Proceedings of the ASME 2017 International Mechanical Engineering Congress and Exposition
IMECE2017
November 3-9, 2017, Tampa, Florida, USA

IMECE2017-71877

PARAMETER IDENTIFICATION AND CLOSED LOOP CONTROL OF A FLYWHEEL MOUNTED HOVERING ROBOT

Akin Tatoglu
Mechanical Engineering
University of Hartford
West Hartford, CT, USA

ABSTRACT

A prototype of a hovering multi-terrain mobile robot platform that makes use of a flywheel for stabilization and heading control for rapid maneuverability was developed and presented in a prior paper. It was shown that flywheel stored energy could be transferred to the overall body to generate rapid angular motion once wheel is instantaneously stopped. Solution improved localization accuracy and reduced the overall sensitivity with respect to external disturbances such as non-flat terrain. In this paper, we present a feedback control system to measure dynamic parameters before and after the wheel is stopped. System is designed to follow a predefined path plan and instantaneous torque change causes oscillation after a waypoint is reached. To address this issue, we updated system with an inertial measurement unit (IMU) as a feedback sensor. Then, we investigate the feedback control of individual forward thrust vectors as well as wheel braking timing to minimize amplitude of transient response oscillation and to reduce the steady-state error to an acceptable level that differential drive fans could compensate this error and correct the heading after the rotation around a waypoint occurs. In addition to that, previous mechanical system could transfer all energy stored at once and was not adjustable. In this research, we also investigate varying amount of angular inertia generated by fans and wheel individually and together. To do so, system is modified with stronger forward thrusters. Prior to running the system with a full dynamic model with real mechanism, we implemented a simulation to empirically extract system parameters and adjust controller gains to follow a predefined path with open and closed loop control schemas with objective of minimizing localization error. Finally system is tested with real mechanism. Governing equations, simulation and empirical results comparison are presented and generated trajectories of various simulation and real world settings are listed. Test results verify that, with a closed loop control system, overshoot and total error about a waypoint can be minimized to an acceptable level at and after transient response phase.

INTRODUCTION

Mobile robots are getting parts of our life increasingly. While most of the co-robots robots [1-2] are designed to traverse on a flat surface –such as a house or library floor, there are other studies for larger scale multi-terrain outdoor and even agriculture robots [3-4]. Autonomous ground vehicles are one of the most studied systems. They are capable of dealing with heavy payload and could be used for various purposes from remote planet observation [5] to transportation systems [6]. While most of the ground vehicles are wheeled, there are other vehicle types for specific requirements such as emergency situations [7-8]. Another common type of autonomous vehicle studied is aerial vehicles. While most of the research focuses on quadcopter like systems [9], there are other systems bio-mimicking the nature [10] and human body [11].

While these vehicles can complete common missions, they have limitations. Ground vehicles are solid systems with enhanced battery life and can reach to a high speed. However, when systems faces with an obstacle –such as a river—, it won't be able to cross the obstacle. A quadcopter is an ideal system that can deal with such obstacle, however it has pretty limited durability. Again, humanoid like systems are designed for structures designed for human beings, but they have limited locomotion skills and suffer from battery life as well.

To overcome these issues, we initiated a new project and had built a hovering robot [12]. The advantage of a hovering body is that it can traverse through nearly any non-porous surface such as sandy and icy terrain as well as water surface [13]. However, heading of such mechanisms are controlled by a differential drive forward thrusters fans which makes them not suitable for agile maneuvers: to change the angular positions of the system, one of the fans is required to be stopped and then rotate on the opposite direction which takes time. Moreover, this type of

rotation requires a wide turn radius since system has no friction other than drag and causes a drift.

