Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Cascaded Position and Heading Control of a Robotic Helicopter

Marcel Bergerman, Omead Amidi, James Ryan Miller, Nicholas Vallidis, Todd Dudek
Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213, USA
email: marcel@cmu.edu

Abstract— **We present a cascaded control architecture for a Yamaha RMAX robotic helicopter. The controller is composed of an inner-loop that stabilizes the unstable poles of the helicopter's linear dynamic model; and an outer-loop that decouples the dynamics of the lateral, longitudinal, vertical, and heading axes and enables trajectory tracking. Actual flight results are presented to demonstrate the validity of the method. A discussion on the method's limitations and our plans on how to overcome them are also presented.**

I. Introduction and Problem Statement

A robotic helicopter is an aircraft equipped with a sensing, computing, actuation, and communication infrastructure that allows it to execute a variety of tasks in autonomous mode. Tasks may range from simple monitoring with a video camera to sophisticated terrain modeling and identification/tracking of individuals and objects, to name a few. A ground station is used to issue mission plans to the helicopter, and to receive and record telemetry and sensor data. Mission plans may be as low-level as a set of waypoints through which the helicopter must fly or as high-level as "find all specimens of baobab[1] within the boundaries of this national park and return their geographic coordinates."

In this article we are concerned with the problem of controlling the position and heading of a robotic helicopter to make it fly along a pre-planned trajectory. Our focus is on the Yamaha RMAX, one of the most sophisticated remote control helicopters that can be retrofitted as a robotic vehicle, in use by the Carnegie Mellon [3], Georgia Institute of Technology [6], UC Berkeley [8], and NASA [14] robotic helicopter research groups. The trajectory is assumed to be safe in the sense that it respects pre-defined bounds on the helicopter's velocities and accelerations, and in that it does not pose the risk of a collision with obstacles. Helicopter obstacle avoidance is dealt with, for example, in [13].

The RMAX control problem is made difficult by the unstable, nonlinear, and coupled nature of this machine. In the early days of robotic helicopter control, this problem was solved by proportional-integral-derivative (PID) controllers in a two-loop architecture. In this type of controller, the inner-loop is responsible for stabilization of the roll and pitch axes, and the outer-loop is responsible for tracking in the x, y, z, and yaw control axes. These PID-type controllers were implemented and are still in use in a variety of robotic helicopter projects that use the RMAX. Because these controllers do not take into account the dynamics of the helicopter, especially the coupling between the control axes, they offer limited performance. As dynamic models of a similar robotic helicopter, the Yamaha R50, became available [11], model-based controllers were proposed and demonstrated to be practical. While some of these model-based controllers were only tested in simulation environments [8], others were validated in actual field experiments [11]. One of the most sophisticated model-based controllers to be flown on an actual RMAX is probably the H_∞-based controller by La Civita [10]. Much like it PID predecessor, this controller is also based on a cascaded architecture, where a multiple-input-multiple-output (MIMO) inner-loop stabilizes the helicopter's attitude, and four independent single-input-single-output (SISO) controllers are responsible for trajectory tracking. Again, this architecture ignores the coupling between the four control axes of the RMAX, and although it provided for very good flight performance, it can still be enhanced by considering the full dynamic coupling of the aircraft. Learning controllers have also been proposed in the literature, albeit for smaller helicopters (see, e.g., [1], [2]).

We propose an original model-based cascaded position and heading controller for a robotic RMAX helicopter. Much like other works published in the literature, the controller is also based on an inner-loop that stabilizes the poles of the helicopter's dynamic model. Our choice for this loop is a linear quadratic regulator (LQR), which brings the unstable poles of the RMAX model to the left-hand plane. Unlike past work, however, in the outer-loop we fully take into account the coupling between the control axes to design a feedback linearization controller (FLC). This combination of a model-based LQR and a model-based FLC form the basis of our contribution to the state-of-the-art. Actual flight data show promising results, and indicate that we can expect even more impressive results as we improve our dynamic models of the Yamaha RMAX to refine the controllers.

II. Controller Design Model

The cascaded controller design is based on Cheng et al.'s linear dynamic model for the RMAX in hover mode [4], given by:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{1}$$

where the state $\mathbf{x}$ and input $\mathbf{u}$ are given by:

[1]The baobab are trees native to Madagascar, Africa and Australia.

