

Seminar on retrofit technologies for autonomous mobile robot

2023.3.1

特別講座 #02

NEDO特別講座

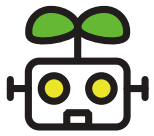
# 中小建設業ROS活用人材育成講座

シミュレータを利用したダンプ智能化ソフトウェアの実習

講演者 小島 匠太郎

所属 東北大学未来科学技術共同研究センター





# 概要

---

- 目標:

- シミュレータ上でダンプトラックを自動走行させ、知能化ソフトウェアに関する理解を深める

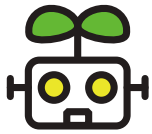
- 講習内容:

- 知能化ソフトウェアの概要説明(10分程度)

1. 知能化ソフトウェアでできること+今回の実習でやること
2. ソフトウェアの構成

- シミュレータを利用した自動走行

1. プログラムの起動
2. 目的地を指令して自動走行
3. 自動走行用の地図を編集する
4. joypadでダンプを操作してみる

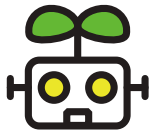


Seminar on  
Retrofit  
Technologies  
for Autonomous  
Mobile Robot

# 知能化ソフトウェアでできること

複数のソフトウェアを統合して土砂積み込み・運搬・積み下ろしを実現



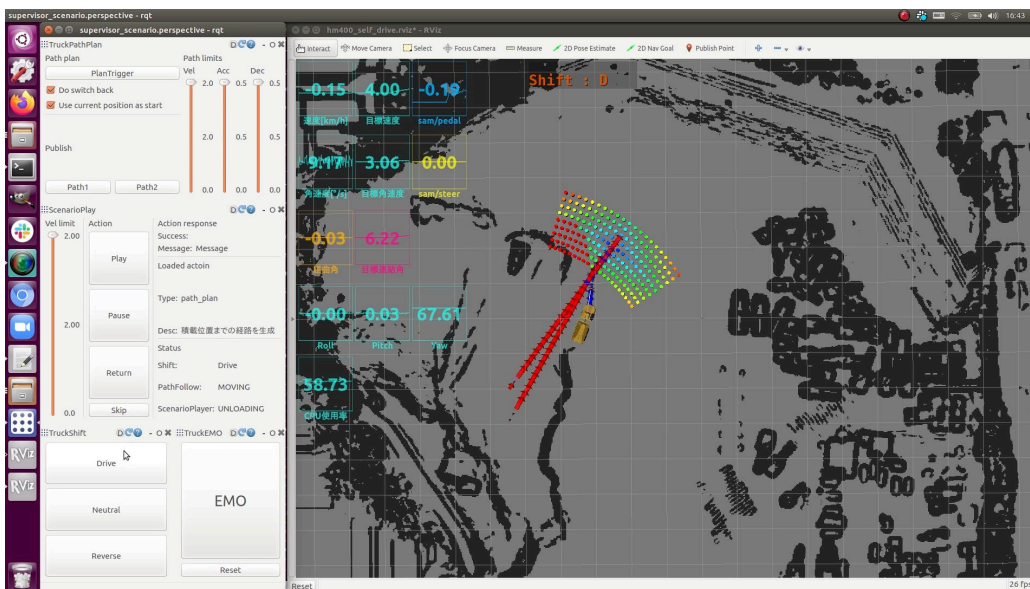
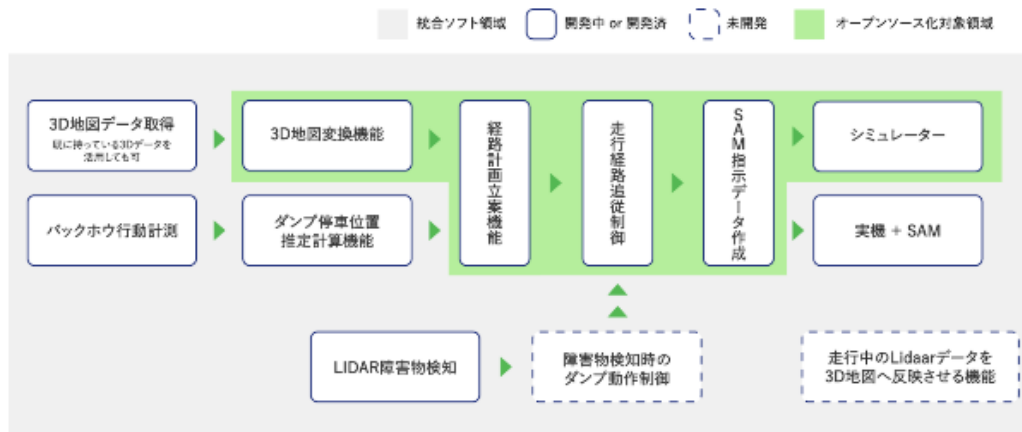
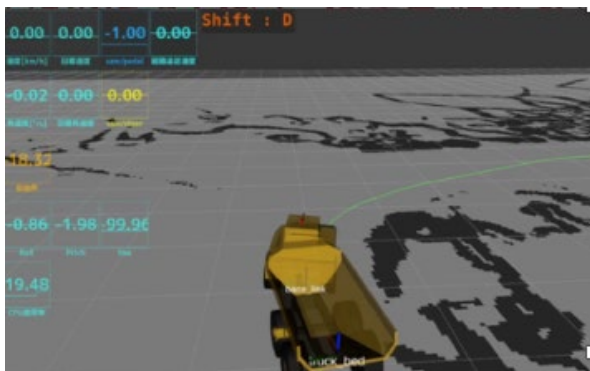


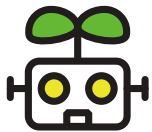
Seminar on  
Retrofit  
Technologies  
for Autonomous  
Mobile Robot

# 今回の実習：シミュレータを利用した自動運転

- ご自身のPCにシミュレータをインストールして開発を体験※
- Windows PC, WMWare Player、開発環境イメージ

※下記の緑の部分の利用した自動運転開発を体験





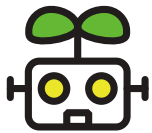
# 実習でやること①: シミュレータ上でダンプを自動走行

ゴールまでの経路を計画し、シミュレータ上のダンプトラックを自動で走行させる

The screenshot displays a simulation environment with a path planning interface. The interface includes a 'TruckPathPlan' window with the following settings:

- Path plan: PlanTrigger
- Path limits: Vel (2.0), Acc (0.5), Dec (0.5)
- Do switch back:
- Use current position as start:
- Buttons: Drive, Neutral, Reverse, EMO, Reset

