An ILS Inspired Approach and Departure System Utilizing Monocular Vision

H. Claus Christmann
Graduate Research Assistant
Georgia Institute of Technology, Atlanta, GA, USA

Eric N. Johnson
Lockheed Martin Associate Professor of Avionics Integration
Georgia Institute of Technology, Atlanta, GA, USA

ABSTRACT

This paper introduces a simple system to provide relative position between a base unit and an active unit. The proposed system is directional and allows the active unit to approach or depart from the base unit along a linear path, determined by the orientation of the base unit. The system does not require a data link between the base and the active unit, just a clear line of sight. The proposed system utilizes monocular vision on the active unit and requires the availability of enough computational power to perform simple computer vision algorithms.
Part I describes the physical characteristics of the beacon utilized on the base unit, Part II describes the algorithms utilized to compute the relative position of the active unit to the base, utilizing the vision data. Part III presents simulation results. Part IV discusses the results and findings and proposes future work.

INTRODUCTION

Indoor UAV utilization most often has to deal with effects resulting from GPS denial. Beside the obvious lack of an absolute position fix, not having GPS available also diminishes the availably of relative positions to units a UAV interacts and communicates with – and to whom the UAV could get a relative position by comparing the respective absolute positions. Relative position is necessary for at least two main reasons: firstly, it is necessary to avoid collisions, and, secondly, it is necessary to precisely approach an object – the later being the core topic of this paper.

In commercial flight the Instrument Landing System (ILS) has been a long used method to allow safe landings under instrumental flight conditions. ILS provides this safe landing capability by emitting a radio pattern which, if interpreted correctly, creates a path in space an aircraft with ILS equipment can glide along until touchdown. ILS equipment also allows a measurement of deviation from this path and hence can give qualitative information on how to get back on track. For more information on ILS see [1] or [2].

However, the ILS system is based on radio interferometry and both, the parts installed on the ground as well as the airborne units, are technologically complex and not necessarily suitable for use with indoor UAVs. The main task at hand is to find a way to replicate the ILS features in a way that requires less complex equipment but still provides relative position and, if possible, relative attitude.

Revisiting the sensor suits of generic UAVs, monocular vision most often is present as live video of remote locations is a main driver for UAV utilization. Based upon the assumption of availability of The proposed beaconsome kind of monocular vision on most indoor UAVs, a system had to be created that could make use of this sensor.

PART I – BASE UNIT BEACON

Basic Principle

Resulting from the limitation of monocular vision, the beacon at the base station has to be shaped in a way that allows relative distance and relative attitude to be gathered from a 2D picture of that beacon.

The proposed beacon consists of a pattern of 5 light sources, 4 of them arranged in a rectangular pattern, the 5th one recessed. The line originating from the recessed light source passing through the centroid of the 4 in-plane sources determines the path (the nav-aid path) in space the active unit (the UAV) uses as an approach or departure path. To allow for easy differentiation of the light sources in a picture, the colors are different. The recessed source is of one color (e.g. yellow), diagonally opposed sources in the rectangle have the same color (e.g. red and green, resp.).

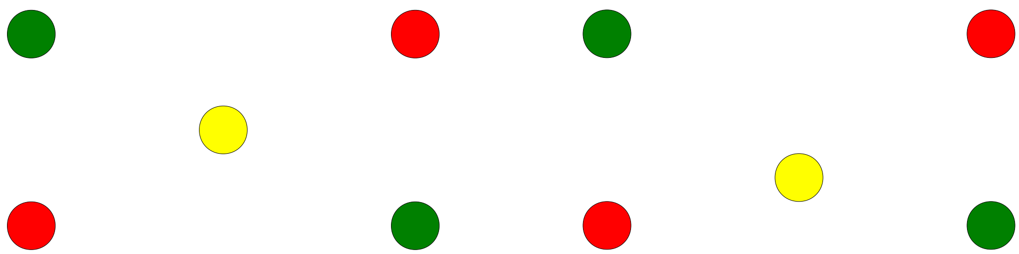


Figure 1: Base Unit Light Pattern

The arrangement of the light sources in 3D space results in a changing 2D picture of the pattern, depending on the orientation of the camera to the beacon. For example, as long as the camera is located along the nav-aid path, the recessed light source will appear in the centroid of the 4 sources that make up the base of the navigation aid pyramid and the relative position of the recessed source to the centroid will change with the relative position of the camera to the path.
In the left part of Fig. 1, the camera is aligned with the nav-aid path, in the right part, the camera is below the nav-aid path (the recessed yellow light source appears below the centroid of the red and green light sources).

As the system will be directional, the authors propose to use light sources which also are directional to a certain level. Medium beam width LEDs of different colors seem to be a suitable choice. LEDs also have the benefit of a known (to a certain extent) spectral intensity distribution, allowing image processing algorithms to be tweaked to the exact wavelength of the LEDs.

Key Characteristics

As the base unit beacon essentially is a pyramid with the base light sources forming the base of the pyramid and the recessed light source the tip, the geometry presents the first set of key characteristics. A larger base will result in a generally larger pattern in the camera image, a larger height of the pyramid, i.e. the distance of the recessed light source to the base, determines how much image of the recessed source moves with respect to the centroid of the base when the camera deviates from the nav-aid path.

A second set of characteristics is the position and attitude of the base unit. Obviously, position and "pitch" and "yaw" determine the nav-aid path and hence are determined by the path the active unit (the UAV) has to follow. Hence, the parameters of this set are the orientation of the nav-aid path and the "roll" of the base unit.

Initial Parameters

The research for this paper was conducted in support of the 2009 entry of the Georgia Tech Aerial Robotics Team (GTAR, [3]) to the 5th mission of the International Aerial Robotics Competition (IARC, [4]) of the Association for Unmanned Vehicle Systems International.

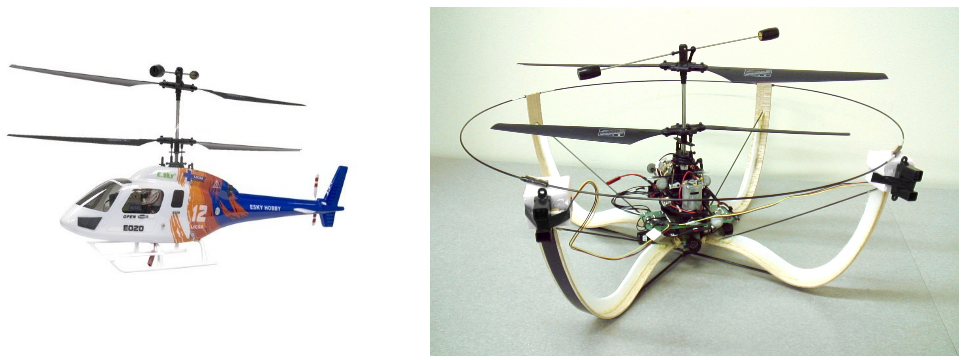


Figure 2: GTAR Vehicle

The GTAR vehicle, Fig. 2, is a stable coaxial helicopter, based upon the Esky Big Lama. The described ILS system is intended to guide the GTAR vehicle into a window at a distance of approximately 3m to 4m. During this phase of flight, the vehicle will fly relatively slow and we assume pitch and roll of the vehicle to be negligibly small. The GTAR vehicle is equipped with a small camera, providing the monocular vision sensor necessary for the described system.

The position and attitude of the base unit were picked in order to provide a horizontal nav-aid path, leading to a point centered in the lower third of the window. The off center point was chosen as the camera is not mounted in the geometric center of the vehicle, however, the vehicle should clear the window equally on all sides (top, bottom, left, right).
The "roll" of the base unit was chosen to be aligned with the orientation of the camera frame. This setup allows the image processor to determine the position of the base unit with only two light sources visible. The four options, above, below, to the right, to the left, correspond to "red to the left of green", "green to the left of red", "green above red", and "red above green" in Fig. 1.

The field of view (FOV) of the camera at the assumed start location determined the size of the base unit and the width and height were chosen to maximize the pattern size in the initial image. This is necessary as with larger distance from the nav-aid the perceived separation of the light sources shrinks in the image, leading to a lower resolution with increasing distance. However, in order to allow the camera to be mounted in either landscape or portrait orientation, the base was chosen to be square. Further comments are given in Part IV.

PART II – PROCESSING NAV-AID PATTERNS

The general assumption for this research is that the base unit is utilized in conjunction with an active unit that has a sufficiently developed controller to interpret higher level guidance commands. The nav-aid software will hence only issue these guidance commands to the underlying flight management system (FMS).

Guidance Zones

The nav-aid generally is utilized in guidance of the active unit. Independent of an approach or a departure from or to the base unit, respectively, eight zones can be identified, pictured in Fig. 3. The zones are determined by the intersection of the lines through same color light sources and the boundary of the base of the base unit pyramid.