To address this challenge, we assembled a flywheel on top of a hovering robot, as shown in Fig. 1. Once the flywheel rotates at high speed, it stores energy. Robot is expected to follow a planned path and when hard turn –such as 45 to 90˚ is required— a braking mechanism stops the wheel instantaneously to transfer the inertia to the whole body. However, this sudden change causes a highly dynamic transient response and overshoot in some cases. This behavior is controlled by a feedback system which uses an IMU as the sensor. On the following sections of the paper, governing equations, initial simulations are discussed followed by hardware and experimental setup. Finally generated trajectories are compared, conclusions and future work are discussed.

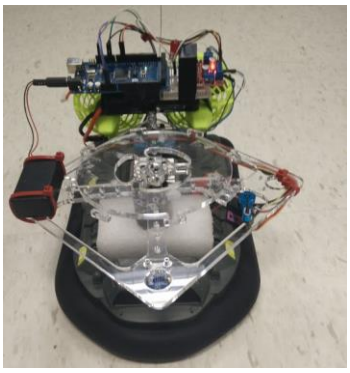

Fig.1 Hovering mobile body and reaction wheel mounted. IMU on the tip.

SYSTEM EQUATIONS AND SIMULATIONS

Overall System, Equation of Motion

System model uses global and body fixed local frames represented by $\{X_G, Y_G\}$ and $\{x_L, y_L\}$ respectively. The velocities on x and y axes are given by

$$\dot{x} = u\cos\psi - v\sin\psi \tag{1}$$

$$\dot{y} = u\sin\psi + v\cos\psi \tag{2}$$

where ψ is projection angle between frames. u (surge speed) and v (sway speed) represent the velocities on x and y directions. The angular velocity of the overall body is given by Ω_H and is equal to first derivative of vehicle orientation ψ given by

$$\dot{\psi} = \Omega_H \tag{3}$$

where subscript {H} represents the hovering body. The controller input u_1 is the sum of forward thruster fan forces which is given by

$$u_1 = F_L + F_R = m\dot{u} - mv\Omega_H + d_v u \tag{4}$$

where m is the mass of the vehicle. This follows the second equation on the sway direction:

$$m\dot{v} + mu\Omega_H + d_v u = 0 \tag{5}$$

where d_v is the coefficient of viscous friction. To simplify the governing equations and simulations modeling, couple assumptions are made. First one is that braking mechanism— presented in Fig. 7— can stop the flywheel instantaneously and second one is that rotation caused by the flywheel occurs around a fixed point. Next one is that the distance of forward thruster force vectors is half of the flywheel disk radius as shown in Fig. 2. Finally, second controller input u_2 is given by

$$u_2 = \frac{r}{2}(F_L - F_R) + M_w\, r = J\,\dot{\Omega_H} + d_r\Omega_H \tag{6}$$

where J is the overall vehicle inertia, M_w is the rotational torque released by the flywheel and d_r is the coefficients of rotational friction.

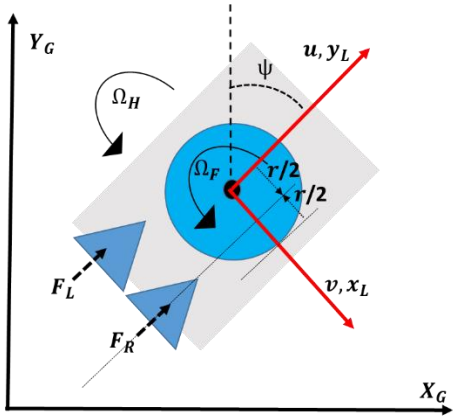

Fig.2 Global and local frame definitions.

