SELF-ORGANIZED TRAFFIC FLOWS: A SEQUENTIAL CONFLICT RESOLUTION APPROACH

A Thesis
Presented to
The Academic Faculty

by

Troy Hand

In Partial Fulfillment
of the Requirements for the Degree
Masters in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2013

SELF-ORGANIZED TRAFFIC FLOWS: A SEQUENTIAL CONFLICT RESOLUTION APPROACH

Approved by:

Dr. Eric Feron, Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Dr. J.-P. Clarke
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Zhi-Hong Mao
Swanson School of Engineering
University of Pittsburgh

Date Approved: April 25, 2013

To my family,

Troy

ACKNOWLEDGEMENTS

It gives me great pleasure in acknowledging the never-ending support and impeccable expertise of Dr. Eric Feron, Professor of Aerospace Engineering and Advisor to my Masters Study and Research. His patience, motivation, enthusiasm, and immense knowledge proved invaluable to me. Dr. Ferons guidance steered me in the right direction while his "laissez faire" approach allowed me to pursue my own interest. Thank you, Eric, for our laid-back conversations and sharing your insight and experience with life in general during my time at Ga.Tech. I leave with a Masters but more importantly the life lessons that made it possible.

It is with immense gratitude and respect, I acknowledge Dr. Zhi-Hong Mao, Associate Professor in the Departments of Electrical and Computer Engineering and Bioengineering at the University of Pittsburgh for graciously welcoming me to Pittsburgh for a summer of research. I consider it an honor and privilege to have collaborated with such a highly valued colleague who was instrumental in helping my research materialize.

I offer my sincerest thanks to Dr. J. P. Clarke, Thesis Committee, for his relevant discussions, inspiration, and perceptive comments.

Finally, I am forever indebted to my family and friends for their understanding, patience and reassurance when it was most required. And a heartfelt thanks to Gretchen for always being available for me, through the tough classes, short deadlines, late nights, and grumpy mornings.

The guidance and backing received from all who contributed to my thesis was vital for its success. I will be forever grateful for their tireless efforts and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Intersecting flows of physical agents are ubiquitous in nature and human activities. They include road intersections, pedestrian traffic, maritime traffic and air transportation. Much research has been conducted on these intersections, where guaranteed collision avoidance among aircraft going to different destinations constitutes a cornerstone of its credibility. The efficiency and safety of these systems depend upon the synergies between the physical agents and the computational framework controlling these agents. To date, however, very little is proven about how these flows actually behave under different control mechanisms, such as those involving autonomous control. Depending on the control and intersection of these flows the traffic congestion can become overwhelming. This is a major externality in many cities around the world, which imposes a negative economic, environmental, and social impact. The cost of congestion in urban areas in the United States, in terms of delay and excess fuel, was about $90 billion in 2007 according to the 2009 FHWA Urban Mobility Report. Similarly, for air traffic the total cost of delays to U.S. economy is estimated to be around $30 billion a year [1]. Of this, 30% is due to capacity limitations in en route airspace, where flows of aircraft intersect and need to be dynamically controlled.

The major cause of congestion, in any type of setting, is conflicting or competing needs of different traffic flows. Hence, a decrease in congestion levels would be generated by any increase in the capacity of a location where space-time needs intersect. However, the control mechanism in place for that space-time element will determine the capacity of the intersection. These include traffic signs and rules for ground traffic, as well as human controller actions for air and maritime traffic.

The current use of centralized air traffic control to ensure aircraft separation is a safe option that has been proven over the years. However, increasing air traffic volume steadily increases complexity [2] and produces many drawbacks in current air traffic control techniques. The Federal Aviation Administration (FAA) and airlines have proposed the concept of "Free Flight" [3] to eliminate restrictions imposed by the current system and allow for more navigation freedom as well as direct routing. Indications of a significant increase in air traffic volume, ranging from a factor of two to three by 2025 [4] [5], pushed the Unites States' NextGen (Next Generation Air Transportation System) and Europe's SESAR (Single European Sky Air-traffic-management Research program) to begin addressing challenges of increasing air traffic volume, limitations on operational flexibility, as well as current air traffic operations [4] [6] [7].

Automation is a key element necessary to achieve the goals set by NextGen and allow the concept of Free Flight to become more viable [8–10]. A decentralized air traffic control architecture would alleviate some of the drawbacks associated with current air traffic control while laying the groundwork for Free Flight. This would require each aircraft to determine its maneuver based on information shared between aircraft, such as position and velocity supplied by the Global Positioning System (GPS). A switch from current systems to GPS would alleviate limitations associated with the ground-based navigation infrastructure. However, there would be a large transition to move from the current centralized control to a decentralized one. This transition would likely consist of a hybrid architecture of centralized control with decentralized control employing more and more automation ultimately leading to decentralized control. For either a centralized or decentralized architecture, a sequential conflict resolution algorithm would decrease the total number of conflicts, allowing human controllers to manage considerably more aircraft.

