J Intell Robot Syst (2010) 57:351–369
DOI 10.1007/s10846-009-9366-2

Autonomous Autorotation of Unmanned Rotorcraft using Nonlinear Model Predictive Control

Konstantinos Dalamagkidis · Kimon P. Valavanis · Les A. Piegl

Received: 1 February 2009 / Accepted: 1 August 2009 / Published online: 16 September 2009

Abstract Safe operations of unmanned rotorcraft hinge on successfully accommodating failures during flight, either via control reconfiguration or by terminating flight early in a controlled manner. This paper focuses on autorotation, a common maneuver used to bring helicopters safely to the ground even in the case of loss of power to the main rotor. A novel nonlinear model predictive controller augmented with a recurrent neural network is presented that is capable of performing an autonomous autorotation. Main advantages of the proposed approach are on-line, real-time trajectory optimization and reduced hardware requirements.

Keywords Autonomous autorotation · Unmanned rotorcraft · Nonlinear model predictive control

Nomenclature

A	Rotor disc area
$\mathcal{C}$	Constraint function
C_d	Blade drag coefficient
$C_{d,0}$	Average blade drag coefficient
C_l	Blade lift coefficient
$C_{l,a}$	Lift curve slope
C_T	Thrust coefficient
E	Neural network epoch size
f_e	Equivalent unit drag coefficient area

K. Dalamagkidis (✉) · L. A. Piegl
Computer Science and Engineering Department, University of South Florida,
4202 East Fowler Avenue, ENB 118, Tampa, FL 33620, USA
e-mail: kdalamag@mail.usf.edu

K. P. Valavanis
Department of Electrical and Computer Engineering, University of Denver,
Clarence M. Knudson Hall, 300, 2390 S. York Street, Denver, CO 80208, USA

f_i Inflow velocity factor
f_r Ratio of maximum to nominal rotor rpm
g Acceleration of gravity
I_R Rotor moment of inertia
J Jacobian matrix
L Cost function
M Helicopter mass
N_c Control horizon
N_s Prediction horizon
R Rotor radius
T Thrust
t_f Final time
t_s Sampling period
u Action
v Velocity
v_H Helicopter sink rate
w Weight factor
χ Auxiliary vector
x State
z Helicopter altitude
z_0 Helicopter initial altitude

Greek Letters

γ Learning rate
θ Blade pitch at 3/4 of its length
κ Induced power correction factor
λ Inflow ratio
ρ_α Air density
σ Rotor solidity factor
Ω Rotor speed of rotation
Ω_0 Nominal rotor speed of rotation

Subscripts

h At hover
i Induced
M Maximum
m Minimum
s Steady-state

1 Introduction

Small unmanned rotorcraft are very attractive both in the military and the civilian application domains mainly due to two key characteristics; maneuverability and portability. Specifically helicopters can maneuver in tight spaces, hover over areas of interest and can take-off from and land almost anywhere. On the other hand they have been described as “ungainly, aerodynamic mavericks” [4], since they exhibit a complex aerodynamic performance that is extremely difficult to accurately predict.

Penetration of such systems into the market is contingent on resolution of safety issues among others. This paper will not go into details on these issues, the interested reader is referred to the literature [2, 5–7, 9, 10, 16]. Nevertheless, solution of such issues will undoubtedly involve provisions to handle on-board failures in a manner that minimizes the risks to the public and third party property to acceptable levels. The focus of this work is on failures that would jeopardize or completely preclude continued flight under manual control, but can be accommodated by an on-board emergency flight controller. Such failures include loss of power or control to the tail rotor and possibly main rotor.

In manned helicopters, a pilot faced with such a failure can employ the autorotation maneuver to safely land the aircraft. Due to the aerodynamics of the main rotor, even when no power is supplied to it, it is possible to maintain a steady rate of descent. This is accomplished by using the air flowing through the rotor disk to rotate the main rotor—the reverse process from normal flight. In this case the main rotor acts as a parachute, breaking the helicopter. Just before touchdown the rotor rpm is exchanged for a reduction in the descent rate thus allowing the helicopter to land safely.

The proposed emergency controller is an independent system on-board unmanned helicopters that is capable of performing the autorotation maneuver autonomously. Its primary purpose is to minimize the probability of fatalities or injuries to people on the ground. Secondary goals are to minimize the risk of collision with other aircraft or stationary objects, as well as minimize the damage to the aircraft itself. As a result, vertical autorotation is preferred in this case, to minimize the airspace volume through which the helicopter will need to fly. Additionally the sink rate is reduced during the last 3 m of descent to minimize the effects of impact even if there are people in the area.

2 History and State of the Art of Autonomous Autorotation

In 1977, Johnson derived an autorotation model that includes vertical as well as longitudinal movement [12]. The optimal control was derived using a cost function that depended on horizontal and vertical speed at touchdown. The derivation of the control law was based on iterative numerical integration forwards and then backwards between the two boundary points and updating using the steepest descent method. The results were then compared with the performance of a modified Bell OH-58A carrying a High Energy Rotor System (HERS).

A few years later in Stanford, Allan Yeow-Nam Lee improved on Johnson's work by introducing state inequality constraints that were converted to equality using slack variables [13]. The controller was derived by numerical parameter optimization using the Sequential Gradient Restoration technique (an iterative method).

Although Johnson and Lee derived optimal autorotation trajectories, the issue of autonomous autorotation was not addressed until almost a decade later. In Japan, Hazawa et al. [11] derived two autorotation models (one linear and one non-linear) and used a PI controller to land a small unmanned helicopter.

In a continuation of the work of Johnson and Lee, Aponso et al. presented their own method to optimize the trajectory and control inputs for a full-size helicopter during an autorotation landing [3]. The goal of their work was to ensure the survival

of sensitive sensors and data stored on board the helicopter in the case of non-catastrophic failures. A significant drawback of their method is that it precalculates the control inputs and the trajectory before entering the autorotation maneuver and as a result is not robust with respect to modeling errors and outside interference. Because of this mismatch between model and simulation a flare law was necessary, that forces the flare to occur at 30 ft. Their work was evaluated against a high fidelity Bell 206 simulator.

During 2008, two groups presented results for autonomous autorotation using machine learning techniques. In the first approach [1], the controller was trained using pre-recorded pilot reference autorotations that provided a model of the aircraft and the "ideal" trajectory. The landing itself was achieved by forcing the helicopter to hover at 0.5 m. The performance of the controller was demonstrated using a small unmanned helicopter (XCell Tempest). The second approach was a straightforward application of reinforcement learning to train a controller using the Johnson model, cost function and experimental data.The final state-action space has 10 dimensions and was covered using RBFs, whose parameters were updated using backpropagation. After 9000 epochs the number of RBFs was about 19,000 and the success rate around 80%.

3 Proposed Approach

Current applications for unmanned helicopters typically require hovering at relatively low altitudes of a few hundred meters. Since the sink rate during autorotation can be significant, the whole maneuver may take only a few seconds to complete making manual intervention difficult if not impossible. Furthermore, on-board processing capacity of small unmanned helicopters is limited, which in turn puts bounds on the computational complexity of the controller if real-time operation is to be achieved. To meet these performance requirements the use of a nonlinear, model-predictive controller augmented by a recurrent neural network is proposed.

