Networked Control of Unmanned Air Vehicles

Daniel Risberg and Peter Henningsson

***Abstract*— It is not just our phones that are getting smarter, it is the buildings as well. Connecting multiple sensors in a house to a main control unit allows this smart building to control things that a human normally would have to. Unmanned Air Vehicles (UAVs) are aircrafts that are capable of piloting without a human on board and is a great addition to the smart building environment. The main goal of this project is to investigate the possibility of integrating UAVs to smart buildings by implementing a PID steering controller and an obstacle avoidance algorithm. In order to make the smart building more autonomous new applications for UAVs are discussed. As a conclusion we are able to show that the UAV used in this project successfully avoided a wall and then managed to navigate through a window in order to reach a reference point located at the other side of the wall.**

***Index Terms*—Automatic control, Avoidance, Detection, PID, Quadrotor, Sensor, Smart building, Unmanned air vehicles**

I. Introduction

UNMANNED Air Vehicles (UAVs) will in a short period of time have a significant role in our modern society. Due to their ability to reach and scan areas human would have a hard time doing on their own, the autonomous UAV is perfect for use in a smart building. This smart building contains many different kinds of sensors which are connected to the main control unit of the house capable of controlling all household electronic devices. With automatic control this can save both time and energy whilst simultaneously providing a better standard of living.

The focus of this project has been to integrate the UAV to the smart building by designing and implementing a steering controller and an obstacle avoidance system. These are two important prerequisites for the use of UAVs in houses to become a reality. A test was derived to check that the steering controller and the obstacle avoidance system worked properly. It was based on letting the UAV fly to a point behind a wall with a corresponding window, thus having to detect and avoid the wall whilst it has to find the window and fly through it to get to the given reference point behind the wall. To pass this test was set to be the main goal of this project.

A quadrotor is an UAV with four rotors controlled individually which leads to a great potential for steering it in tight situations that can emerge in a smart building. Due to this attribute it is an easy choice to use the quadrotor as the UAV that the steering controller and the obstacle avoidance system shall be implemented on.

A study of the quadrotor and its available sensors is made in Section II. A mathematical model of the quadrotor is described in Section III. This quadrotor has been controlled with four different PID controllers and they can be found in Section IV. These controllers were integrated with an obstacle avoidance system which is shown in Section V. How the communication between the control program and the quadrotor works is shown in Section VI. The results of this project can be seen in Section VII. In Section VIII and IX both existing and new applications is discussed. A conclusion is made in Section X and in Section XI some topics for future work is suggested.

SA104X Degree Project in Engineering Physics
Daniel Risberg (dri@kth.se)
Peter Henningsson (phenni@kth.se)

II. Study of the Quadrotor

As said before the quadrotor is an UAV with four symmetrically placed rotors that can be controlled independently. The quadrotor used in this project can be seen in Fig. 1 below. With control of the speed of these four rotors an on board controller is able to produce the roll, pitch and yaw angles which can be seen in Fig. 1. When implementing the steering controller these three angles plus the total throttle of the four rotors combined are used as the input parameters. Before the quadrotor is integrated to the smart building it first has to be equipped with some useful sensors [1]. Some of these sensors can measure parameters such as position, velocity, acceleration and attitude angles while others can provide a virtual image of the surrounding which is vital for obstacle avoidance. In this section the sensors that correlate to the parameters above will be investigated even though not every sensor is available for this project.

Fig. 1. Figure displaying the quadrotor used in this project.

1. Magnetometer

A magnetometer is used to measure the size and direction of the magnetic field at the current position. With this the attitude angles of the UAV can be determined. When calibrating the magnetometer the declination at the current position must be given for the calibration to be accurate [2].

2. Accelerometer

An accelerometer is a device for measuring different accelerations. The acceleration can either be static or dynamic. The static accelerations are due to constant forces acting on an object such as the gravitation field. Dynamic acceleration occurs because of changes in velocity of an object. This device is especially useful for UAVs because by measuring accelerations it is possible to determine in which direction the UAV moves and calculate the force needed to change the movement of it. It can also be used to determine the attitude angles of the UAV in respect the gravitation field [3].

3. Gyroscope

A gyroscope is a frequently used device in aircraft for measuring the pitch, roll and yaw angles based on the principles of angular momentum [3]. This device can of course be implemented into an UAV and is a huge addition to the accelerometer when determining the angles.

4. Global Positioning System

Global positioning system, or GPS as it is called, is a navigation system based on satellites which is used for determining an object's latitude and longitude position. The GPS device can be used for navigation all around the world with high precision. This is great for steering and navigating an UAV outside, but the problem with GPS is that it will not work indoors since the roof and the walls tend to block the signals from the satellites.
Since the goal is to implement UAVs in a smart building it has to relay on other sensors for navigating inside the building. However, the GPS device is a great addition for navigating through big areas outdoors, and can be turn on when it is needed [4].

5. Sonar

Sonar is a sensor that can be used to measure the distance to an object. This is done by sending sound waves and measuring the time it takes for these waves to return to the UAV and then the distance to the object is calculated. The surface of the object has to be smooth for this to work properly and that is why it is mostly used to determine the height of the UAV by sending the sound waves towards the ground. This combined with a GPS would give an accurate position of the UAV as long as it is relatively close to the ground.

6. Camera

For surveillance and monitoring applications a camera may be installed on the UAV. This allows for constant footage of the desired area where the data can be sent to a computer for processing. It is also possible to use the camera for the detection of obstacles which is crucial when implementing the quadrotor in any unknown or dynamic environment. There can also be a recognition part of the camera which allows it to specify what object that has been detected and then acting based on that information. When the UAV is indoors the main purpose of the camera is to navigate through the house based on the recognition of some fixed reference points that have been placed in the smart building. It could also be given the motion patterns of humans and specific animals to be able to tell them apart. With a camera capable of this the number of applications of the UAV drastically increases.