With the use of automation in a safety-critical configuration there must be guarantees of safety and efficiency of the control being utilized. Therefore, for intersecting flows, the conflict resolution schemes used need to be thoroughly studied. Along with this, as the idea of Free Flight becomes more and more possible, a need to anticipate how these flows are going to interact is pertinent.

Apart from air traffic, much research has been conducted on increasing the capacity of intersecting flows of physical agents. Considering ground traffic, traditional traffic safety infrastructure such as warning signs, traffic lights, metal railings, curbs or speed bumps may not only be often unnecessary, but can even be dangerous. In some implementations where such controls were removed, traffic accident rates were observed to fall almost by half, while traffic volumes increased by a third [11]. Under certain circumstances, roundabouts are found to be very advantageous, e.g. safety improvements and delay reductions, compared to other alternatives such as signalized intersections [12]. A study on 24 intersections in the United States reported a 39% overall decrease in crashes and a 90% decrease in collisions involving fatal or incapacitating injuries [13]. Similar arguments can be made for traffic systems in countries such as India and Vietnam, where few or no traffic control systems are utilized at intersections. In these cases, traffic flow rate is typically very high as depicted in Figure 1 [14], but at the same time virtually without any accidents. Similar characteristics also exist for pedestrian traffic in these and other countries [15] [16].

From these observations it may be noted that under certain setups such minimally controlled self-organizing flow can be safer and more efficient than centralized fully controlled traffic flows. This leads to the idea of applying some level of control in a way that mimics the behavior of these agents acting in a self-organized manner. This idea is where my research begins, as I discover different aspects of self-organized flows as well as determining a simple control architecture that can be utilized to mimic some self-organizing behavior.

Figure 1: YouTube video snapshot of Vietnam traffic.

1.1 Objectives of the thesis

There are two main objectives, firstly, is to provide analysis of self-organized traffic flows occurring in current traffic scenarios. Secondly, is generating an analytical base for understanding conflict resolution and its limits when sequential control is utilized in an aircraft flow where vehicles are not allowed to stop to avoid conflicts.

- How do intersecting flows of agents interact in a self-organized environment?

- Can sequential conflict resolution schemes mimic any behavior noticed in that of self-organized systems?

- What are the limits of sequential conflict resolution maneuvers?

1.2 Contributions

The main contributions of this thesis are:

- Development of an automated tool to extract properties of moving agents from a video file.

- Analysis of self-organized traffic flows of ground based vehicles

- Stability analysis of sequential conflict resolution schemes involving arbitrary encounter geometries of arbitrary flow thickness.

1.3 Thesis outline

The organization of this thesis starts with a review of some of the related research. Methods used to build a video analysis tool with the purpose of extracting information about objects in motion are then discussed. Also, I provide an example and walk-through of how to use the tool I created. Next I present some findings while using the tool which include the types of videos that work well and which ones do not. Remedies are also provided for some issues that may show up. I then present some data visualization tools I created for analysis purposes, this includes the analysis of self-organized traffic flows. The thesis switches gears somewhat at this point, as I expand on previous research of sequential control of intersecting aircraft flows by examining arbitrary flow geometries of arbitrary flow thickness. Simulations are conducted to determine how flow thickness effects the aircraft flow during conflict resolution. Stability is proven to be achieved with analytical solutions derived for bounds on the conflict resolution maneuvers. Lastly, comparisons of the simulation results and the analysis of self-organized flows are discussed.

CHAPTER II

RELATED RESEARCH IN EXISTING LITERATURE

There has been a good bit of research over the years on crowd dynamics of pedestrian traffic with the aim of analyzing the behavior of agents within these flows. The authors of [17] develop a "social force model" to successfully describe many observed collective effects of pedestrian behavior. Experiments are conducted in [18] with students to observe video of intersections, bottlenecks, and counterflow. They further develop the model to realistically reproduce the phenomena in a simulation environment, and observe the phenomenon of stripe formation with this model. The stripes or corridors are also reproduced in this research as well as research in which vehicles are under a given conflict avoidance rule as described by [19, 20]. The authors of [18] go on to provide improved design solutions for pedestrian facilities. The authors of [21] discuss the effect of groups (i.e. family and friends) on pedestrian traffic, and show how group size effects walking speed and moreover the average patterns of organization for each size of group under different crowd densities. The motivation of most of the research on crowd dynamics is for use in prediction models for the design of urban infrastructures, traffic management of crowd safety during mass events or evacuation processes. The authors of [22] video pilgrimage related crowds and use video analysis techniques to extract useful information. They were mostly interested in the density of the crowd since the safety in crowds is usually determined by the maximum occurring density. The authors also give some measures to improve crowd safety.

