A method for modeling error-driven adaptive control of an aircraft. Normal aircraft plant dynamics is modeled, using an original plant description in which a controller responds to a tracking error $e(k)$ to drive the component to a normal reference value according to an asymptote curve. Where the system senses that (1) at least one aircraft plant component is experiencing an excursion and (2) the return of this component value toward its reference value is not proceeding according to the expected controller characteristics, neural network (NN) modeling of aircraft plant operation may be changed. However, if (1) is satisfied but the error component is returning toward its reference value according to expected controller characteristics, the NN will continue to model operation of the aircraft plant according to an original description.

3 Claims, 11 Drawing Sheets
OTHER PUBLICATIONS


* cited by examiner
FIG. 1  Adaptive Control Architecture

Pilot Stick Input for Mach 0.75, Alt 20K - Right
Stab Lock at 4 deg - Failure Inserted at t = 10 sec

FIG. 2A

FIG. 2B
Mach 0.75, Alt 20K - Right Stab Lock at 4 deg - Failure Inserted at t = 10 sec

**FIG. 3A**

![Graph showing roll rate (q) over time](image)

**FIG. 3B**

![Graph showing normal acceleration (N) over time](image)

**FIG. 3C**

![Graph showing angle of attack (α) over time](image)
Mach 0.75, Alt 20K - Right Stab Lock at 4 deg - Failure Inserted at t = 10 sec
Mach 0.75, Alt 20K - Right Stab Lock at 4 deg - Failure Inserted at $t = 10$ sec

**FIG. 5A**

![Graph showing yaw rate (deg/s) over time (seconds)]

**FIG. 5B**

![Graph showing $\dot{\alpha}$ over time (seconds)]

**FIG. 5C**

![Graph showing $\beta$ over time (seconds)]
Mach 0.75, Alt 20K - Right Stab Lock at 4 deg - Failure Inserted at t = 10 sec

**FIG. 6A**

Roll channel performance error

**FIG. 6B**

Pitch channel performance error

**FIG. 6C**

Yaw channel performance error
Mach 0.75, Alt 20K - Right Stab Lock at 4 deg - Failure Inserted at t = 10 sec

![Graph 7A](image)

![Graph 7B](image)

![Graph 7C](image)

![Graph 7D](image)
Mach 0.75, Alt 20K - Canard Multiplier at -1 - Failure Inserted at t = 10 sec

**FIG. 8A**

![Graph of q (deg/s) vs. time (seconds)]

**FIG. 8B**

![Graph of NN ad q vs. time (seconds)]

**FIG. 8C**

![Graph of alpha (deg) vs. time (seconds)]
Mach 0.75, Alt 20K - Canard Multiplier at -1 - Failure Inserted at $t = 10$ sec

FIG. 9A

FIG. 9B

FIG. 9C
Mach 0.75, Alt 20K - Canard Multiplier at -1 - Failure Inserted at t = 10 sec

**FIG. 10A**

**FIG. 10B**

**FIG. 10C**
Mach 0.75, Alt 20K - Canard Multiplier at -1 - Failure Inserted at t = 10 sec

FIG. 11A

FIG. 11B

FIG. 11C
Mach 0.75, Alt 20K - Canard Multiplier at -1 - Failure Inserted at t = 10 sec
AIRCRAFT SYSTEM MODELING ERROR AND CONTROL ERROR

ORIGIN OF THE INVENTION

This invention was made, in part, by one or more employees of the U.S. government. The U.S. government has the right to make, use and/or sell the invention described herein without payment of compensation, including but not limited to payment of royalties.

FIELD OF THE INVENTION

This invention relates to minimization of modeling error and control error for an aircraft system.

BACKGROUND OF THE INVENTION

If an aircraft/spacecraft vehicle encounters a failure (such as a jammed control surface or loss of a surface), most controllers cannot adapt to the failure and a crash may occur. In most cases, the vehicle has enough redundant actuation mechanisms to salvage the vehicle. Several airplane crashes have occurred in the past where the pilot was unable to control the damaged airplane due to the pilot’s inability to learn to fly this altered aircraft configuration in the very short time available. The flight computer, however, may have the necessary information as well as bandwidth available to learn the new dynamics, and control the vehicle within a reasonable time interval.

