Near minimum-time direct voltage control algorithms for wheeled mobile robots with current and voltage constraints

Jong-Suk Choi and Byung Kook Kim*

Department of Electrical Engineering, KAIST 373–1 Kusong-dong Yusong-gu, Taejon 305–701 (Korea)
e-mail: bkkim@ee.kaist.ac.kr

(Received in Final Form: May 2, 2000)

SUMMARY
Near minimum-time direct voltage control (DVC) algorithms synthesizing path-planning and path-following are proposed for wheeled mobile robots (WMRs) satisfying (i) initial and final postures and velocities as well as (ii) voltage and current constraints. To overcome nonholonomic and nonlinear properties of WMRs, we divide our control algorithm for cornering motion into three sections: TSD (Translational Section of Deceleration), RS (Rotational Section), and TSA (Translational Section of Acceleration). We developed off-line DVC algorithms using the quadratic constraint in RS and velocity/torque-servo modules, while satisfying the current constraints. Two methods of searching for the two control parameters (number of steps for RS $M_R$ and velocity constraint in RS $\beta$) were considered: The one is composed of one simple 1-dimensional search for $\beta$, and the other is composed of two 1-dimensional searches for $M_R$ and $\beta$ which has better performance. Performances of the proposed control algorithms are validated via various simulations.

KEYWORDS: Mobile robot; Current and voltage constraints; Path planning; Near minimum time control

1. INTRODUCTION
Interest in wheeled mobile robots (WMRs) is growing rapidly due to much broad range of their potential applications – industrial automation, undersea/planet exploration, nuclear/explosives handling, warehousing, security, agricultural machinery, military, education, mobility for the disabled, and personal robots, etc.

It is well known that WMR is one of nonholonomic system where the system cannot be stabilized with smooth state feedback. Kinematic and dynamic modeling of WMRs has been addressed by several researchers. Control of WMRs is generally divided into two categories – path-planning (PP) and path-following (PF).

PP is to plan a trajectory connecting the given initial and final positions with or without obstacle avoidance. Dubins set the problem of characterizing the shortest path for a particle moving forward with a constant linear velocity and simple kinematic model of Dubins’ car under the constraint of bounded curvature. Later, Reeds and Shepp considered the same problem, where backward motions are allowed. A shortest path synthesis of Dubins’ car was determined according to Pontryagin’s Maximum Principle (PMP) by Soueres et al. Bicchi et al. extended the Reeds and Shepp’s results to the case where obstacles are present. However, in those researches based on the Dubins’ model, the curvature along the trajectory does not vary continuously since the optimal solutions are sequences of line segments and arcs of circle of minimal radius. One choice for path preserving curvature continuity is the clothoid, of which the curvature is linearly increased and then decreased with running distance along the curve. It was used as splines in computer aided design (CAD) and introduced in robotics by Kanayama et al.11,12

PF is to make stable control for mobile robots to follow the given path (trajectory). Kanayama et al. proposed critically damped controller which is Lyapunov stable. In Soueres’ research, obstacle avoidance is also included in the PF during transition phase using a sliding mode technique. Hamel studied robustness with respect to errors in path using a compact attractive domain around zero error. Dynamic robot model should be considered for fast moving.

Time-optimal control synthesis for PP and PF has been studied by several researchers but remains an open problem yet. It was initially addressed by Jacobs et al. in which minimum-time trajectories based on Hilare-like model are necessarily made up with bang-bang pieces proved with PMP. Reister made a numerical study of bang-bang trajectories containing only five elementary pieces with interesting time parameterization and having at most four control switches with no mathematical proof to bound the number of control switches, which has been invalidated by Renaud, who showed that certain configurations could not be reached by extreme trajectories containing only five elementary pieces and pointed out existence of extreme solutions allowing to reach those configurations and containing more than four switches. The above researches based on Hilare-like model include only linear and angular acceleration bounds.

Most researches considering dynamic model dealt with dynamic constraints of input torques only. In a few cases, just limitations of velocities or accelerations are included. However, since there are limits on motor’s
performance and battery’s power, WMR systems have motor armature current constraints as well as battery voltage constraint in practice. In previous researches, control inputs are velocities or accelerations with or without bounds where, for the low level control of motors, velocity-servo modules or torque-servo modules are used to generate desired control inputs. In practice, since final control inputs are voltages (PWM duty ratios) generated by those servo modules, there may exist bad cases where those modules cannot track the desired velocity/acceleration commands due to voltage and current constraints. Hence, efficient control algorithm for WMR systems considering those constraints is essentially required.

In this paper, near minimum-time direct voltage control (DVC) algorithms synthesizing path-planning and path-following are proposed for WMRs to satisfy given initial and final postures (positions and angles) and velocities as well as to consider both voltage and current constraints. DVC means that normalized voltages (equivalent to PWM duty ratios) which will be applied to both motors are controlled directly without velocity-servo or torque-servo module, while satisfying the current constraints. To overcome nonholonomic and nonlinear properties, we divide our control algorithm for turning motion into three sections: The first is RS (Rotational Section) which is focused on the rotational motion with the required turning angle, and the others are TSD (Translational Section of Deceleration) and TSA (Translational Section of Acceleration) which are secondary procedures focused on translational motion. No obstacles but exit-distance requirements (predetermined by user) are considered. We developed off-line DVC algorithms using the quadratic problem satisfying current and voltage constraints with the object function minimizing the total time. Two methods of searching for the two control parameters (number of steps for \( M_r \) and velocity constraint in RS \( \beta \)) were considered: The one is composed of one simple 1-dimensional search for \( \beta \), and the other is composed of two 1-dimensional searches for \( M_r \) and \( \beta \) which has better performance. To show the performance of the proposed controller, various simulations for different cases are presented.