Although the research of [17] is conducted on pedestrian traffic there are still many similarities with this research. For instance, this research aims at showing the

use of sequential conflict resolution scheme can mimic that of self-organizing traffic flows. Methodologically, I specifically differentiate this research on traffic intersections from pedestrian traffic related work of [17] via the necessity to obtain proofs of the stability of the conflict resolution scheme utilized. Theoretical validity and performance bounds before system deployment are required in many of automated applications as they are safety-critical configurations.

The use of video analysis tools are becoming more and more prevalent not only in academia, but in the Wide World of Sports as well. Many sports enthusiast are using these tools to analyze their swing, pitch, bowling, shot, etc. As far as the techniques behind the software is concerned, most all video analysis research has the objective to gain some information about an object in motion within the video frame. And all of the motion extraction techniques involve comparing a current frame of the video with a previous one to estimate the motion between the two. Much research has been done in this field which has produced many available softwares that can extract motion from video. Some of the freely available software includes, DataPoint [23], SwisTrack [24], and ObjectTracker [25]. Of these, ObjectTracker is the most advanced as it offers tracking with use of a particle filter. However, all of these softwares fail at some aspect needed for this research. DataPoint software requires the user to click each object they want to track from frame to frame; a very time consuming manual motion extraction process. SwisTrack developed a modular system which made customization possible (i.e. apply different techniques to extract motion), it also has the ability to retrieve data via TCP/IP. However, the user interface of SwisTrack is somewhat confusing and sometimes crashes. Also, DataPoint and SwisTrack only operate on AVI file types where ObjectTracker can handle a few others. ObjectTracker, while being the most advanced, was only able to analyze completely stable video with slowly moving objects. There was also the inability to save the data for further analysis, or make modifications to obtain other properties of the objects. All of these

programs mentioned were designed to track objects and not necessarily retrieve any other information about the objects. As I tried different software available, I came to the conclusion that I must create my own tool which will be user friendly and easily modifiable while extracting as much information about the objects as possible.

Along with the video analysis there is a necessity for good data visualization to enable the user to see what is happening. As larger and larger data sets are being used to capture wanted aspects, huge files of endless data are being produced and the idea of data visualization becomes a necessity. This idea has always been present in the media and in academia, for use of educating the public in a simple understandable way. One example of developing a data visualization tool for understanding flows of aircraft comes from the authors of [26], who develop a data visualization tool to show the trajectories of thousands of aircraft over a sector of airspace by use of a large archive of aircraft data. My idea is similar in the fact that I need to see what is going on with all of the data I extract from the video. I can also apply further analysis to determine different aspects about the flows such as speed, heading, trajectories, etc.

I plan to apply what I learn from the video analysis to determine if intersecting aircraft flows under sequential control exhibit any of the observed behaviors of objects in self-organized traffic flows. In addition, I want to provide a strong analytical base for understanding conflict resolution and its limits when sequential control is utilized in an aircraft flow where vehicles are not allowed to stop in order to avoid conflicts. The authors of [27] have built an analytical base for the probability density function using a "closed-loop" approach, taking into consideration the conflict resolution algorithm. The probability density function shows a dissymmetry between the positive and negative lateral deviation, which is also shown in simulations in this thesis as well as in the video analysis. In other words, the aircraft/objects tend to move toward the other flow to prevent a conflict [27]. The authors of [19] and [20] derive analytical solutions of the displacement bounds of conflict resolutions in different aircraft flow

models stating that a stable conflict resolution scheme must have a bounded conflict resolution maneuver. The mathematical derivation of these results involved rigorous non-trivial analysis indicating the challenges and the need for more intensive research for more complicated traffic configurations.

