JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 6, November–December 1997

Autopilot Design for Bank-to-Turn Missiles Using Receding Horizon Predictive Control Scheme

Myung-Joon Kim,* Wook Hyun Kwon,† and Yong Ho Kim‡
Seoul National University, Seoul 151-742, Republic of Korea
and
Chanho Song§
Agency for Defence Development, Taejon 305-600, Republic of Korea

Receding horizon predictive control (RHPC) methodology is applied to the design of an autopilot for a bank-to-turn missile. The main control objective is high tracking performance for guidance commands. The proposed RHPC-based autopilot consists of the RHPC and the design algorithm of future commands. The design algorithm of future commands determines future guidance commands required in the RHPC. It is based on the first-order approximation of the current guidance commands generated from the existing guidance law. The RHPC is designed for tracking control of pitch acceleration, yaw acceleration, and roll rate. It is shown by computer simulation that the RHPC-based autopilot offers good tracking performance in six-degree-of-freedom environments, which results in low terminal miss distances even for maneuvering targets. Implementation issues such as memory requirements and computing power are also discussed. We demonstrate that the RHPC-based autopilot can be implemented in real time.

Nomenclature

a_x, a_y, a_z	= actual acceleration along body axes
a_{xc}, a_{yc}, a_{zc}	= commanded acceleration along body axes
$a^I_{xc}, a^I_{yc}, a^I_{zc}$	= commanded acceleration along inertial axes
I_{xx}, I_{yy}, I_{zz}	= principal moments of inertia
p, p_c	= actual and commanded roll rate, respectively
q, q_c	= actual and commanded pitch rate, respectively
r, r_c	= actual and commanded yaw rate, respectively
u, v, w	= velocity along body axes
u_δ	= commanded surface deflection
$u_{\delta_q}, u_{\delta_r}, u_{\delta_p}$	= commanded effective surface deflection in pitch, yaw, and roll, respectively
V	= missile velocity
α	= angle of attack
β	= angle of sideslip
δ	= actual surface deflection
$\delta_q, \delta_r, \delta_p$	= effective control surface deflection in pitch, yaw, and roll, respectively
ζ	= actuator damping factor
ω	= actuator natural frequency

I. Introduction

A CONVENTIONAL missile has two distinct subsystems designed independently, a guidance system and a flight control system. The guidance system generates suitable guidance commands for the flight control system. The flight control system, which is called an autopilot, steers the missile to track the guidance commands.[1,2] Specifically, the autopilot is required to obtain high tracking performance such as a good transient response for arbitrary guidance commands, so that the missile may have a high intercept capability with little miss distance.[3]

The majority of existing autopilots do not take the tracking performance for arbitrary guidance commands into account.[4–10] Instead, they deal with zero steady-state error for constant guidance commands. For arbitrary guidance commands, these approaches may yield tracking errors between commanded outputs and achieved outputs. The tracking errors are possibly a major factor contributing to excessive terminal miss distances.[11]

Generally, the tracking performance for arbitrary reference commands is known to be good in predictive controls when finite future reference commands are available in advance. Predictive control has been applied successfully in many control applications, and its good tracking performance has been reported, particularly in process control.[12–14] The receding horizon predictive control (RHPC) algorithm is known to be a general predictive control scheme that is based on general multi-input/multi-output state-space models.[14] Hence, it is expected that the application of the RHPC to a missile autopilot will provide improved target intercept capability.

There have been few results on the design of missile autopilots using the predictive control approach. Bendotti and M'Saad[15] applied generalized predictive control to the single-input/single-output yaw channel of a skid-to-turn missile. Future guidance commands required in predictive controls were assumed to be known in advance, and how to compute future guidance commands was not provided. Note that finite future guidance commands are required to obtain the current control input in predictive controls. Future guidance commands can be computed only when future states of the missile can be determined, if the existing guidance law is used. However, future states of the missile can be determined when future control inputs of the autopilot are given. Therefore, it is difficult to solve future guidance commands in the case where one of the existing guidance laws is used for the guidance system and the predictive control scheme is used for the autopilot. Thus, a method for determining future guidance commands is needed for the successful application of the RHPC to missile autopilots.

An autopilot design based on RHPC methodology is proposed with the design algorithm of future commands. The design algorithm of future commands determines future guidance commands by the first-order approximation of the current guidance commands generated from the existing guidance law. The approximated future guidance commands are fed to the RHPC, which is designed for tracking control of pitch acceleration, yaw acceleration, and roll rate in a bank-to-turn (BTT) missile. We will call the integrated system composed of the RHPC and the design algorithm of future commands the RHPC-based autopilot. Good performance in terms of low miss distances is verified through six-degree-of-freedom (DOF) simulations for intercept scenarios. Because computational burden

Received Nov. 21, 1996; revision received May 5, 1997; accepted for publication May 20, 1997.

*Ph.D. Student, School of Engineering, San 56-1, Shilim-dong, Kwanak-gu. E-mail: kmj@cisl.snu.ac.kr.

†Professor, School of Engineering, San 56-1, Shilim-dong, Kwanak-gu. E-mail: whkwon@cisl.snu.ac.kr.

‡Ph.D. Student, School of Engineering, San 56-1, Shilim-dong, Kwanak-gu. E-mail: yongho@isltg.snu.ac.kr.

§Principal Researcher, Guidance and Control Technology Division.

could be a detrimental aspect of predictive controls, the requirements of memory and computing power for the implementation of the RHPC-based autopilot are also analyzed.

In Sec. II, we explain a dynamic model of a BTT missile and obtain two linear subsystems for design: one for pitch/yaw and one for roll. In Sec. III, the designs of the RHPCs are investigated for each subsystem and the design algorithm of future commands is proposed. The simulation results are shown in Sec. IV for a six-DOF missile model, and the characteristics of the RHPC-based autopilot are discussed. Implementation issues such as memory requirements and computing power are discussed in Sec. V. Finally, Sec. VI presents our conclusions.

