Development of Batteries Smart Management System Using Machine Learning

Ahmed Shokry & Eric Moulines

Center for Applied Mathematics, Ecole Polytechnique, France

Outlines on work done/ in progress

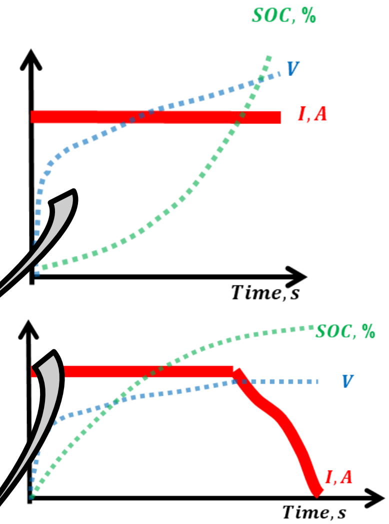
1) **Charging Control Based on ML** (conference and journal paper)

2) **State Of-Health (SOH) Estimation** (standard battery tests, MIT & NASA datasets)

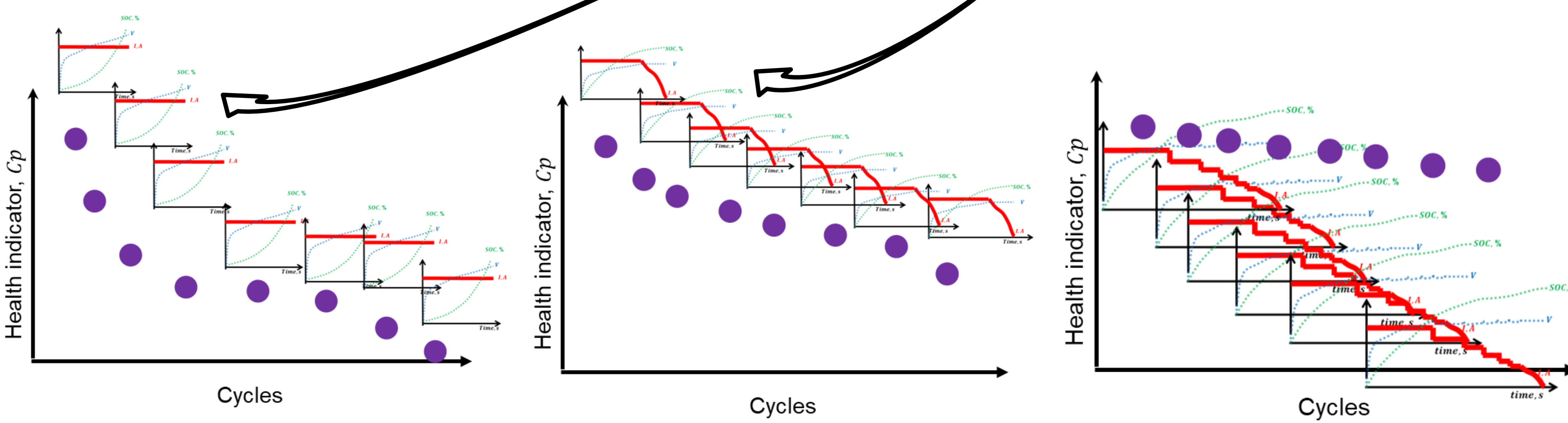
3) **SOH & Remaining Useful Life (RUL) Estimation** (field data, in progress)

4) **Report on SOH and RUL Estimation from field data** (in progress)

Importance of Battery Charging Protocols

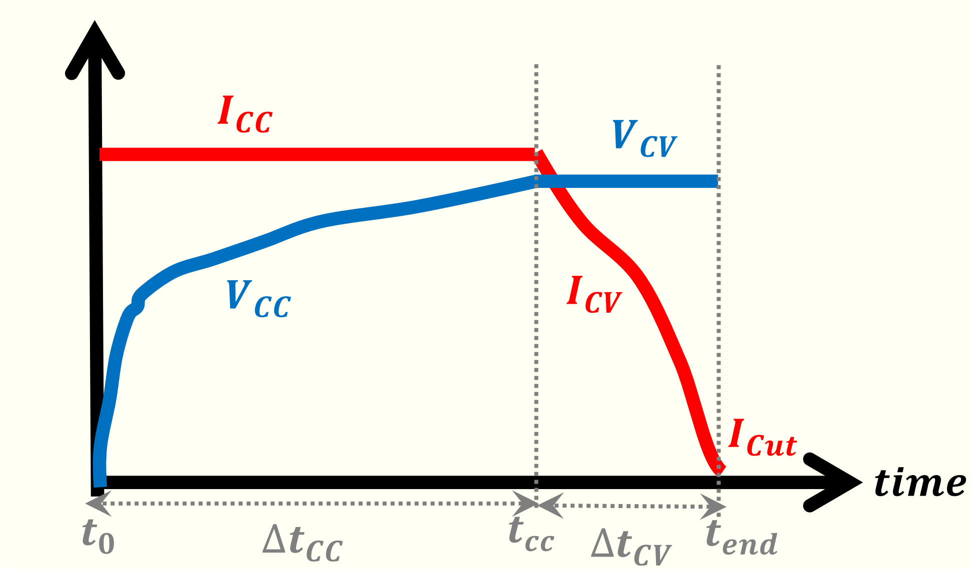


- Batteries withstand **1000+s** of charging actions
- **Degradation rate** $\xLeftarrow{dependent}$ **charging manner**
  - Heat generated, temperature rise, over voltage
  - Irreversible damage and capacity loss