1-4244-0912-8/07/$25.00 ©2007 IEEE.

$$\mathbf{x} = \begin{bmatrix} u^B & v^B & w^B & p & q & r & r_{fb} \\ & & & \beta_{1c} & \beta_{1s} & \kappa_{1c} & \kappa_{1s} & \dot{\beta}_0 & \beta_0 & \nu \end{bmatrix}^T \quad (2)$$

$$\mathbf{u} = \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{col} & \delta_{ped} \end{bmatrix}^T \quad (3)$$

and the individual state and input components represent:

- u^B, v^B, w^B: linear velocities of the helicopter in body frame.
- p, q, r: angular velocities of the helicopter.
- r_{fb}: yaw rate feedback (a gyro-like mechanism internal to the Yamaha RMAX).
- β_{1c}, β_{1s}: lateral and longitudinal cyclic pitch of the main blades.
- κ_{1c}, κ_{1s}: lateral and longitudinal cyclic pitch of the stabilizer bar.
- β_0: main blades' coning angle.
- ν: inflow.
- δ_{lat}: lateral command.
- δ_{lon}: longitudinal command.
- δ_{col}: collective command.
- δ_{ped}: pedal (yaw) command.

We refer the reader to the literature on helicopter design and control for a full explanation of the seven last elements of the state vector, as well as the elements of the control vector [7]. Here it suffices to say that the lateral command generates rolling moments, the longitudinal command generates pitching moments, the collective command generates vertical force, and the pedal (yaw) command generates yawing moments, all with respect to the helicopter's center of gravity.

To be able to control directly the position and heading (yaw) angle of the helicopter, we need to add its position and heading angle to the state vector. Note that if we add the position in the navigation frame, $\begin{bmatrix} x^N & y^N & z^N \end{bmatrix}^T$, the system becomes nonlinear, as the position in navigation frame is obtained by pre-multiplying the position in body frame by a nonlinear rotation matrix. Therefore, we chose to add the position in body frame to the state vector, $\begin{bmatrix} x^B & y^B & z^B \end{bmatrix}^T$, to keep the model linear. (We will later show how the desired trajectory, normally defined in navigation frame, can be easily transformed to the body frame for guidance and control purposes.) On the same token, instead of adding to the model the exact, nonlinear transformation from angular rates to attitude (which includes the heading angle), we chose to approximate the attitude angles as the time integral of the angular rates. This approximation is valid as long as the helicopter flies with small ($< 10^o$) roll and pitch angles, which is true most of the time in non-aggressive flight. Note that the accuracy of this approximation (and of the entire state estimation process) is dependent on the quality of the sensors used. In our case the RMAX carries a high-precision Litton inertial measurement unit, which provides very accurate angular rate measurements in all three axes.

With these considerations taken into account we extend the state vector to include the following variables:

$$x^B = \int u^B dt, y^B = \int v^B dt, z^B = \int w^B dt \quad (4)$$

$$\phi \approx \int p\, dt, \theta \approx \int q\, dt, \psi \approx \int r\, dt \quad (5)$$

Therefore,

$$\mathbf{x} = \begin{bmatrix} x^B & y^B & z^B & u^B & v^B & w^B \\ & \phi & \theta & \psi & p & q & r & r_{fb} \\ & \beta_{1c} & \beta_{1s} & \kappa_{1c} & \kappa_{1s} & \dot{\beta}_0 & \beta_0 & \nu \end{bmatrix}^T \quad (6)$$

Note that, by the definition above, x^B, y^B, z^B represent the position of the helicopter in a frame which rotates with the helicopter's body but whose origin is the same as the origin of the navigation frame; ϕ, θ, ψ are the helicopter's roll, pitch, and yaw attitude angles.