CHAPTER III

MOTION EXTRACTION AND VISUALIZATION TOOL
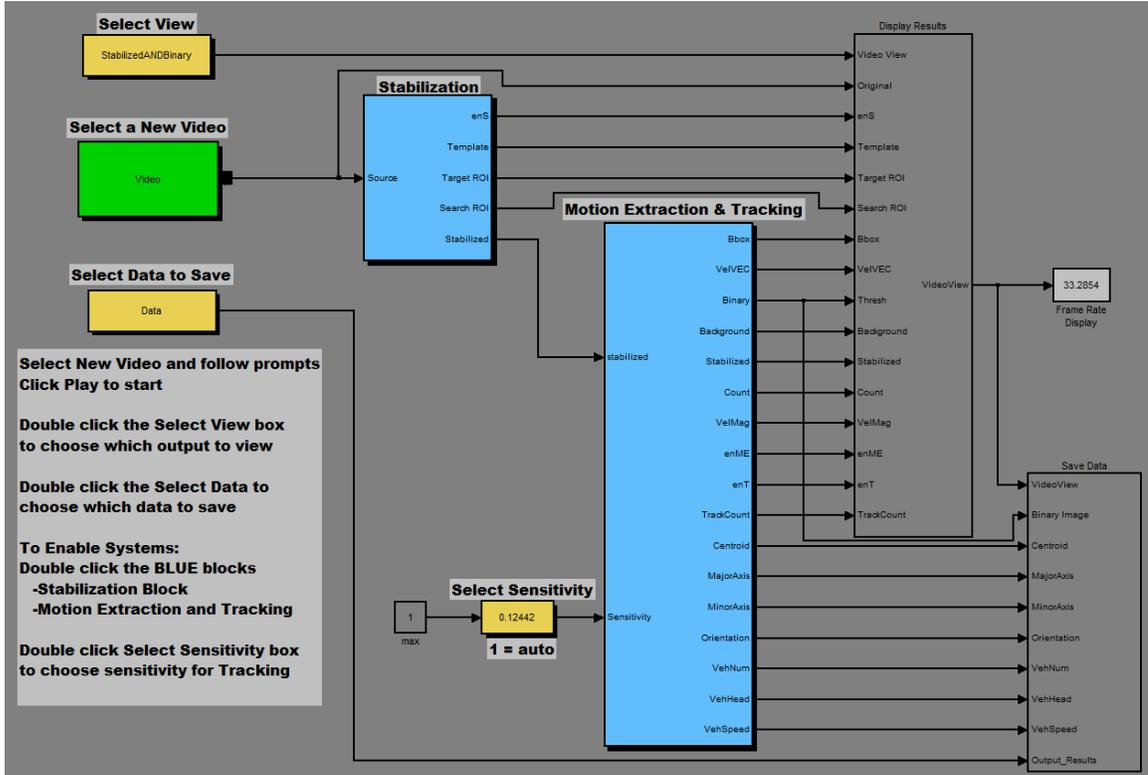
Figure 2: User interface of the motion extraction tool

3.1   Tool description

The goal of this tool is to obtain properties of objects in motion in recorded video. In other words, automatically extract useful information about different flows of agents. As mentioned previously, there are freely available tools such as Swistrack and ObjectTracker which aim at extracting motion from a video and tracking those objects in motion. However, they are not easily modifiable or adjustable for the user and are not always stable, and are also specific on file types supported. A user friendly
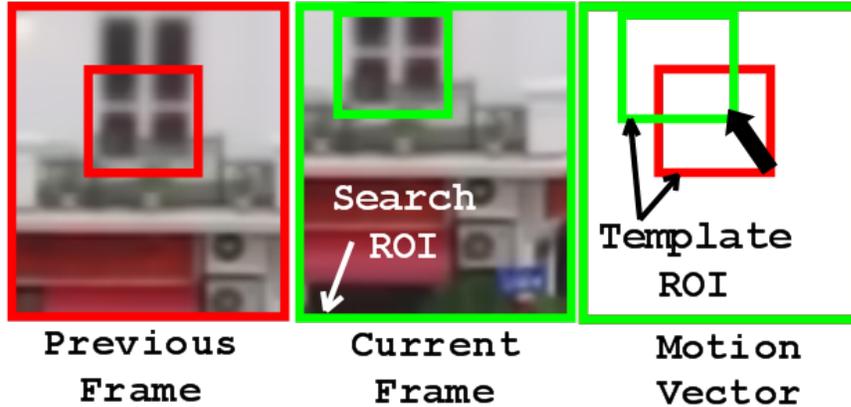
Figure 3: Example of how the motion vector is estimated

interface is needed while at the same time allowing for easy modification by the user if desired. I decided to use the Simulink interface to build a tool because it is an intuitive environment that is easily modifiable, with some built in functions for vision based applications. I will go though each sub-system within this tool and discuss the purpose of them. Figure 2 illustrates the Simulink interface of the tool.

3.1.1 Stabilization

This is a necessary portion of the tool if the goal is to estimate position, speed and heading of the extracted motion. For example, if we do not know the motion of the camera it would be impractical to estimate the location of an object as we wouldn't be able to discern if the object or the camera was in motion. Template matching is used to determine the motion vector to translate each video frame for stabilization.

The user will be prompted to specify some template region of interest (ROI) to keep stationary. The ROI would need to be a portion of the video without motion (i.e. a building's window). A search boarder is also given by the user to specify the area around and including the template ROI to search during template matching. The template matching minimizes the difference of the pixel values within the ROI of the previous frame with pixels within the search ROI of the current frame, see Figure 3. A motion vector is estimated by comparing the locations of the previous ROI and the

11

result of the template matching. The motion vector is in turn used to translate the video. The template and search ROI are then updated for the new translated video.

Now, with a stabilized video, the pixel locations are stationary and therefore better estimates of the speed and heading of the motion of the extracted objects can be determined. This also allows for creation of a background image for motion extraction discussed in the next subsection.