II. Plant Description

The motion of a missile in flight is described with nonlinear differential equations including aerodynamic terms. Let us assume that the control actuators in the pitch and yaw channels are expressed by

$$\frac{\delta}{u_\delta} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2}$$

To simplify the derivation of the state equations, we define

$$\alpha = w/u, \qquad \beta = v/u$$

under the assumptions that α and β are small. Then the dynamic equations are derived as follows[6]:

$$\dot{a}_z = (Z_\alpha/V)a_z + Z_\alpha q + Z_\delta\dot{\delta}_q - p(Z_\alpha/Y_\beta)[a_y - Y_\delta\delta_r] \tag{1}$$

$$\dot{q} = \frac{M_\alpha}{Z_\alpha}a_z + M_q q + \left(M_\delta - \frac{Z_\delta M_\alpha}{Z_\alpha}\right)\delta_q + \left(\frac{I_{zz} - I_{xx}}{I_{yy}}\right)pr \tag{2}$$

$$\dot{\delta}_q = \frac{\mathrm{d}\delta_q}{\mathrm{d}t} \tag{3}$$

$$\ddot{\delta}_q = \omega^2\left(u_{\delta_q} - \delta_q\right) - 2\zeta\omega\frac{\mathrm{d}\delta_q}{\mathrm{d}t} \tag{4}$$

$$\dot{a}_y = (Y_\beta/V)a_y - Y_\beta r + Y_\delta\dot{\delta}_r + p(Y_\beta/Z_\alpha)[a_z - Z_\delta\delta_q] \tag{5}$$

$$\dot{r} = \frac{N_\beta}{Y_\beta}a_y + N_r r + \left(N_\delta - \frac{Y_\delta N_\beta}{Y_\beta}\right)\delta_r + \left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right)pq \tag{6}$$

$$\dot{\delta}_r = \frac{\mathrm{d}\delta_r}{\mathrm{d}t} \tag{7}$$

$$\ddot{\delta}_r = \omega^2\left(u_{\delta_r} - \delta_r\right) - 2\zeta\omega\frac{\mathrm{d}\delta_r}{\mathrm{d}t} \tag{8}$$

$$\dot{p} = L_p p + L_\delta u_{\delta_p} + L_\beta\beta \tag{9}$$

where Y_β, Y_δ, Z_α, Z_δ, L_β, L_p, L_δ, M_α, M_q, M_δ, N_β, N_r, and N_δ are aerodynamic derivatives, which are complicated functions of Mach number, dynamic pressure, angle of attack, angle of sideslip, etc. The real values of aerodynamic derivatives at a flight condition are given in the Appendix. In Eqs. (1–9), cross-coupling terms proportional to the roll rate appear due to coriolis and gyroscopic moments. When the roll rate p is constant, it is possible to separate Eqs. (1–9) into the pitch/yaw channel of Eqs. (1–8) and the roll channel of Eq. (9). At an appropriate operating point, we can linearize the models of each channel and obtain two linear time-invariant state-space models. The state, input, and output variables of the pitch/yaw channel are as follows:

$$\boldsymbol{x}^{P/Y} = [a_z \quad q \quad \delta_q \quad \dot{\delta}_q \quad a_y \quad r \quad \delta_r \quad \dot{\delta}_r]^T \tag{10}$$
$$\boldsymbol{u}^{P/Y} = \left[u_{\delta_q} \quad u_{\delta_r}\right]^T, \qquad \boldsymbol{y}^{P/Y} = [a_z \quad q \quad a_y \quad r]^T$$

By neglecting the β term in Eq. (9), we can also obtain the model of the roll channel whose variables are as follows:

$$x^R = p, \qquad u^R = u_{\delta_p}, \qquad y^R = p \tag{11}$$

III. Design of the RHPC-Based Autopilot

A. Design of the Pitch/Yaw and the Roll RHPCs

We start designing the RHPCs by discretizing the linear models of Eqs. (10) and (11). By augmenting integrators, the discrete models are converted to the form of

$$\boldsymbol{x}(t+1) = A\boldsymbol{x}(t) + B\Delta\boldsymbol{u}(t) + D\boldsymbol{\xi}_1(t)$$
$$\boldsymbol{y}(t) = H\boldsymbol{x}(t) + \boldsymbol{\xi}_2(t)$$

where $\boldsymbol{x}(t) \in R^n$, $\Delta\boldsymbol{u}(t) \in R^l$, and $\boldsymbol{y}(t) \in R^m$ and A, B, D, and H are $n \times n$, $n \times l$, $n \times l$, and $m \times n$ matrices, respectively. Δ is $1 - q^{-1}$, and q^{-1} is the unit delay operator. Also, $\boldsymbol{\xi}_1(t) \in R^l$ and $\boldsymbol{\xi}_2(t) \in R^m$ are zero-mean, white noise sequences with variances given by

$$\mathrm{cov}[\boldsymbol{\xi}_1] = Q_f, \qquad \mathrm{cov}[\boldsymbol{\xi}_2] = R_f$$

where $Q_f \geq 0$ and $R_f > 0$.

The RHPC scheme takes the following strategy.[14]

1) At the present time t, it is assumed that the finite values of the command signal $\boldsymbol{y}_r(t+1), \boldsymbol{y}_r(t+2), \ldots, \boldsymbol{y}_r(t+N)$, over the future horizon N are available.

2) A certain type of output predictor $\hat{\boldsymbol{y}}(t+j)$, $j = 1, 2, \ldots, N$, is introduced. (The Kalman predictor will be employed to make the output prediction.)

3) Control increments $\Delta\boldsymbol{u}(t), \Delta\boldsymbol{u}(t+1), \ldots, \Delta\boldsymbol{u}(t+N)$, are obtained to minimize the following finite quadratic cost function under the assumption that $\Delta\boldsymbol{u}(t+j)$, $j = N+1, N+2, \ldots, N+N_F$, are zero:

$$J = \sum_{j=1}^{N} \|\boldsymbol{y}_r(t+j) - \hat{\boldsymbol{y}}(t+j)\|_{Q_c} + \sum_{j=0}^{N} \|\Delta\boldsymbol{u}(t+j)\|_{R_c}$$
$$+ \sum_{j=N+1}^{N+N_F} \|\boldsymbol{y}_r(t+j) - \hat{\boldsymbol{y}}(t+j)\|_{Q_F}$$

where $\|\boldsymbol{x}\|_Q$ is $\boldsymbol{x}^T Q\boldsymbol{x}$ and $\hat{\boldsymbol{y}}(t+j)$ is the estimation of $\boldsymbol{y}(t+j)$, which is obtained at step 2.