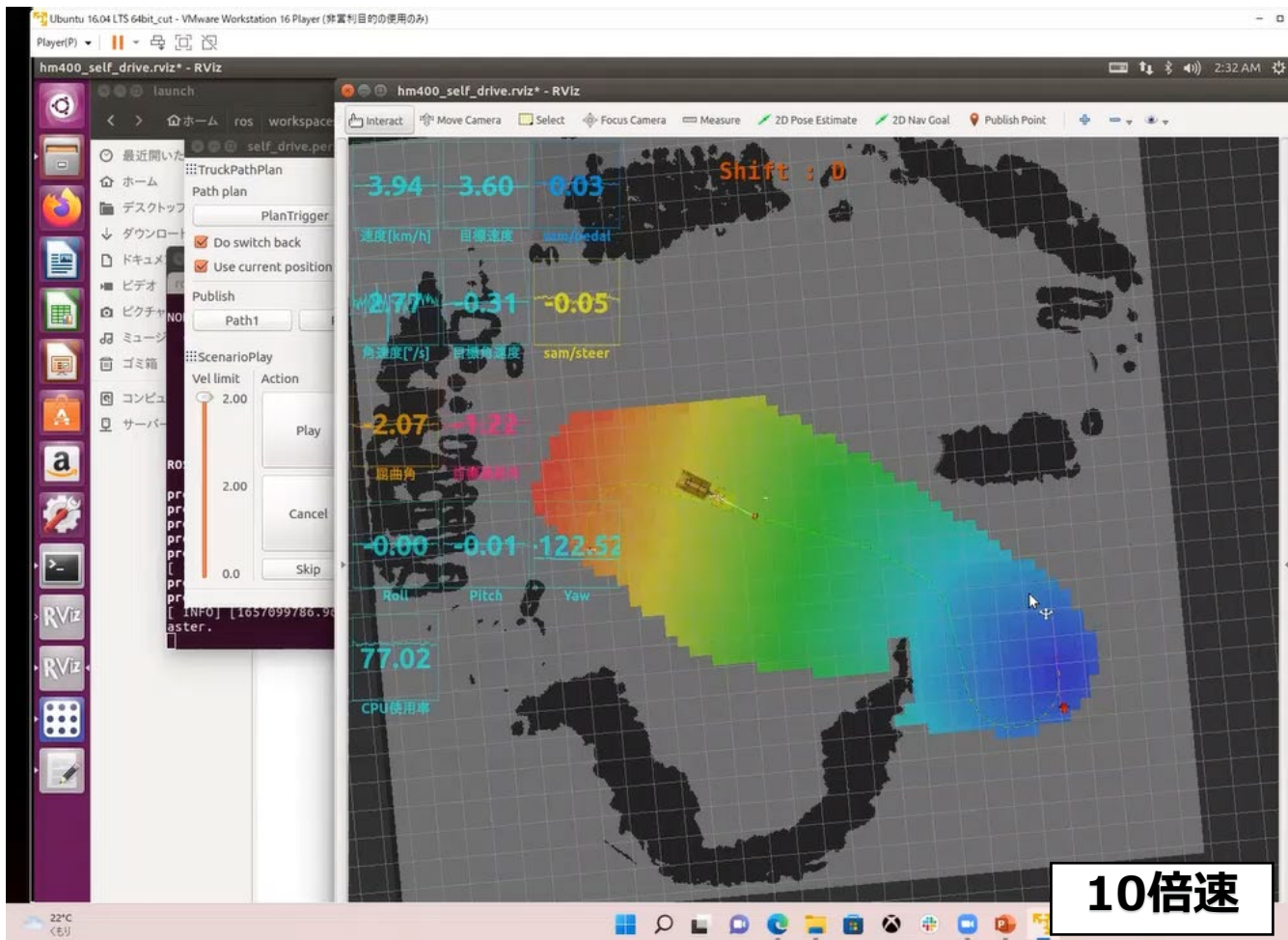
The 'ScenarioPlay' window shows a velocity limit of 2.00 and a 'Play' button. The background shows a 2D map with a path and a speedometer. A box in the bottom right corner indicates '5倍速' (5x speed).

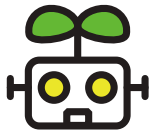


# 実習でやること②: 地図の変更

自動走行に使用する地図を画像エディタで編集して障害物を消してみる

## 2. 編集した地図上で自動走行

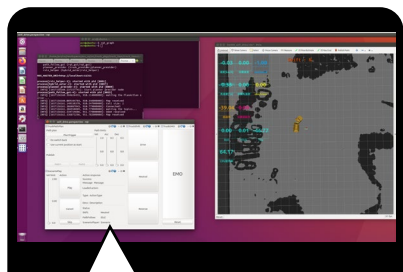




# 実習でやること③: ゲームコントローラでダンプを遠隔操縦

シミュレータ上のダンプを自動走行モードから遠隔操縦モードに切替え、プログラムの構成がどのように変わるか解説。ダンプの操作を体験。

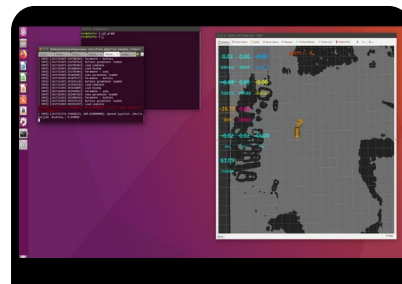
### 自動走行モード



画面上のパネルで操作

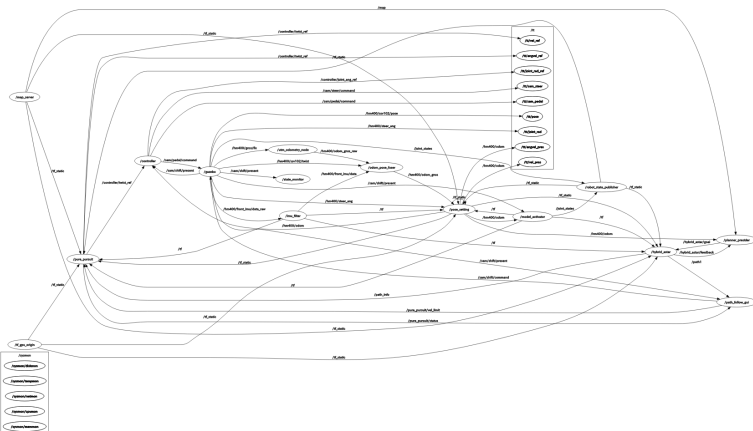
モード  
切替

### 遠隔操縦モード

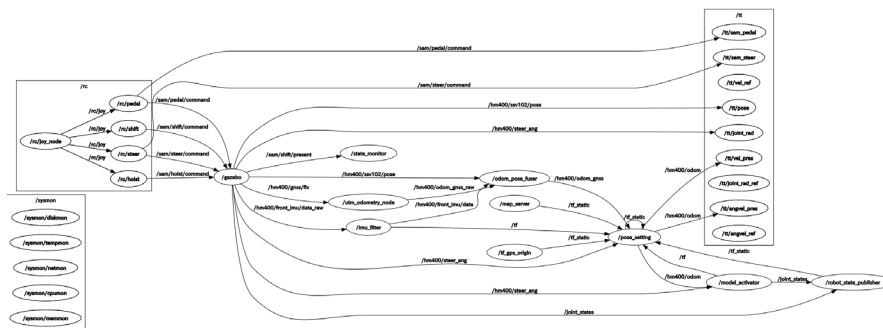


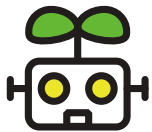
ゲームコントローラで操作

### ソフトウェア構成(自動)



### ソフトウェア構成(遠隔)

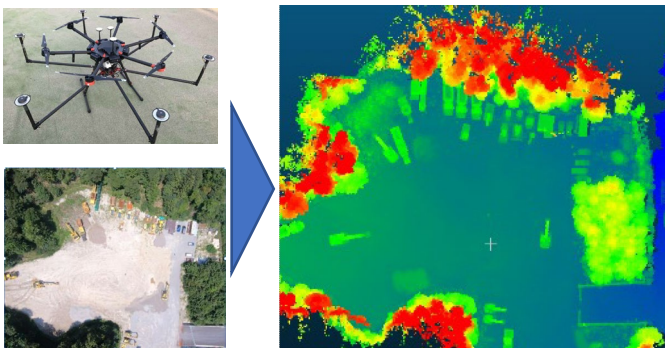




# 参考:3次元の地形データから自動走行用地図を生成

- ドローンで計測した3次元データを自動走行用の2次元地図に変換する機能
  - オープンソースで公開:  
<https://github.com/TohokuACV/TohokuACV>
- 今回の実習では変換済の2次元地図を使用します。