The flight computer needs an intelligent controller that flies the vehicle with the baseline controller during nominal conditions, and adapts the design, when the vehicle suffers damage. Thus, given the information about the vehicle from all the available sensors, the control system needs to determine whether the vehicle is in its nominal state or is damaged. One approach to deal with this is to utilize smart algorithms that attempt to identify the vehicle characteristics and to change the control system, if necessary. This approach is known as Indirect Adaptive Control. For systems such as airplanes, there is usually very little time available to make changes to the control system, and this indirect approach is often insufficient to achieve the desired safety metrics. Another approach, known as the direct adaptive control (“DAC”), looks directly at the errors, and updates the control law accordingly. This is typically much faster and meets the timing considerations for airplane system implementations.

The current state of the art implementation consists of the Intelligent Flight Control Architecture that uses a DAC approach. This has been implemented by us at the NASA Ames Research Center, and has been test flown on the F-15 research aircraft at the Dryden Flight Research Center. The update law uses tracking error to change the control law. This approach is based on the work at the Georgia Tech Aerospace Engineering Department, under R. T. Rysdyk and A. J. Calise, “Fault Tolerant Flight Control Via Adaptive . . . Augmentation” AIAA 98-4483.

When operating in the real world, an airplane will always have tracking errors associated with its states. For example, when an pilot provides an aggressive stick command, there is always a large transient tracking error that eventually disappears as the controller continues to perform. Adaptation should typically occur only when the aircraft experiences damage or change in its flight configuration, which the original control design cannot deal with. Usually much effort goes into the design of the nominal baseline control design, which should be changed only if necessary.

What is needed is an approach that implements DAC that looks not just at the tracking error, but rather its characteristics or evolution over time to determine whether the controller needs to be adapted or left alone. The time evolution of the tracking error provides clues for investigating whether the system is in good health or has undergone damage/faults. This crucial piece of available information remains un-utilized in all the existing DAC approaches.

SUMMARY OF THE INVENTION

This invention presents a novel stable discrete-time adaptive law that is designed and implemented for flight control to targets damages/modeling errors in a direct adaptive control (DAC) framework. The approach is based on the observation that, where modeling errors are not present, the original control design has been tuned to achieve the desired performance. The adaptive control should, therefore, work towards achieving back this performance only when the design has modeling uncertainties/errors or when the vehicle suffers damage or substantial flight configuration change. In this work, the baseline controller uses dynamic inversion with proportional-integral augmentation. Dynamic inversion is carried out using the assumed system model. On-line adaptation of this control law is achieved by providing a parameterized augmentation signal to a dynamic inversion block. The parameters of this augmentation signal are updated to achieve the nominal desired error dynamics. Contrary to the typical Lyapunov-based adaptive approaches that guarantee only stability, the current approach investigates conditions for stability and for performance. A high-fidelity F-15 aircraft model is used to illustrate the overall approach.

Operationally, the aircraft plant dynamics is modeled, using the original plant description without changes, and the parameters representing the plant components are monitored. Under normal conditions, the controller responds to an excursion in the tracking error e(k), which is the difference between the desired and the actual aircraft behavior, and drives this tracking error toward a zero value according to a asymptote curve that is characteristic of the controller. If the tracking error does not conform to, or lie close to, this asymptotic curve, a resulting error (difference between desired error behavior and actual error behavior) is observed. This difference, called the performance error E(k), represents a difference between normal aircraft parameters and damaged aircraft parameters, and its components are monitored.

Assume that the system senses that (1) at least one component of aircraft tracking error e(k) is experiencing an excursion and (2) the return of this component value toward a reference value (e.g., a constant, such as 0) is not proceeding according to the expected controller characteristics (which gives rise to a non-zero magnitude IE(k)I above an expected threshold magnitude). Only when both conditions (1) and (2) are satisfied will the system reactivate the neural network (NN), change the plant dynamics according to the NN, and change the modeling of aircraft plant operation. Where condition (1) is satisfied but the return of the vector component e(k) toward the reference value proceeds according to the controller characteristics (E(k)=0), or within a selected neighborhood of this asymptote, so that condition (2) is unsatisfied, the system will not change modeling of the plant operation. In this latter instance, the NN will continue to model operation of the aircraft plant according to the original model.