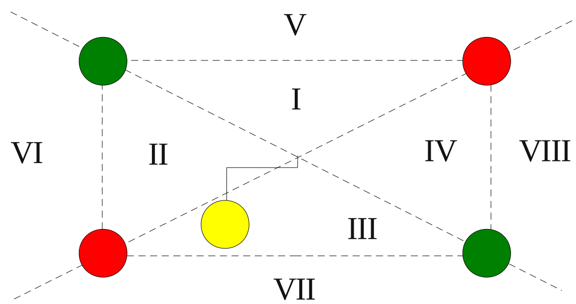


Figure 3: Guidance Zone Example

Dependent on where the recessed light source appears, the following general guidance commands can be identified:

Table 1:Guidance Zones

Zone	Vehicle Position (relative to nav-aid path)	Command
I	"above"	"descent"
II	"to the left"	"translate to the right"
III	"below"	"climb"
IV	"to the right"	"translate to the left"
V	"way above"	"stop longitudinal motion and descend"
VI	"way to the left"	"stop longitudinal motion and translate to the right"
VII	"way below"	"stop longitudinal motion and climb"
VIII	"way to the right"	"stop longitudinal motion and translate to the left"

Based upon these zone commands, the vehicle is expected to reach a point on the nav-aid path. Starting in the position which results in the pattern in Fig. 3, the nav-aid software would issue commands guiding the vehicle first to climb. Once the yellow light source reaches zone II (with an expected overshoot), the command would change from "climb" to "translate to the right". Once again in zone III, "translate to the right" would be changed back to "climb" again, resulting in aligning the recessed light source with the centroid of base which in return indicates a position of the vehicle on the nav-aid path.

Image Frame Location of the Nav-Aid Pattern

Beside the actual pattern of the base unit in the captured image, the general location of the pattern also is needed to guide the vehicle. The guidance zones only deal with translatory motion (i.e. the velocities u, v, w in classic flight mechanics notation), the attitude in terms of Euler angles (φ, θ, ψ) needs to be addressed from a different source.

The general location of the complete pattern in the image frame of the camera can be used for that.

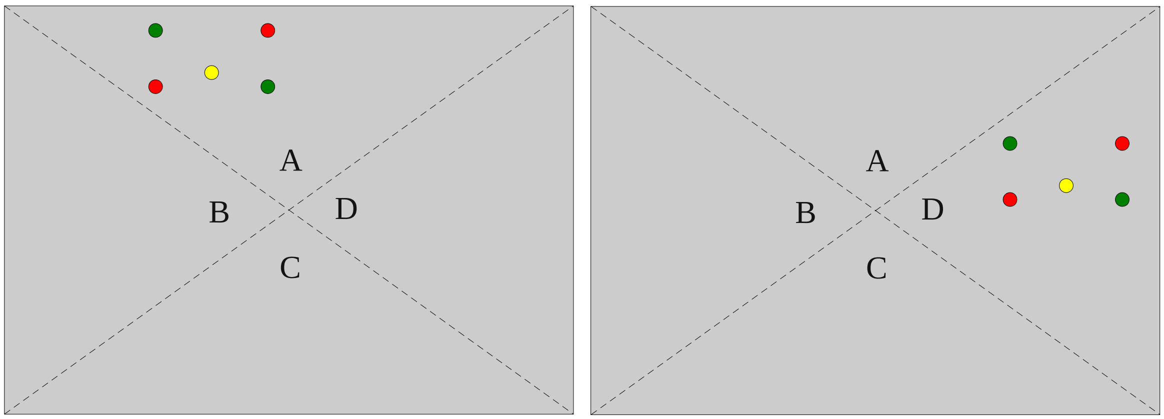


Figure 4: Location in the Image Frame

Comparable to the guidance zones I to VIII, the image frame of the camera can also be divided into four zones: *A-D*. This leads to the following extension of Table 1:

Table 2: Guidance Zone Extension

Zone	Vehicle Attitude (relative to nav-aid position)	Command
A	"pointed under"	"pitch up"
B	"pointed to the right"	"yaw left"
C	"pointed above"	"pitch down"
D	"pointed to the left"	"yaw right"

As an example, the situation depicted in the left part of Fig. 4 would correspond to a guidance zone of *AIII*, meaning the vehicle would need to "pitch up" and "climb" in order to reach an aligned attitude at a point on the nav-aid path. Correspondingly, the situation depicted in the right part of Fig. 4 would result in the guidance commands "yaw right" and "climb", in accordance with the zone *DIII*.

The combination of the zones *I-VIII* for the velocities u,v,w and *A-D* for the Euler angle rates q and r provide all the necessary information to estimate the relative position of the vehicle with respect to the base unit for the application at hand (the window entry problem).
In extension to the described data, the pixel separation of the base light sources could be used to estimate distance from the nav-aid base unit. The roll attitude could also be estimated from the image by comparing the alignment of the base edges with the image frame boarders, however, this can be skipped for the GTAR vehicle.

GTAR Specific Adaptations to the Guidance Scheme

The GTAR vehicle allows certain assumptions which simplify the guidance scheme. One benefit of the chosen platform is the inherent stability, provided by the Bell bar of the top rotor. Another benefit is that the resulting pitch and roll angles for longitudinal motion along the vehicle's body *x* axis and lateral motion along the vehicle's body *y* axis, respectively, can be assumed to be very close to zero for small velocities. However, as pitch and roll are directly coupled with translational velocities, an independent commanding of *p*, *q* and *u*, *v*, *w* is not possible.

The problem at hand is a window entry problem,which in a nav-aid perspective this equals a departure. The GTAR team chose to perform this entry at a fixed longitudinal velocity which can be achieved by manipulating the trim setting of the vehicle. This sets *u* at a predefined fixed value. As described above, *p* and *q* can also not be commanded and can be assumed to be zero throughout the departure.
This leave *v*, *w*, and *r* to be determined by the nav-aid guidance.

Under the given assumptions of a horizontal vehicle ($\varphi=\theta=0$) and a horizontal nav-aid path, the determination of w (climb or descent) can be changed from zones I, III, V, and VII to just zones A and C. This also has the benefit that the resolution of the relative vertical position estimate is greatly increased, as the measurement of relative pixel distance of the recessed light source and the centroid of the base can be replaced by a measurement of the average pixel coordinates of all light sources with respect to the center of the image frame.

GTAR Nav-Aid Guidance Look-Up Table

Making the adaptations mentioned above, this leads to the following guidance zones:

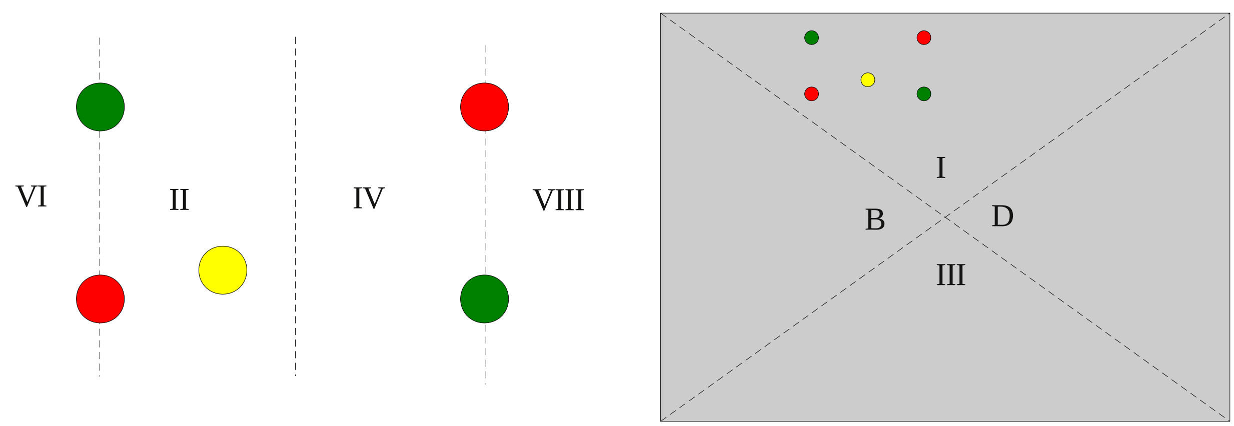


Figure 5: GTAR Guidance Zones

The implemented control architecture however does not use just the zone information, but actually three independent P-controller loops for the not predefined variables *v*, *w*, and *r*. For *v* the input is the horizontal pixel distance from the recessed light source compared to the centroid of the base, for *w* the input is the vertical pixel distance of the mean of the pixel locations of all detected light sources compared to the center of the image, and the input for *r* is the horizontal pixel distance of the same measure.

