Iterative Dynamic Programming for Optimal Control Problem with Isoperimetric Constraint and Its Application to Optimal Eco-driving Control of Electric Vehicle

Van-Duc Doan*)
Takafumi Koseki
Hiroyuki Kishi**

(Manuscript received May 23, 2017)

An iterative dynamic programming (iDP) is proposed along with an adaptive objective function for solving optimal control problem (OCP) with isoperimetric constraint. Its application is investigated for optimal eco-driving control problem in electric vehicle (EV). The proposed method reduces the computational effort and enhances the global convergence of using iDP. Numerical calculations show that for the same computational time, the proposed method guarantees higher quality of solution when compared to basic iDP. In addition, the proposed method also achieves high accuracy of optimal solution with much less computational time than basic dynamic programming (DP), thus allowing for potential online implementation. Furthermore, the proposed iDP is successful in solving the optimal eco-driving control problem of EV with very long operational range.

Keywords: dynamic programming, iterative dynamic programming, optimal control problem, isoperimetric constraint, electric vehicle, eco-driving

Nomenclature

- $n$: Size of state vector.
- $m$: Size of control vector.
- $k$: Size of isoperimetric constraint.
- $N$: Number of stages.
- $N_i$: Size of iterations.
- $N_p$: Number of passes.
- $N_f$: Number of iterations for updating shifting vector.
- $M$: Number of desired feasible state trajectories.
- $n_u^\text{min}$: Minimal size of control grid ($u$).
- $n_u^\text{max}$: Maximal size of control grid ($u$).
- $n_x^\text{min}$: Minimal size of state grid ($x$).
- $n_x^\text{max}$: Maximal size of state grid ($x$).
- $\Delta_x^\text{min}$: Minimal resolution of state ($x$).
- $\Delta_x^\text{max}$: Minimal resolution of state ($x$).
- $R_u^\text{min}$: Minimal radius of state ($x$).
- $R_u^\text{max}$: Maximal radius of state ($x$).
- $X_u^\text{min}$: Minimal state bound adjusted ($n \times N$).
- $X_u^\text{max}$: Maximal state bound adjusted ($n \times N$).
- $x_i^\text{min}$: Minimal state bound at time stage $i$ ($n$).
- $x_i^\text{max}$: Maximal state bound at time stage $i$ ($n$).
- $u_i^\text{min}$: Minimal control bound at time stage $i$ ($m$).
- $u_i^\text{max}$: Maximal control bound at time stage $i$ ($m$).
- $r_u^\text{min}$: Minimal radius of control ($u$).
- $r_u^\text{max}$: Maximal radius of control ($u$).
- $\theta$: Penalty factor for final state constraint.
- $\beta$: Expansion factor of reachable state space.
- $\gamma$: Contraction factor of control and state bounds.
- $\eta$: Restoration factor of control and state bounds.
- $\alpha$: Convergence factor for isoperimetric constraint.
- $\Phi$: Weight factor for isoperimetric constraint ($k$).
- $N_u$: Size of control grid ($m \times N$).
- $N_x$: Size of state grid ($n \times N$).
- $X_u^\text{min}$: Minimal state bound approximated ($n \times N$).
- $X_u^\text{max}$: Maximal state bound approximated ($n \times N$).
- $\bar{x}^\text{min}$: Minimal state bound adjusted ($n \times N$).
- $\bar{x}^\text{max}$: Maximal state bound adjusted ($n \times N$).
- $U_u^\text{min}$: Discretized control ($m \times m \times N$).
- $U_u^\text{max}$: Discretized control ($m \times m \times N$).
- $U_x^\text{min}$: Discretized control ($m \times m \times N$).
- $U_x^\text{max}$: Discretized control ($m \times m \times N$).
- $x^*$: Optimized state ($n \times N$).
- $U^*$: Optimized control ($n \times N$).
- $C_u^\text{min}$: Center of control space ($n \times N$).
- $C_u^\text{max}$: Center of control space ($n \times N$).
- $R_u^\text{max}$: Radius of control space ($n \times N$).
- $R_u^\text{min}$: Radius of control space ($n \times N$).
- $n_u^\text{rid}$: Size of state grid at time stage $i$ ($n$).
- $n_u^\text{rid}$: Size of control grid at time stage $i$ ($m$).
- $x_i^\text{rid}$: Minimal state bound approximated at stage $i$ ($n$).

*) Correspondence to: Van-Duc Doan. E-mail: doanvanduc.ac.hut@gmail.com
** The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
** Technova Inc.
The Imperial Hotel Tower
1-1-1, Uchisaiwai-cho, Chiyoda-ku, Tokyo 100-0011, Japan
Due to increasing concerns for environment issues, strategies of using energy effectively in transportation system are becoming more important than before; especially, when the number of vehicles on the road has increased worldwide. For EV, whose battery capacity is limited, the effective use of energy by reducing its energy consumption also solves the problem of short mileage per charge. For example, range extension control system (RECS) is proposed to extend the cruising range of a vehicle by motion control. They assumed that an EV has more than one motor and that suitable torque distribution results in improved efficiency.

Another approach to reduce energy consumption is to optimize the speed profile of EV, and it is named as "eco-driving". In this approach, there are mainly three solution methods that are used popularly in automotive area, namely Pontryagin’s maximum principle (PMP), analytical solutions and dynamic programming (DP). PMP method is to transform the original OCP to a two-point boundary-value problem (TPBVP) that is time-consuming if multiple shooting algorithm is used. Furthermore, it is difficult to handle with state constraints. With the method based on analytical solutions, the solution of OCP is in a closed form; thus, it is suitable for online solving of OCP. However, the vehicle model must be simplified for the global optimality of solution. For that reason, we propose the global solution with an allowed accuracy. These ideas are mainly based on mechanism of iDP. First, if the cost-to-go problem can reach from the initial state. In addition, using the same grid sizes of each control directions of DP such as state increment, neuro, iterative, approximate, adaptive were also proposed to reduce the computational effort, however only applicable to some specific classes of problems. Notably, iterative dynamic programming (iDP) was improved considerably, and applied successfully in chemical process engineering with high dimensions. Although iDP has been proved to be successful in obtaining the global optimum with the use of only single state grid point for many typical chemical problems, where calculation time is very small. But, there are some problems such as the fed-batch reactor problem, where iDP with single state grid point can not find the global optimal solution. Although the global optimality of solution can be enhanced by increasing the number of state grid points, the computational time becomes serious like basic DP, which is not desired with iDP. Therefore, the global optimality of solution with iDP is still not guaranteed completely.