## 3次元データ(ドローンで計測)



## 2次元データ(自動走行に使用)



黒: 走行不可  
白: 走行可能

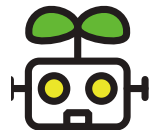
## ↓オープンソース公開ページより引用

### 1. 地図情報の変換

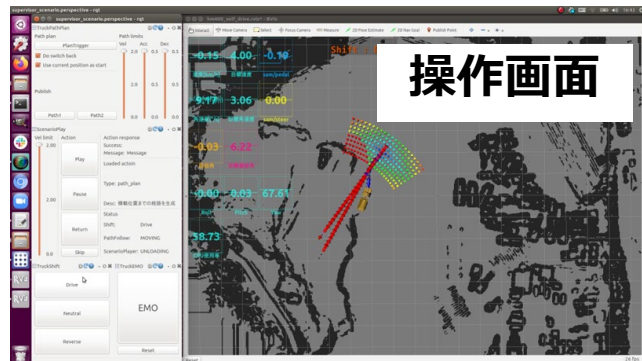
- UAVによる測定で得られた点群(Point Cloud)情報を、経路計画で利用できるOccupancy Gridに変換する作業を行う。
- 地図情報の変換は経路計画のたびに行う必要はなく、測定で得られた点群情報に対して一回行えばよい。

機能	プロジェクト/ソース	処理概要
①点群情報の読み込み	convert_to_dem/ text_to_dem.py	• UAVにより測定され、XYZの各座標に分けて記した3つのCSVファイルに記載されている点群情報を読み込む • CSVファイルから読み込んだデータをDataframeオブジェクト(サイズの動的変更が可能な2次元配列)に展開してから、DEM(数値標高モデル)に変換する • 作成したDEMをトピックとして配信する
②DEM情報へ変換		
③変換後のフィルター処理	convert_to_dem/ filters_demo	• 作成したDEMに対して情報を付加する場合に用いるフィルターチェーン
④Occupancy Gridへ変換	dem_to_occupancy/ dem_to_occupancy. cpp	• DEM情報を受け取り、Occupancy Grid(車両の通過可能/不可能を色付きのドットで示した地図情報)に変換して返す • ROSのサービスノードとして動作する
	convert_to_dem/ dem_to_occupancy_ client.py	• text_to_dem.pyから受け取ったDEM情報を受け取る • サービスノードであるdem_to_occupancyを呼び出し、DEM情報をOccupancy Grid情報に変換する • Occupancy Gridをトピックとして配信する



