23’ Optimal Control Final Term Project

Submission Deadline: December 18, 2023

Required Submission Materials:

- Result Report
- Matlab Code (must be operational)

Instruction

The model presented below represents a lateral dynamic motion and position model of a vehicle, assuming a small wheel angle. This model is intended for use in designing controllers for the given problems Q1 and Q2. It takes the vehicle's steering wheel as input and models global lateral position, lateral velocity, heading, and yaw rate as its states. It is assumed that all states are observable and that the vehicle maintains a constant longitudinal velocity. It is important to emphasize that this model is specifically for the design of controllers and differs from the Plant model provided in the matlab simulation program. Please keep this information in mind while addressing the problems outlined below.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}\mathbf{u}$$
$$\mathbf{y} = \mathbf{c}\mathbf{x}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & V_x & 0 \\ 0 & -\frac{2C_{af}+2C_{ar}}{mv_x} & 0 & -V_x - \frac{2C_{af}L_f-2C_{ar}L_r}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f-2C_{\alpha r}l_r}{I_z v_x} & 0 & -\frac{2C_{\alpha f}L_f^2+2C_{\alpha r}L_r^2}{I_z v_x} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}L_f}{I_z} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} Y & v_y & \psi & \dot{\psi} \end{bmatrix}^T \quad \mathbf{u} = \delta$$

Q1) MPC-Regulator

This project focuses on the design of a lateral controller that ensures a vehicle follows a straight path ($Y = 0$) using a Model Predictive Control (MPC) approach, specifically the Regulator method. The scenario assumes the vehicle initially maintains a constant lateral offset($Y = 1$) and travels at a constant speed. The primary objective of this exam is to design a controller using MATLAB that enables the vehicle to converge to and stably follow the path within the simulation time. The MPC controller must satisfy the following optimization problem:

$$\text{minimize } J(x,u) = \frac{1}{2}x_N^T S_N x_N + \frac{1}{2}\sum_{k=0}^{N-1} x_k^T Q_k x_k + u_k^T R_k u_k$$
$$\text{subject to } \mathbf{x_{k+1}} = \mathbf{A}\mathbf{x_k} + \mathbf{b}\mathbf{u_k}$$
$$\mathbf{y_k} = \mathbf{C}\mathbf{x_k}$$
$$\mathbf{x_0} = given$$

[Q1.1] In this task, you are required to develop the `fMPC_regulating()` function, as detailed in the provided code comments. It should implement an MPC (Regulator) based algorithm tailored to the specific lateral control problem at hand. In addition to designing the controller, you are expected to generate graphs of the vehicle's states and input signals but also plotting the cost function J over time.

*** Please ensure to utilize MATLAB's `quadprog` function for this task. ***

[Q2.2] Discuss the role of 'Terminal Cost Tuning' in enhancing the convergence of a Model Predictive Control (MPC) system. Provide a clear explanation of the concept and its implementation in MPC. Afterwards, apply 'Terminal Cost Tuning' in your MPC code and present the results through comparative graphs.

Q2) MPC-Tracking

This project involves the development of a lateral controller for a vehicle, employing a Model Predictive Control (MPC) strategy, specifically tailored for Tracking applications. Unlike the previous scenario (Q1), here the reference path is not a straight line ($Y \neq 0$), but a lane-changing trajectory. Additionally, the vehicle starts without any initial lateral offset.

The core objective of this task is to design an MPC (Tracking) controller using MATLAB, ensuring that the vehicle precisely follows the given lane-changing path. The controller must enable the vehicle to accurately track the dynamic path and maintain stability throughout the simulation duration.

$$\begin{aligned}
&\text{minimize } J(e,u) = \mathbf{\frac{1}{2}e_N^T S_N e_N} + \mathbf{\frac{1}{2}\sum_{k=0}^{N-1} e_k^T Q_k e_k + u_k^T R_k u_k} \\
&\text{subject to } \mathbf{x_{k+1}} = \mathbf{A x_k + B u_k} \\
&\qquad\qquad\quad \mathbf{e_k} = \mathbf{r_k - C x_k} \\
&\qquad\qquad\quad \mathbf{x_0} = given
\end{aligned}$$

[Q2.1] In this assignment, you are tasked with crafting the `fMPC_tracking()` function, following the guidelines outlined in the code commentary. This function must implement an MPC (Tracking) based algorithm, designed specifically for the given control problem. Unlike the previous task, this function should utilize an augmented model where the changes of system input(Δu) are considered as the inputs to the controller.