The remainder of the paper is organized as follows. Section 2 gives problem statements with WMR dynamics. Near minimum-time control algorithms synthesizing path-planning and path-following are developed in Section 3 using MATLAB optimization toolbox. Simulation results are shown and discussed in Section 4. Section 5 is for concluding remarks.

2. PROBLEM STATEMENT

2.1 Dynamic model for WMRs

Assume that WMR has symmetrical structure driven by two identical DC motors as shown in Figure 1. Define posture as \([x(t) y(t) \theta(t)]^T\) and states \(x\) as \([i^1 \dot{i}^1 \dot{w} \dot{w}^T]\) of which elements are right motor current \(i^1\), left motor current \(\dot{i}\), linear velocity of WMR \(u\) and angular velocity of WMR \(w\). Then, WMR’s kinematics is defined as

When \(\cos \theta \neq 0\) and \(\sin \theta \neq 0\), the WMR dynamics are given as

\[
\begin{pmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{pmatrix} =
\begin{bmatrix}
\cos \theta & 0 & 0 \\
\sin \theta & 0 & 1
\end{bmatrix}
\begin{bmatrix}
u \\
w
\end{bmatrix}
\] (1)

If we select normalized right and left voltages to motors, \(u^1\) and \(u^2\), as control inputs, then WMR dynamics is derived as

\[
\begin{pmatrix}
x \end{pmatrix} =
A
\begin{pmatrix}
x \end{pmatrix} +
B
\begin{pmatrix}
u \\
w
\end{pmatrix}
\] (2)

Details of the derivation is presented in Appendix A. Overall dynamics of WMR system is illustrated in Figure 2, where \(I_z = \text{diag}(1,1)\).

We can convert to discrete-time version with sampling interval \(T_s\) as

\[
x_{k+1} = Gx_k + Hu_k, \quad x_k = [i^1 \dot{i}^1 \dot{w} \dot{w}]^T, \quad u_k = [u^1 \dot{u}^1]^T
\] (3)

where

\[
G = e^{AT_s}, \quad H = \left( \int_0^{T_s} e^{\lambda \Delta \lambda} d\lambda \right)B
\]

Using motor parameters in Table I, the values of \(G\) and \(H\) are decided as

\[
G =
\begin{pmatrix}
-0.0018 & -0.0013 & -2.0280 & -0.4126 \\
-0.0013 & -0.0023 & 0.0199 & -0.1887 \\
0.0007 & 0.0007 & 0.9051 & 0.0000 \\
0.0006 & -0.0006 & 0.0000 & 0.9853
\end{pmatrix}
\] \hspace{1cm}

\[
H =
\begin{pmatrix}
4.1679 & -0.0723 \\
-0.0723 & 4.1679 \\
0.0414 & 0.0414 \\
0.0341 & -0.0341
\end{pmatrix}
\]
2.2 Configurations and exit-distance

We can classify configurations of initial and final postures into two categories. The one is primary configuration (PC) where it is unnecessary to change the sign of rotational velocity for path-planning, and the other is compounded configuration (CC) where necessary to change that sign as shown in Figure 3. For simplicity, we consider the PC only in this research. Because of nonlinear and nonholonomic properties, a graphical approach is necessary to solve path-planning problem. Since all-time optimal paths for PC are expected to be made up with one rotational section and two translational sections surrounding the rotational one, we divide our control algorithm for cornering motion into three sections – The first is RS (Rotational Section) which is focused on the required turning angle, and the others are TSD (Translational Section of Declaration) and TSA (Translational Section of Acceleration) which are secondary procedures focused on condition of positions. Furthermore, obstacles are not considered explicitly. Instead we add constraint that both exit-distances \( d_R \) and \( d_{\bar{R}} \) should be bounded with \( D_R \) in RS as shown in Figure 4, which limits deviations from the given configuration, and hence obstacles can be avoided.

![Fig. 3. Classification of configurations.](image)

![Fig. 4. Sections in path and requirements of exit-distances.](image)

### Table I. Parameters of WMR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i_{\text{max}} )</td>
<td>4 A</td>
<td>( K_0 )</td>
<td>0.0801 Nm/A</td>
</tr>
<tr>
<td>( u_{\text{max}} )</td>
<td>1</td>
<td>( K_s )</td>
<td>0.0801 V/(rad/s)</td>
</tr>
<tr>
<td>( F_c )</td>
<td>0.8 Nm/(rad/s)</td>
<td>( R_s )</td>
<td>5.625 ( \Omega )</td>
</tr>
<tr>
<td>( b )</td>
<td>0.188 m</td>
<td>( L_s )</td>
<td>0.00384 H</td>
</tr>
<tr>
<td>( r )</td>
<td>0.08 m</td>
<td>( m_s )</td>
<td>10 kg</td>
</tr>
<tr>
<td>( t_c )</td>
<td>0.03 m</td>
<td>( m_c )</td>
<td>0.5 kg</td>
</tr>
<tr>
<td>( V_f )</td>
<td>24 V</td>
<td>( T_s )</td>
<td>10 ms</td>
</tr>
<tr>
<td>( \rho )</td>
<td>12.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ |u_j| \leq u_{\text{max}}, \quad j = 1,2 \]  
\[ |i_j| \leq i_{\text{max}}, \quad j = 1,2 \]  

