JAST, Vol. 12, No. 1, pp. 37-51

JAST
Journal of Aerospace Science and Technology

Examination of Quadrotor Inverse Simulation Problem Using Trust-Region Dogleg Solution Method

H. Jafari[1] and F. Shahmiri[2*]

1, 2. Department of Aerospace Engineering, Maleke Ashtar University

*Postal Code: Tehran, IRAN

prof.shahmiri@gmail.com

In this paper, the particular solution technique for inverse simulation applied to the quadrotor maneuvering flight is investigated. The trust-region dogleg (DL) technique which is proposed alleviates the weakness of Newton's method used for numerical differentiation of system states in the solution process. The proposed technique emphasizes global convergence solution to the inverse simulation problem. This algorithm is evaluated by calculating the control inputs necessary to enable the quadrotor to follow a specified trajectory including climb-hover and cruise-hover maneuvers. The trajectory is generated by the direct simulation using a linear optimal control developed for the quadrotor. The rotors for the quadrotor are of a nonlinear model developed based on blade element theory (BET), linear aerodynamics, and non uniform inflow over the rotor disc. The results show that the control inputs obtained from the inverse simulation are in a good agreement with control inputs estimated by direct simulation. The results also confirm that the maximum difference between the prescribed trajectory and the trajectory generated by the direct simulation is less than 0.02%, and thus, the potential application of the inverse simulation with the trust-region dogleg optimization is evident.

Keywords: Quadrotor, Flight dynamics, Inverse simulation, Trust-region Dogleg method

Introduction

The inverse technique for dynamic model simulation, the one in which control inputs are determined based on the prescribed maneuver, has been receiving much attention in aerospace engineering and in other application areas, such as robotics, biomechanics, computer animations, and aerospace [1-11]. The potential of the inverse simulation approach for external validation of nonlinear simulation models is also given particular considerations. The external validation of nonlinear models is conducted using time history data collected from the experiments on the corresponding real system. Comparison of the measurement input variables with equivalent variables from simulation model in inverse mode by the measured output response data from the real system can provide insight about model deficiencies which may not be so obvious from conventional output response comparisons. A review of some available methods and algorithms in the field of aircraft flight mechanics and surveys on a number of typical applications have been reported earlier in the literature [12].

The available solution methods of inverse simulation are divided into techniques involving numerical differentiation and iterative techniques based on numerical integration processes. The differentiation method tends to be at least an order of magnitude faster than integration and the two approaches tend, therefore, to have different areas of application.

Kato and Suguira [13], Kato [14], and Thomson [15] have developed inverse simulation algorithms for particular air vehicles. The

1. Assistant Professor (Corresponding Author)

2. PhD Student

algorithms in [13-15] share a common feature in which they call numerical time differentiation of some vehicle state variables in the solution process. The numerical time differentiation approach is that of finding an appropriate approximation for the function to be differentiated and also differentiating the approximated function. However, the problem with time differentiation approach is that even the small errors in the approximated function are increasingly intensified, meaning that the unreliability and probability occurrence of solution instability is increased with the errors in the solution process [16].

Even though much attention has been focused on inverse simulation issues [17-22], Thomson has directly emphasized the deficiency in the solution process occurred by differentiation approach [15] and then, asserted that the numerical integration methods are more reliable and stable for inverse simulations.

Hess [23] has followed and restated the inverse simulation in a valuable procedure which eliminates time differentiation in the solution process. Although the resulting formulations have advantages over the algorithms requiring time differentiation, the direct simulation algorithm with control inputs is also called to obtain output vectors in a particular interval. Consequently, time integration method is essentially much slower than the time differentiation and is computationally time consuming. Some examples of the integration method have been examined in helicopter maneuvering flight, inspecting the flight and handling qualities of helicopters, as well as examining the aircraft performance [24-27]. Overall, solutions to inverse problems are generally impractical for all but restricted to the simplest problems with small computational domains.

Solutions to inverse simulation problems, particularly the non-linear ones, also face deficiencies of Newton-Raphson method (NR) in both time differentiation and integration process. The problem with this approach is that the NR method is rarely converged into non-convex and ill-conditioned problems [28, 29]. Because the small changes in input variables lead to large variations of output variables in complex problems, the global convergence cannot be achieved in the solution process. To alleviate the problem in time differentiation technique, Powell's dogleg (DL) algorithm is proposed and applied to the nonlinear inverse simulation problem in this paper. The DL algorithm is formulated based on trust-region methods implemented for unconstrained minimization through the combinations of the Gauss-Newton and steepest descent directions [30]. This algorithm is explicitly controlled through the use of the trust-region method. Accordingly, in this work, the potential of DL algorithm as compared with NR method in global convergence issue is exercised by estimating the control inputs necessary for the developed quadrotor model to follow a prescribed trajectory defined as climb-hover and cruise-hover maneuvers. The process of inverse simulation is presented in Fig. 1.

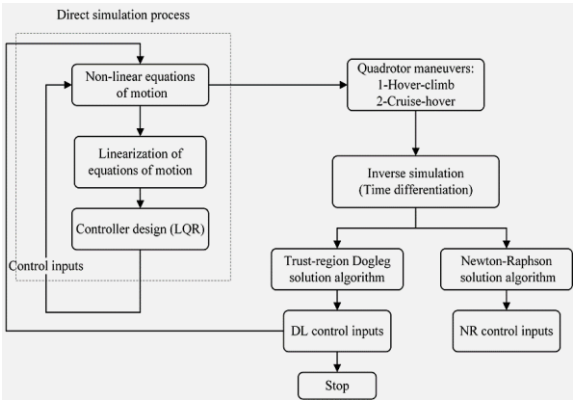


Fig. 1. Direct and inverse simulation process

Inverse simulation algorithm

The inverse problem is presented in [14] and will only be summarized here. In nonlinear equation form, the motion of a quadrotor is obtained as:

$$\dot{\boldsymbol{x}}(kT) = \boldsymbol{f}\left(\boldsymbol{x}(kT), \boldsymbol{u}(kT)\right) \quad (1)$$

where T is discretization interval, $\boldsymbol{u}(kT)$ is the input vector with four control inputs, and $\boldsymbol{x}(kT)$ is the state vector. Using Newton method, the following error vector for Eq. (1) is defined as:

$$\boldsymbol{F}_E\left(\boldsymbol{x}(kT), \boldsymbol{u}(kT)\right) = \dot{\boldsymbol{x}}(kT) - \boldsymbol{f}\left(\boldsymbol{x}(kT), \boldsymbol{u}(kT)\right) = 0 \quad (2)$$

or

$$\boldsymbol{F}_E\left(\boldsymbol{Y}(kT)\right) = \dot{\boldsymbol{x}}(kT) - \boldsymbol{f}\left(\boldsymbol{Y}(kT)\right) = 0 \quad (3)$$

where $\boldsymbol{Y} = [y_1 \quad \cdots \quad y_n]^{\mathrm{T}}$ is the unknown vector including control inputs and certain state variables. In an iterative procedure, the unknown vector $\boldsymbol{Y}$ is obtained as:

$$\boldsymbol{Y}_{m+1}(kT) = \boldsymbol{Y}_m(kT) + \boldsymbol{d}_{GN} \tag{4}$$

where m is an iteration number. Eq. (4) is the linear approximation of $\boldsymbol{Y}$, where $(\boldsymbol{d}_{GN})$ is Gauss-Newton step and error vector is updated where the error estimator is $|\boldsymbol{Y}_{m+1}(kT) - \boldsymbol{Y}_m(kT)|$. Normally, the convergence occurs if $|\boldsymbol{Y}_{m+1}(kT) - \boldsymbol{Y}_m(kT)| \leq 10^{-6}$. In this paper,$\boldsymbol{d}_{GN}$ is defined as:

$$\boldsymbol{d}_{GN} = -\left[\boldsymbol{J}\left[\boldsymbol{F}_E[\boldsymbol{Y}_{\boldsymbol{m}}(kT)]\right]\right]^{-1} \boldsymbol{F}_E[\boldsymbol{Y}_{\boldsymbol{m}}(kT)] \tag{5}$$

where

$$\boldsymbol{J}_{ij} = \frac{\partial \boldsymbol{F}_{E_i}[\boldsymbol{Y}(kT)]}{\partial y_j(kT)} i, j = 1, n \tag{6}$$

Here, $\boldsymbol{J}[\,]$is the Jacobian matrix which shows the derivative in the single variable Newton method. As the partial derivatives in Eq. (6) cannot be determined analytically, these derivatives are therefore approximated as:

$$\frac{\partial \boldsymbol{F}_E[\boldsymbol{Y}(kT)]}{\partial y_j(kT)} = \frac{\boldsymbol{F}_{E_i}[y_j + \Delta y_i/2] - \boldsymbol{F}_{E_i}[y_j - \Delta y_i/2]}{\Delta y_i} i, j = 1, \ldots, n \tag{7}$$

where Δy_i is perturbation in y_i, and can be generated as fixed percentage of y, at kT. It should be noted when the Jacobian inverse is singular, the Moore-Penrose pseudoinverse of the Jacobian $(\boldsymbol{J}^+)$ is used instead of $\boldsymbol{J}^{-1}$. More details on pseudoinverse are given in [31, 32].

$$(\boldsymbol{J}\{\boldsymbol{F}_E[\boldsymbol{Y}_m(kT)]\})^{-1} = (\boldsymbol{J}^+\{\boldsymbol{F}_E[\boldsymbol{Y}_m(kT)]\}) \tag{8}$$

The time differentiation algorithm is outlined in Fig. 2. This process is started with a given trajectory over a specified time interval, and then the time differentiation process is performed as the next step of the solution. Following this process, the iterative NR method is used to calculate unknowns of nonlinear equations of motion. The control inputs and other unknowns are changed and improved, in step fashion, every kT seconds on the basis of a solution of Eq. (4). When convergence condition is fulfilled, the control inputs can be computed and the solution continues at the next step time. The values of the control inputs are stored to enable a good first estimate to be made at the interval of next time point.

