Proceedings of the 2014 International Conference on
Advanced Mechatronic Systems, Kumamoto, Japan, August 10-12, 2014

Particle Filter-Model Predictive Control of Quadcopters

Kento SHIMADA
Graduate School of Engineering,
Kyushu Institute of Technology
1-1 Tobata, Kitakyushu, Fukuoka, Japan

Takeshi NISHIDA
Department of Mechanical and Control Engineering,
Faculty of Engineering, Kyushu Institute of Technology
1-1 Tobata, Kitakyushu, Fukuoka, Japan
Email: nishida@cntl.kyutech.ac.jp

***Abstract*—In recent years, extensive research has been conducted on the design and control of unmanned aerial vehicles having multiple propellers, called multicopters (especially quadcopters, which has four propellers). In this paper, we propose a method for applying particle filter-model predictive control that can take disturbance and measurement noise into consideration explicitly to the flight control of a quadcopter. Moreover, the validity of the proposed method is verified by simulation.**

***Keywords*—model predictive control, particle filter, quadcopter.**

I. INTRODUCTION

In recent years, extensive research has been conducted on the design [1] and control [2], [3] of unmanned aerial vehicles having multiple propellers, called multicopters. Especially, research on quadcopters, which have four propellers, has been extensive. A quadcopter is classified as a vertical takeoff and landing vehicle and has a cross body frame lifted and propelled by four rotors. A quadcopter uses two sets of identical fixed pitch propellers, and its motion is controlled by altering the rotation rate of one or more propellers, thereby changing its torque load and thrust/lift characteristics. Generally, a quadcopter carries a three-axis gyroscope sensor, geomagnetism sensor, and three-axis accelerometer, which are used for the stabilization of the flight attitude.

The system model of a quadcopter is represented as a nonlinear system with input saturation, and wind disturbance strongly affects its motion. Moreover, the control input becomes excessive or is saturated when a simple control system such as a position error feedback control system is used [4]. Therefore, in this research, the MPC (model predictive control) method, which is effective in solving such problems, is introduced. However, in conventional MPC, because the prediction input to a reference trajectory is computed definitely, the probable element of a prediction horizon cannot be taken into consideration. Thus, in this paper, PF-MPC (particle filter-model predictive control) [5], [6] that considers the above-mentioned uncertainly is introduced for the predictive control of a quadcopter. This method improves the stability of the quadcopter system and prevents the generation of an excessive control input.

II. DYNAMICS OF QUADCOPTER

A. Continuous time model

The relation between the inertial system Σ_O and the quadcopter coordinate system Σ_r is shown in Fig. 1. Here, the upper part of the z axis in Σ_r is positive. The posture of Σ_r relative to Σ_O is represented by rotation transformation as follows:

$$\boldsymbol{R}_r^O(\theta_z, \theta_y, \theta_x) = \boldsymbol{R}_z(\theta_z)\boldsymbol{R}_y(\theta_y)\boldsymbol{R}_x(\theta_x) \tag{1}$$

where θ_y, θ_r, and θ_p represent the yaw, roll, and pitch angles, respectively. Moreover, $\boldsymbol{R}_z(\theta_z)$, $\boldsymbol{R}_y(\theta_y)$, and $\boldsymbol{R}_x(\theta_x)$ represent the rotation transformation around the z axis, y axis, and x axis, respectively. The shift vector from the origin of Σ_O to the origin of Σ_r is represented as $\boldsymbol{z}(t) \triangleq [x(t)\ y(t)\ z(t)]^T$. The quadcopter has four propellers, the angular velocity of the i-th propeller is expressed as $s_i(t)$, and the perpendicular upward thrust can be expressed as follows:

$$f_i(t) = bs_i^2(t) \qquad (i = 1,\ 2,\ 3,\ 4) \tag{2}$$

where b is the thrust constant. The dynamic model of the quadcopter in Σ_O is represented as follows:

$$m\ddot{\boldsymbol{z}}(t) = \boldsymbol{R}_r^O \begin{bmatrix} 0 \\ 0 \\ F(t) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - \boldsymbol{K} \begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} \tag{3}$$

where g is the gravitational acceleration, m is the mass of the quadcopter, and $F(t) = \sum f_i(t)$ is the sum of the perpendicular upward thrusts. The first term on the right-hand side of Eq. (3) represents the sum of the thrusts in Σ_O. The second term represents the force generated by gravity, and the third term represents about air resistance. $\boldsymbol{K} \in \mathbb{R}^{3\times1}$ is the proportionality constant vector, and $\boldsymbol{v}(t) = [v_x(t)\ v_y(t)\ v_z(t)]^T \in \Sigma_O$ is