Since too high velocities in RS may cause WMRs to go over the maximum exit-distance \( D_R \), velocity-constraint is added in RS using a parameter \( \beta \) (velocity scale factor) for requirements of exit-distances (Figure 5).

\[ \frac{|v^n|}{v_{\text{max}}} + \frac{|w^n|}{w_{\text{max}}} \leq \beta, \quad 0 < \beta \leq 2 \]  

![Fig. 5. Added velocity constraints in RS.](image)
where 
\[ \begin{align*}
\nu^\alpha, \nu^\beta &\text{: linear and angular velocity of WMR in RS, respectively,} \\
v^\max &\text{: maximum translational velocity of WMR in steady state when } u^\alpha = u^\beta, \\
w^\max &\text{: maximum angular velocity of WMR in steady state when } u^\alpha = -u^\beta.
\end{align*} \]

\( \beta = 2 \) means that there are no added constraints about velocities in RS when a sufficiently large \( D_k \) is given. However, when \( D_k \) is smaller, we should plan RS with reduced velocities using a certain value of \( \beta > 2 \).

Two constraint steps of Eqs. (5) and (6) can be written by corresponding hyperplane inequalities with inequality matrices \( [D_\alpha, e_\alpha] \) and \( [D_\beta, e_\beta] \), respectively, as
\[ \begin{align*}
C_\alpha &= \{ \mathbf{x} : D_\alpha \mathbf{x} \leq e_\alpha \} \\
C_\beta &= \{ \mathbf{x} : D_\beta \mathbf{x} \leq e_\beta \}
\end{align*} \]
where
\[ \begin{align*}
D_\alpha &= \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
-1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0
\end{pmatrix},
\quad e_\alpha = \begin{pmatrix}
i_{\max} \\
i_{\max} \\
i_{\max}
\end{pmatrix}, \\
D_\beta &= \begin{pmatrix}
0 & 0 & 1/v_{\max} & 1/v_{\max} \\
0 & 0 & 1/v_{\max} & 1/v_{\max} \\
0 & 0 & 1/v_{\max} & 1/v_{\max}
\end{pmatrix},
\quad e_\beta = \begin{pmatrix}
\beta \\
\beta \\
\beta
\end{pmatrix}
\end{align*} \]

If we are concerned with WMR dynamics using \( M \) steps, describe control sequences of \( \mathbf{u}_k \) from zero steps to \((M-1)\) the step as
\[ \mathbf{u}_{\text{seq}} = (\mathbf{u}_0^T \ldots \mathbf{u}_{M-1}^T)^T \]
Then, each state \( \mathbf{x}_k \) can be described by \( \mathbf{u}_{\text{seq}} \) as
\[ \mathbf{x}_k = G^T \mathbf{x}_0 + E_k \mathbf{u}_{\text{seq}}, \quad k = 1, \ldots , M \]
where
\[ E_k = \begin{pmatrix}
0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & \ldots & 0
\end{pmatrix} \]
Since all of \( \{ \mathbf{x}_k, k = 1, \ldots , M \} \) should be within both constraint sets \( C_\alpha \) and \( C_\beta \), corresponding constraints can be expressed with hyperplane inequality descriptions of \( \mathbf{u}_{\text{seq}} \) using Eq. (8) as
\[ \begin{align*}
D_{\alpha,\text{seq}} \mathbf{u}_{\text{seq}} &\leq e_{\alpha,\text{seq}} \\
D_{\beta,\text{seq}} \mathbf{u}_{\text{seq}} &\leq e_{\beta,\text{seq}}
\end{align*} \]
where
\[ \begin{align*}
D_{\alpha,\text{seq}} &= \begin{pmatrix}
D_\alpha & E_1 & \ldots & E_M \\
D_\alpha & E_1 & \ldots & E_M \\
D_\alpha & E_1 & \ldots & E_M \\
D_\alpha & E_1 & \ldots & E_M
\end{pmatrix},
\quad e_{\alpha,\text{seq}} = \begin{pmatrix}
e_\alpha - D_\alpha G \mathbf{x}_0 \\
e_\beta - D_\beta G \mathbf{x}_0
\end{pmatrix}, \\
D_{\beta,\text{seq}} &= \begin{pmatrix}
D_\beta & E_1 & \ldots & E_M \\
D_\beta & E_1 & \ldots & E_M \\
D_\beta & E_1 & \ldots & E_M \\
D_\beta & E_1 & \ldots & E_M
\end{pmatrix},
\quad e_{\beta,\text{seq}} = \begin{pmatrix}
e_\beta - D_\beta G \mathbf{x}_0 \\
e_\beta - D_\beta G \mathbf{x}_0
\end{pmatrix}
\end{align*} \]