For control of hybrid electric vehicle (HEV), iDP was used as dynamic optimization in a model predictive controller (MPC). With a short range of vehicle operation, iDP was reported to have potential in real-time applications. To reduce the required memory and computational time, the state space is reduced as narrow as possible. However, in many cases, where the state space can not be reduced, basic iDP must calculate all points in the state space, even they are not in the forward-reachable state space. Notice that forward-reachable state space is defined to contain feasible states from which the process can reach the final feasible state zone. Other directions of DP such as state increment, neuro, iterative, approximate, adaptive were also proposed to reduce the computational effort, however only applicable to some specific classes of problems. Notably, iterative dynamic programming (iDP) was improved considerably, and applied successfully in chemical process engineering with high dimensions. Although iDP has been proved to be successful in obtaining the global optimum with the use of only single state grid point for many typical chemical problems, where calculation time is very small. But, there are some problems such as the fed-batch reactor problem, where iDP with single state grid point can not find the global optimal solution. Although the global optimality of solution can be enhanced by increasing the number of state grid points, the computational time becomes serious like basic DP, which is not desired with iDP. Therefore, the global optimality of solution with iDP is still not guaranteed completely.

For control of hybrid electric vehicle (HEV), iDP was used as dynamic optimization in a model predictive controller (MPC). With a short range of vehicle operation, iDP was reported to have potential in real-time applications. To reduce the required memory and computational time, the state space is reduced as narrow as possible. However, in many cases, where the state space can not be reduced, basic iDP must calculate all points in the state space, even they are not in the forward-reachable state space. Notice that forward-reachable state space is defined to contain feasible states from which the process can reach from the initial state. In addition, using the same grid sizes of each control/state at every time stage enlarges the grid sizes and consequently, computational time.

To the best of our knowledge, there is no proper method to reduce computational effort of iDP for OCP where the state space is large, or iDP with single grid point can not guarantee the global optimality of solution. For that reason, we proposed an improved iDP, named as AHIDP, whose ideas are to reduce the computational time while still guaranteeing the global solution with an allowed accuracy. These ideas are mainly based on mechanism of iDP. First, if the cost-to-go
function is calculated in only forward-reachable state space, the computational effort is reduced. Therefore, we proposed an approximate generation of forward-reachable state space. Second, since the radii of either controls or states in iDP at every time stage are different, grid sizes of controls or states should be adapted at each time stage accordingly. To achieve the same fine resolution, these grid sizes are assigned based on the radii of controls or states at every time stage in each iteration. Since these proposals are already presented in the other paper of ours, they are briefly restated in Appendix.

Although our AHiDP for optimal eco-driving control of EV is proved to outperform DP/iDP(s) in terms of computational effort or solution accuracy, when the operational range of EV becomes longer even from several ten kilometers to more than hundred kilometers, DP/iDP(s) or AHiDP cannot be applied due to their required memory and computational time. In that scenario, an improvement of AHiDP for optimal eco-driving control of EV with very long operational range is required. Therefore, this paper presents a method to define new state variables, i.e., \( z(t) \) satisfying:

\[
\frac{dz(t)}{dt} = y(x(t), u(t), t); t \in [0; T].
\]

And, the additional state constraints are derived as below:

\[
z(0) = z_0; z(T) = c.
\]

Therefore, we obtain a canonical OCP without isoperimetric constraint, where the state vector of the OCP are \( \mathbf{x} = (x, z) \). The solution methods for this OCP using Pontryagin’s maximum principle can be found in any textbook of optimal control theory. However, these analytical methods face difficulties in considering complicated constraints and dynamic equations. Whereas, DP/iDP(s) can easily handle these constraints and complicated models. The next sections are about reviews of these DP/iDP(s).

### 3. Conventional Approaches for Solving Optimal Control Problem with Isoperimetric Constraint

#### 3.1 Calculus of Variations

The oldest isoperimetric problem well known in calculus of variations is Dido’s problem, whose solution is easily obtained by using Euler-Lagrange equations. From viewpoint of optimal control problem, the isoperimetric problem can be regarded as a simple case of an OCP, where \( f \equiv u \), and there is no inequality \( h(x(t), u(t), t) \leq 0 \) involved. In other words, the OCP with isoperimetric constraint is much more complicated than the original isoperimetric problem. Therefore, the method based on Euler-Lagrange equations cannot be directly used for solving the OCP Eq. (1), s.t. Eq. (2)~Eq. (7). A simple way to consider the isoperimetric constraint (4) in the OCP is to define new state variables, i.e., \( z(t) \) satisfying:

\[
\frac{dz(t)}{dt} = y(x(t), u(t), t); t \in [0; T].
\]

#### 3.2 Basic Dynamic Programming

The discretized form of the OCP obtained in Section 3.1 is derived as below:

\[
\arg\min_{u(t) \in U} \left\{ J = \varphi(x(T), T) + \sum_{i=0}^{N-1} \tilde{g}(x_i, u_i, t_i) \right\}
\]

s.t.,

\[
\begin{align*}
\frac{dx(t)}{dt} &= f(x(t), u(t), t); t \in [0; T] \quad \text{.........(2)} \\
h(x(t), u(t), t) &\leq 0 \quad \text{.........(3)} \\
z(T) &= \int_0^T y(x(t), u(t), t) dt = c > 0 \quad \text{.........(4)} \\
x(t) &\in X(t) \subseteq \mathbb{R}^m \quad \text{.........(5)} \\
u(t) &\in U(t) \subseteq \mathbb{R}^m \quad \text{.........(6)} \\
x(0) &= x_0; x(T) = x_f \quad \text{.........(7)}
\end{align*}
\]

Where, \( \psi \) is a scalar function. Beside bounds of both state and control in Eq. (5) and Eq. (6), the path constraint Eq. (3) and the equality constraint Eq. (4) (i.e., isoperimetric constraint) are also considered. Notice that size of \( z(T) \) is assumed to be \( k \in \mathbb{N}; c \) in Eq. (4) a vector of constants.