- **BMSs rely on <u>empirical</u> charging protocols, such as:**

Constant Current-Constant Voltage (CC-CV)

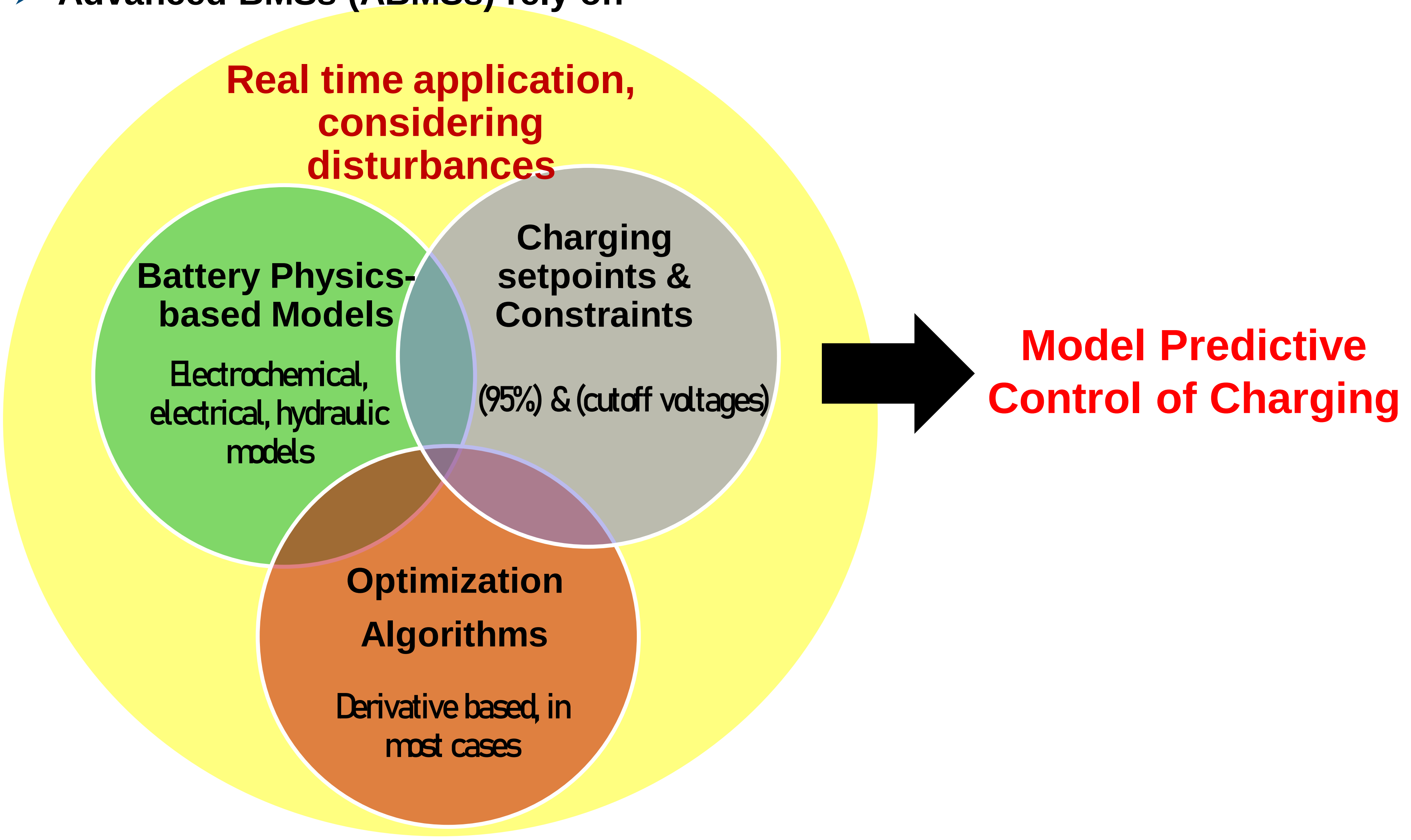


- Experimentally designed protocols
- **<u>Fair</u>** performance (speed & safety), but **<u>not optimal</u>**, since
  - do not exploit knowledge about battery dynamic (electrochemical, thermal, etc.)
  - do not consider real-time conditions (initial SOC, temperature, etc.)