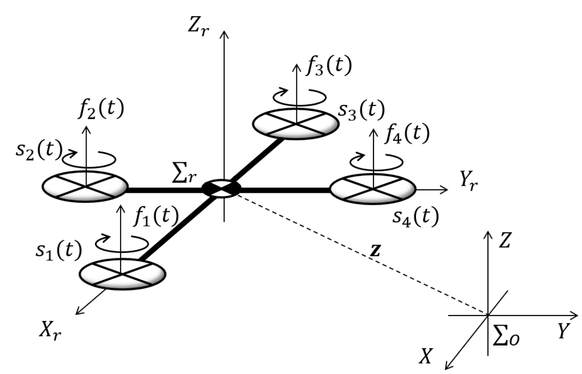


Fig. 1. Relation between the inertial system and the quadcopter coordinate system.

the velocity of the quadcopter.

Next, on the basis of Euler's equation, the rotation dynamics are given as follows:

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}(t) = -\boldsymbol{\omega}(t) \times \boldsymbol{J}\boldsymbol{\omega}(t) + \boldsymbol{T}(t) - \boldsymbol{K}_e\boldsymbol{\omega}(t) \tag{4}$$

where $\boldsymbol{J} \in \mathbb{R}^{3\times3}$ is an inertia matrix, $\boldsymbol{\omega}(t)$ is the angular velocity vector, and $\boldsymbol{T}(t) \triangleq [\tau_{x_r}(t)\ \tau_{y_r}(t)\ \tau_{z_r}(t)]^T$ is the running torque. The fourth term represents force generated by air resistance and $\boldsymbol{K}_e = [K_{e_x}\ K_{e_y}\ K_{e_z}]^T$ is a proportionality constant vector. The inertia matrix is a positive-definite symmetric matrix, and it can be diagonalized by choosing a suitable orthogonal coordinate system as follows:

$$\boldsymbol{J} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{5}$$

Using this, Eq. (4) can be expressed as follows:

$$\begin{aligned} I_{xx}\dot{\omega_1}(t) + (I_{zz} - I_{yy})\,\omega_2(t)\omega_3(t) &= \tau_x(t) - K_{e_x}\omega_1 \\ I_{yy}\dot{\omega_2}(t) + (I_{xx} - I_{zz})\,\omega_3(t)\omega_1(t) &= \tau_y(t) - K_{e_y}\omega_2 \\ I_{zz}\dot{\omega_3}(t) + (I_{yy} - I_{xx})\,\omega_1(t)\omega_2(t) &= \tau_z(t) - K_{e_z}\omega_3 \end{aligned}$$

where $\dot{\boldsymbol{\omega}}(t) \triangleq [\dot{\omega}_1(t)\ \dot{\omega}_2(t)\ \dot{\omega}_3(t)]^T$ and $\boldsymbol{\omega}(t) \triangleq [\omega_1(t)\ \omega_2(t)\ \omega_3(t)]^T$.

A dynamic model of the quadcopter is derived from Eq. (3) and Eq. (4).

$$\begin{bmatrix} F(t) \\ \boldsymbol{T}(t) \end{bmatrix} = \boldsymbol{A}\boldsymbol{s}^2(t) \tag{6}$$

where

$$\boldsymbol{A} \triangleq \begin{bmatrix} b & b & b & b \\ 0 & -db & 0 & db \\ -db & 0 & db & 0 \\ -k & k & -k & k \end{bmatrix} \tag{7}$$

$$\boldsymbol{s}^2(t) \triangleq \begin{bmatrix} s_1^2(t) & s_2^2(t) & s_3^2(t) & s_4^2(t) \end{bmatrix}^T \tag{8}$$

and $\boldsymbol{s}(t)$ is a function of the rotational frequency of rotors. If $b,\ k,\ d > 0$, $\boldsymbol{A}$ is a full-rank matrix, and the rotational frequency of rotors that would give the specified thrust and moment to the body is given as follows:

$$\boldsymbol{s}^2(t) = \boldsymbol{A}^{-1}\begin{bmatrix} F(t) & \tau_{x_r}(t) & \tau_{y_r}(t) & \tau_{z_r}(t) \end{bmatrix}^T \tag{9}$$

B. Discrete time state equation

An actual quadcopter system is driven as a discrete time system, and the predictive control method targets discrete time systems. Here, the abovementioned dynamic model is re-described as a discrete time system.

