

Nama : Siti An-nisaa

NIM : 1103190023

## UTS Robotika dan Sistem Cerdas

# UTS – Robot Autonomy I

### • Reproduce Codes and Make Technical Documentation

#### • Target (Choose 1):

1. <https://github.com/laszlo-schaffer/webots-szte-robocup-2022>
2. [https://github.com/salvatorezam/motion\\_planning\\_and\\_target\\_searching\\_in\\_complex\\_environments](https://github.com/salvatorezam/motion_planning_and_target_searching_in_complex_environments)
3. <https://github.com/silvery107/auto-docking-vessels>
4. <https://symforce.org/>
5. <https://carla.org/> or Drive Sim <https://www.nvidia.com/en-us/self-driving-cars/simulation/>

#### • Deliverables:

- Clone the codes in your git
- Make technical documentation in PDF (PPTX turn to PDF) and upload to git
- Send your git link to google drive (will be given)

#### • Deadline: 2 Week (Nov 19<sup>th</sup> to Dec 3<sup>rd</sup>)

Berdasarkan petunjuk soal diatas saya memilih Symforce yang digunakan dalam pengerjaan UTS kali ini.

### Symforce

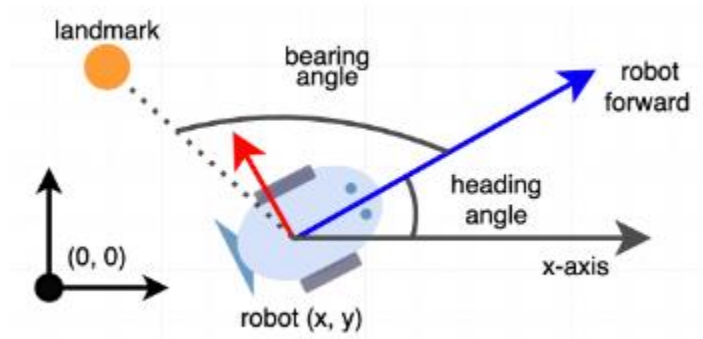
*Symforce* merupakan sebuah wadah untuk mengaplikasikan pembuatan kode untuk aplikasi robotika seperti *computer vision*, *state estimation*, *motion planning*, dan *controls*. *Symforce* ini menggabungkan kecepatan pengembangan dan fleksibilitas matematika dengan kinerja kode yang dibuat secara otomatis dan sangat dioptimalkan pada bahasa C++. *Symforce* ini juga merupakan open source yang memiliki library algoritma kode seperti SLAM, MPC, calibration, bundle adjustment, dan controls, yang menghasilkan :

1. *Faster runtime performance*, dengan *highly performant code generator*
2. *Faster development cycles*, yang menggunakan perhitungan turunan matematika
3. *Fewer line of code*, yang memantaint suatu masalah terhadap suatu formula dan mengautogeneratekan.

*Symforce* ini sendiri terdiri atas tiga workflow, yakni :

1. *Symbolic Math*, yang mana dimodelkan menggunakan bahasa python dengan sympy API yan termasuk dalam perhitungan geometri, camera models, dan perhitungan.
2. *Code Generation*, yang mana dapat melakukan introduct fungs simbolik dengan performa yang tinggi dan dibuat secara otomatis untuk bahsa apapun.

3. *Optimization*, dilakukan untuk mengenerated faktor realtime terhadap pengaplikasian robot menggunakan C++ atau Python.



Sebagai contoh, dapat dilihat pada gambar diatas bahwa sebuah robot harus menentukan :

1. Jarak
2. Dan arah yang dituju

Tahap ini merupakan tahap pertama yaitu *symbolic math* dalam menentukan pergerakan dari robot.

In [1]:

```
from symforce import sympy as sm
from symforce import geo
```

In [2]:

```
pose = geo.Pose2(
    t=geo.V2.symbolic("t"),
    R=geo.Rot2.symbolic("R")
)
landmark = geo.V2.symbolic("L")
```

Tahapan kedua yang dilakukan adalah code generation yang mana tahapan ini dilakukan untuk melakukan generated runtime function dari bentuk simbolik menjadi one command.

In [14]:

```
from symforce.codegen import Codegen, CppConfig

codegen = Codegen.function(bearing_residual, config=CppConfig())
```

Dan tahapan terakhir untuk melihat apakah tahapan yang dilakukan dalam *symforce* sudah akurat ialah optimizer. Yang mana optimizer ini dilakukan untuk apakah estimasi arah yang telah ditentukan sebelumnya telah akurat.

In [10]:

```
from symforce.opt.optimizer import Optimizer

optimizer = Optimizer(
    factors=factors,
    optimized_keys=[f"poses[{i}]" for i in range(num_poses)],
    # So that we save more information about each iteration, to visualize later:
    debug_stats=True,
)
```

In [13]:

```
from symforce.examples.robot_2d_triangulation.plotting import plot_solution
plot_solution(optimizer, result)
```