1) Charging Control Based on ML

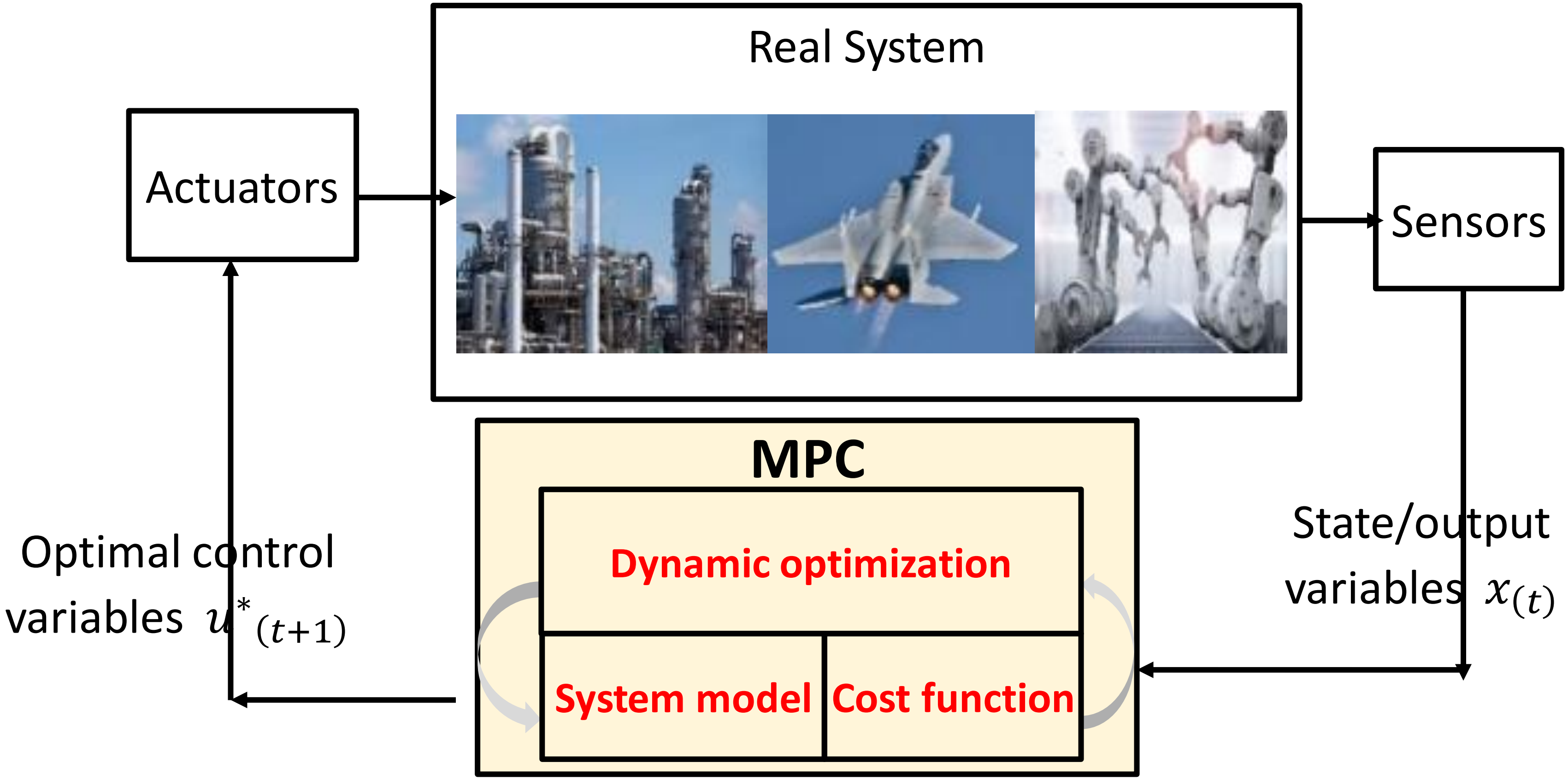
1.1) Problem Context

- **Advanced BMSs (ABMSs) rely on**



1) Charging Control Based on ML

1.2) Model Predictive Control (MPC)



Advantages:

- Multivariate nonlinear systems: complex Multiple-Inputs-Multiple Outputs (MIMO) relations
- Hard constraint on state and/or control variables

Perfectly suit for batteries (MIMO, nonlinear systems; operational, safety and health constraints)

1) Charging Control Based on ML

1.2) Model Predictive Control (MPC)

$$\min_{u_0,\dots,u_{N_p}} J$$

$$= x'_{N_p}\, P\, x_{N_p} + \sum_{k=0}^{N_p-1} \left[(x_k - \check{r})' Q\, (x_k - \check{r}) + \Delta u'_k\, \mathcal{R}\, \Delta u_k \right]$$

S.T.:

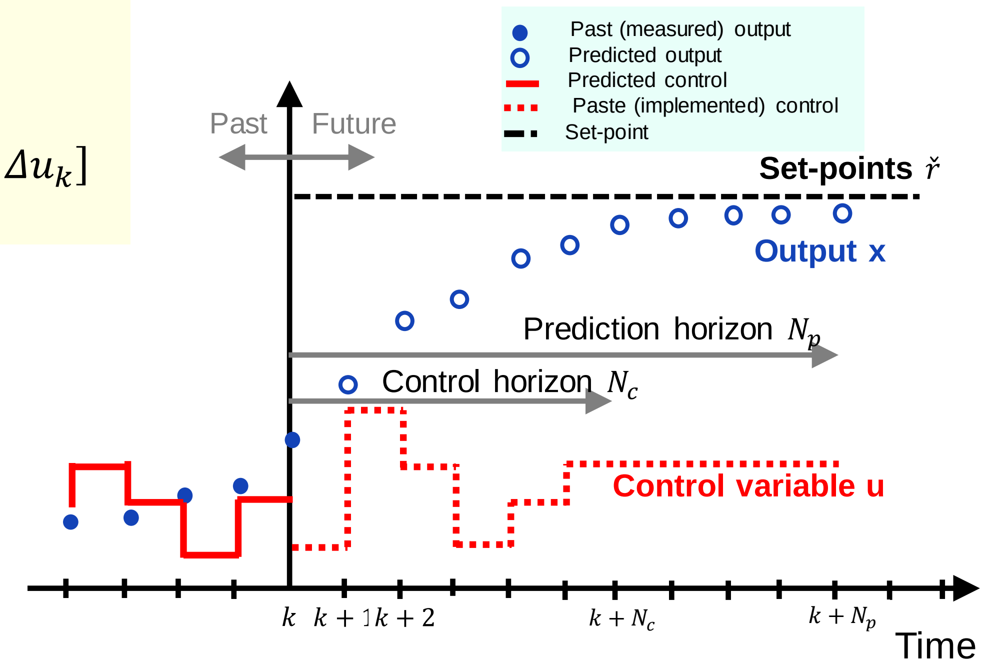
$$x_{min} \le x_k \le x_{max},$$

$$u_{min} \le u_k \le u_{max},$$

$$x_{k+1} = \boldsymbol{F}(x_k, u_k), \qquad x \in R^m, u \in R^v,$$

$$g_l(x_k, u_k) \le 0\,, \qquad l = 1,2,\dots L$$

- x_k: State/output variables
- u_k: Control variables
- $\check{r} \in R^m$: Setpoints
- Δu_k: Control increment $\Delta u_k = u_k - u_{k-1}, k = 0{:}N_p$
- $\boldsymbol{F}$: System model
- g_l: Constraints
- $Q, P, \mathcal{R}$: coefficient matrices