III. MATHEMATICAL MODEL

To fully understand how to create an obstacle avoidance system it is necessary to describe the quadrotor with a mathematical model. This model will develop the differential equations that describe the dynamic system of the Unmanned Air Vehicle and it will also present the restrictions of the quantities in these equations.

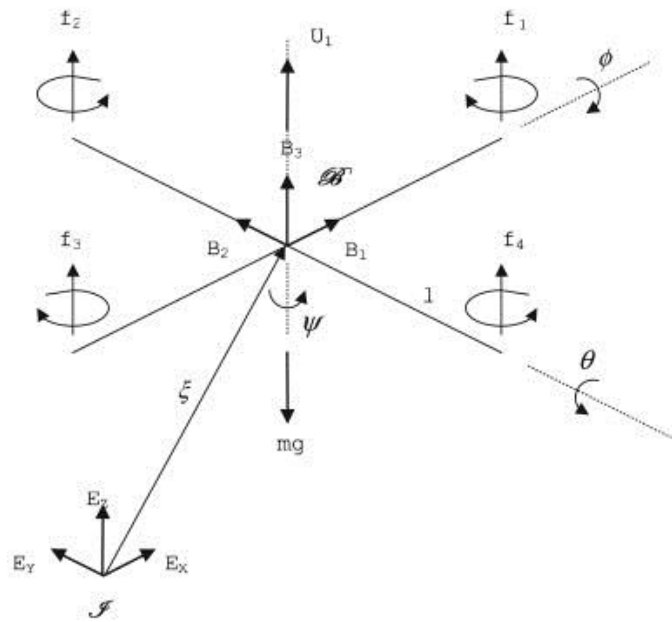

Fig. 2. Figure showing the earth fixed coordinate system *E* and the quadrotor body fixed coordinate system *B* and how they are linked together.

1. Coordinate systems

In order to develop the mathematical model, two coordinate systems must first be defined. The first one is a system fixed to the earth and the second one is a system fixed to the body of the quadrotor where the x and the y axis will correspond to the arms of the quadrotor. This is illustrated in Fig. 2. The angles between the different axes of the two coordinate systems are defined as following: The pitch, θ, is the vertical angle between the two x-axes and the roll, ϕ, is the vertical angle between the two y-axes whilst the yaw, ψ, is defined as the horizontal angle between the two x-axes.
It can easily be seen from Fig. 2 that it is smart to model the quadrotor as a second order differential system with the gravitational force in the negative Z_E-direction and the force generated from the rotors in the positive Z_B-direction. To do this differential equation it is necessary to be able to project

the forces in the Z_B-direction onto the unitary vector n_{ZE}, pointing in the positive Z_E-direction, depending on the quadrotor angles θ, ϕ and ψ. This is done with help from the rotational matrixes Rot_X, Rot_Y and Rot_Z [5] and the unitary vector $n_{ZE.}$

$$Rot_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{1}$$

$$Rot_Y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{2}$$

$$Rot_Z = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$n_{ZE} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{4}$$

To express this unitary vector n_{ZB}, pointing in the Z_B direction, in the fixed coordinate system the unitary vector n_{ZE} is rotated around the Z_E-axis, then the X_E-axis and lastly a rotation around the Y_E-axis and thus getting [4]:

$$n_{ZB} = Rot_Z * Rot_X * Rot_Y * n_Z = \begin{bmatrix} \cos\theta\sin\phi\sin\psi + \sin\theta\cos\psi \\ -\cos\theta\sin\phi\cos\psi + \sin\theta\sin\psi \\ \cos\theta\cos\phi \end{bmatrix} \tag{5}$$

2. Limitations

This model assumes direct control over the parameters θ, ϕ and ψ whilst in reality there are some limitations to these variables. The relationship between the actual value and the desired value is modeled as a first order differential equation:

$$\dot{\theta} = k_\theta\left(\theta_{des} - \theta\right) \tag{6}$$

$$\dot{\phi} = k_\phi\left(\phi_{des} - \phi\right) \tag{7}$$

$$\dot{\psi} = k_\psi\left(\psi_{des} - \psi\right) \tag{8}$$

The fact that the rise time of (6), (7) and (8) must be limited imposes that the rate of change of θ, ϕ and ψ must have an upper and a lower limit. This will lead to an upper and a lower limit to k_ϑ, k_ϕ and k_ψ. There will also be a saturation of the control parameters θ and ψ which implies a maximum and a minimum limit of these values which will hinder the quadrotor from tipping over.

3. Differential equations

Adding the gravitational force for a quadrotor with mass *m* combined with the total force *F* from the rotors to (5) the following relationship is derived:

$$\ddot{x} = \frac{F}{m}\left(\cos\theta\sin\phi\sin\psi + \sin\theta\cos\psi\right) \tag{9}$$

$$\ddot{y} = \frac{F}{m}\left(-\cos\theta\sin\phi\cos\psi + \sin\theta\sin\psi\right) \tag{10}$$

$$\ddot{z} = \frac{F}{m}\left(\cos\theta\cos\phi\right) - g \tag{11}$$

With this finished the step of writing the differential equations (9), (10) and (11) on the form of $\dot{\bar{X}} = A\bar{X} + \bar{B}$ is easily done:

$$\frac{\partial}{\partial t}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{F}{m}(\cos\theta\sin\phi\sin\psi + \sin\theta\cos\psi) \\ \frac{F}{m}(-\cos\theta\sin\phi\cos\psi + \sin\theta\sin\psi) \\ \frac{F}{m}(\cos\theta\cos\phi) - g \end{bmatrix} \tag{12}$$

IV. NAVIGATION CONTROLLER DESIGN

In order to control the quadrotor a steering controller must be implemented. Since this is a project for obstacle avoidance, it is really important that the steering is exact with no large oscillation around the reference point. The choice of controllers will be documented in this section.

