49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida

AIAA 2011-78

Cooperative Control of Swarms of Unmanned Aerial Vehicles

Erik de Vries[*]

Delft University of Technology, Netherlands, Email:e.devries-lr@student.tudelft.nl

Kamesh Subbarao[†]

The University of Texas at Arlington, Arlington, TX - 76019, USA

I. Abstract

Potential function based swarm control is a technique using artificial potential functions to generate steering commands resulting in swarming behavior. This means that the vehicles in the swarm autonomously take care of flying in formation, resulting in steering the swarm, instead of all the individual vehicles being tasked separately. This form of cooperative control is very effective for large groups of unmanned aerial vehicles (UAVs). To test this technique, a simulation tool (in SIMULINK) has been developed using a swarm of quadrotors with nonlinear dynamics. Quadrotors are suitable testbeds due to their mechanical simplicity and can be described by reasonably simple dynamical equations of motion. The inner loop control laws for the quadrotor are based on a backstepping like technique with a nested multi-loop structure. After verification of the individual vehicle model, a multi-UAV controller was designed using potential functions to generate the swarming behavior. The results show that the potential function based method is effective in swarm aggregation. The swarm controller is able to cope with inflight changes of the number of agents if, due to communication limits, one or more agents drift in or out of range. The ability to avoid obstacles by adding obstacle avoidance terms to the steering commands is also shown. These terms are sensitive to the velocities of the swarm members and the gains. However, much better results are expected when a path shaping module is added. This will not only add the ability to fly along a path, but it will also solve the problem of saturated velocity commands due to large differences between the quadrotor position and the goal position.

II. Introduction

This paper uses potential functions to create stable swarm behavior with the members (agents) modeled as quadrotors based on nonlinear dynamics using a backstepping based nested multi-loop controller (Ref. [1]). By explicitly modeling the agents, instead of using simple point mass dynamics, this paper shows that the application of potential functions to create swarm aggregation is a practical solution for multiple quadrotor UAV control. The quadrotor platform is an under-actuated platform, using four rotors as (force) inputs to control the six (coordinate) output states. Research into the quadrotor control problem has led to many potential solutions, several of which have been implemented successfully on hardware testbeds. PID control is used by, for example Refs. [2,3]. For the same problem, a feedback linearization approach was used by Refs. [4] and [5]. In addition, sliding mode controllers ([6]) and H_∞ controllers ([7, 8]) have been used to stabilize and control quadrotor platforms. Finally, another popular nonlinear controller technique, backstepping procedure has been used by Refs. [4, 6, 9] and forms the motivation for our present approach. It is to be said that the previously mentioned control approaches deal with several aspects of robust control design such as disturbance mitigation and improved stability. Ref. [10] and [11] studied the reference trajectory problem using dynamic inversion and neural networks for providing feed-forward compensation of disturbances. In

[*] Graduate Student, Faculty of Aerospace Engineering

[†] Associate Professor, Department of Mechanical & Aerospace Engineering, Senior Member AIAA, Email:subbarao@uta.edu

this paper we wish to highlight a natural hierarchy (order) in the dynamic states that dictate which state variables need to be controlled ahead of the others. In a typical multi-loop structure, this translates to identifying the fast and slow loops. Also, contrary to traditional multi-loop designs for aircraft, the quadrotor control presents a strongly coupled set of rotational and translational dynamics (the forward speed can only be changed via the pitch angle).

This paper uses the above-mentioned quadrotor equations of motion with local controllers developed based linearized dynamics about the hover equilibrium condition. Using the linear equations of motion, a backstepping based nested multi-loop controller is designed, applicable to both the linear and nonlinear quad rotor model.

A. Centralized control versus decentralized control

There are a number of methods available for formation flight control. In the context of a (small) swarm there are two main categories. Centralized control and decentralized control. Centralized control is a control method where each swarm member is controlled by a central controller. The members are fully dependent on the control inputs from the central controller. For a formation this means that each member is guided along its own trajectory, provided by the central controller. In decentralized control, the individual members have their own controller. This allows a member of a swarm to be autonomous. Instead of following a trajectory planned by a central controller, the members now react to each others movements to stay out of harms way and safely reach their goal.

Usually a mix of centralized and decentralized control is applied. This means that control of the vehicles is spread between a central controller and a local controller. For example, consider a simple 'A to B' mission with four UAV's. The mission is to fly the four UAV's from a point A to a point B while staying inside a certain region with radius R and keep safe distance from each other as specified by a sphere Ω_r for each UAV. These spheres may not overlap in order to keep a safe distance. This mission is depicted in Fig. 1. In a mix of centralized and decentralized control, a central controller could provide the swarm with the start point A and end point B, while the individual controllers would fulfill the formation and safe distance requirements. A set-up like this minimizes the information the central controller has to provide and it is a more safe and robust means of control. A loss of communication, does not have immediate disastrous consequences.

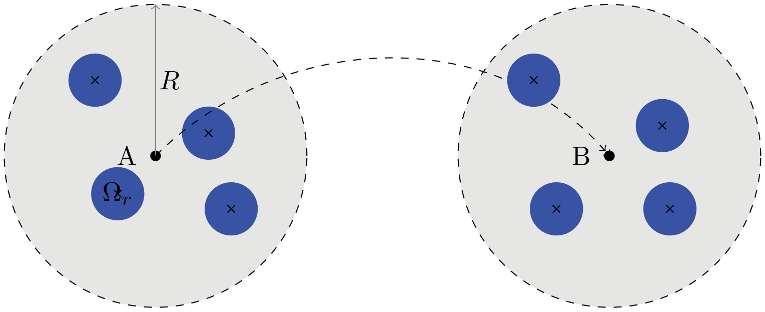


Figure 1: 'A' to 'B' mission

B. Swarm behavior in nature

Swarm behavior is often observed in nature: flocks of birds, schools of fish and herds of animals are well known examples. Swarm behavior often helps animals in defending them against predators and finding food. Their gracious motion seems coordinated but research has shown that simple rules obeyed by each individual are the driving force behind the swarm behavior. The idea of using local laws to simulate swarm behavior was made famous by Ref. [12]. Reynolds came up with three rules to mimic swarm aggregation:

1. Collision avoidance
2. Velocity matching

3. Flock centering

[12] used simulations based on the above for animation purposes, but the same method can be applied to create (local) control laws to ensure automated formation flight in UAVs, micro-UAVs or other vehicles. [13] set up rules for collision avoidance, velocity matching and flock centering and added flock joining and seek maneuver rules. [14] applies potential functions to create local laws for repulsion and attraction in order to create swarming behavior in robots. [15–17] proved stable swarm behavior can be attained using potential functions. Most of the research has been done using simple point mass dynamics modeling of the swarm members. In this paper we use the full nonlinear dynamics of the quadrotor to represent the swarm members.

III. Potential function based control laws

Suppose N is the number of members in a swarm and $\mathbf{r}^i \in \Re^{3\times 1}$ is the position vector of a member, and $\mathbf{r}^T = [r^{1,T}, ..., r^{N,T}]$ is the vector containing all the position vectors of the members, then the steering command can be given by:

$$\dot{\mathbf{r}}^i_{pot} = -\nabla_{\mathbf{r}^i} J(\mathbf{r}), i = 1, \dots, N \tag{1}$$

Here, $J(\mathbf{r})$ is a potential to be chosen by the system designer. For swarm aggregation in quadrotor formations a suitable potential function is of the shape:

$$J(\mathbf{r}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} J_{ij}(\|\mathbf{r}^i - \mathbf{r}^j\|) \tag{2}$$

and fulfills the following assumptions [16, 18]:

1. The potential function is symmetric with respect to the position vector. This means:

$$\nabla_{\mathbf{r}^i} J_{ij}(\|\mathbf{r}^i - \mathbf{r}^j\|) = -\nabla_{\mathbf{r}^j} J_{ij}(\|\mathbf{r}^i - \mathbf{r}^j\|)$$

2. The potential function has an attractive and a repulsive part.

$$J(\mathbf{r}) = J_a(\|\mathbf{r}^i - \mathbf{r}^j\|) + J_r(\|\mathbf{r}^i - \mathbf{r}^j\|)$$

   and its gradients are of the form $\nabla_{\mathbf{y}} J_a(\|\mathbf{y}\|) = \mathbf{y} g_a(\|\mathbf{y}\|)$ and $\nabla_{\mathbf{y}} J_r(\|\mathbf{y}\|) = \mathbf{y} g_r(\|\mathbf{y}\|)$, with $\mathbf{y} = \mathbf{r}^i - \mathbf{r}^j$

3. There exist a unique distance δ_{ij} where the attractive part and the repulsive part balance each other, such that: $g^{ij}_{ar}(\delta_{ij}) = 0$ and $g^{ij}_{ar}(\|\mathbf{y}\|) < 0$ if $\|\mathbf{y}\| < \delta_{ij}$ and $g^{ij}_{ar}(\|\mathbf{y}\|) > 0$ if $\|\mathbf{y}\| > \delta_{ij}$, with $g^{ij}_{ar} = g_a + g_r$

The above assumptions lead to the following theorems [16].