Reaction Wheel, Equation of Motion

With θ_b as the angular position of the body, and θ_f as the angular position of the flywheel with respect to the body, equations of motion are given by

$$\ddot{\theta}_b = \frac{(m_b l_b + m_f l_f)*g*\sin(\theta_b) - \tau - f_b\dot{\theta}_b + f_f\dot{\theta}_f}{I_b + m_f {l_f}^2} \tag{7}$$

$$\ddot{\theta}_f = \frac{(I_b + I_f + m_f {l_f}^2)*(\tau - f_f\dot{\theta}_f)}{I_f*(I_b + m_f {l_f}^2)} - \frac{(m_b l_b + m_f l_f)*g*\sin(\theta_b) - f_b\dot{\theta}_b}{I_b + m_f {l_f}^2} \tag{8}$$

where m_b is the mass of the body, including any mass attached, such as the braking system. m_f and I_b represent the mass of the flywheel and moment of inertia of the body around the pivot point respectively. I_f is the moment of inertia of the flywheel around the axis of the motor. The lengths $l_b$and l_f represent the length from the pivot point to the center of mass of the body and the pivot point to the center of mass of the wheel, respectively. τ is the torque of the motor. Dynamic friction coefficients of the

system are represented by f_b – friction where the body is mounted— and f_f , friction of the motor when it is not being torqued.

In the equation of motion for the body, the first term "$(m_b l_b + m_f l_f) * g * \sin(\theta_b)$" represents the torque created by both the body and the wheel due to gravity. They are essentially acting as an inverted pendulum, pulling the body downwards away from the vertical axis. The terms "$f_b \dot{\theta}_b$" and "$f_f \dot{\theta}_f$" are the impact of the friction as a result of the rotational speed of the body and flywheel respectively. The denominator represents the sum inertia of the system with respect to the pivot point, at one corner of the body.

The equation of motion for the flywheel has been simplified so that a term could be created that was very similar to the equation of motion for the body. This shows that the flywheel's acceleration is directly influenced by the motion of the body.

INITIAL SIMULATIONS

Waypoint Following with Flywheel Only Rotation

Prior to developing control system a set of simulations were run which are similar to the real world experiments which will be discussed in the following sections. First simulation was to observe the effect of flywheel itself. A set of waypoints are defined, which are marked with red crosses in Fig. 3.

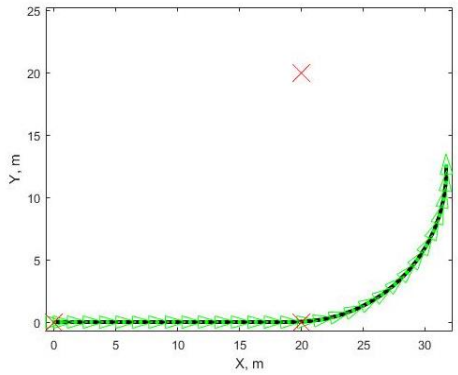

Fig.3 Waypoint following. Rotation is executed with flywheel only.

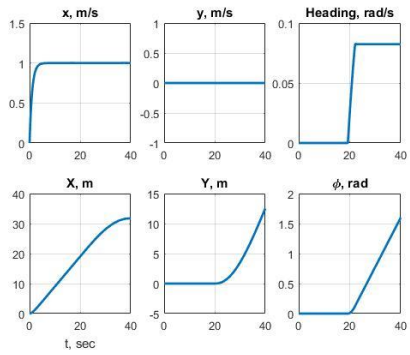

Fig.4 Local and global frames, individual axis linear and angular positions with flywheel only rotation.

Based on the above equations, only flywheel generated torque is calculated and motion of the system is generated which is represented by black line with green arrows superimposed on top representing the heading. Individual displacement plots are given by Fig. 4. While forward thrusters generate constant force, flywheel is stopped at a safe distance to waypoint which is defined by capture radius of 1m. It is shown that flywheel is capable of generating roughly 45˚ angular displacement itself.

Waypoint Following with Flywheel and Feedback Controller

A second simulation which incorporates differential control of left and right forward fans with a feedback controller is executed. This time, once the capture radius is reached, not only flywheel is stopped, but also both fans are controlled individually with the feedback control system. Trajectory generated and individual local and global displacement plots are presented in Fig .5 and 6.

