/huckl3b3rry87/PhD.git (fetch)
upstream https://github.com/huckl3b3rry87/PhD.git (push)
```

**Example 1**

To make sure that you are using the most recent version of the upstream (or original repo) you need to get the latest code and merge it into your repo. Use the terminal to navigate to the git folder with the repo that you are working on. Then type:

```
git merge upstream/master
```

**Note:** The above command attempts to automatically merge, and if there are merge issues they can easily be resolved using the Atom text editor.

**Note:** If you run this example just after setting everything up there should be no differences in the upstream repo.

**Example 2**

Each day that you make changes you can push them to your local repository.

**Option 1 (using Atom)**

If you are using the Atom text editor, this is very easy to do.

1) open the Packages tab and scroll down to Github and click Toggle Git Tab.
2) Click Stage All to stage the changes
3) Write a commit message and click Commit
4) Under the Commit button push the up arrow then click Push
5) Put in your git user info
Option 2 (using terminal)

1) add changes:

```
git add -A
```

2) commit changes

```
git commit -m "updated docs"
```

3) push changes

```
git push origin master
```

Example 3

This example is for when you are ready to commit to the upstream repo, this example shows you how to make a pull request.

Assuming that, your local changes have all been committed to the local repo you can easily make a pull request at::

```
https://github.com/TulgaErsal/PhD.git
```

Just click the New Pull Request button.

This will then alert the original repo owner and they can then merge your changes.

```
>>>>>> 583ccf8843b87e3b5ba41467af531075812f8d41
```

1.7.3 Create a disconnected git branch

1) start with a fresh copy of the repo

2) Create a new disconnected branch:

```
git checkout --orphan gh-pages
```

3) hop onto that branch:

```
git checkout -b gh-pages
```

4) At this point there are no commits but lots of files from whatever branch you were on. Have git remove those files:

```
git rm -rf .
```

then follow the rest here:

```
https://coderwall.com/p/0n3soa/create-a-disconnected-git-branch
```

```
 julia> Pkg.clone("https://github.com/JuliaMPC/MPCDocs.jl")
 INFO: Cloning MPCDocs from https://github.com/JuliaMPC/MPCDocs.jl
 INFO: Computing changes...
 INFO: No packages to install, update or remove

 julia> febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git checkout --orphan gh-
 ...pagesSwitched to a new branch 'gh-pages'
```

(continues on next page)
The program 'branch' is currently not installed. You can install it by typing:

```
sudo apt install rheolef
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git branch
  master
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git checkout gh-pages
error: pathspec 'gh-pages' did not match any file(s) known to git.
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git branch
  master
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git checkout -b gh-pages
```

```
Switched to a new branch 'gh-pages'
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git rm -rf .
```

```
fatal: pathspec '.' did not match any files
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ ls
```

```
MPCDocs  MPCDocs.jl
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ cd MPCDocs
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ ls
```

```
appveyor.yml  LICENSE.md  README.md  REQUIRE  src  test
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git branch
  * gh-pages
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git clean -fdx
```

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/MPCDocs$ git branch
  * gh-pages
```

```
(continues on next page)
```

(continues on next page)
1.7.4 Forking a Repository

Follow what this page talks about
also if you are doing this in julia see Another way to connect to github it using ssh
do a:

```
git branch
```

Initially the error was:

```
Permission denied (publickey).
fatal: Could not read from remote repository.
```

- This was obtained when initially setting up the git repositories in julia after cloning a package and trying to push modifications back up to the remote repository.
- Information on this can be founds at, or by following the two steps a fix may be obtained:

FOLLOW:

https://help.github.com/articles/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent/

NOTE: just hit enter, don’t change the default location!!! THEN:

https://help.github.com/articles/adding-a-new-ssh-key-to-your-github-account/

1. Make an ssh key and add it to github, following.
2. Check out this, or use the following commands:
   - A program to hold private keys for public authentication.
     type:

     ```
     ssh-agent
     ```

   - Initially the agent does not hold any private keys.
     So run:

     ```
     ssh-add
     ```
1.7.5 Mistakes I Made

- Make sure that you are working on the master branch!
  - Do not check out a tag and start making changes only to realize that you are not on the master branch!
- Trying to connect to github using ssh
  1) Create a github repository, with the name (for example: huckl3b3rry87/LiDAR.jl)
  2) Then
     Type this in the terminal:
     ```
     febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/LiDAR$ git remote add origin git@github.com:huckl3b3rry87/LiDAR.jl
     ```
  3) Then
     Try this:
     ```
     febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/LiDAR$ git pull master
     ```
  4) Next
     Get this:
     ```
     fatal: 'master' does not appear to be a git repository
     fatal: Could not read from remote repository.
     Please make sure you have the correct access rights and the repository exists.
     ```

Next we are going to test the ssh connection

5) Attempt to ssh to GitHub By typing:

```
febbo@febbo-HP-ZBook-17-G2:~/.julia/v0.5/LiDAR$ ssh -T git@github.com
Hi huckl3b3rry87! You've successfully authenticated, but GitHub does not provide shell access.
```  
6) realize that you messed up by typing:

```
git pull master
```
and not:

```
git pull origin master
```  

1.8 Documentation

1.8.1 Creating Documentation

**Sphinx**

**Installation**

In the terminal run:
pip install sphinx


**Useful Resources**

- an awesome video introduction
- toctree
- basic commands and syntax
- useful resource
- making sections

**Common Commands**

To start documenting:

```bash
sphinx-quickstart
```

To manually build documentation:

```bash
make html
```

To clean out an old build folder when things have changed significantly:

```bash
make clean
```

**Images:**

```bash
.. image:: gnu.png
```

To make a comment:

```bash
.. Comment here
```

Something

**To make a warning**

Type:

```bash
.. warning:: this is a warning
```

Which looks like this:

**Warning**: this is a warning
To make a note

Type:

```bash
.. note:: please be careful with spacing!
```

Which looks like this:

Note: please be careful with spacing!

To make a todo box

Type:

```bash
.. todo:: testing
```

Which looks like this:

Todo: testing

changing the themes

- In `conf.py` change:

```bash
html_theme = 'haiku'
```

- Or a main one:

```python
import sphinx_rtd_theme
html_theme = "sphinx_rtd_theme"
html_theme_path = [sphinx_rtd_theme.get_html_theme_path()]
```

- An awesome theme

Manging References

Properly managing references is a critical habit and these are some of the useful software tool that I use to manage my references.