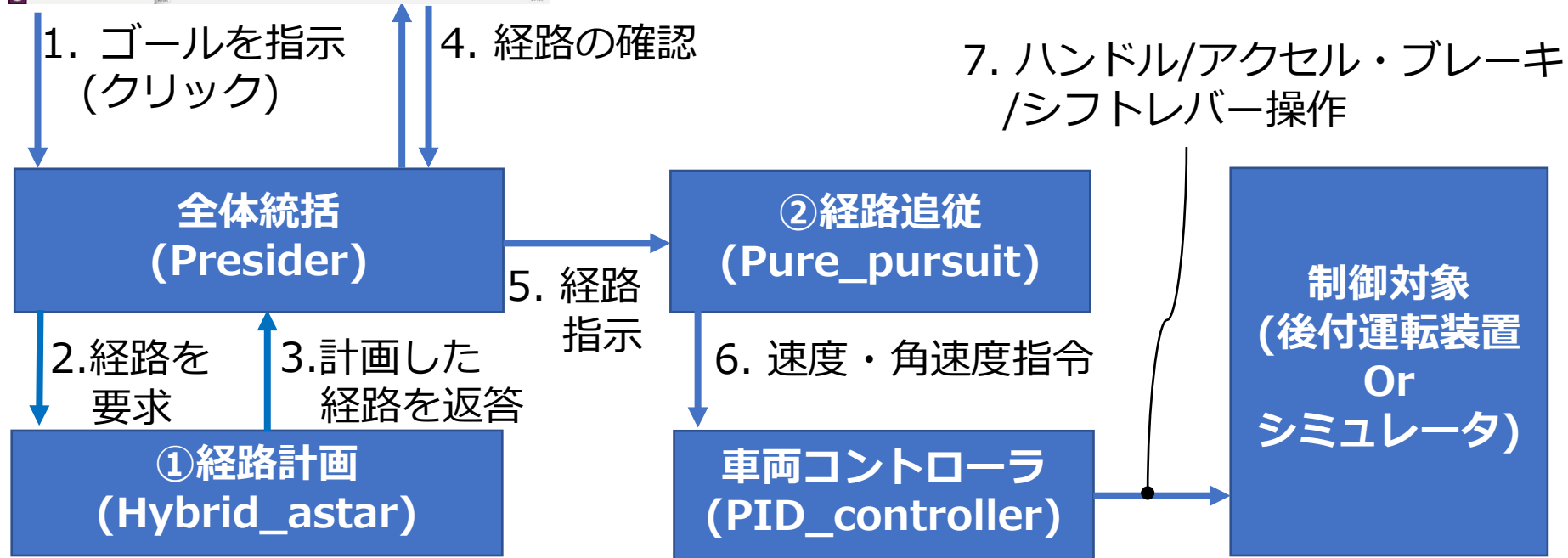


# ソフトウェアの構成:全体像

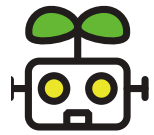


複数のROSノード(機能のブロック)が互いに通信・連携することで全体を構成

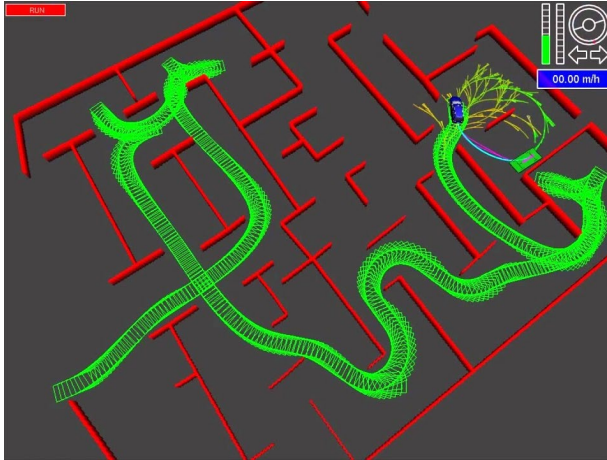
■ : ROSノード  
→ : トピック通信



※コアとなる機能①経路計画 と ②経路追従について次以降のスライドで解説

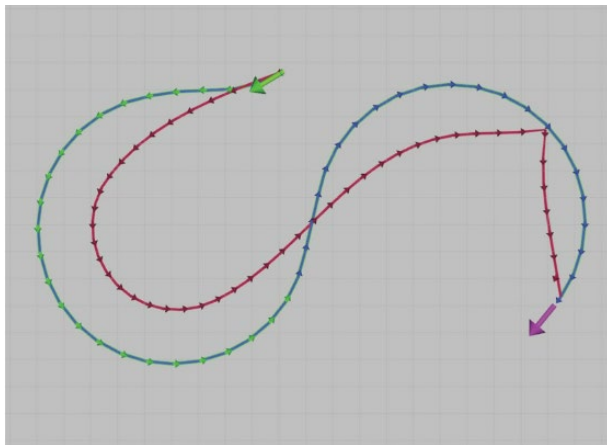


# ソフトウェアの構成:経路計画(Hybrid astar<sup>[1]</sup>)



<https://blog.habrador.com/2015/11/explaining-hybrid-star-pathfinding.html>

- Hybrid astar<sup>[1]</sup>
  - 自動車などの経路計画に用いられる経路計画の方法
  - 操舵角に制約がある自動車が追従可能な経路を計画

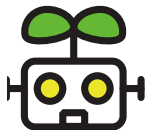


赤: 改善前 青: 改善後

- スムーザーの改良<sup>[2]</sup>
  - 計画後の経路を滑らかにする“スムーザー”機能を独自に改良
  - 経路の形によってはスムージングに失敗するという課題を解決

[1] Dolgov, Dmitri, et al. "Path planning for autonomous vehicles in unknown semi-structured environments." *The international journal of robotics research* 29.5 (2010): 485-501.

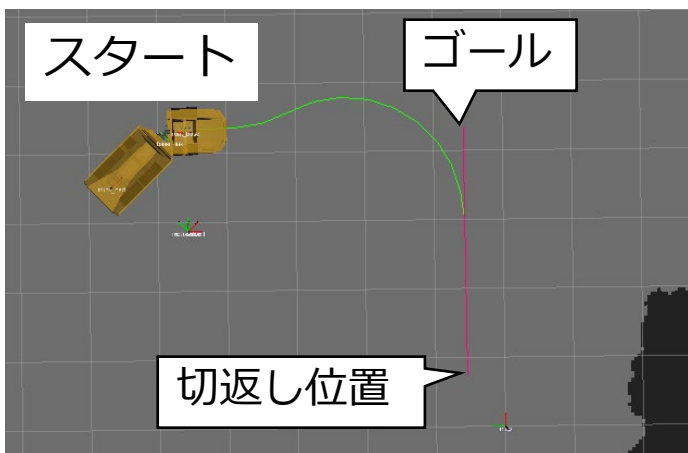
[2] Mizuno, Naoki, et al. "Enhanced path smoothing based on conjugate gradient descent for firefighting robots in petrochemical complexes." *Advanced Robotics* 33.14 (2019): 687-698.



# ソフトウェアの構成:Hybrid astarのダンプへの適用

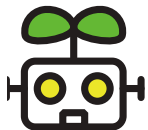


- 積込・積降の際の切返し動作
  - 停車位置に荷台側から近づくため、停車位置の正面で切返し
  - 自動でも現場で運用されているダンプに近い動作を実現したい

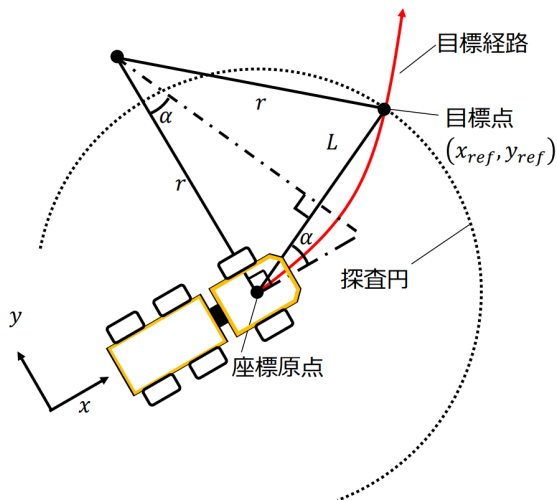


- 停車位置の正面で切り返す経路の計画
  - ゴール正面に経由点を置くことで、必ず停車位置の正面で切り返す経路を生成
  - 現場のダンプに近い切返し動作が可能

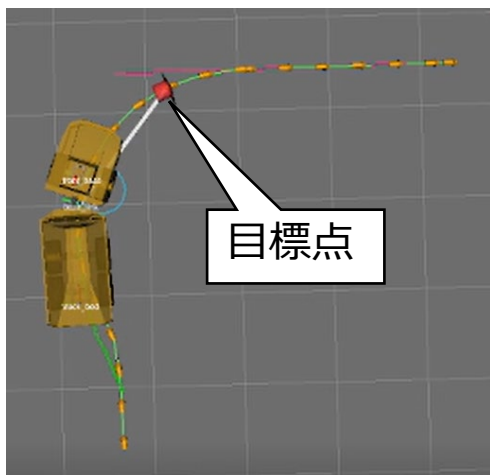
緑: 経路1(スタート-> 切返し)  
紫: 経路2(切り返し-> ゴール)



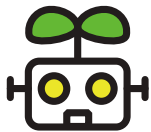
# ソフトウェアの構成:経路追従(Pure pursuit<sub>[3]</sub>)



- Pure pursuit<sub>[3]</sub>
  - ロボットを中心とする探査円と追従する経路の交点を目標点に設定
  - 目標点に到達するための並進速度 $v$ , 角速度 $w$  を算出、ロボットに指令



- 実習で使用するソフトでの表示
  - 自動走行中のダンプの正面に、pure pursuitの目標点が表示される
  - ロボットの移動に伴い目標点も移動する様子が確認できる



# ここから実習に移ります！

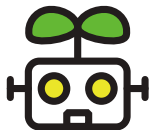
## 前提：

- インストールマニュアル10Pまでの手順が完了していること
- [https://www.dropbox.com/s/y4ztoqchtog5xsg/20221030\\_ACV%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB%E3%83%9E%E3%83%8B%E3%83%A5%E3%82%A2%E3%83%AB\\_VMWare%E3%82%A4%E3%83%A1%E3%83%BC%E3%82%B8\\_V0.2.pdf?dl=0](https://www.dropbox.com/s/y4ztoqchtog5xsg/20221030_ACV%E3%82%A4%E3%83%B3%E3%82%B9%E3%83%88%E3%83%BC%E3%83%AB%E3%83%9E%E3%83%8B%E3%83%A5%E3%82%A2%E3%83%AB_VMWare%E3%82%A4%E3%83%A1%E3%83%BC%E3%82%B8_V0.2.pdf?dl=0)