Iterative Dynamic Programming Isoperimetric Constraint Electric Vehicle Eco-driving (Van-Duc Doan et al.)

Fig. 1. Backward calculation of basic DP in 2-D state space of $x_1$ and $x_2$; $\Delta x_{1i}$ and $\Delta x_{2i}$ are resolutions of $x_1$ and $x_2$ at time stage $i$, respectively

Fig. 2. Backward calculation of iDP-RG in 2-D state space of $x_1$ and $x_2$; $\Delta x_{1i}$ and $\Delta x_{2i}$ are resolutions of $x_1$ and $x_2$ at time stage $i$, respectively. The yellow curve is the optimized state trajectory obtained at each iteration.

Fig. 3. Backward calculation of iDP-IG in 2-D state space of $x_1$ and $x_2$. The yellow curve is the optimized state trajectory obtained at each iteration.

Eq. (11)

In the first version of iDP, named as iDP-RG (i.e., iDP-RG uses Regular Grid of state points), both control and state grids are generated independently around their optimized profiles obtained in the previous iteration with coarse resolution, then the control and state regions are contracted with their respective rate of $\gamma$. The procedure stated above is repeated until the solution is obtained (17). Since calculating the cost-to-go function is only executed at exact forward-reachable state points, and the number of these points is small regardless of system dimension, iDP-IG can be applied easily in high dimensions with small memory used. This is its big advantage. However, using less state grid points for some problems such as fed-batch reactor (22) did not guarantee the global optimality of solution. Although increasing the state grid points can enhance the possibility of obtaining the global solution, computational time increases. The reason is that unlike iDP-RG, since the state grid points are irregularly distributed, bilinear interpolation can not be used; instead, exact calculation of the cost-to-go function is executed by integrating the differential equations using the best controls of the nearest state point forward to the final stage. This process costs time if many stages and grid points are investigated. That could be a drawback of iDP-IG.

4. Iterative Dynamic Programming with Adaptive Objective Function for Optimal Control Problem with Isoperimetric Constraint

In Section 3.2 and Section 3.3, with the definition of new state variables $z(t)$ in Eq. (8) and additional fixed state constraints in Eq. (9), DP/IDP(s) can be used to solve the derived
Iterative Dynamic Programming Isoperimetric Constraint Electric Vehicle Eco-driving (Van-Duc Doan et al.)

OCP Eq. (10), s.t. Eq. (11)–Eq. (15). However, this approach enlarges the size of the original OCP, leading to its corresponding demerits. Especially, if $T$ is quite long, or space of $z(t)$ is wide, these disadvantages become more serious. Therefore, in order to consider the isoperimetric constraint Eq. (4) while still keeping the size of OCP as original, we define a new objective function as Eq. (17).

$$I = J + \Phi^T z_N = J + \Phi^T \sum_{i=0}^{N-1} y_i(x_i, u_i, t_i)$$

$$= \psi(x_f, T) + \sum_{i=0}^{N-1} \left( g_i(x_i, u_i, t_i) + \Phi^T y_i(x_i, u_i, t_i) \right)$$

$$\text{subject to} \quad y_i \in \mathbb{R}^n, u_i \in \mathbb{U}, e_i \in \mathbb{E}.$$ (17)

Where, $\Phi$ is a vector, whose size is the same as that of $y$ (i.e., $k$); $T$ is matrix transpose; $z_N$ is formulated as follows:

$$z_N = \sum_{i=0}^{N-1} y_i(x_i, u_i, t_i) \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdots \cdOTS (24) $
\[ F_{\text{mot}} = \begin{cases} u_{\text{mot}} \times F_0, & \text{if } 0 \leq v \leq v_0 \\ u_{\text{mot}} \times \frac{F_0 v_0}{v}, & \text{if } v_0 < v \leq v_{\text{max}} \end{cases} \]

Braking: \( u_{\text{mot}} \in [-1; 0] \) (\( F_1, v_1 \) depend motor characteristics)

\[ F_{\text{mot}} = \begin{cases} u_{\text{mot}} \times F_1, & \text{if } 0 \leq v \leq v_1 \\ u_{\text{mot}} \times \frac{F_1 v_1}{v}, & \text{if } v_1 < v \leq v_{\text{max}} \end{cases} \]

\( v_0, v_1 \) and \( F_0, F_1 \) are base speeds and forces of motor. The motor power of EV i.e. \( P_{\text{mot}} \) is given by the equation:

\[ P_{\text{mot}} = \begin{cases} \eta_{\text{f}} \frac{F_{\text{mot}} \times v}{v}, & \text{if } F_{\text{mot}} \geq 0 \\ \eta_{\text{f}} \frac{F_{\text{mot}} \times v}{v}, & \text{if } F_{\text{mot}} < 0 \end{cases} \]

Where, \( \eta_{\text{f}} \) is inverter efficiency, \( \eta_{\text{mot}} \) is motor efficiency, it depends on motor torque and speed, as shown in Fig. 4. Thus, power supplied by battery to EV is:

\[ P_{\text{bat}} = P_{\text{mot}} + P_{\text{aux}} \]

From Eq. (25), we get:

\[ \left( \frac{dv}{dt} \right) \left( \frac{dx}{dt} \right) = \frac{dv}{dx} \frac{1}{\gamma_m \times m \left( F_{\text{mot}} - R(x, v) \right)} \]