First, the velocity, acceleration, and angular velocity, angular acceleration of the center of gravity of the quadcopter are expressed as follows:

$$\ddot{\boldsymbol{z}}(t) \triangleq \dot{\boldsymbol{v}}(t) \triangleq \boldsymbol{a}(t) \tag{10}$$

$$\ddot{\boldsymbol{\theta}}(t) \triangleq \dot{\boldsymbol{\omega}}(t) \triangleq \boldsymbol{\alpha}(t) \tag{11}$$

Taylor's expansion is applied to these expressions, and the terms that express the available sensor signals are used for the construction of a state equation as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{\Phi}\boldsymbol{x}_k + \boldsymbol{\Gamma}\boldsymbol{u}_k \tag{12}$$

where

$$\boldsymbol{x}_k \triangleq \begin{bmatrix} \boldsymbol{z}_k \\ \boldsymbol{v}_k \\ \boldsymbol{\theta}_k \\ \boldsymbol{\omega}_k \end{bmatrix},\quad \boldsymbol{\Phi} \triangleq \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{\Delta}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 & \boldsymbol{\Delta}_3 \\ \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{0}_3 & \boldsymbol{I}_3 \end{bmatrix} \tag{13}$$

$$\boldsymbol{\Gamma} \triangleq \begin{bmatrix} \frac{\boldsymbol{\Delta}_3^2}{2} & \boldsymbol{0}_3 \\ \boldsymbol{\Delta}_3 & \boldsymbol{0} \\ \boldsymbol{0}_3 & \frac{\boldsymbol{\Delta}_3^2}{2} \\ \boldsymbol{0}_3 & \boldsymbol{\Delta}_3 \end{bmatrix},\quad \boldsymbol{u}_k \triangleq \begin{bmatrix} \boldsymbol{a}_k \\ \boldsymbol{\alpha}_k \end{bmatrix} \tag{14}$$

k is the discrete time, Δ is the sampling time, $\boldsymbol{a}_k \in \Sigma_O$ is the acceleration, $\boldsymbol{\omega}_k$ is the angular velocity, $\boldsymbol{\alpha}_k$ is the angular acceleration, $\boldsymbol{I}_3 \in \mathbb{R}^{3\times3}$ is an identity matrix, $\boldsymbol{0}_3 \in \mathbb{R}^{3\times3}$ is a zero matrix, $\boldsymbol{\Delta}_3 = \text{diag}[\Delta, \Delta, \Delta]$, and $\boldsymbol{\Delta}_3^2 = \text{diag}[\Delta^2, \Delta^2, \Delta^2]$. Furthermore, the equation can be expressed as follows:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k \\ &+ \begin{bmatrix} \boldsymbol{\Delta}_3\boldsymbol{v}_k + \frac{\boldsymbol{\Delta}_3^2}{2}\left\{\boldsymbol{R}_r^O(\boldsymbol{\theta}_k)\boldsymbol{F}_k(\boldsymbol{s}_k) - \boldsymbol{g} - \boldsymbol{K}\frac{\boldsymbol{v}_k}{m}\right\} \\ \boldsymbol{\Delta}_3\left\{\boldsymbol{R}_r^O(\boldsymbol{\theta}_k)\boldsymbol{F}_k(\boldsymbol{s}_k) - \boldsymbol{g} - \boldsymbol{K}\frac{\boldsymbol{v}_k}{m}\right\} \\ \boldsymbol{\Delta}_3\boldsymbol{\omega}_k + \frac{\boldsymbol{\Delta}_3^2}{2}\left\{\boldsymbol{J}^{-1}(-\boldsymbol{\omega}_k \times \boldsymbol{J}\boldsymbol{\omega}_k + \boldsymbol{T}_k(\boldsymbol{s}_k) - \boldsymbol{K}_e\boldsymbol{\omega}_k)\right\} \\ \boldsymbol{\Delta}_3\left\{\boldsymbol{J}^{-1}(-\boldsymbol{\omega}_k \times \boldsymbol{J}\boldsymbol{\omega}_k + \boldsymbol{T}_k(\boldsymbol{s}_k) - \boldsymbol{K}_e\boldsymbol{\omega}_k)\right\} \end{bmatrix} \end{aligned}$$