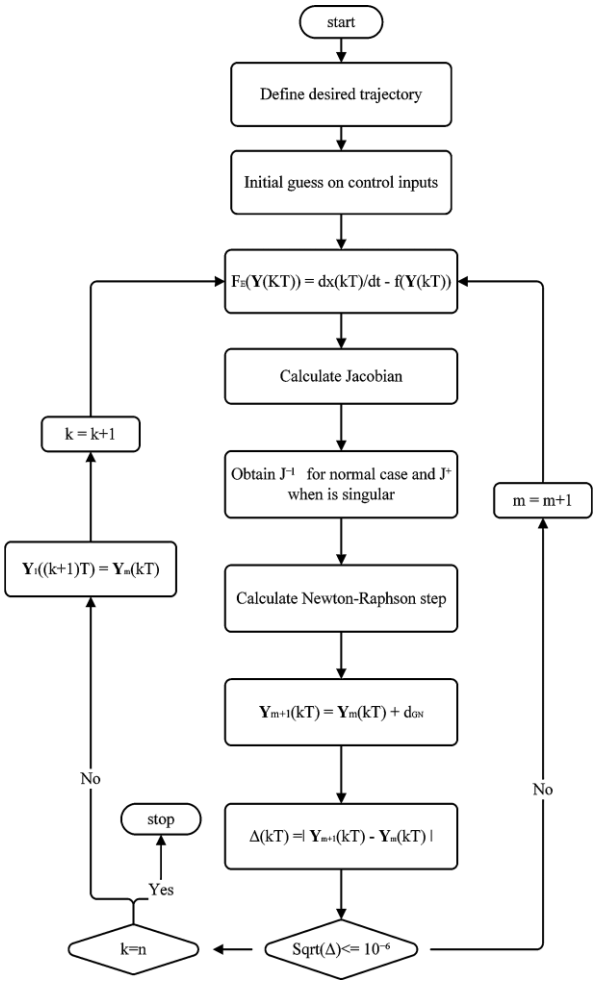


Fig. 2. Diagram for inverse simulation using Newton-Raphson method

Trust-region dogleg algorithm

Among different numerical solution methods, trust-region dogleg algorithm [30 & 34] is used to achieve the solutions of inverse problem in this paper. This method eliminates the singularity within the Jacobian inverse mostly occurring in NR approach. Furthermore, NR method may not converge when the starting point is far from the solution and a good estimation of the solution is not available. To avoid this, dogleg algorithm based on trust-region method is proposed here. Thus, the global convergence of the solution for any initial guess is obtained. Dogleg algorithm combines the properties of the steepest descent

method with the benefits of Gauss-Newton method [33].To use a trust-region strategy, a merit function is needed to decide if $\boldsymbol{Y}_{m+1}$ is better or worse than$\boldsymbol{Y}_{m+1}$. The merit function is defined as:

$$\begin{aligned}&\boldsymbol{m}_k(\boldsymbol{d})\\&=\frac{1}{2}\boldsymbol{F}_E^T[\boldsymbol{Y}(kT)]\boldsymbol{F}_E[\,\boldsymbol{Y}(kT)]\\&+\boldsymbol{d}^T\boldsymbol{J}^T[\boldsymbol{F}_E[\boldsymbol{Y}(kT)]]\boldsymbol{F}_E\,[\boldsymbol{Y}(kT)]\\&+\frac{1}{2}\boldsymbol{d}^T\boldsymbol{J}^T[\boldsymbol{F}_E[\boldsymbol{Y}(kT)]]\,\boldsymbol{J}\,[\boldsymbol{F}_E[\boldsymbol{Y}(kT)]]\boldsymbol{d}\end{aligned} \tag{9}$$

where the sub problem is corresponded to:

$$\text{Min.}[\boldsymbol{m}_k(\boldsymbol{d})]\quad \text{subject to } \|\boldsymbol{d}\| \le \Delta_k \tag{10}$$

where Δ_k is the radius of the trust-region,$\boldsymbol{d}$ is a diagonal scaling matrix (dog-leg step) and $\|\dots\|$ is the norm. This sub problem can be efficiently solved using a dogleg technique using Cauchy step ($\boldsymbol{d}_C$) and unconstrained minimizer or Gauss-Newton step ($\boldsymbol{d}_{GN}$). Thus, the exact solution of Eq. (10) is obtained as:

$$\boldsymbol{d} = \boldsymbol{d}_C + \lambda\,(\boldsymbol{d}_{GN} - \boldsymbol{d}_C) \tag{11}$$

where λ is the largest value between [0,1] when $\|\boldsymbol{d}\| \le \Delta_k$. The Cauchy step is calculated as:

$$\boldsymbol{d}_C = -\tau_k\Delta_k\frac{\boldsymbol{J}^T\{\boldsymbol{F_E}[\boldsymbol{Y}(kT)]\}.\boldsymbol{F_E}[\boldsymbol{Y}(kT)]}{\|\boldsymbol{J}^T\{\boldsymbol{F_E}[\boldsymbol{Y}(kT)]\}.\boldsymbol{F_E}[\boldsymbol{Y}(kT)\|} \tag{12}$$

where τ_k is step-size and is calculated as follow:

$$\tau_k = \begin{cases}1 & (F_E^T\boldsymbol{J})(\boldsymbol{J}^T\boldsymbol{J})(\boldsymbol{J}^T\boldsymbol{F}_E) \le 0\\ \min\left\{1,\frac{\|\boldsymbol{J}^T.\boldsymbol{F}_E\|^3}{\Delta_k(F_E^T.\boldsymbol{J})(\boldsymbol{J}^T.\boldsymbol{J})(\boldsymbol{J}^T.\boldsymbol{F}_E)}\right\} & (F_E^T\boldsymbol{J})(\boldsymbol{J}^T\boldsymbol{J})(\boldsymbol{J}^T\boldsymbol{F}_E) > 0\end{cases} \tag{13}$$

$\boldsymbol{J}\,[\boldsymbol{F}_E[\boldsymbol{Y}(kT)]]$ and $\boldsymbol{F}_E[\boldsymbol{Y}(kT)]$ are respectively abbreviated as $\boldsymbol{J}$ and $\boldsymbol{F}_E$ in Eq. (13). Gauss-Newton step is calculated as:

$$\boldsymbol{d}_{GN} = -\boldsymbol{J}^{-1}\{\boldsymbol{F}_E[\boldsymbol{Y}(kT)]\}\boldsymbol{F}_E[\boldsymbol{Y}(kT)] \tag{14}$$

The dogleg method allows successively better estimates of $\boldsymbol{Y}_{m+1}(kT)$ to be made at each value of k. Accordingly, the solution of Eq. (3) can be represented as:

$$\boldsymbol{Y}_{m+1}(kT) = \boldsymbol{Y}_m(kT) + \boldsymbol{d} \tag{15}$$

Eq. (15) is the linear approximation of Y, where (d) is dog-leg step and is updated where the error estimator is corresponded to $|\boldsymbol{Y}_{m+1}(kT) - \boldsymbol{Y}_m(kT)|$. The convergence is obtained if $|\boldsymbol{Y}_{m+1}(kT) - \boldsymbol{Y}_m(kT)| \le 10^{-6}$. In the case where $\boldsymbol{J}^{-1}$ is singular, $\boldsymbol{d}$ is just Cauchy direction, thus, Eq. (11) can also be written as:

$$\boldsymbol{d} = \boldsymbol{d}_C(1-\lambda) \tag{16}$$

What is to be said is that the inverse simulation algorithm based on DL approach is approximately similar to the NR algorithm presented in Fig. 2. However, in DL approach, there is no need to calculate the pseudoinverse of Jacobian and also the value of DL step is replaced by Newton step value. Furthermore, after the $m-th$ iteration where the convergence criterion in the internal loop is satisfied, the control inputs and other unknowns are determined, in step fashion, every kT seconds.