PART III – SIMULATION

The main goal of the simulation was to evaluate the feasibility of the proposed nav-aid system. Crucial to that is the determination of the minimal update rate of the image processing based controller and, playing into that, the determination of the minimal resolution of the captured images.

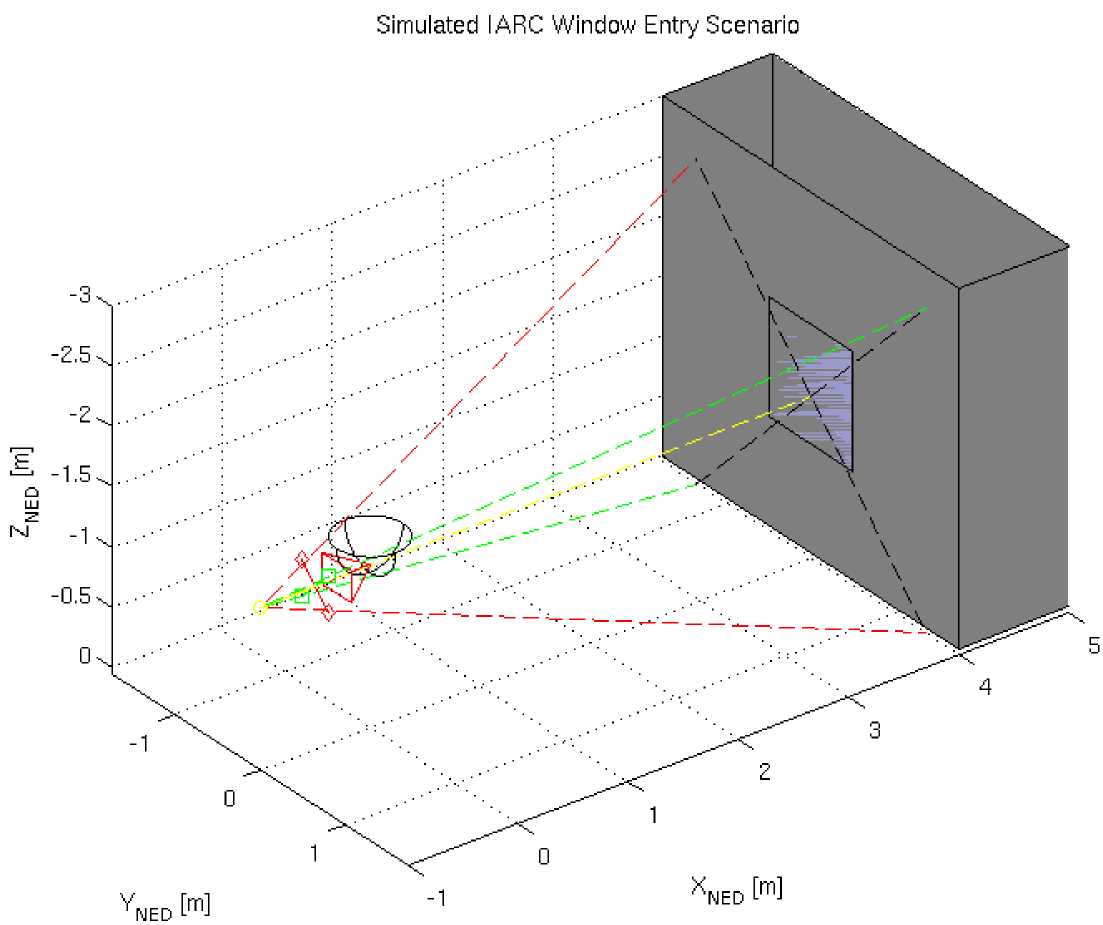


Figure 6: MATLAB Simulation Scenario

In order to mimic the IARC scenario, the vehicle starts outside a 3m boundary around the building to be entered. The nav-aid path is horizontal, aimed at the center of the lower part of the window opening. The vehicle is assumed to be manually piloted into a suitable position and then to be switched over to auto control.

In Fig. 6 the black wireframe represents the GTAR vehicle, the red pyramid attached to the vehicle represents the FOV of the utilized camera system. The base of the base unit beacon is represented by green rectangles and red diamonds, the recessed light source is depicted as a yellow circle.
Also drawn are the rays that delimit the original sectors *I-IV*. The dashed yellow ray represents the nav-aid path.

Appendix A outlines the core of `main.h` of the MATLAB based simulator: An outer loop goes through several predefined start positions, marking the corners of a box in which the manual pilot has to

maneuver the vehicle. If the vehicle is inside this box and all light sources are captured by the camera, the manual pilot is assumed to switch over to auto.
The main simulation loop steps forward through time, updating the vehicle position based upon the issued guidance commands. At a selectable update rate the simulated image processing is performed and a new guidance command is computed.

Start Position Box

The eight corners of the start position box were chosen to replicate the possible extrema. The vehicle is horizontally positioned as close or as far as possible from the base unit beacon, limited by either the proximity to the beacon or the “do-not-cross” line 3m in front of the window, respectively. The altitude and vehicle attitude are given by moving the light source pattern to the extreme corners of the camera FOV.
The start positions to the right of the beacon have the camera facing outward with respect to the beacon, the start positions to the left have the camera facing inward.

The optimal start position in (x, y, z) coordinates matching Fig. 6 would be (0, 0, 0) with an attitude (φ, θ, ψ) of (0, 0, π).Simulation Results

Simulation Results

The following pages present the results for all nine simulation runs (the eight corners plus the optimal start position). Shown are the top view, the front view (from the beacon toward the window) after the run is terminated, and the initial camera image.

The naming convention of the scenarios is as follows:

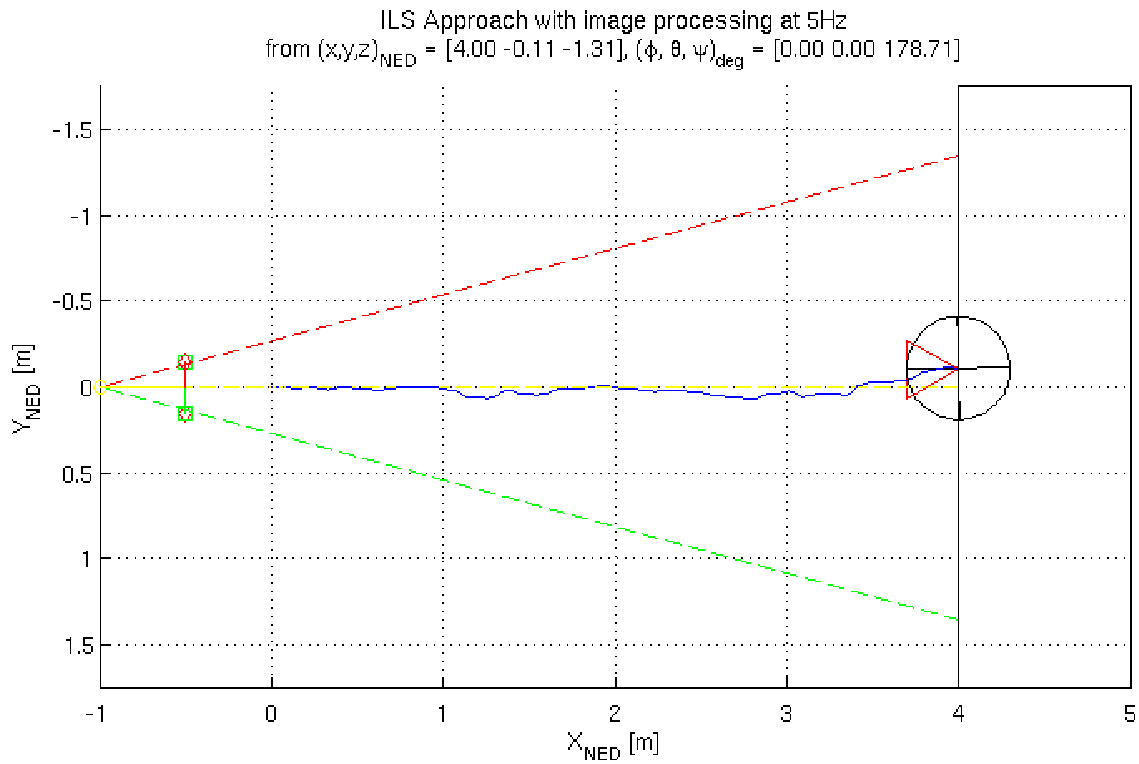
- *near/far* indicate the horizontal distance of the vehicle to the beacon
- *top left, e.g.,* indicates the corner of a rectangle when looked at the beacon from the window.
- *inward/outward* relate to the camera pointing either towards the nav-aid path (inward) or away from it (outward).

