Name: Cuong Ha; Matriculation Number: 03749410

Part 1: CMake

• set(CMAKE_MODULE_PATH "\${CMAKE_CURRENT_SOURCE_DIR}/cmake_modules/" \${CMAKE_MODULE_PATH})

// specify a search path for CMake modules to be loaded by the the include() or find_package()

set(CMAKE_CXX_STANDARD 14)set(CMAKE_CXX_STANDARD_REQUIRED ON)

set(CMAKE CXX EXTENSIONS OFF)

// Specify that C++ 14 standard is required, disable C++ extensions (e.g. use e.g. -std=c++11 rather than -std=gnu++11) for broadest compatibility across compilers

• set(CMAKE_CXX_FLAGS_DEBUG "-00 -g -DEIGEN_INITIALIZE_MATRICES_BY_NAN")

// set compile flag for building in Debug mode: optimize level 0 and init Eigen matrix by
Nan to track for the problem of uninitialized data being used.

set(CMAKE_CXX_FLAGS_RELWITHDEBINFO "-O3 -DNDEBUG -g - DEIGEN_INITIALIZE_MATRICES_BY_NAN")

// set compile flag for building in Release with Debug information mode: Optimization level 3, disable assert, and debug info a.k.a additional symbol table information for debugging and tracking for uninitialized Eigen matrix.

set(CMAKE_CXX_FLAGS_RELEASE "-O3 -DNDEBUG")

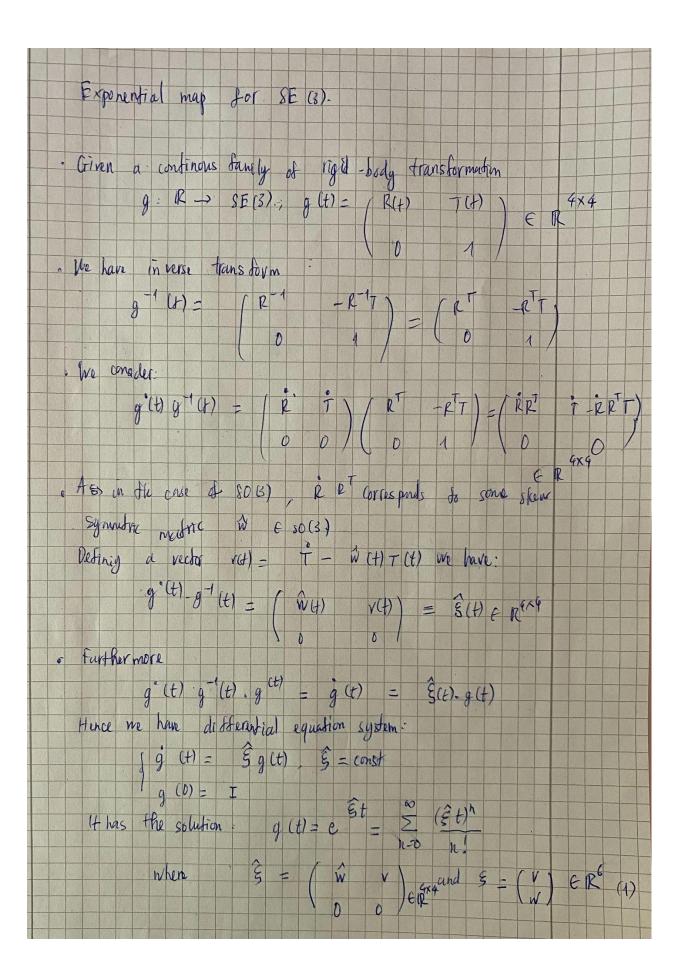
// set compile flag for building in Release mode: Optimization level 3, disable assert.