1. Choice of controllers

The complete steering controller will be divided into four different sub controllers which are yaw, pitch, roll and height controller. All controllers will be a proportional-integral-derivative (PID) controller on the form [6]:

$$u(t) = K_P e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{d}{dt}e(t) \tag{13}$$

The yaw angle which corresponds to the alignment between the earth fixed coordinate system and the quadrotor body fixed coordinate system is controlled by changing the yaw output.
By examining (12) it is possible to see that by setting the yaw angle to zero the following relationship is derived:

$$\ddot{X}_E = \sin\theta \tag{14}$$

$$\ddot{Y}_E = \sin\phi cos\theta \tag{15}$$

With this it is easily seen that for small θ and ψ the acceleration in the X_E-direction will be directly proportional to the pitch angle and the acceleration in the Y_E-direction will be directly proportional to the roll angle. That is why the input of the pitch controller will be the difference in the X_E-direction between the quadrotor and the reference point and the input of the roll controller will be the difference in the Y_E-direction between the quadrotor and the reference point. The input of the height controller will be the difference in Z_E between the reference point and the quadrotor and its process variable will be the throttle output. Each controller is independent of the others and does only affect the difference from the quadrotor's current position to the reference point in its own direction. The four controllers will be able to work simultaneously as a combined unit in a feedback system shown in Fig. 3.

2. Pitch and roll controller design

Since the quadrotor is symmetrical the pitch and roll controller will be identical. For controlling pitch and roll with high precision it is decided to use a complete PID controller.
It is quite unusual to have an integral part for a basic steering controller [7] because they are usually allowed to have a larger error when following a trajectory but for this project the I part serves a big role for stabilization around the reference point. As mentioned earlier stabilization is prioritized in this project since the goal is to pass through a window with small error margins where no sideways drifting is allowed.
The PID parameters will be tuned in a way that the rise time of the system will become very small and thusly minimizing the total sum of error. This was due to the fact that a big integral part may cause large oscillations.

3. Yaw controller design

The yaw reference angle will always be set to zero in order to keep the quadrotor's coordinate system aligned with the earth fixed coordinate system and therefore allowing the pitch controller to only control movement in the X_E-direction and the roll controller to only control movement in the Y_E-direction. To achieve this, a PI controller will be used with the only purpose of keeping the yaw angle stable at zero degrees throughout the flight [8]. The D part will be excluded to avoid small oscillation problems around the reference point.

4. Height controller design

The height will also be controlled by a complete PID controller. The request for this specific controller is that the rise time will be small and therefore minimizes the error for the integral part. A small overshoot is allowed as long as the quadrotor quickly stabilizes around the reference point. Since a short rise time is requested the P part has to be relatively large.

5. Tuning parameters

The tuning of the PID parameters of each controller will be done by experimental tests. After each tests the results will carefully be studied and with the knowledge of PID controller tuning new parameters for better performance can then be produced until a satisfying result is achieved.

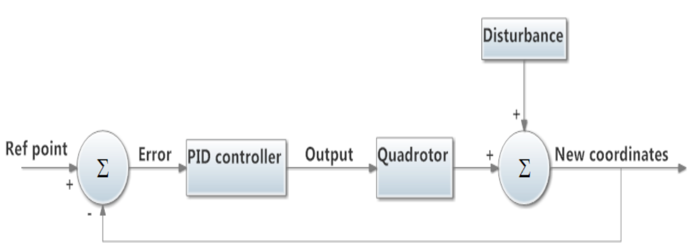

Fig. 3. The figure shows a simple block diagram that represents all four controllers. Some disturbances due to uneven airflows may occur during the flight and is therefore also included in the block diagram.

V. Obstacle avoidance system

To be able to avoid obstacles is crucial for any implementation of the autonomous quadrotor. Without it the areas of usage for the UAV would be drastically decreased. In this section an algorithm to avoid obstacles is described. It is also shown how the implementation of this was done.

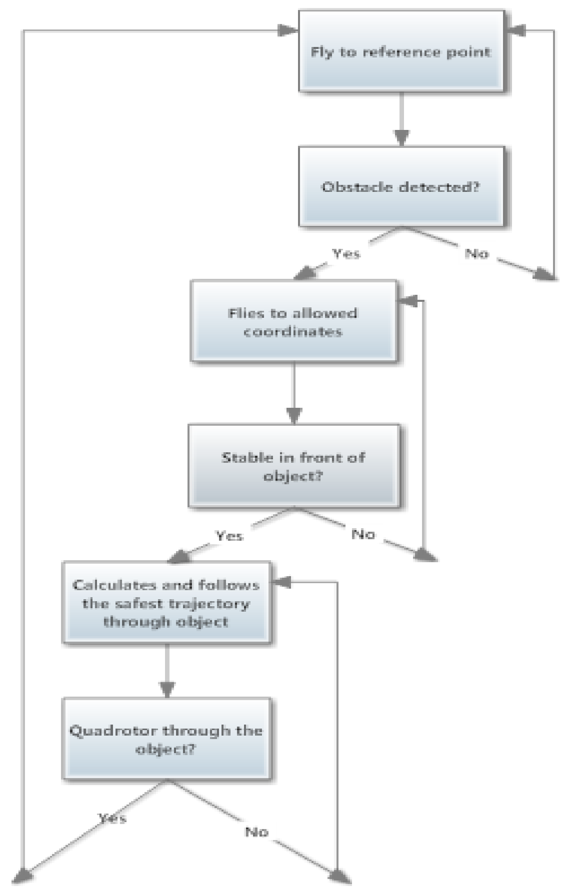

Fig. 4. Algorithm describing how to set reference points to avoid an obstacle.