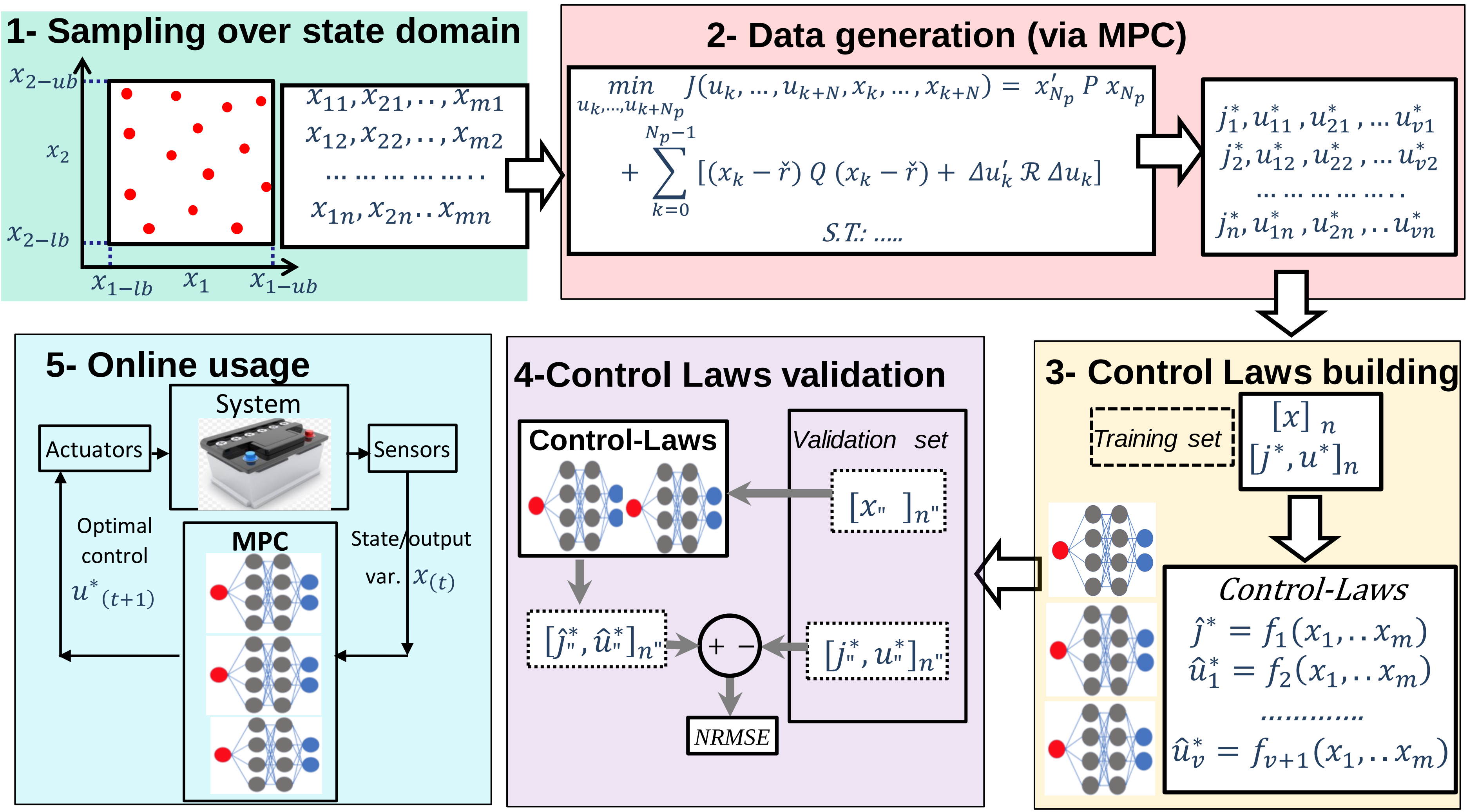
Shortcomings:

- Very demanding computations
  - Repeated solution of an *optimal control optimization*
- Infeasible for operational implementation in many practical cases

1) Charging Control Based on ML

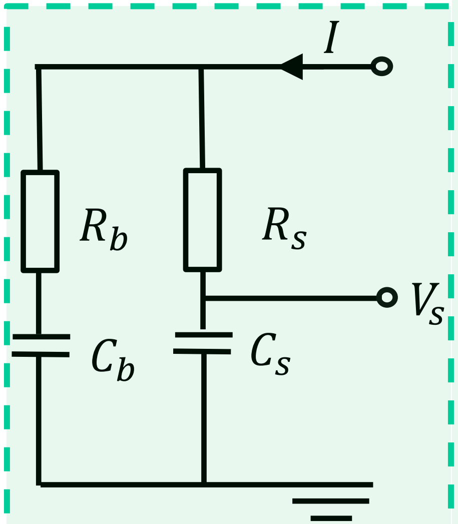
1.3) Machine-learning-based Explicit MPC