In the model given by Equations (1), (3), and (6), some states are not directly measured; they are the yaw rate feedback, the main blades' and the stabilizer bar's cyclic pitch, the coning angle, and the inflow. Coincidentally, these states are also the ones associated with the least dominant poles of the system, and therefore produce the least modeling error if eliminated from the model. We therefore generated a lower-order model with only the helicopter's position and attitude, linear and angular velocities, and yaw rate feedback:[2]

$$\mathbf{x}_c = \begin{bmatrix} x^B & y^B & z^B & u^B & v^B & w^B \\ & \phi & \theta & \psi & p & q & r & r_{fb} \end{bmatrix}^T \quad (7)$$

The helicopter's control synthesis design model is, finally, given by:

$$\dot{\mathbf{x}}_c = A_c \mathbf{x}_c + B_c \mathbf{u}_c \quad (8)$$

$$\mathbf{y}_c = C_c \mathbf{x}_c \quad (9)$$

with

$$\mathbf{u}_c = \mathbf{u} = \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{col} & \delta_{ped} \end{bmatrix}^T$$

$$\mathbf{y}_c = \mathbf{y} = \begin{bmatrix} x^B & y^B & z^B & \psi \end{bmatrix}^T$$

In Equations (8)-(9) the subscript 'c' indicates that this is the model that we will use for control synthesis purposes. The elements of the matrices in (8) can be found in [4].

As is usual in helicopter modeling, this control synthesis design model is unstable, as can be easily verified by computing its poles (the eigenvalues of A_c) and plotting them in the complex plane. Figure 1 shows the poles of this system that are near the origin of the complex plane. Note the unstable poles on the right-hand plane, corresponding to the pitch axis.

[2]The yaw rate feedback cannot be disregarded because it is electromechanically coupled with the yaw dynamics, which is a dominant pole of the system. We obtain this variable using estimation techniques as shown in Section III-E.

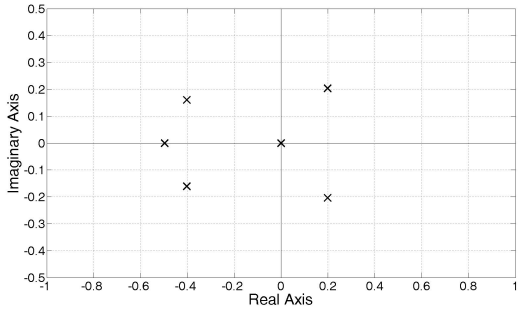

Fig. 1. Near-origin poles of the (unstable) nominal control system design model (Equations (7), (8), (9)).

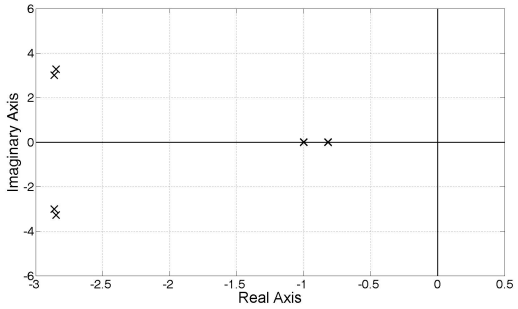

Fig. 2. Near-origin poles of the nominal control system design model (Equations (7), (8), (9)) with LQR state feedback (11).

III. Control Methodology

Our approach to the problem of controlling the RMAX's position and heading is based on the cascaded combination of three well-established control methodologies. Initially, we use a linear quadratic regulator to stabilize the right-hand plane poles; then we use a feedback linearization controller to decouple the input/output pairs; finally, we use a proportional-derivative controller to enable trajectory tracking. Each of these techniques are described in the sequel.

A. Linear Quadratic Regulator

The linear quadratic regulator (LQR for short) has its origins in the optimal control theory field. Optimal control is concerned with the optimization (usually minimization) of a system's energy or energy-like function, when the system evolves from an initial state $\mathbf{x}(t_0)$ to a final state $\mathbf{x}(t_f)$. In particular, consider a dynamic system described by Equation (8), and assume the system must reach a desired state $\mathbf{x}^d = \mathbf{0}$ while the following energy-like cost function is minimized:

$$J = \int_{t_0}^{\infty} \left(\mathbf{x}_c^T Q \mathbf{x}_c + \mathbf{u}_c^T R \mathbf{u}_c \right) dt \tag{10}$$

The LQR methodology finds a fixed gain matrix K_{LQR} such that the state feedback control law

$$\mathbf{u}_c = -K_{\mathrm{LQR}} \mathbf{x}_c \tag{11}$$

minimizes J while the system evolves along (1) to $\mathbf{x}^d$. The (usually) diagonal, positive-definite matrices Q and R define the "contribution" from each input and state to the cost J; the larger a term is in either Q or R, the more the LQR will try to minimize the corresponding input or state. "Large" matrices R will therefore tend to "save" energy by applying small inputs to the system, and "small" matrices R will have the opposite effect.