Th1 The center of the swarm, given by $\bar{\mathbf{r}} = \frac{1}{N}\sum \mathbf{r}^i$ is stationary for all time, t

Th2 If $J(\mathbf{r})$ is bounded below, then for any initial condition $\mathbf{r}(0)$ if $t \to \infty, \mathbf{r}(t) \to \Omega$

Th3 The swarm size will be bounded and all the agents will converge to a small region around its center.

The first theorem follows from the symmetric behavior of the vehicles with respect to each other. This is caused by the symmetry of the gradient of the potential function. The second and third theorem lead to a bounded swarm size that will ultimately settle into a finite, small region, in which the vehicles have come to a rest. These theorems are fulfilling the collision avoidance and flock center rule. It can also be shown that the velocity matching rule is fulfilled using a suitable potential function. Assuming that the swarm will need to follow a certain path, the steering command can be extended with a path component:

$$\dot{\mathbf{r}}^i = \dot{\mathbf{r}}^i_{pot} + \dot{\mathbf{r}}_{path} \tag{3}$$

Because the velocity command from the potential function is the negative gradient of $J(\mathbf{r})$, and Th2 shows that $\mathbf{r}_{pot}$ goes to the invariant set Ω, then the only velocity command received by each quadrotor is the path velocity command. Thus over time, all velocities will match the velocity of the path command, satisfying the velocity matching rule.

In this paper the potential function proposed by [15] is used. It is a function consisting of a quadratic attractive part and an exponential repulsive part:

$$J(\mathbf{r}) = \frac{1}{2}a\|\mathbf{r}^i - \mathbf{r}^j\|^2 + \frac{bc}{2}e^{-\frac{\|\mathbf{r}^i - \mathbf{r}^j\|^2}{c}} \tag{4}$$

The potential function steering command is then given by:

$$\dot{\mathbf{r}}^i_{pot} = -\nabla J(\mathbf{r}) = -\sum_{j=1, j\neq i}^{N} (\mathbf{r}^i - \mathbf{r}^j)\left[a - be^{-\frac{\|\mathbf{r}^i - \mathbf{r}^j\|^2}{c}}\right] \tag{5}$$

The equilibrium distance δ_{ij} is found by setting Eq. 5 to zero:

$$\delta_{ij} = \sqrt{c\ln(\frac{b}{a})} \tag{6}$$

IV. Obstacle avoidance

In order to avoid obstacles, an extra term needs to be added to the multi-UAV controller. When an obstacle, or threat, is modeled as a repulsive potential function based on the difference of the position vectors of threat and quadrotor position, each quadrotor will automatically try to avoid the obstacle due to the steering command generated by the added potential term. The obstacle will basically be behaving like a stationary quadrotor with only a repulsive potential function. A threat/obstacle is modeled with:

$$\dot{\mathbf{r}}^{\,i}_{obs} = -\nabla J(\vec{x}) = -\sum_{j=1, j\neq i}^{N} (\mathbf{r}^{\,i} - \mathbf{r}^{j}_{obs})\left[b_{obs}e^{-\frac{\|\mathbf{r}^{\,i} - \mathbf{r}^{j}_{obs}\|^2}{c_{obs}}}\right] \tag{7}$$

where the parameters, b_{obs} and c_{obs} can be used to set the strength and region of influence of the obstacle, the same way as can be done for Eq. 5. Note that the obstacle function is very close to this function. The difference is that the exponential term is now used as repulsive part and there is no attraction.

However, when a quadrotor encounters a threat on, or very close to its path, that threat will generate a large velocity in the opposite direction causing overshoots and possibly a local minimum. Picture for example a simple situation with just two quadrotors, placed so far apart, that they will attract each other. If there is an obstacle between the two quadrotors, this obstacle will be exactly aligned with the two quadrotor's paths. The obstacle avoidance steering command will now generate a command with the size determined by the exponential part and the direction based on the difference vector of the quadrotor and the obstacle. This will result in a vector trying to steer the quadrotor backwards. The net effect is that (in this case) over time both quadrotors will come to a stand still when the attraction of each other matches the repulsion of the obstacle. There is no steering command that generates lateral movement. If the obstacle is close to the path, but not on it, it will have a very small lateral component, insufficient to safely steer the quadrotor around the obstacle. To avoid this kind of behavior, an extra term $\dot{\mathbf{r}}^{\,i}_{on}$ is introduced. This term creates a steering command normal to the vector $\mathbf{r}^{\,i} - \mathbf{r}_{obs}$, causing the quadrotor to take evasive action. The strength of this command varies linearly with the distance to the threat. This ensures that the normal component starts acting before the quadrotor gets dangerously close. This situation is depicted in figure 2. Here, the green and red lines depict the attraction between the two agents, the blue, continuous lines indicate the repulsion force due to the obstacle (black dot) generated by the obstacle avoidance term. The dashed blue line indicates the steering command generated by the normal component. The black vectors are the sum vectors based on the three steering commands just discussed. In the simulation a check is performed whether or not the quadrotor is in close alignment with a threat. If not, there is no need to activate the extra term. The net effect of the two commands will be a smoother avoidance of the threat. This extra avoidance term is implemented as follows:

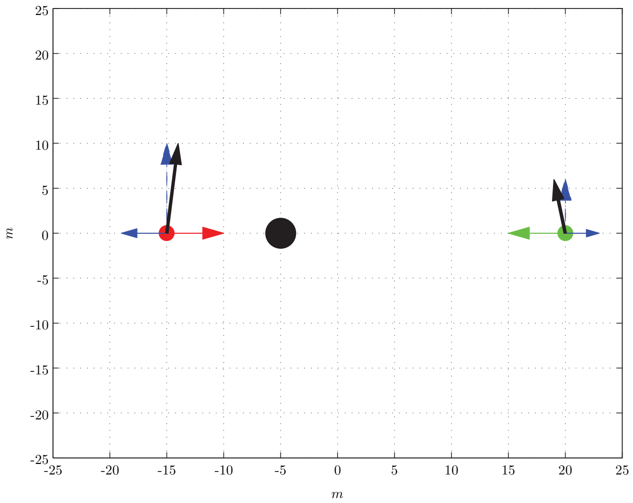


Figure 2: Effect of normal component on total steering command

- First, we check for all obstacles if they are within, or close to the path of the quadrotor by calculating the inner-product ϵ:

$$\epsilon = \frac{\vec{r}^{\,i} - \vec{r}^{\,j}_{obs}}{\|\vec{r}^{\,i} - \vec{r}^{\,j}_{obs}\|} \cdot \frac{\vec{r}^{\,i} - \vec{r}_d}{\|\vec{r}^{\,i} - \vec{r}_d\|}$$

  If $\epsilon > \epsilon_d$, the obstacle is close enough that $\dot{r}_{on}$ needs to be activated. The user can set the sensitivity by varying the value of ϵ_d. A value of of .98 for example corresponds with a cone of 11.5°.

- If multiple obstacles are in close alignment with the quadrotor path, the one closest to the quadrotor is selected.

- Using a linear control law (Eq. 8) a velocity command is generated.

$$\dot{r}^{\,i}_{on} = k_{on} \frac{\vec{r}^{\,i} - \vec{r}^{\,j}_{obs}}{\|\vec{r}^{\,i} - \vec{r}^{\,j}_{obs}\|} \qquad (8)$$

V. Communication constraints

In order to create more realistic behavior, a limit has been placed on the detection range of objects and quadrotors. The steering commands are only based on quadrotors and obstacles that are within range. There is a separate range set for communication between quadrotors and for detection of obstacles. The control algorithm within each vehicle checks the distance between itself and other swarm members. Whenever a member is out of communication range, it will not contribute to the steering command. The same holds for obstacles that are out of range. This is a simplification, as in real life communication will not suddenly be cut off, but the signal strength received by onboard sensors will decay. However, for a first estimate of how the swarm operates under limited sensor range this approach is sufficient. In future work more realistic sensor models will be included. For example field of vision/detection, communication lags and signal quality, strength and decay. It is interesting to note that in animal swarms, the maneuvering is usually based on the nearest neighbors. In case animal behavior needs to be simulated, the algorithm should reflect this filtering [12].

