UTS Robotics and Intelligent System

- Nama : Brilliant Friezka Aina
- NIM : 1103194186
- Kelas : TK-43-GAB

```
pip install symforce
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting symforce
  Downloading symforce-0.7.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.4 MB)
     |████████████████████████████████| 4.4 MB 5.4 MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.8/dist-packages (from symforce) (1.21.6)
Collecting skymarshal==0.7.0
  Downloading skymarshal-0.7.0-py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 312 kB/s
Collecting sympy~=1.11.1
  Downloading sympy-1.11.1-py3-none-any.whl (6.5 MB)
     |████████████████████████████████| 6.5 MB 42.5 MB/s
Requirement already satisfied: scipy in /usr/local/lib/python3.8/dist-packages (from symforce) (1.7.3)
Collecting clang-format
  Downloading clang_format-15.0.4-py2.py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 23.6 MB/s
Collecting black
  Downloading black-22.10.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 50.7 MB/s
Requirement already satisfied: jinja2 in /usr/local/lib/python3.8/dist-packages (from symforce) (2.11.3)
Collecting symforce-sym==0.7.0
  Downloading symforce_sym-0.7.0-py3-none-any.whl (70 kB)
     |████████████████████████████████| 70 kB 4.6 MB/s
Requirement already satisfied: graphviz in /usr/local/lib/python3.8/dist-packages (from symforce) (0.10.1)
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from skymarshal==0.7.0->symforce) (1.15.0)
Collecting ply
  Downloading ply-3.11-py2.py3-none-any.whl (49 kB)
     |████████████████████████████████| 49 kB 3.9 MB/s
Collecting argh
  Downloading argh-0.26.2-py2.py3-none-any.whl (30 kB)
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.8/dist-packages (from sympy~=1.11.1->symforce) (1.2.1)
Requirement already satisfied: typing-extensions>=3.10.0.0 in /usr/local/lib/python3.8/dist-packages (from black->symforce) (4.1.1)
Requirement already satisfied: tomli>=1.1.0 in /usr/local/lib/python3.8/dist-packages (from black->symforce) (2.0.1)
Collecting platformdirs>=2
  Downloading platformdirs-2.5.4-py3-none-any.whl (14 kB)
Collecting pathspec>=0.9.0
  Downloading pathspec-0.10.2-py3-none-any.whl (28 kB)
Collecting click>=8.0.0
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
     |████████████████████████████████| 96 kB 3.9 MB/s
Collecting mypy-extensions>=0.4.3
  Downloading mypy_extensions-0.4.3-py2.py3-none-any.whl (4.5 kB)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.8/dist-packages (from jinja2->symforce) (2.0.1)
Installing collected packages: ply, platformdirs, pathspec, mypy-extensions, click, argh, sympy, symforce-sym, skymarshal, clang-format,
  Attempting uninstall: click
    Found existing installation: click 7.1.2
    Uninstalling click-7.1.2:
      Successfully uninstalled click-7.1.2
  Attempting uninstall: sympy
    Found existing installation: sympy 1.7.1
    Uninstalling sympy-1.7.1:
      Successfully uninstalled sympy-1.7.1
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source
flask 1.1.4 requires click<8.0,>=5.1, but you have click 8.1.3 which is incompatible.
Successfully installed argh-0.26.2 black-22.10.0 clang-format-15.0.4 click-8.1.3 mypy-extensions-0.4.3 pathspec-0.10.2 platformdirs-2.5.
```

```
import numpy as np
import os
import symforce

symforce.set_symbolic_api("symengine")
symforce.set_log_level("warning")

# https://symforce.org/tutorials/epsilon_tutorial.html
symforce.set_epsilon_to_symbol()

from symforce import codegen
from symforce.codegen import codegen_util
from symforce import ops
import symforce.symbolic as sf
```

```
from symforce.values import Values
from symforce.notebook_util import display, display_code, display_code_file


def az_el_from_point(
    nav_T_cam: sf.Pose3, nav_t_point: sf.Vector3, epsilon: sf.Scalar = 0
) -> sf.Vector2:
    """
    Transform a nav point into azimuth / elevation angles in the
    camera frame.