## 項目：

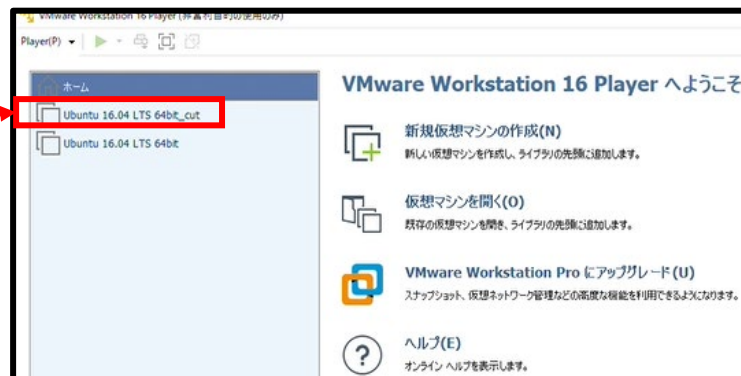
1. 仮想マシンとプログラムの起動
2. 目的地を指令して自動走行
3. 自動走行用の地図を編集する
4. joypadでダンプを操作してみる

※トラブル等についてこれない場合は、聴講のみに切り替えていただく場合があります。ご了承ください。

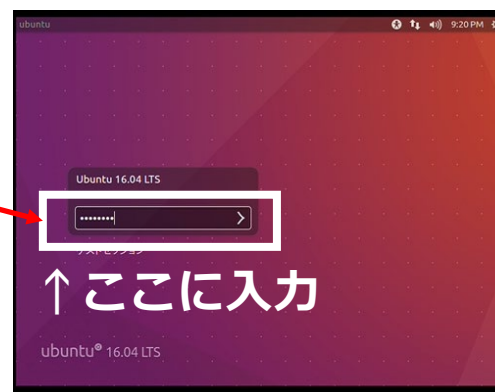


# 1. 仮想マシンとプログラムの起動：仮想マシン

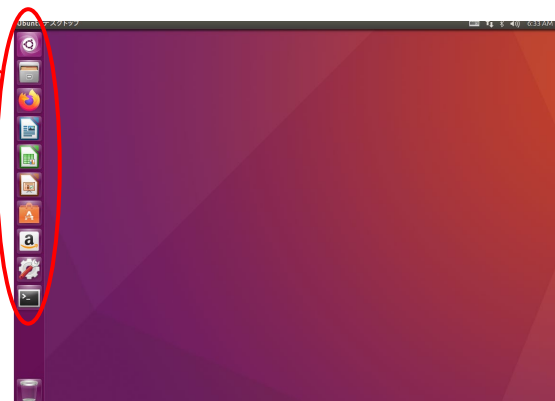
1. VMware playerを開き,  
"Ubuntu 16.04 LTS 64bit\_cut"  
をクリック

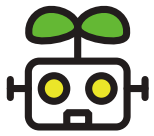


2. パスワードを聞かれるので,  
"nedokouza"と入力



3. 画面の左端に縦に並んだアイコン  
が出てきたら起動完了





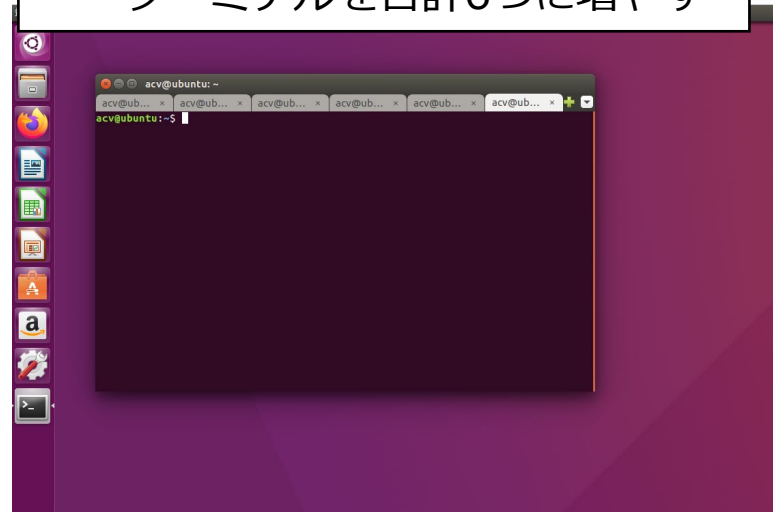
# 1. 仮想マシンとプログラムの起動：プログラム

1. 画面左端の黒いアイコンをクリック



2. “新しい端末”をクリック

3. Ctrl + Shift + t を5回押し、ターミナルを合計6つに増やす



4. ターミナル6つそれぞれで下記のコマンドを実行

“roscore”

“roslaunch truck\_launch simulator.launch”

“roslaunch truck\_launch hm400.launch”

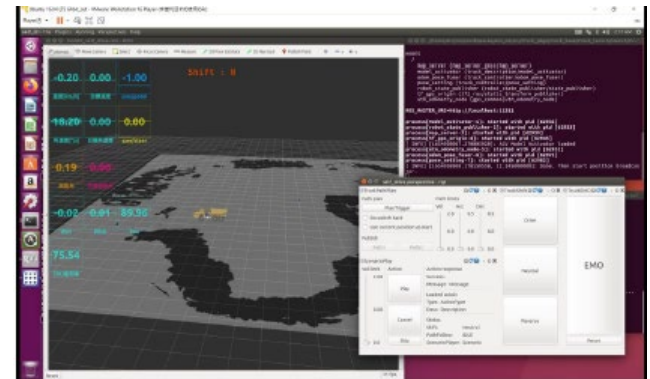
“roslaunch truck\_launch truck\_visualize.launch”

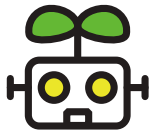
“roslaunch truck\_launch controller.launch”

“roslaunch truck\_launch self\_drive.launch”