A. Variable gain selection

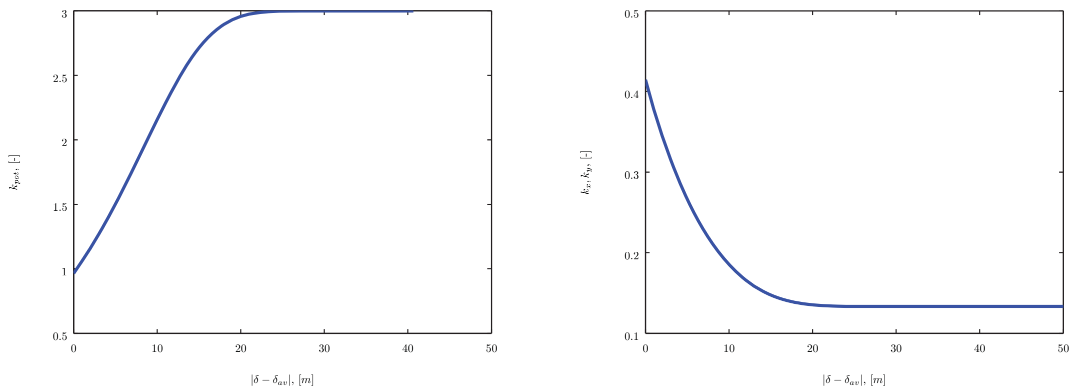
In order to optimize the performance of the swarm, the relative importance of the swarming task and the 'reaching-the-goal' task needs to be made variable. If not, it will only perform one task satisfactory. Consider a swarm of quadrotors that start out at some random configuration. The swarm is not in formation. The preferred behavior would be to first form a formation and then fly, as a fully formed swarm, towards the goal. In order for the swarm to behave like this, the value of the swarming control input $\dot{\vec{r}}^{\,i}_{pot}$ needs to dominate the value of the velocity command generated by the outerloop of the quadrotor controller. See figure 6. This results in a very slow flying swarm once the formation has has been achieved. Therefore, it is necessary to make the position gains and the swarm gains variable based on the state of the formation. This can be done by setting up the following variable swarm gain and position gains:

$$k_{pot} = k_{max} \tanh\left(\frac{1}{k_{max}} e^{\frac{|\delta-\delta_{av}|}{10}}\right) \tag{9}$$

$$k_x = \frac{k_{x_{max}}}{k_{pot}} \tag{10}$$

$$k_y = \frac{k_{y_{max}}}{k_{pot}} \tag{11}$$

The multi-UAV gain is now dependent on the distance between the agents. A larger difference between the desired distance δ and the averaged distance between the swarm members δ_{av} leads to a higher gain being asigned and vice versa. In Eq. 9 the exponential term causes the gain to grow with an increase in $|\delta - \delta_{av}|$ and as before, the tangens hyperbolicus ensures that the gain will not exceed the value assigned to k_{max}. Fig. 3a shows this behavior. A scaled, inverse relationship between the position gains k_x and k_y and k_{pot} (Eqs. 10 and 11) will then ensure the opposite behavior for the position gains. This is shown in Fig. 3b. The values $k_{x_{max}}$ and $k_{x_{max}}$ are the user specified maximum values for the position gains.



(a) k_{pot} versus $|\delta - \delta_{av}|$

(b) k_x and k_y versus $|\delta - \delta_{av}|$

Figure 3: Inverse relationship between position gains and multi-UAV gain with $k_{max} = 3$

VI. Quadrotor model

Fig. 4 shows the quadrotor free body diagram (FBD). Here F_1, F_2, F_3, F_4 indicate the quadrotor's engine thrust and M_1, M_2, M_3, M_4 the rotor torques. Two reference frames are used and are indicated by subscripts b and E indicating body-fixed frame and Earth-fixed frame respectively. Quadrotor mass is indicated by m and g is the Earth's gravitational constant. Then using a standard ZYX-rotation, the following twelve equations govern the motion of the quadrotor [1, 4]:

$$\dot{p} = I_{xx}^{-1}\left[l(-F_1 + F_2 + F_3 - F_4) - qr(I_{zz} - I_{yy})\right] \tag{12a}$$

$$\dot{q} = I_{yy}^{-1}\left[l(-F_1 - F_2 + F_3 + F_4) - pr(I_{xx} - I_{zz})\right] \tag{12b}$$

$$\dot{r} = I_{zz}^{-1}[C(F_1 - F_2 + F_3 - F_4) - pr(I_{yy} - I_{xx})] \tag{12c}$$

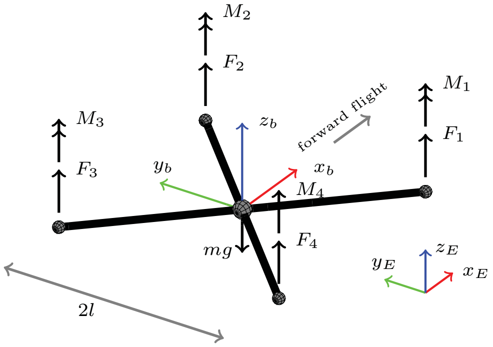


Figure 4: Free Body Diagram of a symmetrical quadrotor

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{12d}$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \tag{12e}$$

$$\dot{\psi} = \frac{\sin\phi}{\cos\theta} q + \frac{\cos\phi}{\cos\theta} r \tag{12f}$$

$$\dot{u} = \sin\theta g + rv - qw \tag{12g}$$

$$\dot{v} = -\cos\theta \sin\phi g + pw - ur \tag{12h}$$

$$\dot{w} = -\cos\theta \cos\phi g + qu - pv + \frac{1}{m}(F_1 + F_2 + F_3 + F_4) \tag{12i}$$

$$\dot{x} = [\cos\psi \cos\theta]u + [\cos\psi \sin\theta \sin\phi - \sin\psi \cos\phi]v + [\cos\psi \sin\theta \cos\phi + \sin\psi \;\; \sin\phi]w \tag{12j}$$

$$\dot{y} = [\sin\psi \cos\theta]u + [\sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi]v + [\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi]w \tag{12k}$$

$$\dot{z} = [-\sin\theta]u + [\cos\theta \sin\phi]v + [\cos\theta \cos\phi]w \tag{12l}$$

Here, x, y, z are the quadrotor position coordinates in the Earth reference frame, ϕ, θ, ψ are standard Euler angles representing the body-fixed frame orientation with respect to the inertial frame and p, q, r the angular rates, also expressed in the body-fixed frame. I_{xx}, I_{yy}, I_{zz} are the moments of inertia of the quadrotor. In Eq. 12c, C is a force-to-moment coefficient, converting the (individual) rotor forces into rotor torques. This torque is needed to control the yaw motion. The rotors are setup in two counter rotating pairs (see Fig. 4) so that at equal rotor speeds the torques cancel each other. By changing the rotor speeds, a net torque can be generated to stabilize and control the yaw motion.

A. Stabilization

The quadrotor controller uses multi-loop nested backstepping to track the position and stabilize the quadrotor with the control laws based on the linearized equations of motion 1. Eqs. 13a - 16b show the control laws as used to stabilize the quadrotor (without swarm aggregation implemented) [1]. To keep demanded angles and velocities within limits the hyperbolic tangent function is used to impose saturation limits. Using trial and error, a velocity of 6 m/s is set as the limit. Note that in this setup, heading information will be lost when both u- and v-components become saturated (the angle between the two will reach 45 degrees). Roll and pitch angles are limited to 0.5 rad (about 29 degrees).

$$u_d = 6\tanh(\frac{1}{6}\left[\dot{x}_d - k_x(x - x_d)\right]) \tag{13a}$$

$$\theta_d = \frac{1}{2}\tanh(\frac{2}{g}\left[\dot{u}_d - k_u(u - u_d)\right]) \tag{13b}$$

$$q_d = \dot{\theta}_d - k_\theta(\theta - \theta_d) \tag{13c}$$

$$\dot{q} = \dot{q}_d - k_q(q - q_d) \tag{13d}$$

Defining $\dot{q} = I_{yy}^{-1}\left[l(-F_1 - F_2 + F_3 + F_4) - pr(I_{xx} - I_{zz})\right]$ gives the x-controller.

$$y_d = 6\tanh(\frac{1}{6}\left[\dot{y}_d - k_y(y - y_d)\right]) \tag{14a}$$

$$\phi_d = \frac{1}{2}\tanh(\frac{-2}{g}\left[\dot{v}_d - k_v(v_d - v_d)\right]) \tag{14b}$$

$$p_d = \dot{\phi}_d - k_\phi(\phi - \phi_d) \tag{14c}$$

$$\dot{p} = \dot{p}_d - k_p(p - p_d) \tag{14d}$$

Defining $\dot{p} = l[-F_1 + F_2 + F_3 - F4]/I_{xx} = M_x/I_{xx}$ gives the y-controller.

$$w_d = 6\tanh(\frac{1}{6}\left[\dot{z}_d - k_z(z - z_d)\right]) \tag{15a}$$

$$\dot{w} = \tanh(\dot{w}_d - k_w(w - w_d)) \tag{15b}$$

Defining $\dot{w} = (F_1 + F_2 + F_3 + F_4)/m = T/m$ gives the z-controller.

$$\psi_d = \dot{r}_d - k_r(r - r_d) \tag{16a}$$

$$\dot{\psi} = \dot{\psi}_d - k_\psi(\psi - \psi_d) \tag{16b}$$

Defining $\dot{\psi} = C[F_1 - F_2 + F_3 - F_4]/I_{zz} = M_z/I_{zz}$ gives the ψ-controller:

VII. Swarming controller implementation

Implementing the steering commands for swarming into the existing quadrotor controller only takes a few alterations to the original controller. Fig. 5 shows the unaltered block diagram for the x-mode. First, the potential based steering command, responsible for the swarming behavior is be added to the signal going into the second loop. This is a velocity command. The new (perceived) velocity command is now a summation of a position based command from the outer loop and a swarm command directly added by the swarm controller. For the x-direction, this gives the following updated, attitude control law:

$$\theta_d = \frac{1}{2}\tanh\left(\frac{2}{g}\left(k_u\left[u - (u_d + r^i_{pot,x})\right]\right)\right) \tag{17}$$

with $r^i_{pot,x}$ being the x-component of $\mathbf{r}^i_{pot}$.

