Optimal tuning and neural emulation of MPC for power management in fuel cell hybrid electric vehicles

Optimal tuning and neural emulation of MPC for power management in fuel cell hybrid electric vehicles

L. Calogero[1], M. Pagone[1], F. Cianflone[2], E. Gandino[2], C. Karam[2], and A. Rizzo*[,1]

Introduction

Power management in a fuel cell hybrid electric vehicle (FCHEV) consists of splitting efficiently the power generated by the battery and the fuel cell (FC), ensuring that the net delivered power meets the total power requested by the vehicle. In this work, we investigate the use of Model Predictive Control (MPC) to perform such power management task. The proposed MPC scheme features a novel approach for the optimal choice of the cost function weights, based on Particle Swarm Optimization (PSO), in order to achieve multiple control objectives, such as requested power tracking and minimum supplies consumption. Also, the MPC controller employs a neural data-driven approximation of the real plant as internal prediction model; this makes the controller employable in real-case scenarios, where only input-output plant measurements are available. We also present an alternative controller, based on feedforward neural networks (NNs), which emulates the whole MPC-based optimal control law. The NN-MPC controller is able to reliably reproduce the control action of the original controller, with a significantly lower computation time, making it suitable for real-time implementation on low-end control units.

1. Power supplies model

The FCHEV power supplies (i.e., battery and FC) are the plant under control (Fig. 1). The plant model takes as inputs the battery/FC power (P_b, P_{fc}) and provides as output the present battery state of charge ($SOC \equiv \xi$), the remaining hydrogen mass (m_h), and the net delivered power (P_{tot}). The model is discretized with time step T_s, i.e.,

$$\boldsymbol{x} = [\xi, m_h]^\top, \quad \boldsymbol{u} = [P_b, P_{fc}]^\top, \quad \boldsymbol{y} = [\xi, m_h, P_{tot}]^\top \tag{1}$$

$$\begin{cases} \boldsymbol{x}_{k+1} = \begin{bmatrix} \xi_{k+1} \\ m_{h,k+1} \end{bmatrix} = \begin{bmatrix} \xi_k + T_s \cdot f_1(\xi_k, P_{b,k}) \\ m_{h,k} + T_s \cdot f_2(P_{fc,k}) \end{bmatrix} \\ \boldsymbol{y}_k = [\xi_k, \;\; m_{h,k}, \;\; \eta_b P_{b,k} + \eta_{fc} P_{fc,k}]^\top, \end{cases} \tag{2}$$

with $k = t/T_s$, $t = nT_s$, $n \in \mathbb{N}_{\geq 0}$.

2. Data-driven MPC controller

The MPC prediction model approximates the plant model (2) by means of two feedforward neural networks (FNNs) for the plant state and output equations, respectively. The NNs are trained through plant input-output data. The prediction model is discretized with time step $T_s' = mT_s$, $m \in \mathbb{N}_{\geq 1}$ (i.e., the control time step), and step-by-step linearized at each control time instant $h = t/T_s'$, $t = nT_s'$, $n \in \mathbb{N}_{\geq 0}$, obtaining affine time-variant prediction model constraints in the MPC optimal control problem.

*Corresponding author

[1]Department of Electronics and Telecommunications, Politecnico di Torino, Turin, Italy

[2]Punch Softronix, Turin, Italy

E-mail addresses: lorenzo.calogero@polito.it (L. Calogero), alessandro.rizzo@polito.it (A. Rizzo)

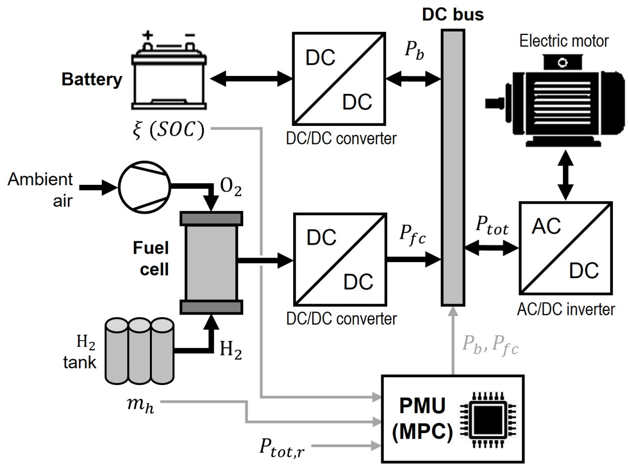

Figure 1. FCHEV power management system.

