Integrated Model predictive control and Moving Horizon estimation for Stabilization at Goal location

Manasvi Saxena

Mercedes Benz R&D India Pvt. Ltd., Bangalore, India

saxena.manasvi06@gmail.com

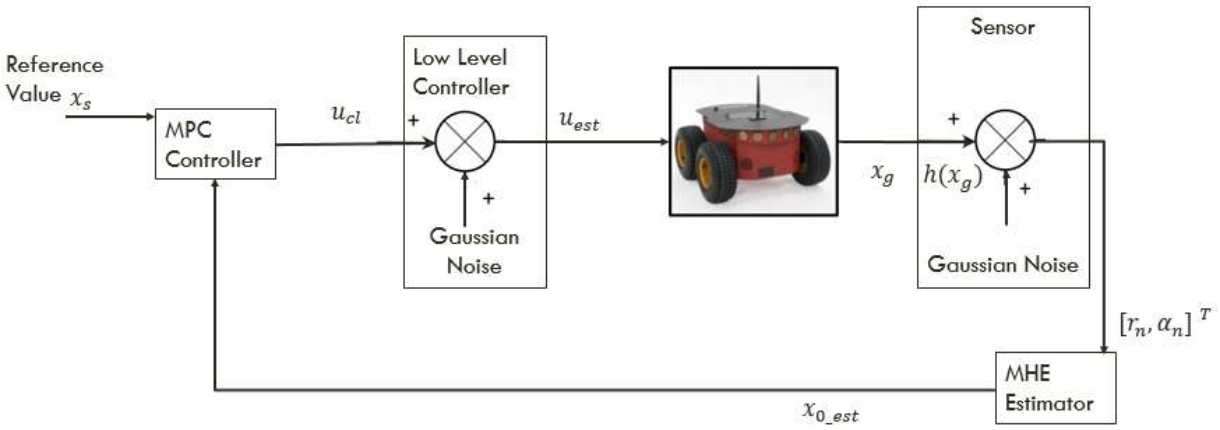
INTRODUCTION

In this paper, I demonstrate how to "Integrate Model predictive control and Moving Horizon estimation for Mobile robots" to achieve the control objective of Stabilization at Goal location.

My codes are further advanced implementation of the codes already explained in the workshop of M.W. Mehrez [1] because of following two reasons 1. I consider that both the estimates of current states and the applied control inputs are noisy which were not considered together in the examples of the workshop 2. I integrate together the Estimation and the Control problem to work simultaneously which were addressed as two separate problems in the workshop. In fact it was also recommended in the workshop that "Combining MHE with MPC. A very good Exercise!" as the next step. I am grateful to M.W. Mehrez's workshop which helped me to get started in the domain of solving Nonlinear Model predictive control using CASADI.

PROBLEM FORMULATION for Stabilization at Goal Location:

We have the below scenario for our control objective i.e. Stabilization at the Goal location defined as x_s. The current position of the robot is not known and is measured by a range-bearing sensor which measures the ground truth states of the robot but has an unknown noise associated with it (assumed Gaussian here). The best estimate of the current position is determined by the MHE estimator which is then passed to MPC and compared with the desired goal location for the MPC controller to determine the control trajectory for the entire pre-defined prediction horizon length. The first control action of the trajectory is applied to the low level controller of the robot which applies the higher level control action given to it by MPC but with an unknown noise associated with it (assumed Gaussian here). Therefore, the control input which ultimately goes to the robot is unknown, the effect of which can only be quantified by measuring states of the robot by range-bearing sensor which itself has noise associated with it. The goal is to make the robot reach the goal location under these Sensor and Actuator noise. The basic flowchart is given below for our problem. Next, important aspects of each of the Key blocks are explained and the *line no.* where they are implemented in the code MPC_MHE_Robot_PS_mul_shooting_vc1_stabilisation.m and the associated functions is also highlighted.