Using Sphinx BibTex extension

Sphinx BibTex extension

- To install

  Do this:

  ```bash
  pip install sphinxcontrib-bibtex
  ```
In `conf.py` add:

```python
extensions = ['sphinxcontrib.bibtex']
```

**Warning:** that it is *added* to the rest of extensions!! Not just at the top, or it will be removed!!

Like this:

```python
extensions = [  
    'sphinx.ext.autodoc',  
    'sphinx.ext.doctest',  
    'sphinx.ext.intersphinx',  
    'sphinx.ext.todo',  
    'sphinx.ext.coverage',  
    'sphinx.ext.mathjax',  
    'sphinx.ext.ifconfig',  
    'sphinxcontrib.bibtex',  
]
```

Also, you can avoid these errors on readthedocs.com:

```bash
python /home/docs/checkouts/readthedocs.org/user_builds/nloptcontroljl/envs/latest/  
˓→bin/pip install --exists-action=w --cache-dir /home/docs/checkouts/readthedocs.org/  
˓→user_builds/nloptcontroljl/.cache/pip -r pip install sphinxcontrib-bibtex  
Could not open requirements file: [Errno 2] No such file or directory: 'pip install  
˓→sphinxcontrib-bibtex'  
You are using pip version 8.1.2, however version 9.0.1 is available.  
You should consider upgrading via the 'pip install --upgrade pip' command.  
Command time: 0s Return: 1
```

By typing:

```bash
requirements.txt
```

Into the Advanced Settings Page and making a `requirements.txt` file with:

```bash
pip install --upgrade pip  
pip install sphinxcontrib-bibtex
```

More on this is [here](#)

Example: To cite:

```plaintext
according to :cite:`someone` yada yada..
```

Then, at the end of the document include:

```plaintext
.. bibliography:: references.bib
```

**Useful Work-flow Technique**

Setting up a server to build the documentation when a change is detected.

Installation:
pip install sphinx-autobuild

Then go into your main directory and type:

sphinx-autobuild docs docs/_build/html

Or the directory that contains conf.py and type:

sphinx-autobuild . _build_html

Then visit the website: http://127.0.0.1:8000/
  • This will show you the live changes (after each save!!)

More information can be found using this resource.

Useful Software

Read The Docs

To get it:

pip install sphinx_rtd_theme

Read the Docs is useful resource to host the webpage.

Some Issues:

  • The git repository for OCP is private and this website only hosts public repositories.
  • Had to remove the last line here in conf.py for the code to work with ReadtheDocs.

like this:

```python
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.doctest',
    'sphinx.ext.intersphinx',
    'sphinx.ext.todo',
    'sphinx.ext.coverage',
    'sphinx.ext.mathjax',
    'sphinx.ext.ifconfig',
    'sphinx.ext.viewcode',
    'sphinxcontrib.bibtex',
]
#    'sphinx.ext.githubpages',
```

  • Had to make sure that the name of the project was correct
  • Had to make sure that the webhook was activated on github
Potential Issues

1. Make sure that the “toctree” is indented by 3 space characters. I listed the .rst files by 4 space characters and this created an issue. To resolve the issue, you need to have the same indentation level.

2. When cross referencing things in the document make sure you skip a space when you define the ref.
   - like this
     works:
     ```
     .. _ploy_div:
     Polynomial Division
     ---------------
     ```
   - not like this
     fails:
     ```
     .. _ploy_div:
     Polynomial Division
     ---------------
     ```

3. Trying to uninstall sphinx so that I could get a newer version without bugs:

   ```
   febbo@febbo-HP-ZBook-17-G2:~/Desktop/useful downloads$ pip uninstall sphinx
   Cannot remove entries from nonexistent file /home/febbo/anaconda3/lib/python3.5/site-packages/easy-install.pth
   ```

   The problem: This probably won’t work:

   ```
   conda update setuptools
   ```

   The fix, was to download ez_setup.py from this link and run it in the terminal:

   ```
   febbo@febbo-HP-ZBook-17-G2:~$ python ez_setup.py
   ```

   Then I could uninstall Sphinx-1.4.1:

   ```
   febbo@febbo-HP-ZBook-17-G2:~$ pip uninstall sphinx
   Uninstalling Sphinx-1.4.1:
   /home/febbo/anaconda3/bin/sphinx-apidoc
   /home/febbo/anaconda3/bin/sphinx-autogen
   /home/febbo/anaconda3/bin/sphinx-build
   /home/febbo/anaconda3/bin/sphinx-quickstart
   /home/febbo/anaconda3/lib/python3.5/site-packages/Sphinx-1.4.1-py3.5.egg
   Proceed (y/n)? y
   Successfully uninstalled Sphinx-1.4.1
   ```

   There was reason I needed to uninstall this version of Sphinx was because the citations where not working.

   Finally I installed Sphinx-1.5.2:

   ```
   #Install from newest dev version in stable branch: :
   pip install git+https://github.com/sphinx-doc/sphinx@stable
   ```
1.8.2 Miscellaneous

Pandoc

Software used to convert documentation from one format to another.

1.9 LaTeX

1.9.1 Install

Use Texmaker and when you change the build directory, make sure that you add build/ before the % to all the commands in configure.

Install on Ubuntu:

```
sudo apt-get install texlive-full
```

Other packages:

```
sudo apt-get install texlive-latex-extra
dsour apt-get install texlive-bibtex-extra biber
dsour apt-get install texlive-fonts-recommended
dsour apt-get install texlive-publishers
```

misc:

```
sudo apt-get install pdflatex
```

1.9.2 Beemer

Adobe Reader


Convert to Individual Files

https://tex.stackexchange.com/questions/240243/getting-gif-and-or-moving-images-into-a-latex-presentation

type:
Autonomous Vehicle Control Documentation, Release 0.0.1-rc1

```
convert -coalesce something.gif something.png
```

**Syntax Highlighting: listings**


**Syntax Highlighting: minted**

```
usepackage{minted}

which needs:
```
```
sudo apt-get install python-pygments
```

**OLD**


**FFmpeg**


Change quality:

```
ffmpeg -i input.mp4 -qscale:v 2 output_%03d.jpg
```

**Embedding movies1:**


code:

```
\usepackage[loop,controls,buttonsize=0.24cm,buttonbg=0.8,autoplay]{animate}
\%
\begin{frame}{Why julia?}
\begin{figure}
\centering
\animategraphics[loop,width=\textwidth,every=1\{100\}{videos/v1/out_}\{1\}
\rightarrow{1199}
\caption{Our Algorithms where not Solving in Real-Time!}
\end{figure}
\end{frame}
```

**Embedding movies2:**

https://tex.stackexchange.com/questions/240243/getting-gif-and-or-moving-images-into-a-latex-presentation

1.9. LaTeX
Embedding movies3:


animate package

http://mirror.hmc.edu/ctan/macros/latex/contrib/animate/animate.pdf

Viewing slides

Okular:

```
sudo apt-get install okular
```

xPDF

install:

```
sudo apt-get update
sudo apt-get install xpdf
```

1.9.3 Software Tools

Texmaker

A useful program to use LaTeX on Ubuntu.