3.1.2 Motion extraction

All motion extraction techniques compare a current frame with a previous frame or background image. The purpose being to find what pixel values change from one frame to the next and hence see what is moving from frame to frame. Many different methods are available in the Simulink environment, including block matching, optical flow, and background subtraction. The basics of block matching and optical flow are similar in the fact that they estimate the motion of the pixels that change from a previous frame. This results in mostly the edges of the objects to be extracted, more specifically, the edges (Figure 7a) that are in the front and back of the direction of motion. With this method you do not see the full objects size/shape, some information that may be of importance in certain situations (i.e. extracting motion for use in path planning for robots, where size/shape of the object is important). This method is convenient in the fact that you do not have to generate/have a background image, however, the ability to detect objects in a blurry video is more difficult. Figure 7a shows the extracted motion by use of optical flow as the white portions of the figure.

I chose to develop my motion extraction around the background subtraction method which gives the ability to see the full object, which are objects that are different from the background. The downside here is that we now need a background image. This is accomplished by firstly taking the mean of all of the previous frames of a stable video to generate the background image. Then both the current video frame

(a) Initial frame

(b) Mean of a few frames

(c) Generated background image

Figure 4: Generating the background image

and the background image are converted to an intensity image, a linear distribution where black=0 and white=1. Finally, the background image is subtracted from the current frame and the absolute value is applied resulting in the darker portion being the background and lighter portion being the motion. In other words, the objects in motion will show up as white blobs on a black background. The use of the absolute difference of the current frame and the background results in the ability to see objects lighter and darker than the background. An example of what the generation of the background would look like is depicted in Figure 4.

A stable video is critical in the creation of a background image as you want the background to remain in the same location throughout the video. For some very shaky videos the stabilization may fail to stabilize the video if the search ROI is too small, therefore, the mean of the previous frames may no longer be a good representation

(a) Example of typical extent   (b) Example of a very small extent
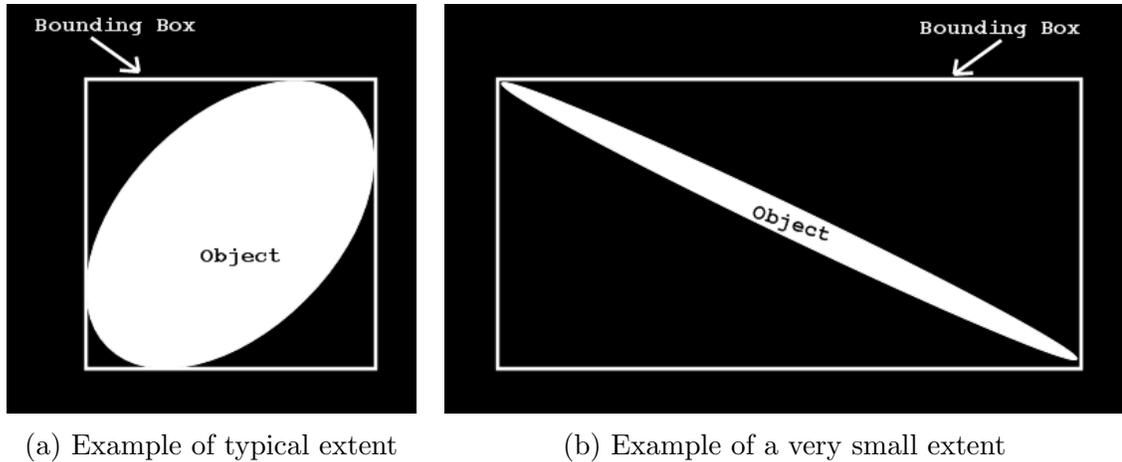
Figure 5: Examples of blob extents

of the background image. This is also likely to occur when the intensity of the image changes drastically. For example, a cloudy day with the sun shining through and being obstructed again or when transitioning from daytime to nighttime. In this event it is a good idea to reset the mean and begin generating a new background image. This could be accomplished manually by specifying when to reset the background. I develop an automatic background subtraction by analyzing these blobs of motion and examining the extent of each one. Simulink has a built in blob analysis block which I will discuss later in this subsection. The extent of a blob is the ratio of the blob area to the area of the bounding box around the blob. Examples of blob extents are shown in Figure 5. This is useful to detect edges that show up if the stabilization cannot handle the amount/type of shaking. When the minimum of the blob extents becomes less than a specified value the background generation will be reset.