    Args:
        nav_T_cam (sf.Pose3): camera pose in the world
        nav_t_point (sf.Matrix): nav point
        epsilon (Scalar): small number to avoid singularities

    Returns:
        sf.Matrix: (azimuth, elevation)
    """
    cam_t_point = nav_T_cam.inverse() * nav_t_point
    x, y, z = cam_t_point
    theta = sf.atan2(y, x + epsilon)
    phi = sf.pi / 2 - sf.acos(z / (cam_t_point.norm() + epsilon))
    return sf.V2(theta, phi)


az_el_codegen = codegen.Codegen.function(
    func=az_el_from_point,
    config=codegen.CppConfig(),
)
az_el_codegen_data = az_el_codegen.generate_function()

print("Files generated in {}:\n".format(az_el_codegen_data.output_dir))
for f in az_el_codegen_data.generated_files:
    print("  |- {}".format(os.path.relpath(f, az_el_codegen_data.output_dir)))

display_code_file(az_el_codegen_data.generated_files[0], "C++")
```

```
Files generated in /tmp/sf_codegen_az_el_from_point_wt9h7hp8:

  |- cpp/symforce/sym/az_el_from_point.h
```

```
// ----------------------------------------------------------------------------
// This file was autogenerated by symforce from template:
//     function/FUNCTION.h.jinja
// Do NOT modify by hand.
// ----------------------------------------------------------------------------

#pragma once

#include <Eigen/Dense>

#include <sym/pose3.h>

namespace sym {

/**
 * Transform a nav point into azimuth / elevation angles in the
 * camera frame.
 *
 * Args:
 *     nav_T_cam (sf.Pose3): camera pose in the world
 *     nav_t_point (sf.Matrix): nav point
 *     epsilon (Scalar): small number to avoid singularities
 *
 * Returns:
 *     sf.Matrix: (azimuth, elevation)
 */
template <typename Scalar>
Eigen::Matrix<Scalar, 2, 1> AzElFromPoint(const sym::Pose3<Scalar>& nav_T_cam,
                                          const Eigen::Matrix<Scalar, 3, 1>& nav_t_point,
                                          const Scalar epsilon) {
  // Total ops: 78

  // Input arrays
  const Eigen::Matrix<Scalar, 7, 1>& _nav_T_cam = nav_T_cam.Data();

  // Intermediate terms (24)
  const Scalar _tmp0 = 2 * _nav_T_cam[0];
  const Scalar _tmp1 = _nav_T_cam[3] * _tmp0;
  const Scalar _tmp2 = 2 * _nav_T_cam[1];
```

```
codegen_with_jacobians = az_el_codegen.with_jacobians(
    which_args=["nav_T_cam", "nav_t_point"],
    include_results=True,
)

data = codegen_with_jacobians.generate_function()
from symforce.notebook_util import display_code_file

display_code_file(data.generated_files[0], "C++")
```

```
// ----------------------------------------------------------------------------
// This file was autogenerated by symforce from template:
//     function/FUNCTION.h.jinja
// Do NOT modify by hand.
// ----------------------------------------------------------------------------

#pragma once

#include <Eigen/Dense>

#include <sym/pose3.h>

namespace sym {

/**
 * Transform a nav point into azimuth / elevation angles in the
 * camera frame.
 *
 * Args:
 *     nav_T_cam (sf.Pose3): camera pose in the world
 *     nav_t_point (sf.Matrix): nav point
 *     epsilon (Scalar): small number to avoid singularities
 *
 * Returns:
 *     sf.Matrix: (azimuth, elevation)
 *     res_D_nav_T_cam: (2x6) jacobian of res (2) wrt arg nav_T_cam (6)
 *     res_D_nav_t_point: (2x3) jacobian of res (2) wrt arg nav_t_point (3)
```

```
L = sf.V2.symbolic("L").T
m = sf.V2.symbolic("m").T
ang = sf.V2.symbolic("a").T
dang = sf.V2.symbolic("da").T
g = sf.Symbol("g")
```

```
// Total ops: 289
```

```
ddang_0 = (
    -g * (2 * m[0] + m[1]) * sf.sin(ang[0])
    - m[1] * g * sf.sin(ang[0] - 2 * ang[1])
    - 2
    * sf.sin(ang[0] - ang[1])
    * m[1]
    * (dang[1] * 2 * L[1] + dang[0] * 2 * L[0] * sf.cos(ang[0] - ang[1]))
) / (L[0] * (2 * m[0] + m[1] - m[1] * sf.cos(2 * ang[0] - 2 * ang[1])))
display(ddang_0)
```

$$\frac{-gm_1 \sin\left(a_0 - 2a_1\right) - g\left(2m_0 + m_1\right)\sin\left(a_0\right) - 2m_1 \cdot \left(2L_0 da_0 \cos\left(a_0 - a_1\right) + 2L_1 da_1\right)\sin\left(a_0 - a_1\right)}{L_0 \cdot \left(2m_0 - m_1 \cos\left(2a_0 - 2a_1\right) + m_1\right)}$$