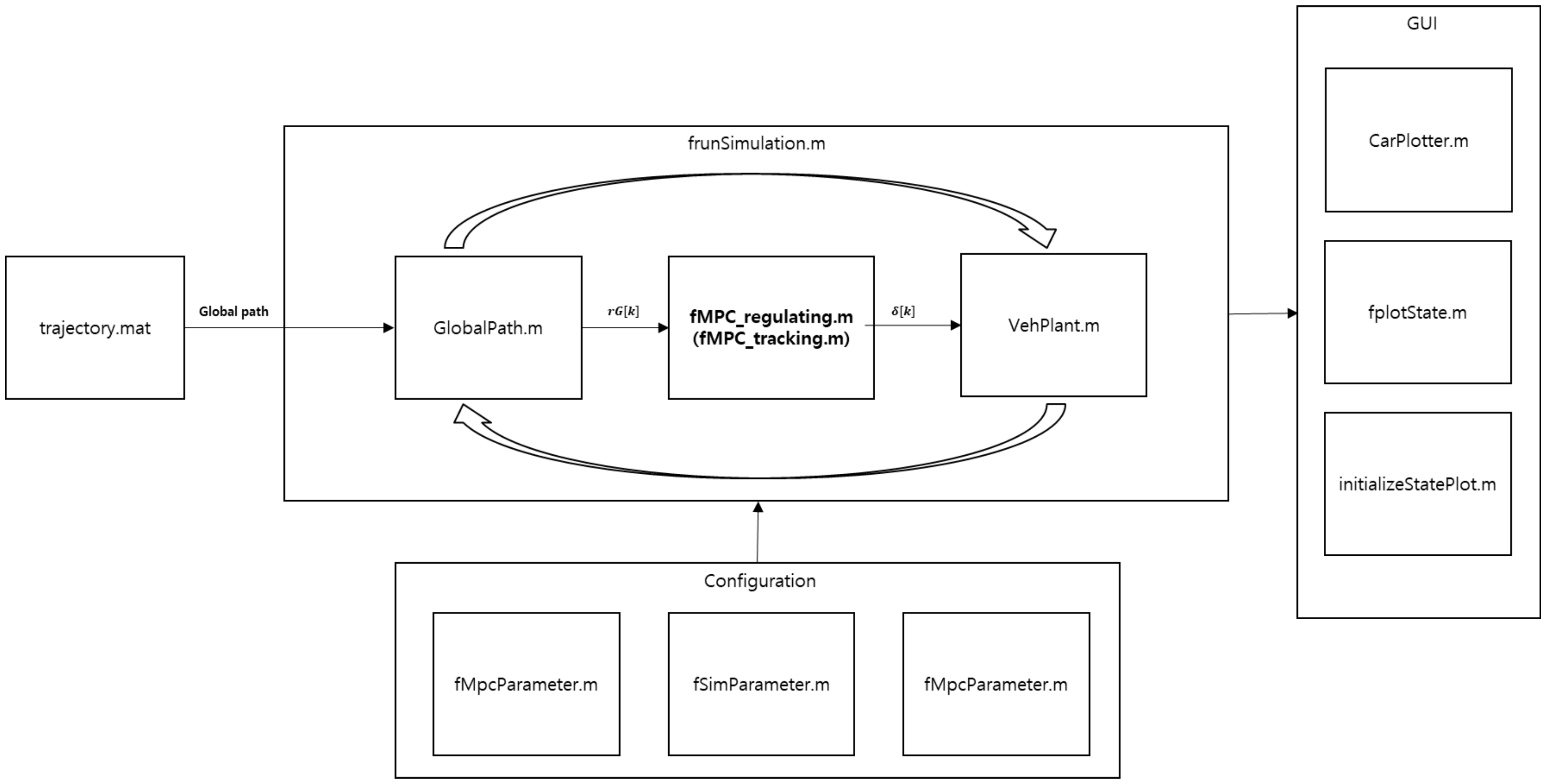
Your responsibility includes not only the development of the controller but also the graphical representation of its performance. This includes plotting both the state error and the cost function J over time, providing a comprehensive view of the controller's accuracy in tracking the desired path.

*** It is crucial for this task to employ MATLAB's `quadprog` function. ***

[Q2.2] Apply Input (u) and Input Rate (Δu) Inequality Constraints to the Controller Designed Above. Investigate the implementation of Input (u) and Input Rate (Δu) inequality constraints in the previously designed Model Predictive Control (MPC) system. To observe the impact of these constraints, define arbitrary constraint values and compare the performance of the MPC system before and after applying them. Analyze and discuss the results to demonstrate the effect of incorporating u and Δu constraints.

Simulation Configuration & Programming Guide

The structure of the Matlab program provided for the MPC controller simulation is designed as follows. As described in the earlier problem, the majority of your task involves writing the `fMPC_regulating.m` (`fMPC_tracking.m`) file. Additionally, you may modify `fMPCParameter.m` and `fSimParameter.m` if necessary for tuning your controller design or changing the simulation environment. Furthermore, you are free to make any additional modifications or additions to demonstrate the analysis results of the performance of your designed controller.


GUI
CarPlotter.m
fplotState.m
initializeStatePlot.m
frunSimulation.m
trajectory.mat
Global path
GlobalPath.m
$rG[k]$
fMPC_regulating.m
(fMPC_tracking.m)
$\delta[k]$
VehPlant.m
Configuration
fMpcParameter.m
fSimParameter.m
fMpcParameter.m