In a prior art approach, as long as condition (1) is satisfied, modeling of the aircraft plant dynamics is changed, irrespec-
The time index (k+1) for this reference signal is not necessary computed at time index (k+1), but is the desired reference value of the output the time index (k+1) that is computed at time index (k). The controller is designed to achieve a prescribed second order error dynamics with respect to this reference command. Let this error dynamics, in a discrete form, be given in scalar form as:

\[ e(k+1) = K_p e(k) + K_i e_i(k), \] (1A)

\[ e(k) = y(k) - y_{ref}(k) \] (1B)

where \( e_{IA} \) represents the integrated error until time index \( k \). \( K_p \) and \( K_i \) are gains, chosen appropriately to have the desired transient response characteristics. Equation (1), with the definition of the error \( e(k) \), is used to compute the control input to achieve the desired error dynamics as follows. Equation (1) can be re-expressed as

\[ y(k+1) = y_{ref}(k+1) + K_p [y_{ref}(k) - y(k)] - K_i e_i(k) \] (2)

The plant output \( y(k+1) \) must satisfy Eq. (2) to achieve the prescribed second order error dynamics. The right hand side of Eq. (2) can thus be labeled as \( y_{ad}(k+1) \), the desired plant output. Thus,

\[ y_{ad}(k+1) = y_{ref}(k+1) + K_p [y_{ref}(k) - y(k)] - K_i e_i(k) \] (3)

Again, note that this value of the desired output at time index \( (k+1) \) is computed at time index \( (k) \). Let the plant dynamics be given as:

\[ y(k+1) = f(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)); u(k-1), \ldots, u(k-p_u)) \] (4)

\[ g(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)); u(k-1), \ldots, u(k-p_u)) \]

We can thus invert the dynamics represented by Eq. (4) to compute the control function \( u(k) \) to achieve the desired error dynamics, Eq. (1), as:

\[ u(k) = \left[ y_{ref}(k+1) + K_p [y_{ref}(k) - y(k)] - K_i e_i(k) - f(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)); u(k-1), \ldots, u(k-p_u)) \right] / g(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)) \] (5)

where \( f \) and \( g \) are functions characterizing the plant. This control input, with exact knowledge of the plant \( f \) and \( g \), will help achieve the desired second order error dynamics. With modeling uncertainties and other errors, we will not know \( f \) and \( g \) exactly, but only their estimates given by the model, \( f \) and \( g \). The adaptive augmentation is now designed to offset these modeling errors, so that we can get the same error dynamics or the desired performance. With the adaptive augmentation, as shown in FIG. 1, the desired output \( y_{ad}(k+1) \) is given as:

\[ y_{ad}(k+1) = y_{ref}(k+1) + K_p [y_{ref}(k) - y(k)] - K_i e_i(k) - y_{ad}(k) \] (6)

The control input is given as:

\[ u(k) = \left[ y_{ref}(k+1) + K_p [y_{ref}(k) - y(k)] - K_i e_i(k) - f^*(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)); u(k-1), \ldots, u(k-p_u)) \right] / g(y(k), y(k-1), \ldots, y(k-p_y; u(k-1), \ldots, u(k-p_u)) \] (7)