The idea behind model predictive control is to start with a fixed prediction horizon (N_s), using the current state of the plant as the initial state. An optimal control sequence of length N_c ($N_s \geq N_c$) is then obtained that minimizes an objective function while at the same time satisfying posed constraints. After applying the first element of that sequence as an input to the plant, the new state is observed and used as an initial state repeating the process.

Model predictive control is used extensively for the control of production processes and the properties of linear model predictive controllers are well understood. In the case of nonlinear problems, linearization techniques are usually preferred because NMPC suffers from the same issues as any nonlinear optimization approach; convergence, stability and computational complexity. Nevertheless many NMPC-based solutions to various nonlinear problems have been proposed in the literature, with encouraging results.

The NMPC problem can be expressed as an optimization problem, specifically:

$$\min L(\boldsymbol{u}, \boldsymbol{x}) \qquad \text{s.t.} \quad \mathcal{C}(\boldsymbol{u}, \boldsymbol{x}) \leq 0 \tag{1}$$

where a control sequence $\boldsymbol{u}$ is determined that minimizes the objective function L and satisfies the inequality constraints $\mathcal{C}$. This optimization problem needs to be solved each time a new state is observed. To solve such problems, Xia et al.

have proposed a series of recurrent neural networks [17–20]. The idea behind their approach is to build a neural network that models an ODE whose equilibrium point is the optimal solution to the problem (1). This approach has two major advantages; guarantied exponential convergence in the case of convex problems and fast execution speed when using hardware that can perform parallel computations.

The update rule of the recurrent neural network used in this paper was adapted from [20] and is given by:

$$d\begin{pmatrix}\boldsymbol{u}\\ \boldsymbol{\chi}\end{pmatrix} = \gamma\begin{pmatrix}-\boldsymbol{u}+\left(\boldsymbol{u}-\dfrac{\mathrm{d}L}{\mathrm{d}\boldsymbol{u}}-\dfrac{\mathrm{d}\mathcal{C}}{\mathrm{d}\boldsymbol{u}}\boldsymbol{\chi}\right)^{+}\\ -\boldsymbol{\chi}+(\boldsymbol{\chi}-\mathcal{C}(\boldsymbol{u}))^{+}\end{pmatrix} \tag{2}$$

where γ is a learning rate parameter, $(\cdot)^{+}$ is an activation function and $\boldsymbol{\chi}$ is an auxiliary vector with size equal to the number of constraints.

3.1 Vertical Autorotation Model

The vertical autorotation model used for the controller is given by:

$$\dot{v_H} = g - \frac{\rho_\alpha A R^2\Omega^2}{2M}\sigma C_{l,a}\left[\frac{\theta}{3}-\frac{v_i - v_H}{2\Omega R}\right] - \frac{\rho_\alpha f_e}{2M}{v_H}^2 \tag{3}$$

$$\dot{\mathsf{z}} = -v_H \tag{4}$$

$$\dot{\Omega} = -\frac{\rho_\alpha A R^3\Omega^2}{I_R}\frac{v_i - v_H}{R\Omega}\sigma C_{l,a}\left[\frac{\theta}{3}-\frac{v_i - v_H}{2\Omega R}\right] - \frac{\rho_\alpha A R^3\Omega^2}{8I_R}\sigma C_{d,0} \tag{5}$$

where Eq. 3 is derived from the balance of the three forces acting on the helicopter; the thrust provided by the main rotor, the aerodynamic drag from moving through the air and the weight of the aircraft. Similarly Eq. 5 is obtained from a torque balance between the torque added or subtracted by the inflow and torque subtracted due to blade drag. For the detailed derivation of these equations the reader is referred to [13].

Since the inflow dynamics are generally not measurable during flight, the model used internally by the controller will be simplified to include only the measurable states namely v_H, z and Ω. As a result the controller assumes a steady-state inflow velocity, that is:

$$v_i = v_{i,s} = v_{i,h} f_i \tag{6}$$

where $v_{i,h} = \sqrt{\dfrac{Mg}{2\rho_\alpha A}}$ and $f_i = \begin{cases}\kappa + 0.75\dfrac{v_H}{v_{i,h}} & , 0 \le \dfrac{v_H}{v_{i,h}} \le \dfrac{8\kappa}{4\kappa+1}\\ 7\kappa - 3\kappa\dfrac{v_H}{v_{i,h}} & , \dfrac{8\kappa}{4\kappa+1} \le \dfrac{v_H}{v_{i,h}} \le 2\\ \kappa\dfrac{v_H}{2v_{i,h}} - \kappa\sqrt{\left(\dfrac{v_H}{2v_{i,h}}\right)^2 - 1} & , 2 \le \dfrac{v_H}{v_{i,h}}\end{cases}$

The control input is scaled to the [0, 1] range, while the model equations are non-dimensionalized using the nominal rotor angular velocity Ω_0 and the rotor radius R:

$$\tau = \frac{\Omega_0 t}{100} \Rightarrow \qquad \frac{\mathrm{d}}{\mathrm{d}t} = \frac{\Omega_0}{100}\frac{\mathrm{d}}{\mathrm{d}\tau} \tag{7}$$

$$x_1 = \frac{100 v_H}{\Omega_0 R} \Rightarrow \qquad v_H = \frac{\Omega_0 R}{100} x_1 \tag{8}$$

$$x_2 = \frac{\mathsf{z}}{10R} \Rightarrow \qquad \mathsf{z} = 10 R x_2 \tag{9}$$

$$x_3 = \frac{\Omega}{\Omega_0} \Rightarrow \qquad \Omega = \Omega_0 x_3 \tag{10}$$

$$x_4 = \frac{100 v_{i,s}}{\Omega_0 R} \Rightarrow \qquad v_{i,s} = \frac{\Omega_0 R}{100} x_4 \tag{11}$$

$$u = \frac{\theta + \theta_m}{\theta_M - \theta_m} \Rightarrow \qquad \theta = u(\theta_M - \theta_m) + \theta_m \tag{12}$$

The final controller model equations are:

$$\dot{x}_1 = \frac{10^4}{{\Omega_0}^2 R} g - \frac{\rho_\alpha f_e R}{2M} x_1^2 + \frac{25 \rho_\alpha A R \sigma C_{l,a}}{M} x_3 (x_4 - x_1) - \frac{29.1 \rho_\alpha A R \sigma C_{l,a} \theta_m}{M} x_3^2 - \frac{29.1 \rho_\alpha A R \sigma C_{l,a} (\theta_M - \theta_m)}{M} x_3^2 u \tag{13}$$

$$\dot{x}_2 = -\frac{1}{10} x_1 \tag{14}$$

$$\dot{x}_3 = -\frac{100 \rho_\alpha A R^3 \sigma C_{d,0}}{8 I_R} x_3^2 + \frac{\rho_\alpha A R^3 \sigma C_{l,a}}{400 I_R} (x_4 - x_1)^2 - \frac{2.91 \rho_\alpha A R^3 \sigma C_{l,a} \theta_m}{10^3 I_R} (x_4 - x_1) x_3 - \frac{2.91 \rho_\alpha A R^3 \sigma C_{l,a} (\theta_M - \theta_m)}{10^3 I_R} (x_4 - x_1) x_3 u \tag{15}$$

There are three types of constraints imposed on the controller. The first concerns the physical limits of the actuator:

$$\theta_m \le \theta \le \theta_M \Rightarrow 0 \le \boldsymbol{u} \le 1 \tag{16}$$

This type of constraint is easily handled by appropriate design of the neural network activation function. In this case the activation function is a saturation function that limits the input to the range [0, 1], so that the control sequence is always within the actuator limits:

$$(\cdot)^+ = \min(1, \max(\cdot, 0)) \tag{17}$$

The second constraint is incorporated to avoid blade stall:

$$C_T \le \frac{\sigma}{8} \Rightarrow \frac{1}{2}\sigma C_l \left(\frac{1}{3} u(\theta_M - \theta_m) + \frac{1}{3}\theta_m - \frac{1}{200} \frac{x_4(i) - x_1(i)}{x_3(i)} \right) \le \frac{\sigma}{8} \tag{18}$$

The last constraint is designed to protect the main rotor from mechanical stress. This is because for very low blade pitch the rotor angular velocity may increase above nominal, possibly damaging the rotor assembly.

$$\Omega \leq f_r \Omega_0 \Rightarrow x_3 \leq f_r \tag{19}$$

where from the literature f_r typically takes values between 1.05 and 1.25 [3, 14].