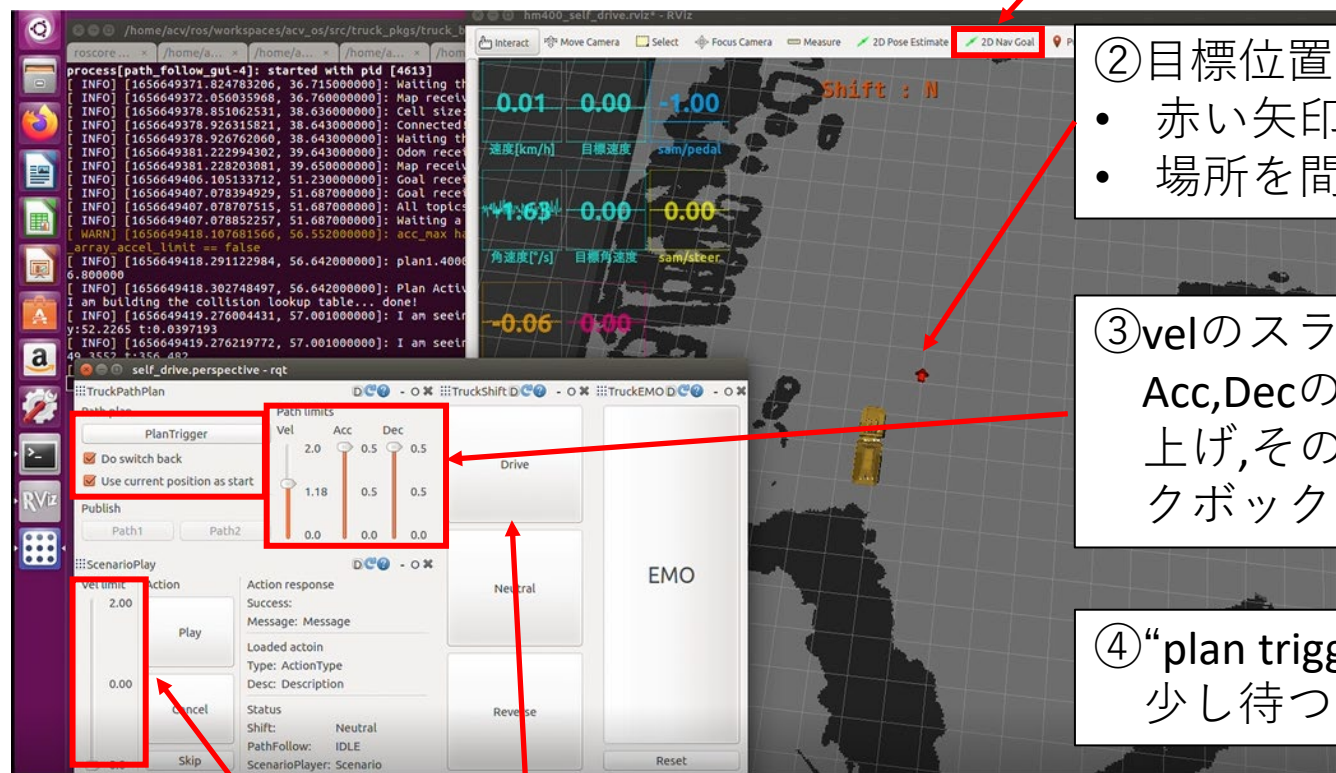
※前後の””はコマンドに含みません。

↓全部起動するとこうなります





## 2.目的地を指令して自動走行



① 2d nav goalをクリック

② 目標位置をクリック

- 赤い矢印が地図上に出現
- 場所を間違たらやり直し可能

③ velのスライダを半分まで、Acc,Decのスライダを最大まで上げ、その左側の2つのチェックボックスにチェック

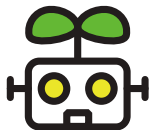
④ “plan trigger”をクリック  
少し待つと経路が生成される

⑥ Driveボタンを押してシフトをドライブへ入れ，“Vel limit”スライダを上げると走行開始。

⑤ 経路が生成されたらPath1ボタンをクリック

⑦ ゴールに到着したらvel\_limitを0まで下げ、Neutralボタンをクリック。





# 3.自動走行用の地図を編集する:地図データのコピー

1. 画面左端の上から2番目のアイコンをクリック

2. 下記フォルダに移動  
ros/workspaces/acv\_os/src/truck\_pkgs/truck\_base/truck\_launch/map

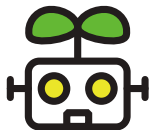
3. sanbongi.pgmを右クリックして「指定先にコピー」を選択

4. 右の画面が出たら「選択」をクリック  
「sanbongi(コピー).pgm」ができる

5. 「sanbongi(コピー).pgm」を右クリックし、ファイル名を「sanbongi\_edit.pgm」に変更

6. 1-5と同様の手順で「sanbongi.yaml」の  
コピー「sanbongi\_edit.yaml」を作成



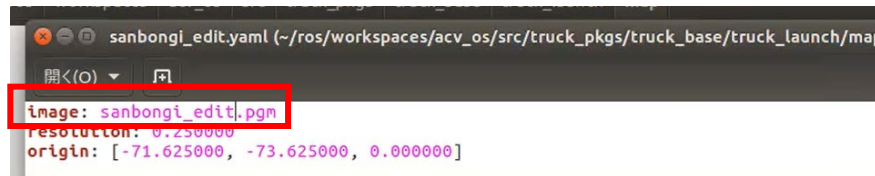


### 3.自動走行用の地図を編集する:使用する地図の切り替え

コピーした地図を自動走行に使用するように設定を変更します。

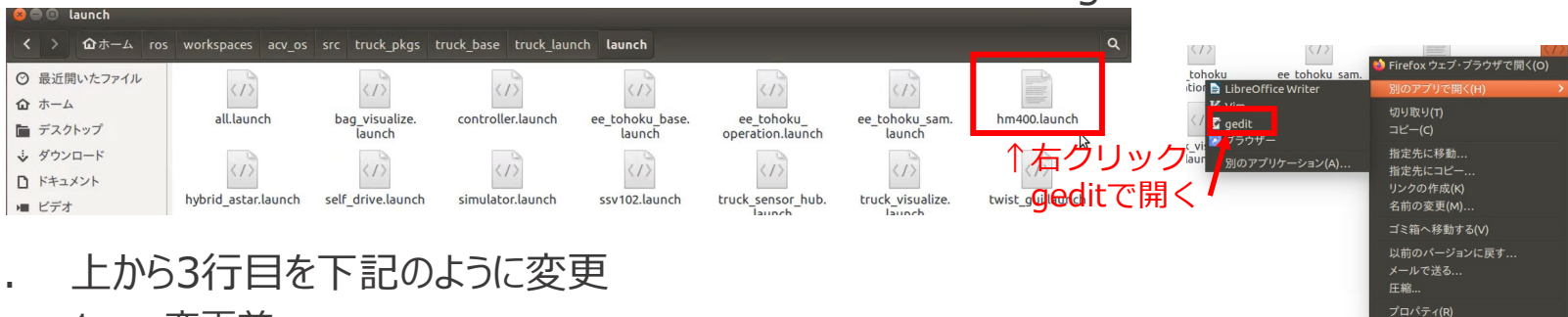
#### 1. Sanbongi\_edit.yamlの変更

1. ひとつ前のスライドで作成したsanbongi\_edit.yamlをダブルクリックで開く
2. 一番最初の行を変更
  1. 変更前: image: sanbongi.pgm
  2. 変更後: image: sanbongi\_\*\*edit\*\*.pgm



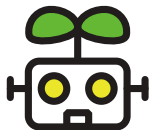
#### 2. Hm400.launchの変更

1. ファイルエクスプローラで“ホーム /ros/workspaces/acv\_os/src/truck\_pkg/truck\_base/truck\_launch/launch”に移動、hm400.launchを右クリックし、“別のアプリで開く->gedit”を選択