4) Only the first control increment $\Delta u(t)$ is applied at the present time. At the next time, $t+1$, the overall procedure is repeated.

The RHPC law comprises a state estimator, a feedforward gain, and a feedback gain. To consider the effect of computational delay in the real implementation, we adopt the Kalman predictor as a state estimator; then $\hat{\boldsymbol{x}}(t \mid t-1)$ is obtained by[16]

$$\hat{\boldsymbol{x}}(t \mid t-1) = A(I - KH)\hat{\boldsymbol{x}}(t-1 \mid t-2)$$
$$+ B\Delta\boldsymbol{u}(t-1) + AK\boldsymbol{y}(t-1) \tag{12}$$

where

$$K = PH^T\left(HPH^T + R_f\right)^{-1} \tag{13}$$

and P satisfies

$$P = APA^T - APH^T\left(HPH^T + R_f\right)^{-1}HPA^T + DQ_fD^T \tag{14}$$

Then the RHPC law can be written as

$$\Delta u(t) = -\left[R_c + B^T F(0)B\right]^{-1} B^T[F(0)A\hat{\boldsymbol{x}}(t \mid t-1) + \boldsymbol{g}(t)] \tag{15}$$

where $F(0)$ and $\boldsymbol{g}(t)$ are obtained from the following recursion:

$$F(i) = A^T F(i+1)A - A^T F(i+1)B\left[R_c + B^T F(i+1)B\right]^{-1}$$
$$\times B^T F(i+1)A + H^T Q_c H, \qquad i < N \tag{16}$$

$$F(N) = \sum_{j=0}^{N_F - 1} A^{j^T} H^T Q_F H^T A^j \tag{17}$$

$$\boldsymbol{g}(t+i) = \begin{cases} A^T\left[I + F(i+1)BR_c^{-1}B^T\right]^{-1}\boldsymbol{g}(t+i+1) \\ \quad -H^T Q_c\boldsymbol{y}_r(t+i+1) \qquad \text{for} \quad i < N \\ A^T\boldsymbol{g}(t+i+1) - H^T Q_F\boldsymbol{y}_r(t+i+1) \\ \qquad\qquad \text{for} \quad N \leq i < N + N_F - 1 \end{cases} \tag{18}$$

with $\boldsymbol{g}(t+N+N_F-1) = -H^T Q_F \boldsymbol{y}_r(t+N+N_F)$. The real values of filter gain AK in Eq. (12) and feedback control gain $-[R_c + B^T F(0) B]^{-1} B^T F(0) A$ in Eq. (15) are given in the Appendix for a flight condition.

Here $\boldsymbol{y}_r(\cdot)$ are reference commands to the outputs of the model. They are composed of a_{zc} and a_{yc}, which are generated by a guidance algorithm. In BTT missiles, null command is used as a_{yc} because the motion in the horizontal plane is achieved not by yaw acceleration but by pitch acceleration and rolling. Thus, $\boldsymbol{y}_r$ of the pitch/yaw model is given by

$$\boldsymbol{y}_r^{P/Y} \triangleq \begin{bmatrix} a_{zc} & q_c & a_{yc} & r_c \end{bmatrix}^T = [a_{zc} \quad -(a_{zc}/V) \quad 0 \quad 0]^T \tag{19}$$

The zero reference command to a_y plays a role in preventing the yaw action that may result in a large sideslip angle. In the RHPC for the roll channel,

$$y_r^R \triangleq p_c = -K_r \arctan(a_{yc}/a_{zc}) \tag{20}$$

is used as the reference command, where K_r is a proportional constant.

The final weighting horizon N_F and the control horizon N are design parameters peculiar to the RHPC, which do not appear in the infinite horizon control problems such as the linear quadratic Gaussian (LQG). They should be greater than or equal to the system order. When the exact future commands are known in advance, the RHPC makes the response smoother and produces less tracking error as the horizon becomes larger. In real applications, however, exact future guidance commands are not available. Instead, we use approximated future commands, which will be explained later. As N_F and N become larger, approximated future guidance commands for longer times are required. In this case, abrupt target maneuvering may cause significant error in the approximated future guidance commands. Except for the feedforward gain, the structure of the RHPC is similar to that of the well-known LQG control, which is generated from the interconnection of a feedback control gain and a state estimator.[16] Thus, the physical meaning of Q and R except N and N_F in the RHPC are basically equal to those in the LQG control.

B. Design of Future Commands

Note that the RHPC needs $\boldsymbol{y}_r(t+1 \mid t), \ldots, \boldsymbol{y}_r(t+N+N_F \mid t)$ to obtain control increment $\Delta\boldsymbol{u}(t)$, whereas general nonpredictive control laws need only $\boldsymbol{y}_r(t+1 \mid t)$. Thus, it is required to predict future guidance commands. The method of determining future guidance commands greatly affects the performance of the RHPC-based autopilot.

Once the current command, $\boldsymbol{y}_r(t+1 \mid t)$, is defined by Eqs. (19) and (20) using the existing guidance law, future commands are composed as follows:

$$\boldsymbol{y}_r(t+i \mid t) = \boldsymbol{y}_r(t+i-1 \mid t) + \dot{\boldsymbol{y}}_r(t) T_s \quad \text{for} \quad 2 \le i \le N + N_F \tag{21}$$

where T_s is the sampling period and $\dot{\boldsymbol{y}}_r(t)$ is obtained by

$$\dot{\boldsymbol{y}}_r(t) = \frac{\boldsymbol{y}_r(t+1 \mid t) - \boldsymbol{y}_r(t \mid t-1)}{T_s} \tag{22}$$

and $\boldsymbol{y}_r(t \mid t-1)$ is composed of the acceleration commands computed from the existing guidance law at the preceding discrete time step. Both $\boldsymbol{y}_r^{P/Y}(\cdot)$ and $y_r^R(\cdot)$ are approximated via this method.

