Research Article **ISSN: 2395 – 5597**

International Journal of Engineering and Robot Technology

Journal home page: www.ijerobot.com

https://doi.org/10.36673/IJEROBOT.2021.v08.i02.A08

FLYING ROBOTS USED AS HAPTIC INTERFACES

Robert L. Williams II[*1]

[1*]Department of Mechanical Engineering, Ohio University, Athens, Ohio, USA.

ABSTRACT

This paper presents a novel concept for using flying robots (e.g. quadcopters) as versatile, mobile haptic interfaces when interacting with human users immersed in virtual reality (VR). The proposed system will significantly increase the workspace of haptics-augmented VR, when compared to using fixed-base, limited workspace haptic interfaces. A quadcopter dynamics model is presented, a haptics model is presented wherein the 4-dof quadcopter is used to effect 6-dof haptic feedback (impossible to do all 6 at once), followed by preliminary experiments focusing on interaction of a quadcopter and wrench on the environment (human hand).

KEYWORDS

Flying robots, Quadcopters, Dynamic model, Virtual reality, Haptics and Human-machine interface.

Author for Correspondence:

Robert L. Williams II,

Department of Mechanical Engineering,

Ohio University, Athens, Ohio, USA.

Email: williar4@ohio.edu

INTRODUCTION

The popularity of drones (unmanned aerial vehicles, UAVs) has exploded in the last five years. Overwhelmingly, they are being used for outdoor teleoperated surveillance via on-board cameras. Hobbyists with no technical nor pilot background can easily participate via off-the-shelf controllers and standard joysticks. Quadcopter drones are by far the most popular design, but tri-rotors and hexacopters have also been designed and implemented. Interestingly, the popular term quadcopter is a hybridized English word composed of a Latin prefix and a Greek root. The full Greek term would be tetracopter, and the full Latin term is quadrotor[1].

There appears to be no standard dynamic model for quadcopters for navigation and control. Many authors have developed their own model, focusing on the most pertinent aspects for the topic at hand. Authors have linearized these models to a greater or lesser degree and also have included aerodynamics effects to a greater or lesser degree. Luukkonen[2] presents quad copter dynamic and aerodynamic modeling, stabilization, and trajectory control, all implemented in simulation only. Mahony *et al*[1]. Present a tutorial on quadrotor dynamics for estimation and control of small quadrotors. The well-known feedback linearization technique from robotics has been successfully applied by many authors to drone control.

Drone applications to date include "exploring and mapping 3-D environments; transporting, manipulating, and assembling objects; and acrobatic tricks such as juggling, balancing, and flips"[1]. Additional current or near-term drone applications include inspections, photographs and filming, interactive museums, personal tour guides, public safety, agricultural crops/livestock monitoring, commercial deliveries, real-estate marketing, news gathering, hybrid flying/land vehicles, search and rescue, police and military uses, and entertainment. GPS is commonly used for translational navigation, and gyros and accelerometers are often used for attitude (orientational) feedback.

Apvrille, *et al*[3] present the project Drone4u to enable autonomous navigation of micro- and mini-UAVs for indoor surveillance, where GPS cannot be used effectively. These authors use economical sensors for flight planning and collision avoidance. Indoor applications appear to be on the rise.

The vast majority of UAV applications to date are controlled by teleoperation, where a human operator (pilot) on the ground commands all flight motions via hand-held input device, either in a direct line of sight or via the camera feedback. The higher quality teleoperated off-the-shelf quadcopters have automatic yaw stabilization and automatic leveling or attitude maintaining to assist the human pilot. More-demanding current and near-term quadcopter applications require more or full autonomy, which is under development by several authors.

Some authors have presented using a haptic interface for bilateral teleoperation of UAVs, for feeling the flight dynamics and/or for assistance in obstacle avoidance, e.g[4,5].

Our literature and patent searches turned up no cases where the quadcopter or UAV itself is used as the haptic interface. This means that our proposed use of quadcopters as mobile, flying haptic interfaces in VR caves, HMD-based VR simulations and related applications is unique. This would enable haptic feedback over a large indoor (or outdoor) range, overcoming a fundamental limitation of current haptic interfaces in such VR environments. Also, success in the proposed project will enable many more UAV applications for contact with the environment, greatly increasing the potential applications of these flying robots. (The genesis of this project was in 2014 and we were slow to publish, thanks to our 2016 in-house invention disclosure for the technology[6]. Since then some similar quadcopter/haptics works have come to our attention[7-11]; these excellent references are pushing the state-of-the-art in this new area. However, they had no role in affecting our development, due to our pre-dating their work).

This paper first presents a quadcopter dynamics model, then a haptics model, ending with preliminary experimental data, interacting with a quadcopter.

QUADCOPTER HAPTIC INTERFACE CONCEPT

Current haptic interfaces are expensive and, more importantly, have a very small workspace. With unmanned rotorcraft, the haptic interface can fly around the entire space for VR simulation without limitations. The human user (see Figure No.1) physically grasps a handle on one or more autonomously-controlled rotorcraft to: 1. Give pose (position and orientation) inputs to the VR computer, and 2. Feel force/moment and/or touch feedback from the VR simulation enabled by the rotorcraft motors and propellers. Indoor and outdoor VR environments will benefit. Applications include telerobotics, remote control of unmanned aerial vehicles (UAVs), education, training, haptic-enabled VR simulations, design, collaboration, VR gaming, and entertainment.

Haptics has been enabled in the past primarily for sit-down VR simulations, where the user does not move significantly, so commercial base-mounted haptic interfaces with limited workspace may be used for the input/haptic output. This innovation represents the first device to allow haptic feedback free-ranging across the whole space (e.g. 20' x 20' x 20' room), where the haptic interfaces literally fly unobtrusively around with the user, allowing pose inputs and haptic feedback throughout the entire VR simulation space. Multiple flying haptic interfaces can be deployed and programmed in the same VR simulation for maximum flexibility.

Unmanned rotorcraft haptic interfaces can also be widely applied as aerial robots, interacting in contact with the environment to achieve useful tasks in ways that are not possible currently. Many tasks that industrial, manufacturing, service, and hazardous-environment robots currently do can be done by unmanned rotorcraft robots that contact the environment safely and effectively while hovering.

QUADCOPTER MODEL DESCRIPTION

The quadcopter, or quad rotor, is arguably the most popular unmanned aerial vehicle (UAV). It consists of 4 motors/rotors/propellers arranged as shown in Figure No.2. By applying a voltage to each motor, current is applied which causes a torque τ_i on each propeller. Other possible input variables for each motor are thrust $\mathbf{f}_i$ and angular velocity $\dot{\theta}_i$.

As seen in the diagram above, a quadcopter must have alternating directions for the propellers in order to nullify the yaw torque. In this model motors 1 and 3 are chosen to drive their propellers in the *ccw* direction (angular velocity direction, from the top view), while model motors 2 and 4 are chosen to drive their propellers in the *cw* direction. Due to the propeller blade angle design, all 4 propellers impart a vertical lift, opposite gravity g. The nominal flight direction is along X_1 (but the quadcopter can fly along any vector).

The six-d of quad copter is capable of *XYZ* translational and roll-pitch-yaw rotational control during flight. One interesting problem of the quadcopter is the fact that it is underactuated, i.e. only 4 motors (in standard designs, all pointing in the same direction as shown) to control the 6-dof of general flight. Therefore, some of the rotational and translational motions are coupled. The variable and constant parameters used in the quadcopter dynamics model are summarized in Table No.1.