\[ \begin{align*}
2.4 \text{ Model reduction in translational motion} \\
\text{In translational motion, WMR dynamics are reduced into simpler ones since } u^\alpha = u^\beta, \quad i_\alpha = i_\beta \quad \text{and } w_k = 0. \quad \text{The reduced models are}
\end{align*} \]
\[ \begin{align*}
\mathbf{x}_{k+1} = G' \mathbf{x}_k + H' u_k, \quad \mathbf{x}_k = [i_k, \mathbf{v}_k]^T, \quad u_k = u_1 = u_2
\end{align*} \]
where
\[ \begin{align*}
G' &= \begin{pmatrix}
G_{11} + G_{12} & G_{13} \\
G_{31} + G_{32} & G_{33}
\end{pmatrix}, \\
H' &= \begin{pmatrix}
H_{11} + H_{12} \\
H_{31} + H_{32}
\end{pmatrix}
\end{align*} \]
\[ \begin{align*}
G_{ij}: \text{element of } i^{th} \text{row and } j^{th} \text{column in } G \\
H_{ij}: \text{element of } i^{th} \text{row and } j^{th} \text{column in } H
\end{align*} \]
Current and voltage constraints are also reduced as
\[ \begin{align*}
|i_k| &\leq i_{\max}, \quad |u_k| \leq u_{\max}
\end{align*} \]
which also can be expressed with hyperplane inequality descriptions of \( \mathbf{u}_{\text{seq}} = [u_0, u_1, \ldots , u_{M-1}]^T \) using similar methods in Section 2.4. Using values in Table I, the reduced model of dynamics is
\[ \begin{align*}
(u_{k+1}) &= \begin{pmatrix}
0.0039 & -0.1887 \\
0.0176 & 0.8609
\end{pmatrix}
\begin{pmatrix}
i_k \\
u_k
\end{pmatrix} + \begin{pmatrix}
4.9611 \\
1.1471
\end{pmatrix}
\end{align*} \]

\[ \begin{align*}
2.5 \text{ Problem statement} \\
\text{Two postures of } P_s = [x_s, y_s, \theta_s] \text{ and } P_f = [x_f, y_f, \theta_f] \text{ are made up in PIC. Assume that WMR is in steady state (x=0) at both postures running in translation motion with its maximum velocity, i.e.,}
\end{align*} \]
\[ \begin{align*}
x_s &= [i_s, i_s, u_s, w_s]^T \\
x_f &= [i_f, i_f, u_f, w_f]^T
\end{align*} \]
where
\[ \begin{align*}
u_s = u_f = u_{\max} \\
w_s = w_f = 0
\end{align*} \]
\[ \begin{align*}
[i_i, i_f]^T &= \frac{L}{F} T_q^{-1} [u_s, w_s]^T \\
[i_f, i_f]^T &= \frac{L}{F} T_q^{-1} [u_f, w_f]^T
\end{align*} \]
Let the state \( \mathbf{z}_k \) defined as
\[ \mathbf{z}_k = [x, y, \theta, u, w]^T \]
Then initial and final configurations composed of postures and velocities are given as
\[ \begin{align*}
\mathbf{z}_s = [x_s, y_s, \theta_s, u_{\max}, 0]^T \\
\mathbf{z}_f = [x_f, y_f, \theta_f, u_{\max}, 0]^T
\end{align*} \]
In addition, both exit-distances in rotational section are assumed to be bounded by \( D_k \) as shown in Figure 4.

**Problem:** Given \( \mathbf{z}_s \) and \( \mathbf{z}_f \), find
\[ \mathbf{u}_{\text{seq}} = \{ \mathbf{u}_k, k = 0, 1, \ldots , k_f \} \]
minimizing the number of steps \( k_f \) s.t.
3. DIRECT VOLTAGE CONTROL ALGORITHMS

In this section, two near minimum-time DVC (DVC-1, DVC-2) algorithms synthesizing path-planning and path-following are proposed for WMRs to satisfy given initial and final postures (positions and angles) and velocities as well as voltage and current constraints. First, we find a control pair composed of the number of steps and control sequence in RS (M, \(x_k\)) which satisfy the required turning angle and also satisfy exit-distance bound using proper \(D\) found with binary search. Then, we find another two control pairs in TSD and TSA, \((M_p, u^p)\) and \((M_v, u^v)\), respectively surrounding the first planned RS control sequence, which satisfy the required translational motion. Control algorithms for each section are established with quadratic programming, and solved using MATLAB optimization toolbox.

Algorithm DVC-2 is an extension of Algorithm DVC-1 in that \(M_p\) is increased from the first value found in Algorithm DVC-1 for possible reduction of total number of steps.