The control system is given a command, \( y_{con}(k+1) \) (e.g., pitch rate command from the pilot’s stick). The time index \( (k+1) \) refers to the desired value at the next time index \( (k+1) \). Given the knowledge of how fast or slow the aircraft plant can handle such a command, it is typically taken through a second order reference model, with appropriate damping and natural frequency to obtain the corresponding achievable reference command \( y_{ref}(k+1) \). It is important to note that the value for the time index \( (k+1) \) for this reference signal is not necessary.
To analyze the effect of this control input, we look at the modeling error, which is defined as the difference $e(k+1)$ between the actual plant output and that predicted by the model:

$$e(k+1) = y(k+1) - f(x(k), u(k) - p_x, \ldots, u(k-p_u))$$

Substituting the expression for the control input, given by Eq. (7), in Eq. (8) gives:

$$e(k+1) = y(k+1) - f(x(k), u(k) - p_x, \ldots, u(k-p_u)) = y(k+1)$$

In terms of the definition of the tracking error, Eq. (9) can be written as:

$$e(k+1) = y(k+1) - K_p e(k) - K_l e_i(k)$$

Equation (10) represents a key equation of this approach. The left hand side of Eq. (10) is the desired second order error dynamics. The right hand side of Eq. (10) is the difference between the modeling error and adaptive augmentation signal input. Equation (10) indicates that, if the adaptive augmentation signal can learn the modeling error and cancel this error, the error dynamics of this control loop will be restored to its desired nature. In other words, we will recapture the performance desired from this control loop. We, therefore, define the left hand side of Eq. (10) as the performance error, $E(k)$, which is more realistically expressed as a vector of performance error components.

$$E(k+1) = e(k+1) - K_p e(k) - K_l e_i(k)$$

We can now form a Lyapunov function of the performance error as:

$$V = |E(k)|^2$$

An update law now can be devised for the adaptive augmentation input, $y_{ad}$, that imposes monotonically decreasing behavior on this Lyapunov function.

Parameterization and Update Laws for the Adaptive Augmentation.

In this section, we investigate two questions. The first relates to the parameterization of the modeling error, and the second relates to the choices for designing stable update laws.

1. Parameterization for a Linear System:

Consider a linear system of the form:

$$x(k+1) = Ax(k) + Bu(k)$$

where $x$ and $u$ are vector components of the plant variables and the control inputs and $A$ and $B$ are system matrices. In a manner similar to that illustrated by Eqs. (3-5), the control input is computed as:

$$u(k) = B^{-1} [y_{ad}(k) + K_p e(k) + K_l e_i(k) - x_{ad}(k) - Ax(k)]$$

where $A$' and $B$' are estimates of the system $A$ and $B$ matrices. If the system matrices ($A$, $B$) are known, adaptive augmentation is not needed, and the control input is computed as:

$$u(k) = B^{-1} [y_{ad}(k) + K_p e(k) + K_l e_i(k) - x_{ad}(k)]$$

If these control inputs are to provide the same desired error dynamics, they must be equated, which gives the form of the idealized value of the augmentation signal $x_{ad}(k)$.

2. Parameterization for a Non-Linear System Affine in Control:

Consider a non-linear system that is affine in control, and whose dynamics can be written as linear in parameters.

$$x(k+1) = W_f (k)x(k) + W_y (k)u(k)$$

where $W_f$ is the linear dynamic weight matrix, and the vector $\beta$ corresponds to the linear and/or non-linear functions of the state system. The control input is computed in a similar manner as:

$$u(k) = B^{-1} [y_{ad}(k) + K_p e(k) + K_l e_i(k) - x_{ad}(k) - W_f (k)x(k) + W_y (k)u(k)]$$

where $W_f$ and $B$ are the corresponding estimates of the system matrices. By carrying out the analysis similar to the linear system case, the ideal augmentation signal can be computed to be:

$$x_{ad}(k) = ((I - B^* B^{-1}) W_f (k)x(k) + W_y (k)u(k))$$

Equations (17) and (20) imply that the ideal augmentation signal can be written as:

$$x_{ad}(k) = W_{x_{ad}} (k)$$

with the ideal weights, $W_{x_{ad}}$, and the basis functions, $\beta$, as given in Eqs. (16) and (19). These are the same basis functions used in Rysdyk and Calise, ibid. Thus, we can parameterize a neural network in this form, and compute the ideal weights iteratively using an appropriate update algorithm.


