cmake\_minimum\_required(VERSION 3.0.2)

project(sensor)

## Compile as C++11, supported in ROS Kinetic and newer

add\_compile\_options(-std=c++11)

## Find catkin macros and libraries

## if COMPONENTS list like find\_package(catkin REQUIRED COMPONENTS xyz)

## is used, also find other catkin packages

find\_package(catkin REQUIRED COMPONENTS

roscpp

rospy

std\_msgs

wittenstein\_msgs

rosbag

ourSensor\_msgs

)

# Generate messages in the 'msg' folder

# add\_message\_files(

# FILES

# )

## Generate services in the 'srv' folder

# add\_service\_files(

# FILES

# Service1.srv

# Service2.srv

# )

## Generate actions in the 'action' folder

# add\_action\_files(

# FILES

# Action1.action

# Action2.action

# )

## Generate added messages and services with any dependencies listed here

# generate\_messages(

# DEPENDENCIES

# std\_msgs

# )

## Generate dynamic reconfigure parameters in the 'cfg' folder

# generate\_dynamic\_reconfigure\_options(

# cfg/DynReconf1.cfg

# cfg/DynReconf2.cfg

# )

catkin\_package(

#INCLUDE\_DIRS include

LIBRARIES ${PROJECT\_NAME}

CATKIN\_DEPENDS roscpp rospy std\_msgs wittenstein\_msgs ourSensor\_msgs

#DEPENDS system\_lib

)

catkin\_install\_python(PROGRAMS src/accel\_pub.py

DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)

###########

## Build ##

###########

## Specify additional locations of header files

## Your package locations should be listed before other locations

include\_directories(

${catkin\_INCLUDE\_DIRS}

include

)

#add\_executable(wittenstein\_main src/main.cpp src/HexFT.cpp)

#target\_link\_libraries(wittenstein\_main ${catkin\_LIBRARIES})

# OUTDATED

#add\_executable(active\_data src/data\_call.cpp src/HexFT.cpp)

#add\_dependencies(active\_data ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

#target\_link\_libraries(active\_data

# ${catkin\_LIBRARIES}

#)

add\_executable(get\_force src/get\_force.cpp src/HexFT.cpp)

add\_dependencies(get\_force ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(get\_force

${catkin\_LIBRARIES}

)

add\_executable(force\_control src/force\_control.cpp src/HexFT.cpp)

add\_dependencies(force\_control ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(force\_control

${catkin\_LIBRARIES}

)

# OUTDATED

#add\_executable(data\_force src/data\_force.cpp src/HexFT.cpp)

#add\_dependencies(data\_force ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

#target\_link\_libraries(data\_force

# ${catkin\_LIBRARIES}

#)

add\_executable(active\_position src/data\_call\_position\_control.cpp src/HexFT.cpp)

add\_dependencies(active\_position ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(active\_position

${catkin\_LIBRARIES}

)

add\_executable(impulse\_coll src/impulse\_coll.cpp src/HexFT.cpp)

add\_dependencies(impulse\_coll ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(impulse\_coll

${catkin\_LIBRARIES}

)

add\_executable(force\_pub src/force\_pub.cpp src/HexFT.cpp)

add\_dependencies(force\_pub ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(force\_pub

${catkin\_LIBRARIES}

)

add\_executable(my\_test src/test.cpp src/HexFT.cpp)

add\_dependencies(my\_test ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(my\_test

${catkin\_LIBRARIES}

)

add\_executable(accel\_sub src/accel\_sub.cpp)

add\_dependencies(accel\_sub ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(accel\_sub

${catkin\_LIBRARIES}

)

add\_executable(hall\_calibrate src/hall\_calibrate.cpp)

add\_dependencies(hall\_calibrate ${${PROJECT\_NAME}\_EXPORTED\_TARGETS} ${catkin\_EXPORTED\_TARGETS})

target\_link\_libraries(hall\_calibrate

${catkin\_LIBRARIES}

)