**Warning:** make sure that you play with the build option if it is saying that the references were not found.

For instance, after a while this worked:

```
LuaLaTeX View PDF
```

Then, the above option did not work and this worked:

```
LaTeXMk View PDF
```

To insert .svg files automatically

Useful links:

- svg package
• shell-escape command

Convert SVG to PDF

```
sudo apt-get install inkscape
```

https://tex.stackexchange.com/questions/194148/how-to-make-latex-automatically-call-inkscape-to-export-figure-to-pdf-tex


Making Figures With LaTeX

http://www.texample.net/tikz/examples/tag/diagrams/

1.9.4 Plotting

PGFPlots

Documented here

TikZ and PGF

Documentation is here

With julia

• a recent package is PGFPlotsX
• the original package is PGFPlots

Miscellaneous

How to include a `.svg` into LaTeX

```latex
\documentclass[]{memoir}
\usepackage{svg}
\setsvg{inkscape=inkscape -z -D,svgpath=figs/}
\begin{document}
\begin{figure}[!ht]
\centering
\includesvg[width=0.5\columnwidth, svgpath = ./relativePATHtoSVG/]{filename}
\end{figure}
\end{document}
```

1.9. LaTeX
1.10 C

1.10.1 Including files

There are 2 kinds of include directives:

- double quoted ones (#include “xyz.h”)
  - the working directory
- angle bracket ones (#include <xyz.h>)
  - usually /usr/include/

Other Notes

- Place the header file in the same directory as your .c file and use -I when compiling
- You shouldn't place your header files in /usr/include that is meant for the system headers.

Useful resources

- https://stackoverflow.com/questions/27660713/including-header-file-from-static-library

1.11 Operating Systems

1.11.1 Linux

Useful Programs

In the terminal:

```bash
sudo dpkg --install atom-amd64.deb
sudo apt install ffmpeg
sudo apt-get install cmake bar time binutils make libssl-dev gfortran libunwind8-dev
    --gcc g++ #clang curl perl wget m4 patch pkg-config
sudo apt-get install gsfonts-x11
sudo apt-get install notepadqq
sudo apt-get update
sudo apt-get install gfortran
sudo apt-get install libnlopt0
sudo apt-get install openssh-server
sudo apt install ipmiutil
sudo apt-get install hdf5-tools
apm install latex # Atom package manager for latex!
sudo apt-get install pdf2svg # for PGFPlots
pip install sphinxcontrib-bibtex
sudo apt-get install texlive-luatex
sudo apt install aptitude
sudo apt-get install freecad
sudo apt install blender
sudo apt install rpm
apm install latex
sudo apt autoremove
```
Basic Linux Commands

Never run an executable in root!
Some more info can be found on 'this page '<https://www.cyberciti.biz/faq/tar-extract-linux/>'.
To move a folder an all of its contents:

```
mv /path/sourcefolder/* /path/destinationfolder/
```

To unpack a tar file:

```
tar -xvf file.tar
```

To extract a .tar.gz:

```
tar -xzvf file.tar.gz
```

Note: Modern tar can recognize the format automatically:

```
sudo tar xf armadillo-8.300.3.tar.xz -C /opt
```

Unzip:

```
unzip stuff.zip -d /destination/folder
```

directory comment:

```
$HOME = /home/febbo
```

Run a jar file: java -jar lad

Useful Tools

- Synaptic Package Manager

/etc/bash.bashrc vs. $HOME/.bashrc

Be careful there are two files that configure the terminal!
http://unix.stackexchange.com/questions/140207/messed-up-bash-bashrc-file-commands-not-working-anymore

in my $HOME/.bashrc:

```
alias blender='/opt/blender/blender'
alias julia='/opt/julia-6445c82d00/bin/julia'
alias docs='sphinx-autobuild . _build_html'
export DYLD_LIBRARY_PATH="$HOME/knitro/lib:$DYLD_LIBRARY_PATH"
export LD_LIBRARY_PATH="$HOME/knitro/lib:$LD_LIBRARY_PATH"
export PYTHONPATH=$PYTHONPATH:/usr/lib/python2.7/dist-packages
export GAZEBO_PLUGIN_PATH=/home/febbo/.julia/v0.5/OCP/MAVs/workspace/src/velodyne_plugin/build/:${GAZEBO_PLUGIN_PATH}
source /opt/ros/kinetic/setup.bash
export ROS_PACKAGE_PATH=/home/febbo/.julia/v0.5/OCP/MAVs/workspace:$ROS_PACKAGE_PATH
```
Ubuntu package manager

https://launchpad.net/ubuntu/+source/matplotlib

Video Stuff

to convert to MP4

Type:

```bash
ffmpeg -f gif -i mainSimPath.gif RESULT.mp4
```

Quickly convert to .gif files:

Following this link

Add this to your ~/.bash_profile file:

```bash
# Convert video to gif file.
# Usage: video2gif video_file (scale) (fps)
video2gif() {
    ffmpeg -y -i "${1}" -vf fps=${3:-30},scale=${2:-320}:-1:flags=lanczos,palettegen "${1}.png"
    ffmpeg -i "${1}" -i "${1}.png" -filter_complex "fps=${3:-10},scale=${2:-320}:-1:flags=lanczos[x];[x][1:v]paletteuse" "$1.gif"
    rm "${1}.png"
}
```

Then load it in the .bashrc file:

```bash
. ~/.bash_profile
```

### 1.11.2 Setting up a machine for Dual Boot

Todo: transcribe notes from black book on this

### 1.12 misc

#### 1.12.1 MPT3

- MPT3 is a MATLAB based optimization tool that comes out of The Automatic Controls Laboratory at ETH.
- Lots of other useful software that come out out ETH in Zurich, Switzerland and can be found [aqui](#).
- Additionally, I worked with Michal Kvasnica (a former membeber of The Automatic Controls Laboratory at ETH) developing Moving Obstacle avoidance code using MPT3.
  - The result of this work is a tool named OptiPlan and the code can be found [here](#).
1.12.2 BARC

http://www.barc-project.com/

1.12.3 AUTORALLY

http://autorally.github.io/
Georgia tech
https://collab.cc.gatech.edu/borg/home?destination=home

GTSAM

Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions
https://www.youtube.com/watch?v=CsjKci5Ifco&feature=youtu.be

1.12.4 Different Text Editors

I tried a few different text editors, but this one I really like and it is very similar to coding in MATLAB. It is a tool called JUNO that basically links a nice text editor called Atom to julia, so that you can run julia interactively.