In the preceding method, future commands are determined by using the tangent vector of the current value during the finite future horizon. This method is successfully applied under the condition that the derivatives of a_{yc} and a_{zc} will not change severely during the horizon. For many guidance laws in which the time histories of guidance commands are smooth and continuous, such as the conventional proportional navigation guidance, we can easily select the length of the future horizon to satisfy the described condition.

IV. Performance Evaluation

We introduce closed-loop, air-to-air intercept scenarios to assess the performance of the designed autopilot. We adopt a guidance law as follows[2,6]:

$$\boldsymbol{A}_c = \frac{K_n(\boldsymbol{R}_{\text{rel}} + \boldsymbol{V}_{\text{rel}} t_{\text{go}})}{t_{\text{go}}^2} \tag{23}$$

where $\boldsymbol{A}_c = [a_{xc}^I \ a_{yc}^I \ a_{zc}^I]^T$, $\boldsymbol{R}_{\text{rel}}$ is the three-component vector of relative position represented in the inertial axes, $\boldsymbol{V}_{\text{rel}}$ is the three-component vector of relative velocity represented in the inertial axes, t_{go} is the time-to-go, and K_n is the navigation constant. In the simulation, the inertial x axis is established along the initial line-of-sight (LOS) and the LOS angle is assumed not to be much different from the initial one over the whole engagement. The commanded acceleration $[a_{xc}^I \ a_{yc}^I \ a_{zc}^I]^T$ is converted to $[a_{xc} \ a_{yc} \ a_{zc}]^T$ for Eqs. (19) and (20). The time-to-go is calculated using the formula[6]

$$t_{\text{go}} = \frac{2r_x}{-v_x + \left[v_x^2 + (4 r_x a_x / K_n)\right]^{\frac{1}{2}}} \tag{24}$$

where r_x and v_x are the relative position and the relative velocity in the direction of the missile x axis.

Two cases are considered for comparison. Case 1 uses constant future commands as follows:

$$\boldsymbol{y}_r(t+1 \mid t) = \boldsymbol{y}_r(t+2 \mid t) = \cdots = \boldsymbol{y}_r(t+N+N_F \mid t)$$

and case 2 uses future commands generated from Eqs. (21) and (22). The same RHPC is used in both cases. Note that case 1 has no prediction concept and is basically similar to existing set-point tracking controllers. Case 2 is the proposed RHPC-based autopilot. The RHPC is designed on various operating points. In the simulation, the gain is scheduled as a function of dynamic pressure, roll rate, and angle of attack. To minimize the effect of the guidance error, all of the quantities required in Eqs. (23) and (24) are assumed to be known exactly. A g-bias is used to counteract the effect of gravity. Performance for two cases was evaluated by six-DOF simulation including all of the gyroscopic and coriolis couplings and the variation of aerodynamic parameters. The aerodynamic parameters were obtained from tabular aerodynamic data via multidimensional linear interpolation routines. The limiting values of pitch acceleration and roll rate are assumed to be ±30 g and ±8 rad/s. The following results were obtained with MATLAB. The six-DOF simulation block is shown in Fig. 1, where integrators are used to obtain control input $\boldsymbol{u}(t)$ from control increment $\Delta\boldsymbol{u}(t)$ of Eq. (15).

The first scenario is shown in Fig. 2. The initial altitudes of the missile and the target are 10,000 and 11,000 ft, respectively. The target is modeled as a point mass. When the time-to-go drops to less than 1 s, it starts maneuvering according to

$$\boldsymbol{A}_T(t) = \begin{bmatrix} -7\,g[1 - t_{\text{go}}(t)]^2 \\ 0 \\ -7\,g[1 - t_{\text{go}}(t)]^2 \end{bmatrix} \tag{25}$$

where $\boldsymbol{A}_T = [a_{tx} \ a_{ty} \ a_{tz}]^T$ denotes the vector of target acceleration represented in the initial missile body axes. This scenario causes only pitch motion of the missile, and it helps to evaluate the effectiveness of the proposed design algorithm of future commands. The initial z-axis acceleration of the missile is $-10\,g$. Figure 3 shows the time responses of a_z for two pitch/yaw autopilots. At a glance, case 2 gives a response closer to the command than case 1. The miss distance is only 0.2527 ft. In case 1, however, greater tracking error gives birth to the miss distance of 1.4234 ft. From Fig. 3, it appears that the prediction is useful when the derivative of the command changes severely as well as when it is nearly constant. Table 1 summarizes the miss distances for different target maneuverings in scenario 1. Case 2 is shown to be superior to case 1 for all runs.

To evaluate the designed autopilots when roll motion is also present, the second scenario (Fig. 4) is introduced. The altitude of the missile and the target is 10,000 ft. The target starts a 9-g turn

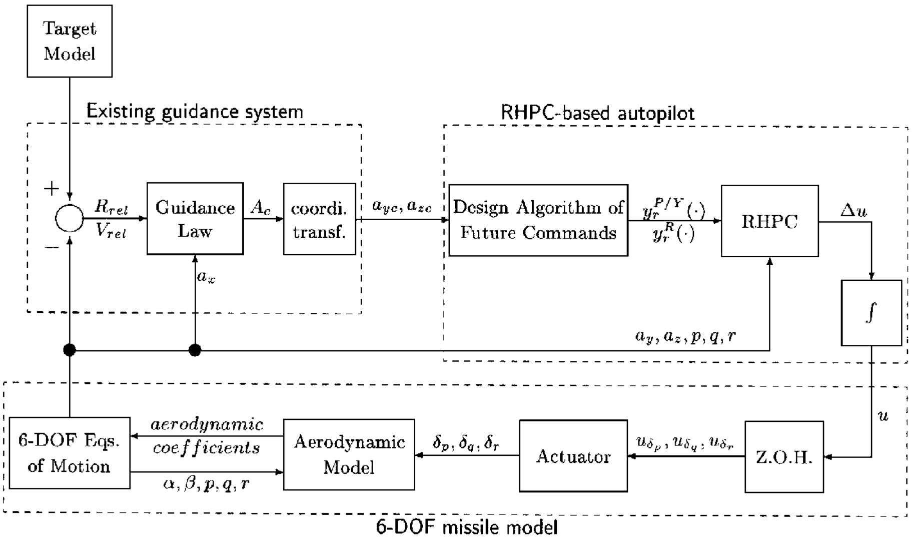


Fig. 1 Six-DOF simulation block diagram.