```
const Scalar _tmp9 = 2 * _nav_t_cam[2];
```

```
ddang_1 = (
    2
    * sf.sin(ang[0] - ang[1])
    * (
        dang[0] ** 2 * L[0] * (m[0] + m[1])
        + g * (m[0] + m[1]) * sf.cos(ang[0])
        + dang[1] ** 2 * L[1] * m[1] * sf.cos(ang[0] - ang[1])
    )
) / (L[1] * (2 * m[0] + m[1] - m[1] * sf.cos(2 * ang[0] - 2 * ang[1])))
display(ddang_1)
```

$$\frac{2\left(L_0 da_0^2\left(m_0 + m_1\right) + L_1 da_1^2 m_1 \cos\left(a_0 - a_1\right) + g\left(m_0 + m_1\right)\cos\left(a_0\right)\right)\sin\left(a_0 - a_1\right)}{L_1 \cdot \left(2m_0 - m_1 \cos\left(2a_0 - 2a_1\right) + m_1\right)}$$

```
const Scalar _tmp24 = _tmp13 + _tmp14;
```

```
inputs = Values()

inputs["ang"] = ang
inputs["dang"] = dang

with inputs.scope("constants"):
    inputs["g"] = g

with inputs.scope("params"):
    inputs["L"] = L
    inputs["m"] = m

display(inputs)
```

```
      Values(
        ang: [a0, a1],
        dang: [da0, da1],
        constants:   Values(
          g: g,
        ),
        params:   Values(
          L: [L0, L1],
          m: [m0, m1],
        ),
      const Scalar _tmp56 = -_tmp26;
```

```
outputs = Values(ddang=sf.V2(ddang_0, ddang_1))

display(outputs)
```

```
      Values(
        ddang: [(-g*(2*m0 + m1)*sin(a0) + g*sin(-a0 + 2*a1)*m1 + 2*sin(-a0 + a1)*m1*(2*L1*da1 + 2*cos(-a0 + a1)*L0*da0))/(L0*
      (2*m0 + m1 - cos(-2*a0 + 2*a1)*m1))]
      [-2*sin(-a0 + a1)*(g*(m0 + m1)*cos(a0) + (m0 + m1)*L0*da0**2 + cos(-a0 + a1)*m1*L1*da1**2)/(L1*(2*m0 + m1 - cos(-2*a0 +
      2*a1)*m1))],
      ),
      const Scalar _tmp60 = _tmp58 + _tmp59;
```

```
double_pendulum = codegen.Codegen(
    inputs=inputs,
    outputs=outputs,
    config=codegen.CppConfig(),
    name="double_pendulum",
    return_key="ddang",
)
double_pendulum_data = double_pendulum.generate_function()

print("Files generated in {}:\n".format(double_pendulum_data.output_dir))
for f in double_pendulum_data.generated_files:
    print("  |- {}".format(os.path.relpath(f, double_pendulum_data.output_dir)))
```

```
     Files generated in /tmp/sf_codegen_double_pendulum_kbz33tag:

       |- lcmtypes/double_pendulum.lcm
       |- cpp/symforce/sym/double_pendulum.h
       const Scalar _tmp76 = _tmp59 + _tmp75;
```

```
display_code_file(double_pendulum_data.function_dir / "double_pendulum.h", "C++")
```

```
// ----------------------------------------------------------------------------
// This file was autogenerated by symforce from template:
//     function/FUNCTION.h.jinja
// Do NOT modify by hand.
// ----------------------------------------------------------------------------

#pragma once

#include <Eigen/Dense>

#include <lcmtypes/sym/constants_t.hpp>
#include <lcmtypes/sym/params_t.hpp>

namespace sym {

/**
 * This function was autogenerated. Do not modify by hand.
 *
 * Args:
 *     ang: Matrix12
 *     dang: Matrix12
 *     constants: Values
 *     params: Values
 *
 * Outputs:
 *     ddang: Matrix21
 */
template <typename Scalar>
```

```
input_values = Values(inputs=inputs)
output_values = Values(outputs=outputs)
namespace = "double_pendulum"
double_pendulum_values = codegen.Codegen(
    inputs=input_values,
    outputs=output_values,
    config=codegen.CppConfig(),
    name="double_pendulum",
)
double_pendulum_values_data = double_pendulum_values.generate_function(
    namespace=namespace,
)

print("Files generated in {}:\n".format(double_pendulum_values_data.output_dir))
for f in double_pendulum_values_data.generated_files:
    print("  |- {}".format(os.path.relpath(f, double_pendulum_values_data.output_dir)))

display_code_file(
    double_pendulum_values_data.function_dir / "double_pendulum.h",
    "C++",
)
```

```
Files generated in /tmp/sf_codegen_double_pendulum_byezx1n3:

  |- lcmtypes/double_pendulum.lcm
  |- cpp/symforce/double_pendulum/double_pendulum.h
// -----------------------------------------------------------------------------
// This file was autogenerated by symforce from template:
//     function/FUNCTION.h.jinja
// Do NOT modify by hand.