#### 2. 上から3行目を下記のように変更

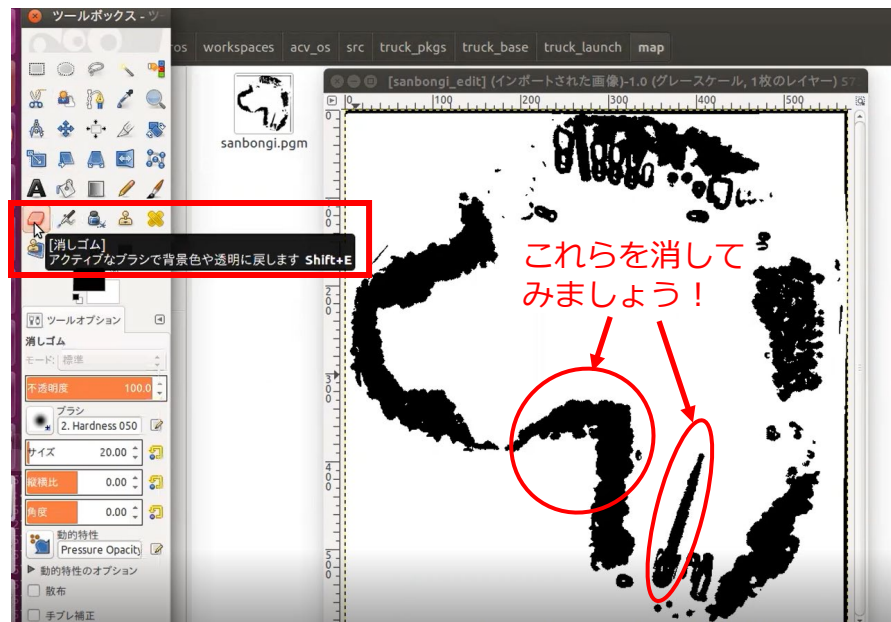
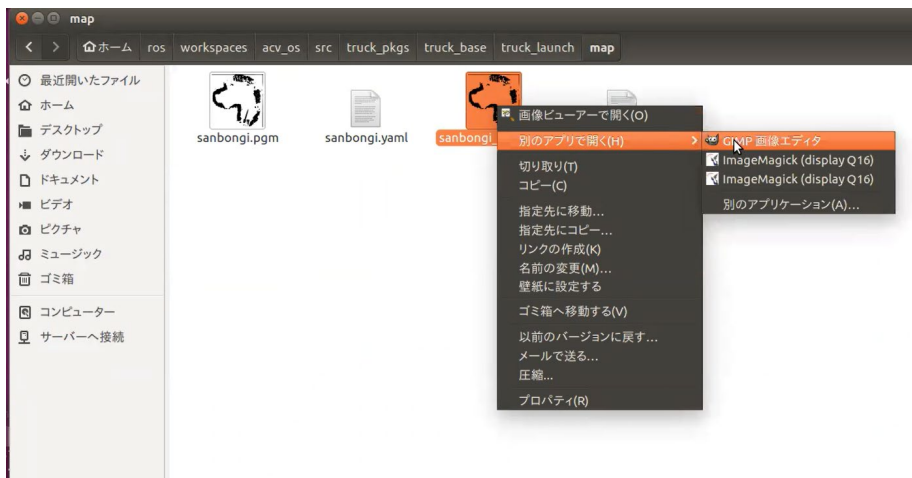
1. 変更前:  
`<arg name="map_file" default="$ (find truck_launch)/map/sanbongi.yaml" />`
2. 変更後:  
`<arg name="map_file" default="$ (find truck_launch)/map/sanbongi_**edit**.yaml" />`

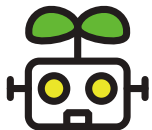


# 3.自動走行用の地図を編集する:地図の編集

## 地図データを編集

1. 以前のスライドでコピーした「sanbongi\_edit.pgm」を右クリックし、別のアプリで開く->gimp画像エディタ を 選択
2. 消しゴムを選択し、右下の図の赤丸で囲んだ障害物を消してみましよう。
3. 最後に ファイル->上書きエクスポート で保存





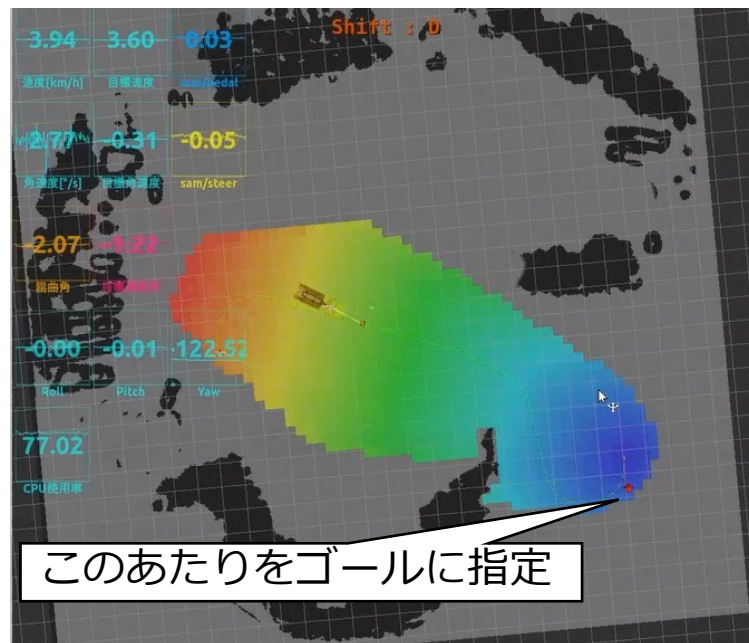
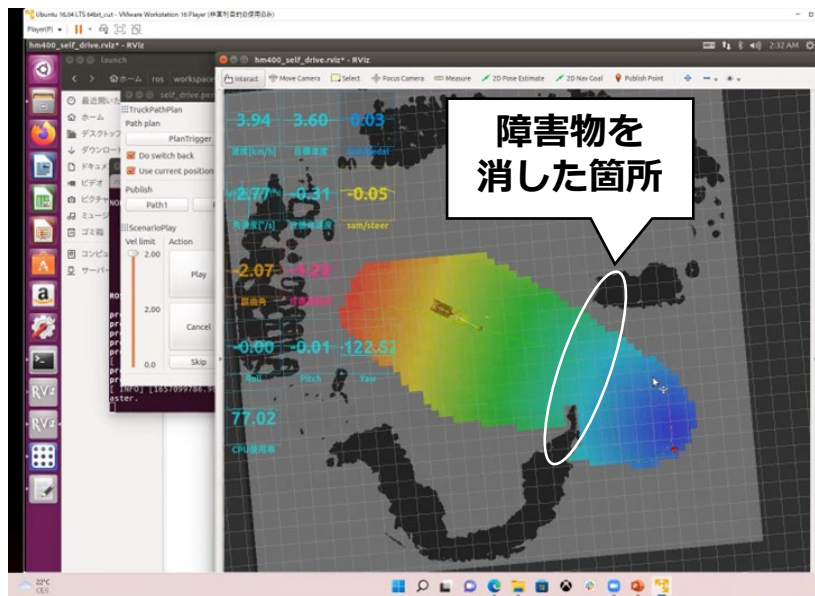
### 3.自動走行用の地図を編集する:新しい地図で自動走行

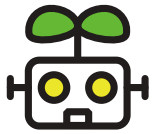
#### 1. 地図のロード

1. 最初に起動したプログラムのうち、“roslaunch truck\_launch hm400.launch”を再起動  
ターミナルを選択して “Ctrl + c”で終了、上矢印キーで” roslaunch … hm400.launch”を選択してenterキーを押すと再起動できます。
2. 左下の図のように、マップ上にあった障害物（黒塗りの部分）が消えているはずです。