1. MPC Controller:

We use Multiple shooting to convert MPC problem in to NLP which will be solved by CASADI. In Multiple shooting both the System states and the control inputs for the entire prediction horizon are defined as Optimization variables as defined in *line no. 44* in the code. The system model as shown below is first defined as a function f in *line no. 20* and then used to define equality constraints between states and control inputs at the two consecutive time steps as shown below in the eq. (1) and implemented in *line no. 41*. The initial condition x_0 is also defined as a constraint in *line no. 34*.

$$\begin{bmatrix} x(k+1) \\ y(k+1) \\ \theta(k+1) \end{bmatrix} = \begin{bmatrix} x(k) \\ y(k) \\ \theta(k) \end{bmatrix} + \Delta T \begin{bmatrix} v(k)\cos(\theta(k)) \\ v(k)\sin(\theta(k)) \\ \omega(k) \end{bmatrix} \tag{1}$$

The objective function for MPC controller is to reach the goal location defined by x_s. Therefore objective function is defined as below in eq. (2) where the quadratic error between the states in the entire prediction horizon length w.r.t goal location is minimized. Since the objective is to reach and stabilize at the goal location, the Quadratic error of the control action w.r.t zero ($u_s = 0$) is also included in the objective function and is implemented in *line no. 37*. The lower and upper bounds of state and control inputs are implemented in *line no. 61-75*.

$$\min_{x,u} J_N(x,u) = \sum_{k=0}^{N-1} \|x(k) - x_s\|_Q^2 + \|u(k) - u_s\|_R^2 \tag{2}$$

$$s.t.\ x(k+1) = f\big(x(k), u(k)\big)$$

$$x(0) = x_0$$

$$x(k) \in X\ \ \forall\, k \in [0, N]$$

$$u(k) \in U\ \ \forall\, k \in [0, N-1]$$

2. MHE Estimator:

We use Multiple shooting also to convert MHE problem in to NLP which will be solved by CASADI. In Multiple shooting both the System states and the control inputs for the entire prediction horizon are defined as Optimization variables as defined in *line no. 119* in the code. The system model as shown above is first defined as a function f in *line no. 20* and then used to define equality constraints between states and control inputs at the two consecutive time steps as shown above in the eq. (1) and implemented in *line no. 114* for MHE.

The objective for MHE estimator as defined below in eq. (5) is to estimate the current state of the robot which might not move as predicted by the motion model and MPC control inputs because of the unknown disturbances present in the Low level controller. Range and bearing sensor is used to measure states of the robot but also has noise associated with it. Therefore the objective function is to estimate the states of the robot by minimizing the quadratic error between the range and bearing measurements and the predicted measurements from the sensor model defined below in eq. (3) and implemented in *line no. 81*.

$$y(k) = h(x(k)) \tag{3}$$

$$\begin{bmatrix} r \\ \alpha \end{bmatrix} = \begin{bmatrix} \sqrt[2]{x^2 + y^2} \\ arctan(\frac{y}{x}) \end{bmatrix} \tag{4}$$

The quadratic error between the range and bearing measurements and the predicted measurements from the sensor model is implemented in *line no. 99*. The objective of MHE estimator may also be to estimate the control inputs that Low level controller is applying to the robot. Therefore we can add quadratic error between the nominal control inputs calculated by MPC and the predicted control outputs from the Low level controller as an additional objective which is implemented in *line no. 105*. The lower and upper bounds of state and control inputs are implemented in *line no. 128-141*.

$$\underset{x_{est}, u_{est}}{min} J_N(x,u) = \sum_{i=k-N}^{i} \|\tilde{y}(i) - h(x_{est}(i))\|_Q^2 + \|u_{est}(i) - u_{cl}\|_R^2 \tag{5}$$

$$s.t.\ x_{est}(k+1) = f\big(x_{est}(k), u_{est}(k)\big)$$

$$x_{est}(k) \in X \ \ \forall\, k \in [k-N, k]$$

$$u_{est}(k) \in U \ \ \forall\, k \in [k-N, k-1]$$

3. Simulation Loop:

Once the MPC & MHE problem setup is done we can start with our simulation loop demonstrating the effectiveness of our formulation in making the robot reach the goal location under the presence of Sensor and Actuator uncertainties.