Having looked at the question of parameterization, we now construct a stable update law for the parameters $W_{x_{ad}}$. Parameterizing the adaptive augmentation signal in the form given by Eq. (20), and using the definition of the performance error as given in Eq. (11), Eq. (10) can be rewritten in vector form as:

$$E(k+1) = e(k+1) - x_{ad}(k)$$

Compared to Eq. (10), this is written for an error vector, $E$, corresponding to the general case of multiple control loops. Written in this form, the equation indicates that one estimates the vector modeling error, $e(k)$ (for all loops) using the adaptive augmentation signal $x_{ad}(k)$. The vector $E(k+1)$ is a corresponding error in the estimate. This error dynamics for the performance error $E(k)$ corresponds to a system identification like problem. This opens up a host of approaches for doing this online system identification. In this work, we consider a normalized gradient update approach.


Let $E(k)$ correspond to the $i$th element of the vector performance error $E(k)$. Let $W_{x_{ad}}^*$ represent the $i$th column
vector of the weight matrix $W_{a_{t}}^{*}$, which corresponds to the ideal weights that minimize the performance error vector components $E_{i}(k)$ to $\Delta^{e} = \{0^{*}, \ldots, 0^{*}\}$.

Similarly, let $W_{a_{t}}^{t}$ represent the $t$th column vector of the current estimate of the ideal weight matrix. The update law for each of these column vectors of the weight matrix is given as:

$$W_{a_{t}}^{*}(k) = W_{a_{t}}^{*}(k-1) ([\gamma E_{i}(k)^{T}B_{i}(k-1)]/[1 + P_{i}^{T}(k-1)P_{i}(k-1)])$$

(23)

The parameter $\gamma$ (Eq. (12)) corresponds to the learning rate that lies in a range $0 < \gamma < 1$.

Reference [11] proves that with this weight update law, the performance error, $E(k)$, is monotonically decreasing for all $i$. Further, it is known that if the system experiences sufficient persistent excitation, the weights $W_{a_{t}}^{*}$ approach the ideal weights $W_{a_{t}}^{*}$.

5. What Happens to Tracking Error?

The final part of this analysis investigates the behavior of the system error $e(k)$. This work provides an update only when modeling error is present, as opposed to presence of tracking error. However, tracking error is what is ultimately important. It is, therefore, appropriate to analyze the asymptotic behavior of the tracking error given the behavior of the performance error. For simplicity, in this analysis we consider the case where the desired error dynamics is first order given as:

$$E_{i}(k) - e_{i}(k+1) - K_{P_{i}}e_{i}(k) = 0$$

(25)

Let $E_{i}(k) < 0$, after time $k$, where $\delta$ is some small positive scalar. This implies

$$e_{i}(k+1) + K_{P_{i}}e_{i}(k) < 0$$

(26)

From the Cauchy-Schwarz inequality,

$$e_{i}(k+1) + K_{P_{i}}e_{i}(k) < \frac{-\delta}{1 + |K_{P_{i}}|}$$

(27)

Equations (26) and (27) imply:

$$|e_{i}(k+2)| < |K_{P_{i}}|e_{i}(k+1) + \delta$$

(28-1)

$$< |K_{P_{i}}|e_{i}(k) + \delta(1 + |K_{P_{i}}|)$$

$$e_{i}(k+\tau) < |K_{P_{i}}|e_{i}(k) + \delta(1 + |K_{P_{i}}|) + \cdots + |K_{P_{i}}|^{\tau-1}$$

(28-a)

Because $|K_{P_{i}}| < 1$ for stable error dynamics, as $k \to \infty$, $|e_{i}(k)|$ is bounded above as:

$$|e_{i}(k)| < \delta(1 - |K_{P_{i}}|)$$

(29)

Thus, if the performance error is bounded, Eq. (29) establishes bounds on the tracking errors. A similar analysis can be carried out for second order error dynamics. The result summarizes that as long as the desired error dynamics (first or second order) is stable, the tracking error will be bounded above, given that the performance error is bounded.