The MPC controller solves, at each control time instant h, the following QP finite-horizon optimal control problem

$$\min_{\hat{\boldsymbol{u}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{x}}} \sum_{i=0}^{N_p-1} \left(\|\hat{\boldsymbol{y}}_{i|h} - \boldsymbol{y}_{r,h}\|^2_{\boldsymbol{Q}} + \|\hat{\boldsymbol{u}}_{i|h}\|^2_{\boldsymbol{R}} \right) + \sum_{i=1}^{N_p-1} \left(\|\hat{\boldsymbol{y}}_{i|h} - \hat{\boldsymbol{y}}_{i-1|h}\|^2_{\boldsymbol{Q}_\Delta} + \|\hat{\boldsymbol{u}}_{i|h} - \hat{\boldsymbol{u}}_{i-1|h}\|^2_{\boldsymbol{R}_\Delta} \right) + \|\hat{\boldsymbol{y}}_{N_p|h} - \boldsymbol{y}_{r,h}\|^2_{\boldsymbol{P}} + \rho |\boldsymbol{\varepsilon}|^2 \tag{3a}$$

s.t.

$$\hat{\boldsymbol{x}}_{i+1|h} = \boldsymbol{A}_h \hat{\boldsymbol{x}}_{i|h} + \boldsymbol{B}_h \hat{\boldsymbol{u}}_{i|h} + \boldsymbol{b}_h, \quad i = 0, 1, \dots, N_p \tag{3b}$$

$$\hat{\boldsymbol{y}}_{i|h} = \boldsymbol{C}_h \hat{\boldsymbol{x}}_{i|h} + \boldsymbol{D}_h \hat{\boldsymbol{u}}_{i|h} + \boldsymbol{d}_h, \quad i = 0, 1, \dots, N_p \tag{3c}$$

$$\hat{\boldsymbol{x}}_{0|h} = \boldsymbol{x}_h \tag{3d}$$

$$\hat{\boldsymbol{u}}_{i|h} = \hat{\boldsymbol{u}}_{N_c-1|h}, \quad i = N_c, \dots, N_p \tag{3e}$$

$$\underline{\boldsymbol{y}}_h - \boldsymbol{\varepsilon} \leq \hat{\boldsymbol{y}}_{i|h} \leq \overline{\boldsymbol{y}}_h + \boldsymbol{\varepsilon}, \quad i = 0, 1, \dots, N_p \tag{3f}$$

$$\underline{\boldsymbol{u}}_h \leq \hat{\boldsymbol{u}}_{i|h} \leq \overline{\boldsymbol{u}}_h, \quad i = 0, 1, \dots, N_c - 1 \tag{3g}$$

$$\underline{\boldsymbol{u}}_{\Delta,h} \leq \tfrac{1}{T_s'} (\hat{\boldsymbol{u}}_{i|h} - \hat{\boldsymbol{u}}_{i-1|h}) \leq \overline{\boldsymbol{u}}_{\Delta,h}, \quad i = 1, \dots, N_c - 1 \tag{3h}$$

$$\boldsymbol{\varepsilon} \geq \boldsymbol{0} \tag{3i}$$

where: N_p, N_c are the prediction and control horizons; (3a) is the cost function; (3b)-(3d) is the affine time-variant prediction model; (3f)-(3h) are bound constraints on the output, input, and input rate; (3i) is the slack variable constraint.

The cost function (3a) features a reference tracking term for the output, penalization terms for the input, output rate, input rate, and slack variable, and a terminal cost. The terminal cost enforces closed-loop asymptotic stability for suitable values of N_p and $\boldsymbol{P}$ [1]. In presence of a bounded error on the optimal control input, it is ensured only finite-time convergence [1].

The control policy is a 1-step receding horizon, applied to the plant (2) over a control time step.

3. MPC controller tuning

The tuning of the MPC weights $\boldsymbol{w}$ (i.e., the elements of the diagonal weighting matrices $\boldsymbol{Q}$, $\boldsymbol{R}$, $\boldsymbol{Q}_\Delta$, $\boldsymbol{R}_\Delta$, $\boldsymbol{P}$ in (3a)) is formulated as a multi-objective optimization problem, minimizing an objective function f that is the weighted sum of several components c_i, i.e.,

$$\min_{\boldsymbol{w}\in\Omega_w} f(\boldsymbol{w}), \quad f(\boldsymbol{w}) = \sum_{i=1}^{n_c} \alpha_i \cdot c_i(\boldsymbol{w}). \tag{4}$$

Each component c_i quantifies a different aspect of control performance. The weights $\boldsymbol{\alpha} = [\alpha_i]_{i=1}^{n_c}$ set the priority of each term. For our purposes, the objective function has three components, associated with: 1) reference power tracking, 2) consumption of the battery, 3) consumption of hydrogen.

To solve the tuning optimization problem (4), Particle Swarm Optimization (PSO) is employed [2].

4. Neural network emulation of the MPC controller

The tuned MPC controller (3)-(4) is emulated by means of a FNN, taking as input the present state $\boldsymbol{x}_h$ and reference output $\boldsymbol{y}_{r,h}$ and providing as output the optimal control input $\boldsymbol{u}_h$ [3]. The universal approximation theorem of FNNs ensures boundedness of the error of the NN output [4], meaning that the NN-MPC controller grants finite-time convergence [1].

5. Results and simulations

The MPC and NN-MPC controllers are tested to control the plant model (2) with a predefined vehicle power request $P_{tot,r}$ (Fig. 1, 2). The plant and the MPC prediction model are perturbed with measurement noises, to empirically assess the controller robustness.