6. Derivation of Optimal Control Problem with Isoperimetric Constraint for Eco-driving Control of Electric Vehicle

From Eq. (25), we get:

\[ \left( \frac{dv}{dt} \right) \left( \frac{dx}{dt} \right) = \frac{dv}{dx} \frac{1}{\gamma_m \times m \left( F_{\text{mot}} - R(x, v) \right)} \]

We define new variables, named as \( \tau \) and \( V \), where \( \tau \equiv x \), and \( V \equiv v^2 \). By some mathematical derivations, we obtain:

\[ \frac{dV}{d\tau} = \frac{2 \gamma_m \times m \left( F_{\text{mot}} - R(x, v) \right)}{\sqrt{V}} \]

And Eq. (32) can be transformed as below:

\[ \min_{-1 \leq \text{mot}(\tau) \leq 1} E_{\text{tot}} = \int_0^L f(P_{\text{bat}}) d\tau \]

\[ \min_{-1 \leq \text{mot}(\tau) \leq 1} J = E_{\text{tot}} + \phi \times T_{\text{act}} \]

\[ \int_0^L \left( f(P_{\text{bat}}) + \phi \right) d\tau \]

Where, \( \phi \in \mathbb{R} \) is a coefficient supposed to be changed as Eq. (40) so that Eq. (38) is satisfied. Where, \( \kappa(x) : \mathbb{R}^+ \rightarrow \mathbb{R}/\kappa(0) = 0 \) is a monotonically increasing function for controlling the convergence speed of \( \phi \) or \( T_{\text{act}} \).

\[ \phi^{(i+1)} = \phi^{(i)} + \text{sign}(T_{\text{act}}^{(i)} - T) \times \kappa(T_{\text{act}}^{(i)} - T) \]

According to Eq. (40), if \( T_{\text{act}} > T \), \( \phi \) increases; therefore, minimizing \( J \) in Eq. (39) results in reducing \( T_{\text{act}} \). Whereas, if \( T_{\text{act}} < T \), \( \phi \) decreases, which increases \( T_{\text{act}} \) when solving Eq. (39). In other words, adaptation mechanism of \( \phi \) like Eq. (40) guarantees that \( T_{\text{act}} \) convergences to \( T \) after a certain number of iterations. The function \( \kappa(x) \) can be chosen from well known functions described in Fig. 5. For example, if \( \kappa(x) \) is a linear function of \( x \) (i.e. \( \kappa(x) = \alpha x \)), control mechanism of \( \phi \) becomes an integral type as shown in Fig. 6. This mechanism guarantees error convergence of running time to zero when the number of iteration of AHiDP is large enough.
7. Numerical Results

7.1 Short Operational Range of Electric Vehicle

7.1.1 Convergence Characteristics of AHiDP with Adaptive Objective Function (AHiDP-1D)  A scenario of EV operation is examined as Fig. 7, where practical operational conditions of EV are considered such as: sectional-speed constraints, road slope, etc. The desired running time of EV is assumed to be $T = 185$ [s]; EV speeds before starting and after finishing its routine are zero (i.e., $v_{0} = v_{f} = 0$ [km/h]; $x_{0} = x_{f} = 3500$ [m]). Parameter settings of AHiDP-1D are given in Table 2.

Figure 8 and Fig. 9 demonstrate optimized profiles of control signal and EV speed respectively at each iteration. They show that when iteration passes, these optimized profiles converge to a curve that is the desired solution.

Figure 10 and Fig. 11 show the numbers of grid points for control signal and EV speed at every stage, respectively. They are minimal radii of control signal and EV speed; (2) $r_{min}$ are settings of AHiDP-1D. Whereas, $v' = 0$ [km/h], $x' = 3500$ [m], $r' = 185$ [s] are chosen for optimal solution; $v_{scale} = 150$ [km/h], $x_{scale} = 3500$ [m], $t_{scale} = 185$ [s] are chosen for scaling; these parameters are fixed. Figure 13 displays accumulative computational time $t_{CPU}$, final state error $\delta_f$ and energy error $\delta_E$ obtained at each iteration. At the 9th iteration, $\delta_f < 1\%$ and $\delta_E < 1\%$ with accumulative computational time of $t_{CPU}$ $\approx 229$ [s], which means that high accuracy of solution can be obtained with no need of executing AHiDP-1D until its final iteration. At final iteration, the optimal energy consumption of $E^* \approx 0.639984$ [kWh].

7.1.2 Comparison with Basic Dynamic Programming

For comparing the computational effort and solution accuracy of basic DP and AHiDP-1D, the same scenario of EV operation as Section 7.1.1 is investigated, where both DP and AHiDP-1D are used to solve the OCP. Parameter settings of DP and AHiDP-1D are tabulated in Table 3. We consider four cases, Case 1~Case 4 when using DP and two cases,

| Table 2. Parameter settings for AHiDP-1D with short operational range of EV. Notice that (1) $n_{u, min}$ (15 ~ 45) are ranges of the number of control signal and EV speed; (2) $\Delta u_{min}$ and $\Delta v_{min}$ are minimal resolutions of control signal and EV speed; (3) $r_{u, min}$ and $r_{v, min}$ are minimal radii of control signal and EV speed, respectively |

<table>
<thead>
<tr>
<th>$N$</th>
<th>$N_i$</th>
<th>$N_s$</th>
<th>$N$</th>
<th>$n_{u, min}$</th>
<th>$n_{v, min}$</th>
<th>$\Delta u_{min}$</th>
<th>$\Delta v_{min}$</th>
<th>$r_{u, min}$</th>
<th>$r_{v, min}$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>25</td>
<td>1</td>
<td>15</td>
<td>15 ~ 45</td>
<td>15 ~ 45</td>
<td>0.001</td>
<td>0.36</td>
<td>0.007</td>
<td>2.52</td>
<td>0.1</td>
<td>0.2</td>
<td>50.0</td>
<td>0.75</td>
</tr>
</tbody>
</table>
Table 3. Comparison of computational effort and accuracy of solution between DP and AHiDP-1D. Notice that (1) in DP, \( n_u, n_v, \) and \( n_x \) are numbers of grid points of control signal, EV speed, and EV position, respectively. In AHiDP-1D, \( n_u, n_v, \) and \( n_x \) are ranges of the number of control signal and EV speed, respectively; single-pass scheme is adopted \( (N_u = 25; N_v = 1) \); (2) \( n_{RK} \) is number of times for solving the differential equation by the fourth-order Runge-Kutta (RK4) method \(^{10} \); (3) \( t_{CPU} \) is computational time measured by clock of CPU; (4) \( v_f, x_f, T_{act} \) and \( E_{tot} \) are the final speed, final position, actual running time and energy consumption of EV; (5) \( \delta \) is solution error, where the numbers in parentheses at the \( \delta \) [%] column describe indices of iteration of AHiDP(s).