1. Algorithm

The obstacle avoidance algorithm main idea is to set new reference points around or through the obstacle to get to the wanted coordinates. A flowchart of this algorithm can be seen in Fig. 4 and in Fig. 5 an example of how the algorithm gets the quadrotor through a window is showed. The following paragraph will refer to the points in Fig. 5 when describing this algorithm.
At the start of every flight the quadrotor flies from the starting point *S* toward the desired point *F*. It keeps this path as long as it doesn't detect any obstacles. When an obstacle has been detected at the point *D*, the quadrotor finds the point *W*, where it is safe to pass. It then sets the reference point an orthogonal safety distance, *r*, in front of these coordinates which corresponds to point *A*.

When it has reached these coordinates it has to stay within a margin of error of that point for a specified time, where the margin and the specified time is based on how much clearance there is between the quadrotor and the object. For example if the quadrotor is to pass through a small window, the margin of error will be small and the time will be large to avoid crashes, whilst if the quadrotor is to pass a larger doorway the margin can be greater and the time a bit smaller.
If the quadrotor has achieved this it is classified as stable and starts to calculate and follow the safest trajectory through the object. The safest trajectory is defined as the trajectory where the clearance between the quadrotor and the object is maximized in each of the X_E, Y_E and Z_E-directions. When this trajectory has been followed and the quadrotor is through the object at point *B* the algorithm starts over again by setting the reference point to the coordinates it had in the start of the flight i.e. point *F*.

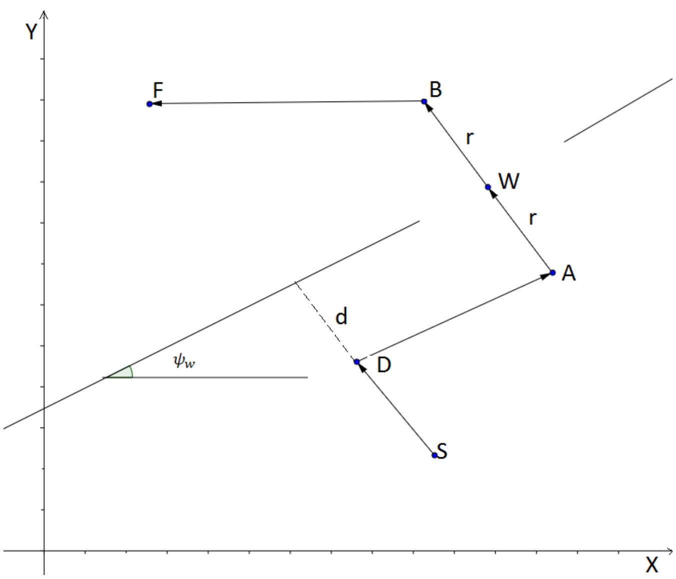

Fig. 5. Describing how to get from *S* to *F* whilst having to detect and fly through the window *W*.

2. Window

From Fig. 5 with a given position. (X_W, Y_W, Z_W), and a certain yaw angle, ψ_W, for the window *W* the coordinates for point *A* and *B* are easily obtained:

$$A = \left(X_W + r\sin\psi_w, Y_w - r\cos\psi_w, Z_w \right) \tag{16}$$

$$B = \left(X_W - r\cos\psi_w, Y_w + r\sin\psi_w, Z_w \right) \tag{17}$$

When calculating the safest way through the window the waypoints are interpolated between point *A* and *B* which will create a smooth and secure flight through this object.

3. Implementation

Since no detection sensors were available for this project both the window and the sensors had to be created with help from the external positioning system. A wall was added beside and in the same yaw angle as the window by interpolating points between the position of the window and the minimum and maximum coordinates covered by camera system. To create an imaginary sensor it was decided that the obstacle was detected when the distance from the quadrotor to the closest point of the obstacle was less than a detection distance *d*.

VI. Implementation Testbed

The communication is a vital part for the use of UAVs in smart buildings to become a reality. This section describes in detail the communication between the controller and quadrotor and how the positioning system works. A short description of the software LabVIEW is also done.

1. LabVIEW

For implementing the controllers a graphical programming software named LabVIEW was used. It is a great software for small practical implementations which this project consists of. LabVIEW also has a built in toolkit for creating and implementing PID controllers which will be a great asset. Furthermore there were already existing programs that handled the communication between the computer and the quadrotor as well as the part of retrieving data from the external positioning system.

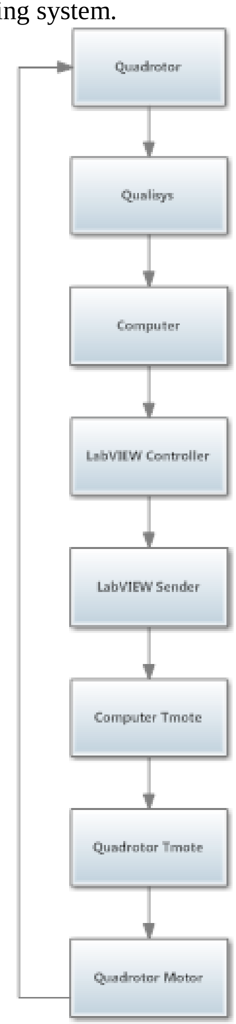

Fig. 6. Block diagram showing how the control loop is implemented.

2. Tmote communication

Since the steering controller in this project is implemented using the programming software LabVIEW the controller will be running on an ordinary computer. Then in order to control the flying quadrotor a wireless communication path between the controller program and quadrotor must be established. In this project the communication problem will be solved by using two Tmotes which are shown in Fig. 7. These motes can send and receive data using radio communication. One Tmote is connected to the computer on which the control program is running whilst the other one is attached and connected to the quadrotor.