3.2 Simulation Model

As mentioned earlier the inflow dynamics and the ground effect phenomenon are ignored by the controller. Nevertheless both are taken into account by the simulator developed to test the controller. The response of the induced velocity to thrust changes is not instantaneous but exhibits a dynamic behavior. This behavior can be modeled using an inertia model with an apparent mass equal to 63.7% that of a sphere of air with the same radius as that of the rotor ($m_a = 0.637\rho_\alpha \frac{4}{3}\pi R^3$) [15]. As a result the thrust will be given by:

$$\begin{aligned} T &= m_a \dot{v}_i + 2\rho_\alpha A v_i (v_i - v_H) \\ &= 0.849\rho_\alpha A R \dot{v}_i + 2\rho_\alpha A v_i (v_i - v_H) \end{aligned} \tag{20}$$

In steady-state conditions the thrust is given by

$$T = 2\rho_\alpha A (v_i - v_H) v_i \tag{21}$$

which when combined with Eq. 20 produces:

$$\begin{aligned} 2\rho_\alpha A (v_H + v_{i,s}) v_{i,s} &= 0.849\rho_\alpha A R \dot{v}_i + 2\rho_\alpha A v_i (v_i - v_H) \\ (v_H + v_{i,s}) v_{i,s} &= 0.4245 R \dot{v}_i + v_i (v_i - v_H) \\ \dot{v}_i &= -\frac{2.356}{R} \left[v_i (v_i - v_H) - v_{i,s} (v_{i,s} - v_H) \right] \\ \dot{v}_i &= -\frac{2.356}{R} \left(v_i - v_{i,s} \right) \left(v_i + v_{i,s} - v_H \right) \end{aligned} \tag{22}$$

where $v_{i,s}$ is the steady state induced velocity typically calculated from empirical models as a function of v_H. It is obvious that the derived equation has two possible steady state solutions; $v_i = v_{i,s}$ and $v_i = v_H - v_{i,s}$ although only the former is of interest. To overcome this problem, it is assumed that $v_i v_H \simeq v_{i,s} v_H$ and as a result the final induced velocity model is given by:

$$\dot{v}_i = -\frac{2.356}{R} \left({v_i}^2 - {v_{i,s}}^2 \right) \tag{23}$$

An additional correction factor is required for hovering near the ground, due to a phenomenon called ground effect. Because the rotor wake meets the ground the pressure below the rotor rises resulting in higher thrust generation for the same power. There are several empirical models of ground effects, the one used in this work is given by:

$$\left[\frac{T}{T_\infty} \right]_{P=\text{const}} = \frac{1}{1 - \frac{\sigma C_l \lambda_i}{4 C_T} \left(\frac{R}{4z} \right)^2} \tag{24}$$

4 Controller Derivation

The vertical autorotation model can be expressed as a general SIMO nonlinear affine in the control problem given by:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})u$$

$$\min_{u\in U} \int_0^{t_f} L(\boldsymbol{x}, u)$$

which in discrete time becomes:

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + t_s \boldsymbol{f}(\boldsymbol{x}(t)) + t_s \boldsymbol{g}(\boldsymbol{x}(t))u(t) \tag{25}$$

$$\min_{u(i)\in U} \sum_{i=0}^{t_f} L(\boldsymbol{x}(i), u(i)) \tag{26}$$

To find the optimal sequence using the recurrent neural network of Eq. 2, the $\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{u}}$, $\mathcal{C}(\boldsymbol{u})$ and $\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\boldsymbol{u}}$ quantities need first be calculated.

Taking into account that actions can only affect future states and don't depend on past or future actions:

$$\frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}u(i)} = 0, \forall x(k), u(i) : i \geq k \qquad \frac{\mathrm{d}u(k)}{\mathrm{d}u(i)} = \begin{cases} 1 & \text{if } i = k \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Differentiating Eq. 25 with respect to a control action $u(i)$:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{x}(k+1)}{\mathrm{d}u(i)} &= \frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}u(i)} + t_s J_f(\boldsymbol{x}(k)) \frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}u(i)} + t_s J_g(\boldsymbol{x}(k)) \frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}u(i)} u(k) + t_s g(\boldsymbol{x}(k)) \frac{\mathrm{d}u(k)}{\mathrm{d}u(i)} \\ &= \left[I + t_s J_f(\boldsymbol{x}(k)) + t_s J_g(\boldsymbol{x}(k))u(k) \right] \frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}u(i)} + t_s g(\boldsymbol{x}(k)) \frac{\mathrm{d}u(k)}{\mathrm{d}u(i)} \end{aligned} \tag{28}$$

Using Eqs. 28 and 27 the following update rule for calculating the $\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}u(i)}$ from the previous prediction step is obtained:

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}u(i)} = \begin{cases} 0 & \text{if } t < i \\ t_s g(\boldsymbol{x}(k)) & \text{if } t = i \\ \left[I + t_s J_f(\boldsymbol{x}(k)) + t_s J_g(\boldsymbol{x}(k))u(k) \right] \frac{\mathrm{d}\boldsymbol{x}(k)}{\mathrm{d}\boldsymbol{u}(i)} & \text{otherwise} \end{cases} \tag{29}$$

If an objective function of the following form is assumed:

$$L = \sum_{j=1}^{N_s} L^*(\boldsymbol{x}(j)) + w\boldsymbol{u}^T\boldsymbol{u} \tag{30}$$

where L^* is a function of the state and w is a positive weight factor, then:

$$\frac{\mathrm{d}L}{\mathrm{d}\boldsymbol{u}} = \sum_{i=1}^{N_s} \left(\frac{\partial L^*}{\partial \boldsymbol{x}(i)} \frac{\mathrm{d}\boldsymbol{x}(i)}{\mathrm{d}\boldsymbol{u}} \right) + 2w\boldsymbol{u} \tag{31}$$

The constraints in the vector form required by the recurrent neural network are given by:

$$\mathcal{C}(\boldsymbol{u}) = \begin{bmatrix} \left[C_T(i) - \frac{\sigma}{8}\right]_{1\le i\le N_c} \\ \\ \left[\Omega(j) - f_r\Omega_0\right]_{1\le j\le N_c} \end{bmatrix} \tag{32}$$

or

$$\mathcal{C}(\boldsymbol{u}) = \begin{bmatrix} \left[\frac{\sigma C_{l,a}(\theta_M-\theta_m)}{6}\left(\boldsymbol{u} + \frac{\theta_m}{\theta_M-\theta_m} - \frac{3}{4}\frac{1}{\theta_M-\theta_m}\frac{1}{C_{l,a}} - \frac{3}{400}\frac{1}{\theta_M-\theta_m}\frac{x_4(i)-x_1(i)}{x_3(i)}\right)\right]_{1\le i\le N_c} \\ \\ \left[x_3(j) - f_r\right]_{1\le j\le N_c} \end{bmatrix} \tag{33}$$