<table>
<thead>
<tr>
<th>Method</th>
<th>Case</th>
<th>( n_u )</th>
<th>( n_v )</th>
<th>( n_x )</th>
<th>( n_{RK} )</th>
<th>( t_{CPU} ) [s]</th>
<th>( v_f ) [km/h]</th>
<th>( x_f ) [m]</th>
<th>( T_{act} ) [s]</th>
<th>( E_{tot} ) [kWh]</th>
<th>( \delta ) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>Case 1</td>
<td>75</td>
<td>75</td>
<td>105</td>
<td>10.9279 \times 10^{3}</td>
<td>649.9230</td>
<td>1.8543</td>
<td>3458.6911</td>
<td>185.0000</td>
<td>0.712431</td>
<td>5.7242</td>
</tr>
<tr>
<td></td>
<td>Case 2</td>
<td>75</td>
<td>75</td>
<td>305</td>
<td>31.7084 \times 10^{3}</td>
<td>1915.7260</td>
<td>0.4704</td>
<td>3498.8725</td>
<td>185.0000</td>
<td>0.661473</td>
<td>1.6064</td>
</tr>
<tr>
<td></td>
<td>Case 3</td>
<td>75</td>
<td>75</td>
<td>405</td>
<td>42.1467 \times 10^{4}</td>
<td>2410.0970</td>
<td>0.2351</td>
<td>3498.5168</td>
<td>185.0000</td>
<td>0.661466</td>
<td>1.6062</td>
</tr>
<tr>
<td></td>
<td>Case 4</td>
<td>75</td>
<td>75</td>
<td>805</td>
<td>83.7717 \times 10^{5}</td>
<td>6068.9870</td>
<td>0.0381</td>
<td>3493.7683</td>
<td>185.0000</td>
<td>0.652158</td>
<td>0.9553</td>
</tr>
<tr>
<td>AHiDP-1D</td>
<td>Case 5</td>
<td>15 \sim 35</td>
<td>15 \sim 35</td>
<td>-</td>
<td>4.6638 \times 10^{4}</td>
<td>229.3840</td>
<td>1.0897</td>
<td>350.0000</td>
<td>187.3663</td>
<td>0.638790</td>
<td>0.7414</td>
</tr>
<tr>
<td></td>
<td>Case 6</td>
<td>15 \sim 45</td>
<td>15 \sim 45</td>
<td>-</td>
<td>5.6046 \times 10^{4}</td>
<td>243.4880</td>
<td>0.9392</td>
<td>350.0000</td>
<td>187.1809</td>
<td>0.635260</td>
<td>0.7626</td>
</tr>
</tbody>
</table>

Fig. 10. Grid resolution adaptation of control at every iteration of AHiDP-1D with short operational range of EV.

Fig. 11. Grid resolution adaptation of speed at every iteration of AHiDP-1D with short operational range of EV.

Fig. 12. Actual time and coefficient \( \phi \) at every iteration of AHiDP-1D with short operational range of EV.

Case 5 ~ Case 6 when using AHiDP-1D.

To compare the computational effort of DP and AHiDP-1D, we count the number of times for solving the differential equation Eq. (25) by the fourth-order Runge-Kutta (RK4) method \(^{10} \), named as \( n_{RK} \). Notice that the step size \(^{1} \) of RK4 is chosen to be 0.1 [m]. Besides, the accumulative computational time of DP or AHiDP-1D, named as \( t_{CPU} \) [s], is measured by clock of CPU \(^{11} \).

\(^{1}\) The step size of RK4 method is chosen to be smaller than the step of DP to guarantee the accuracy of solving the differential equation.

\(^{11}\) Basic DP/AHiDP(s) are implemented in C/C++ language. All programs are run on Intel(R) Core(TM) i7-4790 CPU @ 3.6 GHz 3.6 GHz, RAM 32 GB, 64-bit OS, x64-based processor.
To evaluate the accuracy of solution obtained by AHiDP-1D, we define $\delta$ formulated by Eq. (43) as solution error.

$$\delta = \frac{1}{4} \left( \frac{v_f - v^*}{v_{\text{scale}}} \right)^2 + \left( \frac{x_f - x^*}{x_{\text{scale}}} \right)^2 + \left( \frac{T_{\text{act}} - T^*}{T_{\text{scale}}} \right)^2 + \left( \frac{E_{\text{tot}} - E^*}{E_{\text{scale}}} \right)^2 $$  \hspace{1cm} (43)

Where, $v_f$ [km/h] is the final speed; $T_{\text{act}}$ [s] is the actual running time; $x_f$ [km] is the final position; $E_{\text{tot}}$ is energy consumption of EV; these parameters are calculated by DP and AHiDP-1D. Whereas, $v^*$ [km/h], $x^*$ [km] is the final position; $E^*$ is chosen for optimal solution; $v_{\text{scale}} = 150$ [km/h], $x_{\text{scale}} = 3500$ [m], $T_{\text{scale}} = 185$ [s], and $E_{\text{scale}} = 0.639984$ [kWh] are chosen for scaling; these parameters are fixed. Notice that $E^*$ and $E_{\text{scale}}$ are chosen to be 0.639984 [kWh] that is calculated in Section 7.1.1.