In practice, one of the side effects of applying the LQR methodology is that the poles of the system change from those given by the matrix A_c to those given by $A_c - B_c K_{\mathrm{LQR}}$, since

$$\begin{aligned} \dot{\mathbf{x}}_c &= A_c \mathbf{x}_c + B_c \mathbf{u}_c = \\ &= A_c \mathbf{x}_c - B_c K_{\mathrm{LQR}} \mathbf{x}_c = \\ &= \left(A_c - B_c K_{\mathrm{LQR}} \right) \mathbf{x}_c \end{aligned} \tag{12}$$

In our case, this results in all poles being moved to the left-hand side of the complex plane, thus rendering the combined dynamic model + LQR state feedback a (mathematically, at least) stable system. This can be seen in Figure 2, where we show the near-origin poles of the control design system (Equations (7), (8), (9)) under the LQR state feedback, with K_{LQR} computed with the Matlab command `lqr` for $Q = \mathrm{diag}(1,1,\ldots,1)$ and $R = \mathrm{diag}(1,1,\ldots,1)$. Note that the originally unstable poles are now in the left-hand plane.

Since the control design system is stable under the LQR state feedback, we can now utilize other control methodologies to drive the system to any admissible output. We choose here the technique known as feedback linearization, as explained in the sequel.

B. Feedback Linearization Controller

The feedback linearization control methodology (FLC) has its origins in the nonlinear control theory field. It finds a state transformation and a state feedback law that (mathematically) decouples and linearizes the system, thus allowing one to (again, mathematically) control each output independently of all others with a simple linear controller (e.g., a PD controller). This is particularly interesting for the position and heading control of a helicopter, where all axis (and especially the lateral and longitudinal ones) are highly coupled. From a simplified point of view, the methodology requires one to compute successive time derivatives of the system's output, until all inputs appear explicitly in the resulting differential equations. At this point, the equation is inverted and the input is computed as an explicit function of the system's matrices and state, as shown in the sequel. (For a more formal description of the FLC and the Lie algebra concepts behind it, we refer the reader to [5]).

We start with the control synthesis model (Equations (7), (8), (9)):

$$\dot{\mathbf{x}}_c = A_c\mathbf{x}_c + B_c\mathbf{u}_c \quad (13)$$
$$\mathbf{y}_c = C_c\mathbf{x}_c \quad (14)$$

(Recall that the reduced state $\mathbf{x}_c$ is given by Equation (7) and that $\mathbf{y}_c = \mathbf{y}$, $\mathbf{u}_c = \mathbf{u}$.) The control input $\mathbf{u}_c$ is chosen as a combination of the LQR state feedback and an auxiliary input $\mathbf{v}_c$:

$$\mathbf{u}_c = \mathbf{v}_c - K_{\mathrm{LQR}}\mathbf{x}_c$$

Therefore, the system can be described by:

$$\dot{\mathbf{x}}_c = (A_c - B_cK_{\mathrm{LQR}})\,\mathbf{x}_c + B_c\mathbf{v}_c \quad (15)$$
$$\mathbf{y}_c = C_c\mathbf{x}_c \quad (16)$$

We now compute $\dot{\mathbf{y}}_c$ as:

$$\begin{aligned}\dot{\mathbf{y}}_c &= C_c\dot{\mathbf{x}}_c = \\ &= C_c\,(A_c - B_cK_{\mathrm{LQR}})\,\mathbf{x}_c + C_cB_c\mathbf{v}_c \\ &= C_c\,(A_c - B_cK_{\mathrm{LQR}})\,\mathbf{x}_c \end{aligned} \quad (17)$$

because, for this particular system, C_cB_c is equal to the null matrix. Since $\mathbf{v}_c$ does not appear explicitly in Equation (17), we differentiate the output once again to obtain:

$$\begin{aligned}\ddot{\mathbf{y}}_c &= C_c\,(A_c - B_cK_{\mathrm{LQR}})\,\dot{\mathbf{x}}_c = \\ &= C_c\,(A_c - B_cK_{\mathrm{LQR}})^2\,\mathbf{x}_c + \\ &\quad C_c\,(A_c - B_cK_{\mathrm{LQR}})\,B_c\mathbf{v}_c \end{aligned} \quad (18)$$

