1. Introduction

The authors’ research group has discussed the Manual Motion Control (MMC). MMC problems are seen in the conveyance of a large amount of products in factories and stores. One of the most successful example of MMC is power assist which can be used to reduce the muscle load on operators\(^5\). For example, Colgate proposed “Cobot” for making the virtual guide to assist the conveyance\(^2\). We studied on the power assist controller design as an application of power assist system to suppress vibration\(^1\) and prevent falling\(^4\) for carried objects.

Simultaneously, the work in factories by using power assist systems needs higher efficiency. Then, such a mechanical system also needs an automatic operation mode taking the collision avoidance into account when respond to call or necessities of charging battery. Furthermore, fully automatic robot has been spread all over the world. For successful example, irobot invented the automatic cleaner “Roomba”\(^5\). The system should avoid collision with obstacles to ensure the safety in automatic mode. If a mechanical system itself can determine where to go to charge battery automatically, it improves efficiency of work in the production site. Therefore, this study proposes a control method to allow the mechanical system to avoid collision with obstacles in automatic mode as mentioned above.

In this paper, we deal with the problem as an example of mechanical system collision avoiding control methods. This study adopts the self-moving cart as a controlled object example which is described in Fig. 1. The cart starts to decelerate at the moment it detects the obstacle to avoid collision. This paper applies Updating Final-State Control (UFSC)\(^7\) to the problem as described above in order to generate appropriate motion trajectories for collision avoidance. We compare UFSC with Model Predictive Control (MPC)\(^7\) which is widely applied as a control method for obstacle collision avoidance problems and consider the superiority of the UFSC.

2. Model Predictive Control

MPC is one of control methods in which the current control action is obtained by solving the constrained optimal control problem online at each sample until the specified future time. The control input is determined by solving the optimal control problem at each sample and the optimization yields an optimal control sequence. The only first control input in the sequence is applied to the controlled system. This method is called Receding Horizon (RH) control because the prediction horizon keeps being shifted forward. The scheme of MPC is shown in Fig. 2.

MPC was often used for chemical plants that are slow in reaction due to the larger computational load. In recent years, this method has become a typical control method for collision avoidance because computers have been developed and many effective algorithms have been proposed actively.

3. Updating Final-State Control

3.1 Final-State Control (FSC)

The self-moving cart should decelerate gradually for obstacle collision avoidance to prevent a bad influence on the conveyance. Therefore, we try to apply Updating FSC based on FSC\(^5\). This section mentions about FSC before applying UFSC. FSC is the control method which transits a dynamic system to a specified state in a specified time. Generally, a single-input liner time-invariant discrete-time system is represented as the following equation:

\[
\dot{x}(k + 1) = Ax(k) + Bu(k) \tag{1}
\]

where, \(k\), \(x(k)\) and \(u(k)\) mean the number of samples, the state vector of the controlled object and the control input, respectively. Here, let us consider the external input \(u(k)\) that transfers the state \(x(k)\) from the initial state \(x(0)\) to the final state \(x(N)\) within \(N\) samples. Such an input must meet the following boundary conditions:

\[
X = \Sigma U \tag{2}
\]

where,

\[
X = x(N) - A^N x(0) \tag{3}
\]

\[
\Sigma = [A^{N-1} b \ A^{N-2} b \ ... \ b] \tag{4}
\]

\[
U = [u(0) \ u(1) \ ... \ u(N - 1)]^T. \tag{5}
\]

In general, the solution of the above equation is not unique. So, criterion function \(J\) is set as follows:

\[
J = U^T QU. \quad Q > 0 \tag{6}
\]

\[
U = Q^{-1} \Sigma^T (\Sigma Q^{-1} \Sigma^T)^{-1} X. \tag{7}
\]

At each sample, the computed input is applied in order and the ideal final states are realized at the terminal sample.

3.2 Updating Final-State Control (UFSC)

In order to permit FSC to deal with the time-varying ideal final states, UFSC updates the initial and final states in Eq. (7) at each sample. Because of updating, UFSC cannot satisfy minimizing the criterion function Eq. (6). However, it inherits other characteristics from FSC. At each sample, initial and objective final states are updated as the following equation at \(k\)-sample:

---

**Key Words:** Motion Control, Collision Avoidance, Final-State Control, Model Predictive Control

---

Fig. 1 Controlled object example

![Fig. 1 Controlled object example](image1)

Fig. 2 Scheme of MPC

![Fig. 2 Scheme of MPC](image2)
Dynamics of a cart model:

\[ x(k) = [x(k) \; \dot{x}(k)]^T \]  
\[ x(N) = [\dot{x}(k) \cdot (N-k) \cdot t_{samp} \; 0]^T \]  

where \( N \) is the number of samples for UFSC.

Then, UFSC input at \( k \)-sample is given by the first row of the following equation:

\[ U = Q^{-1} \Sigma_k^{-1} (\Sigma_k Q^{-1} \Sigma_k^T)^{-1} [x(N) - A^{(N-k)}x(k)] \]  

The control input is determined by solving Eq. (10). This input is applied to the controlled system at each sample and the objective final state is realized in \( N \) samples. In addition, large computational load is required because of the existence of inverse matrix. Therefore, it is difficult to apply UFSC in many actual cases. However, it is possible when a simple controlled system model is treated\(^6\). A cart as shown in Fig. 3 can be modeled as a double-integrator-system and corresponds to such a simple controlled system.

4. Simulation Study

4.1 MPC

In simulations, the cart moves forward automatically for 6 s and control horizon is set to 1 s between 6 s and 10 s. After the cart detects an obstacle at 6 s, the cart starts to decelerate in order to avoid collision. The final objective displacement is the 20 mm point this side of the obstacle. This paper follows the algorithm of MPC in Ref. (7). The control sequence is recalculated at each sample. The sampling time is set to 1.0 s in this simulation because of large computational load. Trajectory generation results by MPC are shown in Fig. 4. As indicated in Fig. 4, right figure shows that the displacement can stop in front of the obstacle. Red dotted line corresponds to the position of the obstacle. Overshoot occurs in after 6 s. Left figure shows velocity reference. It is decelerated to zero. The computation time for realizing these reference generations by MPC requires 1.99 s by using a PC (OS: MS-Windows 7; CPU: Intel Core i5 3.2 GHz; RAM: 8.0 GB) and MATLAB Ver. R2015a.

4.2 UFSC

The validation of UFSC for obstacle collision avoidance is validated by simulations in this subsection. Once the objective final state is achieved in any samples, the objective initial and final states are updated. The control input is recalculated in each 1 ms sample to realize ideal motion trajectory. Trajectory generation results by UFSC are indicated in Fig. 5. As shown in Fig. 5, right figure shows that the displacement can stop in front of the obstacle and red dotted line of the obstacle. Left figure indicates that the velocity reference is decelerated to zero more smoothly. The computation time for realizing these reference generations by UFSC requires 0.67 s by using the same condition as that of the previous subsection to solve the UFSC problem. Therefore, it is verified that UFSC can reduce the computational load effectively in comparison with the use of MPC although the sampling step of UFSC is 1000 times larger than that of MPC.

5. Conclusion

This paper discussed a control problem of a self-moving cart as an example. Concretely, we considered the validity of the UFSC which a part of the authors proposed in Ref. (6) by compared with MPC. The validity of UFSC should be confirmed experimentally as a future subject.

Acknowledgment

This work was supported in part by the Tatematsu Foundation. The authors thank Prof. Yoji Yamada from Nagoya University for his valuable advice.

References