6. Application to Aircraft Control.

The modeling error-driven performance-seeking adaptive control design was implemented for aircraft roll, pitch, and yaw rate control. The NASA Intelligent Flight Controller (IFC) design has been tested, and is currently undergoing various modifications for being flight-tested on the research F-15 aircraft. The IFC design has been implementing the adaptive control design as outlined by Rysdyk and Calise, ibid. For implementing the performance seeking adaptive augmentation, the requirement was that it needed to fit within the existing architecture. The main issue in the implementation is that the baseline controller in the IFC architecture uses continuous-time aircraft dynamic inversion, whereas the proposed design has been outlined in the discrete-time. The equations outlined in the preceding sections have been formulated for a discrete-time model inversion. We realized, however, that after reducing the problem to the core error dynamics, the problems became identical. The error equation for the continuous-time implementation for a desired second-order error dynamics for a scalar error $e$ is given as:

$$E(k) = e_{a_{t}}(k)$$

(30)

Defining the left hand side of Eq. (31) as the modified performance error, $E'(k)$, one obtains

$$E'(k) = e_{a_{t}}(k)$$

(32)

This modified performance error equation is identical to the discrete-time version given by Eq. (21). The adaptive augmentation acceleration signal $U_{a_{t}}(k)$ can be parameterized in a similar manner, and the same update laws remain valid for the parameters of the augmentation signal for reducing $E'(k)$.

A zero value of this modified performance error restores the second order error dynamics (LHS of Eq. (31)) to zero, and thereby regains the desired performance from the control loops. Formulated in this manner, this adaptive approach fits within the existing IFC framework, and is considered as an alternate approach for flight testing. In the following discussion, we present some results of this implementation on the high-fidelity model of the modified F-15 aircraft used at the NASA Dryden Flight Research Center. The adaptive control architecture is kept the same as in the original IFC design. This design has three loops, one for each of the pitch, yaw, and roll, respectively. Adaptive augmentation is provided to each loop. Kaneshige, and Burken, “Enhancements to a Neural Adaptive Flight Control System for a Modified F-15 Aircraft,” AIAA-2008-4098, gives details on the implementation approach such as choice of the basis functions etc. The only difference is that the update law is given by Eq. (23). In this study, we look at two cases. In the first case, the right stabilator is locked at 4 degrees at $t=10$ sec into the flight experiment. In the second case, the canard multiplier is set at $-1$, again at $t=10$ sec into the flight experiment. The behavior of the aircraft and update algorithm is examined for the longitudinal and lateral pilot stick inputs given by FIGS. 2A-2B.

FIGS. 3A-3B through 7A-7D display the behavior of the aircraft and the neural net signal for right stabilator failure at $t=10$ sec. In FIGS. 3A-3C, the learning in the pitch channel is observable, when the aircraft actual pitch rate follows the desired pitch rate. FIGS. 4A-4C and 5A-5C display learning behavior in the roll and yaw channels, respectively. FIGS. 6A-6C displays the behavior of the performance errors in each of the three axes as learning proceeds. These performance errors drive the updates in each of the three axes with performance errors becoming smaller as time increases. FIGS. 7A-7D display aircraft surface commands for this
maneuver. In a similar manner, FIGS. 8A-8C, 9A-9C, 10A-10C and 11A-11C display the results for insertion of an incorrect canard multiplier. FIGS. 12A-12D display aircraft surface commands for the incorrect canard multiplier situation. Note the good learning achieved in the pitch and roll channels.

What is claimed is:

1. A method for modeling error-driven adaptive control of an aircraft, the method comprising:
   - providing a selected aircraft variable, \( y(k+1) \), at a time index having a value \( k+1 \), as a matrix sum of \( W_f \) \( \beta \) (k) and \( B \) \( u(k) \), where \( \beta \) (k) includes the at least one aircraft variable \( y(k) \) in a linear or nonlinear format, \( W_f \) is a matrix of selected aircraft variable weighting coefficients, \( u(k) \) is a control variable vector for the at least one aircraft variable, and \( B \) is a matrix of control variable weighting coefficients that is not yet known;
   - modeling an aircraft plant operation using a first neural network modeling mechanism, where the first neural network mechanism incorporates an assumption that the aircraft plant is operating within a normal range, without perturbations and without a tracking error vector \( e(k) \) that would cause the aircraft plant to experience an excursion outside a normal range of operation;
   - providing a finite bound for the tracking error vector \( e(k) \) for operation of the aircraft within the normal range;
   - when (1) at least one component of the tracking error vector \( e(k) \) is experiencing an excursion, determining if (2) return of the at least one component of the tracking error vector \( e(k) \) toward a selected reference vector does not lie on or adjacent to a selected controller error characteristic;
   - when the conditions (1) and (2) are satisfied for at least one value of the time index \( k \), introducing at least one change in at least one parameter of the neural network modeling mechanism and modeling the aircraft plant operation according to a modified neural network mechanism with the at least one changed modeling parameter; and
   - when the conditions (1) is satisfied and condition (2) is not satisfied, continuing to model the aircraft plant operation using the first neural network mechanism, with little or no change in any modeling parameter of the first neural network mechanism.

2. The method of claim 1, further comprising:
   - expressing said tracking error \( e(k) \) as a difference between a reference vector \( y_{ref}(k) \) and said vector representing said at least one aircraft variable \( y(k) \) at said time index \( k \);
   - expressing a change in said tracking error vector \( e(k) \) between said time index value \( k \) and said time index value \( k+1 \) in a tracking error vector equation
     \[
     e(k+1) + K_{pe} e(k) + K_{ie} e_i(k) = 0,
     \]
   - where \( e_i(k) \) represents an integrated tracking error up to and including said time index \( k \), and \( K_{pe} \) and \( K_{ie} \) are gain values; expressing a changed value in said aircraft variable \( y(k) \) as a sum
     \[
     y(k+1) = y_{ref}(k+1) + K_{ sire}(y_{ref}(k)+\Delta b\tau e_i(k));
     \]
   - interpreting said control relationship as a linear or affine relationship
     \[
     y(k+1) = A y(k) + B u(k),
     \]
   - providing estimates, \( A^* \) and \( B^* \), of the matrix \( A \) and said matrix \( B \);
   - expressing said control variable vector \( u(k) \) as a first sum
     \[
     u(k) = B^{-1} \{ y_{ref}(k+1) + K_{w} e(k) + \Delta s(k) \};
     \]
   - expressing said control variable \( u(k) \) as a second sum
     \[
     u(k) = B^{-1} \{ y_{ref}(k+1) + K_{w} e(k) + \Delta s(k) \} - A y(k);
     \]
   - where \( y_{ref}(k) \) is an adaptive augmentation vector that is included to encourage said aircraft to satisfy the tracking error vector equation:
     \[
     y_{ad}(k) = \{ B B^{-1} A - B y(k) \} - ((I B - B^{-1} A - A) y_{ref}(k+1) + K_{w} e(k) + \Delta s(k)) - (A y(k));
     \]
   - where \( f \) is a function of one or more of the variables \( y(k') \) (\( k' = k, k-1, \ldots, k-p, \)) and one or more of the variables \( u(k') \), \( g \) is a function of one or more of the variables \( y(k') \) and one or more of the variables \( u(k') \), and \( f \) and \( g \) characterize the aircraft plant and are as yet unknown;
   - inventing the control relation to express the control vector \( u(k) \) in a form needed to achieve a desired aircraft control dynamics as
     \[
     (A - B B^{-1} A - A) y_{ref}(k+1) + K_{w} e(k) + \Delta s(k) - (y(k) - y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*)) - f[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))],g[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))];
     \]
   - providing an approximate model of the aircraft plant dynamics, with corresponding estimates, \( f^* \) and \( g^* \), for the respective functions \( f \) and \( g \);
   - expressing an estimate \( u^*(k) \) of a control vector that will achieve a desired aircraft control dynamics as
     \[
     u^*(k) = (y_{ref}(k+1) + K_{w} e(k) + \Delta s(k) - f[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))],g[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))];
     \]
   - and providing an estimate \( e'(k) \) of said tracking error as said tracking error
     \[
     e'(k) = y(k+1) - f'[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))],g[y(k), y(k-1), \ldots, y(k-p, u(k-1), \ldots, u(k-p,*))].
     \]