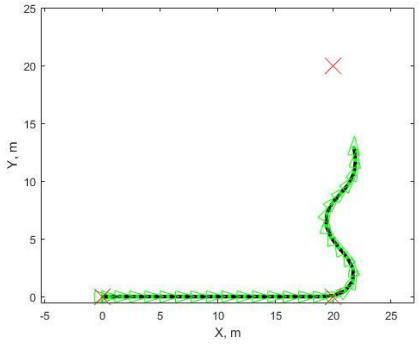

Fig.5 Waypoint following with feedback control system. Rotation is executed with flywheel and forward fans.

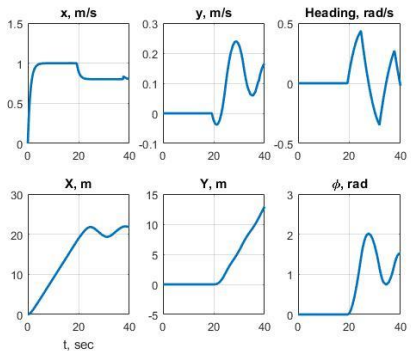

Fig.6 Local and global frames, individual axis linear and angular positions with feedback control system.

In this second experiment, it is observed that forward fans attempt to minimize the error and causes a smooth oscillatory motion during the transient response period.

HARDWARE SETUP

Hardware Overview

Several different components were used for the construction of this system. There were four DC motors. Additionally, two

motor drivers were used to control these motors. An internal measurement unit (IMU) was also included in the system. Two batteries where used to power different aspects of the system. The flywheel was another part of the system, as well as a servo. Finally, the system was controlled by a microcontroller.

Motors

As previously mentioned three brushless DC motors were included in the system. Two of these motors are small. These motors where designed to be used for a quad copter, but have been repurposed to provide thrust for the hovercraft. These motors were fastened to the back of the craft's existing propellers, which are unused. The appeal of these motors is that they have a high max RPM and the speed of them can be controlled with pulse width modulation (PWM). Additionally, there small size is ideal as they can be mounted more easily and reduce the overall weight of the system

One of the remaining two DC motors is the motor behind the rotation of the flywheel. This motor is considerably larger than the two small motors previously mentioned. The reason for this is that the fly wheel needs to spin fast enough so that sufficient fore can be transferred to the hovercraft. This motor is controlled in the same way as the motors used for the propulsion of the system. This motor is placed into a hollow section of the hovercraft and is held in place with a foam block. The block is placed into the hollow and then a circle was cut out to put the motor in.

The last DC motor was built into the initial hovercraft. This motor drove a fan that inflates the skirt under the hovercraft. This is how the hovercraft gets its upward propulsion making it "hover". This motor was not controlled and was tied directly to the power supply to drive it. It did not need to be controlled as full thrust is what was desired, any less and the system's weight may cause it to stay on the ground.

The last motor is a small servo motor. This is mounted onto the flywheel construction. This servo is controlled directly by the microcontroller. The servo has a metal rod attached to its arm. This servo has two positions during operation, open and locked. This servo is responsible for stopping the flywheel. When the servo is open then the flywheel is free to rotate without any issue. When the servo is in the locked position then it stops the flywheels rotation.

Motor Divers

In order to control the motors two motor divers were used. These were the L298N dual h-bridge motor controller modules. As the name suggests there are two h-bridges in the module, allowing it to control two motors at the same time. There are three motors that need to be controlled with these modules; the two smaller motors for propulsion and the motor driving the flywheel. These motors are connected to the two controllers, the flywheel motor on one and the propellers on the other. The drivers are then connected to the microcontroller and a power supply. The motor driver takes three inputs for each motor. Two of them determine the direction of the motor, one high the other low and vise-versa. The last input is the PWM signal. This is what controls the speed and can be a value 0 - 255. These were mounted on the hovercraft in front of the old propellers. This module has two input voltage ports to power it. One is for 12v and up the other is for everything below. If the 12v pin is used an additional 5v output is available which was used to power the microcontroller.