#### 2. 新しい地図上での自動走行

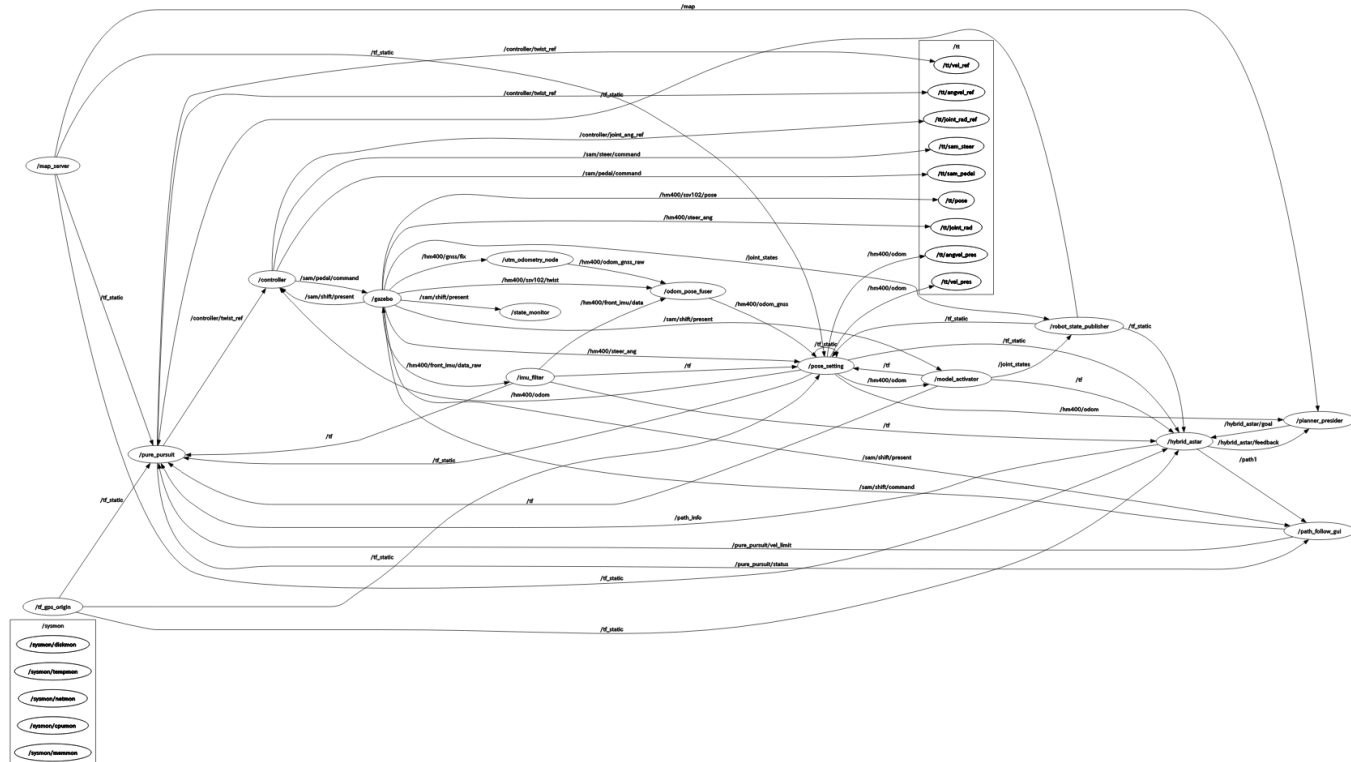
1. 障害物を消した場所を通過して、フィールドの外側に自動走行で出てみましょう。  
◆ 地図上のフィールド外の場所を自動走行のゴールとして指定します(右下)。

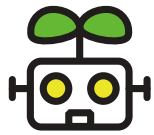




## 4. 遠隔操作でダンプを操作してみる:事前準備

1. 自動走行操作パネルで“neutral”ボタンを押してシフトレバーをneutralにし、左下の“vel limit”のスライダを0に下げる
  - 確実にダンプを静止させるため。
2. 新しいターミナルを開き、“rqt\_graph”コマンドを実行
  - 自動走行時のソフトウェア構成図が出てきます。内容は講義で簡単に説明します。
  - 後ほど、遠隔操縦に切り替えたら再度このグラフを見てみます。





## 4. 遠隔操作でダンプを操作してみる:自動/遠隔操縦の切替

### 1. Joypadの接続

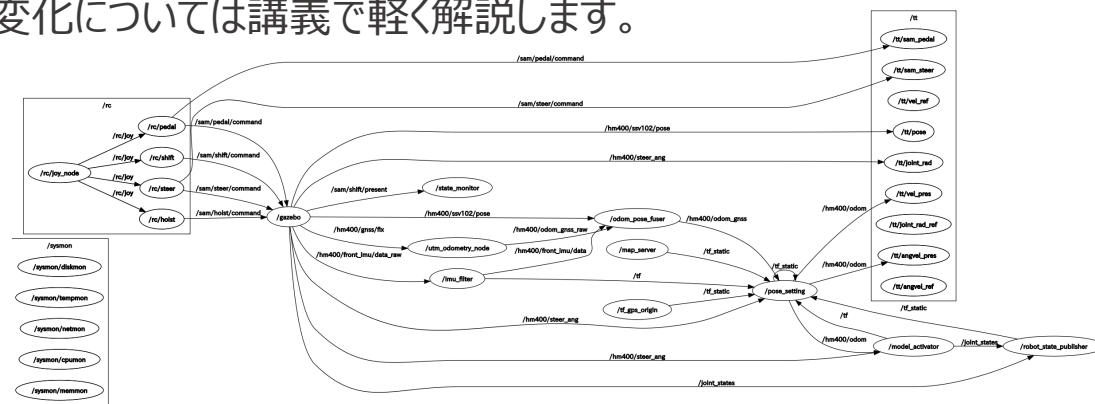
1. usbケーブルでPS4のコントローラ(dual shock 4)をパソコンに接続します。
2. 仮想マシンに接続するか聞かれた場合は、「仮想マシンに接続」を選択します。
3. 聞かれない場合はVMwareの左上のプルダウンから「取り外し可能デバイス」を選択し、「Sony wireless controller」を仮想マシンに接続します。

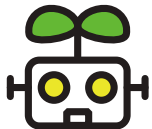
### 2. ソフトウェアの切替

1. 自動走行時に起動したプログラムのうち、下記2つをCtrl + c で停止します  
“roslaunch truck\_launch controller.launch”  
“roslaunch truck\_launch self\_drive.launch”
2. 新しいターミナルを開き、下記を実行します。  
“roslaunch joy\_rc sam\_rc.launch”

### 3. Rqt\_graphをもう一度見てみましょう

1. 左上の更新ボタンを押すと、変更後のソフトウェア構成が表示されます。自動走行時からの変化については講義で軽く解説します。





## 4. 遠隔操作でダンプを操作してみる:実際に走行させてみましょう!

### 1. 発車の手順

1. 左スティックを後ろに倒しながらR1と○ボタンの同時押しでギアがドライブに切り替わります。
2. 左スティックを前に倒すとアクセルが踏まれ、前進します。
3. 右スティックを左右に倒すとハンドルが操作できます。
4. 止めたいときは左スティックを後ろに倒してブレーキを掛けます。

### 2. 停車の手順

1. 左スティックを後ろに倒してブレーキを掛けます。
2. R1と×ボタンの同時押しでギアがニュートラルに切り替わります。
3. これでアクセルを踏んでも前進しなくなります。  
※実際のダンプにはパーキングボタンが別にあります  
が、シミュレータ上のダンプにはありません。



ダンプ	コントローラー	操作	備考
ペダル	L ジョイスティック上	アクセル	-
	L ジョイスティック下	ブレーキ	-
ステアリング	R ジョイスティック左	左ハンドル	-
	R ジョイスティック右	右ハンドル	-
シフトチェンジ	R1+○	Drive	ブレーキをかけて実行
	R1+×	Neutral	ブレーキをかけて実行
	R1+△	Reverse	ブレーキをかけて実行