Using the background subtraction method the current frame is reduced to an intensity image with higher intensity values representing the motion. Other operations can be applied to this image to result in a more accurate motion extraction. The tool I developed gives options to apply smoothing and morphological operations. I built in median filters to smooth the image; these filters take the median of a neighborhood of the intensity image resulting in removing noise while preserving edges. The user has

the option to choose how many smoothing operations to apply. I have also embedded multiple dilation and erosion morphological operations. The dilation operation (Figure 6b) adds values to a given neighborhood of pixels while the erosion removes values from a given neighborhood of pixels. For example, if the motion extraction is missing some areas on an object in motion, the dilation operation would be beneficial in the fact that it could increase the size of the blob to incorporate the whole object. This becomes dangerous when dealing with many objects in close proximity as they can start to become one blob. This is when the erosions are necessary to remove pixel values from a given neighborhood with the goal of separating the individual blobs (Figure 6c). The erosion operations are also convenient when dealing with a very noisy intensity image. The tedious work is determining the order of the operations to obtain the best result. The Simulink interface allows the user to select the order of the operations. The neighborhoods of the morphological and smoothing operations are also easily editable. Addition of other morphological operations such as opening, closing, top-hat, and bottom-hat is just a matter of adding a block from the Simulink's vision library. The Simulink tool can also include operations from the OpenCV library via the MEX interface. The effect of these operations are easier to view in a binary image discussed in the next paragraph.

The newly generated intensity image is converted to a binary image by use of applying a threshold. This threshold is the sensitivity of the motion detected. Smaller values of the threshold result in a more sensitive motion extracted where larger values of the threshold result in less sensitivity. The user can define the threshold with a slider gain in the Simulink interface to determine the amount of sensitivity they desire. Alternatively, they can choose to use the built in AutoThreshold which determines the threshold by splitting the histogram of the input image such that the variance for each of the pixel groups is minimized. After completing the motion extraction process the image is reduced to a binary image where a motion is represented as

(a) Initial binary image   (b) Dilation only

(c) Erosion only   (d) Final binary image after smoothing
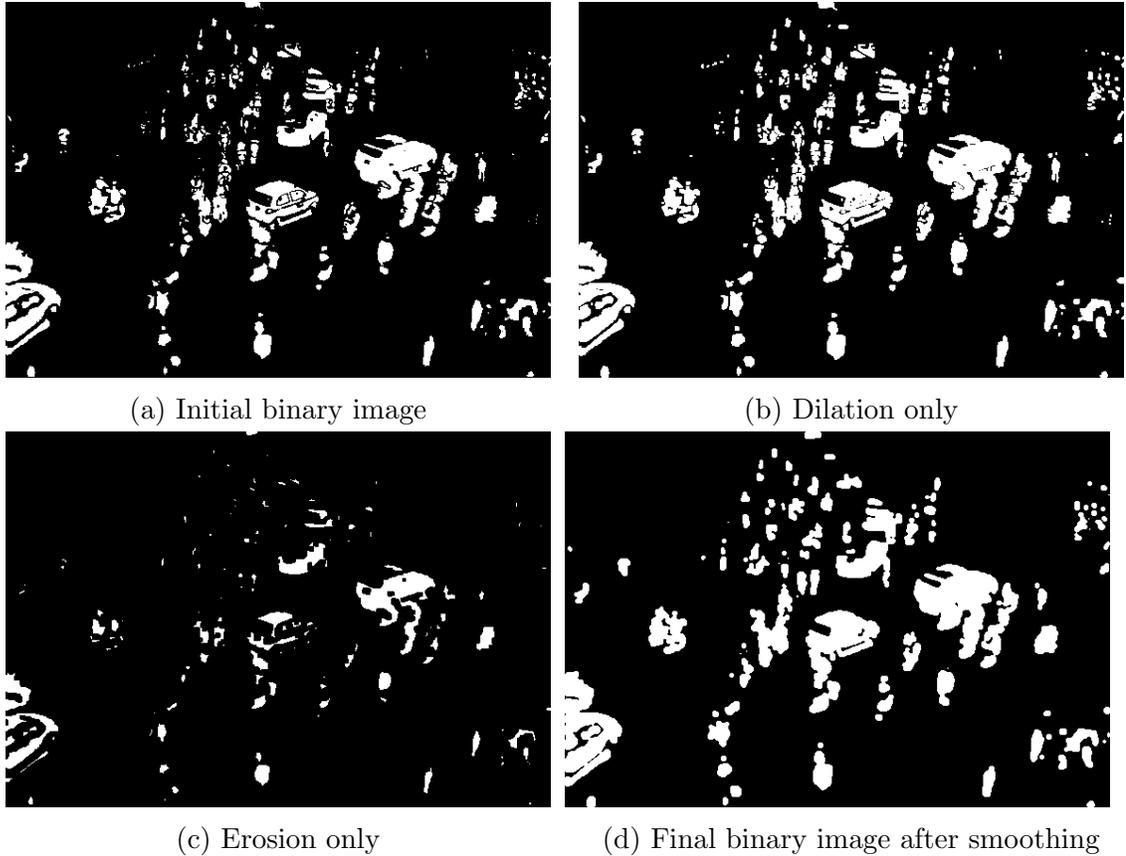
Figure 6: Example of morphological operations and resulting binary image after all operations and smoothing are preformed (white=object in motion)

a 1 and the background is represented as a 0. Figure 6 gives an example of these different operations and the resulting binary image. A comparison of the optical flow method and the background subtraction method is shown in Figure 7 by overlaying the motion (i.e. white portion) on the frame of the movie.