Quadrotor model

The equations of motion of the fuselage are those of a rigid body, formulated in the body-fixed coordinate system (see Fig. 3). The terms on the left hand side represent the externally applied loads $(\underline{F}^B, \underline{M}^B)$ which include contributions from the rotors and aerodynamic loads directly applied to the fuselage.

$$\underline{F}^B = m(\dot{\underline{V}}^B + \underline{\omega}_{IB}^B \times \underline{V}^B) \tag{17}$$

$$\underline{M}^B = I^B\dot{\underline{\omega}}_{IB}^B + \underline{\omega}_{IB}^B \times I^B\underline{\omega}_{IB}^B \tag{18}$$

where$m$ is the mass of quadrotor, $\underline{V}^B = [\dot{x}\quad \dot{y}\quad \dot{z}]^T$, $\underline{\omega}_{IB}^B = [\omega_X\quad \omega_Y\quad \omega_Z]$, and $I^B = I_{3\times3}$ is linear velocity vector, angular velocity vector, and the mass moment inertia matrix represented in body-fixed coordinate system, respectively. The superscript B refers to body-fixed coordinate system. The forces and moments on the left hand side of Eqs. (17) and (18) are then given by:

$$\begin{aligned}\underline{F}^B &= [F_{X_r}+F_{X_b}\quad F_{Y_r}+F_{Y_b}\quad F_{Z_r}+F_{Z_b}]^T\\&= [-(H+D_X)D_Y(D_Z\\&\quad -T)]^T\end{aligned} \tag{19}$$

$$\begin{aligned}&\underline{M}^B\\&= [M_{X_r}+M_{X_b}\quad M_{Y_r}+M_{Y_b}\quad M_{Z_r}+M_{Z_b}]^T\\&\cong [M_{X_b}\quad M_{Y_b}\quad Q]^T = [\tau_X\quad \tau_Y\quad \tau_Z]^T\end{aligned}$$

$$\begin{aligned}\underline{F}^B &= [F_{X_r}+F_{X_b}\quad F_{Y_r}+F_{Y_b}\quad F_{Z_r}+F_{Z_b}]^T\\&= [-(H+D_X)D_Y(D_Z\\&\quad -T)]^T\end{aligned} \tag{20}$$

$$\begin{aligned}&\underline{M}^B\\&= [M_{X_r}+M_{X_b}\quad M_{Y_r}+M_{Y_b}\quad M_{Z_r}+M_{Z_b}]^T\\&\cong [M_{X_b}\quad M_{Y_b}\quad Q]^T = [\tau_X\quad \tau_Y\quad \tau_Z]^T\end{aligned}$$

where the subscript r refers to the rotor and b to the helicopter body. The rotor blade is of a rectangular plan form with NACA 0012 cross sections and has no taper. All blades were designed

and built with no flapping, lead-lag, and feathering degrees of freedom.

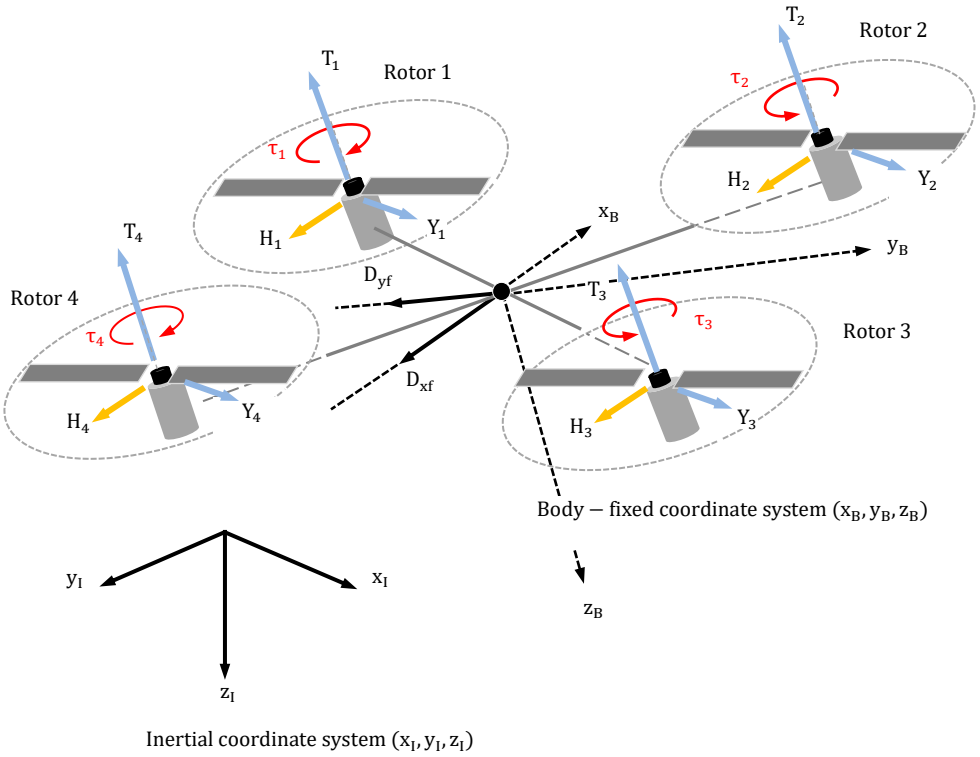


Fig. 3. Presentation of coordinate systems for a typical quadrotor.

The representation of Eq. (17) in inertial coordinate system is:

$$C_B^I \underline{F}^B = m\dot{\underline{V}}^I = mC_B^I(\dot{\underline{V}}^B + \underline{\omega}_{IB}^B \times \underline{V}^B) \qquad (21)$$

where C_B^I is the transformation matrix from body-fixed to inertial coordinate and $\dot{\underline{V}}^I = [a_X a_Y a_Z]^T$.

$$C_B^I = \begin{bmatrix} C_\theta C_\psi & S_\theta S_\phi C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \qquad (22)$$

Here, $S \equiv \sin(\dots)$ and $C = \cos(\dots)$ are Sine and Cosine operators. To model the rotor, the forces generated by each rotor are calculated by blade element theory (BET). This theory determines the forces of the blade element at (r_e, ψ_b) and then integrates over the blade radius from root cutout to the blade effective radius, BR, where $B\left(=1-\sqrt{2C_T}/b\right)$ is the blade tip loss factor, C_T is thrust coefficient and b is the number of blades. In forward flight, the rotor forces have to be averaged over a rotor revolution to obtain the trim condition. The aerodynamic load on the blade element provides a shear load and a moment at the blade root. In a conventional rotor, the flapping hinge at the blade root avoids much of this load from being transferred to the fuselage. A stiff rotor, such as those on a typical quadrotor, however, transfer these moments, and thus, are calculated in this work. Accordingly, the rotor thrust, drag, and side force coefficients are corresponded to:

$$C_T = \frac{a\sigma}{2}\left(\frac{B^3}{3}\theta_0 - \frac{B^2}{2}\mu_z - \frac{B^2}{2}\lambda_0 + \frac{B^4}{4}\theta_1 + \frac{B}{2}\mu_x^2\theta_0 + \frac{B}{2}\mu_y^2\theta_0 + \frac{B^2}{4}\mu_x^2\theta_1 + \frac{B^2}{4}\mu_y^2\theta_1\right)$$

$$
\begin{aligned}
C_H = B\frac{\sigma a}{2}\Big(B\frac{C_{d_0}}{2a}\mu_x - \frac{1}{2}\lambda_0\mu_x\theta_0 \\
+\frac{1}{2}\mu_x\mu_z\theta_0 \\
-\frac{1}{4}B\lambda_0\mu_x\theta_1 \\
+\frac{1}{4}B\mu_x\mu_z\theta_1\Big) \\
C_Y = B\frac{\sigma a}{2}\Big(B\frac{C_{d_0}}{2a}\mu_y - \frac{1}{2}\lambda_0\mu_y\theta_0 \\
+\frac{1}{2}\mu_y\mu_z\theta_0 \\
-\frac{1}{4}B\lambda_0\mu_y\theta_1 \\
+\frac{1}{4}B\mu_y\mu_z\theta_1\Big)
\end{aligned} \tag{23}
$$

Also, the moment coefficients are given as (see Fig. 4):

$$
\begin{aligned}
CM_x = -B^2\frac{\sigma a}{2}\Big(\frac{1}{4}\lambda_0\mu_x - \frac{1}{4}\mu_x\mu_z \\
+\frac{1}{3}B\mu_x\theta_0 + \frac{1}{4}B^2\mu_x\theta_1\Big) \\
CM_y = B^2\frac{\sigma a}{2}\Big(\frac{1}{4}\lambda_0\mu_y - \frac{1}{4}\mu_y\mu_z + \frac{1}{3}B\mu_y\theta_0 \\
+\frac{1}{4}B^2\mu_y\theta_1\Big) \\
CM_z = B^2\frac{\sigma a}{2}\Big(\frac{C_{d_0}}{4a}B^2 - \frac{1}{2}\lambda_0^2 - \frac{1}{2}\mu_z^2 \\
+\frac{C_{d_0}}{4a}\mu_x^2 + \frac{C_{d_0}}{4a}\mu_y^2 \\
+\lambda_0\mu_z - \frac{1}{4}B^2\lambda_0\theta_1 \\
+\frac{1}{4}B^2\mu_z\theta_1 - \frac{1}{3}B\lambda_0\theta_0 \\
+\frac{1}{3}B\mu_z\theta_0\Big)
\end{aligned} \tag{24}
$$

where $\mu_x = v_x^{hub}/v_t$, $\mu_y = v_y^{hub}/v_t$, and $\mu_z = v_z^{hub}/v_t$.