3.1 Rotational section

In rotational section, only angle component of \(\theta\) and velocity components of \([v, w]^T\) in the given configurations are controlled since they keep linear relations with motor voltages while the others don’t (see Eqs. (1) and (2)). Since we don’t know in advance how many number of steps is required to reach final angle and velocities, \(M=1\) is supposed to be the minimum number of steps at first and increased step by step until a solution to the following Problem RS exists. In addition, to minimize overall steps for satisfying whole configurations (including positions), we introduce an object function \(O\) to be minimized with final step \(M\) fixed as:

Minimize \[ O_M = \sum_{k=1}^{M} (v_k - v_{\text{max}})^2 \] (15)

Minimizing \(O_M\) means minimizing the sum of square error from the maximum translational velocity, which makes WMRs move as fast as possible, hence performs near minimum-time control. Minimizing the \(O_M\) is equivalent to minimizing the following quadratic function of \(u\) (see Appendix B):

Minimize \[ u^T Q u + f^T u \] (16)

where

\[
Q = \sum_{k=1}^{M} E_k^T A_k^T A_k E_k \\
= \sum_{k=1}^{M} 2A_k G_k x_k A_k E_k - \sum_{k=1}^{M} 2v_{\text{max}} A_k E_k \\
A_v = (0 \ 0 \ 1 \ 0) \\
u_{\text{seq}} = [u_k, k=0, \ldots, M - 1] \\
\Delta = (u_0^T \ldots u_{M-1}^T)^T
\]

\[
\begin{align*}
\{v_0 = \min(1, \beta) v_{\text{max}}, w_0 = 0, [i_0^T i_0^T] = \frac{i_0^T}{T_q}[v_0 w_0]^T, \text{ if } D_k > 0 \\
\} \\
\{v_0 = w_0 = i_0^T = i_0^T = 0, \text{ if } D_k = 0 \}
\end{align*}
\]

(18)

find a control sequence \(u_{\text{seq}} = [u^R, k=0, \ldots, M - 1] s.t.\) conditions of:

(i) \(\theta_0 = \theta_f - \theta_i\)
(ii) \(w_f = 0\)
(iii) Voltage and current constraints of Eqs. (4) and (9)
(iv) Added velocity constraints of Eq. (10), if \(D_k > 0\)

are satisfied with the object function

\[ u^R \in \{u^R \} = \min_{u} \left( \frac{u^T Q u}{2} + f^T u \right) \]

Minimize \(u_{\text{seq}}^T Q u_{\text{seq}} + f^T u_{\text{seq}}\) to be minimized, where \(Q\) and \(f\) are from Eq. (17).

Angular velocity and displacement in the Problem RS can be written with respect to \(u_{\text{seq}}\) as

\[
w_k = A_v [E_k u_{\text{seq}}^R + G_k x_k] \\
\theta_k = \sum_{i=0}^{k=M-1} w_i T_s
\]

where \(A_v = \begin{bmatrix} 0 \ 0 \ 1 \ 0 \end{bmatrix}\).

We can translate the Problem RS into the following quadratic programming problem of Problem QPRS using MATLAB optimization toolbox:

\[
U = \min_{u} \left( \frac{u^T Q u}{2} + f^T u \right) \text{ subject to } U = Q P(Q, f, D, e, VLB, VUB, n)
\]

\[
\begin{align*}
VLB & \leq u \leq VUB \\\nDu & = e \text{ for the first } n \text{ rows of } D, e \\\nDu & \leq e \text{ the later rows of } D, e
\end{align*}
\]

Problem QPRS (Quadratic Problem for Rotational Section): Given \(M\) and \(x_0\) of Eq. (18), find a control sequence \(u_{\text{seq}} = [u^R, k=0, \ldots, M - 1]\) by solving \(u_{\text{seq}} = Q P(Q, f, D, e, VLB, VUB, n)\) where

\[
\begin{align*}
VUB &= \frac{2M}{[u_{\text{max}} \ldots u_{\text{max}}]^T} \\
VLB &= \frac{2M}{[-u_{\text{max}} \ldots -u_{\text{max}}]^T}
\end{align*}
\]

and if \(D_k > 0\) then
$D = \left( \sum_{k=0}^{M-1} T_k A_n E_k \right)^T (A_n E_M)^T \begin{bmatrix} T_{seq} & D_{seq}^r \end{bmatrix}$

$e = \left( \theta_f - \theta_s - \sum_{k=0}^{M-1} T_k A_n G^r x_0 - A_n G^r x_0 \right)^T$

$n = 2$

if $D_n = 0$ then

$D = \left( \sum_{k=0}^{M-1} T_k A_n E_k \right)^T (A_n E_M)^T \begin{bmatrix} T_{seq} & D_{seq}^r \end{bmatrix}$

$e = \left( \theta_f - \theta_s - \sum_{k=0}^{M-1} T_k A_n G^r x_0 - A_n G^r x_0 \right)^T$

$n = M + 2$

Construct an internal algorithm in RS as

**Algorithm RS:** Given $x_0$.

Step 1: Set $M = 1$.

Step 2: Increase $M$ until a solution to Problem QPRS with $M = M_R$ is found.

3.2 Translational Section

After planning in RS, two translational motions before and after RS should be planned to cover both remaining distances $d_D$ and $d_A$ for satisfaction of positions’ conditions (Figure 6). Given the required distances of $d_D$ and $d_A$, we can make each simpler control both in TSD and in TSA using reduced model of Eq. (11). Basic procedures are established as follows:

**Algorithm TSD:** Given $d_D$ and $x_0 = x_{s_D}$.

Step 1: Set $M_D = 1$.

Step 2: Increase $M_D$ until a control sequence $U_{seq}^D = \{u_k^D, k = 1, \ldots, M_D \}$ is found s.t. conditions of

(i) $\sum_{i=0}^{M_D-1} u_i T_s = d_D$

(ii) $w_M = 0$

(iii) $u_{M_D} = u_{max}$

(iv) current and voltage constraint (Eq. (12)) are satisfied.