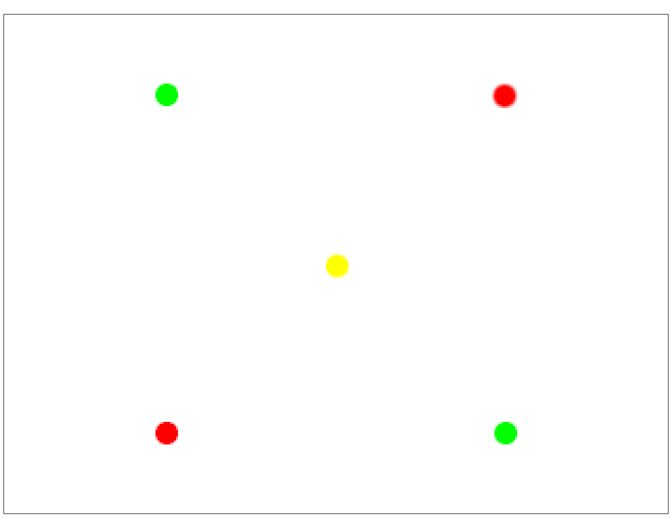
All these runs were performed with a simulated camera caption resolution of 320x240 pixel and a simulated update rate of 5Hz for the image processing loop. Furthermore it is assumed that the image processing algorithm is able to detect the center of the light sources with a sub-pixel resolution of 1/10 of a pixel and that that detection is prone to a white noise uncertainty of +/- 2 pixel.

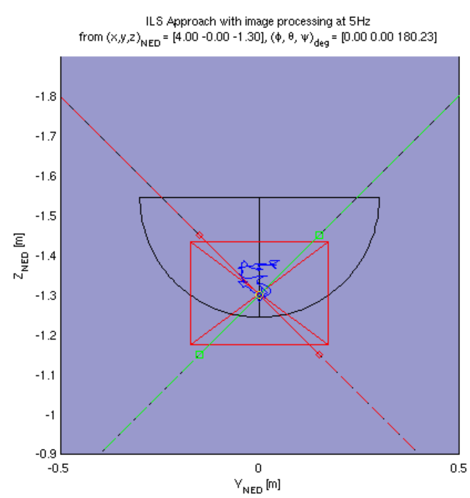
As the scenario is symmetric, the outward/inward orientation is altered with the right/left orientation of the start position with respect to the base unit beacon. Hence the eight presented cases reflect all 16 possible combinations due to symmetry.

A white noise error is added at every time step in order to simulate drift of the vehicle.

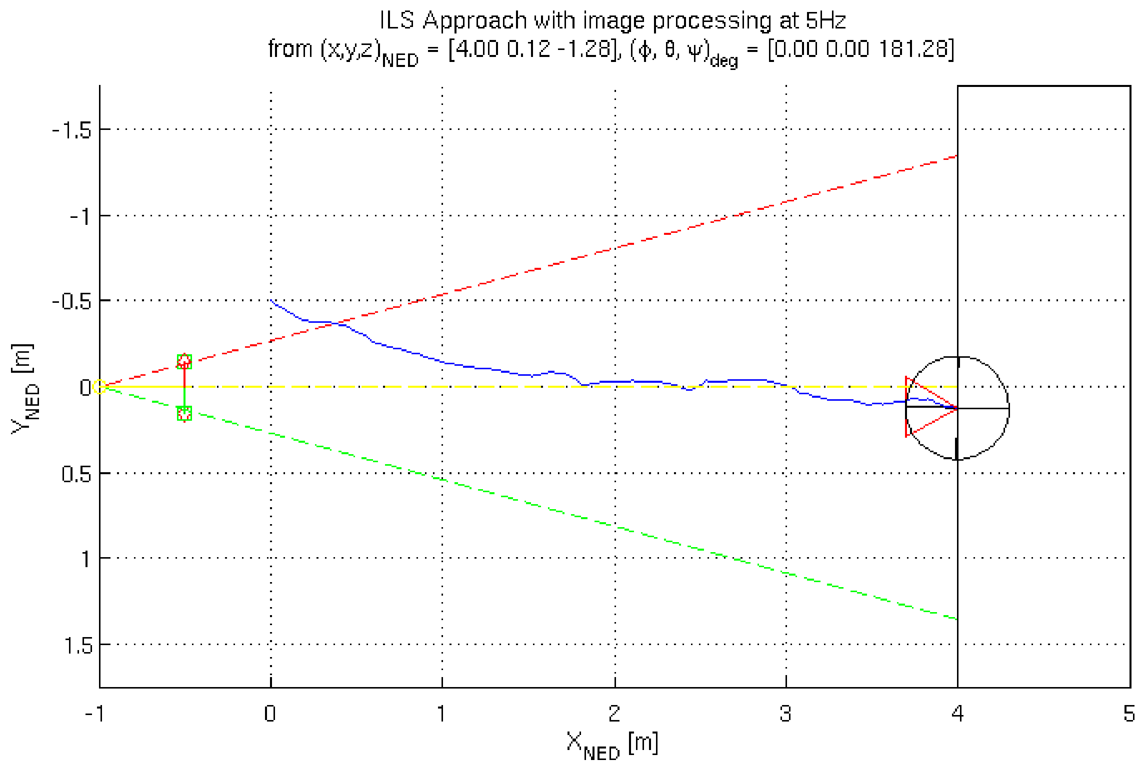
Optimal Start Position

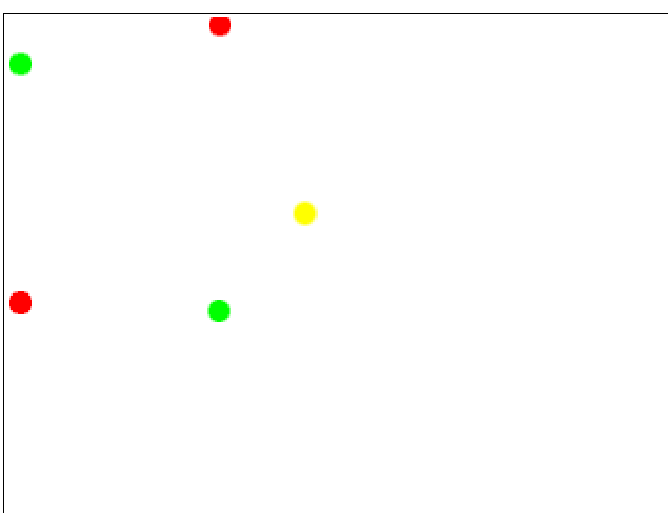


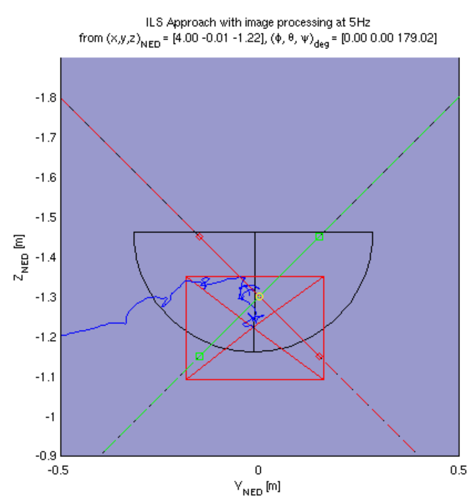




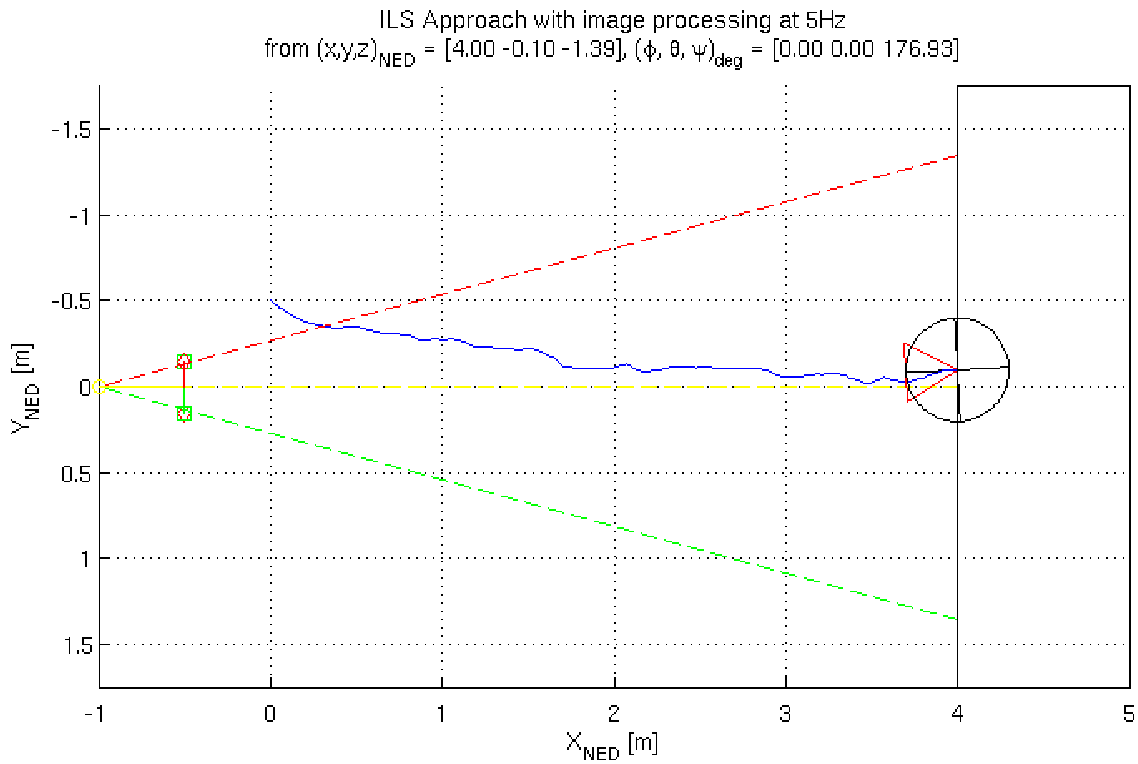
Near Bottom Right Corner (Outward)