Atom

Atom Flight Manual
Useful things:
spell check <https://github.com/atom/spell-check>:

ctrl+shift+;

Markdown Preview Package:

ctrl+shift+m

https://github.com/atom/markdown-preview

Subscripts:

variable\_1+tab

Using /LaTeX with Atom. It was suggested to use Tex Live <https://www.tug.org/texlive/acquire-netinstall.html>
Where the quick install instructions are here<https://www.tug.org/texlive/quickinstall.html> https://atom.io/packages/latex

1.12. misc
**TeX Live**

I use this with Atom.


To do this change the bashrc file:

```
gksu gedit /etc/bash.bashrc
```

Paste this at the end:

```
PATH=/usr/local/texlive/2016/bin/x86_64-linux:$PATH; export PATH # make sure there are NO spaces
MANPATH=/usr/local/texlive/2016/texmf-dist/doc/man:$MANPATH; export MANPATH
INFOPATH=/usr/local/texlive/2016/texmf-dist/doc/info:$INFOPATH; export INFOPATH
```

Then in Atom, go to settings and add the path to the texlive:

```
/usr/local/texlive/2016/bin/x86_64-linux
```

**Potential Issues**

- If the julia binary is not where it belongs (for instance I put it in my /opt/... folder)
  - Change the julia .config file (in Atom) to the proper folder (where julia binaries are)

- Otherwise
  - leave it as: juliaPath: “julia”

I had to:

```
Pkg.clone("https://github.com/JunoLab/Juno.jl")
Pkg.update()
```

Make sure you close both atom and julia after installation. Now you should be able to run julia from Atom!

**1.12.5 Transferring Files To A Website**

http://www.umich.edu/~umweb/how-to/homepage.html


**type:**

```
sftp sftp.itd.umich.edu
```

Or just do:

```
scp -r /home/febbo/Documents/workspace/PhD/docs/_build/html sftp.itd.umich.edu:/afs/umich.edu/user/f/e/febbo/Private/html
```
Setting Permissions:
https://mfile.umich.edu/?path=/afs/umich.edu/user/f/e/febbo/Private/html

1.12.6 Community Help

There is a tremendous community of people online that will help you with your projects. I have helped when I can, but largely other more experienced people usually help me. Below I have listed a few of the communities that I have been involved with.

Stackoverflow

A more general venue for asking questions to the online community of experts in various fields.

GitHub

If you have issues with people's software, you can submit an issue.

Gitter

Basicaly an instant message service for developers. I have used it for julia, plots.jl, and juno.jl.

1.12.7 Making A Thesis

https://github.com/jterrace/sphinxtr

1.12.8 Miscellaneous

Making the Screen Brighter with an Apple Monitor on Ubuntu

Ubuntu Freezing and not Booting with Monitors Plugged in

Blender

3D modeling tool

chmod

changes permissions. For instance:

```bash
chmod +x filename
```

Allows the file to be executed.

1.12.9 MATLAB

I used MATLAB to develop the work in my 2016 ASME paper. Great software, but julia is faster, free and just plain better:)

1.12. misc
1.12.10 Build Tools

**catkin**

This is the official build system of ROS, it uses the CMake macros and Python to add functionality to Cmake.

**Installation Instructions**

**catkin-tools**

These tools are used to facilitate a merged build process. That is several inter-dependent but separately developed CMake projects

**Installation:**

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu `lsb_release -sc` main" > /etc/apt/sources.list.d/ros-latest.list'
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
sudo apt-get update
sudo apt-get install python-catkin-tools
```

**Warning:** If you this was built on an old machine or paths got updated (e.g. a new version of julia) then it may be necessary to clean out the build:

```
rm -rf build
```

If you move a package, delete the build and the devel folders then do a catkin_make

**CMake**

**Ninja**

a small build system:

```
sudo apt-get install ninja-build
```

**Linuxbrew**

For working with Linux and Mac install:

```
sudo apt install linuxbrew-wrapper
```

then:

```
brew install libyaml
```
1.12.11 Reference Management

**JabRef**

BibTex files can be managed with a free software called JabRef.

First I tried a few things including:

```
sudo apt-get install default-jre
sudo apt-get update
```

And:

```
sudo add-apt-repository ppa:openjdk-r/ppa
sudo apt-get update
sudo apt-get install openjdk-7-jre
```

This messed stuff up and then I deleted all of my old java versions using:

```
sudo natilus
```

Ended up needing to follow this advice <http://askubuntu.com/questions/251213/unable-to-install-default-jdk>:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java8-installer
java -version
```

This finally worked and I was able to run JabRef with no problem using:

```
java -jar JabRef-3.8.jar
```

If something is broken:

```
sudo apt-get -f install
```

### Using JabRef with LibreOffice

http://help.jabref.org/en/OpenOfficeIntegration  
http://homepage.usys.ethz.ch/eugsterw/knowhow/jabref-libreoffice/
ls $(which soffice) ls $(which soffice) -l locate libreoffice https://onetransistor.blogspot.com/2015/04/libreoffice-bibliography-jabref.html
http://homepage.usys.ethz.ch/eugsterw/knowhow/jabref-libreoffice/

```
cd /usr/lib/openoffice/program/soffice
```

```
febbo@febbo-HP-ZBook-17-G2:/usr/lib/libreoffice/program$ locate unoil.jar /usr/lib/libreoffice/program/classes/unoil.jar
```

```
febbo@febbo-HP-ZBook-17-G2:/usr/lib/libreoffice/program$ locate jurt.jar /usr/lib/libreoffice/program/classes/jurt.jar
```