$$
\begin{aligned}
v_x^{hub} &= u + ry_i \\
v_y^{hub} &= v + rx_i \\
v_z^{hub} &= w - py_i - qx_i
\end{aligned} \tag{25}
$$

Here, $x_i$and $y_i$are the coordinate of the $i-th$ rotor. The rotor coefficients are $C_T(= T/\rho A v_t^2)$, $C_H(= H/\rho A v_t^2)$, and $C_Y(= Y/\rho A v_t^2)$, and the rotor solidity is $\sigma = bc/\pi R$. In addition, a is lift curve slope, θ_0 is blade collective pitch, θ_1 is blade twist, C_{d_0} is profile drag coefficient, and λ_0 represents the rotor inflow ratio. Also, b is the number of blades, c denotes blade chord length, V_∞ forward flight speed, α_{TPP} is tip path plane angle of attack (see Fig. 4), and $v_t = R\Omega$ the blade tip speed. The inflow ratio is then obtained as [35]:

$$
\lambda_0 = \mu\tan\alpha_{TPP} + \frac{C_T}{2\sqrt{\mu^2+\lambda_0^2}} \tag{26}
$$

For a given advance ratio, thrust coefficient, and α_{TPP}, the solution to inflow ratio was provided using NR method in this paper where the iteration started with $\lambda_{n=0} = \sqrt{C_T/2}$ and the convergence occurred when $\left\|\frac{\lambda_{n+1}-\lambda_n}{\lambda_{n+1}}\right\| \le 10^{-5}$.

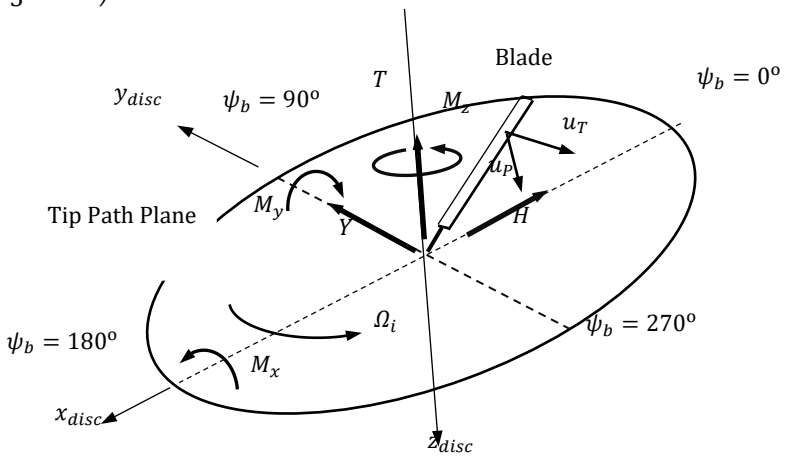


Fig. 4.Presentation of the rotor tip path plane (TPP)

JAST, Vol. 12, No. 1, pp. 37-51

JAST Journal of Aerospace Science and Technology

The equations of motion of the fuselage are those of a rigid body, formulated in the body-fixed coordinate system. In this paper, the magnitudes of $p, q,$ and $r$are assumed to be small, and the net angular momentum of the rotors is also close to zero since the momentum of the clockwise rotors cancels out that of the counter-clockwise motors. Therefore, the gyroscopic moments are small and are neglected. To track the position and velocity as well as the attitudes, a total of 12 states are required:

$$\begin{aligned} -\sum_{i=1}^{4} H_i - X_F &= m(\dot{u} + qw - vr \\ &\quad + g\sin\theta) \\ \sum_{i=1}^{4} Y_i + Y_F &= m(\dot{v} + ru - wp \\ &\quad - g\sin\phi\cos\theta) \\ -\sum_{i=1}^{4} T_i - Z_F &= m(\dot{w} + pv - qu \\ &\quad - g\cos\phi\cos\theta) \end{aligned} \tag{27}$$

and

$$\begin{aligned} \sum_{i=1}^{4} (M_{x_i} + T_i y_i) &= I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ \sum_{i=1}^{4} (M_{y_i} + T_i x_i) &= I_y\dot{q} + (I_{xx} - I_{zz})pr \\ \sum_{i=1}^{4} (M_{z_i} + Y_i x_i - H_i y_i) &= I_{zz}\dot{r} + (I_y - I_x)pq \end{aligned} \tag{28}$$

The authors assumed that the contribution of fuselage in x_B direction can be calculated using the equivalent flat plate area approach. Thus, the fuselage drag is given by $X_F = \frac{1}{2}\rho(\mu\Omega R)^2 f_e$, where f_e represents the equivalent flat plate area, and $Y_F = Z_F = 0$ [35].

Maneuvers Definition

Two different maneuvers for the quadrotor including climb-hover and cruise-hover were designed to prove the capabilities of inverse simulation with trust-region DL algorithm. The climb-hover motion is characterized by a rapid 10 meter vertical translation with a constant lateral and longitudinal position andvariable heading angle starting and terminating in a trimmed hover. The duration of the maneuver is specified as 40 seconds. Heading angle is changed up to 63.4 degrees in 30 seconds and climb velocity is 2 knots.

Cruise-hover is characterized by a rapid 22 meter forward flight with constant altitude and heading, starting and terminating in a trimmed hover. The duration of the maneuver is specified as 60 seconds and forward flight velocity is 3 knots.Thesetwo maneuvers were generated by direct simulation of quadrotor with the linear quadratic controller (LQR). Table 1 compares the two maneuvers for the inverse simulation problem.

Table 3 Presentation of the hover-climb and the cruise-hover maneuvers

Flight phase	States variables	Control inputs
Climb-hover	$\{z, \dot{z}, \psi, \dot{\psi}, \phi, \dot{\phi}\}$	$\{u_1, u_2, u_4\}$
Cruise-hover	$\{x, \dot{x}, z, \dot{z}, \theta, \dot{\theta}, \phi, \dot{\phi}\}$	$\{u_1, u_2, u_3\}$

The four control inputs of $\boldsymbol{u}(t) = \{u_1, u_2, u_3, u_4\}$ based on physical requirements of the quadrotor are defined in Table 1. These control vectors are altitude control ($\Delta\Omega_Z$), roll control ($\Delta\Omega_\phi$), pitch control ($\Delta\Omega_\theta$), and yaw control input ($\Delta\Omega_\psi$), respectively. The relationship of the inputs with the rotation speed of each rotor is defined as [36]:

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = \begin{bmatrix} \Omega_n \\ \Omega_n \\ \Omega_n \\ \Omega_n \end{bmatrix} + \begin{bmatrix} +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & +1 & +1 \end{bmatrix} \begin{bmatrix} \Delta\Omega_Z \\ \Delta\Omega_\phi \\ \Delta\Omega_\theta \\ \Delta\Omega_\psi \end{bmatrix} \tag{29}$$

where Ω_n is the nominal speed of each rotor in hover flight. To generate the control vector $\boldsymbol{u}(t)$, the LQR was developed for the linearized equations of motion represented by $\dot{\boldsymbol{x}} = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$ in which $\boldsymbol{x}(t)$ is the state vector. The satisfactory explanation for control vector was $\boldsymbol{u}(t) = -K[\,\boldsymbol{x}(t) - \boldsymbol{x}_{com}(t_f)]$ which is a multiplication of the optimal feedback gain (K) to the difference between the system state $\boldsymbol{x}(t)$ and the system control command $\boldsymbol{x}_{com}(t_f)$. To satisfy the system controllability, in the first step, variables$z$and$\psi$and their rates were accounted as system variables and u_1 and u_4 were considered as the control inputs. Therefore, the A and B matrices are considered as:

$$A=\begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{bmatrix},\qquad B=\begin{bmatrix}0&0\\-0.0632&0\\0&0\\0&0.0635\end{bmatrix} \tag{30}$$

The performance index is then defined as:

$$J=\frac{1}{2}\int_{t_0}^{t_1}[\,\boldsymbol{x}^T(t)\,Q\boldsymbol{x}(t)+\boldsymbol{u}^T(t)R\boldsymbol{u}(t)]dt \tag{31}$$

where semi-positive and positive definite matrices are respectively estimated as:

$$Q=\begin{bmatrix}10&0&0&0\\0&10&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix},\qquad R=\begin{bmatrix}1&0\\0&1\end{bmatrix} \tag{32}$$

It should be noted that the elements of diagonal matrices Q and R are obtained by trial and error. By solving the Riccati equation $A^TP+PA-PBR^{-1}B^TP+Q=0$, the optimal gain matrix for the first step is obtained as:

$$K=\begin{bmatrix}-3.1623&-10.4915&0&0\\0&0&1&5.7005\end{bmatrix} \tag{33}$$

Since u_2 control input is shared between two maneuvers presented in Table 1, here the A and B matrices are reduced to:

$$\mathrm{A}=\begin{bmatrix}0&1\\0&0\end{bmatrix},\qquad \mathrm{B}=\begin{bmatrix}0\\1.21\end{bmatrix} \tag{34}$$