Finally differentiating Eq. 33 with respect to the control sequence gives $\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\boldsymbol{u}}$:

$$\frac{\mathrm{d}\mathcal{C}}{\mathrm{d}\boldsymbol{u}} = \begin{bmatrix} \frac{\sigma C_{l,a}(\theta_M-\theta_m)}{6}[I]_{N_c\times N_c} \\ -\frac{\sigma C_{l,a}}{800}\left[\frac{\frac{\mathrm{d}x_4(i)}{\mathrm{d}x_1(i)}-1}{x_3(i)}\frac{\mathrm{d}x_1(i)}{\mathrm{d}u(j)} - \frac{x_4(i)-x_1(i)}{x_3^2(i)}\frac{\mathrm{d}x_3(i)}{\mathrm{d}u(j)}\right]_{\substack{1\le i\le N_c \\ 1\le j\le N_c}} \\ \\ \left[\frac{\mathrm{d}x_3(i)}{\mathrm{d}u(j)}\right]_{\substack{1\le i\le N_c \\ 1\le j\le N_c}} \end{bmatrix} \tag{34}$$

Although Eq. 29 requires that each $\frac{\mathrm{d}x(i)}{\mathrm{d}\boldsymbol{u}}, i \in [1, 2, \ldots, Ns]$ is computed in sequence, Eqs. 31–34 are independent and can be calculated in parallel. Furthermore

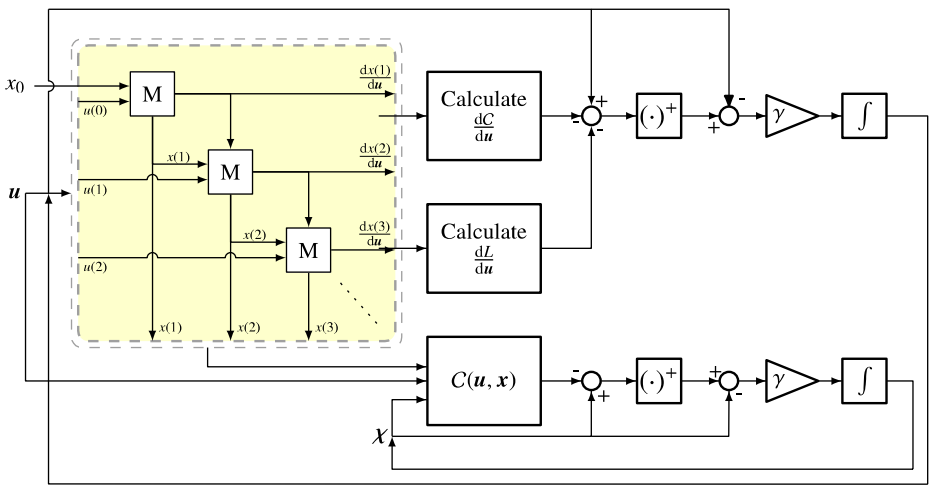

Fig. 1 Block diagram of the controller. With the exception of the shaded block, the other operations can be run in parallel. Inside the shaded block a cascaded connection is used to calculate $\frac{\mathrm{d}x(i)}{\mathrm{d}\boldsymbol{u}}$ from $\frac{\mathrm{d}x(i-1)}{\mathrm{d}\boldsymbol{u}}$

the rest of the neural network also allows parallel computations as can be seen in Fig. 1.

5 Results

The controller was tested using a model of the OH-58A helicopter with high energy rotor system (HERS). Although this helicopter does not represent a small rotorcraft, this model was chosen because it has been used extensively in the literature thus facilitating comparison with other developed controllers. The model parameters used are summarized in Table 1. For this paper a simple objective function was used:

$$L^* = 0.05\,(\max(v_H, 5.5) - 0.5)^2\, e^{1-1.25\min(z,1)} \tag{35}$$

This corresponds to an objective of lowering the sink rate to 0.5 m s^{-1} for the last 3 m of the descent.

5.1 Baseline Scenario

The baseline simulation was carried out modeling a descent from an initial altitude of 120 m using a prediction horizon of $N_s = 10$ and a control horizon of $N_c = 5$. The simulation is performed at an update rate of 1kHz while the controller is run at 10Hz. The neural network parameters are $E = 150$ and $\gamma = 0.08$.

The results, presented in Fig. 2, show that the helicopter accomplished the stated objective, without violating the thrust coefficient and rpm constraints. The whole maneuver lasts for about 14 s, although about half of it corresponds to the last 4 m of the descent. The sudden jump in blade pitch at an altitude of about 60 m is due to the big drop required in the sink rate so that the Ω-constraint is not violated. A smoother transition is also possible, but at the expense of either violating the constraint or requiring a longer prediction horizon.

It should also be noted that although a sink rate of 0.5 m s^{-1} is achieved, towards the end of the maneuver an increase in the sink rate is observed. This is due to the rapid loss of inertial energy in the rotor and the onset of stall in the blades that

Table 1 Vertical autorotation model parameters for a modified OH-58A helicopter with high energy rotor system

Helicopter mass (M)	1,360 kg
Rotor moment of inertia (I_R)	911 kg m^2
Solidity factor (σ)	0.048
Rotor disc radius (R)	5.37 m
Mean drag coefficient (C_d)	0.008,7
Lift coefficient (C_l)	5.73 rad^{-1}
Equivalent unit drag coefficient area (f_e)	2.32 m^2
Induced power correction factor (κ)	1.13
Nominal rotor speed (Ω_0)	354 rpm or 37 rad s^{-1}
Air density (ρ_α)	1.225 kg m^{-3}
Minimum main rotor pitch	−2°
Maximum main rotor pitch	16°
Maximum main rotor rpm	1.15Ω_0

Source: [8, 13]