where $\boldsymbol{F}_k(\boldsymbol{s}_k) = [0\ 0\ F(t)/m]^T$ and $\boldsymbol{g} = [0\ 0\ g]^T$. That is, this state equation is nonlinear.

Therefore, the nonlinear discrete time state equation and measurement equation are represented as follows:

$$\boldsymbol{x}_k = \boldsymbol{\phi}\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{\xi}_{k-1}\right) \tag{15}$$

$$\boldsymbol{y}_k = \boldsymbol{\psi}\left(\boldsymbol{x}_k, \boldsymbol{\eta}_k\right) \tag{16}$$

where $\boldsymbol{\phi}$ is the state translation function, $\boldsymbol{\xi}_k$ is the system noise, $\boldsymbol{\psi}$ is the measurement function, and $\boldsymbol{\eta}_k$ is the measurement noise.

III. Particle filter-model predictive control

Conventional MPC optimizes a finite time horizon. The main advantage of MPC is that it allows the current timeslot to be optimized, while taking future timeslots into account (Fig. 2). However, in MPC, the effect of the measurement noise and indefinite disturbance of a quadcopter control system cannot be taken into consideration. Therefore, we introduce PF-MPC [6], which can consider such effects in the predictive control of a quadcopter.

In PF-MPC, plural predictions $\bar{\boldsymbol{x}}_{k:k+N_P}$ up to N_P steps from the present time are generated in consideration of the system and measurement noise, and the input is optimized by a cost function based on a deviation target trajectory and $\boldsymbol{x}_{k:k+N_P}$. A concrete algorithm is shown below.

(Particle Filtering)

Given Initialize the particles $\boldsymbol{x}_0^{[m]}$ using random numbers and set their weight as $w_0^{[m]} = 1/M$.

Step 1-1 Sampling

$$\boldsymbol{x}_k^{(m)} = \boldsymbol{\phi}(\boldsymbol{x}_{k-1}^{(m)}, \boldsymbol{u}_{k-1}, \boldsymbol{\xi}_{k-1}) \tag{17}$$

Step 1-2 Evaluate the likelihood and update the weights.

$$\bar{w}_k^{(m)} = w_{k-1}^{(m)} h(\boldsymbol{y}_k | \boldsymbol{x}_k^{(m)}) \tag{18}$$

$$w_k^{(m)} = \frac{\bar{w}_k^{(m)}}{\sum_{j=1}^{M} \bar{w}_k^{(j)}} \tag{19}$$

Step 1-3 If $N_{eff} < N_T$, execute resampling.

Execution of resampling is dependent on the effective sample size [7]:

$$N_{eff} = \frac{1}{\sum_{i=1}^{M} \left(w_k^{(i)} \right)^2} \tag{20}$$

$N_{eff} = M$ when the weights of all the particles are equal, and $N_{eff} = 1$ when the variety of weights is the largest. This value is an indicator of the number of particles currently utilized effectively, and it is introduced to control the frequent occurrence of resampling. An appropriate threshold N_T is prepared and resampling is executed if the obtained N_{eff} is less than N_T.

(Particle Filtering-Model Predictive Control)

Given The input of each particle is generated as follows:

$$\bar{\boldsymbol{u}}_k^{(m)} = \boldsymbol{u}_{k-1} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u) \tag{21}$$

Moreover, $\bar{\boldsymbol{x}}_k^{(m)} = \boldsymbol{x}_k^{(m)}$ and $\bar{w}_k^{(m)} = w_k^{(m)}$.