Now $C_c\,(A_c - B_cK_{\mathrm{LQR}})\,B_c$ is a square, full-rank matrix, and thus invertible. If we make $\mathbf{v}_c$ equal to the following state feedback-like term:

$$\begin{aligned}\mathbf{v}_c &= [C_c\,(A_c - B_cK_{\mathrm{LQR}})\,B_c]^{-1}\cdot \\ &\quad \cdot\left[\bar{\mathbf{v}}_c - C_c\,(A_c - B_cK_{\mathrm{LQR}})^2\,\mathbf{x}_c\right] \end{aligned} \quad (19)$$

then the closed-loop behavior of $\mathbf{y}_c$ is dictated by $\ddot{\mathbf{y}}_c = \bar{\mathbf{v}}_c$.

C. PD Controller

Finally, by choosing $\bar{\mathbf{v}}_c$ as a proportional plus derivative control of the form

$$\bar{\mathbf{v}}_c = \ddot{\mathbf{y}}_c^{\mathrm{ref}} + K_d\left(\dot{\mathbf{y}}_c^{\mathrm{ref}} - \dot{\mathbf{y}}_c\right) + K_p\left(\mathbf{y}_c^{\mathrm{ref}} - \mathbf{y}_c\right) \quad (20)$$

where $\mathbf{y}_c^{\mathrm{ref}}$ is the desired (reference) output, the overall dynamic behavior of the error $\mathbf{e}_c = \mathbf{y}_c^{\mathrm{ref}} - \mathbf{y}_c$ becomes:

$$\ddot{\mathbf{e}}_c + K_d\dot{\mathbf{e}}_c + K_p\mathbf{e}_c = \mathbf{0} \quad (21)$$

In the absence of modeling errors, a careful selection of the gain matrices K_p and K_d guarantees that $\mathbf{e}_c \to \mathbf{0}$, or, in other words, that $\mathbf{y}_c$ converges to its reference value with a convergence rate determined by the elements of the gain matrices. If we choose them diagonal, positive definite matrices, then each output is controlled independently of all others and its dynamic behavior can be assigned arbitrarily. Now because $\mathbf{y}_c = \mathbf{y}$ and $\mathbf{u}_c = \mathbf{u}$ by construction, Equation (21) determines that the helicopter's position in the ground-fixed body frame, as well as its heading, converge to the desired references.

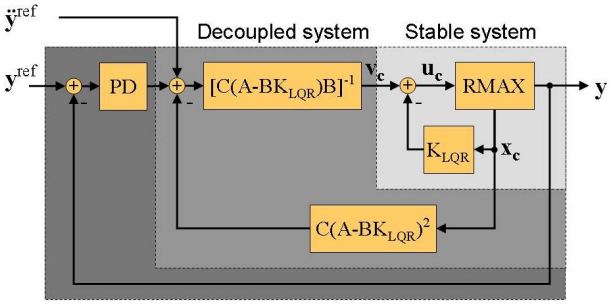

Fig. 3. System-level block diagram of the cascaded controller.

D. Cascaded Control Law

In summary, the overall controller is given by:

$$\begin{aligned}\mathbf{u}_c &= [C_c\,(A_c - B_cK_{\mathrm{LQR}})\,B_c]^{-1}\cdot \\ &\quad \cdot\left[\ddot{\mathbf{y}}_c^{\mathrm{ref}} + K_d\dot{\mathbf{e}}_c + K_p\mathbf{e}_c + \right. \\ &\quad \left. -C_c\,(A_c - B_cK_{\mathrm{LQR}})^2\,\mathbf{x}_c\right] + \\ &\quad -K_{\mathrm{LQR}}\mathbf{x}_c \end{aligned} \quad (22)$$

Because all matrices appearing in Equation (22) can be computed off-line, this control law can be easily implemented in real time. Figure 3 presents a block diagram of the controller.