Internal Measurement Unit (IMU)

The IMU used for this system is the BNO055. This module was used to determine the orientation of the system. It was mounted to the front-underside of the flywheel assembly (non-rotating). This was used as a feedback to the microcontroller as to tell it what direction it was heading. The device was communicated with using an I2C connection to the microcontroller. Using this module the microcontroller along with the motor drivers was is able to control the direction the hovercraft will go. This device is powered off of the 3.3v output on the microcontroller.

Power Supply

The system was powered by two batteries. One was a 9v alkaline battery. This battery's output was fed into the motor controller for the flywheel motor. It was connected via the less then 12v input on the controller. It was mounted on the front of the flywheel structure on the opposite side of the IMU. This battery was placed here to help with the weight distribution to try and balance it out as the hovercraft was back heavy.

Another battery that was used was a 11.1v, 2200mAh lipo battery. This battery was used to power the rest of the system. This includes the second motor controller (and the motors connected), the microcontroller, and the fan motor in the hovercraft. As for the motor controller, the battery was connected to the 12v input as the battery could sufficiently power it threw this port despite the lower voltage. This battery, unlike the previous one, could not be mounted to the hovercraft. This was because of its weight. The battery was too heavy and would cause the hovercraft to not generate enough lift to move effectively. To connect it to the system two long wires were used to tie it into the system. This allowed for the battery to be placed to the side during operation of the hovercraft.

Microcontroller

The microcontroller selected for this system was the Arduino Mega. This was selected for its ease of application. The Arduino IDE, which is used to program the microcontroller, has support material that can be referenced to assist with the programming of the system. From a more practical view the Mega has all of the pin-outs needed to run the system. There is an I2C port for the IMU and sufficient pins to run the motor controllers and servo motor.

Flywheel Construction

The flywheel construction comes in two major parts. The first is the frame. This is a non-moving acrylic square that is mounted to the body of the large DC motor and the hovercraft body. This provides a structure to hold wires, the IMU, 9v

battery, and the servo motor. The servo is fastened to this structure so that it can reach the second part of the flywheel construction, the wheel. The wheel is a disk of acrylic attached to the rotational axis of the large DC motor at its center. The disk has 'T' shaped teeth so that the rod on the servo can catch one of the teeth without it slipping. System is presented in Fig. 7.

Fig.7 Braking mechanism acting on tooth barrier

Miscellaneous

There are several steel plates (3) added to the system. These are to add weight in specific locations. Two of these were attached to the flywheel. They were put there as there was not enough momentum being transferred to the hovercraft when the wheel was stopped, so these weights were added. The other one was used to balance the hovercraft. it was found that the hovercraft was slightly front heavy so that when the rear motors were turned on the nose of the hovercraft would hit the ground. Additionally, the hovercraft leaned to the side. Both of these problems were fixed by adding the third steel plate to a specific location at the rear, balancing the craft.

The other Item added to the system was a blue LED. This was attached directly to the microcontroller in the 3.3v and GND pins. What this was used for was tracking the hovercraft in videos to get an exact position for comparison.

CODE SETUP

Software Overview

To code this system the Arduino IDE was used. The code was structured with a setup, loop, and functions. The setup is a block of code that runs only once at the beginning. This setup takes care of the initializations of the different aspects of the system. This includes the pin-outs of the microcontroller and all of the setup for the IMU and servo. The loop section is what contains the code that is actually executed. The functions are various sections of code that can be called in the main loop.

Motor Implementation

The DC motors were implemented in multiple functions. Each of these functions performs a different action. For example: there is a function for moving forward, turn right, turn left, stop and others. These functions take in a value between 0-255 to determine the speed of that action. This applies to both the propellers and the flywheel. It was found that the thrust generated by the two propellers were not equal. This resulted in the hovercraft drifting to one side when both motors are set to the same speed. To rectify this faster motors speed was set to 50 less than the other. After this the hovercraft moved fairly strait, and in an overall predictable manner.