Methodology 3: ML- based explicit MPC



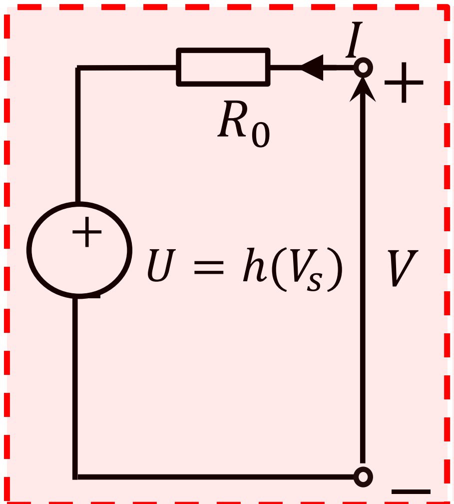
1) Charging Control Based on ML

1.4) Applications : E-MPC of Nonlinear Double-Capacitors



First part

- Two capacitor-resistor circuits
- $R_s - C_s$: electrode surface.
- $R_b - C_b$: electrode's inner bulk.
- Parallel link $\Longrightarrow$ migration of charge through the electrode
- V_s and V_b : voltages across C_s and C_b



Second part

- $U = h(V_s)$: is the open-circuit voltages
- R_0 : internal resistance
- V : terminal voltage
- I : charging/discharging current

Mathematical model

$$\begin{bmatrix} \frac{d\boldsymbol{V_b}(t)}{dt} \\ \frac{d\boldsymbol{V_s}(t)}{dt} \end{bmatrix} = A \begin{bmatrix} V_b(t) \\ V_s(t) \end{bmatrix} + B\, I(t)$$

$$\boldsymbol{SOC}(t) = \frac{C_b V_b(t) + C_s V_s(t)}{C_b + C_s}$$

$$\boldsymbol{V}(t) = V_{oc}(t) + R_0\, I(t)$$

$$\boldsymbol{R_0}(t) = \beta_0 + \beta_1 \exp(-\beta_3(1 - SOC(t)))$$

$$\boldsymbol{V_{oc}}(t) = h(V_s(t)) = \alpha_0 + \alpha_1 V_s(t) + \alpha_2 V_s(t)^2 + \alpha_3 V_s(t)^3 + \alpha_4 V_s(t)^4 + \alpha_5 V_s(t)^5$$

$$A = \begin{bmatrix} \frac{-1}{C_b(R_b + R_s)} & \frac{1}{C_b(R_b + R_s)} \\ \frac{1}{C_s(R_b + R_s)} & \frac{-1}{C_s(R_b + R_s)} \end{bmatrix}, B = \begin{bmatrix} \frac{R_s}{C_b(R_b + R_s)} \\ \frac{R_b}{C_s(R_b + R_s)} \end{bmatrix}$$

MPC

$$\min_{I_0,\dots,I_{N_p}} J = \sum_{k=0}^{N_p-1} [(SOC_k - \check{r})\, Q\, (SOC_k - \check{r}) + \Delta I_k' \,\mathcal{R}\, \Delta I_k]$$

S.T.:

Operational constraints

$$SOC_0 = 20\%\,,\, \check{r} = 90\%$$
$$0 \le I_k \le 3\ Amp,$$
$$V_k \le 4.2\ Volt,$$