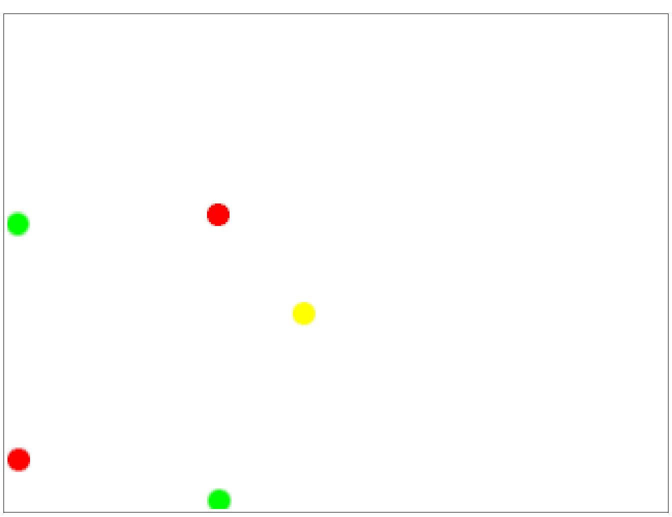


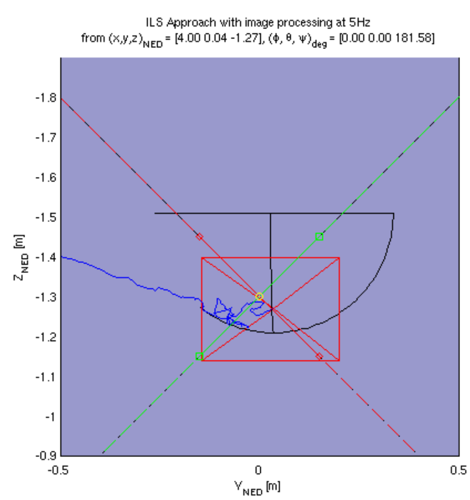




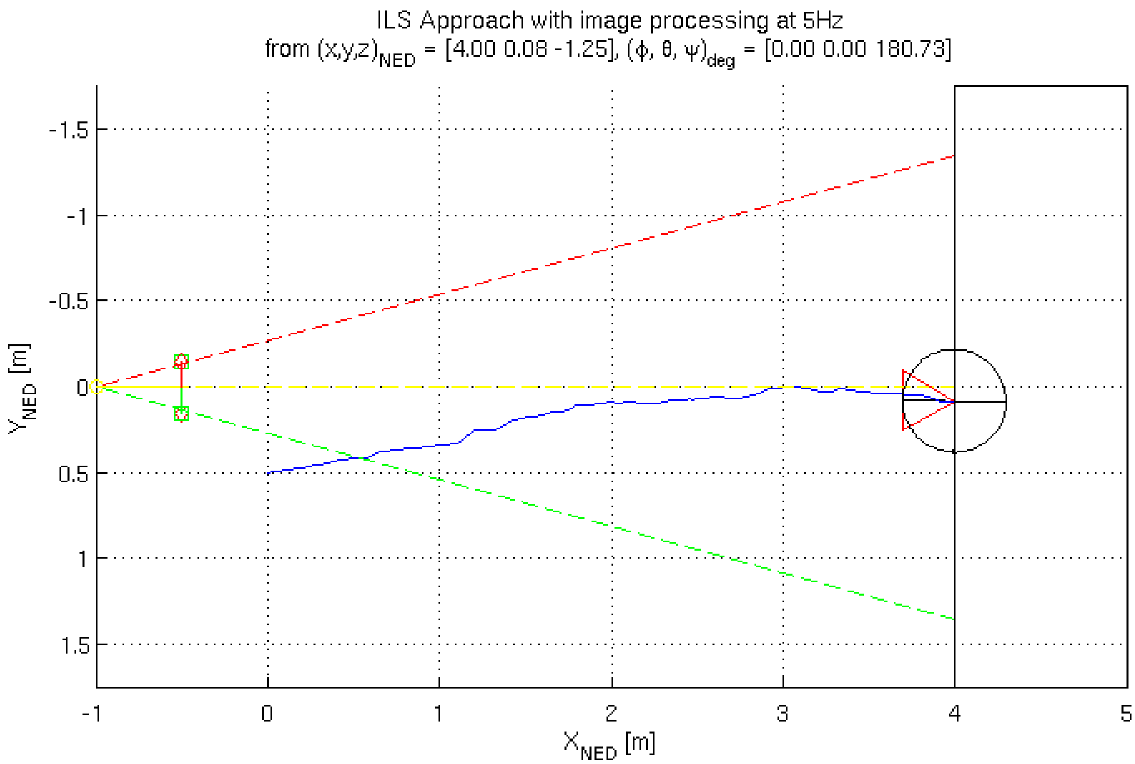
Near Top Right Corner (Outward)