The inertial Cartesian reference frame is {0}. The body-fixed moving Cartesian coordinate frame is {1}, whose origin is located at the quadcopter center of gravity, *CG*. At zero *Z-Y-X* yaw-pitch-roll (ψ–θ–ϕ) Euler angles, the orientation of {1} is identical to that of {0}.

QUADCOPTERKINEMATICSMODEL

Position vectors fixed in the moving frame {1} are:

$$ {}^{1}\mathbf{r}_1=\begin{Bmatrix} r\\ 0\\ 0\end{Bmatrix} \qquad {}^{1}\mathbf{r}_2=\begin{Bmatrix} 0\\ -r\\ 0\end{Bmatrix} \qquad {}^{1}\mathbf{r}_3=\begin{Bmatrix} -r\\ 0\\ 0\end{Bmatrix} $$

$$ {}^{1}\mathbf{r}_4=\begin{Bmatrix} 0\\ r\\ 0\end{Bmatrix} \qquad {}^{1}\left\{{}^{CG}\mathbf{P}_P\right\}=\begin{Bmatrix} p_x\\ p_y\\ p_z\end{Bmatrix} $$

The Cartesian translational position, velocity, and acceleration vectors, with respect to the inertial frame, respectively, are:

$$ {}^{0}\mathbf{P}_1=\begin{Bmatrix} x\\ y\\ z\end{Bmatrix} \qquad {}^{0}\dot{\mathbf{V}}_1=\begin{Bmatrix} \dot{x}\\ \dot{y}\\ \dot{z}\end{Bmatrix} \qquad {}^{0}\ddot{\mathbf{A}}_1=\begin{Bmatrix} \ddot{x}\\ \ddot{y}\\ \ddot{z}\end{Bmatrix} $$

The 3D angular convention used is the yaw-pitch-roll, *Z-Y-X* (ψ–θ–ϕ) Euler angles (see the above diagram):

$$ \{\varphi\}=\begin{Bmatrix} \psi\\ \theta\\ \phi\end{Bmatrix} $$

Euler angle convention involves a series of three rotations about the moving (body) axes {1}. The set of Euler angles φ is NOT a vector (for large angle motions). The *Z-Y-X* (ψ–θ–ϕ) Euler angles convention leads to the following orthonormal rotation matrix $\left[{}^{0}_{1}\mathbf{R}\right]$, which gives the orientation of the moving frame {1} relative to the inertial frame {0}.

$$ \left[{}^{0}_{1}\mathbf{R}\right]=\begin{bmatrix} c\psi c\theta & -s\psi c\phi+c\psi s\theta s\phi & s\psi s\phi+c\psi s\theta c\phi\\ s\psi c\theta & c\psi c\phi+s\psi s\theta s\phi & -c\psi s\phi+s\psi s\theta c\phi\\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} $$

Where the following abbreviations were used, for each of the three Euler angles (ψ–θ–ϕ):

$$c\alpha = \cos\alpha$$

$$s\alpha = \sin\alpha$$

The following transformation, a function of the three Euler angles, relates the unique angular velocity vector $\left\{ {}^{0}\boldsymbol{\omega}_{1} \right\}$ to the set of Euler angle rates $\{\dot{\varphi}\}$:

$$\left\{ {}^{0}\boldsymbol{\omega}_{1} \right\} = [\mathbf{M}]\{\dot{\varphi}\}$$

$$\begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} = \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\phi & c\phi & 0 \\ c\theta c\phi & -s\phi & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{Bmatrix}$$

Where $\dot{\varphi} = \left\{ \dot{\psi} \quad \dot{\theta} \quad \dot{\phi} \right\}$ are the Euler angle rates. The inverse rate relationship for *Z-Y-X* (ψ–θ–ϕ) Euler angle convention is:

$$\{\dot{\varphi}\} = [\mathbf{M}]^{-1} \left\{ {}^{0}\boldsymbol{\omega}_{1} \right\}$$

$$\begin{Bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{Bmatrix} = \begin{bmatrix} 0 & \dfrac{s\phi}{c\theta} & \dfrac{c\phi}{c\theta} \\ 0 & c\phi & -s\phi \\ 1 & s\phi t\theta & c\phi t\theta \end{bmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}$$

Where: $t\theta = \tan\theta = \dfrac{\sin\theta}{\cos\theta}$

Note there is an algorithmic singularity; when $\theta = \pm 90^\circ$ matrix $[\mathbf{M}]$ cannot be inverted in the above transformation.

The angular acceleration vector of the moving frame {1} with respect to the inertial frame {0} is:

$$\left\{ {}^{0}\boldsymbol{\alpha}_{1} \right\} = \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{Bmatrix} = \begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{Bmatrix}$$

$$\left\{ {}^{0}\boldsymbol{\alpha}_{1} \right\} = \left[\dot{\mathbf{M}} \right]\{\dot{\varphi}\} + [\mathbf{M}]\{\ddot{\varphi}\}$$

$$\left\{ {}^{0}\boldsymbol{\alpha}_{1} \right\} = \begin{bmatrix} -c\theta\dot{\theta} & 0 & 0 \\ -s\theta\dot{\theta}s\phi + c\theta c\phi\dot{\phi} & -s\phi\dot{\phi} & 0 \\ -s\theta\dot{\theta}c\phi - c\theta s\phi\dot{\phi} & -c\phi\dot{\phi} & 0 \end{bmatrix} \begin{Bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{Bmatrix} + \begin{bmatrix} -s\theta & 0 & 1 \\ c\theta s\phi & c\phi & 0 \\ c\theta c\phi & -s\phi & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\psi} \\ \ddot{\theta} \\ \ddot{\phi} \end{Bmatrix}$$

The 4x4 homogeneous transformation matrix description of the quadcopter pose (position and orientation) is[12]:

$$\left[{}^{0}_{1}\mathbf{T} \right] = \begin{bmatrix} \left[{}^{0}_{1}\mathbf{R} \right] & & \left\{ {}^{0}\mathbf{P}_{1} \right\} \\ 0 \quad 0 & 0 & 1 \end{bmatrix}$$

QUADCOPTER NEWTON-EULER DYNAMICS EQUATIONS OF MOTION

The quadcopter free-body diagram is Figure No.3. The wrench applied to the environment by the quadcopter at point *P*, in the coordinates of the moving frame {1}, is:

$${}^{1}\mathbf{W}_{E} = {}^{1}\begin{Bmatrix} \mathbf{F}_{E} \\ \mathbf{M}_{E} \end{Bmatrix} = \begin{Bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{Bmatrix}$$

Therefore, by Newton's First Law, the wrench reacting back to the quadcopter at point *P* is (as shown in the FBD Figure No.3):

$$-{}^{1}\mathbf{W}_{E} = {}^{1}\begin{Bmatrix} -\mathbf{F}_{E} \\ -\mathbf{M}_{E} \end{Bmatrix}$$

The individual thrust force magnitude $f_i(t)$ for each motor *i* is proportional to the square of its angular velocity $\dot{\theta}_i(t)$. The constant of proportionality is the propeller lift coefficient C_L.