In practice, of course, the outputs cannot be exactly decoupled because the model we used to synthesize the controller only represents a portion of the helicopter's full dynamic behavior. As we will see in the sequel, however, the controller's performance is quite acceptable.

E. State Estimation

For practical implementation purposes, we need to know all states in $\mathbf{x}_c$. The only state not directly measurable is the yaw rate feedback, r_{fb}. To obtain an estimate for its value, we start by reproducing the corresponding dynamic equation from the original model in [4]:

$$\dot{r}_{fb} = K_r r + K_{r_{fb}} r_{fb} \quad (23)$$

Because this is a stable, first-order filter, we can easily compute an estimate $\hat{r}_{fb}$ with the following recursive formulation:

$$\dot{\hat{r}}_{fb} = K_r r + K_{r_{fb}}\hat{r}_{fb},\ \hat{r}_{fb}(0) = r_{fb}(0) \quad (24)$$

Additionally, because the time constant of this filter is very small compared to the dominant ones in the control design system, we can safely assume that the time derivative of $\hat{r}_{fb}$ is negligible and therefore:

$$\hat{r}_{fb} = -\frac{K_r}{K_{r_{fb}}}r \quad (25)$$

Equation (25) was used in the real-time code implemented onboard the RMAX for the purposes of validating the cascaded controller.

F. Waypoint Following in Ground-Fixed Body Frame

Recall that the triplet x^B, y^B, z^B in the state vector $\mathbf{x}_c$ represents the position of the helicopter in a frame which rotates with the helicopter's body but whose origin is the same as the origin of the navigation frame. Because guidance waypoints are usually defined in the navigation frame, which does not rotate but is rigidly fixed on Earth, we must transform the waypoints to the ground-fixed body frame in which the controller was designed. To do this we simply use the fact that the position in the body frame is equal to:

$$\begin{bmatrix} x^B \\ y^B \\ z^B \end{bmatrix} = C_N^B \begin{bmatrix} x^N \\ y^N \\ z^N \end{bmatrix} \tag{26}$$

where C_N^B is the direction cosine matrix from navigation to body frame, given by (with the usual convention $s\alpha = \sin(\alpha), c\alpha = \cos(\alpha)$):

$$\begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ s\phi s\theta c\psi - c\phi s\psi & s\phi s\theta s\psi + c\phi c\psi & s\phi c\theta \\ c\phi s\theta c\psi + s\phi s\psi & c\phi s\theta s\psi - s\phi c\psi & c\phi c\theta \end{bmatrix}$$

The waypoint following in the ground-fixed body frame x^B, y^B, z^B is then achieved by using as reference output

$$\left(\mathbf{y}_c^{\text{ref}}\right)^B = \begin{bmatrix} C_N^B \\ 0\,0\,0\,1 \end{bmatrix} \left(\mathbf{y}_c^{\text{ref}}\right)^N$$

IV. Experimental Setup and Validation

To validate the cascaded controller proposed we utilized one of the four Yamaha RMAX owned by the CMU Autonomous Helicopter Laboratory. These vehicles have been instrumented over the years with a high-precision inertial navigation system, including two Novatel GPS receivers, a Litton inertial measurement unit, and a custom-designed Pentium-based processing computer, which runs a 13-dimensional state estimation Kalman filter. The system is capable of 2.5 cm position estimation and 0.01 degree attitude estimation accuracy. Figure 4 shows the aircraft used in the experiments described in this section. The two GPS antennas can be seen at the front and aft of the helicopter. The large white box on the port (left) side houses the navigation, guidance, and control computer. An identical box on the starboard (right) side houses the laser processing computer, used for 3D terrain mapping. Both computers run the VxWorks real-time operating system, with processes running at 100 Hz.

We validated the cascaded controller methodology in three steps: (i) in Matlab, with the RMAX represented by the full dynamic model (Equations (1), (3), (6)); (ii) in our hardware-in-the-loop simulation box, also with the RMAX represented by the full dynamic model; and (iii) during actual flights. Due to space restrictions we present here only the actual flight results, which are of course the most relevant.

Fig. 4. Yamaha RMAX.