Need to make symbolic links to the following java files:

```
febbo@febbo-HP-ZBook-17-G2:/usr/lib/libreoffice/ure-link/share/java$ sudo ln -s /usr/lib/libreoffice/program/classes/jurt.jar febbo@febbo-HP-ZBook-17-G2:/usr/lib/libreoffice/ure-link/share/java$
```
sudo ln -s /usr/lib/libreoffice/program/classes/juh.jar febbo@febbo-HP-ZBook-17-G2:/usr/share/libreoffice/ure-link/share/java
sudo ln -s /usr/lib/libreoffice/program/classes/ridl.jar

EndNote

I still use this tool, but it costs money and I do not have it on Ubuntu.
2.1 LiDAR Devices

2.1.1 Velodyne Puck 16

VPL-16 Setup (for Ubuntu 16.04)

Getting on the LiDAR on the Network

The basic tutorial is here, but you may run into issues like I did. In which case, the following notes may be useful. First figure out what the devise name is. To show all devices found:

```
ifconfig -a
```

Which gives something like (desktop):

```
febbo@febbo-HP-Z220-SFF-Workstation:/Documents/workspace/Docs$ ifconfig -a
eno1 Link encap:Ethernet HWaddr b4:b5:2f:a9:52:61
    inet addr:192.168.1.200 Bcast:0.0.0.0 Mask:255.255.255.255
    inet6 addr: fe80::b6b5:2fff:fea9:5261/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
    RX packets:1604067 errors:0 dropped:0 overruns:0 frame:0
    TX packets:22232 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1722487876 (1.7 GB) TX bytes:2422647 (2.4 MB)
    Interrupt:20 Memory:f7100000-f7120000
lo    Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:65536 Metric:1
    RX packets:89494 errors:0 dropped:0 overruns:0 frame:0
```

(continues on next page)
Warning: Note that the device is en01, not eth0. Also, on my laptop the device name is enp0s25, so it is different depending on the machine.

Then do (desktop):

```bash
sudo ifconfig en01 up
sudo ip addr add 192.168.1.200 dev en01
sudo route add -net 192.168.1.0 netmask 255.255.255.0 dev en01
```

Then do (laptop):

```bash
sudo ifconfig enp0s25 up
sudo ip addr add 192.168.1.200 dev enp0s25
sudo route add -net 192.168.1.0 netmask 255.255.255.0 dev enp0s25
```

Next go to: Network -> Options... and click Ethernet Settings

**Ethernet Settings**

Type the LiDAR’s cloned MAC address in the box.

**Note:** To find the cloned MAC address for the LiDAR; look at the bottom of the device. For me it is:

60:76:88:10:47:F9

**Note:**

- Now you can change the name of the network to something, perhaps LiDAR in Edit Connections...
- To see information about the connection check out Connection Information.

**Testing the LiDAR**

First clone the Velodyne LiDAR ros package into a catkin_ws:

```bash
git clone http://wiki.ros.org/velodyne
```

To run it:

```bash
roslaunch velodyne_pointcloud VLP16_points.launch
```

To view data run:
rosrun rviz rviz -f velodyne

Then add try adding LaserScan and PointCloud2 and their topics /velodyne_points and /scan, respectively.

If you have trouble these instructions will help you display the data in rviz.

Getting on the Computer back on the Internet

Note: If you accidentally deleted the Internet connection or messed it up somehow

On the top right of your screen highlight the Network icon and click Edit Connections. Then click Add add back the Internet with the following steps.

**Warning:** Make sure that you name the LiDAR connection and the Internet connection.

First go to: Network -> Options...

**Ethernet Settings**

Note: If you are trying to get the Internet back delete the MAC address in the box.

**IPv4 Settings**

Note: If you are trying to get the Internet back, select the Method to be: Automatic (DHCP)

**Network Commands**

random commands:

`route -n`

### 2.1.2 SLAMTEC

360 degree 2D LiDAR for $500 well documented and comes with plenty of ROS support

---

2.1. LiDAR Devices 69
These are some notes that I took for classes etc.

Contents:

3.1 Navigation and Guidance of Aerospace Vehicles (AERO 584)

Contents:

3.1.1 Cheat Sheet

not well formatted. . . . look at source!

\[
\begin{align*}
(a^2 - b^2) &= (a - b)(a + b) \\
(a^3 - b^3) &= (a - b)(a^2 + ab + b^2) \\
(a^4 - b^4) &= (a - b)(a + b)(a^2 + b^2)
\end{align*}
\]

\[
\begin{align*}
e^a e^b &= e^{a+b} \\
(e^x)^n &= e^{2x}
\end{align*}
\]

\[
(A + B)^T = A^T + B^T
\]

\[
(AB)^T = B^T A^T
\]

\[
AB^T = BA^T
\]

\[
(A^T)^{-1} = (A^{-1})^T
\]

\[
det(A^T) = det(A)
\]
Also, if A is square, then its eigenvalues are equal to the eigenvalues of its transpose. Additionally, if the matrix is also differentiable and nonsingular

\[
\frac{d}{dt}(P^{-1}(t)) = -P^{-1}(t)\dot{P}(t)P^{-1}(t)
\]

which can be found with:

\[
d\frac{d}{dt}(P(t)P^{-1}(t)) = \frac{d}{dt}(I) = 0
\]

\[
A^T = -A
\]

\[
A = \begin{bmatrix}
a_{11} & a_{12} \\
a_{21} & a_{22}
\end{bmatrix}
\]

\[
A^{-1} = \frac{1}{|A|}[\text{tr}(A) - a_{12}a_{22} - a_{11}] - a_{21}a_{11}
\]

\[
\begin{vmatrix}
a & b & c \\
d & e & f \\
g & h & i
\end{vmatrix}
\]

\[
= a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}
\]

\[
sin(2a) = 2\cos(a)\sin(a)
\]

\[
\cos(2a) = 1 - 2\sin^2(a) = 2\cos^2(a) - 1
\]

States that the derivative of the integral is (for constant limits of integration):

\[
\frac{d}{dt}\int_{a}^{b} f(x, t) \, dt = \int_{a}^{b} \frac{\partial}{\partial x} f(x, t) \, dt
\]

States that the derivative of the integral is (for limits of integration that are not constant)

\[
\frac{d}{dt}\int_{f(a)}^{f(b)} f(x, t) \, dt =
\]

\[
= f(x, b(x))\frac{dx}{dx}b(x) - f(x, a(x))\frac{dx}{dx}a(x) + \int_{f(a)}^{f(b)} \frac{\partial}{\partial x} f(x, t) \, dt
\]

Note :: math: ‘t’ is a variable in the integration limit, so the above
formula must be used to derive (t).