$$f_i(t) = C_L \dot{\theta}_i^2(t)$$

This thrust is created upward in the direction of the rotor axis, for all four motors/propellers:

$${}^{1}\mathbf{f}_{i}(t) = \begin{Bmatrix} 0 \\ 0 \\ C_L \dot{\theta}_i^2(t) \end{Bmatrix}$$

$${}^{0}\mathbf{f}_{i}(t) = {}^{0}_{1}\mathbf{R}\,{}^{1}\mathbf{f}_{i}(t) = \begin{Bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{Bmatrix} C_L \dot{\theta}_i^2(t)$$

Newton's Second Law

Newton's Second Law written for the quadcopter is:

$$\sum \mathbf{F} = m\mathbf{A}_G$$

$$\sum_{i=1}^{4} {}^{0}\mathbf{f}_{i} + {}^{0}\mathbf{w} - {}^{0}_{1}\mathbf{R}\,{}^{1}\mathbf{F}_{E} = m\,{}^{0}\mathbf{A}_{G}$$

Where the acceleration of the CG, $\mathbf{A}_G$, is with respect to the inertial frame (in this case, also expressed in the coordinates of that frame, {0}). The total thrust in the direction of the {1} frame *Z* axis is:

$$ {}^{1}\mathbf{F} = \sum_{i=1}^{4} {}^{1}\mathbf{f}_i = \begin{Bmatrix} 0 \\ 0 \\ C_L \sum_{i=1}^{4} \dot{\theta}_i^2 \end{Bmatrix} $$

$$ {}^{0}\mathbf{F} = {}^{0}_{1}\mathbf{R}\,{}^{1}\mathbf{F} = \begin{Bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{Bmatrix} C_L \sum_{i=1}^{4} \dot{\theta}_i^2 $$

The translational dynamics equation then becomes:

$$ \begin{Bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{Bmatrix} T + \begin{Bmatrix} 0 \\ 0 \\ -mg \end{Bmatrix} - \begin{Bmatrix} {}^{0}f_x \\ {}^{0}f_y \\ {}^{0}f_z \end{Bmatrix} = m \begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{Bmatrix} $$

Where *T* is the total thrust in the moving frame:

$$ T = C_L \sum_{i=1}^{4} \dot{\theta}_i^2 $$

And the force $\mathbf{F}_E$ applied to the environment by the quadcopter is expressed in {0} coordinates. It can be given in {0} coordinates directly or transformed from {1} coordinates as follows:

$$ {}^{0}\mathbf{F}_E = {}^{0}_{1}\mathbf{R}\,{}^{1}\mathbf{F}_E $$

$$ \begin{Bmatrix} {}^{0}f_x \\ {}^{0}f_y \\ {}^{0}f_z \end{Bmatrix} = \left[{}^{0}_{1}\mathbf{R} \right] \begin{Bmatrix} {}^{1}f_x \\ {}^{1}f_y \\ {}^{1}f_z \end{Bmatrix} $$

Euler's Rotational Dynamics Law

The rotor angular velocity $\dot{\theta}_i(t)$ and acceleration $\ddot{\theta}_i(t)$ create a torque $\tau_i(t)$ about the rotation axis for each motor *i* shaft:

$$ \tau_i(t) = I_P \ddot{\theta}_i(t) + C_D \dot{\theta}_i^2(t) $$

Where I_P is the scalar mass moment of inertia of the rotor and propeller about the rotational axis, and C_D is the drag coefficient.

The rotor forces and torques create the torque vector about the moving frame axes as follows:

$$ {}^{1}\{\boldsymbol{\tau}_1\} = \begin{Bmatrix} \tau_{X_1} \\ \tau_{Y_1} \\ \tau_{Z_1} \end{Bmatrix} = \begin{Bmatrix} r(f_2 - f_4) \\ r(f_1 - f_3) \\ \sum_{i=1}^{4} \tau_i \end{Bmatrix} = \begin{Bmatrix} rC_L(\dot{\theta}_2^2 - \dot{\theta}_4^2) \\ rC_L(\dot{\theta}_1^2 - \dot{\theta}_3^2) \\ \sum_{i=1}^{4} \tau_i \end{Bmatrix} $$

From Euler's Rotational Dynamics Law, the external torque $\{\boldsymbol{\tau}_1\}$ must be equal to the sum of the inertial torque $[\mathbf{I}_G]\{\boldsymbol{\alpha}\}$, the centripetal forces torque $\{\boldsymbol{\omega}\} \times [\mathbf{I}_G]\{\boldsymbol{\omega}\}$ and the gyroscopic forces torque $\{\boldsymbol{\tau}_{GYR}\}$:

$$ \{\boldsymbol{\tau}_1\} - \{\mathbf{M}_E\} - \{\mathbf{r}_E\} \times \{\mathbf{F}_E\} = [\mathbf{I}_G]\{\boldsymbol{\alpha}\} + \{\boldsymbol{\omega}\} \times [\mathbf{I}_G]\{\boldsymbol{\omega}\} + \{\boldsymbol{\tau}_{GYR}\} $$

If we assume the quadcopter is symmetric about its *CG* and the body axes {1} are principal axes, then the mass moment of inertia tensor $[\mathbf{I}_G]$ is not only symmetric and positive-definite, but is also diagonal; further, $I_{xx} = I_{yy}$:

$$ [\mathbf{I}_G] = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} $$

The torque $\{\boldsymbol{\tau}_{GYR}\}$ due to the quadcopter gyroscopic forces is:

$$ \{\boldsymbol{\tau}_{GYR}\} = I_{zz} \{\boldsymbol{\omega}\} \times \{{}^{1}\mathbf{Z}_1\} \dot{\theta}_{GYR} $$

$$ \{\boldsymbol{\tau}_{GYR}\} = I_{zz} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \times \begin{Bmatrix} 0 \\ 0 \\ 1 \end{Bmatrix} \dot{\theta}_{GYR} = \begin{Bmatrix} \omega_y \\ -\omega_x \\ 0 \end{Bmatrix} I_{zz} \dot{\theta}_{GYR} $$

Where the scalar $\dot{\theta}_{GYR}$ is:

$$ \dot{\theta}_{GYR} = \dot{\theta}_1 - \dot{\theta}_2 + \dot{\theta}_3 - \dot{\theta}_4 $$

It is convenient to express the rotational dynamics equation in the coordinates (basis) of the moving frame {1}, since that is where $\{\boldsymbol{\tau}_1\}$, $[\mathbf{I}_G]$, the centripetal forces torque $\{\boldsymbol{\omega}\} \times [\mathbf{I}_G]\{\boldsymbol{\omega}\}$, and the gyroscopic forces torque $\{\boldsymbol{\tau}_{GYR}\}$ are already expressed:

$$ \{\boldsymbol{\tau}_1\} - \{\mathbf{M}_E\} - \{{}^{CG}\mathbf{P}_P\} \times \{\mathbf{F}_E\} = [\mathbf{I}_G]\{\boldsymbol{\alpha}\} + \{\boldsymbol{\omega}\} \times [\mathbf{I}_G]\{\boldsymbol{\omega}\} + \{\boldsymbol{\tau}_{GYR}\} $$