**Algorithm TSA:** Given $d_A$ and $x_0 = x_{M_R}$ from RS.

Step 1: Set $M_A = 1$ and increase $M_A$ until a control sequence $U_{seq}^A = \{u_k^A, k = 1, \ldots, M_A \}$ is found s.t. conditions of

(i) $\sum_{i=0}^{M_A-1} u_i T_s = d_A$

(ii) $w_{M_A} = 0$

(iii) $u_{M_A} = u_{max}$

(iv) current and voltage constraint (Eq. (12)) are satisfied.

Both Algorithm TSD and Algorithm TSA can be easily solved by using QP function of MATLAB as in Problem RS with arbitrary $Q$ and $f$.

3.3 DVC algorithms

Two DVC algorithms considering all RS, TSD, and TSA are proposed. The first is a simpler one for simple calculations and the second step is made by extension of the first for reducing the number of steps. Both algorithms are established as follows.

**Algorithm DVC-1:** Set $\beta = 2$, $\beta_2 = 2$, $\beta_3 = 0$ and do following steps.

Step 1: Find the optional $\beta$ using the binary search with the following sub-steps.

Step 1–1: Find $(M_R, U_{seq}^R)$ using Algorithm RS and calculate $d_{R_1}$ and $d_{R_2}$ defined as in Fig. 8.

**Fig. 6.** Binary search for $\beta$.

**Fig. 7.** Example for binary search for $\beta$.  

**Wheeled robots**
Fig. 8. Step-increment of $M_R$.

Fig. 9. Simulation with $D_R = 0.30$, $\theta_1 = 90^\circ \rightarrow \beta = 2.0$, $k_f = 178$.

Fig. 10. Simulation with $D_R = 0.15$, $\theta_1 = 90^\circ \rightarrow \beta = 0.531$, $k_f = 200$.

Fig. 11. Simulation with $D_R = 0.15$, $\theta_1 = 135^\circ \rightarrow \beta = 0.469$, $k_f = 288$.

Fig. 12. Simulation with $D_R = 0.15$, $\theta_1 = 45^\circ \rightarrow \beta = 0.531$, $k_f = 170$. 
Step 1–2: Calculate \( d\beta \) and update \( \beta_1 \) (or \( \beta_u \)) as

\[
\begin{align*}
\Delta \beta &= 0.5(\beta_T - \beta_0), \quad \beta_1 = \beta_0, \text{ if } d_\beta < D_R, \text{ }d_\beta > D_R \\
\Delta \beta &= 0.5(\beta_T - \beta_0), \quad \beta_1 = \beta_0, \text{ otherwise }
\end{align*}
\]

Step 1–3: If \( |\Delta \beta| > \beta_{\text{tol}} \) then update \( \beta \) as \( \beta \leftarrow \beta + \Delta \beta \) and go to Step 1–1, otherwise go to Step 2.

Step 2: Find \( \langle M_{D'}, u_{\text{seq}}^D \rangle \) using Algorithm TSD with \( d_{D'} \).

Step 3: Find \( \langle M_{A}, u_{\text{seq}}^A \rangle \) using Algorithm TSA with \( d_A \).

Flow chart of the binary search for optimal \( \beta \) is drawn in Figure 6, and Figure 7 shows a graphical description of an example. Average number of search for optimal \( \beta \) is reduced to \( \log_2(2/\beta_{\text{tol}}) \) in the method of binary search, while \( 2\beta - 1 \) in a method of simple decrement (or increment).

**Algorithm DVC-2**: Set \( \beta = 2, \beta_1 = 2, \beta_u = 0 \) and do following steps.

Step 1: Do Algorithm DVC-1.

Step 2: Let \( M'_{A} = M_{A} + 1 \).

Step 3: Solve the Problem QPRS for \( M'_{A} \) and calculate \( d'_{A} \) and \( d'_{D} \). If both \( d'_{A} < D_A \) and \( d'_{D} < D_D \) then go to Step 4, otherwise stop.

Step 4: Find \( \langle M'_{D'}, u_{\text{seq}}^{D'} \rangle \) using Algorithm TSD with \( d'_{D'} \) and also find \( \langle M'_{A}, u_{\text{seq}}^{A} \rangle \) using Algorithm TSA with \( d'_{A} \) (Figure 8).

If

\[
M_D + M_R + M_A \geq (M'_D + M'_R + M'_A)
\]

then,

\[
\langle M'_{D'}, u_{\text{seq}}^{D'} \rangle \leftarrow \langle M'_{D'}, u_{\text{seq}}^{D'} \rangle,
\langle M'_A, u_{\text{seq}}^{A} \rangle \leftarrow \langle M'_A, u_{\text{seq}}^{A} \rangle,
\langle M'_A, u_{\text{seq}}^{A} \rangle \leftarrow \langle M'_A, u_{\text{seq}}^{A} \rangle
\]

Step 5: Let \( M'_D = M'_D + 1 \) and go to Step 3.