\[
\int x^n \, dx = \frac{1}{n+1} x^{n+1}
\]
\[
\int \frac{1}{x} \, dx = \ln |x|
\]
\[
\int u \dot{v} \, dx = uv - \int v \, du
\]
\[
\int e^x \, dx = e^x
\]
\[
\int a^x \, dx = \frac{1}{\ln a} a^x
\]
\[
\int \ln x \, dx = x \ln x - x
\]
\[
\int \sin x \, dx = -\cos x
\]
\[
\int \cos x \, dx = \sin x
\]
\[
\int \tan x \, dx = \ln |\sec x|
\]
\[
\int \sec x \, dx = \ln |\sec x + \tan x|
\]
\[
\int \sec^2 x \, dx = \tan x
\]
\[
\int \sec(x) \tan(x) \, dx = \sec x
\]

\[
dxdy = rdrd\theta
\]

don’t forget to change the limits of integration!

\[
\frac{d}{dt}(\tan^{-1}(x)) = \frac{1}{1 + x^2}
\]
\[ f(t) = \mathcal{L}[f(t)] = F(s) \]
\[ 1 = \frac{1}{s} \]
\[ \delta(t) = 1 \]
\[ \delta(t - t_0) = e^{-st_0} \]
\[ f'(t) = sF(s) - f(0) \]
\[ f^n(t) = s^nF(s) - s^{(n-1)}f(0) - \cdots - f^{(n-1)}(0) \]
\[ t^n(n = 0, 1, 2, \ldots) = \frac{n!}{s^{n+1}} \]
\[ \sin kt = \frac{k}{s^2 + k^2} \]
\[ \cos kt = \frac{s}{s^2 + k^2} \]
\[ e^{at} = \frac{1}{s - a} \]
\[ t^n e^{at} = \frac{n!}{(s - a)^{n+1}} \]
\[ e^{at} \sin kt = \frac{k}{(s - a)^2 + k^2} \]
\[ e^{at} \cos kt = \frac{s - a}{(s - a)^2 + k^2} \]
\[ t \sin kt = \frac{2ks}{(s^2 + k^2)^2} \]
\[ t \cos kt = \frac{s^2 - k^2}{(s^2 + k^2)^2} \]

First translation theorem:
\[
\mathcal{L} \left[ e^{at} f(t) \right] = F(s - a)
\]
\[
\int_{-\infty}^{\infty} f(x) \delta(x - a) \, dx = f(a)
\]
\[
\int_{-\infty+\epsilon}^{\infty+\epsilon} f(x) \delta(x - a) \, dx = f(a), \epsilon > 0
\]
\[ \delta(x - a) = 0 \]

Given some nonlinear system of the form:
\[ \dot{x}(t) = f(x(t), u(t), t) \]
\[ y(t) = g(x(t), t) \]

A linearization can be performed about nominal trajectories
\[ x^0(t), u^0(t), and u^0(t) \text{ (shorthand is 0) by defining the Jacobian of w.r.t. the state evaluated at the nominal trajectories (0) as: trajectories(0) as:} \]

Finally, the output equation may need to be linearized about 0 as well

\[
C(t) = \left. \frac{\partial y}{\partial x} \right|_0 = \begin{bmatrix}
\frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\
\frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial g_n}{\partial x_1} & \frac{\partial g_n}{\partial x_2} & \cdots & \frac{\partial g_n}{\partial x_n}
\end{bmatrix}
\]
Then the perturbation variables are defined as:

\[
\delta x(t) = x(t) - x^0(t) \\
\delta u(t) = u(t) - u^0(t) \\
\delta y(t) = y(t) - y^0(t)
\]

The final linearized system is:

\[
\begin{align*}
\delta \dot{x}(t) &= A(t)\delta x(t) + B(t)\delta u(t) \\
\delta y(t) &= C(t)\delta x(t) \\
\dot{x}(t) &= A(t)x(t) \\
x(t) &= \Phi(t, t_0)x_0 \\
\Phi(t, t_0) &= x(t)x(t_0)^{-1}
\end{align*}
\]

The system is

\[
\begin{align*}
\dot{x}(t) &= A(t)x(t) + B(t)u(t) \\
y(t) &= C(t)x(t) \\
x(t_0) &= x_0
\end{align*}
\]

where the solution is,

\[
\begin{align*}
x(t) &= \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau) \, d\tau \\
y(t) &= C(t)\Phi(t, t_0)x_0 + \int_{t_0}^{t} C(t)\Phi(t, \tau)B(\tau)u(\tau) \, d\tau
\end{align*}
\]

The above two equations are called the variation of constants formulas. They contain two terms, the first term is the free response which is due to \(x_0\) and the second term is the forced response due to the input \(u(t)\). Additionally, the impulse response is defined as

The basic equation is:

\[
\Phi(t, \tau) = e^{A(t-\tau)}
\]

Which can be calculated using:

\[
\mathcal{L}^{-1}[(sI - A)^{-1}] = e^{At}
\]

After this, math: ‘\(\tau\) must be added in, which can be done with by

taking the inverse of \(\Phi(\tau, 0)\) to get \(\Phi(0, \tau)\) and then multiplying by \(\Phi(t, 0)\) as:

The STM is the unique solution to

\[
\frac{\partial}{\partial t}(\Phi(t, t_0)) = A(t)\Phi(t, t_0)
\]

within initial conditions: math: ‘\(\Phi(t_0, t_0) = I\).

To solve:
• multiply the above matrices out
• take the Laplace Transform of each element in the matrix
• solve the algebraic equation for each $\Phi_{i,i}(s)$
• take the inverse Laplace transform to find $\Phi_{i,i}(t)$

$\Phi(t_0, t_0) = I$
$\Phi(t, t_0)^{-1} = \Phi(t_0, t)$
$\Phi(t, t_0) = \Phi(t, t_1)\Phi(t_1, t_0)$

The system Eqn. [eq:sys1] is stable if, given $x(t_0) = x_0, x(t)$ (Eqn. [eq: sol1]) is bounded as $dt \geq t_0$. In this case, $\Phi_{i,i}(t)$ may not go to zero. Stability can also be determined by looking at each (i,j) component of the STM as:

$|\Phi_{ij}(t, t_0)| \leq k < \infty, \forall t_0 \leq t$

The system Eqn. [eq:sys1] is asymptotically stable if, given $x(t_0) = x_0, x(t)$ (Eqn. [eq: sol1]) decays to zero, that is:

The system, Eqn. [eq:sys1], is BIBO stable if when $x_0 = 0$, the forced output response $y$ to every bounded input $u(t)$ is bounded. This requires that $G(t, \tau)$ is absolutely integrable.

