cmake\_minimum\_required(VERSION 3.0.2)

project(filters)

##############################################################################

# Find dependencies

##############################################################################

find\_package(catkin REQUIRED COMPONENTS pluginlib roslib roscpp rosconsole)

find\_package(Boost REQUIRED)

include\_directories(

include

${catkin\_INCLUDE\_DIRS}

)

##############################################################################

# Define package

##############################################################################

catkin\_package(

INCLUDE\_DIRS include

LIBRARIES mean params increment median transfer\_function

CATKIN\_DEPENDS roslib roscpp rosconsole pluginlib

DEPENDS Boost

)

##############################################################################

# Build

##############################################################################

add\_library(real\_time\_circular\_buffer INTERFACE)

target\_include\_directories(real\_time\_circular\_buffer INTERFACE

include

${Boost\_INCLUDE\_DIRS})

# Plugins

add\_library(mean src/mean.cpp)

target\_link\_libraries(mean

${catkin\_LIBRARIES}

real\_time\_circular\_buffer)

add\_library(params src/test\_params.cpp)

target\_link\_libraries(params

${catkin\_LIBRARIES}

real\_time\_circular\_buffer)

add\_library(increment src/increment.cpp)

target\_link\_libraries(increment

${catkin\_LIBRARIES}

real\_time\_circular\_buffer)

add\_library(median src/median.cpp)

target\_link\_libraries(median

${catkin\_LIBRARIES}

real\_time\_circular\_buffer)

add\_library(transfer\_function src/transfer\_function.cpp)

target\_link\_libraries(transfer\_function

${catkin\_LIBRARIES}

real\_time\_circular\_buffer)

if(CATKIN\_ENABLE\_TESTING)

find\_package(rostest)

# Test median filter

add\_executable(median\_test EXCLUDE\_FROM\_ALL test/test\_median.cpp )

target\_link\_libraries(median\_test median ${catkin\_LIBRARIES} ${GTEST\_LIBRARIES})

add\_rostest(test/test\_median.launch)

# Test transfer function filter

add\_executable(transfer\_function\_test EXCLUDE\_FROM\_ALL test/test\_transfer\_function.cpp)

target\_link\_libraries(transfer\_function\_test transfer\_function ${catkin\_LIBRARIES} ${GTEST\_LIBRARIES})

add\_rostest(test/test\_transfer\_function.launch)

# Test mean filter

add\_executable(mean\_test EXCLUDE\_FROM\_ALL test/test\_mean.cpp)

target\_link\_libraries(mean\_test mean ${catkin\_LIBRARIES} ${GTEST\_LIBRARIES})

add\_rostest(test/test\_mean.launch)

# Test params filter

add\_executable(params\_test EXCLUDE\_FROM\_ALL test/test\_params.cpp)

target\_link\_libraries(params\_test params ${catkin\_LIBRARIES} ${GTEST\_LIBRARIES})

add\_rostest(test/test\_params.launch)

# Test plugin loading into filter chain

add\_executable(chain\_test EXCLUDE\_FROM\_ALL test/test\_chain.cpp)

target\_link\_libraries(chain\_test increment ${catkin\_LIBRARIES} ${GTEST\_LIBRARIES}) # Needed for OSX

add\_rostest(test/test\_chain.launch)

# Test realtime safe buffer class

catkin\_add\_gtest(realtime\_buffer\_test EXCLUDE\_FROM\_ALL test/test\_realtime\_circular\_buffer.cpp)

target\_link\_libraries(realtime\_buffer\_test real\_time\_circular\_buffer)

endif()

##############################################################################

# Install

##############################################################################

# Install libraries

install(TARGETS mean params increment median transfer\_function

ARCHIVE DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

LIBRARY DESTINATION ${CATKIN\_PACKAGE\_LIB\_DESTINATION}

RUNTIME DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)

# Install headers

install(DIRECTORY include/${PROJECT\_NAME}/

DESTINATION ${CATKIN\_PACKAGE\_INCLUDE\_DESTINATION})

# Install plugins xml file

install(FILES default\_plugins.xml

DESTINATION ${CATKIN\_PACKAGE\_SHARE\_DESTINATION}

)

if(CATKIN\_ENABLE\_TESTING)

if(TARGET tests)

add\_dependencies(tests median\_test transfer\_function\_test mean\_test params\_test chain\_test)

endif()

endif()