IMU Implementation

The IMU was fastened on the front of the hovercraft so that the x-axis of the IMU is the horizontal plane of the hovercraft. This is what will be read from to determine the orientation of the hovercraft. The code attained the orientation in degrees (0°-360°). The IMU is used to serve as a controller so that if the hovercraft starts to go off course the IMU will detect that and call the appropriate functions that control the rear fans to correct its orientation.

For this IMU to be implemented four libraries needed to be included. Two of these were for the operation of the IMU giving access to various commands that get access to the data collected. Another library is used to manipulate the data gathered by the IMU and convert it to degrees. The last library needed to be added for the use of the I2C communication. To use this method of serial communication an BNO055 object needs to be made with an address for the IMU's memory location for the desired data, which in this case is 55.

Feedback Controller

This P controller was implemented using the data attained from the IMU. The IMU was initialized and found the starting orientation. Then during operation the delta between the initial orientation and the orientation during the hovercraft's movement. If the delta is found to be negative then the left fan will speed up and the right stops and the opposite is true for a positive delta. When the fans are turned on and off like this the hovercraft is trying to stay as close to the initial orientation as it can. This correction doesn't trigger until the delta has exceeded a threshold.

This controller can also be used to make the hovercraft turn. This was done during some of the tests as described in the testing section. To do this the initial value is modified so that the system tries to correct its orientation to match the new "initial" value. For example to turn right the old value, 20°, can be given a modifier of +90° so that the system will want to turn to get to 110° and then continue on the new path.

Servo Motor

The servo is initialized so that the angle is 0 and the pins that control it are set. This servo is controlled using two functions. One is to move the servo into the open position. This position allows for the flywheel to rotate. The other function moves the servo to the locked position. In this position the peg attached to the servo will catch the flywheel causing the wheel to come to an immediate stop thereby transferring the momentum. The speed of servo cannot be increased as the servo only has one speed. The positions are entered as angles and the servo moves to that position.

EXPERIMENTAL SETUP

Overview

Several tests were conducted to observe the functionality of the system and tune the parameters. Each of them had a mutation of the last to observe how that changed the outcome. The tests were done by placing several equal distant lines of markers on the ground and the hovercraft at one end. A top-view camera was placed so that it would look down onto the lines and hovercraft so that it could record its motion from a top down view. This provides its motion on the one plane. To make the motion of the hovercraft easier to follow an led was added as previously described. This led will allow for better tracking and so more accurate trajectory data. Here is a summary of type of experiments that were executed.

Test 1: The first test was to move in a straight line. This was done without the controller to observe the accuracy of the forward movement. This is the base case for forward movement

Test 2: This test was again to move forward but with the use of the controller to correct its orientation. This was done to compare with Test 1 to look for improvement on forward motion.

Test 3: This test is for the flywheel. The flywheel is spun while the hovercraft remains in place. This will show how the flywheel affects the orientation of the hovercraft.

Test 4: This test was to again move forward with the controller. However, this time the wheel will be spinning. This test is to see if the p-controller can negate the rotation caused by spinning the fly wheel, seen in test 3, to maintain a straight path.

Test 5: This is the first test for turning. Here the test was to move forward for a duration then turn using only the rear fans. After the turn the hovercraft moves forward. This will give a baseline to compare to for future turning tests.

Test 6: This test is to turn using only the flywheel. This test is done by moving forward for a duration with the flywheel spinning, then locking the flywheel with the motor. After the turn the hovercraft moves forward. This was done to observe the effect of the flywheel with forward momentum, which can be compared to other turn tests.

Test 7: This test is a combination of the previous two. The flywheel and the rear motors are used to turn after a duration of moving forward. After the turn the hovercraft moves forward. This test was done to compare the combined use of the fans and flywheel to their individual use.

Test 8: This test is the same as Test 7 only after the turn the p-controller is used to better control the turn. This test is to compare how the p-controller affects the exit angle of the turn.