So the Euler rotational dynamics equation of motion is:

$$ \begin{Bmatrix} \tau_{X_1} \\ \tau_{Y_1} \\ \tau_{Z_1} \end{Bmatrix} - \begin{Bmatrix} {}^{1}m_x \\ {}^{1}m_y \\ {}^{1}m_z \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ p \end{Bmatrix} \times \begin{Bmatrix} {}^{1}f_x \\ {}^{1}f_y \\ {}^{1}f_z \end{Bmatrix} = $$

$$ \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{Bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{Bmatrix} + \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix} + \begin{Bmatrix} I_{zz}\dot{\theta}_{GYR}\omega_y \\ -I_{zz}\dot{\theta}_{GYR}\omega_x \\ 0 \end{Bmatrix} $$

$$ \begin{Bmatrix} \tau_{X_1} - {}^{1}m_x - p\,{}^{1}f_y \\ \tau_{Y_1} - {}^{1}m_y + p\,{}^{1}f_x \\ \tau_{Z_1} - {}^{1}m_z \end{Bmatrix} = \begin{Bmatrix} I_{xx}\alpha_x + (I_{zz} - I_{yy})\omega_z\omega_y + I_{zz}\dot{\theta}_{GYR}\omega_y \\ I_{yy}\alpha_y + (I_{xx} - I_{zz})\omega_x\omega_z - I_{zz}\dot{\theta}_{GYR}\omega_x \\ I_{zz}\alpha_z + (I_{yy} - I_{xx})\omega_y\omega_x \end{Bmatrix} $$

QUADCOPTER PSEUDO STATICS EQUATIONS

Pseudostatics analysis refers to when motion is small enough (sufficiently low velocities and accelerations) to ignore dynamic inertial effects. A quadcopter in slow velocity and low acceleration motion is analyzed using statics principles at each motion snapshot. These pseudostatics equations are necessary to command haptic force and moment feedback to the human user's hand, which is constraining the quadcopter.

It is simple to obtain the pseudostatics equations for the quadcopter once the full Newton-Euler dynamics equations of motion are available. Simply set all the velocity and acceleration terms in the dynamics equation to zero to produce the pseudostatics equations. Note: DO NOT set the rotor angular velocities and accelerations to zero, as these are what generate the required forces and moments pseudostatically.

Force Pseudostatics Equations

These are obtained by setting the translational acceleration of the center of gravity to zero in Newton's Second Law:

$$\mathbf{A}_G = \begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

The translational pseudostatics equations then are (components expressed in the basis of the inertial frame {0}):

$$\begin{Bmatrix} s\psi s\phi + c\psi s\theta c\phi \\ -c\psi s\phi + s\psi s\theta c\phi \\ c\theta c\phi \end{Bmatrix} T = \begin{Bmatrix} {}^0f_x \\ {}^0f_y \\ {}^0f_z + mg \end{Bmatrix}$$

Where $\varphi = \{\psi \quad \theta \quad \phi\}$ are the *Z-Y-X* yaw-pitch-roll Euler angles and *T* is the total thrust in the moving frame:

$$T = C_L \sum_{i=1}^{4} \dot{\theta}_i^2$$

And the force $\mathbf{F}_E = \left\{ {}^0f_x \quad {}^0f_y \quad {}^0f_z \right\}^T$ applied to the environment by the quadcopter is expressed in {0} coordinates. It can be given in {0} coordinates directly or transformed from {1} coordinates as follows:

$${}^0\mathbf{F}_E = {}^0_1\mathbf{R}\,{}^1\mathbf{F}_E$$

$$\begin{Bmatrix} {}^0f_x \\ {}^0f_y \\ {}^0f_z \end{Bmatrix} = \left[{}^0_1\mathbf{R} \right] \begin{Bmatrix} {}^1f_x \\ {}^1f_y \\ {}^1f_z \end{Bmatrix}$$

Moment Pseudostatics Equations

The rotor angular velocity $\dot{\theta}_i(t)$ and acceleration $\ddot{\theta}_i(t)$ are NOT set to zero as they are required to create the haptic forces/moments pseudostatically:

$$\tau_i(t) = I_P \ddot{\theta}_i(t) + C_D \dot{\theta}_i^2(t)$$

Where I_P is the scalar mass moment of inertia of the rotor and propeller about the rotational axis, and C_D is the drag coefficient.

The moment pseudostatics equations are obtained by setting the Cartesian angular velocities ${}^0\boldsymbol{\omega}_1 = \left\{\omega_x \quad \omega_y \quad \omega_z\right\}^T$ and angular accelerations ${}^0\boldsymbol{\alpha}_1 = \left\{\alpha_x \quad \alpha_y \quad \alpha_z\right\}^T$ to zero in the Euler Rotational Dynamics equations:

$$\begin{Bmatrix} {}^1m_x \\ {}^1m_y \\ {}^1m_z \end{Bmatrix} = \begin{Bmatrix} \tau_{X_1} - p\,{}^1f_y \\ \tau_{Y_1} + p\,{}^1f_x \\ \tau_{Z_1} \end{Bmatrix}$$

As presented earlier, the rotor forces and torques create the torque vector about the moving frame axes as follows:

$${}^1\{\boldsymbol{\tau}_1\} = \begin{Bmatrix} \tau_{X_1} \\ \tau_{Y_1} \\ \tau_{Z_1} \end{Bmatrix} = \begin{Bmatrix} r(f_2 - f_4) \\ r(f_1 - f_3) \\ \sum_{i=1}^{4} \tau_i \end{Bmatrix} = \begin{Bmatrix} rC_L(\dot{\theta}_2^2 - \dot{\theta}_4^2) \\ rC_L(\dot{\theta}_1^2 - \dot{\theta}_3^2) \\ \sum_{i=1}^{4} \tau_i \end{Bmatrix}$$

Again, it is convenient to express the rotational pseudostatics equations in the coordinates (basis) of the moving frame {1}, since that is where $\{\boldsymbol{\tau}_1\}$ is already expressed.

Wrench component exertion by a Quadcopter

As pointed out earlier, a challenging problem in quadcopter flight control is the fact that it is underactuated, i.e. only 4 motors to control the 6-d of flight. Therefore, some of the rotational and translational motions are coupled. The same problem extends to pseudostatic control. That is, the quadcopter used as a haptic interface can only output 4 haptic effects: f_y OR m_x; f_x OR m_y; f_z; and m_z. That is, f_y and m_x are coupled, and so are f_x and m_z. So we see that the haptic output will be necessarily limited to 4-d of. However, the benefit is that the haptic interface flies with the human user, allowing pose inputs and haptic feedback in an arbitrarily-large VR workspace. Quadcopter design, with additional rotors, can help alleviate this underactuated haptics issue, but this paper only presents a standard quadcopter.

Figure No.4 show how these various body-frame force and moments can be created pseudostatically for haptic feedback in contact with the human hand via a quadcopter. Only one direction for each wrench component is shown below; the opposite directions are obtained by reversing those shown. The red vectors shown indicate the relative propeller thrust required in each case.

All forces and moments shown in Figures No.4 are for the force/moment exerted on the environment by the quadcopter. Therefore, the user's hand must exert the equal and opposite of these to maintain pseudostatic equilibrium. The user feels the force/moments in the directions shown above. Note that the nominal orientation with all three Euler angles zero is shown in the above eight figures. Each of these forces/moments can be generated in the same manner for general yaw-pitch-roll angles $\varphi = \{\psi \quad \theta \quad \phi\}$ since they are in the body frame $\{1\}$.Further note that only individual forces/moments are shown in these figures. Combined forces/moments are also possible, via the pseudostatics model given above.

