

Report

There are six “sub-goals”:

Going ahead for 3 meters:

```
pose.position.x = 3.0; // say desired x-coord is 3
pose.position.y = 0.0;
pose.position.z = 0.0; // let's hope so!
pose.orientation.x = 0.0; //always, for motion in horizontal plane
pose.orientation.y = 0.0; // ditto
pose.orientation.z = 0.0; // implies oriented at yaw=0, i.e. along x axis
pose.orientation.w = 1.0; //sum of squares of all components of unit quaternion is 1
pose_stamped.pose = pose;
path_srv.request.nav_path.poses.push_back(pose_stamped); //2D vector
```

Turn left about 45 degrees:

```
quat = convertPlanarPhi2Quaternion(0.80); // get a quaterion corresponding to this heading
pose_stamped.pose.orientation = quat;
pose_stamped.pose.position.y=5.0; // say desired y-coord is 5.0
pose_stamped.pose.position.x = 7.9; // say desired y-coord is 4.9
path_srv.request.nav_path.poses.push_back(pose_stamped);
```

Spin left about 130 degrees:

```
quat = convertPlanarPhi2Quaternion(3.08);
pose_stamped.pose.orientation = quat;
//desired position is not updated...just the desired heading
pose_stamped.pose.position.y=5.0; //
pose_stamped.pose.position.x=7.9; //
path_srv.request.nav_path.poses.push_back(pose_stamped);
```

Adjust heading and go horizontally:

```
quat = convertPlanarPhi2Quaternion(3.09);
pose_stamped.pose.orientation = quat;
//desired position is not updated...just the desired heading
pose_stamped.pose.position.y=5.4; //
pose_stamped.pose.position.x=1.0; //
path_srv.request.nav_path.poses.push_back(pose_stamped);
```

Turn right:

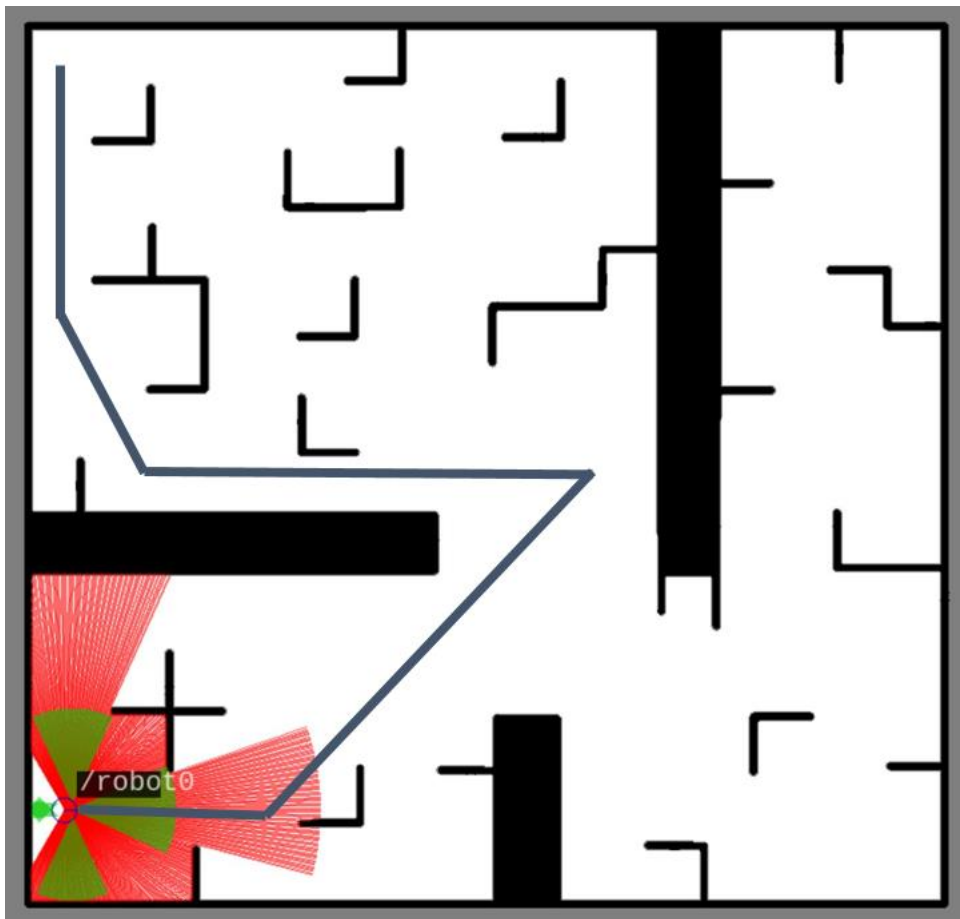
```
quat = convertPlanarPhi2Quaternion(1.70);
pose_stamped.pose.orientation = quat;
```

```
//desired position is not updated...just the desired heading
pose_stamped.pose.position.y=6.5; //
pose_stamped.pose.position.x=0.3; //
path_srv.request.nav_path.poses.push_back(pose_stamped);
```

Go vertically:

```
quat = convertPlanarPhi2Quaternion(1.61);
pose_stamped.pose.orientation = quat;
//desired position is not updated...just the desired heading
pose_stamped.pose.position.y=21.0; //
pose_stamped.pose.position.x=-0.3; //
path_srv.request.nav_path.poses.push_back(pose_stamped);
```

Which will set the trajectory for the robot as following:



The moving distance and heading for each pose are computed as:

```
void get_yaw_and_dist(geometry_msgs::Pose current_pose, geometry_msgs::Pose
goal_pose, double &dist, double &heading) {
    double dx = 0.0;
    double dy = 0.0;
    // dist = 0.0; //FALSE!!
```

```

dx = goal_pose.position.x - current_pose.position.x;
dy = goal_pose.position.y - current_pose.position.y;
dist = sqrt((dx*dx) + (dy*dy));
///heading = convertPlanarQuat2Phi(goal_pose.orientation);

if (dist < g_dist_tol) { //too small of a motion, so just set the heading from goal heading
    heading = convertPlanarQuat2Phi(goal_pose.orientation);
}
else {
    // heading = 0.0; //FALSE!!
    heading = atan2(dy,dx);
}
}

```

Although I guess it will be better to give the more freedom to separate the moving distance and the bearing since it is open loop, which actually will be much more convenient for some poses as long as it's heading to the right spot.