Test 9: This test is the same as Test 8 only this time the p-controller is used during the turn as opposed to after. This test is to see the effects of the p-controller on the exit angle of the turn.

These test results were used to empirically modify the parameters and to run tests which is discussed in the next section.

EXPERIMENTAL RESULTS

Generating the trajectories

A top view camera is set for all experimetns. Hovering robot has a blue led which is used as a moving marker and an image processing technique is used to track initial sample over the period. Markers on the ground are used to measure the real distabce traversed. Fig. 8 illustrates the methodology applied. Blue led location is marked with red dots in Fig. 9-13 to illustrate the trajectory.

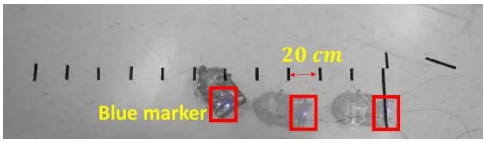

Fig.8 Object Tracking to Generate the Path followed

Analysis

Figures 9-13 represent the outputs of various experiments and a summary is listed in Table 1. While differential drive logic isn't used for first row experiment, flywheel brake isn't used on row 2 experiment. Once flywhell is on, it is stopped at the waypoint. Forward thrustuers are always on, however they are not always controlled with differential drive logic: when they are off, it supplies continuos constant forward thrust vectors. Controller column represents if feedback system kicked in to manipulate the transient response.

Table 1. List of experimental results and explanations

Fig.	Flywheel Brake	Differential Drive	Controller
9	ON	OFF	OFF
10	OFF	ON	OFF
11	ON	ON	OFF
12	ON	ON	ON, at waypoint
13	ON	ON	ON

Figure 9 and Fig. 10 are used to compare the angular displacement capability of individual systems. While flywheel only rotation rotation can generate a smooth and fast rotation, fan only experiment takes more time to rotate and has a negative overshoot. In addition to that, once the real world parameters are empricially identified, real world experiments match with the simulations as could be seen with the comparions of Fig. 3&9 as well as Fig 5&12-13.

Figure 11 represents the maximum rotation capability of system. Not only the flywheel is stopped at the waypoint, but also left and right fans rotate on different directions to generate maximum angular torque. Since there is no controller, system has an increasing negative overshoot naturally while it drifts.

Figure 12 and Fig. 13 are used to compare the effect of controller during the rotation. First trajectory –Fig. 12— starts with a negative overshoot after the rotaion. However, it reaches to steady state point with a smooth oscilation. In the meanwhile, Fig. 13 experiment activates controller from the begining of the process that its trajectory has less oscilation and less overshoot.