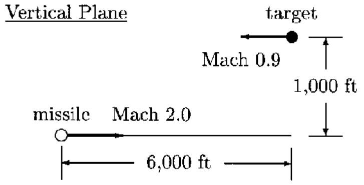
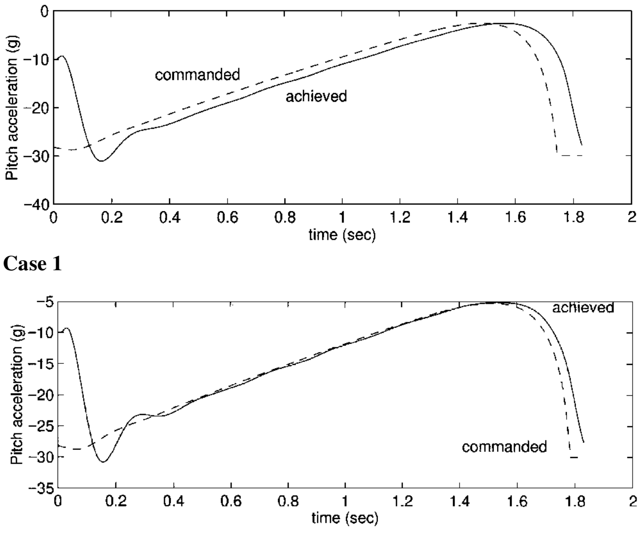


Fig. 2 Scenario 1.



Case 1

Case 2

Fig. 3 Response of scenario 1.

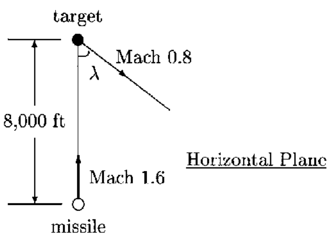
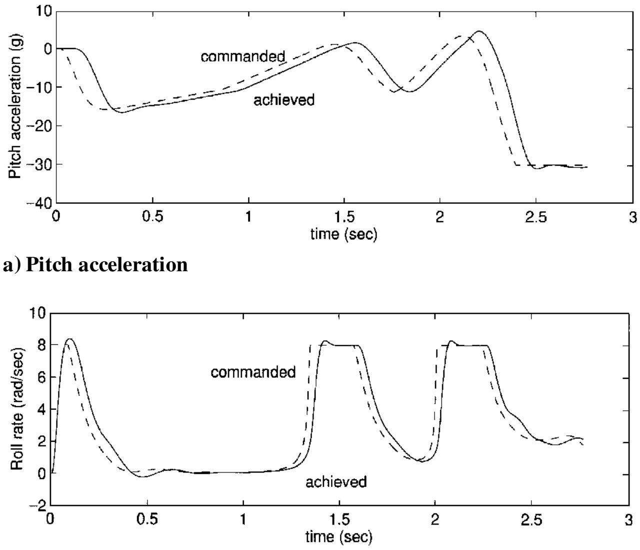


Fig. 4 Scenario 2.



a) Pitch acceleration

b) Roll rate

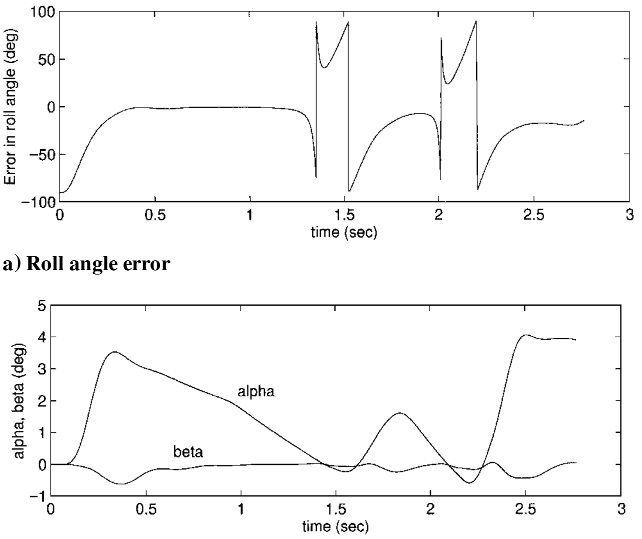
Fig. 5 Response of scenario 2 (case 1).

to the right when the time-to-go is 2 s and a 9-g turn to the left when the time-to-go is 1 s. This target maneuvering is expected to yield marked changes in the acceleration command and to increase the possibility of failure of prediction in the design algorithm of future commands. The time responses of cases 1 and 2 are shown in Figs. 5–8. The pitch acceleration changes more severely in scenario 2 than in scenario 1. Although case 2 yields a larger overshoot when the commanded acceleration changes abruptly, it gives a better overall response. The initial value of the roll angle error, which is defined as

$$e_\phi = \arctan(a_{yc}/a_{zc})$$

is −90 deg due to the component of target velocity. As the roll rate becomes the limiting value, the roll angle error rapidly decays. Also note that, whenever the target acceleration changes abruptly, the roll rate is increased. The angle of attack and the angle of sideslip are shown in Figs. 6b and 8b. Even when a transient occurs, the sideslip is kept below 1 deg. The miss distance of case 2 is only 0.6127 ft, whereas that of case 1 is 5.2774 ft. The difference of the miss distances between two cases comes from their tracking performance. In case 1, greater time lag leads to the acceleration command saturation. It contributes to the severe miss distance because the acceleration command saturation just prior to the impact directly results in miss distance. Table 2 shows the miss distances for various target maneuverings in scenario 2. Case 2 yields better results for all runs than case 1. From these results, we conclude that the designed RHPC-based autopilot provides an excellent intercept capability for the BTT missile. The proposed design algorithm of future commands was verified to work well, even for a highly maneuvering target.