Health constraints

$$V_{s,k} - V_{b,k} \le -0.04\ SOC_k + 0.08$$

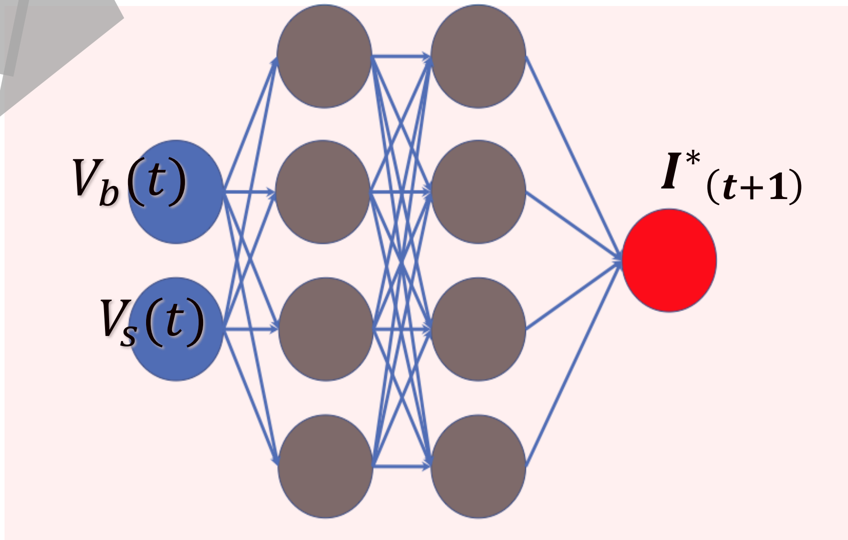
Model

$$x_{k+1} = \boldsymbol{F}(x_k, u_k),$$
$$x \in R^{m=2}, u \in R^{v=1}$$

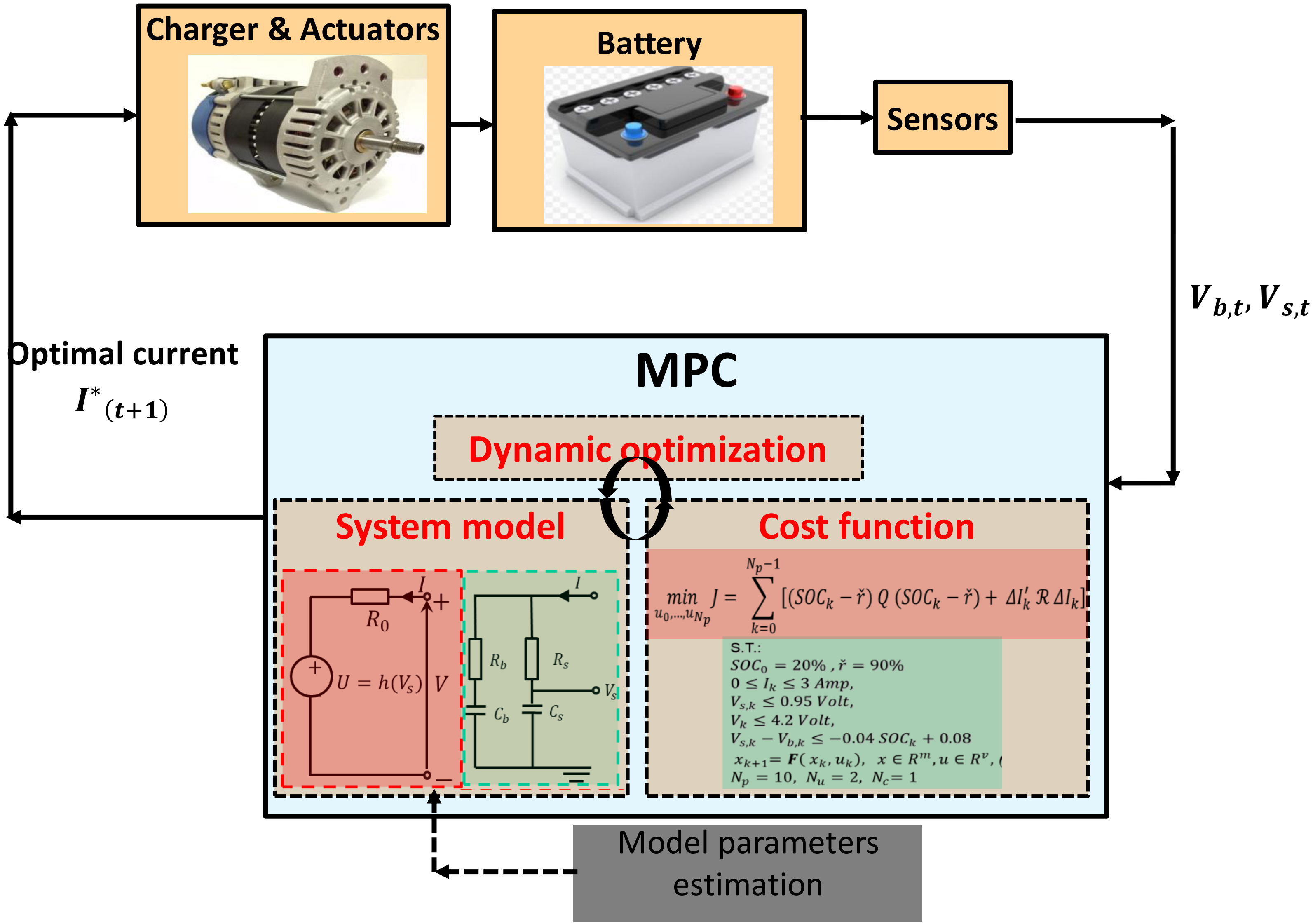