However, before the Tmote that is connected to the computer is able to start sending data, a connection between the controller program and the Tmote must also be established. In this case a TCP server will serve this purpose. In order to accomplish this, a TCP server will be opened on the same computer as the control program and the server will connect to the Tmote that is in the USB-port of the computer. The LabVIEW program where the controller is implemented will function as the client.

Once the control program is started, the first action is to establish a connection to the TCP server. When the TCP server has responded positive for the connection request it is possible to start sending data over the TCP protocol. For every iteration in the control program new data packages will be sent and in this particular project the iteration time interval was set to 100 milliseconds. The control program then processes and sends the data to the Tmote which in turn transmits the data package to the receiving Tmote connected to the quadrotor. This is done, as mention above, with radio communication over a protocol named IEEE.802.15.4 [9].

Fig. 7. Figure displaying a Tmote which is used for communication between the controller and the quadrotor.

3. Qualisys positioning system

Since the quadrotor in this project is not equipped with any kind of positioning sensors, the position of the quadrotor must be determined by an external positioning system. All experimental work in this project is done in the KTH Smart Mobility Lab due to the fact that this lab is equipped with a motion capture system from Qualisys. The setup for this system consists of twelve cameras that are mounted in the ceiling of the lab. The positions of the cameras form a square which allows the positioning system to cover a cubical space.

The Qualisys camera system is able to detect and determine the position of reflective markers. These markers can then be attached to any physical object. Once the markers have been attached to an object it is possible to select those markers in the software to define the object as a rigid body. When a body is created the Qualisys software can keep track of it and its coordinates in real time. It is also possible to define several bodies which can be recognized at the same time.

Once that the bodies are defined, the Qualisys software is then able to send the coordinates of each body to the computer on which the controller is running by using the TCP protocol [10] . The Qualisys software will then work as a server for sending body coordinates. The coordinates can then be retrieved by using Qualisys' own client for receiving coordinates. It is also possible to determine the update frequency for which coordinates are obtained. The whole communication path between controller, positioning system and quadrotor can be seen in Fig. 6.

VII. Results

There have been many successful flights through the window placed at different yaw angles but in this section only one of these will be commented. Some of the flights can be seen at the YouTube channel Eyiwin[11]. The flight is similar to the setup shown in Fig. 4.

	Height	Pitch	Roll	Yaw
K_p	6	8	8	1
K_i	5,17	1,67	1,67	0,007
K_d	0,25	0,3	0,3	0

Table 1. Height, Pitch, Roll and Yaw controller PID parameters

1. Demands

In section V, a margin of error, a stabilizing time and a detection distance is discussed. In this test the margin of error of the displacement from the reference point is set to 5 cm. Why the drift can't be larger than this is due to the clearance between the quadrotor and the window. A measurement yields the horizontal clearance to only 7 cm and the vertical clearance to 8 cm. The time the quadrotor had to stay within this margin of error at the reference point in front of the window was set to 5 seconds. This time can be lower but that would lead to an increased risk of crashes. The detection distance, at which the quadrotor finds the obstacle, is set to 1.4 meters. This parameter would be much larger if real detection sensors were used but being able to avoid obstacles with a lower detection distance will increase the performance when real sensors is implemented.

2. Height controller

In Fig. 9 the throttle output, which is a binary scale from zero to 127, is plotted over time. A base throttle is added to the controller output to decrease large oscillations of the height. This base throttle is the value of which the quadrotor hovers a few cm above the ground and in this project it is set to 54 which is seen in the first part of the plot. Due to a powerful quadrotor the throttle output had an upper limit which was set to 70.

Whilst examining the height plot in Fig. 8 a drop in altitude is seen after a few seconds. This is because of the force from the ground almost vanishes completely at that height and the controller has troubles controlling this. The reference height, shown as a horizontal line, is set to the height of the window which is 0.9 meters. It is also observable that the quadrotor overshoots this reference height, with 10 cm and then stabilizes with oscillations smaller than 5 cm which was the demand to fly through the window. When the quadrotor tries to stabilize in front of the window it is sometimes unrecognizable to the camera system and a new controller output will not be added until the position of it can be tracked again. This leads to a drift away from the reference point and thus increasing the time before it can pass through the window. Since the quadrotor managed to fly through the window this height controller is found acceptable and its PID parameters can be found in Table 1.

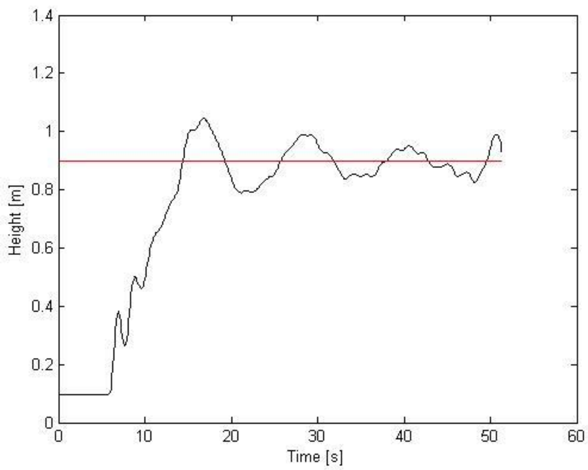

Fig. 8. Step response in Z_E with the horizontal line as the reference point

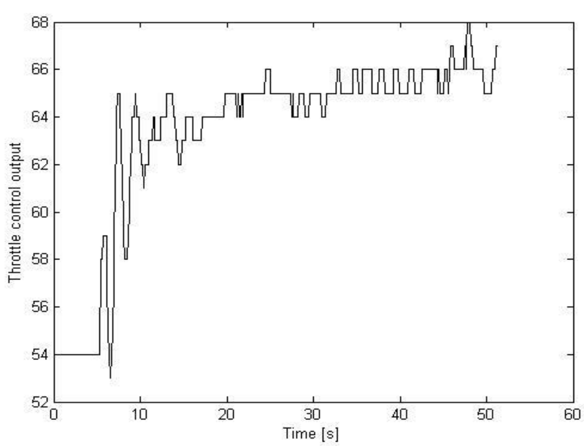

Fig. 9. Throttle control output over time

3. Yaw controller

The scale of the yaw control output in Fig. 11 is in binary and reaches from zero to 127. The basic output, at which the controller output is added, is set to 64 which is the value of which the yaw speed was zero. This can be seen in the first few seconds of the plot.