In order to obtain velocity matching, the quadrotor velocity control laws should no longer be based on the individual's position states, but on the position state of the center of the swarm. This means that the outermost stabilizing loop calculates a velocity command based on the swarm center, $\bar{r}$ and the destination position, $\mathbf{r}_d$. This leads to the following updated velocity control laws in x- and $y$directions:

$$u_d = 6\tanh\left(\frac{1}{6}\left(\dot{x}_d - k_x\left[\bar{r}_x - x_d\right]\right)\right) \tag{18}$$

$$v_d = 6\tanh\left(\frac{1}{6}\left(\dot{y}_d - k_y\left[\bar{r}_y - y_d\right]\right)\right) \tag{19}$$

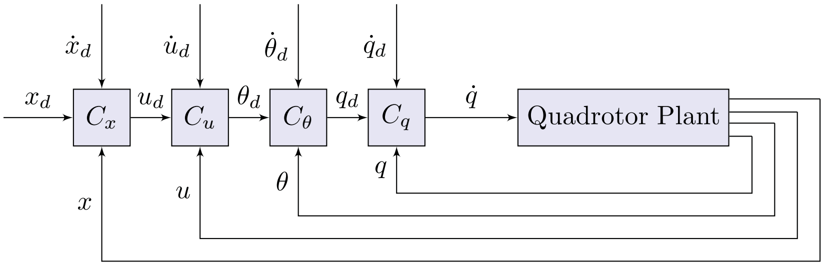


Figure 5: Block diagram of the x-controller and plant, as a pure stabilizer

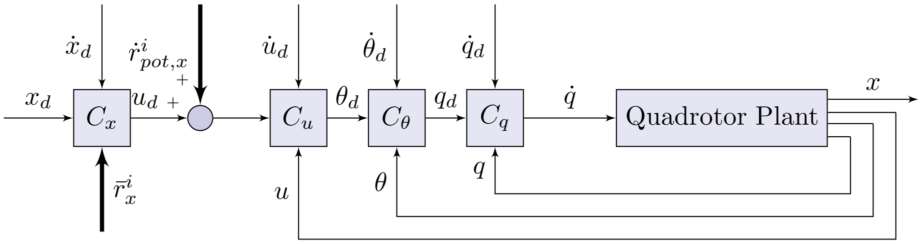


Figure 6: Block diagram of the x-controller and plant, with top level inputs for swarm aggregation and path tracking

where the subscripts x and y indicate the directional components. The velocity in z-direction has been left unaltered on purpose: this way the formation always tries to maintain the predefined altitude. If not, even though the center of the swarm would reach the desired altitude, the individual swarm members might not, resulting in a tilted formation at the destination. When this is not an issue, or when using a large swarm, the w_d control law can be adapted in a similar fashion.

With the above, all quadrotors now base their velocity command on the position of the center of the swarm, leading to the same velocity command for all quadrotors if the swarm has stabilized and $\dot{\mathbf{r}}^i_{pot} \to 0$. The changes in the controller are indicated in Fig. 6. The thick arrows show the new inputs. Note that there is no feedback of the quadrotor position to the outer loop of the stabilizer. The position is only used by the top level controller. Fig. 7 shows how the swarm controller is integrated with the existing quadrotor controller and model, using the x-position control as an example.

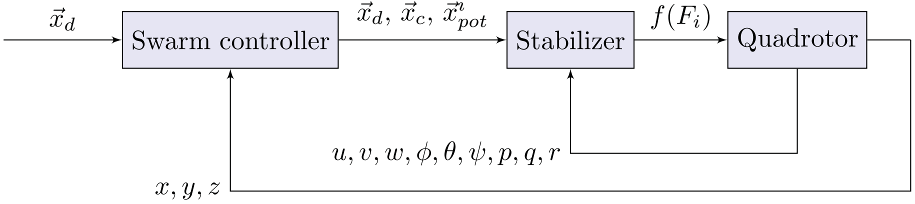

$\vec{x}_d$
Swarm controller
$\vec{x}_d$, $\vec{x}_c$, $\dot{\vec{x}}^i_{pot}$
Stabilizer
$f(F_i)$
Quadrotor
$u, v, w, \phi, \theta, \psi, p, q, r$
x, y, z

Figure 7: Block diagram of the swarm controller and quadrotor stabilizer and plant

VIII. Results

This section will show the results of a number of simulations of a small swarm of three quadrotors, using the controller derived in the previous sections. Unless stated otherwise, the numerical values used in the quadrotor model are based on the quadrotor used by Ref [4]. The first sub-section shows the swarm aggregation term (see Eq. 5). Next, in sub-section B it will be demonstrated that the swarm is able to effectively move from a starting position to a user defined goal position. It also shows the effect of variable position and swarm gains and communication range limits. The next sub-section shows that the model also functions in 3-D. Sub-section D shows the ability of the quadrotor swarm to avoid simple obstacles. The last sub-section shows the effect of a limited sensor range.

A. Case I: Swarm aggregation

The first case demonstrates the swarm aggregating behavior. Feedback gains k_x, k_y and k_z have been set to zero, so the quadrotors move only under the influence of the $\dot{\mathbf{r}}^i_{pot}$-term. Initial altitude is set to zero and the constants for the potential function are set to: $a = 0.03, b = 1, c = 25$. Fig. 8 shows how the quadrotors are being pulled towards each other. A small overshoot is visible, meaning that the inter quadrotor distance is smaller than δ_{ij}. The quadrotors are clearly being pushed away from each other and settle at their equilibrium distance. Calculating the equilibrium distance by using equation 6 and comparing with the distances between the quadrotors after a two minute simulation produces the value of δ_{ij} for each quadrotor. This means the quadrotors form an equilateral triangle as expected. Initial positions are indicated with a star, final positions with a circle. Fig. 10 shows a close up of the position plot. In this figure the position of the swarm center $\bar{r}$ has been plotted. Although not completely stationary as predicted by **Th1** in section III, for all practical purposes, the center can be considered steady. Fig. 9 shows the values of J_{ij}. Note how the overshoot in the value for the third quadrotor in this figure corresponds with the third quadrotor getting too close to the other quadrotors as visible in Fig. 8