The final level of the simulation is the rotor thrust model. The rotor model takes the PWM input from the controllers, and determines the rotor thrust via a second order fit from experimental data[13].

EXPERIMENTAL HARDWARE AND RESULTS

Hardware Setup

The hardware used in this experiment consisted of the following items:

Crazyflie 2.0: The crazyflie as seen in fig#9 was used for all of the test flights.

Vicon Motion Capture System: The Vicon Motion capture system is used to collect position data of the Crazyflie, and also provide position feedback control.

Nema 17 V-slot Linear Actuator Kit: The belt driven stepper motor Linear Actuator as seen in Figure No.9 was used to disturb the Crazyflie by hitting it with a light wooden rod.

Tekpower TP3005E DC Power Supply 30V 5A: This power supply was used to power the Arduino motor controller and the Nema 17 stepper motor.

Arduino Unow/Adafruit Motor Shield V2.3: An Arduino Uno with an Adafruit motorshield was used to control the stepper motor.

Dowel rod: Attached to the s linear actuator mount to disturb the Crazyflie.

The dynamic parameters for our model were obtained from[14]. Another important source was[15].

Experimental Methods

Three different experiments were performed in order to test the accuracy of the quadrotor flight simulator. For each of the experiments the Crazyflie (Figure No.5) would start at a predefined origin, and its Cartesian position with respect to that origin would be recorded during the flight.

The first experiment was to have the Crazyflie 2.0 start at the origin (0,0,0) and fly to a point one meter directly above the origin (0,0,1). The position of the Crazyflie would be recorded until it reached a steady state, and then it would slowly decrease in altitude until it landed on the origin.

For the second experiment the Crazyflie started at the origin (0,0,0), then took off to a point one meter directly above the origin (0,0,1). The Crazyflie would wait for fifteen seconds until it reached a steady state, then it would fly to a point in the XY direction (0.5, 0.5, 1). The Crazyflie would wait for fifteen seconds until it reached a steady state, then moved back to the point directly above the origin (0,0,1). The Crazyflie would then slowly return to the origin (0,0,0).

For the final experiment the Crazyflie would take off to a height (0,0,Z), then it would fly to a point (0,Y,Z) where it would hover in front of a small wooden post attached to a linear actuator. The Crazyflie would reach a steady state then the linear actuator would extend and retract, giving the drone an impulse disturbance in the process.

Experimental Results

Of all of the results obtained, the altitude controller results (see Figure No.6 and Table No.2 and Table No.3) are the cleanest fit. When plotted over top of each other the experimental results and the simulation results agree almost perfectly. Although it is hard to see, the plot actually has a red line and a blue line. This is most likely because this is the simplest controller in the whole simulation, being that it is only a single PID controller with no nested controllers. Unfortunately, this controller is not the most important when considering system stability.

When comparing the simulator translational response with the experimental data (see Figure No.7 and Table No.4 and Table No.5) it is clear that the plots do not overlap as seamlessly as the altitude controller. Most notably, the simulation has significantly more overshoot than the actual response. However the peaks and valleys of both data sets align reasonably well.

Since the whole purpose of this simulation is to create feedback control methods for disturbance rejection, the most important set of data is from the impulse disturbance. Because no feasible way to accurately determine impulse magnitude was developed, only the shape of impulse response can be analyzed with any degree of certainty. When the simulation response is scaled to roughly the same size as the experimental data, the frequencies match considerably well (see Figure No.8 and Table No.6). Although the simulator gives no information on the magnitude of the response, it can still be used to compare the effects of different methods for rejecting disturbances.

We added a disturbance estimator to the Crazyflie control architecture, based on[16]. The disturbance estimator observes the quadrotor acceleration and calculates the inertial forces via inverse dynamics. The estimator then looks at the difference between the estimated forces and the controller input forces to determine the disturbance magnitude.

The estimated disturbance is then filtered and added back into the position controller input. This will provide a greater response when the system is subjected to a disturbance without affecting the controllability of the system.

Shown in Figure No.9 above is a plot of the simulation response to two impulse disturbances and a step disturbance. To quantify the magnitude of the disturbances the distance between the maximum and minimum peaks for each response is measured. For both the impulse and step disturbances the maximum displacement is reduced for the case with the disturbance estimator (see Table No.7). In the case of the step response the steady state error is drastically reduced.

When considering the use of quadrotors as haptic interfaces, the use of a disturbance estimator could potentially improve the quadrotors ability to reject disturbances while providing reaction forces to the user. However, further research must be done to determine the effects of the disturbance estimator on the system's stability margins.

Table No.1: Terminology

S.No	Name	Meaning
1	CG	quadcopter center of gravity
2	g	acceleration due to gravity
3	m	quadcopter mass
4	I_G	quadcopter mass moment of inertia tensor about CG
5	$\mathbf{f}_i$	vector thrust force of motor i
6	$\mathbf{r}_i$	position vector from CG to motor i axis
7	τ_i	torque of motor i
8	T	total thrust in the direction of the {1} frame Z axis

9	$\varphi=\{\psi \quad \theta \quad \phi\}$	*Z-Y-X*yaw-pitch-roll Euler angles
10	$\dot{\varphi}=\{\dot{\psi} \quad \dot{\theta} \quad \dot{\phi}\}$	*Z-Y-X* Euler angle rates
11	$\ddot{\varphi}=\{\ddot{\psi} \quad \ddot{\theta} \quad \ddot{\phi}\}$	*Z-Y-X* Euler angle accelerations
12	${}^{0}\boldsymbol{\omega}_1=\{\omega_x \quad \omega_y \quad \omega_z\}^T$	absolute angular velocity of {1} w.r.t. {0}
13	${}^{0}\boldsymbol{\alpha}_1=\{\alpha_x \quad \alpha_y \quad \alpha_z\}^T$	angular acceleration of {1} w.r.t. {0}
14	$\dot{\theta}_i$	angular speed of motor *i*
15	$\ddot{\theta}_i$	angular acceleration of motor *i*
16	${}^{0}\mathbf{P}_1$	position vector from {0} to {1} origins
17	P	contact point of external force
18	C_L	lift coefficient
19	C_D	drag coefficient
20	$\mathbf{F}_E$	external force applied on the environment by the quadcopter
21	$\mathbf{M}_E$	external moment applied on the environment by the quadcopter
22	$\mathbf{r}_E$	moment arm vector from *CG* to *P*

Table No.2: Comparison of simulation and experiment altitude response parameters

S.No	Data set	Rise Time (seconds)	Percent Overshoot (%)	Settling Time < 2% (seconds)
1	Simulator	1.16	57	11.70
2	Experiment	1.19	57	12.10
3	Percent Difference	2.55	0	2.93

Table No.3: Comparison of simulator and experiment altitude peak/valley times

S.No	Simulator Peak/Valley Times	Experiment Peak/Valley Times	Percent difference	Total difference
1	2	2.05	2.47	0.05
2	3.88	3.92	1.02	0.04
3	5.76	5.80	0.69	0.04
4	7.63	7.89	3.35	0.26
5	9.51	9.60	0.94	0.09
6	11.38	11.71	2.85	0.33

Table No.4: Comparison of simulation and experiment translational response parameters