Step 2-1 Sampling of the inputs

$$\bar{\boldsymbol{u}}_{k+j}^{(m)} = \bar{\boldsymbol{u}}_{k+j-1}^{(m)} + \boldsymbol{v}, \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_u) \tag{22}$$

Step 2-2 Sampling of the particles

$$\bar{\boldsymbol{x}}_{k+j}^{(m)} = \boldsymbol{\phi}\left(\bar{\boldsymbol{x}}_{k+j-1}^{(m)}, \bar{\boldsymbol{u}}_{k+j-1}^{(m)}, \boldsymbol{\xi}_{k+j-1} \right) \tag{23}$$

Step 2-3 Evaluate the likelihood by using a target trajectry $\boldsymbol{s}_{k+j}$, and update the weights as follows:

$$\tilde{w}_{k+j}^{(m)} = \bar{w}_{k+j-1}^{(m)} h_p \left(\boldsymbol{s}_{k+j} | \bar{\boldsymbol{x}}_{k+j}^{(m)} \right) \tag{24}$$

$$\bar{w}_{k+j}^{(m)} = \frac{\tilde{w}_{k+j}^{(m)}}{\sum_{l=1}^{M} \tilde{w}_l^{(m)}} \tag{25}$$

Step 2-4 If $\bar{N}_{eff} < \bar{N}_T$, execute resampling. Register an input sequence $\bar{\boldsymbol{u}}_{k+j}^{[m]}$.

Step 2-5 Repeat the above processing from $j = 1$ to $j = N_P$.

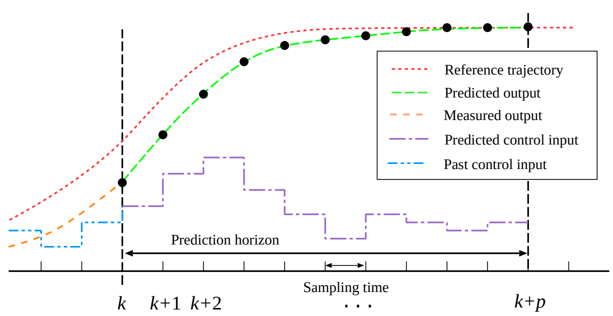


Fig. 2. Model predictive control.

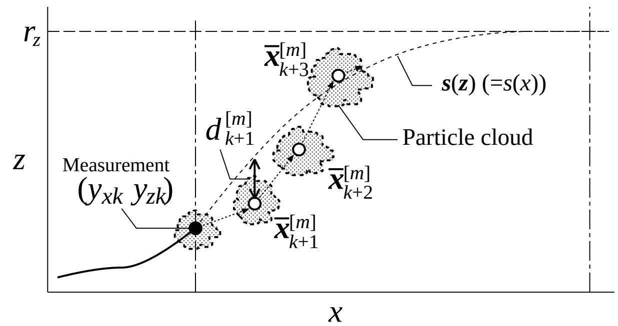


Fig. 3. Relations of the reference trajectory, the target trajectory, and particle clouds.

Step 3 Determine $\boldsymbol{u}_{k+1}$ from

$$\left\{ \bar{\boldsymbol{x}}_{k+1:k+N_P}^{(m)}, \bar{\boldsymbol{u}}_{k+1:k+N_P}^{(m)}, w_{k+1:k+N_P}^{(m)} \right\} \tag{26}$$

based on a following cost function.

IV. Trajectry Tracking Simulation

A. Conditions

1) Parameters: Each parameter was set as follows: $\Delta = 0.01$ [s], $m = 420$ [g], $I_{xx} = I_{yy} = 5.136$ [gm^2], $I_{zz} = 10.16$ [gm^2], $d = 0.21$ [m], $N_P = 40$, $\boldsymbol{\xi}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_v)$, $\boldsymbol{\Sigma}_v = \mathrm{diag}(0.03, 0, 0.03, 0, 0, 0)$, $\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_w)$, $\boldsymbol{\Sigma}_w = \mathrm{diag}(0.001, 0, 0.001, 0, 0, 0)$, $\boldsymbol{\Sigma}_u = \mathrm{diag}(10.5, 0, 10.5, 0, 6, 0)$, and $g = 9.807$ [ms^{-2}]. The number of particles was set to $M = 20$. Experimentally, the proportionality constant vectors were set as $\boldsymbol{K} = [1\ 1\ 1]^T$ and $\boldsymbol{K}_e = [1\ 1\ 1]^T$.

2) Target trajectory: To simplify the simulation, only the composition of the forward, upward, and downward motions was taken into consideration. That is, we set $y_0^{[m]} = v_{y_0}^{[m]} = \theta_{x_0}^{[m]} = \theta_{z_0}^{[m]} = \omega_{x_0}^{[m]} = \omega_{z_0}^{[m]} = 0$.