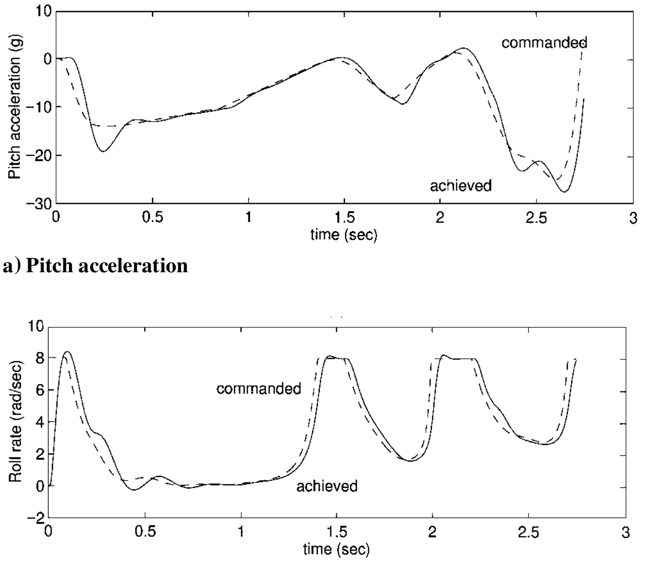
Now let us verify the performance of the RHPC-based autopilots when some seeker noises are present. Instead of the existing



a) Roll angle error

b) Angle of attack and angle of sideslip

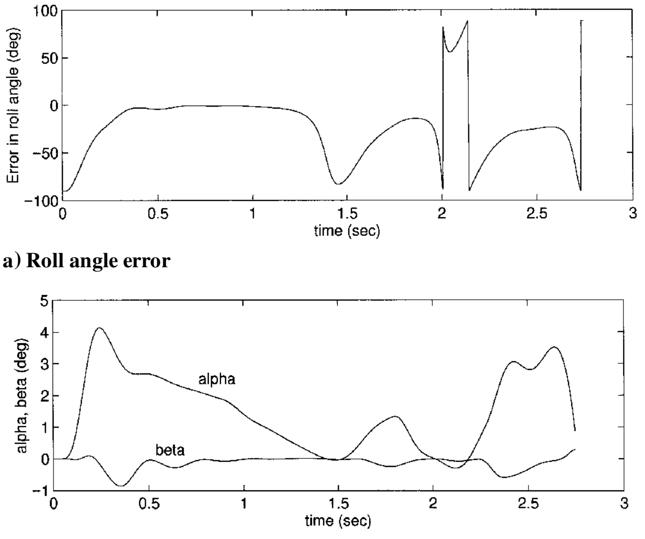
Fig. 6 Response of scenario 2 (case 1).



a) Pitch acceleration

b) Roll rate

Fig. 7 Response of scenario 2 (case 2).



a) Roll angle error

b) Angle of attack and angle of sideslip

Fig. 8 Response of scenario 2 (case 2).

Table 1 Summary of miss distances for scenario 1

	Target maneuver		Miss distance, ft	
Run	1st ($t_{\rm go} = 1$ s)	2nd ($t_{\rm go} = 0.3$ s)	Case 1	Case 2
1	Eq. (25)	none	1.4234	0.2527
2	9-g turn up	9-g turn down	6.2663	2.4998
3	9-g turn down	9-g turn up	7.6228	0.9541

Table 2 Summary of miss distances for scenario 2

	λ,	Target maneuver		Miss distance, ft	
Run	deg	1st ($t_{\rm go} = 2$ s)	2nd ($t_{\rm go} = 1$ s)	Case 1	Case 2
1	30	9-g turn	none	0.2084	0.1431
2	90	to the left		0.2185	0.1881
3	30	9-g turn	none	0.1486	0.0239
4	90	to the right		0.1243	0.1074
5	30	9-g turn	9-g turn	9.2738	5.9942
6	90	to the left	to the right	2.2096	1.3296
7	30	9-g turn	9-g turn	5.2774	0.6127
8	90	to the right	to the left	1.8179	1.3588

Table 3 Summary of miss distances for scenario 2 with sekeer noise

	λ,	Target maneuver		rms miss distance, ft	
Run	deg	1st ($t_{\rm go} = 2$ s)	2nd ($t_{\rm go} = 1$ s)	Case 1	Case 2
1	30	9-g turn	none	3.7066	3.2102
2	90	to the left		2.6907	3.0098
3	30	9-g turn	none	3.0611	2.1803
4	90	to the right		5.4842	4.7025
5	30	9-g turn	9-g turn	25.0194	15.1291
6	90	to the left	to the right	6.9782	6.9322
7	30	9-g turn	9-g turn	32.0045	12.1919
8	90	to the right	to the left	6.3465	6.4554

Table 4 Memory requirements

Matrices to be stored	Number of elements
$A(I - KH)$ in Eq. (12)	$n \times n$
B in Eq. (12)	$n \times l$
AK in Eq. (12)	$n \times m$
$-H^T Q_F$ in Eq. (18)	$n \times m$
A^T in Eq. (18)	$n \times n$
$A^T[I + F(i+1)BR^{-1}B^T]^{-1}$ for $0 \le i < N$ in Eq. (18)	$(n \times n) \times N$
$-H^T Q_c$ in Eq. (18)	$n \times m$
$F(0)A$ in Eq. (15)	$n \times n$
$-[R + B^T F(0) B]^{-1} B^T$ in Eq. (15)	$l \times n$

Table 5 Computational burden

Equations	FLOPs	Real value
$\hat{\boldsymbol{x}}(t \mid t-1)$ in Eq. (12)	$\le 2n(n + l + m)$	320
$\boldsymbol{g}(t+i)$ for $N \le i \le N + N_F - 1$ in Eq. (18)	$\le 2n(n + m)N_F$	2800
$\boldsymbol{g}(t+i)$ for $i < N$ in Eq. (18)	$\le 2n(n + m)N$	5600
$\Delta \boldsymbol{u}(t)$ in Eq. (15)	$\le 2n(n + l)$	240

guidance system block in Fig. 1, we introduce one in Fig. 9, where $\hat{\boldsymbol{R}}_{\rm rel}$, $\hat{\boldsymbol{V}}_{\rm rel}$, $\hat{r}_x$, and $\hat{v}_x$ are noise-corrupted data of $\boldsymbol{R}_{\rm rel}$, $\boldsymbol{V}_{\rm rel}$, r_x, and v_x, respectively. The seeker noise shown in Fig. 9 represents glint noise, which is a position error on $\boldsymbol{R}_{\rm rel}$. The η is three-component noise source, and each component is a zero-mean white Gaussian noise with variance 4 ft. Note that this seeker noise causes not only inaccurate $\hat{\boldsymbol{R}}_{\rm rel}$ and $\hat{\boldsymbol{V}}_{\rm rel}$ but also inaccurate $t_{\rm go}$. The rms miss distances for scenario 2 are shown in Table 3, where each rms miss distance was obtained from 50 trials. In most runs, case 2 is shown to be better than case 1. It means that the performance of the RHPC-based autopilot is satisfactory even with the seeker noise and the inaccurate time-to-go.