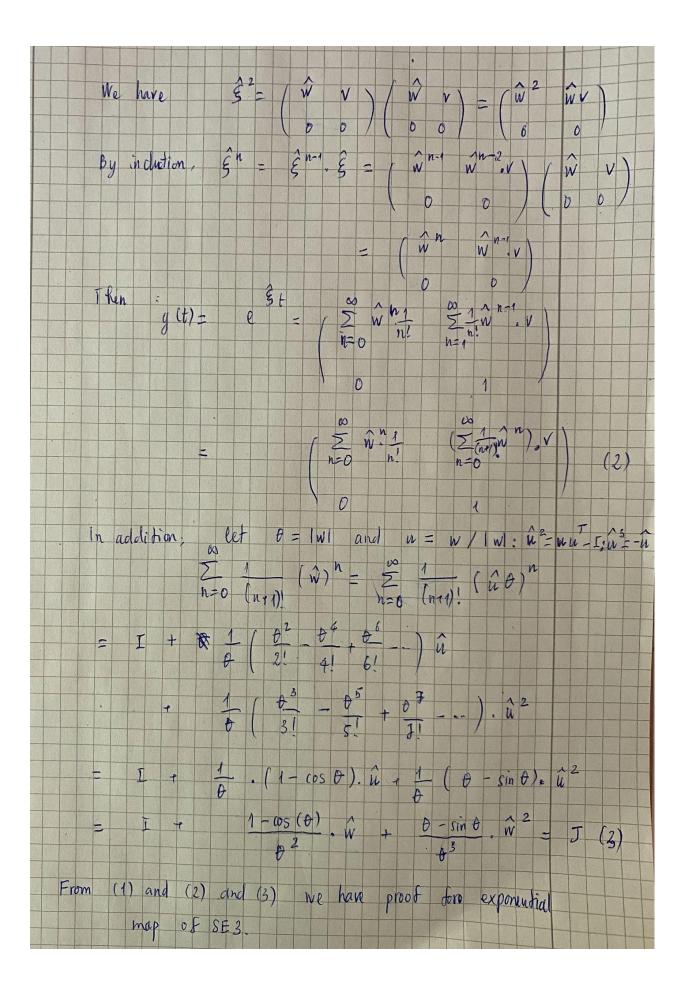
SET(CMAKE_CXX_FLAGS " -ftemplate -backtrace -limit=0 -Wall -Wextra \${ EXTRA_WARNING_FLAGS} -march=\${CXX_MARCH} \${CMAKE_CXX_FLAGS}") // Default compile flag

add_executable(calibration src/calibration.cpp)

target link libraries(calibration Ceres::ceres pangolin TBB)

// Create executable file "calibration" from source file and link it with libraries: Ceres, pangolin and TBB.





Part 3: What is SLAM?

From Cesar Cadena et al. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age".

1. Why would a SLAM system need a map?

A SLAM system requires the use of a map for two reasons:

- To support other tasks such as informing path planning or providing intuitive visualization for a human operator.
- Limit the error committed in estimating the state of the robot. Without a map, dead-reckoning would quickly drift over time; on the other hand, using a map, e.g., a set of distinguishable landmarks, the robot can "reset" its localization error by re-visiting known areas (so-called loop closure).
- 2. How can we apply SLAM technology to real-world applications?
 - SLAM finds applications in all scenarios in which a prior map is not available and needs to be built.
 - It raised in popularity in indoor applications of mobile robotics, where GPS is inaccurate.
 - Furthermore, SLAM provides an appealing alternative to user-built maps, showing that robot operation is possible in the absence of an ad hoc localization infrastructure.
- 3. Describe the history of SLAM.

History of SLAM can be splitted into two ages:

- Classical age: 1986-2004
 - introduction of the main probabilistic formulations for SLAM, including approaches based on Extended Kalman Filters, Rao-Blackwellised Particle Filters, and maximum likelihood estimation.
 - delineated the basic challenges connected to efficiency and robust data association.
- Algorithmic analysis age: 2004-2015
 - study of fundamental properties of SLAM, including observability, convergence, and consistency.
 - the key role of sparsity towards efficient SLAM solvers was also understood, and the main open-source SLAM libraries were developed.