Simulations evaluate the following aspects:

- MPC tuning performance with multiple control objectives;
- MPC/NN-MPC controller performance;
- NN-MPC controller emulation capability of the original MPC controller;
- MPC and NN-MPC execution time.

5.1. MPC controller tuning

Four control objectives are considered to test the MPC tuning capabilities:

1) Lowest tracking error;
2) Low tracking error, low cumulative supplies consumption;
3) Low tracking error, low battery consumption;
4) Low tracking error, low hydrogen consumption.

Setting up accordingly the weights $\boldsymbol{\alpha}$ of the tuning objective function (4), optimal MPC weights $\boldsymbol{w}$ are obtained through PSO. Simulating the tuned MPC controller, the obtained control performance, for each control objective, is reported in Table 1.

Table 1. MPC tuning performance.

Control objective	Battery cons.	H_2 cons.	Cumulative cons.	Track. accuracy	Max track. error
1	80.17%	54.46%	67.31%	100.00%	2.12%
2	68.44%	51.28%	59.86%	100.00%	9.88%
3	49.70%	79.10%	64.40%	99.96%	10.11%
4	94.63%	26.41%	60.52%	100.00%	6.57%

The tuning algorithm has achieved all the given control objectives, in terms of tracking and supplies consumption, outperforming manual tuning, even in presence of uncertainties in the identified MPC prediction model (comparisons are here omitted due to space constraints).

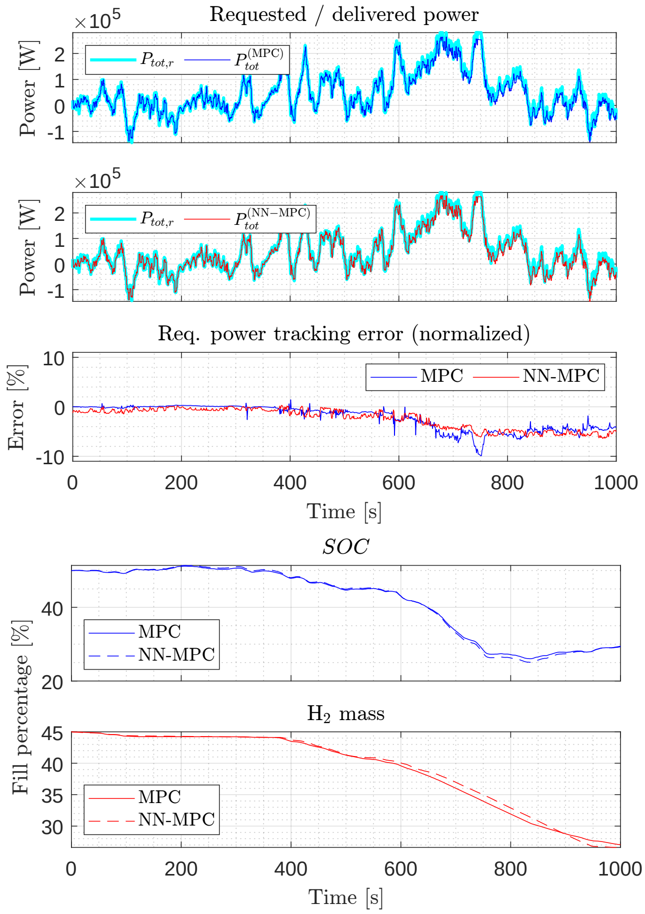

Figure 2. MPC/NN-MPC controllers performance.

5.2. MPC/NN-MPC controllers

Considering control objective 2, we simulate the tuned MPC controller and the corresponding NN-MPC controller.

From Fig. 2, we observe that both controllers achieve excellent tracking accuracy and minimize the cumulative supplies consumption. Also, from Fig. 2, we notice that MPC and NN-MPC achieve comparable performance, proving that NN-MPC is reliable and able to accurately emulate the MPC controller.

5.3. Execution time

The maximum execution time of 1 control step has been measured for both controllers, obtaining 11.44 ms for MPC and 0.096 ms for NN-MPC. NN-MPC outperforms MPC in terms of speed, resulting suitable for automotive real-time implementation, even on control units with low computational resources.

References

[1] T. Parisini and R. Zoppoli, "A Receding-horizon Regulator for Nonlinear Systems and a Neural Approximation," *Automatica*, vol. 31, no. 10, pp. 1443–1451, 1995.

[2] S. Mirjalili, J. Song Dong, A. Lewis, and A. S. Sadiq, "Particle Swarm Optimization: Theory, Literature Review, and Application in Airfoil Design," in *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*, S. Mirjalili, J. Song Dong, and A. Lewis, Eds. Springer, 2020.

[3] S. Wang, T. Dragicevic, Y. Gao, and R. Teodorescu, "Neural Network Based Model Predictive Controllers for Modular Multilevel Converters," *IEEE Transactions on Energy Conversion*, vol. 36, no. 2, pp. 1562–1571, 2020.

[4] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.