Can we estimate a unique $x(t_0) = x_0$, given $u(t)$ and $y(t)$ over the time interval $[t_0, t_1]$? If we have $x_0$, we can solve.

The observability matrix for a LTI system is:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

if the math : 'rank(\mathcal{O}) = n' then the system is observable. The unobservable states are in the null space of the observability matrix, i.e. $Ox_0 = 0$

Observability Gramian

$$M(t_0, t_1) = \int_{t_0}^{t_1} \Phi^T(t, t_0)C^T(t)C(t)\Phi^T(t_0, t)dt$$

a state : math : 'x_0 = x(t_0)' unobservable at time : math : 't_0' if $f$

$$M(t_0, t_1)x_0 = 0, \forall t_1 > t_0$$

so, the unobservable states are in the null space of the observability Gramian. If the only solution that lives in the null-space is the zero vector $x(t_0) = 0$, then the system is completely observable. Also, note that this is not the same as carry out the complete integral to see that the observability Gramian. A system is controllable if we can find a $u(t)$ that drives the state $x(t)$ from $x_0$ in finite time $f$.
The observability matrix for a LTI system is:

\[ C = [B \ BA \ \ldots \ BA^{n-1}] \]

if \ the : \ \text{math} : ' \text{rank}(C) = n' \ \text{then the system is} \ \text{controllable}.

Recall that the rank of a matrix is the number of linearly independent columns.

Controllability Gramian

\[
W(t_0, t_1) = \int_{t_0}^{t_1} \Phi^T(t, t_0)B(t)B^T(t)\Phi^T(t_0, t)dt
\]

this matrix is always symmetric and positive definite. The system is completely controllable if there exists \( t_1 > t_0 \): \( W(t_0, t_1) > 0 \). If the system is controllable at \( t_0 \), then one control that drives the state to the origin.

Given a LTV system as

\[
\dot{x}(t) = A(t)x(t) + B(t)u(t)
\]

\[
y(t) = C(t)x(t)
\]

its dual system is

\[
\dot{x}(t) = -A^T(t)x(t) + C^T(t)u(t)
\]

\[
y(t) = B^T(t)x(t)
\]

The controllable (or uncontrollable) states of one system are the observable (or unobservable) states of the other system.

If \( x \) is a random vector with a PDF \( f(x) \) and \( g: \mathbb{R}^n \rightarrow \mathbb{R}^m \) is a function of \( x \)

\[
\bar{x} = E[x] = \int_{\mathbb{R}^n} x f(x) \, dx \in \mathbb{R}^n
\]

\[
P_{xx} = E[(x - \bar{x})(x - \bar{x})^T]
\]

\[
= \int_{\mathbb{R}^n} (x - \bar{x})(x - \bar{x})^T f(x) \, dx \in \mathbb{R}^{nxn}
\]

Note: the covariance matrix is symmetric as well as positive semidefinite, so

\[
\forall v \in \mathbb{R}_n, v^T P_{xx} v \geq 0
\]

Thus,

\[
v^T P_{xx} v = \int_{\mathbb{R}^n} v^T (x - \bar{x})(x - \bar{x})^T v f(x) \, dx
\]

\[
= \int_{\mathbb{R}^n} (x - \bar{x})^2 f(x) \, dx \geq 0
\]

If \( x_1 \) and \( x_2 \) are subvectors of \( x \)

\[
P_{x_1x_2} = E[(x_1 - \bar{x}_1)(x_2 - \bar{x}_2)^T]
\]

\[
= \int_{\mathbb{R}^n} (x_1 - \bar{x}_1)(x_2 - \bar{x}_2)^T f(x) \, dx \in \mathbb{R}_{n_1x_{n_2}}
\]

3.1. Navigation and Guidance of Aerospace Vehicles (AERO 584)
\[ P^+ = ((P^-)^{-1} + C^T R^{-1} C)^{-1} \]

The relative likelyhood that a random variable \( x \) will take on values on a given interval.

\[ \text{Area} = P(a \leq x \leq b) = \int_a^b f(x)dx \]

If you integrate over the PDF over the entire range then it must equal 1 and the PDF must always be greater than 0.

\[ \int_{\mathbb{R}} f(x)dx = 1 \forall x, f(x) \geq 0 \]

For a PDF uniformly distributed over \([a,b]\), \( f(x) = \text{constant} = c \). \( c \) can then be determined by

\[ \int_a^b c dx = 1 \Rightarrow c = \frac{1}{b-a}. \]

Given the joint PDF \( f_x(x) = f(x_1, x_2) \), the marginal PDF of \( x_1 \) is:

\[ \phi_x(s) \text{is useful to compute the PDF for x and the Gaussian distribution. The expected value can be used to calculate the characteristic function.} \]

The statistical properties of \( x \) are equivalently specified by PDF \( f(x) \) or by the characteristic function \( \phi_x(s) \).

Similar to a Fourier Transform \( \phi_x(s) \) can be put back into the time domain with:

If the following equations are true, then the PDF’s are independent.

\[ f(x, y) = f_x(x)f_y(y) \]

\[ \phi_{xy}(s, r) = \phi_x(s)\phi_y(r) \]

If \( x \) and \( y \) are independent, the conditional density function and conditional mean satisfy:

\[ f(x|y) = \frac{f(x, y)}{f_y(y)} = \frac{f_x(x)f_y(y)}{f_y(y)} = f_x(x) \]

\[ E[x|y] = \int_{\mathbb{R}} x f_x(x) dx = E[x|y] = \bar{x} \]

If \( x \) and \( y \) are independent, then they are uncorrelated. But, if they are uncorrelated, they may not be independent!

If \( x \) and \( y \) are uncorrelated, then they satisfy:

\[ E[xy^T] = E[x]E[y^T] \]

If \( x \) and \( y \) are uncorrelated, the cross-covariances must be zero:

\[ P_{xy} = E[(x - \bar{x})(y - \bar{y})^T] \]
\[ = E[xy^T - xy^T - \bar{x}y + \bar{x}\bar{y}] \]
\[ = E[xy^T] - E[x]E[y^T] + \bar{x}\bar{y}^T \]
\[ = E[xy^T] - \bar{x}\bar{y}^T + \bar{x}\bar{y}^T \]
\[ = E[xy^T] - \bar{x}\bar{y}^T = 0 \]

Why model the probability density function as a Gaussian (or normal) distribution:
• provides a good statistical model for many natural phenomena
• computationally tractable because the statistical properties are described completely by first (mean, \( \mu \)) and second (variance, \( \sigma^2 \)) moments
• normality is preserved through linear transforms (both static and dynamic)

A random vector \( x \in \mathbb{R}^n \) is Gaussian if the characteristic function has the form: \( P = [(x - \bar{x})(x - \bar{x})^T] \)

For such a random vector, we use the notation \( x \sim \text{N}(\mu, \sigma^2) \).