The binary image can now be processed by Simulinks Blob Analysis block to determine blob properties and therefore extract information about the objects in motion (i.e. the blobs). Properties that can be output from the blob analysis block are: area, centroid, bounding box, major and minor axis, orientation, extent, and a few other parameters. The centroid data is an M-by-2 matrix of centroid coordinates, where M represents the number of blobs. The Bounding Box data is an M-by-4 matrix of $[x, y, width, height]$ bounding box coordinates, where M represents the number of

(a) Using optical flow   (b) Using background subtraction

Figure 7: Example of motion extraction

blobs and $[x, y]$ represents the upper left corner of the bounding box. The Major and Minor axis data is a vector that represents the lengths of minor and major axis of ellipses that represent the blobs. The orientation is a vector that represents the angles between the major axes of the ellipses and the x-axis. The extent is a vector representing the results of dividing the areas of the blobs by the area of their bounding boxes. These blob properties can be used to track these objects through the video, frame by frame. The next subsection discusses the tracking algorithm used in the tool.

3.1.3 Tracking

This tool also includes a basic tracking algorithm for the purpose of visualizing the velocity vectors of individual objects in motion. An off-line tracking algorithm will be discussed later in the paper, a tracking algorithm that takes the depth of the figure into account. The tracking algorithm I built into this tool finds the closest centroid from the current video frame to that of the previous video frame. It then detects if any other centroid points to the same previous centroid and chooses the one with the least change in heading. The tracking algorithm (Algorithm 1) is provided. The *Speed* and *Heading* is saved for each object that is tracked. The variable, *Number*, is a reference that records what object number each centroid corresponds to.

17

Algorithm 1 Tracking Algorithm

---

for all frames: $f > 0$ **do**

    Take all centroids at the current frame (C_f) and at the previous frame (C_{f-1})

    for each Centroid $C_f(i)$ **do**

        Compute the minimum distance between all centroids C_{f-1}:

        $D_{min} = \min(C_f(i) - C_{f-1}(:))$

        Find index j such that $D_{min} = (C_f(i) - C_{f-1}(j))$

        if D_{min} is empty OR $D_{min} > D_{max}$ **then**

            $Count = Count + 1$

            $Number_f(i) = Count$

        else

            $Number_f(i) = Number_{f-1}(j)$

            $Speed_f(Number_f(i)) = \text{distance}(C_f(i), C_{f-1}(j))$

            $Heading_f(Number_f(i)) = \text{azimuth}(C_f(i), C_{f-1}(j))$

            $\Delta_{heading} = Heading_f(Number_f(i)) - Heading_{f-1}(Number_f(j))$

            for all previous centroids $k = 1{:}i\text{-}1$ **do**

                if $Number_f(k) = Number_f(i)$ **then**

                    $\Delta_{headingK} = Heading_f(Number_f(k)) - Heading_{f-1}(Number_f(j))$

                    if $\Delta_{heading} < \Delta_{headingK}$ **then**

                        $Count = Count + 1$

                        $Number_f(k) = Count$

                        $Speed_f(Number_f(k)) = \text{distance}(C_f(k), C_{f-1}(j))$

                        $Heading_f(Number_f(k)) = \text{azimuth}(C_f(k), C_{f-1}(j))$

                    else

                        $Count = Count + 1$

                        $Speed_f(Count) = Speed_f(Number_f(i))$

                        $Heading_f(Count) = Heading_f(Number_f(i))$

                        $Number_f(i) = Count$

                  end if

                end if

            end for

        end if

    end for

end for
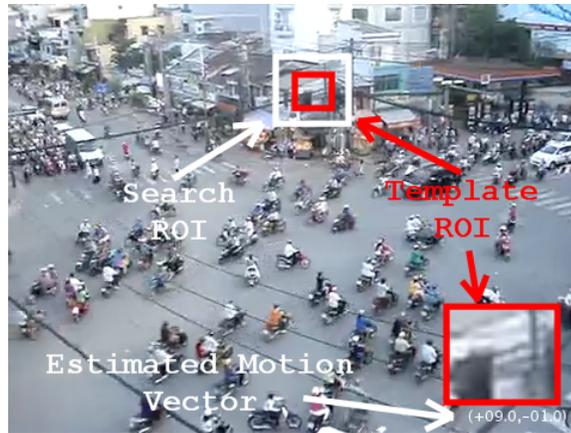
Figure 8: Stabilization view of the output video

With the tracking step complete, a variety of data is now available for analysis. It is also useful to visualize this data during the motion extraction process to determine if good results are obtained. The visualizations provided by the tool are discussed in the next subsection.