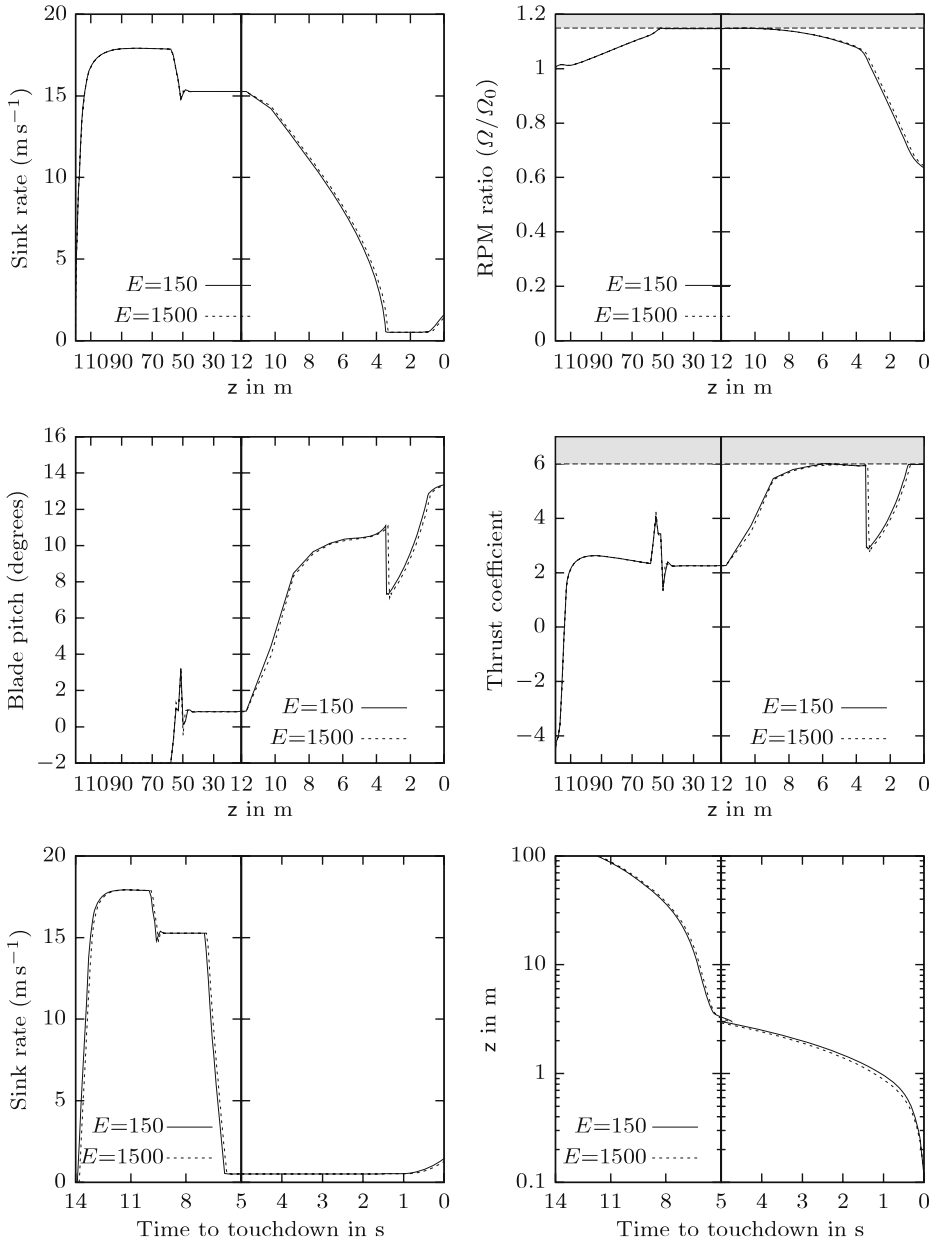

Fig. 2 The sink rate, rotor rpm and control input of the OH-58A for a descent from an initial altitude of 120 m ($N_s = 10$, $N_c = 5$). The *shaded regions* represent the posed constraints. The scale on the *right side of the graphs* has been altered to show the last stage of the descent in higher detail

requires checking the rate of increase of the blade pitch. Nevertheless, and despite the increase, the velocity at touchdown remains within mechanical tolerances of the landing gear, since the latter is typically designed for sink rates up to 3 m s^{-1}.

Figure 2 also presents the resulting trajectories for $E = 1500$. It is obvious that despite the tenfold increase in the time available to the neural network, the output is not significantly different and the results are comparable.

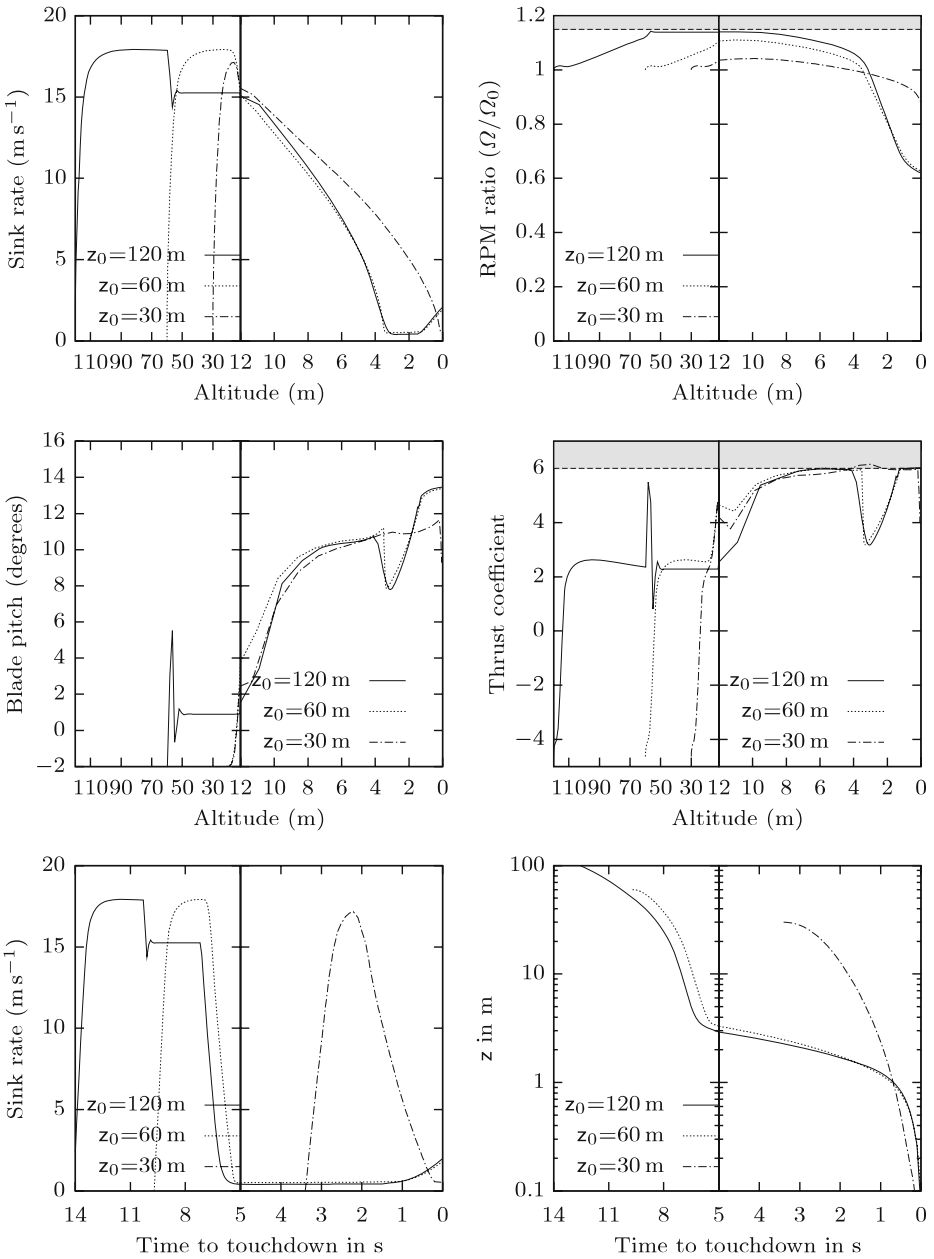

Fig. 3 The effect of different initial altitudes on the performance of the controller

5.2 Initial Altitude

The initial altitude can influence the state of the helicopter at the final stage of the descent and specifically the rotational energy available in the rotor to reduce the sink rate. For initial altitudes exceeding 90 m the helicopter will have time to reach the limit of the allowable rotor rpm. As a result in each case it will reach the flare altitude with $\Omega \simeq 1.15\Omega_0$ and $v_H \simeq 15$ m s^{-1}.