S.No	Data set	Rise Time (seconds)	Percent Overshoot (%)	Settling Time < 2% (seconds)
1	Simulator	.89	35.9	6.8
2	Experiment	1.13	14.4	3.21
3	Percent difference	23.7	85.48	71.7

Table No.5: Comparison of simulator and experiment translational peak/valley times

S.No	Simulator Peak/Valley Times	Experiment Peak/Valley Times	Percent Difference	Total Difference
1	16.38	16.55	1.0	0.17
2	7.59	17.81	1.2	0.22

Table No.6: Times between peaks and valleys of experimental disturbance response

S.No	Impulse	Peak 1 to 2 time	Peak 2 to 3 time
1	1	1.20	1.06
2	2	1.19	1.4
3	3	1.25	1.27
4	4 (plot not shown)	1.29	1.19
5	5 (plot not shown)	1.39	0.72
6	Average	1.26	1.128
7	Simulation results	1.21	1.19
8	Percent difference	4%	5%

Table No.7: Comparison of impulse and step disturbance responses with and without estimator feedback

S.No	Disturbance Type	Peak to peak distance w/ feedback	Peak to Peak Distance w/o feedback	Percent decrease with feedback
1	Impulse	0.1259	0.1678	24.97
2	Step	0.0551	0.1183	114.7