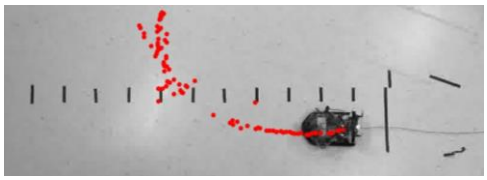

Fig.9 Flywheel only rotation, no controller

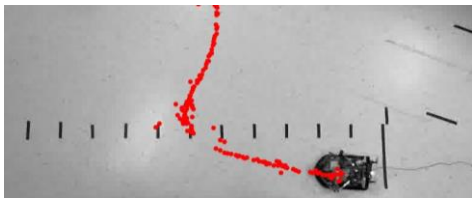

Fig.10 Fan only rotation, no controller

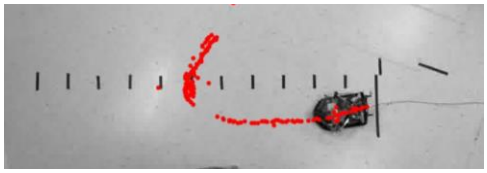

Fig.11 Flywheel and fan operate concurrently, no controller

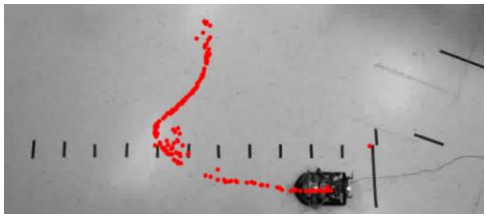

Fig.12 Rotation with flywheel and fans, feedback controller on after rotation

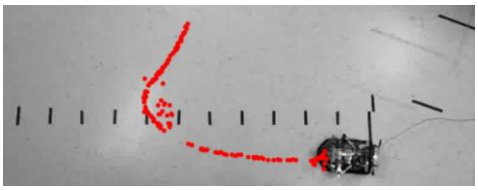

Fig.13 Rotation with flywheel and fans, feedback controller on continuously

CONCLUSION AND FUTURE WORK

Various vehicle types are used and controlled autonomously for various missions. While ground vehicles, humanoids and aerial vehicles have superior abilities, they have limitations such as short durability caused by battery life and/or limited locomotion capabilities to pass through obstacles and rough unstructured.

To address these limitations, we have investigated different vehicle types and developed an alternate mechanism: a hovering robot with a flywheel mounted on top of it. Hovercrafts suffer from wide rotation radius. The goal of the flywheel is to maximize the maneuverability of it while minimizing the turn radius. While the add-on structure—flywheel— can generate agile angular displacement, it is challenging to control the system. We analyzed the system with simulations and empirically tuned the variables. Finally a feedback control system is generated utilizing an IMU as a feedback sensor. Results show that generated trajectory with an active feedback controller of a differential derive mechanism and flywheel structure act faster than traditional solution and can be controlled with acceptable level of oscillation at the transient response phase. In the following steps of our research, we will run experiments to avoid random obstacle in a larger area autonomously with advanced visual perception systems.

ACKNOWLEDGMENTS

We would like to thank Mr. Eric Jacobson and Department of Mechanical Engineering, University of Hartford for their support on this project.

REFERENCES

[1] Chou, Huei-Lin, and Wen-Hsiang Tsai. "A new approach to robot location by house corners." Pattern Recognition 19.6 (1986): 439-451.

[2] Suthakorn, Jackrit, et al. "A robotic library system for an off-site shelving facility." Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 4. IEEE, 2002.

[3] Montemerlo, Michael, et al. "Junior: The stanford entry in the urban challenge." Journal of field Robotics 25.9 (2008): 569-597.

[4] Ruckelshausen, Arno, et al. "BoniRob–an autonomous field robot platform for individual plant phenotyping." Precision agriculture 9.841 (2009): 1

[5] Volpe, Richard, et al. "Rocky 7: A next generation mars rover prototype." Advanced Robotics 11.4 (1996): 341-358.

[6] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite." Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012.

[7] Eriksen, Charles C., et al. "Seaglider: A long-range autonomous underwater vehicle for oceanographic research." IEEE Journal of oceanic Engineering 26.4 (2001): 424-436.

[8] Crespi, Alessandro, et al. "Swimming and crawling with an amphibious snake robot." Robotics and Automation, 2005. ICRA

2005. Proceedings of the 2005 IEEE International Conference on. IEEE, 2005.
[9] Bipin, Kumar, Vishakh Duggal, and K. Madhava Krishna. "Autonomous navigation of generic monocular quadcopter in natural environment." Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015.
[10] Colorado, Julian, et al. "Biomechanics of smart wings in a bat robot: morphing wings using SMA actuators." Bioinspiration & Biomimetics 7.3 (2012): 036006.
[11] Hirai, Kazuo, et al. "The development of Honda humanoid robot." Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on. Vol. 2. IEEE, 1998.
[12] Tatoglu, Akin, Sean Greenhalge, and Kevin Windheuser. "Utilizing Reaction Wheels to Increase Maneuverability and Localization Accuracy of a Hovering Robot." ASME 2016 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers, 2016.
[13] Sira-Ramírez, Hebertt, and Carlos Aguilar Ibáñez. "On the control of the hovercraft system." Dynamics and Control 10.2 (2000): 151-163.