The feedback linearization controller's gains were set, by trial and error, to $K_p = \text{diag}\,(1,1,3,50)$, $K_d = \text{diag}\,(2,2,3,5)$. Initially the results were not satisfactory, because the helicopter's roll and pitch angles oscillated with a high amplitude (on the order of 10^o) while the helicopter hovered. These oscillations may be related to the unstable non-zero dynamics reported by Koo and Sastry [9], and will be further studied in the continuation of this project. For the time being we limited the control action in each axis by creating artificial gains that multiply each of the elements in $\mathbf{u}_c$:

$$\begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{col} & \delta_{ped} \end{bmatrix}^T = K_u \mathbf{u}_c$$

We set $K_u = \text{diag}\,(0.2, 0.2, 0.75, 1.0)$ and obtained satisfactory results: the helicopter hovered stably and was ready to receive trajectory following commands.

We initially commanded flights along straight longitudinal (forward-backward) and lateral (sideways) paths, while keeping constant altitude and a stationary heading, at speeds up to 8 m/s. These trajectories excite very little the coupled longitudinal-lateral dynamics, and therefore were expected to be followed quite "easily" by the cascaded controller, which indeed was the case. The next step was to fly a more dynamically challenging trajectory, which does excite the coupling in the control axes and therefore pushes the controller to a larger extent. (Remember that the cascaded controller was designed on the basis of the hover model of the RMAX, and theoretically should be expected to perform well mostly at hover and low speeds.)

Figure 5 presents results collected on March 30th, 2007 with the helicopter flying an 8-shaped path. The path is generated with third-order (cubic) splines that connect waypoints manually defined by the ground operator. Maximum speed and acceleration along the path are also defined by the operator, and used by a trajectory generation module to define the time separation of the points along the splines. In this test trajectory the helicopter must fly always "looking" ahead, i.e., with heading tangential to the trajectory, and must hold a constant altitude; the speed along the path is 3 m/s. The trajectory was specially designed to excite the coupling in the aircraft, with turns to the right and to the left executed simultaneously with forward flight, altitude holding, and constant heading changes. The average tracking error along the trajectory is 1.5 m. While there is definitely room for improvements, this result can be considered very

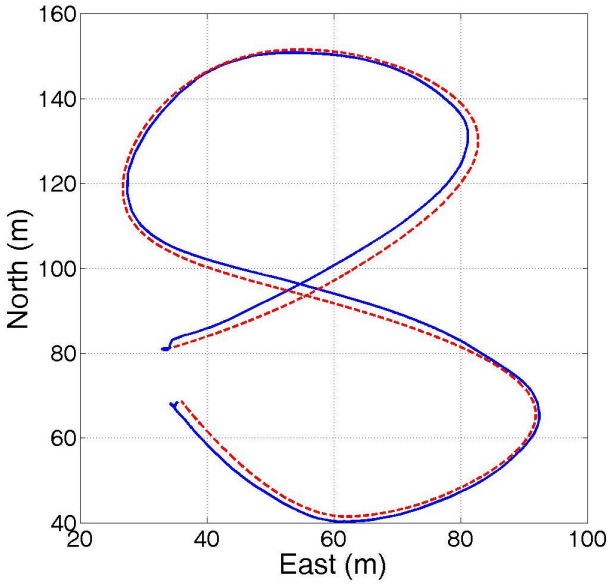

Fig. 5. Field validation of the cascaded controller flying an 8-shaped trajectory. Actual flight path is the solid blue line, while the desired path is the dashed red line.

satisfactory for a controller designed on top of a very simple dynamic model of the RMAX.

V. Conclusion

The cascaded controller designed and validated in the field proves that it is possible to utilize a rather simple dynamic model of the Yamaha RMAX for model-based control design. In fact, it is the author's understanding that this work is the first controller design and validation effort presented in the literature based on Cheng's model. This, and the multiple-input-multiple-output nature of both the inner- and the outer-loop of the controller are the main contributions of this article to the state-of-the-art.