Two vectors \( x \) and \( y \) are jointly Gaussian distributed if \( (x^T, y^T) \) is Gaussian.

When a random vector \( x \) is Gaussian and its covariance matrix \( P \) is nonsingular, its PDF can be evaluated with:

\[
P(x) = \frac{1}{(2\pi)^{n/2}|P|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu)^T P^{-1} (x - \mu)\right)
\]

If \( P \) is singular, then the above equation will not work, but the characteristic function can be found the Gaussian distribution indirectly.

In a random process, we are looking at a family of random vectors \( (x(t), t \in I) \) indexed by time.

\[
P(t, \tau) = E[(x(t) - \bar{x}(t))(x(t) - \bar{x}(t))^T], t \in I
\]

For two random processes \( x(t) \) and \( y(t) \)

\[
P_{xy}(t, \tau) = E[(x(t) - \bar{x}(t))(y(t) - \bar{y}(t))^T], t \in I
\]

Which satisfies:

\[
P(t) = E[x(t)x^T(t)] - \bar{x}(t)\bar{x}^T(t)
\]

Which satisfies:

\[
P_{xy}(t) = E[x(t)y^T(t)] - \bar{x}(t)\bar{y}^T(t)
\]

todo..

todo..

This section combines the ideas of Gaussian distribution and random process

• a random process is Gaussian if all of the vectors \( x_1(t), \ldots x_n(t) \) are jointly Gaussian
• a Gaussian random process is white if the vectors \( x(t_1), \ldots x(t_m) \) are independent, otherwise it is colored
• for a Gaussian and white process, the covariance kernel satisfies

\[
P(t, \tau) = 0, t \neq \tau
\]

\[
P(t, \tau) = Q(t)\delta(t - \tau)
\]

• a random process is Markov if

\[
f(x(t_m)|x(t_m-1), \ldots, x(t_1)) = f(x(t_m)|x(t_m-1))
\]

• a random process is Gauss-Markov if it is both Gauss and Markov
The standard model is:

\[
\dot{x}(t) = A(t)x(t) + B(t)u(t) + w(t), \quad t \geq t_0
\]
\[
y(t) = C(t)x(t) + v(t)
\]

1. the initial condition \(x(t_0)\) is Gaussian

the disturbance \(w(t)\) is a zero-mean, Gaussian, white process that is independent of \(x(t_0)\) covariance \(\sigma_w\) given for these second state variable then

the measurement noise \(v(t)\) is a zero-mean, Gaussian, white process that is independent of \(x(t_0)\)

the processes \(v(t)\) and \(w(t)\) are uncorrelated

\[
E[w(t)v^T(t)] = 0
\]

With the standard model and assumptions, the process \(x(t)\) is Markov.

Recall the variation of constants formula:

\[
x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau + ... + \int_{t_0}^{t} \Phi(t, \tau)w(\tau)d\tau
\]

Then,

\[
x(t_m) = \Phi(t_m, t_{m-1})x(t_{m-1}) + ... + \int_{t_{m-1}}^{t_m} \Phi(t_m, \tau)(B(\tau)u(\tau) + w(\tau))d\tau
\]

Notice that the result does not depend on: \(\mathcal{X}(\tau)\), \(\tau < t_{m-1}\).

Also referred to as the expected value.

\[
\bar{x}(t) = \Phi(t, t_0)\bar{x}(t_0) + \int_{t_0}^{t} \Phi(t, \tau)B(\tau)u(\tau)d\tau
\]
\[
\bar{y}(t) = E[C(t)x(t) + v(t)] = C(t)\bar{x}(t)
\]

Using \(x(t)-(t)\) we can define:

\[
P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \int_{t_0}^{t} \Phi(t, \tau)R_w(\tau)\Phi^T(t, \tau)d\tau
\]

The above equation also satisfies the Lyapunov Equation.
For the output $y(t)$:

$$P_y(t) = C(t)P(t)C^T(t) + R_v(t)$$

with the standard model and assumptions, the covariance matrix satisfies:

$$\dot{P}(t) = A(t)P(t) + P(t)A^T(t) + R_w(t)$$

Which can be derived using the **Leibniz Integral Rule**.

To find the steady state covariance matrix:

- set $(t)=0$
- then solve for $P_{ss}$
  - this will require a computer!

For the state $x(t)$:

$$P_x(t, \tau) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \int_{t_0}^t \Phi(t, \sigma)R_w(\sigma)\Phi^T(t, \sigma)d\sigma$$

For the output $y(t)$:

$$P_y(t, \tau) = C(t)\Phi(t, t_0)P(t_0)\Phi^T(t, t_0)C^T(\tau) + \int_{t_0}^t C(t)\Phi(t, \sigma)R_w(\sigma)\Phi^T(t, \sigma)C^T(\tau)d\sigma + R_v(t)\delta(t - \tau)$$

After evaluating $f(x,y)$, the results is PDF of a Gaussian vector defined as:

$$P_{x|y} = P_x + P_{xy}P_y^{-1}P_{yx}$$

Basic estimation procedure is:

1. determine $\hat{x}$ which is an estimate of the state
2. based off of state equations
3. affected by state uncertainty, $w(t)$

collect measurements from output equation $y(t)$

update estimate of state $\hat{x}$ basedoff newinfo from $y(t)$

based off of output equation

affected by sensor uncertainty, $v(t)$

Measurements are incorporated simultaneously.

$$\dot{\hat{x}}^+ = \hat{x}^- + k(z - C\hat{x}^-)$$

$$P^+ = P^-C^T(CP^-C^T + R)^{-1}$$

where: $k = P^-C^T(CP^-C^T + R)^{-1}$ is obtained algebraically.

Equivalent form,

$$\dot{\hat{x}}^+ = (P^+(P^-)^{-1})\hat{x}^- + (P^+C^TR^{-1})z$$

$$P^+ = ((P^-)^{-1} + C^TR^{-1}C)^{-1}$$

where: $R$ is the covariance of the sensor measurement from...
\( y(t) \)

also note: \( z = y \)

Measurements are incorporated recursively.