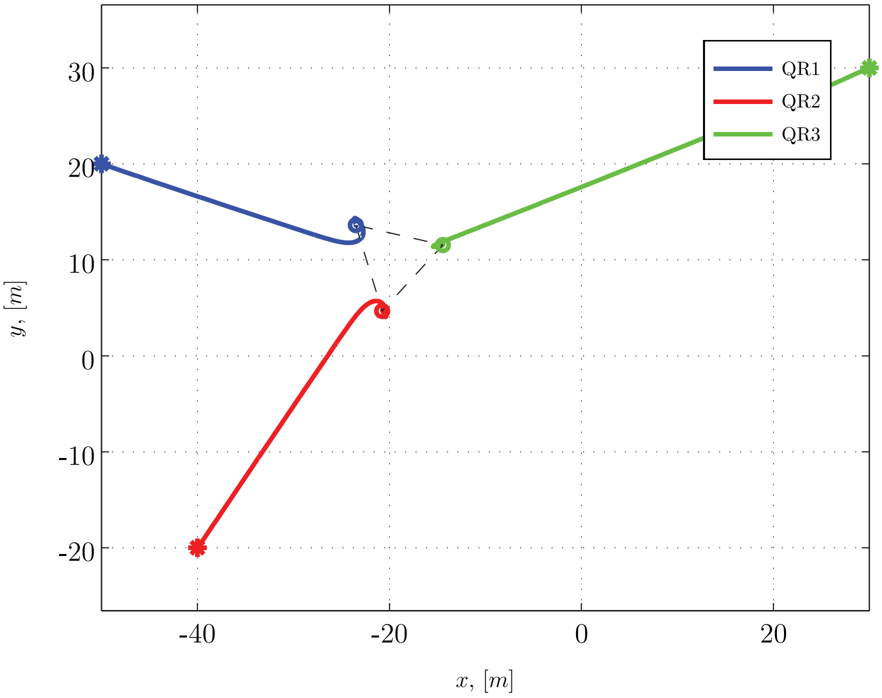


Figure 8: Swarm aggregation under influence of the potential function

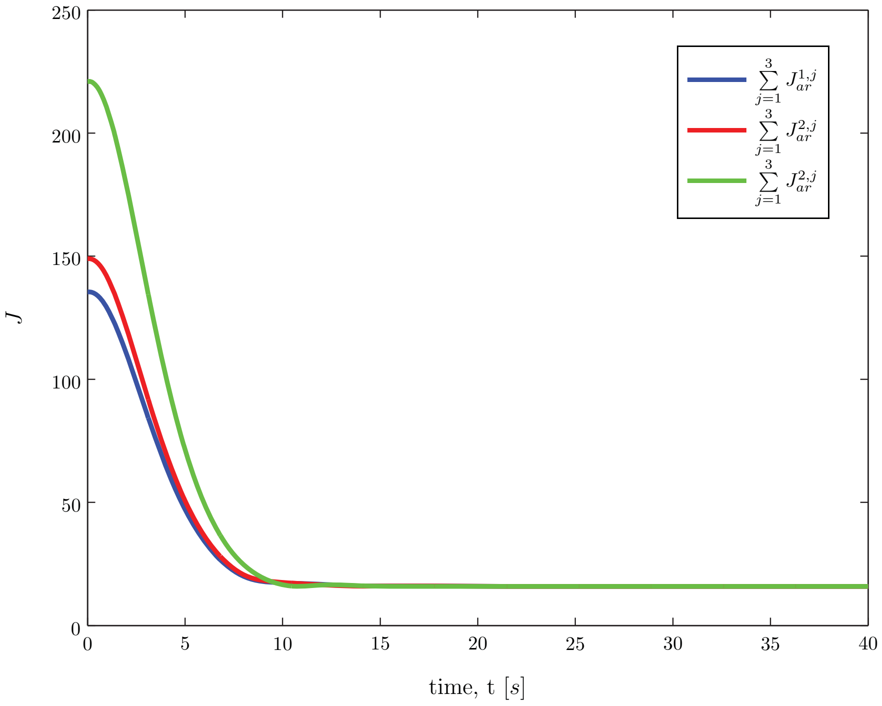


Figure 9: Potential function values corresponding to *case I*

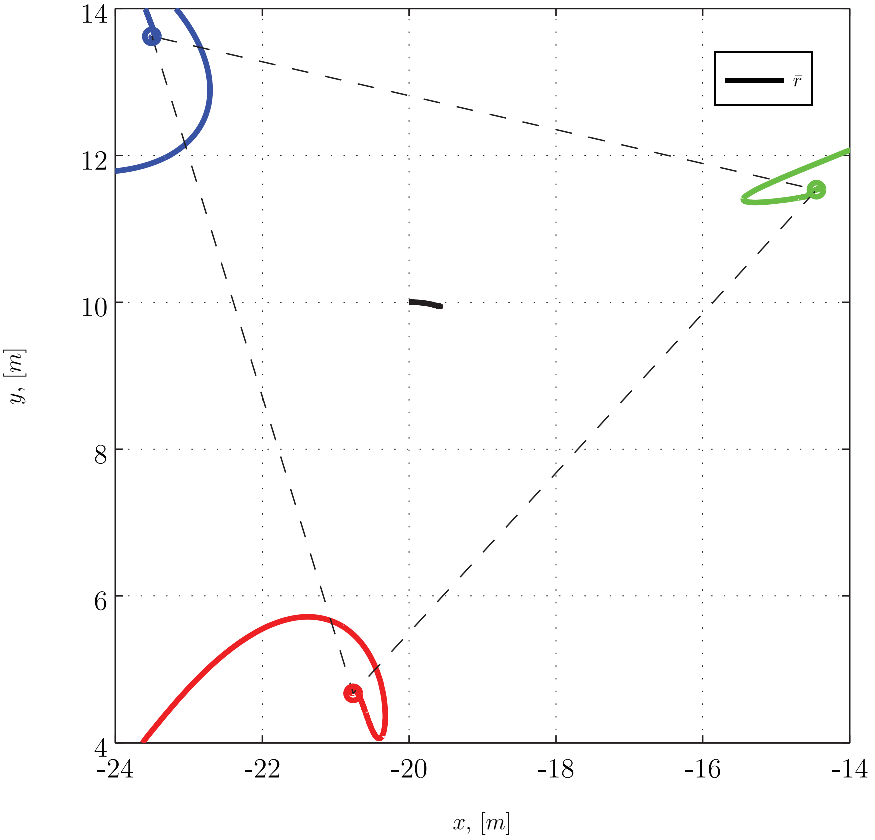


Figure 10: Close up showing steady swarm center

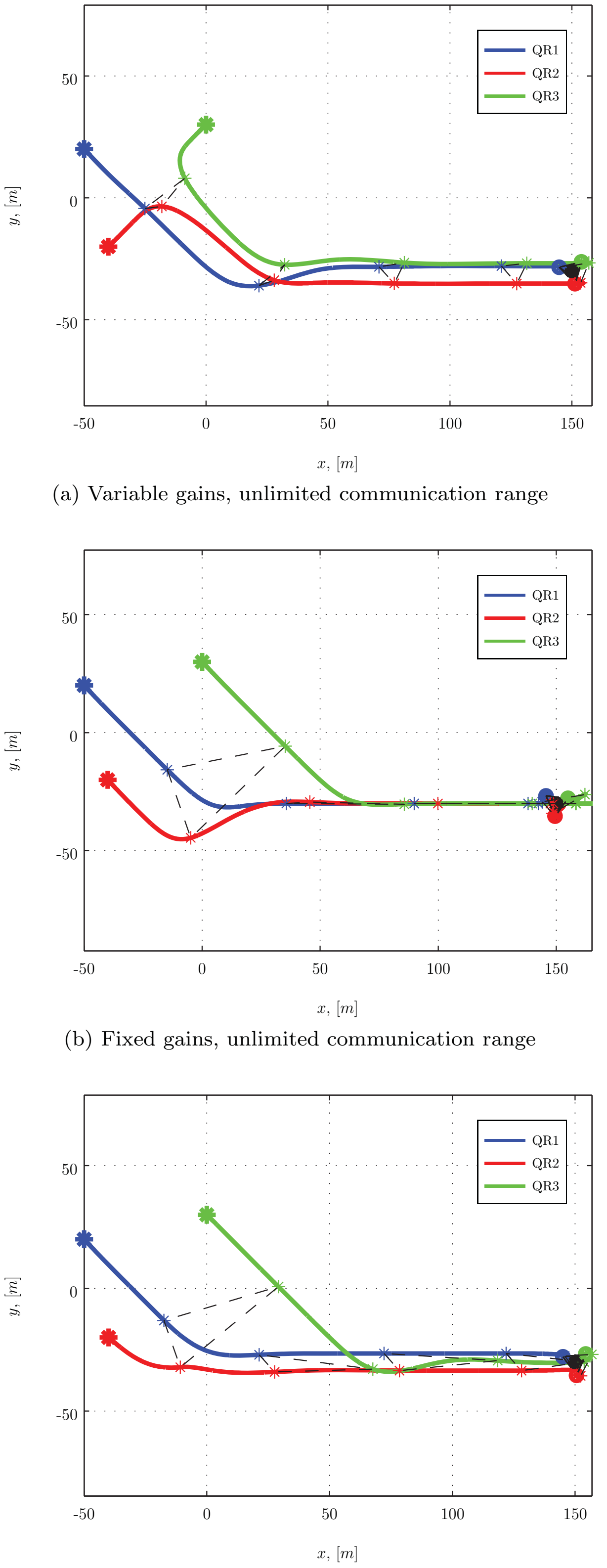
B. Case II: Position control of swarm

This case shows the ability of the controllers to steer the swarm towards a goal. The quadrotors are placed at random positions, far enough that all three quadrotors are free agents, but still under influence of the attractive potential. The desired goal is placed far away from the initial conditions, so that the quadrotors have to move as a swarm, thereby demonstrating velocity matching as well as swarm aggregation. This simulation was run three times, using different setups of the model. In the first case, the fully functioning model is shown. The second time, the variable gain was shut off, demonstrating the influence of this effect. In the last run, the effect of communication range is shown.

The position plots for the three cases are shown in Fig. 11a. The plot shows the best behavior is obtained in the first case. Fig. 11a clearly shows that a fixed gain leads to slow swarm aggregation. In this case k_{pot} has a maximum of 2. Due to the performance limitations of the quadrotor platform, there is only limited control authority. This means that the available control over the craft needs to be shared between the swarm control input, $\dot{\vec{r}}$ and the position control outputs, u_d, v_d and w_d, see Fig. 6. When the gain of $\dot{\vec{r}}$ is increased at the expense of the position gains, the swarm formation is fast, but it results in a very slow flying swarm due to the very low position gains.

The small stars connected by dashed lines indicate the positions of the quadrotors calculated at constant intervals during the simulation. Fig. 11a shows that the quadrotors move along the path in formation as opposed to the other two situations.Fig. 12 shows the potential function values. This figure also shows that influence of the variable gain swarm. The setup with variable gain swarm is able to minimize the potential function value faster than the setup with fixed gains.

The effect of limiting the communication range is clearly visible in Fig. 12c. If a quadrotor is out of communication range, the potential function value reduces to: $J_{ij} = \frac{bc}{2}$. In this case $b = 1$ and $c = 25$ resulting in a constant value of 12.5. When the other quadrotors are within range, the potential function value increases and the steering command brings it down again to its new minimum value. See also section E. Fig. 13 shows the velocity profile for the three cases. The differences between the three situations are due to the different gains applied. Fig. 13b shows that flying with fixed gains does not result in smooth behavior when the goal is reached. This can be resolved by flying slower. This further underscores the importance of variable gains. Notice that due to the limit placed on the outer-loops, the velocities are saturated. In the position plot this is visible as the part where the quadrotors fly diagonal at an angle of 45 degrees: the vector sum of the velocities. This of course is unwanted behavior. In section B a recommendation is made to avoid this.