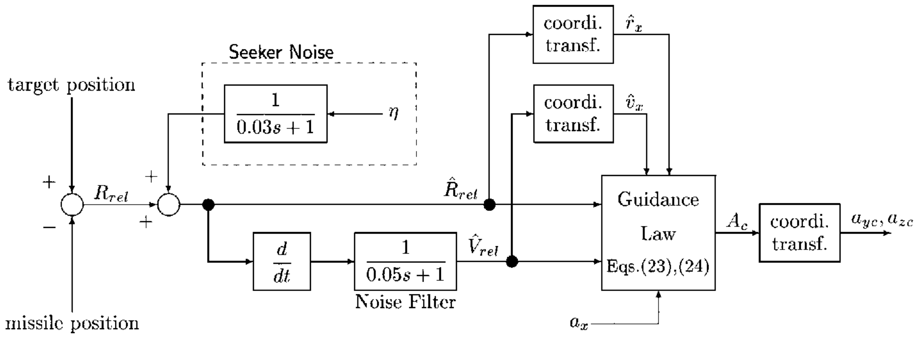


Fig. 9 Guidance system with seeker noise.

Note that we use approximated future guidance commands under the assumption that future commands will vary to the tangent direction of the current value. Any future information is not used. In the case of an air-to-air missile, it is difficult to obtain exact future guidance commands. However, there are many applications such as the flight-path control problem[17] in which reference commands in the future are known in advance. In these cases, RHPC could be applied more naturally and offer better performance with known future information.

V. Implementation Issues

The designed RHPC expressed by Eqs. (12–18) takes its computation time into account. Using $\boldsymbol{y}(t-1)$, $\Delta\boldsymbol{u}(t)$ is computed during one step size between $t-1$ and t, and it is applied from t to $t+1$. We designed the RHPCs for about 1000 operating points to cover all of the flight conditions of the missile. For an operating point, many parts in the computing procedures of $\boldsymbol{g}(t)$ and $\hat{\boldsymbol{x}}(t \mid t-1)$ as well as feedback gain $F(\cdot)$ can be obtained by off-line computation. To minimize the on-line computational burden in the real implementation, these matrices can be stored in read-only memory (ROM) or flash memory and referred to in the on-line computation. Table 4 shows these matrices and the memory requirements to store them per one operating point. Note that most of the on-line computational burden and memory requirements result from the RHPC for the pitch/yaw channel because the order of the RHPC for the pitch/yaw channel is much larger than that of the RHPC for the roll channel. Thus, we consider only the RHPC for the pitch/yaw channel here. In the case of the RHPC for the pitch/yaw channel, the total number of floating-point elements required is 2300 according to $n=10, m=4$, and $l=2$. Because one floating-point number requires 4 bytes, 9200 bytes are required to store the precalculated matrices of the RHPC for the pitch/yaw channel per one operating point. Hence, it amounts to a total of approximately 10 megabytes for all of the designed RHPCs. This amount of memory is feasible, at a low cost by today's standards, by using common flash memory or ROM devices.

The on-line computational burden using this approach is shown in Table 5. The computational burden to obtain future commands $\boldsymbol{y}_r(t+1), \boldsymbol{y}_r(t+2), \ldots, \boldsymbol{y}_r(t+N+N_F)$ is not considered here because this burden is negligible compared with that of the RHPC. The sequence should execute 8960 floating-point operations (FLOPs) in 5 ms, one step size, for the rate of 1.8×10^6 floating-point operations per second (MFLOPS). The TMS32040, a commercially available digital signal processor (DSP), supports 20 MFLOPS (Ref. 18). This means that the designed RHPC-based autopilot could be easily implemented with a DSP.

VI. Conclusions

This study has demonstrated that RHPC methodology can be applied successfully to an autopilot design for a BTT missile while using the existing guidance law. The problem of computing future guidance commands inherent to the predictive control scheme was solved through a design algorithm based on a first-order approximation of the current guidance commands. It was shown by computer simulation that the proposed RHPC-based autopilot offers excellent tracking performance for arbitrary guidance commands and that this contributes to reduced terminal miss distances. The analyses on memory requirements and computational burden showed that the suggested autopilot could be implemented with currently available devices such as a DSP chip and 10 megabytes of ROM devices.

Appendix: Representative Data

The flight conditions are altitude = 10,000 ft, Mach = 2.0, $\alpha = 1.6036$ deg, and $\beta = 0$ deg. The aerodynamic derivatives and design parameters are geven in Tables A1 and A2, respectively.