As a result, using the above modification, the improvement in roll angle value against unwanted disturbance was evident. Choosing Q and R as identity matrices, the optimal feedback gain is obtained as:

$$K=[1\quad 1.6288] \tag{35}$$

For the second maneuver, the three variables of x, z, and θ and their rates are accounted as state variables and u_1 and u_3 are defined as control inputs. It should be noted that u_1 is the control input for altitude adjustment and $u_3$is allocated for controlling the position and pitch angle. Based on the author's examination, the system controllability is sufficiently gained by the selected control inputs and the state variables. Accordingly, A and B matrices can be given:

$$\mathrm{A}=\begin{bmatrix}0&1&0&0&0&0\\0&0&0&0&-9.81&0\\0&0&0&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix},\qquad \mathrm{B}=\begin{bmatrix}0&0\\0&0\\0&0\\-0.0632&0\\0&0\\0&-1.21\end{bmatrix} \tag{36}$$

where elements of diagonal matrices Q and R are obtained by trial and error that corresponded as:

$$\mathrm{Q}=\begin{bmatrix}0.01&0&0&0&0&0\\0&0.01&0&0&0&0\\0&0&1000&0&0&0\\0&0&0&1000&0&0\\0&0&0&0&100&0\\0&0&0&0&0&100\end{bmatrix},\qquad \mathrm{R}=\begin{bmatrix}1&0\\0&1\end{bmatrix} \tag{37}$$

Thus, the optimal control gain for the second maneuver is calculated as:

$$K=\begin{bmatrix}0&0&-31.6228&-44.7294&0&0\\0.1&0.5595&0&0&-14.8623&-11.1609\end{bmatrix} \tag{38}$$

Results

Direct simulation with LQR

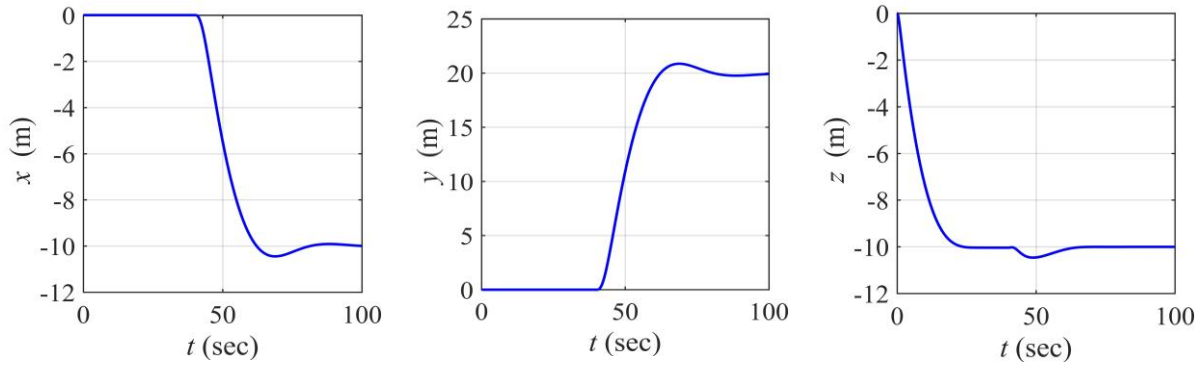
Figs. 5 and 6 show the results of direct simulation for the developed non-linear model of a typical quadrotor with optimal gains determined by Eqs. (33), (35), and (38). The aim of the direct simulation is to achieve the final point at (-10, 20, -10) from the reference position (0,0,0) defined as a starting point of flight. Table 2 provides the general specifications of the quadrotor used in this simulation [36]. Fig. 6 confirms that we have obtained satisfactory results by the direct simulation after 60 seconds where the values of Euler angles are corresponded to $\psi=-63.4°$, $\phi=0°$, and $\theta=-0.4°$.

Table 4 General specifications of selected quadrotor [36].

Parameter	Symbol	Unit	Value
Mass	m	kg	0.52
Moment of inertia about x-axis	I_{xx}	kg.m^2	6.23×10^{-3}
Moment of inertia about y-axis	I_{yy}	kg.m^2	6.23×10^{-3}
Moment of inertia about z-axis	I_{zz}	kg.m^2	1.12×10^{-2}
Hub to hub distance	l	m	0.23
Rotation speed	Ω_0	rad/s	311.7

Parameter	Symbol	Unit	Value
Induced power correction factor	κ		1.13
Profile drag coefficient	C_{d_0}		0.008
Number of blades	b		2
Blade chord	c	m	0.016
Blade radius	R	m	0.13

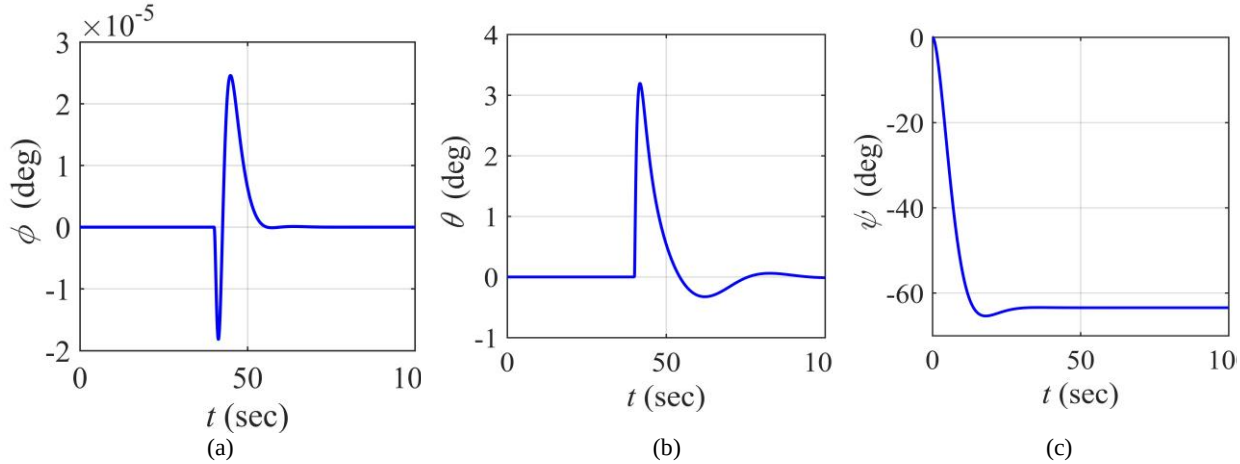
Parameter	Symbol	Unit	Value
Blade twist	θ_1	deg	-10
Blade collective pitch	θ_0	deg	20
Lift curve slope of blade	a	1/rad	5.7
Rotor solidity	σ		0.0784
Blade loading	C_T/σ		0.14



(a) (b) (c)

Fig. 5. Time histories of quadrotor position calculated by direct simulation with LQR controller

Fig. 6 illustrates the trajectory of the quadrotor obtained by direct simulation with LQR controller at $C_T/\sigma = 0.14$. Also, Fig. 6 shows the variations of heading, roll, and pitch angles for the trajectory shown in Fig. 5. It can be seen that the heading angle occurs at -63.4° where the nose of quadrotor is in direction with the line connected to final point. In this case, the altitude also increases 10 m and thus, the quadrotor reaches to the desired altitude. The results show that, after 40 seconds, a positive pitch angle of 3.5° is required to reach the final point while the altitude is hold at the specific of 10 m. As shown in Fig. 6(b), at 62 seconds, when the longitudinal error with respect to the final point is as large as 0.4 m, a pitch angle of -0.4° is required to turn the quadrotor back to the final point. Accordingly, decreasing quadrotor pitch angle is continued for 80 seconds and therefore the hover condition is obtained.



(a) (b) (c)

Fig. 6. Time histories of quadrotor attitude calculated from direct simulation using LQR method.

Inverse simulation with Newton-Raphson algorithm

Fig. 7(a) through (d) compare the four control inputs predicted by inverse simulation based on NR algorithm and direct simulation with LQR controller. The results of inverse simulation are achieved where the unknown vector is defined as $\boldsymbol{Y} = [u_1, u_2, u_3, u_4, \phi, \theta, \psi, \omega_y, \omega_z]$ and the error

vector $\boldsymbol{F}_E$ consists of 6 equations of motion, Eqs. (19) and (20), and three kinematic expressions for Euler angles. Moreover, the initial guess on $\boldsymbol{Y}_0(t = 0)$ is set to $[0.1, 0, 0, 0.1, 0, 0.1, 0, 0, 0.1]$ and the inverse simulation were carried out by the differential method where the step time was adjusted to 0.01 second. The trajectory used in inverse simulation is based on the generated position and attitude previously shown in in Figs. 5 and 6.