Calculation results are summarized in Table 3. It shows that for DP, when the number of grid points is small (i.e. Case 1), the solution obtained does not satisfy EV operation since EV can not finish its routine on time, and the energy consumption is also quite large. And, when the number of grid points increases, the solution error reduces ($\delta < 2\%$ in Case 2–Case 3; $\delta < 1\%$ in Case 4). Notice that in order to achieve low solution error of $\delta < 1\%$, DP costs the least computational times of $t_{CPU} \approx 6069$ [s]. Meanwhile, AHiDP-1D only costs $t_{CPU} \approx 229$ [s] in Case 5, and $t_{CPU} \approx 243$ [s] in Case 6, even with the smaller solution error than that of DP.

Figure 14 and Fig. 15 visualize calculation results by DP and AHiDP-1D. We see that optimized profiles by DP from Case 2 to Case 4 in each figure are similar, and they converge to an optimal profile when the number of grid points increases. Notice that the optimized profiles by AHiDP-1D are slightly different from the ones obtained by DP.

7.1.3 Comparison with Basic Iterative Dynamic Programming  To compare performance of AHiDP(s) with iDP(s), all of these methods are used to solve optimal eco-driving control problem of EV with the same scenario of EV operation as Section 7.1.1. The parameter settings of iDP(s) and AHiDP(s) are given in Table 4; the other parameter settings of AHiDP(s) are the same as Table 2. Notice that:

1. AHiDP-1D means AHiDP with reducing the dimension of OCP (the OCP with isoperimetric constraint).

<table>
<thead>
<tr>
<th>Method</th>
<th>$n_p$ [point]</th>
<th>$n_v$ [point]</th>
<th>$n_t$ [point]</th>
<th>$t_{CPU}$ (time)</th>
<th>$v_f$ (km/h)</th>
<th>$x_f$ (m)</th>
<th>$T_{act}$ [s]</th>
<th>$E_{tot}$ [kWh]</th>
<th>$\delta$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>25</td>
<td>75</td>
<td>305</td>
<td>3.157980 $\times 10^3$</td>
<td>1915.7260</td>
<td>1.0470</td>
<td>3498.0725</td>
<td>185.0000</td>
<td>0.461473</td>
</tr>
<tr>
<td>AHiDP-1D</td>
<td>15 – 35</td>
<td>15 – 35</td>
<td>–</td>
<td>4.6638 $\times 10^6$</td>
<td>229.3840</td>
<td>1.0897</td>
<td>3500.0000</td>
<td>187.3663</td>
<td>0.638790</td>
</tr>
<tr>
<td>AHiDP-2D</td>
<td>25 – 35</td>
<td>25 – 35</td>
<td>–</td>
<td>4.9276 $\times 10^6$</td>
<td>310.5190</td>
<td>0.0023</td>
<td>3498.6672</td>
<td>185.0000</td>
<td>0.653237</td>
</tr>
<tr>
<td>iDP-RG</td>
<td>35</td>
<td>35</td>
<td>35</td>
<td>8.7323 $\times 10^3$</td>
<td>536.4350</td>
<td>0.2807</td>
<td>3498.8306</td>
<td>185.0000</td>
<td>0.680860</td>
</tr>
<tr>
<td>iDP-IG</td>
<td>51</td>
<td>3</td>
<td>3</td>
<td>18.56.99 $\times 10^3$</td>
<td>1880.2310</td>
<td>0.0000</td>
<td>2313.5432</td>
<td>185.0000</td>
<td>0.677938</td>
</tr>
</tbody>
</table>

Fig. 16. Comparison between optimized control profiles obtained by DP, iDP(s) and AHiDP(s) with different resolution.

Fig. 17. Comparison between optimized speed profiles obtained by DP, iDP(s) and AHiDP(s) with different resolution.

Fig. 18. Road map from Gotenba to Hamamatsu. Notice that $L = 146$ [km]; $T = 7800$ [s].

Fig. 19. Road elevation from Gotenba to Hamamatsu.
with the solution error \( δ_{\text{solution}} \) formulated by Eq. (41) and Eq. (42) at each iteration of AHiDP(s). Notice that \( v^0 = 0 \text{[km/h]} \), \( x^0 = 146000 \text{[m]} \), \( r^0 = 7800 \text{[s]} \) are chosen for optimal solution, and \( t_{\text{scale}} = 150 \text{[km/h]} \), \( x_{\text{scale}} = 146000 \text{[m]} \), \( t_{\text{scale}} = 7800 \text{[s]} \) are chosen for scaling. Figure 21 displays accumulative computational time \( t_{\text{CPU}} \), final state error \( δ_f \) and energy error \( δ_E \) obtained at each iteration. It shows that from the 6th iteration, \( t_{\text{CPU}} \approx T = 7800 \text{[s]} \) as desired, and the change of the \( φ \) is small.

To evaluate the accuracy of solution obtained by AHiDP-1D, we calculate \( δ_\text{solution} \) and \( δ_f \) formulated by Eq. (41) and Eq. (42) at each iteration of AHiDP(s). Notice that \( v^0 = 0 \text{[km/h]} \), \( x^0 = 146000 \text{[m]} \), \( r^0 = 7800 \text{[s]} \) are chosen for optimal solution, and \( t_{\text{scale}} = 150 \text{[km/h]} \), \( x_{\text{scale}} = 146000 \text{[m]} \), \( t_{\text{scale}} = 7800 \text{[s]} \) are chosen for scaling. Figure 21 displays accumulative computational time \( t_{\text{CPU}} \), final state error \( δ_f \) and energy error \( δ_E \) obtained at each iteration. It shows that from the 6th iteration, \( δ_f < 1\% \) and \( δ_E < 1\% \) with accumulative computational time \( t_{\text{CPU}} \approx 24351.547 \text{[s]} \approx 6 \text{[hour]} 46 \text{[minute]} \), which means that high accuracy of solution can be obtained with no need of executing AHiDP-1D until its final iteration. At final iteration, the optimal energy consumption of \( E^* \approx 12.008339 \text{[kWh]} \).