Since it was decided to control the position X_E with the pitch controller and the position Y_E with the roll controller it is necessary to keep the yaw angle at zero. The yaw angle is displayed in Fig. 10 but it can be seen that the yaw angle never reaches zero. This is because of differences in what the camera system and the control board of the quadrotor defines as zero yaw.

When examining the yaw angle during the takeoff sequence some spikes are seen which are due to disturbances when the quadrotor is close to the ground. The controller compensates for this and stabilizes the yaw angle after only four seconds of flight time. With that in mind the performance of the yaw controller is considered acceptable and its PID parameters can be found in Table 1.

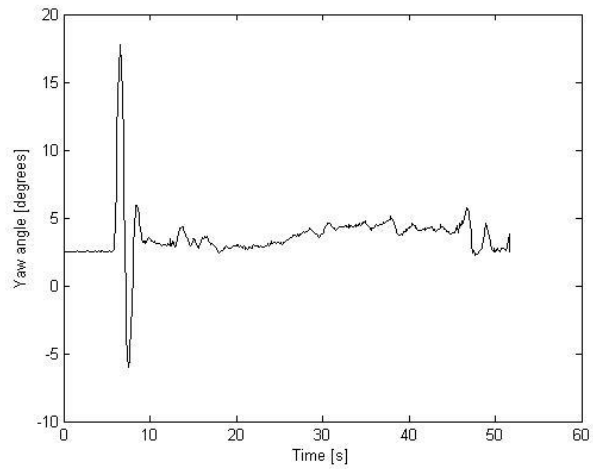

Fig. 10. Step response of the yaw angle ψ with reference point at zero degrees

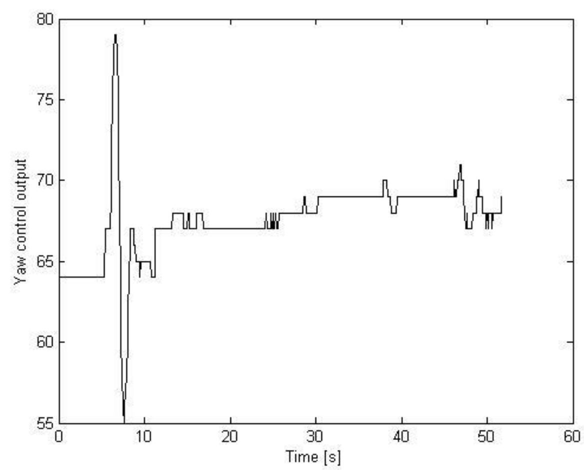

Fig. 11. Yaw control output over time

4. Pitch controller

Since the scale in Fig. 13 reaches from 0 to 127 it could be assumed that the value which gives zero pitch would be the middle value 64 but this is not the case. Due to a displacement in the control board it was found that the pitch controller output that yielded zero pitch angle occurred at the value of 70. The PID control output is added to this base value and is called the pitch control output. Due to safety reasons this output was limited to be within 25 from the base value.

Fig. 12 displays the distance X_E to the reference point over time. As mentioned in section 3 the obstacle avoidance algorithm creates new reference points to avoid obstacles. This is observable when the plot is having large spikes. These spikes are being evened out in a matter of seconds by the controller and then kept stable around the value zero. In the middle of the plot the margin of error is less than 5 cm which was the demand to fly through the window. Since the quadrotor managed to not vary more than 5 cm from the reference point this pitch controller is found acceptable and its PID parameters can be found in Table 1.

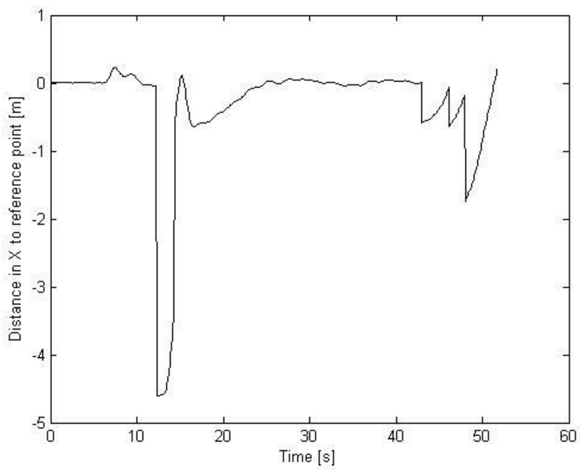

Fig. 12. Step response of the distance in X_E to the reference point. The reference point constantly changes due to the avoidance algorithm but the desired difference, shown in the plot, is always zero.