Coefficients

$$N_p = 10,\ N_u = 2,\ N_c = 1$$
$$Q = 1, \qquad R = 0.1$$

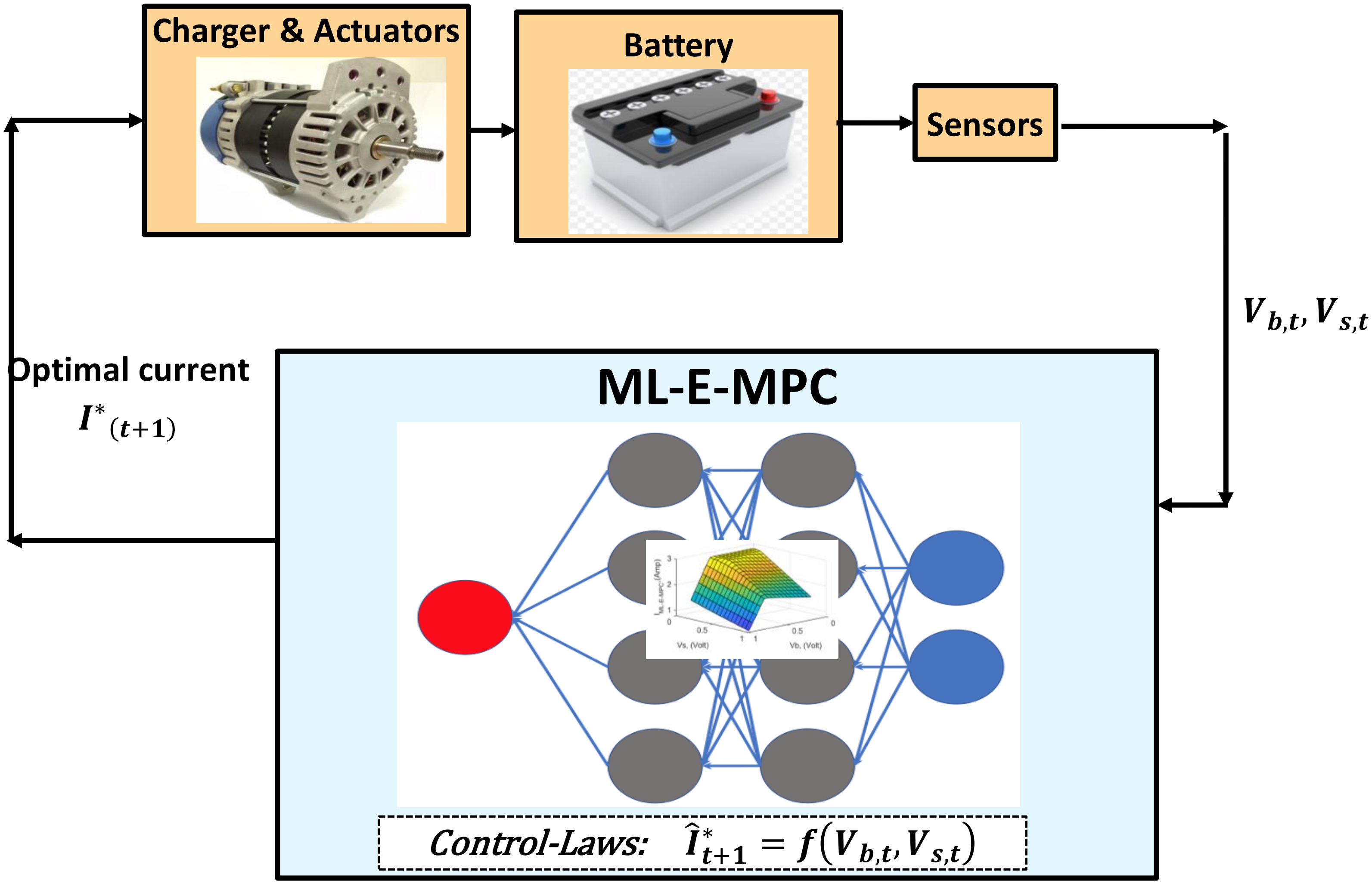
Apply the proposed method



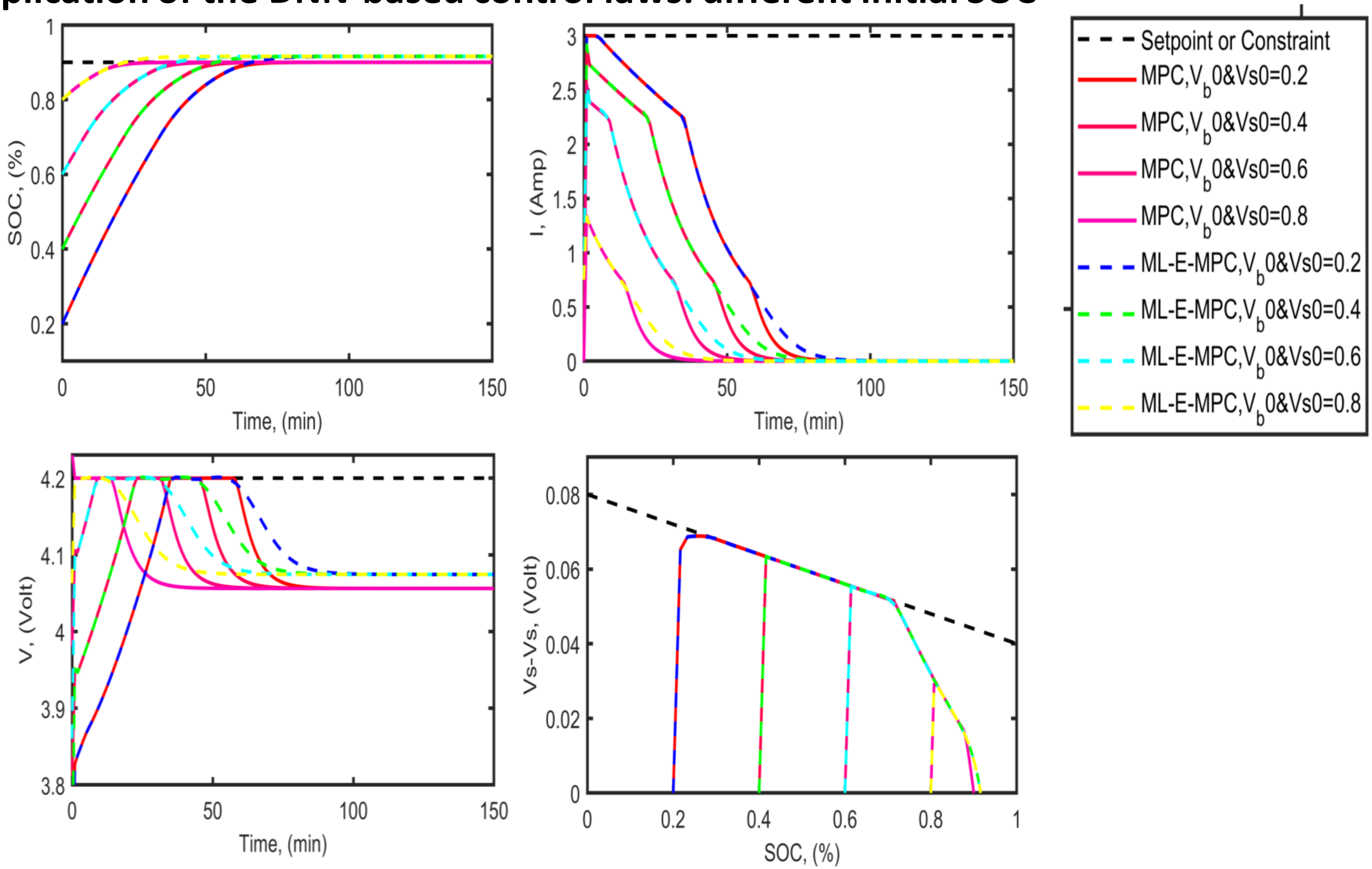
Application of the DNN-based control laws