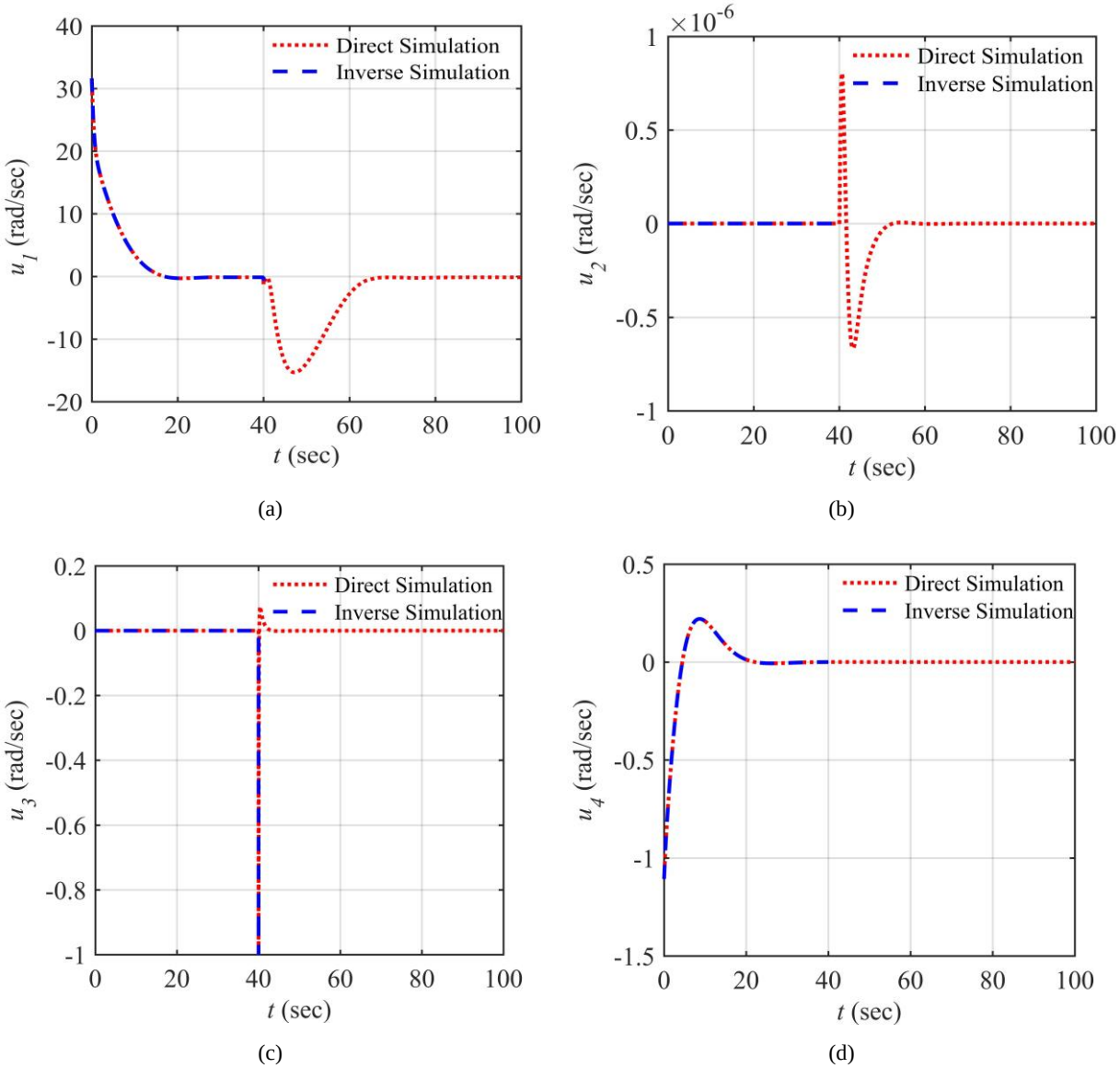


Fig. 7. Control inputs calculated by inverse simulation with NR method and LQR controller

As seen in Fig. 7(a) to (d), in climb-hover maneuver, a good correlation between all four control inputs obtained by inverse simulation and the optimal control inputs designed for direct simulation is obvious. The results confirm that NR method is unable to calculate control inputs after 40 seconds in cruise-hover maneuver. This result is attributed to the location of starting point that is beyond the desired solution.

Inverse simulation with Dogleg algorithm

Fig. 8(a) through (d) compare the time history of the four control inputs determined by inverse simulation based on DL algorithm and direct simulation with LQR controller at $C_T/\sigma = 0.14$. The initial guess on $\boldsymbol{Y}_0$ and the step time of 0.01 second was the same as the NR method.

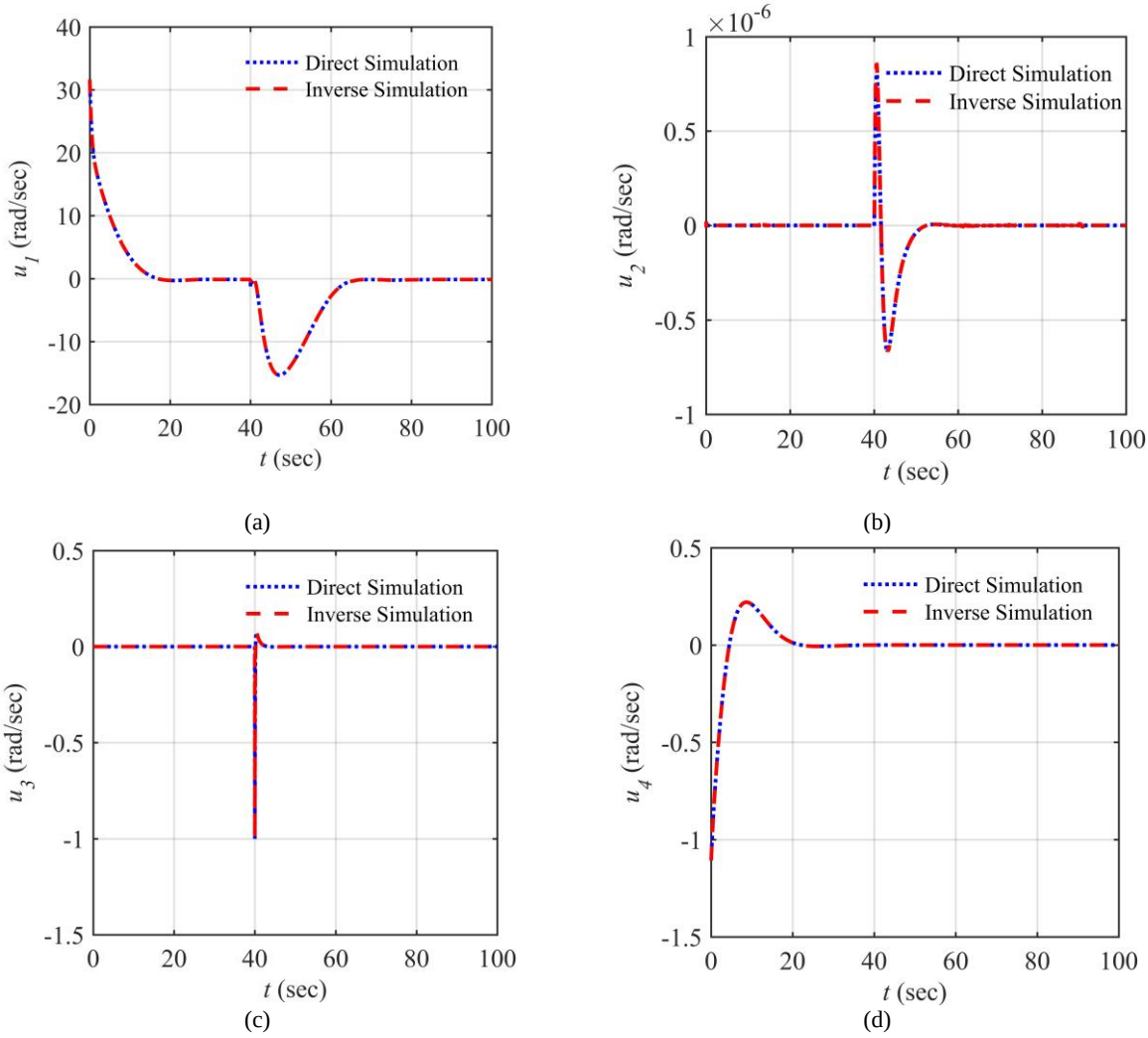


Fig. 8. Control inputs calculated by inverse simulation with trust-region DL method and LQR controller

As it can be seen in Fig. 8(a) to (d), a good correlation is evident between the results of direct simulation with LQR and inverse simulation with DL algorithm in both climb-hover and cruise-hover maneuvers. Unlike the NR method, here the DL solutions involve all four control inputs after 40 seconds for cruise-hover phase. The reason for this result is that the DL algorithm can guarantee the global convergence throughout the computational domain.

Fig. 9 shows iterationloop number (k) in each step (m) for inverse simulation based on NR and DL methods. Figures 9(a) and 9(b) show the first phase of the flight and Fig. 9(c) corresponds to both flight phases as presented in Table 1. Given the total flight time of 100 seconds and the step size of 0.01 seconds for both phases of flight, we will have 10,000 solution steps.