Next, a reference trajectory $s(x)$ that makes a gradually approach from the measurement position $\boldsymbol{y}_k \triangleq [y_{xk}\ 0\ y_{zk}]^T$ of the quadcopter to the target point $\boldsymbol{r} \triangleq [r_x\ 0\ r_z]^T = [20\ 0\ 5]^T$ at discrete time k was constructed as follows:

$$s(x) = (r_z - y_{zk}) \left(1 - e^{-\frac{x - y_{xk}}{T_s}} \right) + y_{zk} \tag{27}$$

These relations are shown in Fig. 3.

3) Likelihood evaluation function: For the likelihood evaluation, we constructed the following cost function:

$$h_p \left(\boldsymbol{s}_{k+j} | \bar{\boldsymbol{x}}_{k+j}^{(m)} \right) \sim \exp \left\{ -\frac{d_{k+j}^{(m)}}{2\sigma_d^2} \right\} \tag{28}$$

where the distance between the predictive particle $\bar{\boldsymbol{x}}_{k+j}^{(m)}$ after the j-th step and the target trajectory is calculated as follows:

$$d_{k+j}^{(m)} \triangleq \left| \bar{z}_{k+j}^{(m)} - (r_z - y_{zk}) \left(1 - e^{-\frac{\bar{x}_{k+j}^{(m)} - y_{xk}}{T_s}} \right) + y_{zk} \right|$$

where $\sigma_d^2 = 0.03$.

4) Calculation of input: On the basis of the particle prediction, an input $\boldsymbol{u}_{k+1}$ at the time step $k+1$ of a particle that has a state nearest to the target trajectory at the time step $k+N_P$ is adopted as the next input. That is, the selection of the input is executed as follows;

$$\boldsymbol{u}_{k+1} = \underset{\bar{\boldsymbol{u}}_{k+1}^{(m)}}{\text{argmin}} \left\{ \lambda \sum_{i=0}^{N_P} d_{k+i}^{(m)} + (1-\lambda) \left\| \boldsymbol{r} - \bar{\boldsymbol{x}}_{k+N_p}^{(m)} \right\| \right\} \tag{29}$$

where $\lambda = 0.46$.

B. Results

The time evolution of the position of the quadcopter and a part of the trajectories of the particles are shown in Fig. 4. The results shown in this figure indicate that the application of the proposed PF-MPC method to the quadcopter control that is affected by disturbance or measurement noise is effective. Moreover, we confirmed that the time evolution of the input sequences was stable. However, investigation of the influence of the value of distribution of the particle filter and a djustment of a feedback gain for the control is future work.

V. Conclusion

In this paper, we proposed a particle predictive control method for a quadcopter control system that is affected by environmental disturbance and measurement noise. Because the proposed method does not require the inverse model, the nonlinear system model of the quadcopter can be used for forward predictive control. Moreover, the results of a simple two-dimensional simulation showed that the developed particle predictive control system operated effectively.

As future work, we intend to extend the simulation conditions are extended to the control of the roll angle, and pitch angle of y-axis orientation's motion, etc.

References

[1] V.Kumar, N.Michael: Opportunities and challenges with autonomous micro aerial vehicles, The International Journal of Robotics Research, Vol. 31, No. 11, pp. 1279–1291, 2012.

[2] B.Sumantri, N.Uchiyama, S.Sano, Y.Kawabata: Robust tracking control of a quad-Rotor helicopter utilizing sliding mode control with a nonlinear sliding surface, Journal of System Design and Dynamics, Vol.7, No.2, pp.226–241, 2013.

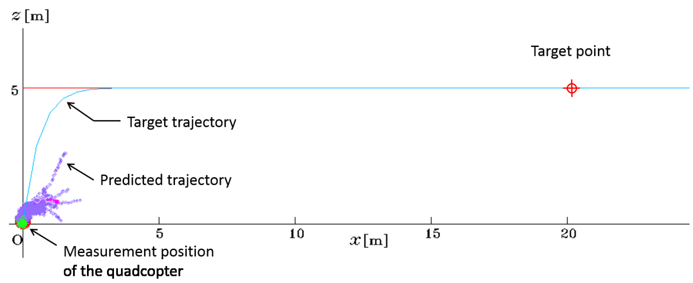
[3] D.Mellinger, N.Michael, V.Kumar: Trajectory generation and control for precise aggressive maneuvers with quadrotors, The International Journal of Robotics Research, Vol.31, No.5, pp.664–674, 2012.