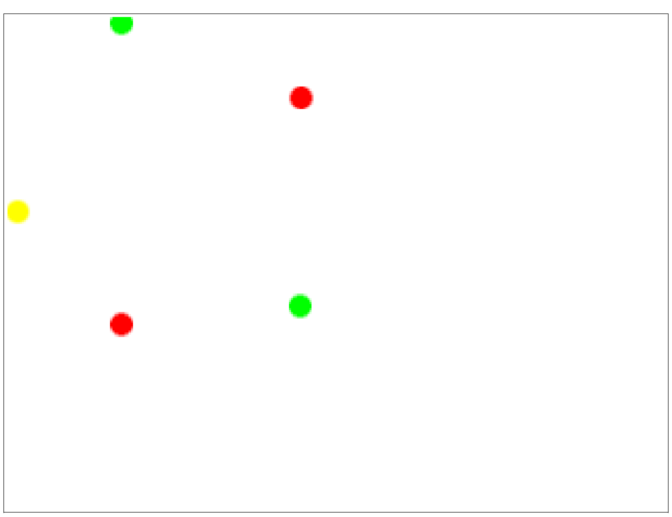


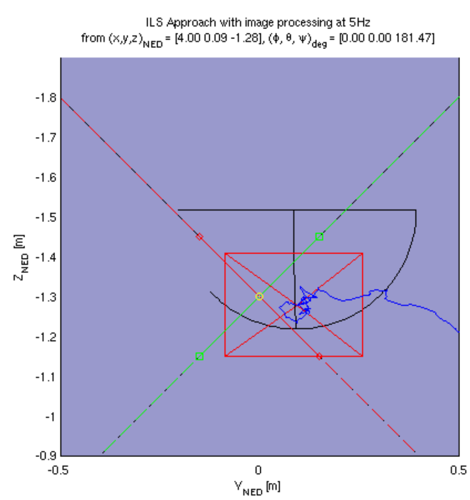




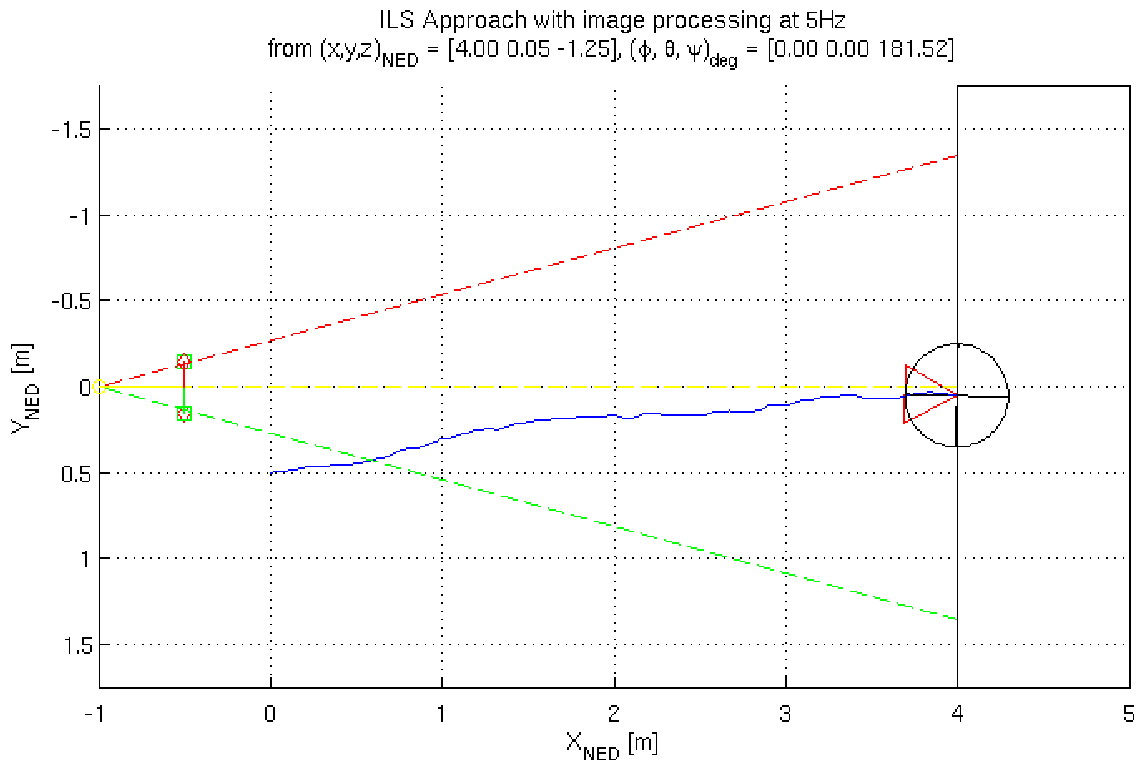
Near Bottom Left Corner (Inward)

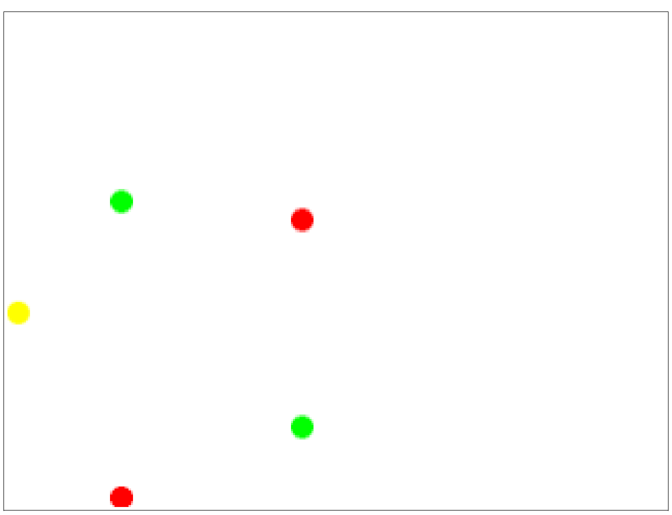


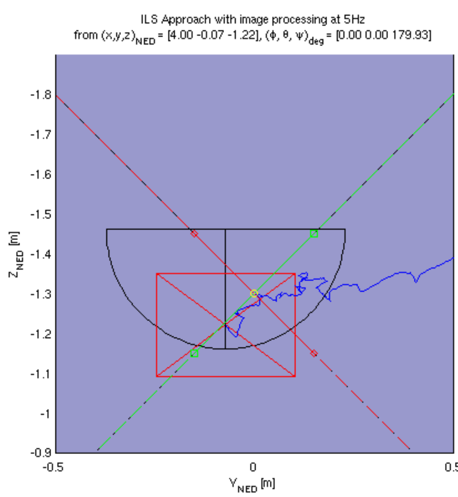




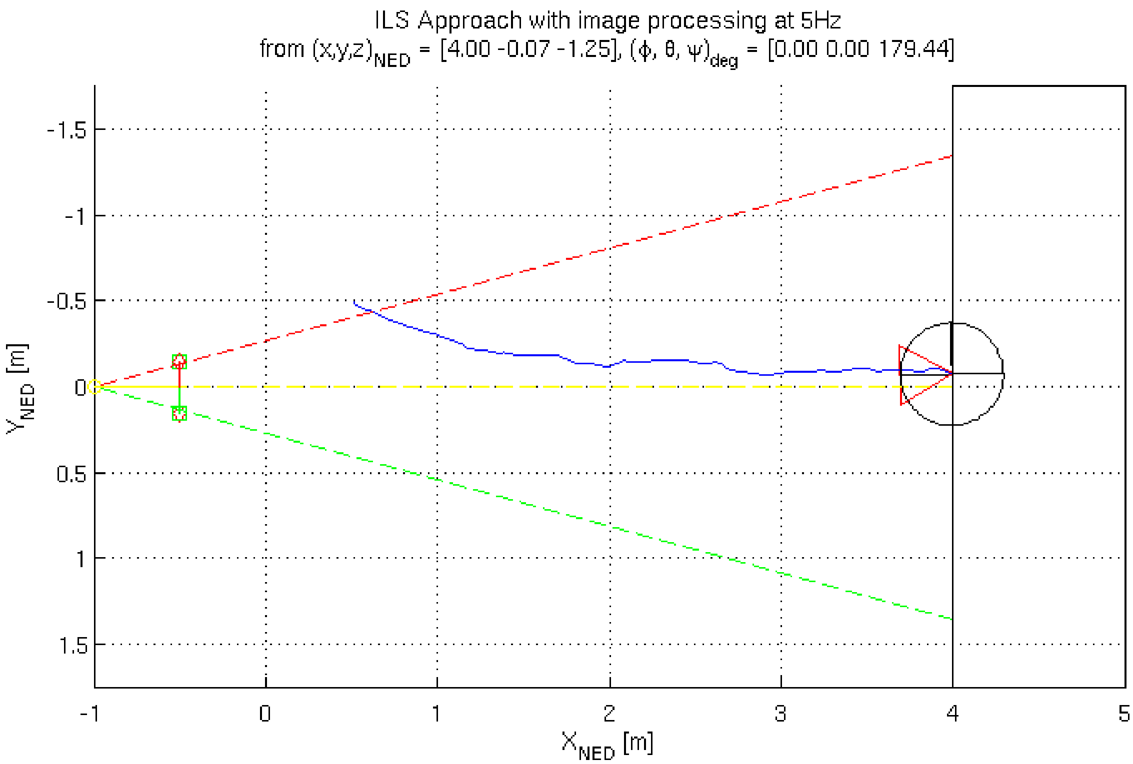
Near Top Left Corner (Inward)