Conversely in the occasions were the initial altitude is lower than 90 m, the kinetic energy stored in the rotor will be lower. The sink rate can be lower than 15 m s^{-1} if the helicopter failed at a very low altitude (< 10 m) or higher for intermediate altitudes. Figure 3 presents the trajectories for three simulations that feature an initial altitude of 120 m, 60 m and 30 m. In the last instance and although the output of the controller does not deviate significantly from that of the other simulations, the target sink rate is achieved only centimeters above the ground. This is because of the thrust coefficient constraint and the lower energy stored in the rotor. To improve the performance in this case, either the prediction horizon would need to be increased or the thrust coefficient constraint be relaxed.

5.3 Noise

To investigate the impact of sensor noise on controller performance, two simulations were carried out using different noise levels. The noise is assumed to be zero-mean, gaussian with varying standard deviation. The noise levels are summarized in Table 2.

Figure 4 shows the simulation results for the first noise level scenario. The trajectory is not significantly changed and the only thing affected is the controller output during the Ω-controlled descent. The latter occurs because of the design of the controller that tries to maintain constant rpm and becomes more pronounced as the noise in the rpm measurements increases as shown in Fig. 5. As the noise level increases the v_H-controlled region starts to get affected as well. This is because the controller is required to maintain a constant velocity in the face of noisy measurements. It should be noted that in the second case the simulation is terminated early. This is because according to the information available to the controller, the helicopter has reached the ground and the simulation terminated.

5.4 Execution Speed

An additional parameter that is important in this problem is the execution speed of the controller. To determine this, the execution time of a single iteration was calculated for different prediction and control horizons. Since a single iteration is very fast, its execution time was estimated by measuring the total time required for 2,000 iterations. The tests were carried out using a single-core Athlon XP 3200+

Table 2 Sensor noise levels

	Standard deviation	
	Level 1	Level 2
Sink rate (m s^{-1})	0.2	0.5
Altitude (m)	0.25	0.75
Rotor speed (RPM)	7	10.5

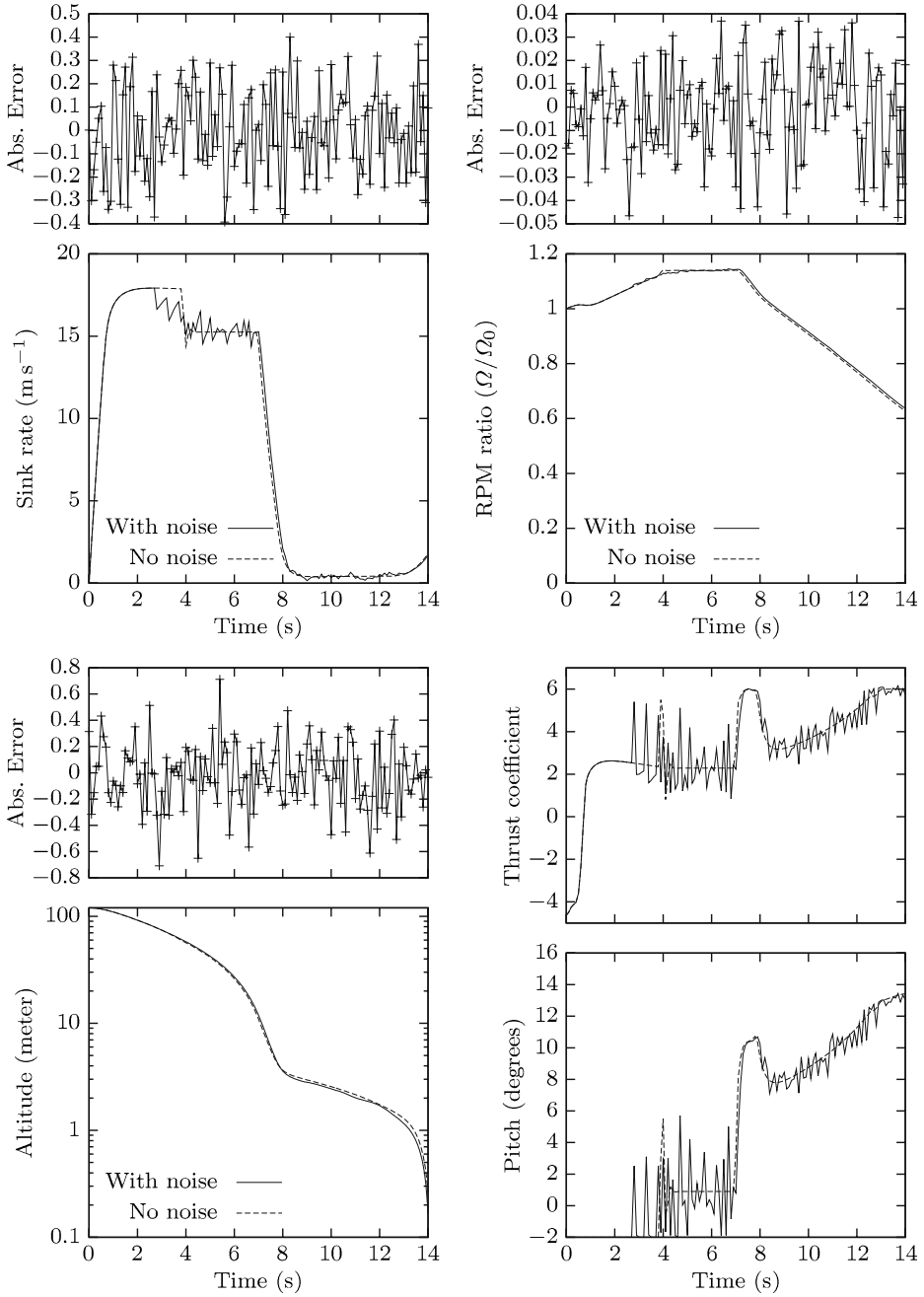

Fig. 4 The trajectory of the helicopter with and without noise under the second noise level. The top graphs present the absolute error of the sensor measurement as a function of time