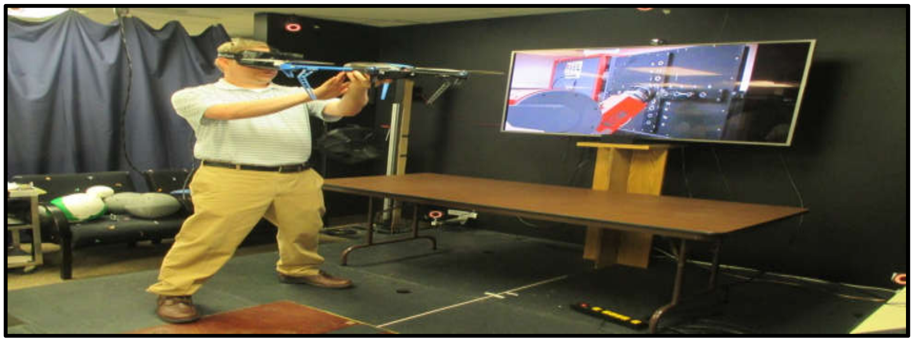

Figure No.1: Quadrotor used as a Haptic Interface in a HMD-based VR Environment

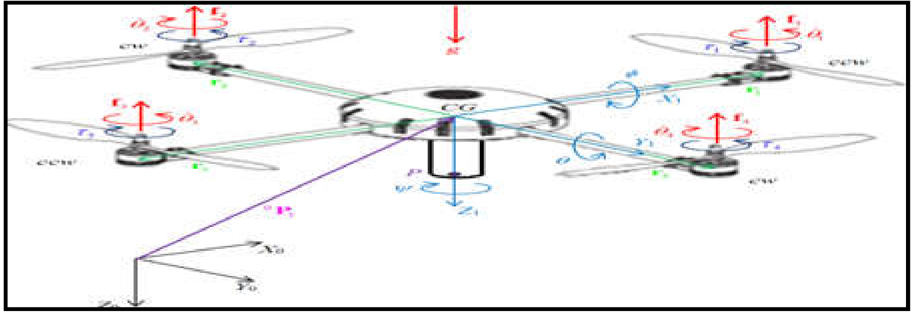

Figure No.2: Quadcopter diagram

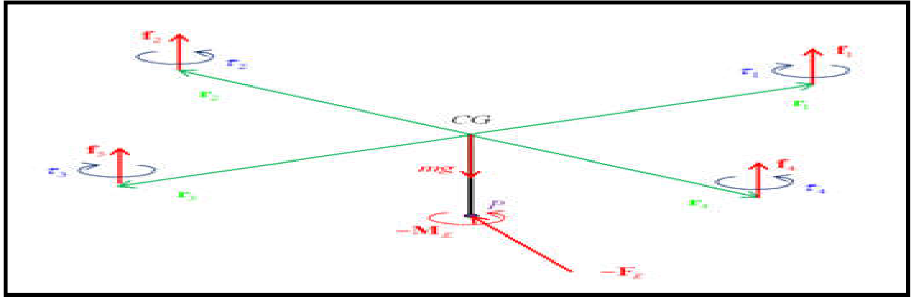

Figure No.3: Quadcopter free-body diagram

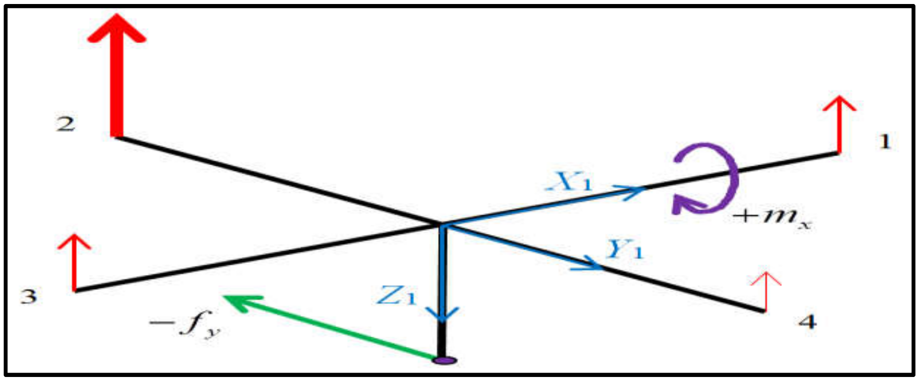

Figure No.4a: Applying f_y OR m_x

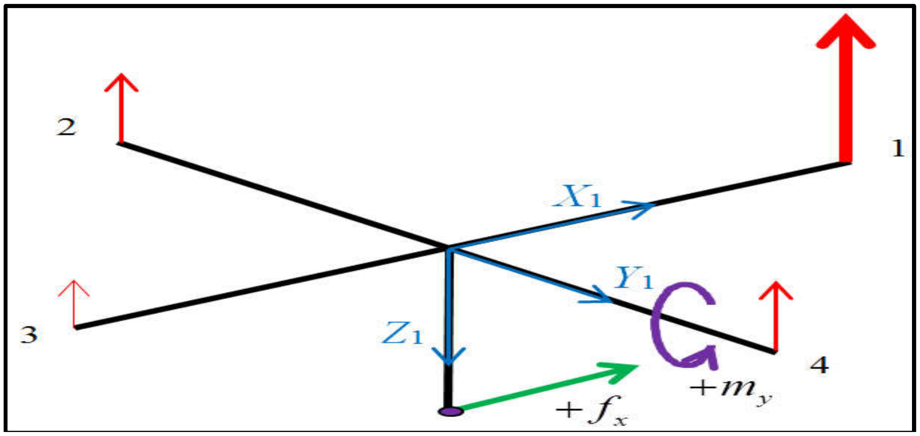

Figure No.4b: Applying f_x OR m_y

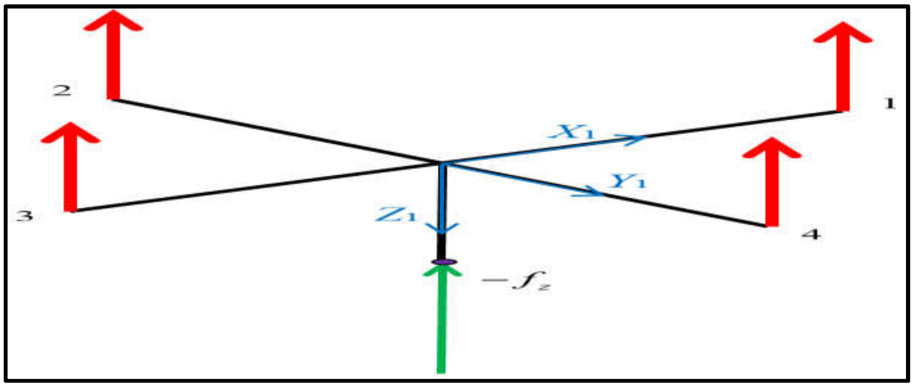

Figure No.4c: Applying f_z

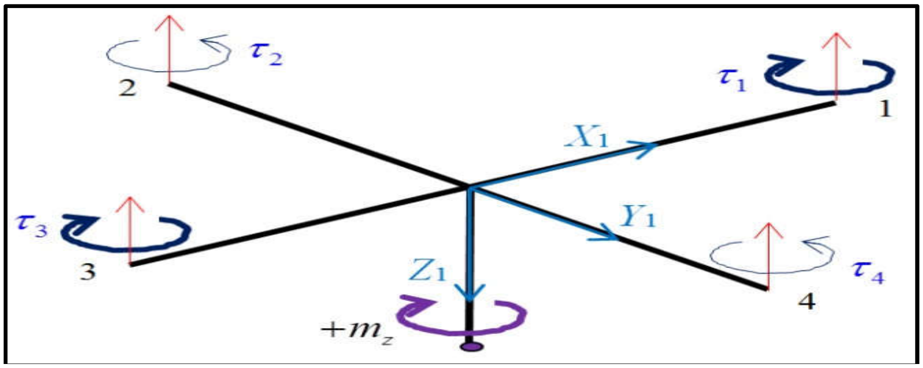

Figure No.4d: Applying m_z

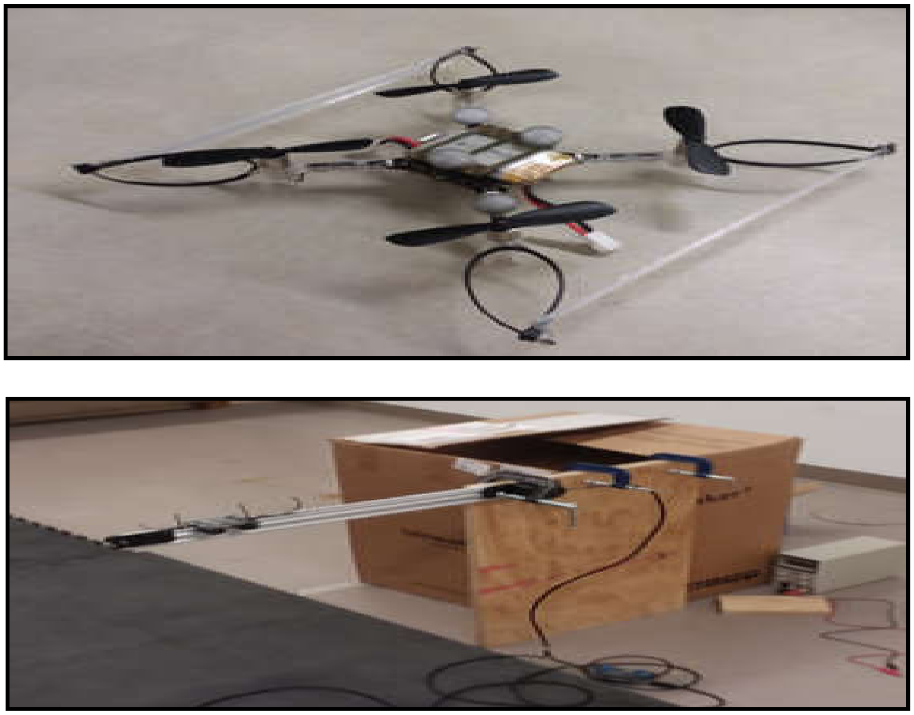

Figure No.5: Crazyflie 2.0 quadrotor used for test flights (top), linear actuator setup for collecting disturbance response data (bottom)

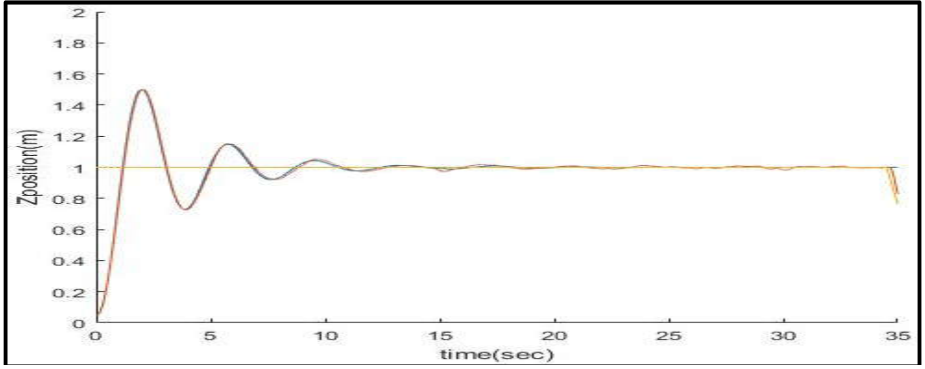

Figure No.6: Comparison of simulator altitude response (blue), actual altitude response (red), and desired altitude (yellow)

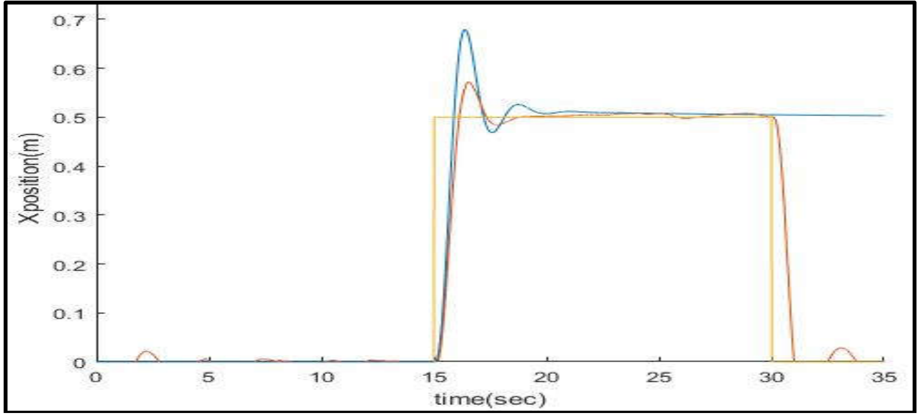

Figure No.7: Comparison of the simulator X direction translational response (blue), actual response (red), and desired position (yellow)