### 7.2.2 Computational Effort and Solution Error of AHiDP-1D with Long Operational Range of EV at Different Grid Sizes of Control and Speed

AHiDP-1D is used to solve the OCP with long operational range of EV, the same scenario of EV operation as Section 7.2.1. Parameter settings of AHiDP-1D are the same as Table 5. Four cases of grid sizes of control and speed are Table 6 are investigated.

To assess the accuracy of solution obtained by AHiDP-1D with different grid sizes of control signal and EV speed at each iteration, we calculate \( δ_f \) formulated by Eq. (43). Notice that \( v^0 = 0 \text{[km/h]} \), \( x^0 = 146000 \text{[m]} \), \( r^0 = 7800 \text{[s]} \), \( E^* = 12.008339 \text{[kWh]} \) are chosen for optimal solution; \( v_{\text{scale}} = 150 \text{[km/h]} \), \( x_{\text{scale}} = 146000 \text{[m]} \), \( t_{\text{scale}} = 7800 \text{[s]} \), \( E_{\text{scale}} = 12.008339 \text{[kWh]} \) are chosen for scaling. Notice that \( E^* \) and \( E_{\text{scale}} \) are chosen to be 12.008339 [kWh] that is calculated in Section 7.2.1.

Figure 22 shows accumulative computational time of AHiDP-1D at different grid sizes of control and speed of EV. It shows that the interruption time (i.e., the necessary computational time to achieve \( δ = 1\% \), error of solution) is much smaller than the total iteration time. In addition, the
computational time drastically reduces when the grid sizes of control and speed are reduced, and high accuracy of solution is still obtained even if the grid sizes of control and speed are small. As shown in Fig. 23, the optimized speed profile in each case of grid sizes is almost the same.

8. Conclusions

In this paper, we demonstrated implementation of AHIDP with our proposal of adaptive objective function for optimal control problem with isoperimetric constraint. Its application was focused on optimal eco-driving control of EV. Verification by numerical calculations showed that AHIDP-1D outperforms DP, iDP(s) and AHIDP-2D in terms of computational effort when all iterations pass; dark red bars show interruption time when the error of solution is smaller than 1%.

Fig. 22. Computational effort of AHIDP-1D with long operational range of EV at different grid sizes of speed and control

Fig. 23. Optimized speed profiles obtained by AHIDP-1D with long operational range of EV at different grid sizes of speed and control

References

(5) D. Grossosell and D. Mezief: ”Practical design of minimal energy controls for an electric bicycle”, in Proc. 9th Int. Conf. Modeling, Simulation and Optimization, Bordeaux, France (2012)
Appendix

1. Iterative Dynamic Programming with Approximate Generation of Forward-Reachable State Space and Grid Resolution Adaptation (AHiDP)

app. Fig. 1 briefly demonstrates AHiDP algorithm, where multi-pass iDP scheme and the shifting method for considering constraints of the final state are adopted (17).

1.1 Proposal of Approximate Generation of Forward-Reachable State Space (GRS)

1.1.1 Generation of Feasible State Trajectories

With the same approach like iDP-IG, feasible state trajectories are generated by integrating the differential equations with each array of the random controls. While iDP-IG only generates several feasible state trajectories, the proposed method generates many feasible state trajectories as possible to guarantee that they can represent the main features of the process, as shown in app. Fig. 2.

1.1.2 Approximation of Forward-reachable State Space

By generating the feasible state trajectories as presented above, we obtain feasible states at every time stage. The forward-reachable state space is approximated by a minimal rectangle containing these feasible states (the red rectangle in Fig. 2). Since the number of state trajectories (i.e., M) is finite, the approximate forward-reachable state space obtained can not reach the actual one. By adjusting the expansion factor \( \beta \in [1; +\infty] \), the approximate forward-reachable state space can be expanded. Although we can get a higher accuracy of the approximate forward-reachable state space by increasing \( M \), computational time becomes larger. Fortunately, \( M \) is not in need of a high value since the forward-reachable state space is re-calculated at each iteration, and it sweeps almost the actual one after a certain number of iterations. In implementation, \( M \) about from 10 ~ 100 is enough.

Notice that if the state space is too narrow, there might not be enough feasible state trajectories obtained with random control inputs. It means that the number of feasible state trajectories generated is smaller than the specified number (i.e., \( M \)), thus the approximate generation of forward-reachable state space obtained becomes not exact enough. In this case, state grid is generated as same as that of iDP-RG. For a general case, the bounds of states in forward-reachable state space can be adjusted as follows:

\[
\begin{align*}
    x_{\text{min}}^i &= \max(\hat{x}_{\text{min}}^i, c_i x_i - r_i) \\
    x_{\text{max}}^i &= \min(\hat{x}_{\text{max}}^i, c_i x_i + r_i)
\end{align*}
\]

Where, \( \min \) and \( \max \) are element-wise functions.

1.2 Proposal of Grid Resolution Adaptation (GRA)

Since the radii of either control or state in iDP-RG at every

app. Fig. 2. Approximate generation of forward-reachable state space. The yellow curve is the optimized state trajectory obtained at each iteration; the red rectangle containing all feasible state points is named as forward-reachable state space

app. Fig. 3. Grid resolution adaptation and backward calculation of iDP-RG in 2-D state space of \( x_1 \) and \( x_2 \). \( \Delta x_1 \) and \( \Delta x_2 \) are resolutions of \( x_1 \) and \( x_2 \) at time stage \( i \), respectively. The yellow curve is the optimized state trajectory obtained at each iteration. State grid points are regularly distributed in forward-reachable state space with an uniform density
time stage, i.e., \( r_{i}^w \) and \( r_{i}^e \) are different, grid sizes of control and state at each time stage, i.e., \( n_{i}^w \) and \( n_{i}^e \) should be chosen proportional to their radii. This guarantees the same resolutions in their entire space. For that reason, grid sizes of control and state are adapted as Eq. (A3) and Eq. (A4). That means the grid size is assigned to be proportional to its radius.