The initial guess of unknowns in the first step ($k = 1$) is considered as $[0.1, 0, 0, 0.1, 0, 0.1, 0, 0, 0.1]$ and in the next step ($k = 2$), the control inputs in the previous steps are considered as the initial guess. As can be seen in Fig. 9 (a), the least iteration loop in the NR method is 2, but in the DL method the convergence is obtained by one iteration only in some steps. The total number of iteration loops for the first phase of the flight is 8202 times in the NR and 7634 in the DL method, indicating a higher speed of DL. In the second phase of the flight (cruise flight) since the initial guess of the control inputs is out of the solution, it will be unstable in the NR procedure but due to the insensitivity of the DL to the initial conditions,

by increasing the number of repeating loops, the unknowns are calculated. The total number of solution steps using DL method is 26421.

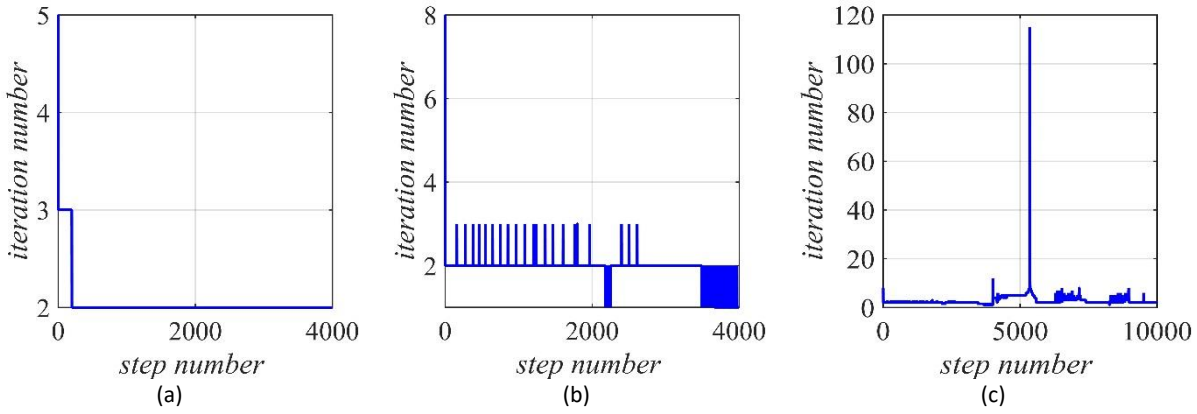
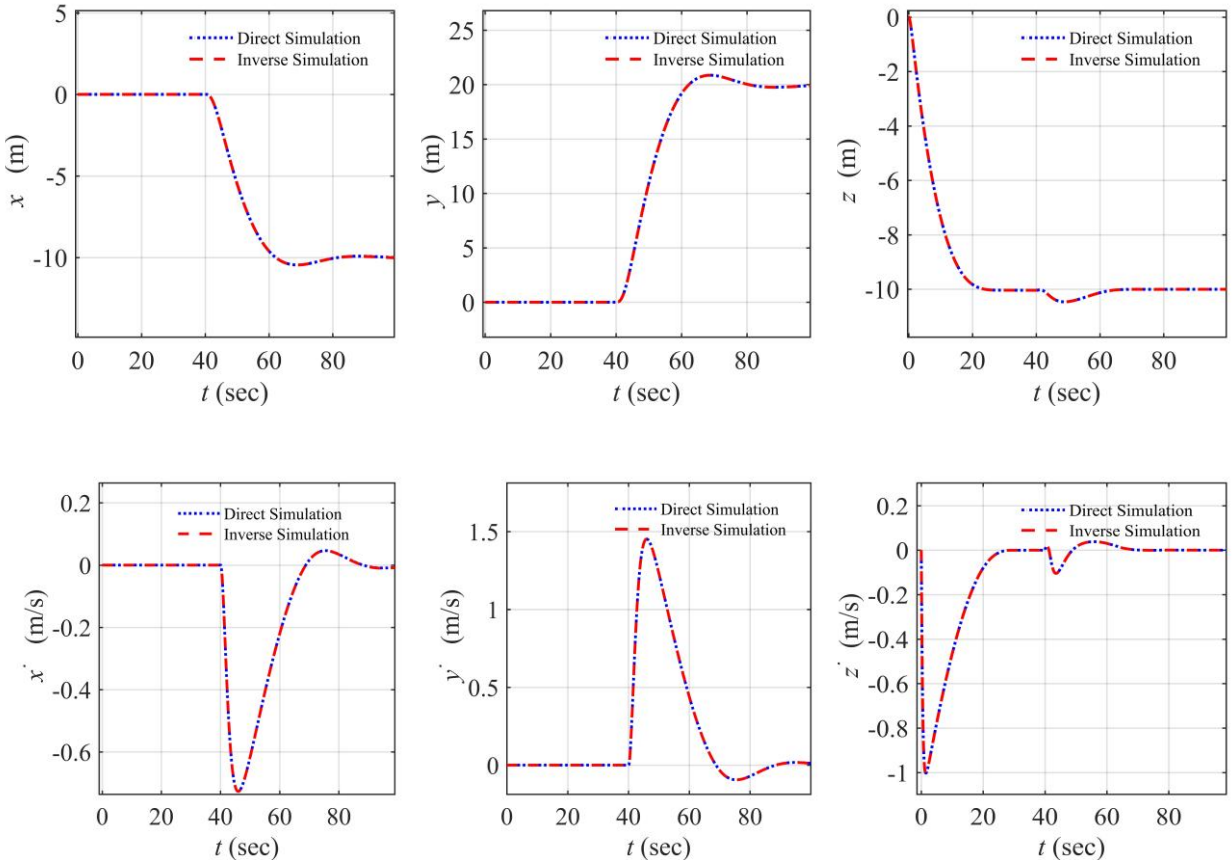


Fig. 9. Comparison of iteration number in each step for inverse simulation based on NR and DL methods

Fig. 10 shows the direct simulation results based on control inputs gained by inverse simulation with DL algorithm. These results are compared to the trajectory generated by LQR controller previously shown in Figs. 5 and 6.



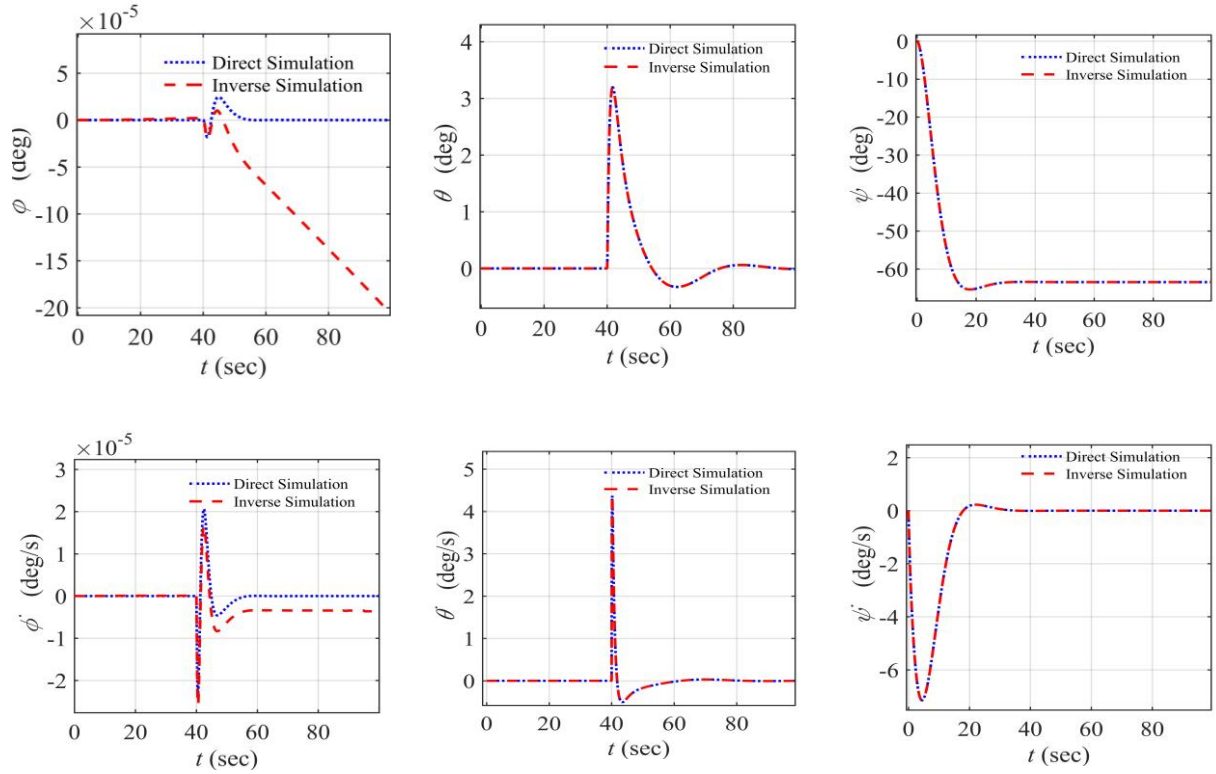


Fig. 10. Comparison of the trajectory for inverse and direct simulaton with LQR controller

We observed that the trajectory obtained by the controller inputs in inverse simulation is in a good agreement with the results of optimal control inputs. The results address that the longitudinal and lateral errors in x and y directions relative to the prescribed trajectory are about 0.04% and 0.02%, respectively. Even though a very slight difference of about 0.0002° between roll angles is obvious, the proposed inverse simulation procedure and the relevant results are reasonably acceptable.