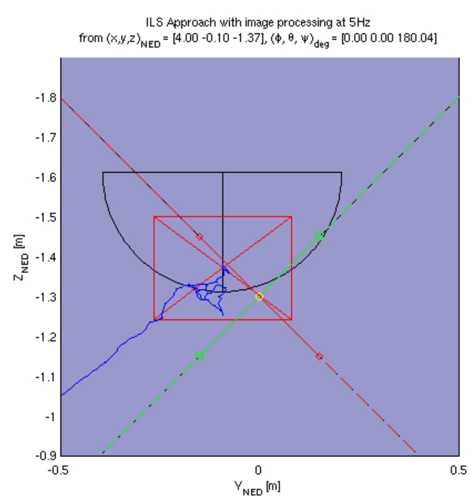




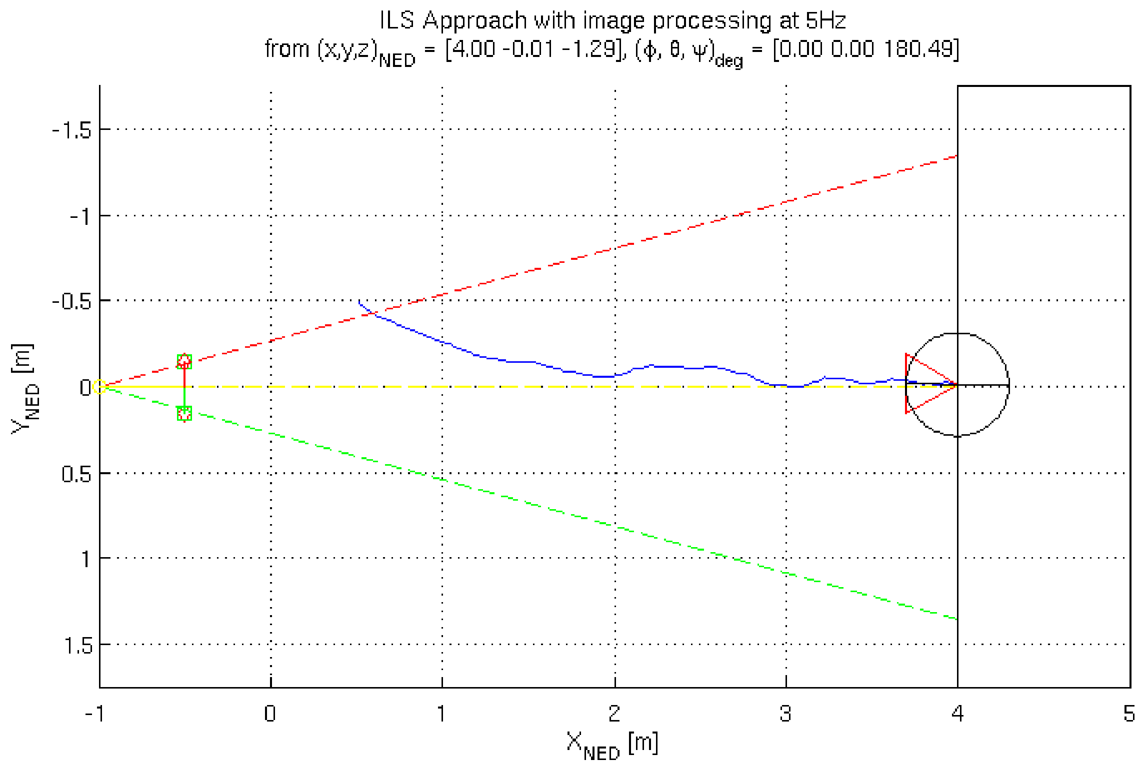


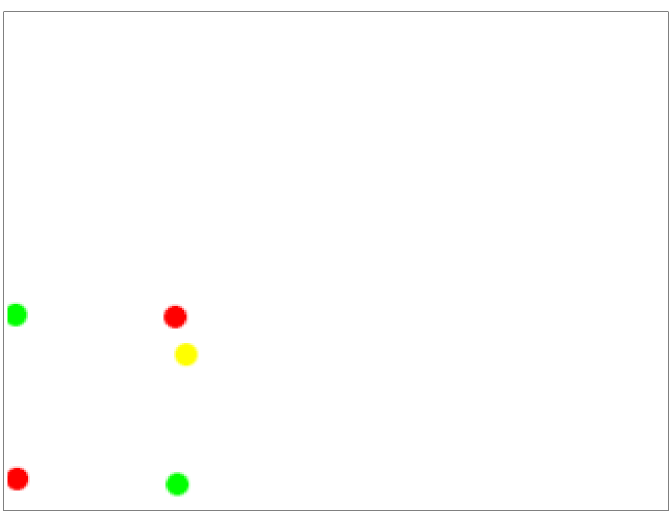
Far Bottom Right Corner (Outward)

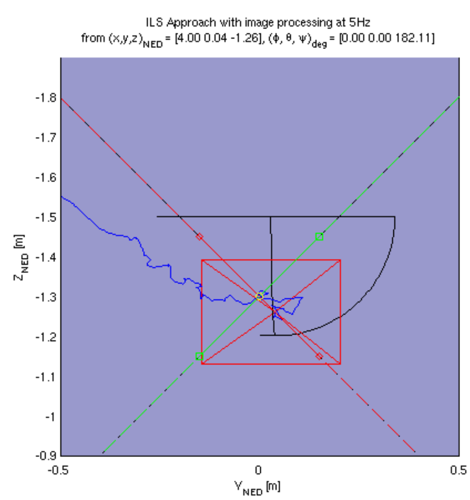




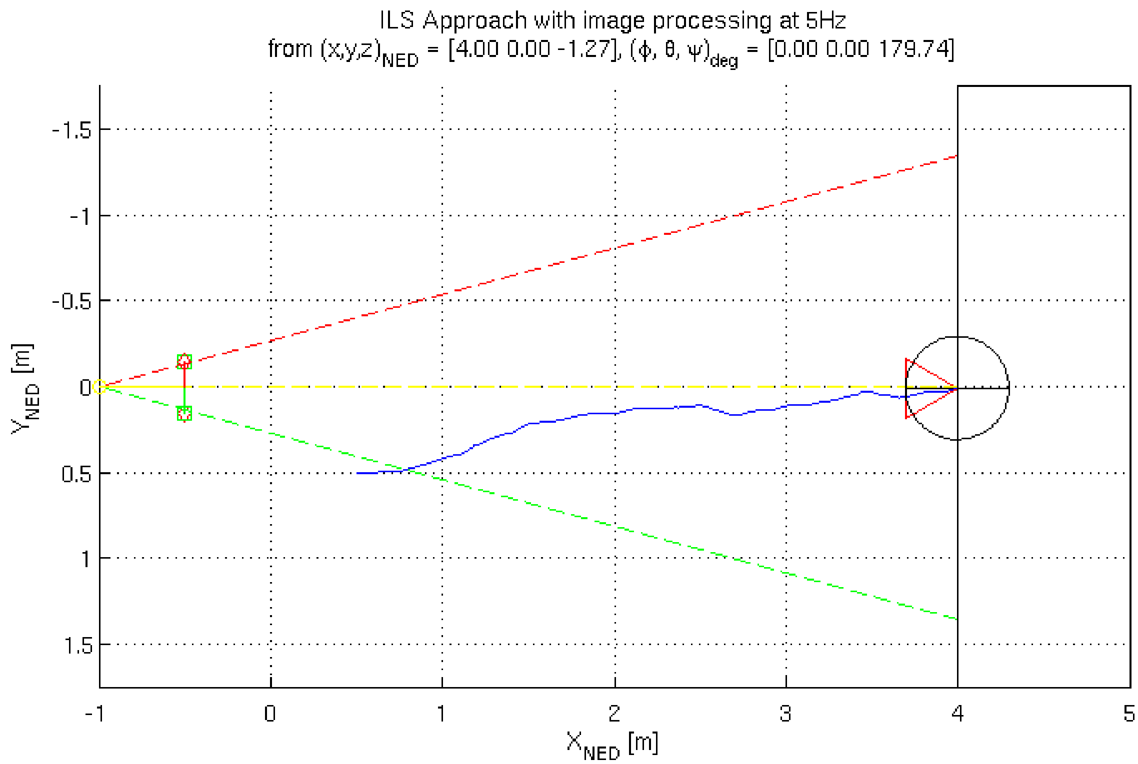
Far Top Right Corner (Outward)