Starting from *line no. 181*, given that the ground truth is known at the initial condition and also the desired goal location, MPC controller calculates the control and possible state trajectory for the entire prediction horizon length in *line no. 184* such that the trajectory minimizes the given objective function satisfying the given initial condition, system model equality constraints, other defined bounds. The first control action is used in *line no. 188* and passed to the Low Level controller which is modeled inside shift_meas_control_noise.m function at *line no. 191*. Inside this function some unknown Gaussian noise is added to the control action in *line no. 8*. This disturbed control action is applied to the robot which is also modeled in the same function in *line no. 11-14*. The ground truth states of the robot is unknown but is still stored in *line no. 20* for evaluating our method later. The only way to measures the ground truth states is by a range-bearing sensor which has an unknown noise associated with it (assumed Gaussian here) and is modeled in *line no. 25-26* of the same function and is returned back to the simulation loop where we will use now the MHE to estimate the complete state vector of the robot.

Please note that MHE estimator needs minimum no of measurements equal to the define horizon length to start estimating for the next step. Therefore in this simulation scenario we assume that for the first N steps the ground truth is known. This is what we implement in *line no. 196-203* where we store the range-bearing measurements without noise in array y_measurements, initialize the trajectory from the MPC solution in *line no. 200-201* and store the current state as per the ground truth for the initialization of next iteration of MPC. Once we have collected the measurements for the defined horizon length, We can execute the first MHE iteration with decision variables initialized as per the values stored in y_measurements without noise and the executed control law like I did in *line no. 209-213*. After the first iteration, we use the solution of the MHE for the initialization of the next MHE iteration like I did in *line no. 215-217*. Going forward, always the last N values stored in the y_measurements and the executed control are used as inputs as in *line no. 220* and the current robot state is estimated from the solution of MHE. These estimated states are used by the MPC controller as in *line no. 227* as the new initial position of the robot for the next MPC iteration and also stored as the best knowledge of the current position of the robot which can be later compared with the ground truth.

4. Results

Fig. 1 (a) shows the effect of noise on the Senor measurements as compared to if the sensor would have been perfect. Fig. 1 (b) shows the ground truth vs estimated states which shows that the MHE estimator was able to estimate all the states very

well even though the last state θ is theoretically unobservable. For the animation of the Exhibited trajectory with the Sensor noise and the ground truth in action please see the associated github page.

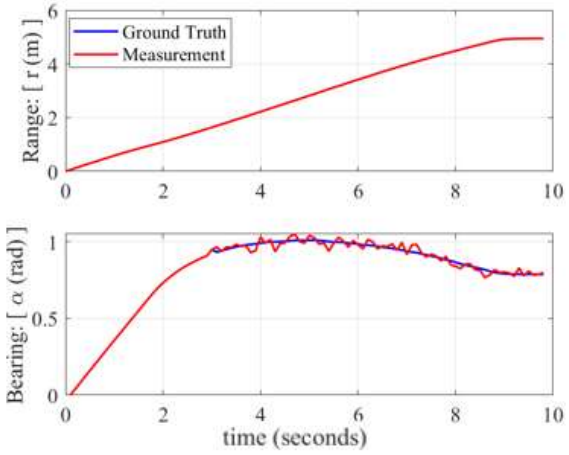


Fig. 1 (a)

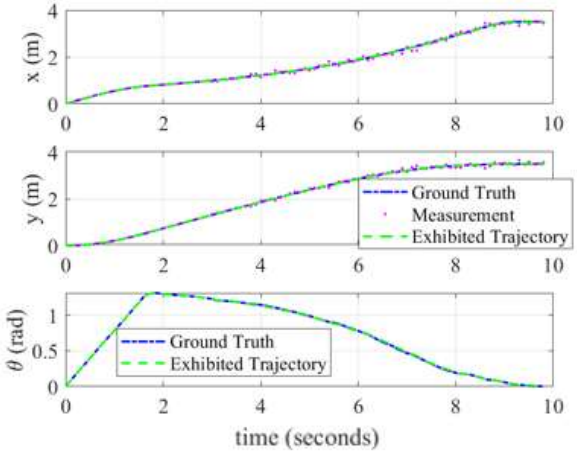


Fig. 1 (b)

REFERENCES

1. M.W. Mehrez Optimization based solutions for control and State estimation in Dynamical systems (Implementation to Mobile Robots): A Workshop

   https://github.com/MMehrez/MPC-and-MHE-implementation-in-MATLAB-using-Casadi