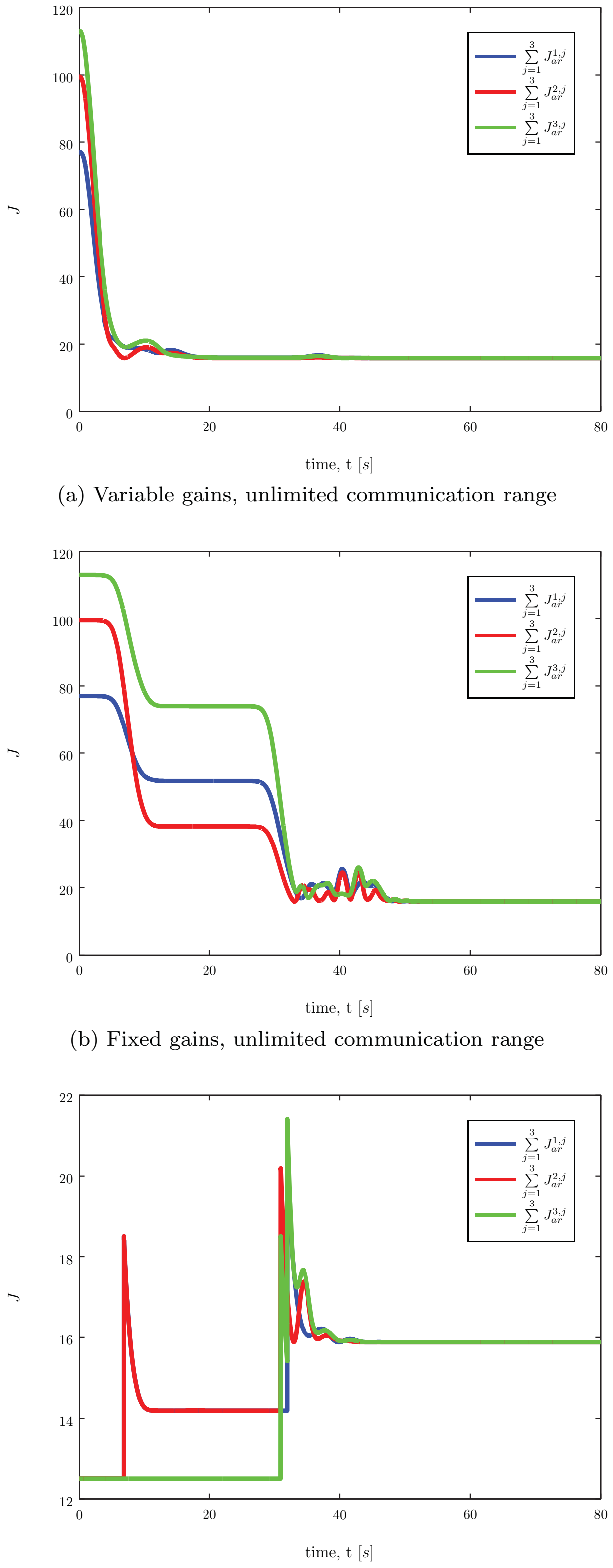


(a) Variable gains, unlimited communication range

(b) Fixed gains, unlimited communication range

(c) Variable gains, limited communication range, $d_{com} = 20m$

Figure 11: Position control for Case II simulations

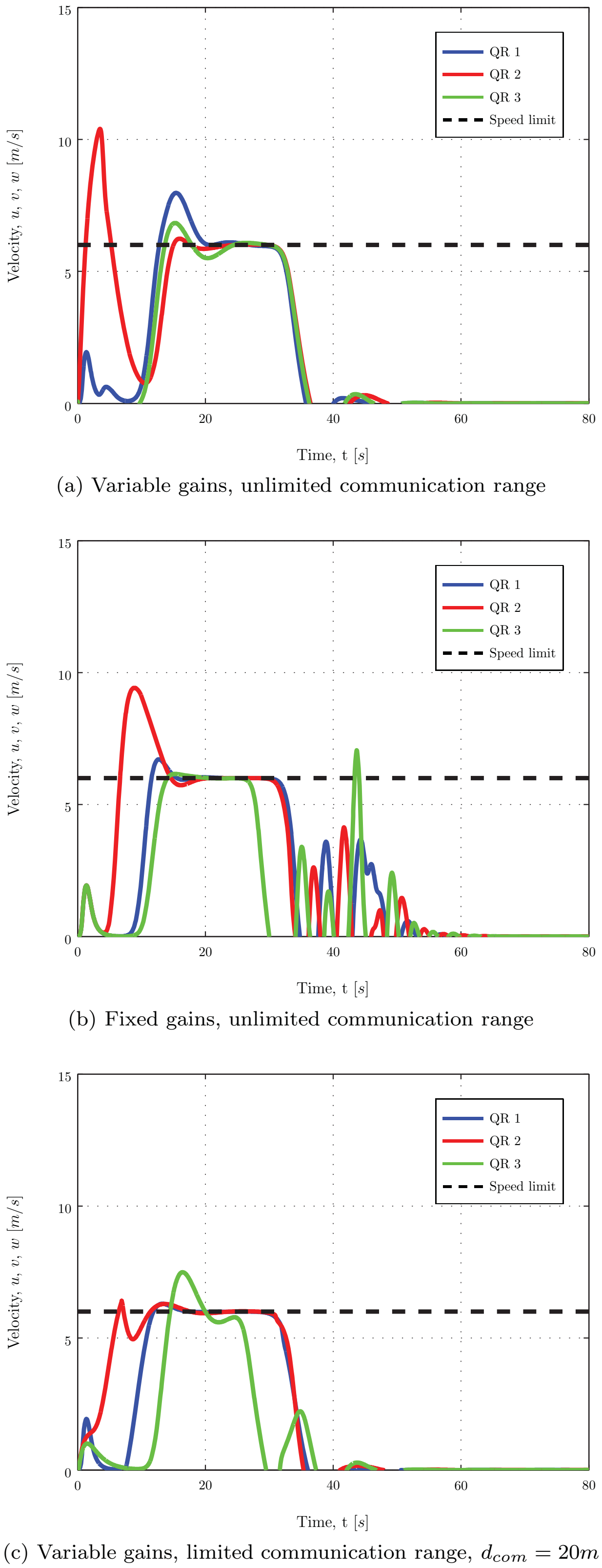


(a) Variable gains, unlimited communication range

(b) Fixed gains, unlimited communication range

(c) Variable gains, limited communication range, $d_{com} = 20m$

Figure 12: Potential function values for Case II simulations



(a) Variable gains, unlimited communication range

(b) Fixed gains, unlimited communication range

(c) Variable gains, limited communication range, $d_{com} = 20m$

Figure 13: Potential function values for Case II simulations

C. Case III: Swarm movement in 3D

This case demonstrates that the swarm is able to move and accurately perform in three dimensions. Fig. 14 shows a 3D plot of the quadrotor positions. Initial positions are equal to that of the previous case, except that the quadrotors now start from different initial heights. The goal is located at $[150, -30, 50]$. The quadrotor displays some peculiar behavior in reaching its altitude. Instead of flying in a straight line, the quadrotors climb towards their altitude before reaching the goal, flying the last part at the desired altitude. This is due to the fact that the control laws for the z-position only consist of two nested loops. The altitude is thus a faster responding state than the x and y states. The thin lines are a 2D projection on the $z = 0$ plane. Fig. 15 shows the velocity profile for the 3D simulation. The figure shows that when the quadrotors reach the desired altitude, they have a little overshoot. This can be avoided if need be, by lowering the value of k_z.

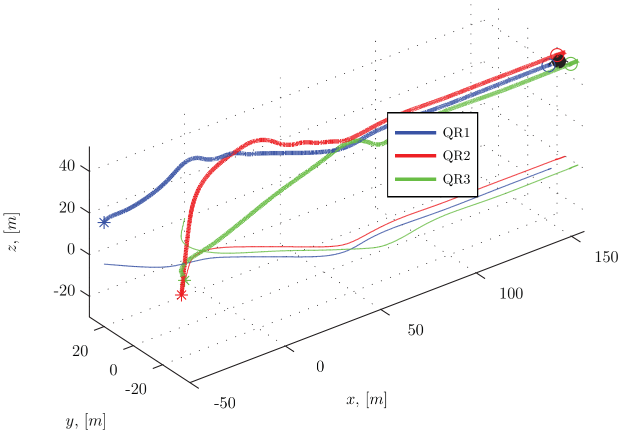


Figure 14: Position control for Case III simulation

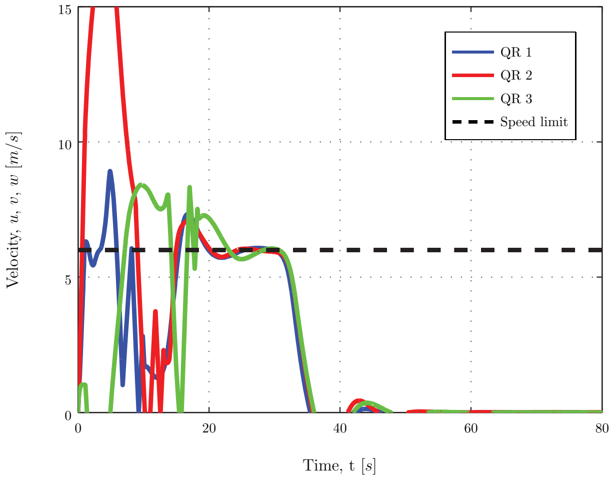


Figure 15: Velocity response for Case III simulation

D. Case IV: Obstacle avoidance

The figures in this section show the ability to avoid simple obstacles. A scenario with two obstacles placed between the starting point and the goal is run twice. In order to give a clear view of how the quadrotors interact with each other and the obstacles, the gains have been lowered to $k_x = k_y = 0.3$ to avoid 'maxing' out the velocity commands. This has been done to avoid confusion whether the quadrotors move under the influence of the Multi-UAV controller or, if the path is influenced by the saturated velocity commands. The two cases correspond to the first time just using the obstacle avoidance function 7 and the second time with the extra normal component activated as well (eq. 8). In both cases one has to adjust the gains to reach satisfactory behavior. This means that the obstacle avoidance is very sensitive to the gains settings.