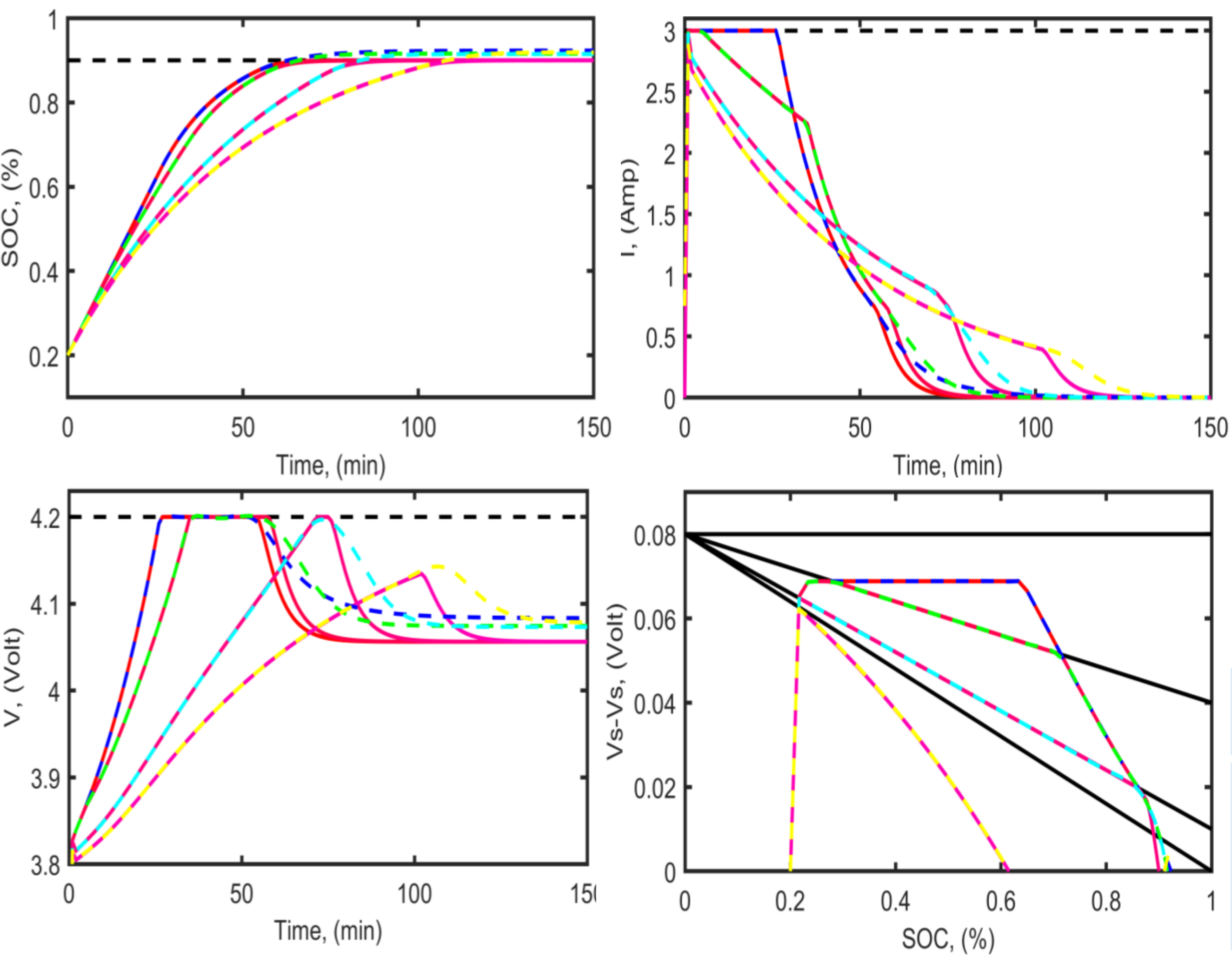
Application of the DNN-based control laws

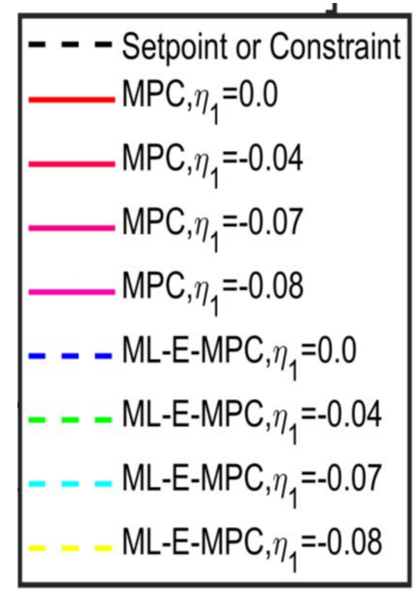


Application of the DNN-based control laws: different initial SOC



Application of the DNN-based control laws: different health-constraints adjustment





Performance

NRMSE (%)	Time (sec)*		Time saved (%)
	ML-E-MPC	*MPC*	
$\ll 1\%$	1.4	89.20	98.40