Especially when \( D_R = 0 \), all the values of \( v_{ij} \) should be zero in RS. Hence, in that situation, Algorithm DVC-1 should be applied without Steps 1–2 and 1–3, since there is no use of increasing \( M_R \) just after the first solution of RS is found. Finally, the overall control sequence is

\[
\mathbf{u}_{\text{seq}} = \{ u_{kj}, k = 0, 1, \ldots, k_f - 1, k_f = M_D + M_R + M_A \}
\]

\[
= \{ u_{\text{seq}}^{A}, u_{\text{seq}}^{A}, u_{\text{seq}}^{A} \}
\]

It is very important to search, at first, two parameters \( \beta \) and \( M_R \) in RS since both TSD and TSA can be planned only after

Table III. Summary of computation time and position error when 90° turning.

<table>
<thead>
<tr>
<th>( T'_i ) (ms)</th>
<th>( \text{comp. time (h)} )</th>
<th>( e_{xy} ) (cm²)</th>
<th>( \text{comp. time (h)} )</th>
<th>( e_{xy} ) (cm²)</th>
<th>( e_{xy}/e_{xy} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.45927</td>
<td>0.8564</td>
<td>0.34577</td>
<td>0.8601</td>
<td>1.004</td>
</tr>
<tr>
<td>10</td>
<td>2.02133</td>
<td>0.1574</td>
<td>0.76562</td>
<td>0.1806</td>
<td>1.147</td>
</tr>
<tr>
<td>5</td>
<td>26.74665</td>
<td>0.0162</td>
<td>7.29261</td>
<td>0.1101</td>
<td>6.796</td>
</tr>
</tbody>
</table>
those parameters have been found. In Algorithm DVC-1, one 1-dimensional search for \( \beta \) is performed, while two 1-dimensional searches in Algorithm DVC-2 - the one is for \( \beta \) and the other is for \( M_{r} \). Surely it is possible to establish a full search (2-dimensional search) for both \( \beta \) and \( M_{r} \), which will make the best performance of the three methods of search, but will be highly time-consuming. Algorithm DVC-2 is a better choice since it is simpler but makes almost the same performance (number of steps \( k_{f} \)) as 2-dimensional search, which will be shown in Section 4 by comparing the performances about number of steps \( k_{f} \). When \( D_{k} \) is small, the simplest one, Algorithm DVC-1 also may be good choice, which will also be discussed in Section 4.

4. SIMULATION RESULTS

In this section, various simulations are performed to show performances of the proposed algorithms (DVC-1, DVC-2) with use of parameters in Table I and \( \beta_{s} = 0.01 \). Initial postures are all set to \([0m 0m 0^\circ]^T\) without loss of generality. Three different final postures, \([0.50m 0.50m 90^\circ]^T\), \([0.25m 0.45m 135^\circ]^T\), and \([0.85m 0.35m 45^\circ]^T\), are used and correspond to Figures 9 to 10, Figure 11, and Figure 12 respectively. All figures are resulted from Algorithm DVC-2.

When the value of \( D_{k} \) is sufficiently large like Figure 9, the maximum value of \( \beta = 2 \) is allowed by binary search satisfying the exit-distance requirement. On the other hand, Figure 10 shows that an appropriate smaller value of \( \beta = 0.531 \) is established since reduced \( D_{k} \) is given, which forces more added constraints into velocities. As we investigate subplots of \((i^1, i^2), (u^1, u^2), \) and \((v, w) \) in each figure, we can see that the planned paths go along with at least one extreme part of constraints all the time, except transient states between TSD and RS and between RS and TSA. It means that WMR tries to use all driving ability as possible as it can for fast moving. Figures 11 and 12 show similar results too.

Binary search for \( \beta \) is based on the conjecture that possibly larger \( \beta \) can do plan in a smaller number of steps. This conjecture is justified by comparison with full 2-dimensional search for both \( \beta \) and \( M_{r} \), which is summarized in Table II. Algorithm DVC-2 shows the same performance, in the number of steps \( k_{f} \), as full 2-dimensional search even though there are some differences in the pair of \( (\beta, M_{r}) \), while Algorithm DVC-1 degrades by 1.78% - the cases of smaller \( D_{k} \) do not affect this degradation at all. With these results, it can be said that Algorithm DVC-2 is a better choice since it makes almost the same performance in spite of simpler process than full 2-dimensional search. When \( D_{k} \) is small, the simplest one, Algorithm DVC-1 may be good choice also.

We did not neglect the armature inductance \( L_{a} \) of each motor in order to consider exact dynamics, even though we used \( T_{s} = 10m \) s in major simulations which is large relative to time constant \( L_{a}/R_{e} = 0.7m \). We summarized computation times and errors for various sampling times in Table III, where the computation time is based on Algorithm DVC-2, and \( e_{xy} \) means \((x_{f} - x_{k})^2 + (y_{f} - y_{k})^2 \). As the sampling time becomes smaller, neglecting \( L_{a} \) requires relatively shorter computation time, but the error \( e_{xy} \) becomes larger. Hence, including \( L_{a} \) gives more accurate results, especially for smaller sampling time.

5. CONCLUSION

WMR systems have nonholonomic and nonlinear properties. In addition, they have motor armature current constraint and battery voltage constraint in practice. If we make a PP first and develop a PF based on the PP, then the PF procedure might be unable to track the planned path due to those constraints. For this reason, we should consider those constraints when making a PP. Consequently, we should include PF at the same time.