Filter Gain in Eq. (12)

Pitch/yaw channel:

$AK =$

$$\begin{bmatrix} -1.5702E\text{-}3 & -2.8180E\text{-}1 & -5.0000E\text{-}14 & 1.1770E\text{-}11 \\ 1.0000E\text{-}15 & 1.4000E\text{-}13 & 1.2076E\text{-}3 & -2.7202E\text{-}1 \\ 2.7073 & -6.3445 & -2.5000E\text{-}13 & -6.2050E\text{-}11 \\ -7.7681E\text{-}3 & 2.0303E\text{-}1 & 1.0000E\text{-}14 & -5.1000E\text{-}13 \\ 2.9672E\text{-}4 & -1.6832E\text{-}3 & -1.0000E\text{-}15 & -1.0000E\text{-}14 \\ -1.1041E\text{-}1 & -3.1904E\text{-}1 & 1.0000E\text{-}14 & 1.1540E\text{-}11 \\ 6.0000E\text{-}14 & 3.3800E\text{-}12 & 2.6467 & 1.7989 \\ 1.0000E\text{-}15 & 1.0000E\text{-}14 & 4.6466E\text{-}3 & 2.6181E\text{-}1 \\ -1.0000E\text{-}15 & -1.0000E\text{-}15 & -3.6349E\text{-}4 & -1.3399E\text{-}3 \\ 1.0000E\text{-}14 & 5.0000E\text{-}13 & 1.3121E\text{-}1 & -1.7047E\text{-}1 \end{bmatrix}$$

Roll channel:

$$AK = \begin{bmatrix} 0.0086 \\ 1.9826 \end{bmatrix}$$

Feedback Control Gain in Eq. (15)

Pitch/yaw channel:

$$-\left[R_c + B^T F(0) B\right]^{-1} B^T F(0) A$$

$$= \begin{bmatrix} -0.2632 & 0 & -0.0008 & 1.2741 & -9.2613 & -0.0396 & 0 & 0 & 0 & 0 \\ 0 & -0.0909 & 0 & 0 & 0 & 0 & -0.0013 & 0.1694 & 1.9964 & -0.0059 \end{bmatrix}$$

Table A1 Aerodynamic derivatives

$$Y_\beta = -3.3651E{+}03 \text{ ft/s}^2 \qquad Y_\delta = 2.1599E{+}03 \text{ ft/s}^2$$
$$Z_\alpha = -1.1691E{+}04 \text{ ft/s}^2 \qquad Z_\delta = -2.2108E{+}03\text{ft/s}^2$$
$$L_\beta = -3.1195E{+}03 \text{ 1/s}^2 \qquad L_p = -2.9066 \text{ 1/s}$$
$$L_\delta = 2.3274E{+}04 \text{ 1/s}^2$$
$$M_\alpha = -91.5918 \text{ 1/s}^2 \qquad M_q = -0.0322 \text{ 1/s}$$
$$M_\delta = -1.0386E{+}03 \text{ 1/s}^2$$
$$N_\beta = 458.4064 \text{ 1/s}^2 \qquad N_r = -0.0322 \text{ 1/s}$$
$$N_\delta = -986.9393 \text{ 1/s}^2$$

Table A2 Design parameters

Sampling period	5 ms
K_n in Eq. (23)	3
K_r in Eq. (20)	7
N (pitch/yaw channel)	20
N_F (pitch/yaw channel)	10
N (roll channel)	4
N_F (roll channel)	2

Roll channel:

$$-[R_c + B^T F(0) B]^{-1} B^T F(0) A = [-60.2784 \quad -0.0774]$$

Acknowledgment

This work was supported by the Agency for Defence Development, Republic of Korea, under Contract UD940124BD.

References

[1] Blakelock, J. H., *Automatic Control of Aircraft and Missiles*, 2nd ed., Wiley, New York, 1991, Chap. 7.

[2] Lin, C. F., *Modern Navigation, Guidance, and Control Processing*, Prentice–Hall, Englewood Cliffs, NJ, 1991, Chap. 2.

[3] Lin, C. F., *Advanced Control Systems Design*, Prentice–Hall, Englewood Cliffs, NJ, 1994, Chap. 1.

[4] Cloutier, J. R., Evers, J. H., and Feely, J. J., "An Assessment of Air-to-Air Missile Guidance and Control Technology," *Proceedings of the 1988 American Control Conference* (Atlanta, GA), 1988, pp. 133–142.

[5] Arrow, A., and Williams, D. E., "Comparison of Classical and Modern Missile Autopilot Design and Analysis Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 12, No. 2, 1989, pp. 220–227.

[6] Williams, D. E., Friedland, B., and Madiwale, A. N., "Modern Control Theory for Design of Autopilots for Bank-to-Turn Missiles," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 378–386.

[7] Wise, K. A., "Bank-to-Turn Missile Autopilot Design Using Loop Transfer Recovery," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 1, 1990, pp. 145–152.

[8] Krause, J., and Stein, G., "A General Adaptive Control Structure with a Missile Application," *Proceedings of the 1988 American Control Conference* (Atlanta, GA), 1988, pp. 561–566.

[9] Sheperd, C. L., and Valavani, L., "Autopilot Design for Bank-to-Turn Missiles Using LQG/LTR Methodology," *Proceedings of the 1988 American Control Conference* (Atlanta, GA), 1988, pp. 579–586.

[10] Sobel, K. M., and Shapiro, E. Y., "Application of Eigenstructure Assignment to Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 1, 1987, pp. 529–531.

[11] Cottrell, R. G., "Optimal Intercept Guidance for Short-Range Tactical Missiles," *AIAA Journal*, Vol. 9, No. 7, 1971, pp. 1414, 1415.

[12] Clarke, D. W., Mohtadi, C., and Tuffs, P. S., "Generalized Predictive Control—Part I. The Basic Algorithm," *Automatica*, Vol. 23, No. 2, 1987, pp. 137–148.

[13] Kwon, W. H., and Byun, D. G., "Receding Horizon Tracking Control as a Predictive Control and Its Stability Properties," *International Journal of Control*, Vol. 50, No. 5, 1989, pp. 1807–1824.

[14] Lee, Y. I., Kwon, W. H., and Noh, S., "A Receding Horizon Predictive Control and Its Related GPC with Stability Properties," *Control Theory and Advanced Technology*, Vol. 10, No. 3, 1994, pp. 523–537.

[15] Bendotti, P., and M'Saad, M., "A Skid-to-Turn Missile Autopilot Design: The Generalized Predictive Adaptive Control Approach," *International Journal of Adaptive Control and Signal Processing*, Vol. 7, 1993, pp. 13–31.

[16] Bitmead, R. R., Gevers, M., and Wertz, V., *Adaptive Optimal Control*, Prentice–Hall, Englewood Cliffs, NJ, 1990, Chap. 3.

[17] Jung, Y. C., and Hess, R. A., "Precise Flight-Path Control Using a Predictive Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 5, 1991, pp. 936–942.

[18] *TMS320C4x User's Guide*, Texas Instruments, 1991.