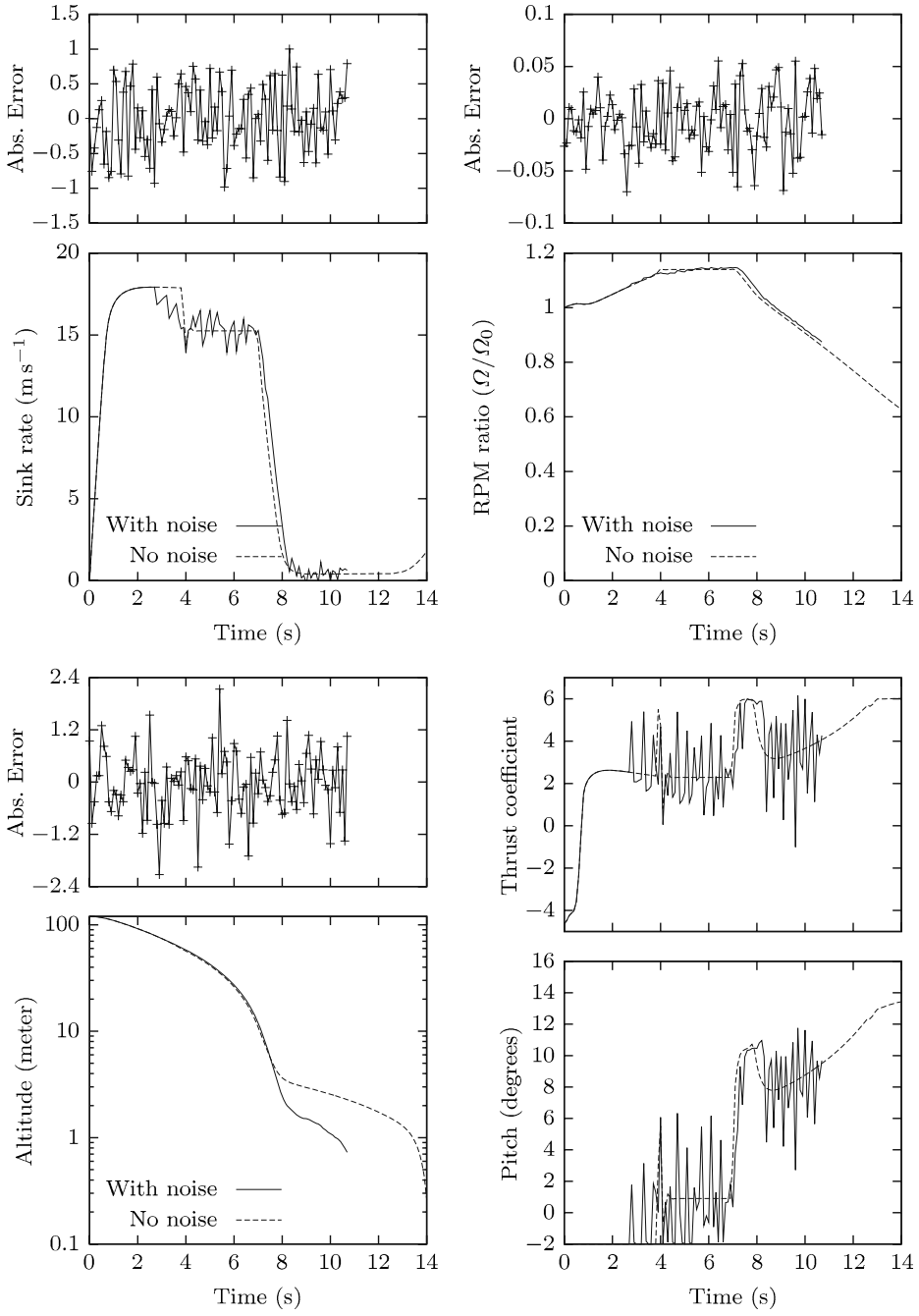

Fig. 5 The trajectory of the helicopter with and without noise under the third noise level. The top graphs present the absolute error of the sensor measurement as a function of time

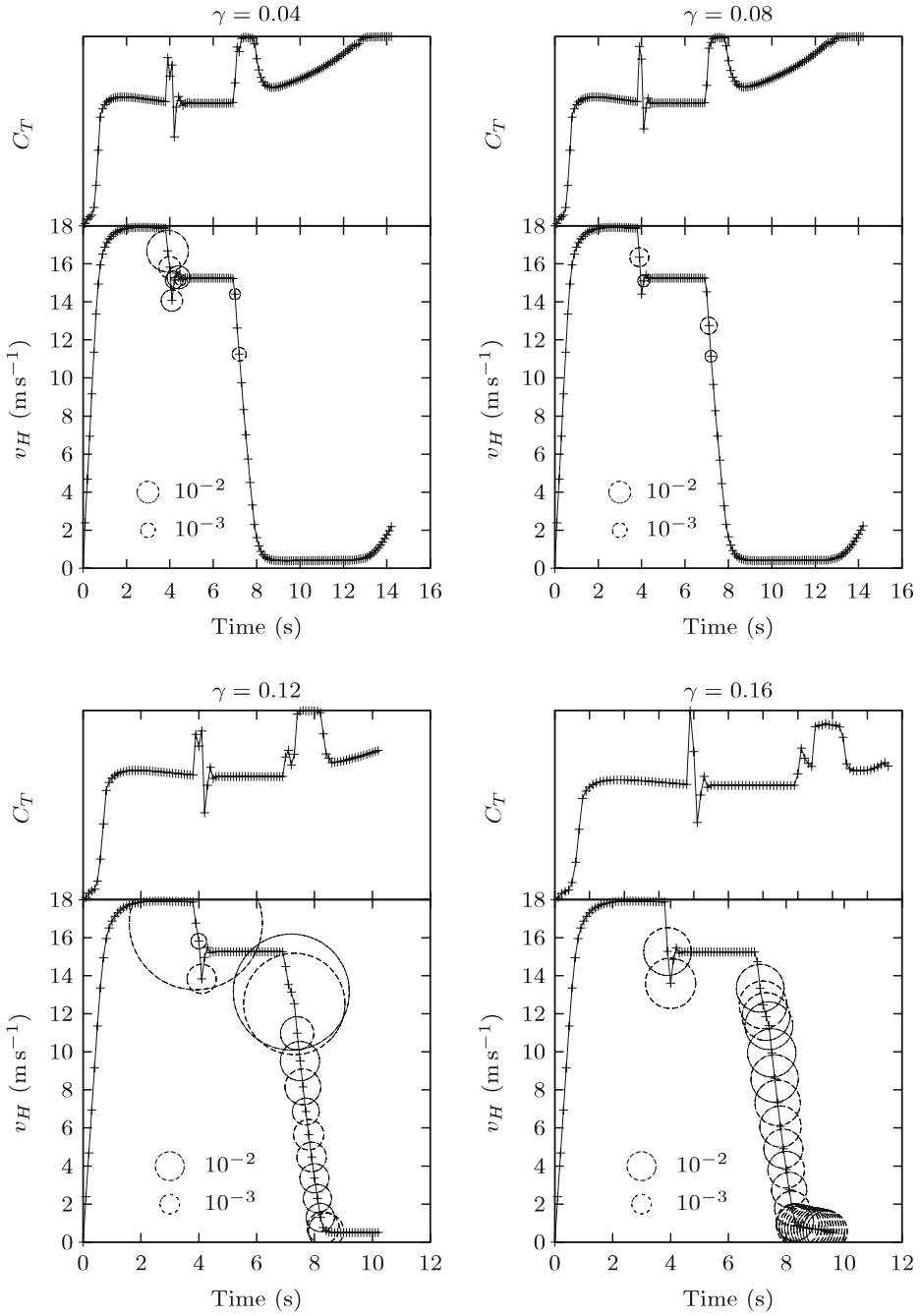

Fig. 6 The variance of the neural network output with respect to time and helicopter sink rate for four values of the learning rate parameter. *Larger circles* mean larger variance while variances smaller than 10^{-4} are not shown. For comparison purposes the thrust coefficient at the corresponding time is also provided

(model of 2005) CPU, throttled to a frequency of 1 GHz running a 32-bit version of Debian Linux. No parallelization or off-line optimization was used to improve the execution time. The cost of the EKF was also calculated under the same conditions and was found to be approximately 128 μs.

For $N_s = 10$ and $N_c = 5$ a controller iteration required 0.88 ms whereas for $N_s = 12$ and $N_c = 6$ this time increased to 1.23 ms. If an update rate of 10 Hz is chosen, under the aforementioned conditions the maximum epoch length is 113 and 81 respectively. For higher prediction and control horizon the number of possible iterations drops considerably, down to 30 for $N_s = 20$ and $N_c = 10$.