In this paper, we have considered real constraints on current and voltage in WMRs. Near minimum-time control DVC algorithms (DVC-1 and DVC-2) synthesizing PP and PF have been proposed, which make it possible to control voltage directly. Control steps are planned in near minimum-time by minimizing the object function \( D_{k} \). Bounded exit-distance is considered by introducing a parameter of \( \beta \) in rotational section as well. Algorithm DVC-2 is a better choice since it makes almost the same performance (number of steps \( k_{f} \)) in spite of simpler process than full 2-dimensional search. When \( D_{k} \) is small, the simplest one, Algorithm DVC-1 may be a good choice also. Various simulations for different cases are performed to show the validity of proposed algorithms.

For a further work, we are going to extend the proposed algorithm to the CC, which is not far from here since CC can be made up with two appropriate PC.

References

motor and left motors respectively. Dynamic relation between angular velocity and motor current considering inertia and viscous friction becomes

\[
(S^T MS) \frac{dw}{dt} + F_w w = K_p \rho \dot{w}
\]  

(22)

where \(F_w\) is viscous friction coefficient and \(M\) is inertia matrix of mobile robot and \(S\) is null space of nonholonomic constraint matrix.

\[S = \begin{pmatrix}
    cb \cos \phi & cb \cos \phi \\
    cb \sin \phi & cb \sin \phi \\
    1 & 0 \\
    0 & 1
\end{pmatrix}
\]

\[M = \begin{pmatrix}
    m & 0 & 0 \\
    0 & m & 0 \\
    0 & 0 & I_c^2 + I_w - I_c^2 \\
    0 & -I_c^2 & I_c^2 + I_w
\end{pmatrix}
\]

\[S^T MS = \begin{pmatrix}
    mc^2 b^2 + I_c^2 + I_w & mc^2 b^2 - I_c^2 \\
    mc^2 b^2 - I_c^2 & mc^2 b^2 + I_c^2 + I_w
\end{pmatrix}
\]

where

\[I_c = \frac{m(4b^2 + r^2)}{12}, \quad I_w = m_n r^2/2, \quad I_m = m_n (3r^2 + t_n^2),
\]

\[I = I_c + 2m_n b^2 + 2I_m, \quad m = m_n + 2m_u
\]

For details, see reference [5] (WMR is assumed to be symmetric in mass, i.e., \(d = 0\)).

Define a state vector as

\[x = (i^1 \ i^2 \ v \ w)^T
\]

where linear velocity of WMR \(v\) and angular velocity of WMR \(w\) are related with \(i^1\) and \(i^2\) as

\[
\begin{pmatrix}
    v \\
    w
\end{pmatrix} = T_q \begin{pmatrix}
    i^1 \\
    i^2
\end{pmatrix}, \quad T_q = \begin{pmatrix}
    \dot{\gamma} \\
    \dot{\phi}
\end{pmatrix}
\]

Then, we get a continuous-time state-space dynamic model

\[
\dot{x} = Ax = Bu
\]  

(23)

where
**B EQUIVALENT QUADRATIC FORM OF $O_M$**

Rewrite the object function $O_M$ as

$$O_M = \sum_{k=1}^{M} (t_k - v_{\text{max}})^2$$  \hspace{1cm} (24)

$$= \sum_{k=1}^{M} (t_k)^2 - 2v_{\text{max}} \sum_{k=1}^{M} t_k + \sum_{k=1}^{M} (v_{\text{max}})^2$$

Since

$$u_k = A_v x_k$$

$$= A_v E_k u_{\text{seq}} + A_v G^k x_0$$  \hspace{1cm} (25)

where $A_v = [0 \ 0 \ 1 \ 0]$ and $E_k$ is from Eq. (8), we can say that

$$\sum u_k = \sum (A_v E_k u_{\text{seq}} + A_v G^k x_0)$$

$$= \sum A_v E_k u_{\text{seq}} + \sum A_v G^k x_0$$

$$\sum (u_k)^2 = \sum (A_v E_k u_{\text{seq}} + A_v G^k x_0)^2$$  \hspace{1cm} (26)

$$= \sum u_{\text{seq}}^T E_k^T A_v^T A_v E_k u_{\text{seq}}$$

$$+ \sum 2A_v G^k x_0 A_v E_k u_{\text{seq}} + \sum (A_v G^k x_0)^2$$

Therefore,

$$O_M = u_{\text{seq}}^T Q u_{\text{seq}} + f^T u_{\text{seq}} + g$$  \hspace{1cm} (27)

where

$$Q = \sum_{k=1}^{M} E_k^T A_v^T A_v E_k$$

$$f^T = \sum_{k=1}^{M} 2A_v G^k x_0 A_v E_k - \sum_{k=1}^{M} 2v_{\text{max}} A_v E_k$$  \hspace{1cm} (28)

$$g = \sum_{k=1}^{M} (A_v G^k x_0)^2 - \sum_{k=1}^{M} 2v_{\text{max}} A_v G^k x_0$$

$$+ \sum_{k=1}^{M} (v_{\text{max}})^2$$

Since $g$ of Eq. (28) is a constant value, minimizing the $O_M$ is equivalent to minimizing

$$u_{\text{seq}}^T Q u_{\text{seq}} + f^T u_{\text{seq}}$$  \hspace{1cm} (29)