The assumptions we made naturally limit the applicability of the cascaded controller. Remember that the base dynamic model was obtained by Cheng et al. for hover maneuvers, and should not be expected to work well at very high speeds (15 m/s or more). We successfully flew the helicopter at 8 m/s, but only over "dynamically decoupled" trajectories (straight lines). When flying the 8-shaped figure at speeds greater than 4 m/s, the helicopter oscillated substantially along the trajectory, indicating poor decoupling between the control axes. We intend to use Cheng's forward flight model as the basis for a controller that we anticipate will allow for more stable high-speed flight. Remember also that we chose to maintain the nominal control system design model linear by approximating the attitude angles as the integrals of the attitude rates (Equation (5)). Since this is true only for roll and pitch angles not exceeding 10^o, we cannot expect the cascaded controller to perform aggressively over trajectories requiring high accelerations (since, in a helicopter, roll and pitch angles are directly proportional to lateral and longitudinal accelerations, respectively). We will relax this approximation and investigate how to extend the current controller to the resulting nonlinear model.

Our future work will focus also on the improvement of the cascaded controller by including the states not directly measured (the main blades' and stabilizer bar's cyclic pitch, the coning angle, and the inflow) in the control synthesis model. We anticipate that this can be accomplished in a straightforward way using established estimation techniques. In the sequel we will develop more complex, nonlinear models of the RMAX which include the aerodynamic effects associated with a rigid body flying through a fluid, and a corresponding nonlinear control design and validation effort.

VI. Acknowledgments

The first author wishes to thank the RMAX safety pilot, Todd Dudek, whose unmatched skills and patience allowed him to execute numerous flight tests with great peace of mind.

References

[1] P. Abbeel, V. Ganapathi, and A. Ng. "Learning vehicular dynamics, with application to modeling helicopters." *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schlkopf. and J. Platt (eds.), MIT Press, Cambridge, MA, 2006, pp. 1-8.

[2] P. Abbeel, A. Coates, M. Quigley, and A.Y. Ng. "An application of reinforcement learning to aerobatic helicopter flight." *Advances in Neural Information Processing Systems 19*, B. Schlkopf, J. Platt, and T. Hoffman (eds.), MIT Press, Cambridge, MA, 2007, pp. 1-8.

[3] O. Amidi, T. Kanade, K. Fujita. "A visual odometer for autonomous helicopter flight." *Robotics and Autonomous Systems*, vol. 28, no. 2-3, August 1999, pp. 185-193.

[4] R.P. Cheng, M.B. Tischler, and G.J. Schulein. "RMAX helicopter state-space model identification for hover and forward-flight." *Journal of the American Helicopter Society*, April 2006, pp. 202-210.

[5] A. Isidori. *Nonlinear control systems*. Springer Verlag, Berlin, 1995, 3rd ed.

[6] E.N. Johnson and S.K. Kannan. "Adaptive trajectory control for autonomous helicopters." *AIAA Journal of Guidance, Control, and Dynamics*, vol. 28, no. 3, May/June 2005, pp. 524-538.

[7] W. Johnson. *Helicopter theory*. Dover, New York, 1994.

[8] H.J. Kim, D.H. Shim, and S. Sastry. "Flying robots: modeling, control and decision making." *IEEE International Conference on Robotics and Automation*, Washington, D.C., USA, May 2002, pp. 66-71.

[9] T.J. Koo and S. Sastry. "Output tracking control design of a helicopter model based on approximate linearization." *IEEE Conference on Decision and Control*, Tampa, FL, USA, December 1998, pp. 3635-3640.

[10] M. La Civita, G. Papageorgiou, W.C. Messner, and T. Kanade. "Design and flight testing of a high bandwidth H_∞ loop shaping controller for a robotic helicopter." *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, August 2002.

[11] B. Mettler, M.B. Tischler, and T. Kanade. "Attitude control optimization for a small-scale unmanned helicopter." *AIAA Guidance, Navigation and Control Conference*, 2000.

[12] J.R. Miller, O. Amidi, C. Thorpe, and T. Kanade. "Precision 3-D modeling for autonomous helicopter flight." *International Symposium of Robotics Research*, Snowbird, Utah, USA, October 1999.

[13] S. Scherer, S. Singh, L. Chamberlain, and S. Saripalli. "Flying fast and low among obstacles." *IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007.

[14] M. Whalley, M. Takahashi, G. Schulein, M. Freed, D. Christian, A. Patterson-Hine, and R. Harris. "The Army/NASA Autonomous Rotorcraft Project." *Proceedings of the American Helicopter Society 59th Annual Forum*, Phoenix, AZ, May 2003.