Fig. 16 shows that the quadrotors are able to avoid the obstacles, but the response is not very smooth. This is due to the effects discussed in section IV. Figs. 18 and 19 show the effect of adding the extra collision avoidance term. It creates an evasive movement before the quadrotors get really close which results in a much smoother path around the obstacles. It must be noted that again, the gain setting corresponding to this term is very important. Consider for example a threat that is placed close to the starting point of a member. Its velocity will not yet be high, this will lead to a relatively larger influence of $\dot{r}_{on}$, resulting in an unnecessarily large evasive action.

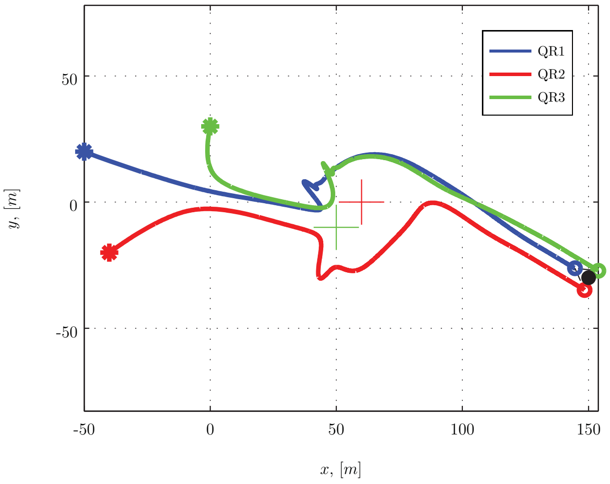


Figure 16: Position response for Case IVa simulation

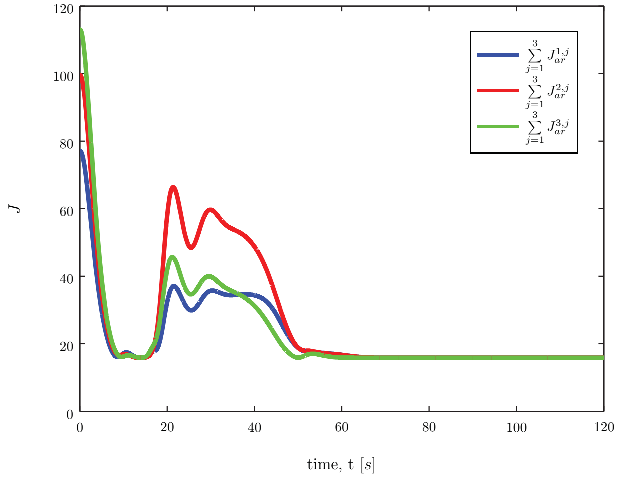


Figure 17: Potential function values for Case IVa simulation

Fig. 19 shows that the quadrotors are able to stay closer together with the extra normal component engaged.

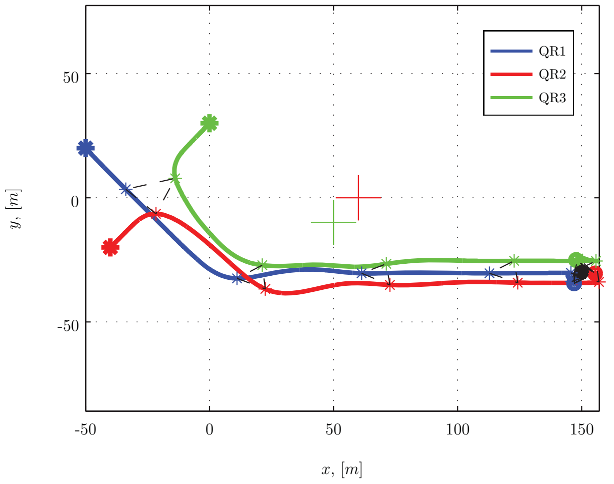


Figure 18: Position response for Case V simulation

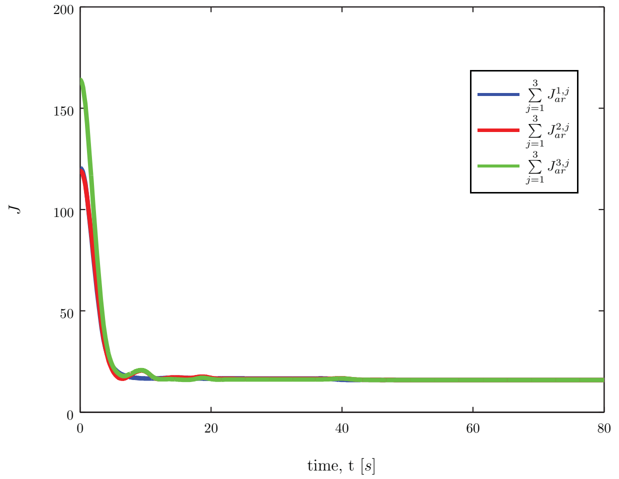


Figure 19: Potential function values for Case V simulation

E. Case V: Effect of limited sensor range

This section shows the effect of limited sensor range on swarm aggregation behavior and the obstacle avoidance performance. The first case starts with one quadrotor out of communication range. The second case shows what happens when the quadrotor has limited obstacle detection range.

1. *Case V-a: Limited communication range*

Fig. 20 shows a situation where two of the three quadrotors start within communication range and the third one is place outside the communication range. The range is set as $40m$. The results demonstrate that the controllers are able to cope with a situation where along the way a swarm member has to be included in the swarm. The figure clearly shows that the third quadrotor is flying in a straight line towards the goal. When other members get within range, the quadrotor tries to position itself at the equilibrium distance from the other members. This is visible as the spikes in figure 21. The first two quadrotors form a formation as indicated by the decreasing red line (the blue one is hidden, because it has the same value). The figure shows that the third quadrotor gets within range of the first quadrotor, roughly a second later, quadrotor-3 is also within range. Again, in this case the saturation limit on the position controls have been lifted and the position gains lowered to $k_x = k_y = 0.3$ so the quadrotors do not fly with saturated position commands.

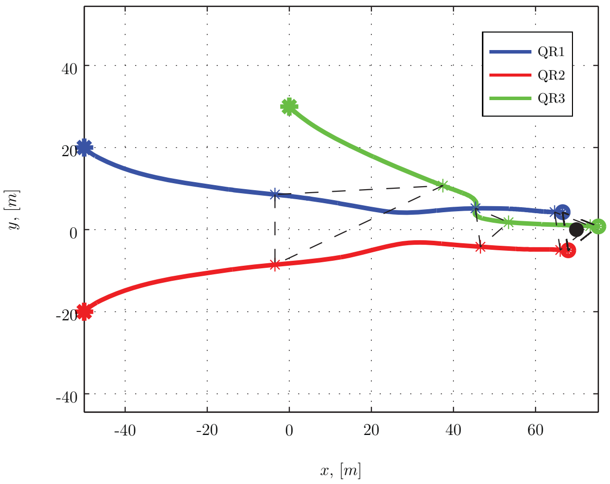


Figure 20: Position response for Case VIa simulation

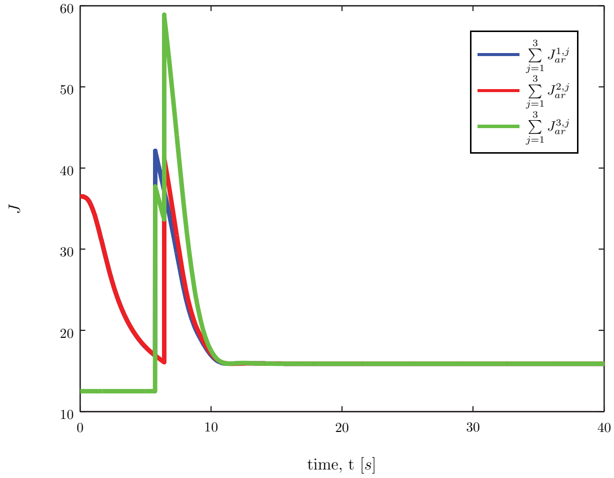


Figure 21: Potential function values for Case VIa simulation

2. Case V-b: limited obstacle detection range

The following figures show how the swarm responds to an obstacle placed about 50 meters away from the start positions, and using an obstacle detection range of 20 meters. The velocity plot in Fig. 22 shows that the response to avoid the obstacle starts too late. The quadrotors largest excitation form a straight line to the goal is little under 20 meters after they have passed the obstacle. Clearly the limited range prevents the quadrotors from taking evasive action in time. This is illustrative of the importance of good, reliable, long range sensors. Of course performance can be boosted by flying slower, to give the quadrotors more time to react.

