STOP TRAJECTORY

namespace SpecialSeqValues

{

enum SpecialSeqValue

{

END\_TRAJECTORY = -1, STOP\_TRAJECTORY = -2

};

}

void JointTrajectoryInterface::trajectoryStop()

{

JointTrajPtMessage jMsg;

SimpleMessage msg, reply;

ROS\_INFO("Joint trajectory handler: entering stopping state");

jMsg.setSequence(SpecialSeqValues::STOP\_TRAJECTORY);

jMsg.toRequest(msg);

ROS\_DEBUG("Sending stop command");

this->connection\_->sendAndReceiveMsg(msg, reply);

}

STOP MOTION

bool MotomanMotionCtrl::stopTrajectory()

{

MotionReply reply;

if (!sendAndReceive(MotionControlCmds::STOP\_MOTION, reply))

{

ROS\_ERROR("Failed to send STOP\_MOTION command");

return false;

}

if (reply.getResult() != MotionReplyResults::SUCCESS)

{

ROS\_ERROR\_STREAM("Failed to Stop Motion: " << getErrorString(reply));

return false;

}

return true;

}