5.5 Learning Rate and Convergence

To investigate the effect of the learning rate parameter as well as the convergence characteristics of the neural network, four simulations were carried out for different values of γ. Higher values of the latter parameter are typically used to improve the convergence speed. On the other hand, such values can lead to over-corrections and oscillations. This is exacerbated by constraint enforcement and results in producing the opposite of the desired effect. The influence of the effect of the controller design parameter γ was evaluated using the variance in the neural network output during the last 20% of the repetitions of each cycle. If the neural network has converged, low output variance is expected. The results summarized in Fig. 6 show that high variance is exhibited mainly in regions where the helicopter transitions from one mode to another. This is expected because the cost and constraint derivatives take their higher values there. Additionally for learning rates 0.12 and above, variance is exhibited first in the region of deceleration and then in the entire velocity-controlled region.

6 Conclusions and Future Work

Current proposed automated autorotation methods have one or more significant drawbacks that don't allow them to be incorporated as they are in current and future aircraft:

- The trajectory is not calculated on-line because the calculations cannot be carried out in real-time. This is significant because any discrepancies between the model and the actual aircraft as well as any external disturbances can lead to accumulating error. This error can be significant by the time the aircraft approaches touchdown and may lead to a catastrophic accident.
- The objective of the autorotation maneuver is typically chosen to be zero vertical and horizontal velocity at zero height. Nevertheless, especially in the case of unmanned helicopters the foremost objective should be to minimize human injuries and fatalities.
- Autonomous autorotation is based on training using pre-recorded attempts by an expert. This can be problematic since the limitations of a human pilot are incorporated into the design and the performance will be as good as a human pilot. Furthermore it does not allow for different objectives and large deviations from the conditions under which the experiments were recorded.

– The controller is designed as a black box, trained through repeated, simulated trial and error. In this approach the accuracy of the simulation model used is very important. Furthermore the controller needs to be repeatedly trained under all possible conditions.

On the other hand the controller presented in this paper is capable of real-time, on-line trajectory optimization using different objective functions without a requirement for training beforehand. Due to the characteristics of the NMPC approach, it is robust to sensor errors and the introduction of a recurrent neural network simplifies the non-linear optimization and significantly improves the speed with which the optimal control sequence is obtained. The convergence speed may be further improved utilizing specialized hardware, that allows parallel computations.

Future work in this area will entail further testing of the controller with different helicopter models, different scenarios (e.g. wind) as well as testing against commercial flight simulators. Gain scheduling or alternate objective functions need to be investigated to determine if it possible to overcome the convergence problems encountered in the regions where the helicopter rapidly changes state. Furthermore better tuning of the objective function and NMPC and neural network parameters is planned, along with an investigation of their impact on the controller performance in terms of both accuracy and speed. Long-term goals include an investigation of the convergence characteristics of the neural network; whether convergence can be guarantied and under what conditions.

In the future, it is envisioned that this controller will be a part of larger emergency system that will be capable of handling a multitude of failures from detection to resolution. Such a system would undoubtedly improve the safety performance of unmanned helicopters in general and pave the way for further integration of such systems into the national airspace system.

Acknowledgement This research has been partially supported by NSF Grant, IIP-0856311 (DU Grant number 36563).

References

1. Abbeel, P., Coates, A., Hunter, T., Ng, A.Y.: Autonomous autorotation of an RC helicopter. In: Proc. 11th International Symposium on Experimental Robotics (2008)
2. Anand, S.: Domestic use of unmanned aircraft systems: evaluation of policy constraints and the role of industry consensus standards. Journal of Engineering and Public Policy **11** (2007)
3. Aponso, B., Bachelder, E., Lee, D.: Automated autorotation for unmanned rotorcraft recovery. Presented at AHS international specialists' meeting on unmanned rotorcraft (2005)
4. Carlson, R.: Helicopter performance—transportation's latest chromosome: the 31st annual Alexander A. Nikolsky lecture. J. Am. Helicopter Soc. **47**(1) (2002)
5. Clothier, R., Walker, R.: Determination and evaluation of UAV safety objectives. In: Proc. 21st International Unmanned Air Vehicle Systems Conference, pp. 18.1–18.16 (2006)
6. Clothier, R., Walker, R., Fulton, N., Campbell, D.: A casualty risk analysis for unmanned aerial system (UAS) operations over inhabited areas. In: Proc. 12th Australian International Aerospace Congress and 2nd Australasian Unmanned Air Vehicles Conference (2007)
7. Dalamagkidis, K., Valavanis, K., Piegl, L.: On Integrating Unmanned Aircraft Systems into the National Airspace System: Issues, Challenges, Operational Restrictions, Certification, and Recommendations, Intelligent Systems, Control and Automation: Science and Engineering, vol. 36. Springer, New York (2009)
8. Dooley, L.W., Yeary, R.D.: Flight test evaluation of the high inertia rotor system. Final report for period 21 September 1976–February 1979 USARTL-TR-79-9, Bell Helicopter Textron (1979)

9. Haddon, D.R., Whittaker, C.J.: UK-CAA policy for light UAV systems. UK Civil Aviation Authority (2004)
10. Hayhurst, K.J., Maddalon, J.M., Miner, P.S., Dewalt, M.P., Mccormick, G.F.: Unmanned aircraft hazards and their implications for regulation. In: Proc. 25th IEEE/AIAA Digital Avionics Systems Conference, pp. 1–12 (2006)
11. Hazawa, K., Shin, J., Fujiwara, D., Igarashi, K., Fernando, D., Nonami, K.: Autonomous autorotation landing of small unmanned helicopter. Transactions of the Japan Society of Mechanical Engineers C **70**(698), 2862–2869 (2004)
12. Johnson, W.: Helicopter optimal descent and landing after power loss. NASA TM 73244, Ames Research Center, National Aeronautics and Space Administration (1977)
13. Lee, A.Y.N.: Optimal landing of a helicopter in autorotation. Ph.D. thesis, Department of Aeronautics and Astronautics, Stanford University (1985)
14. Lee, A.Y.N.: Optimal autorotational descent of a helicopter with control and state inequality constraints. J. Guid. Control Dyn. **13**(5), 922–924 (1990)
15. Leishman, J.G.: Principles of Helicopter Aerodynamics, 2nd edn. Cambridge Aerospace Series. Cambridge University Press, Cambridge (2006)
16. Weibel, R.E.: Safety considerations for operation of different classes of unmanned aerial vehicles in the national airspace system. Master's thesis, Massachusetts Institute of Technology (2005)
17. Xia, Y., Wang, J.: A recurrent neural network for nonlinear convex optimization subject to nonlinear inequality constraints. IEEE Trans. Circuits Syst. I **51**(7), 1385–1394 (2004). doi:10.1109/TCSI.2004.830694
18. Xia, Y., Wang, J.: A recurrent neural network for solving nonlinear convex programs subject to linear constraints. IEEE Trans. Neural Netw. **16**(2), 379–386 (2005). doi:10.1109/TNN.2004.841779
19. Xia, Y., Leung, H., Wang, J.: A projection neural network and its application to constrained optimization problems. IEEE Trans. Circuits Syst. I **49**(4), 447–458 (2002). doi:10.1109/81.995659
20. Xia, Y., Feng, G., Kamel, M.: Development and analysis of a neural dynamical approach to nonlinear programming problems. IEEE Trans. Automat. Contr. **52**(11), 2154–2159 (2007). doi:10.1109/TAC.2007.908342