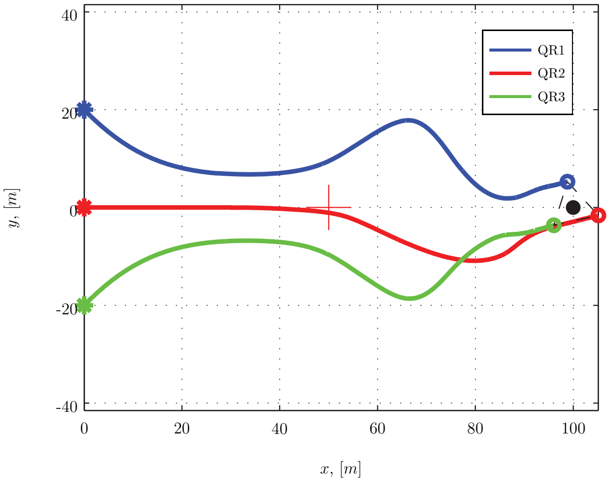


Figure 22: Position response for Case VIb simulation

IX. Conclusions and Recommendations

In this section we discuss some conclusions and recommendations following from the results of the simulations. The first section draws conclusions about the performance of the controllers. The next section introduces a number of recommendations for follow up research.

A. Conclusions

The previous sections have demonstrated that a nested backstepping controller is a suitable position controller for the quadrotor model. The controller is able to accurately steer the quadrotor to user defined positions, as well as control the heading of the vehicle. The method has a number of limitations. When large position differences exist between the actual position and the demanded position, the controller will generate unrealistic control inputs. By implementing saturation limits on the attitude and rate commands using hyperbolic tangent functions, the control commands can be kept reasonable so stability is ensured.

To be able to use one set of gains for a wide range of situations, the velocity command has been limited as well. This method works, but also has a drawback. When flying with saturated position commands, in other words, at maximum speeds, the velocity vector of the quadrotor is not pointed towards the goal anymore, but is just the sum of the three maximum speeds. In a 2D case, using the same saturation limits for the u- and the v-direction, this results in a 45° velocity vector. This means the quadrotor does no longer fly the shortest path to the goal. In the absence of obstacles and other quadrotors, this would be a straight line.

The multi-UAV controllers are effective in creating a stable swarm. Behavior is as would be expected from simulations using the same potential function approach using point mass models to represent swarm members. Small differences were observed between theory and simulation results. Using the quadrotor nonlinear models resulted in a slightly unstable swarm center. However, this movement was insignificant when compared to the displacement of the swarm and therefore did not destabilize the swarm.

A simple obstacle avoidance algorithm was implemented. The swarm is able to cope with obstacles and will reform the formation if broken up by an obstacle. Due to the symmetric nature of the potential function based control laws, the obstacle avoidance term creates little lateral movement whenever an obstacle is closely aligned with the quadrotors' flight path. The implementation of an extra avoidance term, perpendicular to the position-to-obstacle vector improves the avoidance performance. The correct setting of the gain of the term is important.

Also communication and obstacle detection limits have been implemented. They show a distinct influence on the quadrotor's ability to swarm and avoid obstacles. A limited obstacle detection range results in too little time to avoid the obstacle. The simulations show that quadrotors that are out of communication range are able to integrate into the swarm. This means that swarm aggregation does not need to start from a stand still.

The position gains and the swarm aggregation gains have been made variable, so there is an exchange possible between the two types of gains. By measuring the average inter-quadrotor distance the quality of the swarm can be determined. Low quality means an increase in swarm aggregation gain and a decrease in the position gains and vice versa. The effect of this approach is clearly noticeable and necessary when flying with high position gains.

The simulations have shown that potential based swarm control is an adequate method to control swarms of quadrotors. In the following section, a number of recommendations will be made that will increase the functionality of the controllers, so a more realistic simulation can be created, clearing the way for a future testbed.

B. Recommendations

First, the quadrotor model should be extended so it includes engine characteristics, the effects of aerodynamics and rotor dynamics. The responses of such a model would be closer to reality and would be a first step towards test flying.

Second, the functionality of the multi-UAV controller should be extended with an trajectory shaping module. By breaking up the path to the goal in smaller pieces, it is possible to keep the control commands within limits. This also enables path tracking, another must in swarm flying, as applications for simple straight line flying are limited.

Third, the ability of the swarm to handle obstacles should be improved. This research has modeled threats as points, assigned with a certain repulsive strength. This imposes no real distances and is therefore not realistic. Future work should introduce methods to better model threats and obstacles like walls, mountains, etc. Unreachable goals will occur when the goal and one or more obstacles are close to each other: a local minimum in the potential function will occur. In Ref. [19] a method is outlined to prevent this unwanted behavior.

Fourth, simulations using larger number of members should be performed to verify the controller's ability to control larger swarms.

Finally, the maximum velocity components in x-, y- and z-direction should be based on the components of the total maximum allowed velocity of the quadrotor. This makes sure that even when the velocity is at its maximum, the direction of the vector is not influenced.

References

[1] de Vries, E. and Subbarao, K., "Backstepping based Nested Multi-Loop Control Laws for a Quadrotor," *Proceedings of the Eleventh International Conference on Control, Automation, Robotics and Vision (ICARCV 2010)*, Singapore, December 2010.

[2] Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J., "Quadrotor Helicopter Flight Dynamics and Control:Theory and Experiment," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA, August 2007.

[3] Bouabdallah, S., Noth, A., and Siegwart, R., "PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor," *Proc. of The IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2004.

[4] Altuğ, E., Ostrowski, J. P., and Taylor, C. J., "Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback," *The International Journal of Robotics Research*, Vol. 25, No. 5, 2005.

[5] Voos, H., "Nonlinear Control of a Quadrotor Micro-UAV using Feedback-Linearization," *Proceedings of the 2009 IEEE International Conference on Mechatronics*, Málaga, Spain, April 2009.

[6] Bouabdallah, S. and Siegwart, R., "Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor," *Proc. of The IEEE International Conference on Robotics and Automation (ICRA)*, 2005.

[7] Raffo, G. V., Ortega, M. G., and Rubio, F. R., "MPC with Nonlinear H_∞ Control for Path Tracking of a Quad-Rotor Helicopter," *Proceedings of the 17th World Congress The International Federation of Automatic Control (IFAC)*, Seoul, Korea, July 2008.

[8] Chen, M. and Huzmezan, M., "A Combined MBPC / 2 DOF H_∞ Controller for a Quad Rotor UAV," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, August 2003.

[9] Madani, T. and Benallegue, A., "Control of a Quadrotor Mini-Helicopter via Full State Backstepping Technique," *Proceedings of the 45th IEEE Conference on Decision & Control*, San Diego, CA, USA, December 2006.

[10] Das, A., Subbarao, K., and Lewis, F. L., "Dynamic Inversion with Zero-Dynamics Stabilization for Quadrotor Control," *IET Control Theory & Applications*, Vol. 3, No. 3, 2009, pp. 303–314.

[11] Das, A., Lewis, F. L., and Subbarao, K., "Backstepping Approach for Controlling a Quadrotor using Neural Network," *Journal of Intellegent and Robotics Systems (special issue)*, Vol. 56, No. 1, April 2009.

[12] Reynolds, C. W., "Flocks, Herds and Schools: A Distributed Behavioral Model," *Computer Graphics*, Vol. 21, No. 4, July 1987.

[13] Park, C.-S., Tahk, M.-J., and Bang, H., "Multiple Aerial Vehicle Formation Using Swarm Intelligence," *AIAA Guidance, Navigation and Control Conference and Exhibit*, Austin, TX, August 2003.

[14] Kowalczyk, W. and Kozlowski, K., "Artificial Potential Based Control for a Large Scale Formation of Mobile Robots," *Fourth International Workshop on Robot Motion and Control*, June 2004.

[15] Gazi, V. and Passino, K. M., "Stability Analysis of Swarms," *Proceedings of the American Control Conference*, Anchorage, AK, May 2002.

[16] Gazi, V. and Passino, K. M., "A class of Attraction/Repulsion Functions for Stable Swarm Aggregations," *Proceedings of 41st IEEE*, Las Vegas, NV, December 2002.

[17] Gazi, V., "Swarm Aggregations Using Artificial Potentials and Sliding-Mode Control," *IEEE Transactions on Robotics*, Vol. 21, No. 6, December 2005.

[18] Han, K., Lee, J., and Kim, Y., "Unmanned Aerial Vehicle Swarm Control Using Potential Functions and Sliding Mode Control," *Proc. IMechE*, Vol. 222, 2008.

[19] Ge, S. S. and Cui, Y., "New Potential Functions for Mobile Robot Path Plannign," *IEEE Transactions on Robotics and Automation*, Vol. 16, October 2000.