

Nonlinear MPC based Shared Control

Martijn Krijnen
RRIS
NTU

28/02/2020

1 Introduction

This document describes the implementation of Shared Control based on Nonlinear Model Predictive Control (NMPC). The purpose is to build a shared control module for a wheelchair, for patients with limited motory control, perception or cognitive abilities. The method design was partially inspired by [1].

2 Method

The typical MPC definition is as follows: optimize a certain costfunction J , for a number of states x and control variables u over a prediction horizon T , subject to equality and/or inequality constraints:

$$\begin{aligned} \min. \quad & \sum_{t=0}^T J(x, u) \\ \text{s.t.} \quad & \dot{x} = f(x) \\ & c_{ineq} < 0 \\ & c_{eq} = 0 \\ & u \in U \\ & x \in X \end{aligned}$$

NMPC can be used to do local planning, by using the robot's coordinates for the states x and motor inputs as the input variables u . To use this for shared control, the cost function J can be constructed from the following main components:

- Obedience: Follow user input commands
- Comfort: Use a smooth path and smooth acceleration / deceleration
- Safety: Avoid obstacles.

2.1 Model

The wheelchair is modelled as a differential drive robot. The MPC states are the coordinates of the wheelchair: x, y, θ . The input variables are the linear and angular velocities: v, ω . For a certain trajectory, the next state along the trajectory can be calculated from the current state and input variables as follows:

$$x_{t+1} = x_t + v_t * \cos(\theta) \Delta t$$

$$y_{t+1} = y_t + v_t * \sin(\theta) \Delta t$$

$$\theta_{t+1} = \theta + \omega * \Delta t$$

2.2 Constraints

The system model is recast into a list of equality constraints, that can be defined as: For each point along the trajectory the model error should be approximately zero. Mathematically:

$$c_{eq,x}(t) = x_{t+1} - x_t + v_t * \cos(\theta) \Delta t = 0$$

$$c_{eq,y}(t) = y_{t+1} - y_t + v_t * \sin(\theta) \Delta t = 0$$

$$c_{eq,\theta}(t) = \theta_{t+1} - \theta + \omega * \Delta t = 0$$

for $t = 0$ to $t = (T - 1)$.

Inequality constraints are added for obstacle avoidance. The obstacle extraction method models obstacles as circles and ellipses. The added obstacle constraints to the optimization are defined as follows:

$$c_{ineq,circle} = d_{threshold} - (d_{center} - r_{circle}) < 0$$

$$c_{ineq,ellipse} = 1 - \left(\frac{x_e^2}{a^2} + \frac{y_e^2}{b^2} \right) < 0$$

Where a and b are ellipse dimensions and x_e and y_e are robot positions in the ellipse frame of reference. The ellipse inequality constraint is derived from one of the standard algebraic ellipse definitions.

To improve computation time, obstacles are only taken into account if they are within a specified range of the robot.

2.3 Costfunction

The cost function is composed as follows:

$$J_t = C_v * ||v_{user} - v_t||^2 + C_h * ||\theta_{user} - \theta_t||^2 + C_c * (||v_{t+1} - v_t||^2 + ||\omega_{t+1} - \omega_t||^2)$$

Here the first term (C_v) corresponds to 'obedience to user velocity', i.e. it minimizes the error of the velocity along the trajectory to the instantaneous user input velocity. The second term (C_h) corresponds to 'obedience to user heading', i.e. it minimizes the error of the trajectory heading to the current user input. The heading (orientation) variable θ is approximated as $\theta = \text{atan}(\frac{\omega}{v})$. This term should be improved in future versions, since in its current form it prefers circular trajectories, essentially converging to DWA like behavior. The third term (C_c) corresponds to comfort, i.e. it penalizes the change between subsequent commands, enforcing smooth actuation along the trajectory.

2.4 Comparison of methods

As mentioned in the introduction the method of 'Intention-free Shared MPC' is somewhat similar to the Dynamic Window Approach (DWA). DWA samples the input space v, ω and predicts the immediate path based on a fixed set of inputs. For the differential drive model this leads to circular trajectories. MPC is different a.o. because it allows the input variables to vary, enabling more complex trajectories.

An alternative method to guarantee user safety would be a so-called 'command safety filter'. When the safety filter receives command velocities (input variables), it checks whether there is an obstacle within a certain distance, in the commanded direction. The command velocities are only executed if there is no obstacle in that direction. The obvious limitation of this safety filter is that it doesn't help the user to find a way around the obstacle.

2.5 Solver

CPPAD and IPOPT are currently used as Solver for the NMPC problem. CPPAD stands for C++ Algorithmic Differentiation, and is a toolbox which can be used to efficiently calculate the derivative of the problem. IPOPT stands for Interior Point Optimizati~~on~~n, and uses these derivatives are then used for gradient-based nonlinear optimization. IPOPT requires the NMPC problem to be cast as a separate class, often called FGeval. This class contains an operator() function used to evaluate to cost (F) and constraints (G). For more information, see [2].

3 Results

TBD

References

- [1] Pablo Inigo-Blasco, "The Shared Control Dynamic Window Approach for Non-Holonomic Semi-Autonomous Robots" in *ISR/Robotik 2014; 41st International Symposium on Robotics, 2-3 June 2014*
- [2] <https://projects.coin-or.org/CppAD>