[4] P.Corke: Robotics,Vision and Control Fundamental Algorithms in MATLAB, Springer, pp.78–86, 2011.

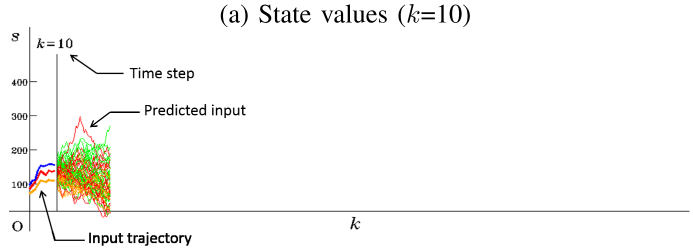
[5] J.P.de Villiers, S.J.Godsill, S.S.Singh: Particle predictive control, Journal of Statistical Planning and Inference, Vol.141, No.5, pp.1753–1763, 2011.

[6] D.Stahl, J.Hauth: PF-MPC particle filter-model predictive control, Systems & Control Letters, Vol.60, No.8, pp.632–643, 2011.

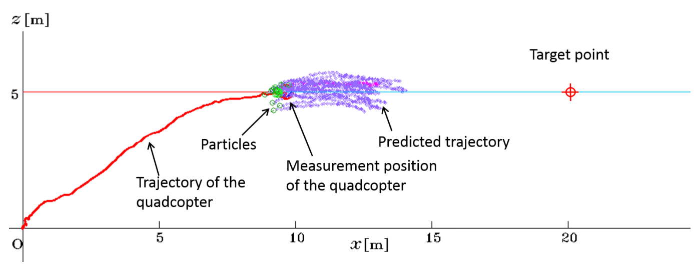
[7] J.S.Liu, R. Chen: Sequential Monte Carlo methods for dynamic systems, Journal of the American Statistical Association, Vol.93, No.443, pp.1032–1044, 1998.



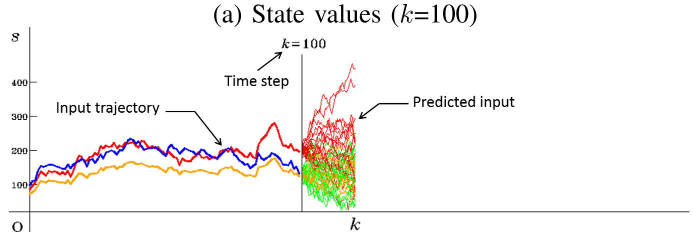
(a) State values (k=10)



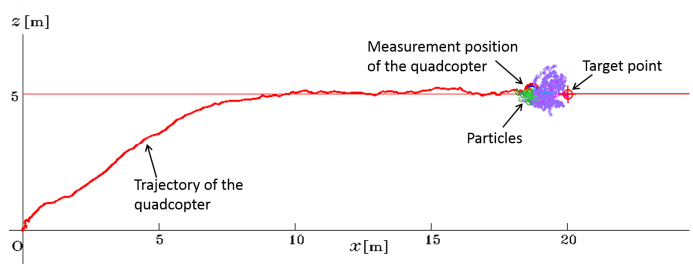
(b) Input (k=10)



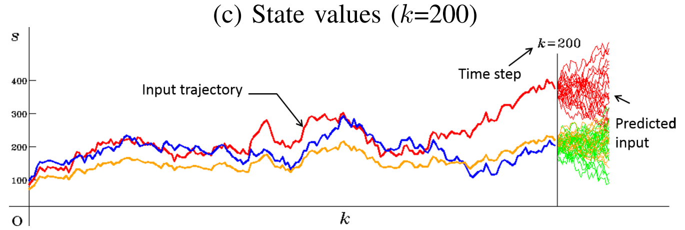
(a) State values (k=100)



(b) Input (k=100)



(c) State values (k=200)



(d) Input (k=200)

Fig. 4. Time evolution of the position of quadcopter and the trajectories of the particles.