// -----------------------------------------------------------------------------

#pragma once

#include <Eigen/Dense>

#include <lcmtypes/double_pendulum/inputs_t.hpp>
#include <lcmtypes/double_pendulum/outputs_t.hpp>

namespace double_pendulum {

/**
 * This function was autogenerated. Do not modify by hand.
 *
 * Args:
 *     inputs: Values
 *
 * Outputs:
 *     outputs: Values
 */
template <typename Scalar>
void DoublePendulum(const double_pendulum::inputs_t& inputs,
                    double_pendulum::outputs_t* const outputs = nullptr) {
  // Total ops: 50

  // Input arrays

  // Intermediate terms (8)
  const Scalar _tmp0 = 2 * inputs.ang.data()[1];
  const Scalar _tmp1 = 2 * inputs.params.m.data()[0] + inputs.params.m.data()[1];
  const Scalar _tmp2 = Scalar(1.0) / (_tmp1 - inputs.params.m.data()[1] *
                                              std::cos( tmp0 - 2 * inputs.ang.data()[0]));
```

```
namespace = "double_pendulum"
double_pendulum_python = codegen.Codegen(
    inputs=inputs,
    outputs=outputs,
    config=codegen.PythonConfig(use_eigen_types=False),
    name="double_pendulum",
    return_key="ddang",
)
double_pendulum_python_data = double_pendulum_python.generate_function(
    namespace=namespace,
)

print("Files generated in {}:\n".format(double_pendulum_python_data.output_dir))
for f in double_pendulum_python_data.generated_files:
    print("  |- {}".format(os.path.relpath(f, double_pendulum_python_data.output_dir)))

display_code_file(
    double_pendulum_python_data.function_dir / "double_pendulum.py",
    "python",
)
```

```
Files generated in /tmp/sf_codegen_double_pendulum_5zfc3z_f:

  |- lcmtypes/double_pendulum.lcm
  |- python/symforce/double_pendulum/double_pendulum.py
  |- python/symforce/double_pendulum/__init__.py
# -----------------------------------------------------------------------------
# This file was autogenerated by symforce from template:
#     function/FUNCTION.py.jinja
# Do NOT modify by hand.
# -----------------------------------------------------------------------------

# pylint: disable=too-many-locals,too-many-lines,too-many-statements,unused-argument,unu

import math
import typing as T

import numpy

import sym


def double_pendulum(ang, dang, constants, params):
    # type: (numpy.ndarray, numpy.ndarray, T.Any, T.Any) -> numpy.ndarray
    """
    This function was autogenerated. Do not modify by hand.

    Args:
        ang: Matrix12
        dang: Matrix12
        constants: Values
        params: Values

    Outputs:
        ddang: Matrix21
    """

    # Total ops: 50

    # Input arrays
    if ang.shape == (2,):
        ang = ang.reshape((1, 2))
    elif ang.shape != (1, 2):
        raise IndexError(
            "ang is expected to have shape (1, 2) or (2,); instead had shape {}".format(
        )

    if dang.shape == (2,):
        dang = dang.reshape((1, 2))
    elif dang.shape != (1, 2):
        raise IndexError(
            "dang is expected to have shape (1, 2) or (2,); instead had shape {}".format(
        )

    # Intermediate terms (8)
    _tmp0 = 2 * ang[0, 1]
    _tmp1 = 2 * params.m[0] + params.m[1]
    _tmp2 = 1 / (_tmp1 - params.m[1] * math.cos(_tmp0 - 2 * ang[0, 0]))
    _tmp3 = -ang[0, 0]
    _tmp4 = _tmp3 + ang[0, 1]
    _tmp5 = math.cos(_tmp4)
    _tmp6 = 2 * math.sin(_tmp4)
    _tmp7 = params.m[0] + params.m[1]

    # Output terms
    _ddang = numpy.zeros((2, 1))
    _ddang[0, 0] = (
        _tmp2
```

```
constants_t = codegen_util.load_generated_lcmtype(
    namespace, "constants_t", double_pendulum_python_data.python_types_dir
)

params_t = codegen_util.load_generated_lcmtype(
    namespace, "params_t", double_pendulum_python_data.python_types_dir
)

ang = np.array([[0.0, 0.5]])
dang = np.array([[0.0, 0.0]])
consts = constants_t()
consts.g = 9.81
params = params_t()
params.L = [0.5, 0.3]
```

```
params.m = [0.3, 0.2]

gen_module = codegen_util.load_generated_package(
    namespace, double_pendulum_python_data.function_dir
)
gen_module.double_pendulum(ang, dang, consts, params)
```

```
array([[  4.77199518],
       [-22.65691471]])
```