\[
\begin{align*}
\min n_{i}^w & = \max \left( n_{i,\text{min}}^w, \left[ n_{u,i}^\text{max} \times \frac{\bar{r}_{i}^w}{r_{i}^w} \right] \right) \quad \text{.......................... (A3)} \\
\max n_{i}^e & = \max \left( n_{i,\text{min}}^e, \left[ n_{u,i}^\text{max} \times \frac{\bar{r}_{i}^e}{r_{i}^e} \right] \right) \quad \text{.......................... (A4)}
\end{align*}
\]

Where,

\[
\begin{align*}
\bar{r}_{i}^w & = 0.5 \times \left( \min (u_{i}^\text{max}, c_{i}^w + r_{i}^w) - \max (u_{i}^\text{min}, c_{i}^w - r_{i}^w) \right) \\
\bar{r}_{i}^e & = 0.5 \times \left( \min (x_{i}^\text{max}, c_{i}^e + r_{i}^e) - \max (x_{i}^\text{min}, c_{i}^e - r_{i}^e) \right) \\
\bar{r}_{u,i}^w & = \max (\bar{r}_{i}^w, i = 0, N) \quad \text{.......................... (A5)} \\
\bar{r}_{u,i}^e & = \max (\bar{r}_{i}^e, i = 0, N) \quad \text{.......................... (A6)} \\
\bar{r}_{w,i}^\text{max} & = \max (\bar{r}_{i}^u, i = 0, N) \quad \text{.......................... (A7)} \\
\bar{r}_{e,i}^\text{max} & = \max (\bar{r}_{i}^e, i = 0, N) \quad \text{.......................... (A8)}
\end{align*}
\]

It should be noted that \( \min, \max \) and \( +, -, \times, /, \lceil \rceil \) in Eq. (A3)–Eq. (A8) are element-wise functions and operators, respectively. In addition, since the grid resolutions of controls and states are gradually reduced, after the certain number of iterations, the grid resolutions might become smaller than their preset minimums (i.e., \( \Delta u_{i}^\text{min} \) and \( \Delta x_{i}^\text{min} \)); thus, the grid sizes should be decreased in order to keep their resolutions at the minimum. In that case, the computational time at latter iterations can be reduced.

Van-Duc Doan (Non-member) received the B.E. degree in Automatic Control, Electrical Engineering from Hanoi University of Science and Technology (HUST), Vietnam, in 2009. After that, he was working as an assistant lecturer in the university in 2012. He received the Master degree in Electric Engineering and Information Systems from The University of Tokyo, Japan, in 2014. He is currently working towards the Doctoral degree in Electrical Engineering and Information Systems, The University of Tokyo. His research interests include infrastructure of wireless power transfer for electric vehicle, energy-saving operation of vehicle, control engineering, optimization, and evolution computation.

Hiroshi Fujimoto (Senior Member) received the Ph.D. degree from the Department of Electrical Engineering, The University of Tokyo, Chiba, Japan, in 2001. In 2001, he joined the Department of Electrical Engineering, Nagaoka University of Technology, Niigata, Japan, as a research associate. From 2002 to 2003, he was a visiting scholar in the School of Mechanical Engineering, Purdue University, U.S.A. In 2004, he joined the Department of Electrical and Computer Engineering, Yokohama National University, Yokohama, Japan, as a lecturer and became an associate professor in 2005. He has been an associate professor of The University of Tokyo since 2010. His interests include control engineering, motion control, nano-scale servo systems, electric vehicle control, and motor drives. Dr. Fujimoto received the Best Paper Award from the IEEE Transactions on Industrial Electronics in 2001 and 2013, the Isao Takahashi Power Electronics Award in 2010, the Best Author Prize of the Society of Instrument and Control Engineers (SICE) in 2010, and the Nagamori Grand Award in 2016. He is a member of IEEE, the SICE, the Robotics Society of Japan, and the Society of Automotive Engineers of Japan, and a Senior Member of the Institute of Electrical Engineers of Japan (IEEE).

Takafumi Koseki (Member) received the Ph.D. degree in electrical engineering from The University of Tokyo, Japan, in 1992. He is currently Professor of the Department of Electrical Engineering and Information Systems, The University of Tokyo. His research interests include public transport systems, particularly linear drives, and the analysis & control of traction systems. Prof. Koseki is a member of IEEE, the Institute of Electrical Engineers of Japan (IEEE), the Japan Society of Mechanical Engineering, the Japan Society of Applied Electromagnetic and Mechanics, the Japan Society of Precision Engineering, and the Japan Railway Engineering Association.

Tomio Yasuda (Member) received a bachelor’s degree from Tokyo Denki University in 1974 and joined Aisin AW Co., Ltd., where he was engaged in the development of automotive electronics and IT systems. He transferred to Technova Inc. in 1999, and is now engaged in survey research on automotive power electronics and system technologies. He was a member of the IEEJ Investigating R&D Committee on Power Electronics for Automobiles from 2007 to 2009. He is a member of JSAE.

Hiroyuki Kishi (Non-member) was born in Tokyo, Japan, in 1979. He received the B.S. degree in agricultural science from Tokyo University of Agriculture in 2002, and M.S. degree in Management of Technology from Shibaura Institute of Technology in 2010. He has been with Technova Inc. since 2004. He is engaged in research of the automotive power electronics, mainly on the wireless power transfer systems for electric vehicles.

Toshiyuki Fujita (Member) received the B.S. degree in electrical engineering and M.S. degree in physical electronics from Tokyo Institute of Technology in 2008 and 2010, respectively. Since 2014, he has been with Technova Inc. and working toward Ph.D. degree in electrical and electronic engineering at the Tokyo Institute of Technology. Prior to it, he was with Panasonic corporation. His research interests include wireless power transfer systems for electric vehicles, AC/DC converters and its control methods.