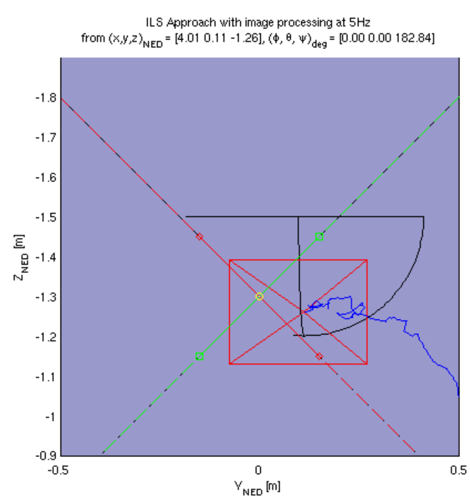




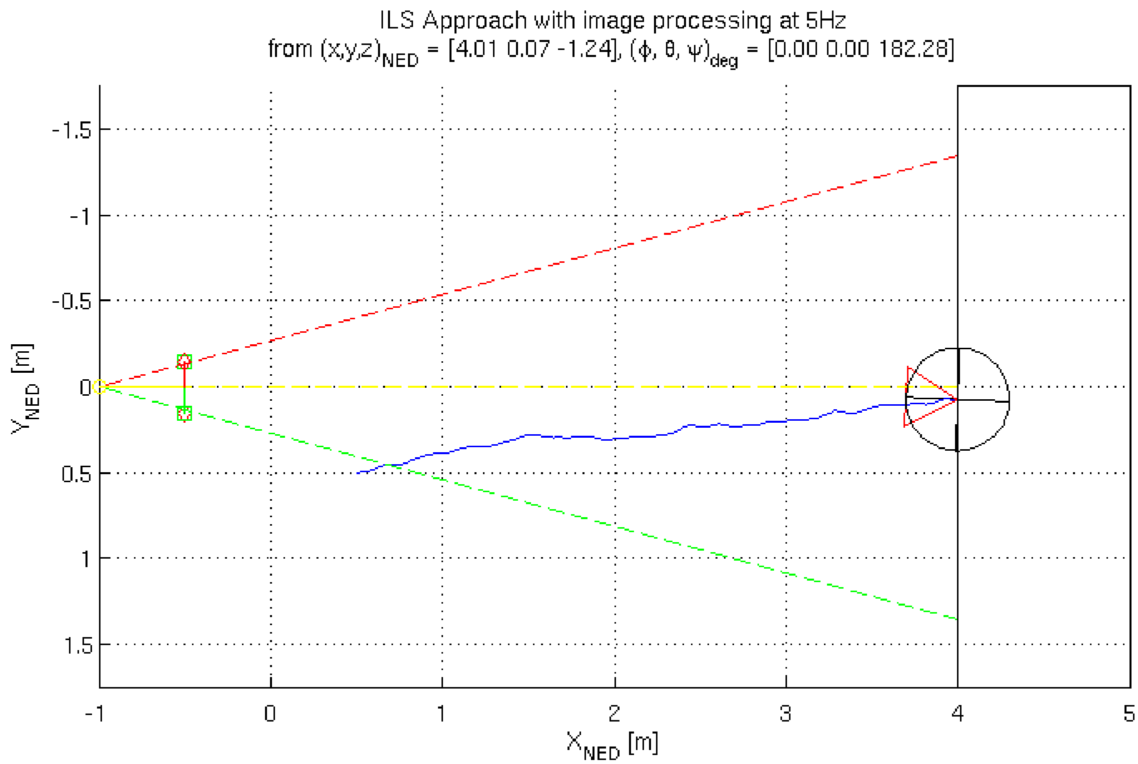


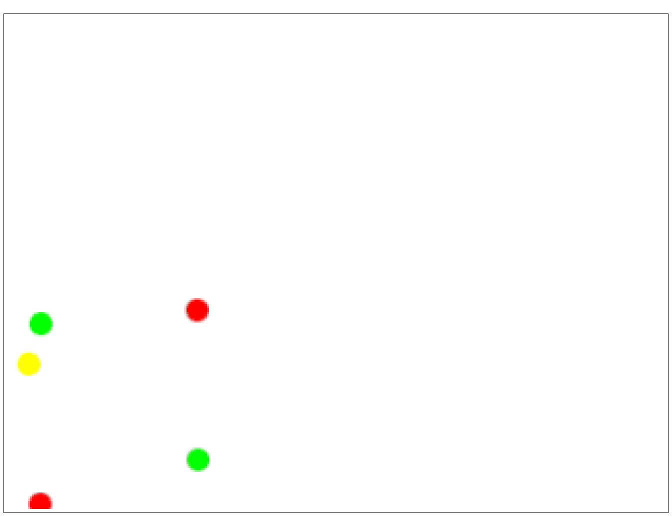
Far Bottom Left Corner (Inward)

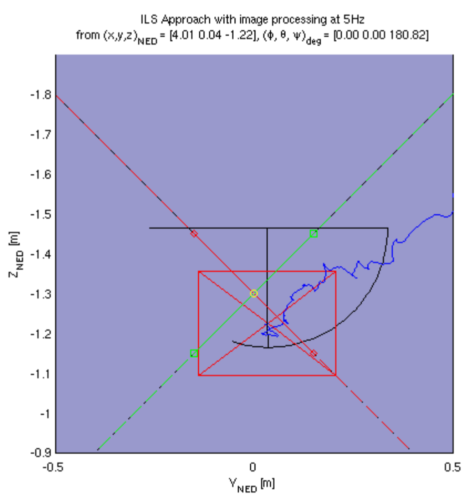




Far Top Left Corner (Inward)







PART IV – DISCUSSION OF THE FINDINGS

General Conclusion

The general result of the simulation seems to indicate the feasibility of the proposed ILS inspired approach and departure system.

Though in general the performance of the controller is weak and a slow approach of the desired navigated path is observed, the implemented controller is of the simplest proportional kind. More sophisticated controller architectures (or the simple inclusion of a differential and integral term) surely would improve metrics such as rise speed and steady state error.

The presented results however show that the system with the given parameters for update rate and resolution performs tolerable. Assuming that computational power is available off-board, processing an analog video stream from the vehicle, a capture resolution of 320x240px and a update rate of 5Hz seems conservative and an increase in either one of the parameters is resulting in a better estimation of the relative position.

Resolution Issues

A major systematic drawback of the proposed system is given for the used departure case. Given by simple geometry, the accuracy of the relative position estimate decreases with an increase in distance – the base unit beacon pattern simply appears smaller in the captured image, resulting in a decrease in resolution as a 1px change now corresponds to a larger spherical angle.
Hence generally the system would perform much better as an approach supporting system.

Camera Orientation

Related to the resolution issue is the choice of the orientation of the on board vehicle. If the utilization of the camera for the ILS system would be the dominant case, the camera should be mounted in a panorama/horizontal fashion. This would allow for a greater horizontal disparity of the left and right light sources of the base which directly affects the accuracy of the lateral translation command as the base could be chosen to be wider in order to completely fill out the initial camera frame at the starting location.

However, as the vehicle is severely limited in pitch actuation (as pitch is coupled with longitudinal velocity) a portrait orientation might be preferred if the dominant use of the camera is video surveillance. With a portrait orientation, a 360deg yaw sweep simply covers more of the surrounding. Due to this fact the base dimensions have been selected to be square, allowing for a panorama or portrait orientation of the camera with the same relative position estimation.

Future Work

In order to improve the performance prediction several steps need to be taken:

- inclusion of a proper dynamical model for the coaxial vehicle of the Georgia Tech Aerial Robotics Team
- improvement of the synthetic camera image
- inclusion of realistic C based vision processing algorithms in order to perform a complete software in the loop simulation

REFERENCES

(1) Kayton and Fried, “*Avionics Navigation Systems*”, Wiley-Interscience, 1997, ISBN 0471547956
(2) Wikipdia, “*Instrument Landing System*”, http://en.wikipedia.org/wiki/Instrument_landing_system
(3) GTAR Website, http://controls.ae.gatech.edu/gtar/
(4) IARC Website, http://iarc.angel-strike.com

APPENDIX A

```
%% Outer Loop
%
% This loop reruns the simulator for all starting positions defined in
% startPositions

disp('Press any key to start the simulation...');pause;

for i=1:length(startPositions(:,1))

X_NED = startPositions(i,:);
disp(sprintf('Setting start position to\nX_NED = [%f %f %f %f %f %f]',X_NED));
updateVehicleGraphics(X_NED,frame_B,cam_B,vehicle_hndl)


%% Main Simulatio Loop
%
% This part is the main outer loop for the simulation. The loop uses a
% fixed time step, stored in the global variable DT. The loop stops at
% either raching SIM_END_TIME or the vehicle center breaching the window
% plane.
% Internal to the sim loop is the controller loop whose update rate can be
% set via CNTRL_RATE in Hz.


C_B = [];          % [u_c v_c w_c p_c q_c r_c] commanded vehicle velocities and angular rates
history = [];

lastControllerUpdate = -1; % -1 ensures that the controller loop runs at t=0
tic % used to measure real time usage for the sim, (goes together with "toc" below)

for t = 0:DT: SIM_END_TIME
    updateVehicleGraphics(X_NED,frame_B,cam_B,vehicle_hndl)

    if (t >= (lastControllerUpdate + 1/CNTRL_RATE))
        cam_image = getNavAidImage(X_NED); % get new camera image , i.e. perform a "frame grab"
        set(pic_hndl,'CData',cam_image); % redraw synthetic camera image
        C_B = computeControls(cam_image);
        lastControllerUpdate = t;
    end

    C_B = C_B + NOISESD*([rand(1,3) 0 0 rand]-0.5*[ones(1,3) 0 0 1]); %  only add noise to u,v,w, and r

    history = [history; ...
               t X_NED C_B]; % store values for post processing

    X_NED = updateVehiclePos(X_NED,C_B);

    if X_NED(1) > WINDOW_CENTER(1)
        break; % stop the simulation loop if the vehicle breaches the window plane
    end

    if DISPLAY
        drawnow; % redraw the figures
    end

end

drawnow;

toc, disp(sprintf('Simulation End Time: %.2fs\n',t)); % output measured real time and simtime

%% Post Processing

figure(scene_hndl);
    hold on;
        hist_hndl=plot3(history(:,2), history(:,3), history(:,4),'-b');
    hold off;
    title(sprintf(['ILS Approach with image processing at %iHz\n',...
                   'from (x,y,z)_{NED} = [%.2f %.2f %.2f], (\\phi, \\theta, \\psi)_{deg} = [%.2f %.2f %.2f]'],...
                   CNTRL_RATE,X_NED(1:3),X_NED(4:6)*180/pi));

view([-90,0]), print(scene_hndl,'-dpng','-r300',sprintf('ILS_Approach_%i_front',i));
view([0,90]), print(scene_hndl,'-dpng','-r300',sprintf('ILS_Approach_%i_top',i));

    delete(hist_hndl);
    view(3);
```

end % end Outer Loop