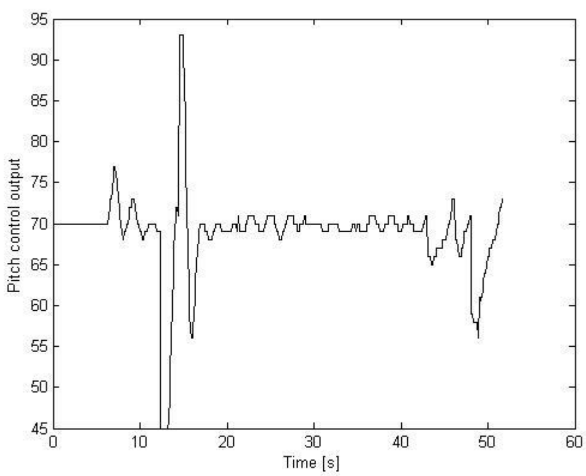

Fig. 13. Pitch control output over time

5. Roll controller

With the same arguments as for the pitch controller, it is seen in Fig. 15 that the base value of the roll output which the roll angle is zero is set to 58. Due to symmetry the roll controller is limited in the same way as the pitch controller is. The distance in Y_E to the reference point in Fig. 14 has the same kind of behavior as the distance X_E to the reference point in Fig. 12. With the same arguments as above it is found out that this roll controller is as equally good as the pitch controller and thusly it is found acceptable. Its PID parameters can be found in Table 1.

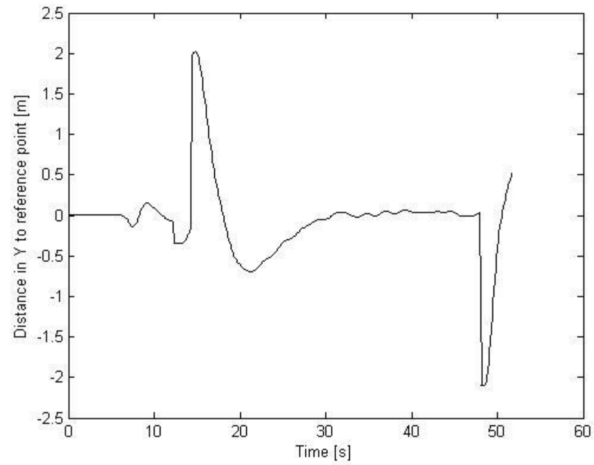

Fig. 14. Step response of the distance in Y_E to the reference point. The reference point constantly changes due to the avoidance algorithm but the desired difference, shown in the plot, is always zero.

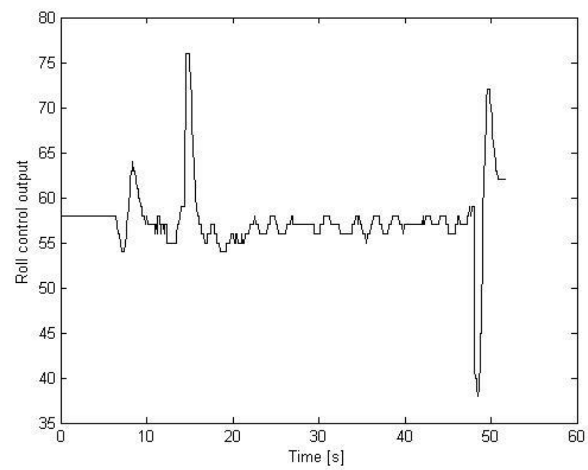

Fig. 15. Roll control output over time

VIII. APPLICATIONS

As UAVs become more and more advanced the applications of them increases. Most of these are for military purposes and a few can be useful to the public. Their small size and their ability to be controlled from a remote location are the two most popular attributes to use in applications. One of the goals for this project was to investigate and propose new application for an UAV. The first three applications shown below are existing state of the art applications and the last three are new applications that have been discussed during this project.

1. Spying

Because UAVs don't need space for humans they can be relatively small in size. Due to their small size it is hard to spot them at larger distances. This makes them perfect for spying where it is crucial not to be seen. UAVs can be equipped with a high resolution camera which is able to take pictures and shoot videos from a far distance. The UAV can then be controlled from a distant computer that is connected to

the UAV through a wireless network. The footages will be sent in real time to the distant computer where it is analyzed and saved as proof. It is also possible to monitor suspects for longer periods of time without him or her even noticing that they are being followed. [12]
Since no humans are on board the UAV there are no risks for human casualties. Because of the wireless network connection which makes it possible to save the evidence directly at a safe location there are no risks of someone destroying it along the way. UAVs with long range communication have also been used to spy on other countries to gather information about military secrets.

2. Attacks

UAVs can be equipped with heavy weapons such as missiles capable of taking out large targets. This is primary for use in warfare where it is a great advantage to strike an enemy without risking any own casualties. These UAVs are also controlled by a distant computer and can be operated from a safe location. [13]
Their small size is a valued characteristic since the UAV might have to travel a large distance to the target which can be very fuel consuming for larger aircrafts. The small size also helps them to avoid being spotted by the enemy's radar when flying behind enemy lines. The size advantage also allows them to make extreme steering maneuvers. Since no humans are on board the UAVs can be able to self-detonate and therefore preventing the enemies from receiving valuable information about the UAV. [14]

3. Surveillance

The UAVs have come to play a big role in the surveillance industry because of their ability to scan and monitor large areas in a short period of time. Since the UAVs can be relatively small they are easy to afford and cheap to operate. As mentioned in the previous section of this report UAVs can be equipped with many different sensors based on the task given [12].
A UAV can be a better solution than static cameras due to the fact that is capable of following tracking the person of interest even if it leaves the covered area.

4. Monitor and defend cattle

The idea of this application is to combine the quadrotor's great ability to survey and monitor large areas with its unique power to intimidate animals. The thought is to let the UAV monitor cattle and other animals existing in a farm and detect if any predator emerges. With an on board camera with motion pattern recognition it will be able to separate predators such as wolves and foxes from cattle and humans.
This will be very useful to farmers that have sheep and chickens because of the many wolf and fox attacks that occur in Sweden. The UAV will patrol in a trajectory around the fold that keeps the sheep in. When a wolf approaches and it is detected by the UAV it will position itself between the sheep and the wolf hopefully intimidating it due to the noise the UAV makes. If the wolf continues to approach the fold the UAV can take drastic measures against the predator and set the reference point somewhere on the wolf where the damages from the rotor blades wouldn't be severe enough to kill it but enough to scare it away.