Conclusions

The trust-region Dogleg solution algorithm has been proposed to improve the inverse simulation process. The case study has performed on a non-linear dynamic model of quadrotor with two different maneuvers (i.e., climb-hover and cruise-hover) generated by direct simulation with LQR controller. This work is a step towards enhancing the inverse simulation process and the principal findings are:

1. An inverse simulation algorithm with time differentiation method and trust-region dogleg solver can be applied to the real problems such as V/STOL aircraft maneuvers.
2. Even though the model of quadrotor studied here consists of nonlinear dynamics and linear rotor aerodynamics with a non-uniform inflow ratio, the solution method proposed in this paper is not restricted to these conditions.
3. The process of inverse simulation with trust-region dogleg solution algorithm can guarantee the global convergence of solution for any problems over the computational domain. This algorithm can be employed instead of the traditional Newton-Raphson which suffers from local convergence occurred in the ill-conditioned problems.

References

[1] W. Kevin, D. Thomson, E. McGookin, Application of inverse simulation to a wheeled mobile robot, 2015 6th International Conference on Automation, Robotics and Applications (ICARA). IEEE, (2015).

[2] G. Avanzini, G. De Matteis, A. Torasso, Assessment of helicopter model accuracy through

inverse simulation, J Aircraft. 54(2) (2016) 535-547. https://doi.org/10.2514/1.C033847.

[3] F. Thaleia, E. McGookin, D. Thomson, Numerical stability of inverse simulation algorithms applied to planetary rover navigation, 2016 24th Mediterranean Conference on Control and Automation (MED), IEEE, (2016).

[4] Abhishek, R. Prasad, Helicopter unsteady maneuver analysis using inverse flight dynamics simulation and comprehensive analysis, J Aircraft. 53(6) (2016) 1614-1625. https://doi.org/10.2514/1.C033707.

[5] W. U. Wei, A general method for closed-loop inverse simulation of helicopter maneuver flight, Chinese J Aeronaut. 30(6) (2017)1799-1808. https://doi.org/10.1016/j.cja.2017.07.010.

[6] J. Zhao, Force tracking control of an electro-hydraulic control loading system on a flight simulator using inverse model control and a damping compensator. T I Meas. Control. 40(1) (2018) 135-147. https://doi.org/ 10. 1177/0142331216651326.

[7] A.M. Elsherbiny, Inverse simulation of symmetric flight of a guided gliding subsonic flying body, 2018 AIAA Modeling and Simulation Technologies Conference. (2018). https://doi.org/10.2514/6.2018-0427.

[8] U. Tarun, S. Raha, S. Srivastava, Inverse simulation for gas turbine engine control through differential algebraic inequality formulation, Int. J Turbo. Jet Eng. 35(4) (2018) 373-383. https://doi.org/10.1515/tjj-2016-0057.

[9] D.J. Murray-Smith, Development of an inverse simulation method for the analysis of train performance, P I Mech Eng. F-J Rai. 232(5) (2018) 1295-1308. https://doi.org/10.1177/0954409717720349.

[10] H. Yingzhi, Robust incremental nonlinear dynamic inversion controller of hexapod flight simulator motion system, Advances in Aerospace Guidance, Navigation and Control. (2018) 87-99. https://doi.org/10.1007/978-3-319-65283-2_5.

[11] S. Hassan, Modeling and inverse simulation of generic helicopter maneuvers, 2018 AIAA Modeling and Simulation Technologies Conference (2018). https://doi.org/10.2514/6.2018-1169.

[12] D. J. Murray-Smith, The inverse simulation approach: a focused review of methods and applications, Math Comput Simul. 53(4-6) (2000) 239-247. https://doi.org/10.1016/S0378-4754 (00)00210-X.

[13] O. Kato, I. Sugiura, An interpretation of airplane general motion and control as inverse problem, J Guid. Control Dynam. 9(2) (1986) 198-204. https://doi.org/10.2514/3.20090.

[14] O. Kato, Attitude projection method for analyzing large-amplitude airplane maneuvers, J Guid. Control Dynam. 13(1) (1990) 22-29. https://doi.org/10.2514/3.20513.

[15] R. Bradley, D.G. Thomson, The development and potential of inverse simulation for the quantitative assessment of helicopter handling qualities, Proceedings of AHS/NASA Conference Piloting Vertical Flight Aircraft: Flying Qualities and Human Factors, (1993), pp. 251-263.

[16] E.W. Cheney, D.R. Kincaid, Numerical mathematics and computing, Cengage Learning; 2012.

[17] S. Potyagaylo ,O. Rand, Autonomous targeted flight of a rotary-wing micro aerial vehicle in indoor, GPS-denied environments, Proceedings of The 29th Congress of the International Council of the Aeronautical Science, St. Petersburg, Russia, September 7-12.

[18] R. Bradley, D.G. Thomson, Handling qualities and performance aspects of the simulation of helicopters flying mission task elements, Proceedings of 18th European Rotorcraft Forum, (1992).

[19] D.G. Thomson, Evaluation of helicopter agility through inverse solution of the equations of motion, Doctoral dissertation, University of Glasgow, 1987.

[20] L. Lu, D. J. Murray-Smith, D.G. Thomson, Issues of numerical accuracy and stability in inverse simulation, Simul. Model Pract. Th. 16 (9) (2008) 1350-1364. https://doi.org/10.1016/j.simpat.2008.07.003.

[21] R. Bradley, G.D. Padfield, D.J. Murray-Smith, D.G. Thomson, Validation of helicopter mathematical models, T I Meas. Control. 12(4) (1990) 186-196. https://doi.org/10.1177/014233129001200405.

[22] D.G. Thomson, F. Coton, R. Galbraith, A simulation study of helicopter ship landing procedures incorporating measured flow-field data, P I Mech Eng G-J Aer. 219(5) (2005) 411-427. https://doi.org/10.1243/095441005X30351.

[23] R.A. Hess, C. Gao, A generalized algorithm for inverse simulation applied to helicopter maneuvering flight, J Am Helicopter Soc. 38(4) (1993) 3-15. https://doi.org/10.4050/JAHS.38.3.

[24] D. Anderson, Modification of a generalized inverse simulation technique for rotorcraft flight, P I Mech Eng G-J Aer. 217(2) (2003) 61-73. https://doi.org/10.1243/095441003765208727.

[25] D.G. Thomson, R. Bradley, Inverse simulation as a tool for flight dynamics research-principles and

applications, Prog Aerosp Sci. 42(3) (2006) 174-210. https://doi.org/10.1016/j.paerosci.2006.07.002.

[26] G. Avanzini, G.D. Thomson, A. Torasso, Model predictive control architecture for rotorcraft inverse simulation, J Guid Control Dynam. 36(1) (2012) 207-217. https://doi.org/10.2514/1.56563.

[27] J. Karelahti, K. Virtanen, J. Öström, Automated generation of realistic near-optimal aircraft trajectories, J Guid Control Dynam. 31(3) (2008) 674-88. https://doi.org/10.2514/1.31159.

[28] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J Sci Comput. 11(3) (1990) 450-481. https://doi.org/10.1137/0911026.

[29] J. Nocedal, S. J. Wright, Nonlinear Equations, Numerical optimization, (2006) 270-302.

[30] Conn, N. Gould, P. Toint. Trust region methods, MPS-SIAM Series On Optimization, SIAM, Philadelphia, 2000. https://doi.org/10.1137/1.9780898719857.

[31] X. Lv, L. Xiao, Z. Tan, Z. Yang, J. Yuan, Improved gradient neural networks for solving Moore–Penrose inverse of full-rank matrix, Neural Processing Letters, (2019) 1-13. https://doi.org/10.1007/s11063-019-09983-x.

[32] G. Duleba, I. W. Karcz-Duleba, A comparison of methods solving repeatable inverse kinematics for robot manipulators, Archives of Control Sciences. 28 (2018). https://doi.org/10.24425/119074.

[33] M. Ahookhosh, K. Amini, M. Kimiaei, A globally convergent trust-region method for large-scale symmetric nonlinear systems, Numerical Functional Analysis and Optimization. 36(7) 2015 830-855. https://doi.org/10.1080/01630563.2015.1046080.

[34] M. Goharian, A. Jegatheesan, G.R. Moran, Dogleg trust-region application in electrical impedance tomography, Physiol Meas. 28(5) 2007 555. https://doi.org/10.1088/0967-3334/28/5/009.

[35] G.J. Leishman, Principles of helicopter aerodynamics with CD extra, Cambridge university press, 2006.

[36] H. R. Jafari, M. Zareh, J. Roshanian, A. Nikkhah, An optimal guidance law applied to quadrotor using LQR method, Transactions of the Japan Society for Aeronautical and Space Sciences. 53(4) (2010) 32-39.