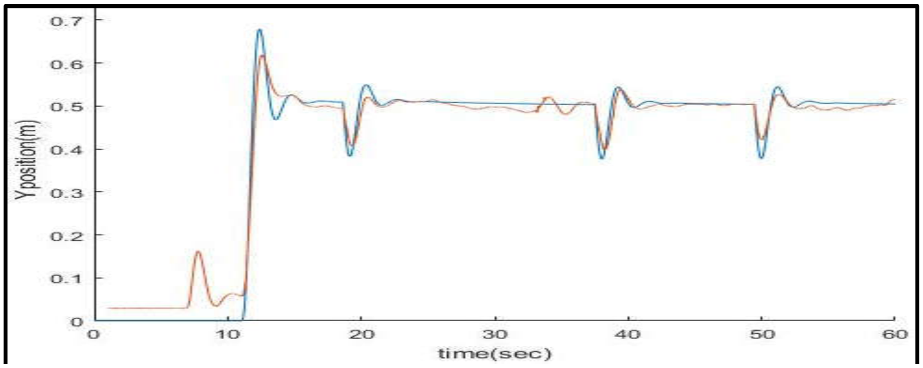

Figure No.8: Simulator disturbance response (blue) and actual disturbance response (red)

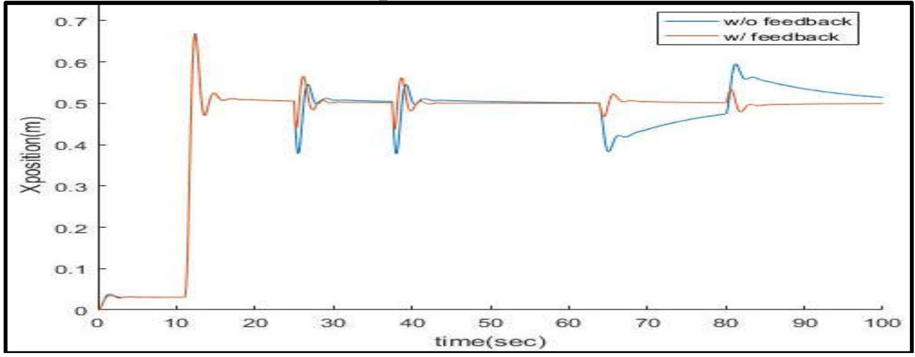

Figure No.9: Comparison of simulation disturbance response with disturbance estimator feedback (red) and without disturbance estimator feedback (blue)

CONCLUSION AND FUTURE WORK

This paper presented a novel concept for using flying robots such as quadcopters for versatile mobile haptic interfaces. Human users immersed in virtual reality (VR) interact with these flying haptic interfaces to feel force and torque feedback. The proposed concept is intended to significantly increase the workspace of haptics-augmented VR, when compared to using fixed-base, limited workspace haptic interfaces. Presented was a quadcopter dynamics model, a quadcopter pseudostatic haptics model, and a summary of our initial experiments focusing on interaction of a quadcopter and wrench on the environment.

Based on the comparison between the simulation and experimental results in this research, this model seems sufficiently accurate to act as a qualitative aid in system design. The true effectiveness of this model cannot be evaluated until it is used to design and implement a real life controller. However, based on the results from the simulator, the use of an acceleration based disturbance estimator for feedback can help reduce the effect of both impulse and step disturbances. In particular, the observer's ability to reject the effect of step disturbances without affecting the controllability of the system could be very useful for turning quadrotors into haptic interfaces. However, the true effectiveness of the estimator and the accuracy of the simulation cannot be fully verified until the controller has been implemented on the actual hardware system.

Further research in this area would require further refinement of the simulation by creating a more accurate model of the Crazyflie 2.0 attitude position and attitude rate controllers. Once the model's response to X and Y waypoint commands better matches the data being collected, further experiment should be performed to compare the stability margins of the simulation the hardware systems stability margins. Finally after the model has been sufficiently refined, the disturbance observer should be implemented in the crazyflie 2.0 firmware. Finally, the performance results of the controller should be compared to the predicted results of the simulation.

ACKNOWLEDGEMENT

The author would like to thank Ohio University assistant professor Dr. Jay Wilhelm for conceptual and lab contributions. The author would also like to thank Ohio University graduate research assistants Ryan Lucas, Garrett Clem, and Gregory Wombold for experimental assistance and data collection in the UAV lab.

CONFLICT OF INTEREST

We declare that we have no conflict of interest.

BIBLIOGRAPHY

1. Mahony R, Kumar V, Corke P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, *IEEE Robotics and Automation Magazine,* 19(3), 2012, 20-32.
2. Luukkonen T. Modelling and Control of Quadcopter, Independent research project in applied mathematics, *Aalto University School of Science, Finland,* 2011.
3. Apvrille L, Dugelay J L, Ranft B. Indoor autonomous navigation of low-cost MAVs using landmarks and 3D perception, *Proceedings of OCOSS'2013,* 2013.
4. Ashour R K, Islam S, Seneviratne L D, Dias J. Internet based bilateral teleoperation system for quad-rotor flying vehicles, *IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS), Abu Dhabi, UAE,* 2013, 80-81.
5. Troy J J, Erignac C A, Murray P. Haptics-enabled UAV teleoperation using motion capture systems, *Jou of Com and Inf Sci in Eng, Trans ASME,* 9(1), 2009, 1-7.
6. Robert L. Williams II, Wilhelm J. Unmanned rotorcraft used as haptic interfaces, *Ohio University Invention Disclosure,* 2016.
7. Abdullah M, Kim M, Hassan W, Kuroda Y, Jeon S. Haptic drone: An encountered-type kinesthetic haptic interface with controllable force feedback: initial example for 1d haptic feedback, *UIST, Quebec City, QC, Canada,* 2017.
8. Hoppe M, Knierim P, Kosch T, Funk M, Futami L, Schneega S. ß, N. Henze, A. Schmidt, Machulla T. "VR haptic drones: Providing haptics in virtual reality through quadcopters, *MUM '18, Cairo, Egypt,* 2018.
9. Abdullah M, Kim M, Hassan W, Kuroda Y, Jeon S. Haptic drone: An encountered-type kinesthetic haptic interface with controllable force feedback: Example of stiffness and weight rendering, *2018 IEEE Haptics Symposium (HAPTICS), San Francisco, CA,* 2018.
10. Abtahi P, Landry B, Yang J, Pavone M, Follmer S, Landay J A. Beyond the force: Using quadcopters to appropriate objects and the environment for haptics in virtual reality, *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19),* 2019.
11. Tsykunov E and Tsetserukou D. Wired swarm: High resolution haptic feedback provided by a swarm of drones to the user's fingers for VR interaction, *25th ACM Symposium on Virtual Reality Software and Technology (VRST '19),* 2019.
12. Craig J J. Introduction to robotics: Mechanics and control, *Pearson Prentice Hall, Upper Saddle River, NJ,* 3rd Edition, 2005.
13. Tobias. Measuring propeller RPM: Part 3, *Bitcraze,* 2018.
14. Landry B. Planning and Control for Quadrotor Flight through Cluttered Environments, *(39) MIT Department of Electrical Engineering and Computer Science,* 2014.
15. https://github.com/bitcraze/crazyflie-firmware.
16. Jung S, Tomizuka M. Attitude control of a quad-rotor system using an acceleration-based disturbance observer: An empirical approach, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* 2012.

Please cite this article in press as: Robert L. Williams II. Flying robots used as haptic interfaces, *International Journal of Engineering and Robot Technology,* 8(2), 2021, 58-71.