5. Lawn mower

One application that has been discussed during this project is to have a quadrotor as an autonomous lawn mower. This will basically be an ordinary quadrotor with just one design requirement – the spinning propellers have to be the part of the quadrotor which is closest to the ground. The propellers work as natural cutting blades for grass however they might need to be reinforced and sharpened. A plastic cover can be placed over the quadrotor to protect the surroundings form the dangerous spinning blades.
It will be easy to set the speed of the quadrotor as it travels through the grass. If the grass is thick the propellers will slow down and this will be registered by the controller which automatically lowers the quadrotor's horizontal speed. A tricky thing to control will be the height of the quadrotor which must be equal to the height of the grass. The height controller has to be very stable and extremely insensitive to disturbances.

6. Help in storages

One application for UAVs in general would be as a help in storages. In storage rooms with a high ceiling it is common to stack goods on top of each other which creates huge stacks that are impossible for a human to reach without some sort of ladder.
Since the UAV easily can travel vertically it will be a great assistance for fetching lighter objects that are placed on the top shelf. For grabbing and holding on to an object a device consisting of Velcro can be applied. This device is attached to the bottom of the UAV and to the top of the object.

IX. CONCLUSION

In this project the aim was to integrate the UAV to the smart building by designing and implementing a steering controller and an obstacle avoidance algorithm. This was to be tested by letting the UAV detect a wall and fly through the corresponding window. Even though no real detection sensor was available the task could still be done by using the external positioning system to receive the coordinates of the obstacle. Based on the experience received from trying to complete this test some new and exciting applications of the UAV were suggested.
With both a study and a mathematical model of the quadrotor the task of designing, implementing and evaluating the steering controller became easier. An investigation was made regarding the communication and it was established that Tmote communication was a good and reliable way to transmit the data from the controller to the quadrotor. It was found that the steering controller combined with the obstacle avoidance algorithm was more than sufficient for completing the task of flying through a small window. With real detection sensors and more experiments it is believed that the integration of the UAV to the smart building will become a reality in a close future.

X. FUTURE WORK

In this project no detection sensors were used and an external positioning system had to be used to detect obstacles. A future project could implement the obstacle avoidance system that was developed in this project with an actual detection sensor and test it on the different types of obstacles that can exist in a house. It should also be able to detect and avoid moving objects such as a human or another UAV. The function for planning the safest trajectory to the desired point could also be improved once real detection sensors are available.

ACKNOWLEDGEMENT

We would like to thank our supervisor Meng Guo for his outstanding feedback and guidance throughout this project. We would also like to thank Mani Amoozadeh, Matteo Vanin, Pedro Filipe Russo De Almeida Lima and João Pedro Ferreira Alvito for a great cooperation and invaluable assistance in the KTH Smart Mobility Lab. Finally we would like to thank our examiner Carlo Fischione for providing us with this project and for giving us great feedback.

REFERENCES

[1] B. Hyon Lim, J. Park, D. Lee, and H.J. Kim, "Build Your Own Quadrotor," *Robotics Automation Magazine, IEEE*, vol. 17, no. 3, pp. 36 –37, sept. 2010.

[2] (2013, April) High accuracy Magnetic Sensor and Compassing Solutions. [Online]. Available: http://www.magneticsensors.com/magnetometers-compasses.php

[3] P. Martin and E. Salaun, "The true role of accelerometer feedback in quadrotor control," in *Proc. IEEE Int. Conf. Robotics and Automation*, Anchorage, AK, May 2010, pp. 1623-1629.

[4] R. Mahony, V. Kumar and P. Corke, "Multirotor Aerial Vehicles," *Robotics Automation Magazine, IEEE*,vol. 17, no. 3, pp. 21, 28, sept. 2010

[5] Bertil Westergren, Lennart Råde. (2010). *Mathematics Handbook for Science and Engineering.* p. 110. Lund: Studentlitteratur AB.

[6] Lennart Ljung, Torkel Glad. (2011). *Reglerteknik - Grundläggande teori.* p. 20. Lund: Studentlitteratur AB.

[7] Marzinotto, A. (2012). Cooperative Control of Ground and Aerial Vehicles. p. 82. Stockholm: Royal Institute of Technology.

[8] Ari Hauksson, Fredrik Svensson, Jens Lycke, Kai Imhäuser, Christina Westermark, Joacim Sundberg, Bashar Mengana. (2013). *A Highly Efficient and Autonomous Goods Terminal.* p. 69. Stockholm: Royal Institute of Technology.

[9] Klingenstein, A. (2012). *Multi-agent testbed development, modelling and control of Quadrotor UAVs. pp. 16-17.* Stockholm: Royal Institute of Technology.

[10] Qualisys AB. (2012). QTM Qualisys Track Manager Manual. Gothenburg, Sweden: Qualisys AB

[11] D.Risberg and P.Henningsson, "Quadrotor Obstacle Avoidance", Available: http://www.youtube.com/user/Eyiwin?feature=watch

[12] (2013, April) Aeryon Scout Quadrotor Spies On Bad Guys From Above, [Online], Available: http://spectrum.ieee.org/automaton/robotics/military-robots/aeryon-scout-quadrotor-spies-on-bad-guys-from-above

[13] (2013, April) Gaze War Crimes Drones, [Online], Available: http://www.guardian.co.uk/world/2009/mar/23/gaza-war-crimes-drones

[14] (2013, April) U.S. Drone That Went Down in Iran Was High Tech Intel Tool, Officials say, [Online], Available: http://www.theatlantic.com/international/archive/2011/12/us-drone-that-went-down-in-iran-was-high-tech-intel-tool-officials-say/249562/