3.1.4   Visualization

This tool allows the user to specify which output they would like to view for feedback while adjusting parameters. The available views are: Stabilization, Background, Binary, Tracking, and stabilized with binary overlay. Each view provides different information about each section of this tool.

The stabilization view (Figure 8) shows the template ROI and the search ROI on top of the main video. The template ROI is enlarged in the bottom right of the view with the estimated motion vector below. I am showing the stabilization view of a different video file that has more shake so the estimated motion vector estimate will be shown on the image.

The background view simply displays the background that is generated during the motion extraction process (Figure 4c). The Binary view displays the binary image that is created during the motion extraction process (Figure 6d). This binary view is also displayed as an overlay on the stabilized video to show how well the motion is

19

Figure 9: Tracking view of the output video

being detected (Figure 7b).

The tracking view (Figure9) displays the bounding boxes that were determined from the blob analysis of the binary image. These bounding boxes show the vehicles that are being passed to the tracking algorithm. There is also a velocity vector shown of all the vehicles that are being tracked. The number of objects detected is displayed in the top left corner along with the number of objects being tracked as well as the average speed (pixels/frame) of those being tracked.

After the user is satisfied with the results, they have the option to save the data. The next subsection discusses the different data that can be saved.

3.1.5 Save data

The user has the option to save the data being extracted with this tool. The user may select a checkbox to save the blob properties, tracking data, and video output separately. The video output will save what view the user has chosen. Table 1 shows the different file names and properties of the data the user can select to save.

Once again, a Simulink interface enables the user to save any other signals very simply. For other data signals the user may want to save, they simply connect the signal they want to save to a "to file" block with a data type of Timeseries. This data can then be used in further analysis.

File Name	Data	Size	Comments
CentroidDATA.mat	Centroid.Data	2-by-M-by-N	dims $= [x,y]^T$
MajorAxisDATA.mat	MajorAxis.Data	M-by-N	M = objects
MinorAxisDATA.mat	MinorAxis.Data	M-by-N	N = frames
OrientationDATA.mat	Orientation.Data	M-by-N	
VehNumDATA.mat	VehNum.Data	M-by-N	(i.e. object 1, 2,3)
VehHeadDATA.mat	VehHead.Data	M-by-N	
VehSpeedDATA.mat	VehSpeed.Data	2-by-M-by-N	dims $= [V_x, V_y]^T$
Output.avi	Output.avi	Input video size	

Table 1: Save Data File Structure

3.2 Tutorial

This section is intended to provide a walk-through on a typical use of the tool I developed.

1. Select a new video by double clicking the green box labeled Video:

- Prompts will appear on screen.

    - Locate the video file you wish to use:
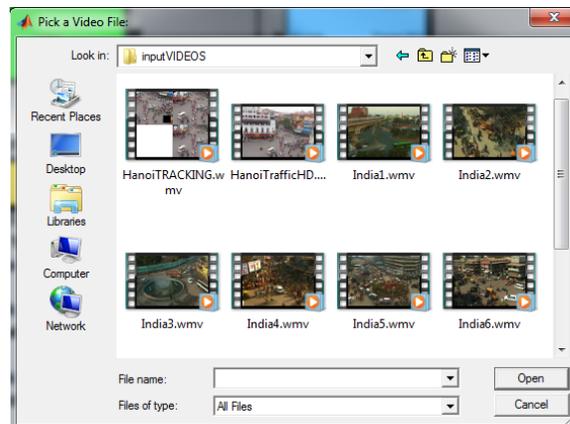
    - A window will pop up with a frame of your video:
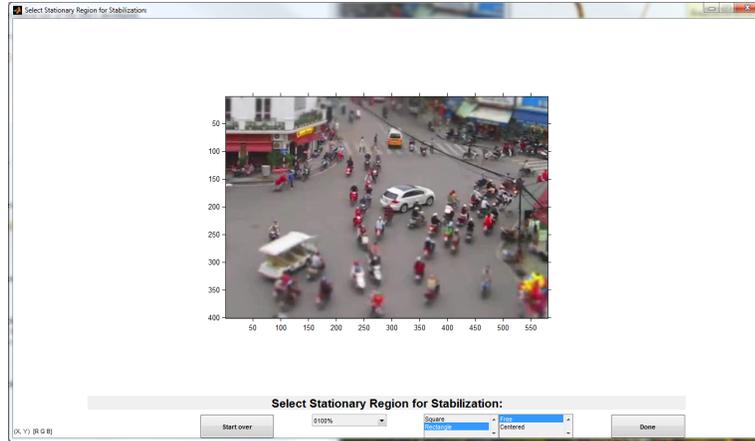
* Select the template ROI (a region without motion)

* Select the largest object in motion (for blob analysis)

* Select the smallest object in motion (for blob analysis)

* Select the largest distance between two frames (for tracking algorithm max speed)

2. Select desired options (these options are editable during execution)

- Double click Stabilization block for options:
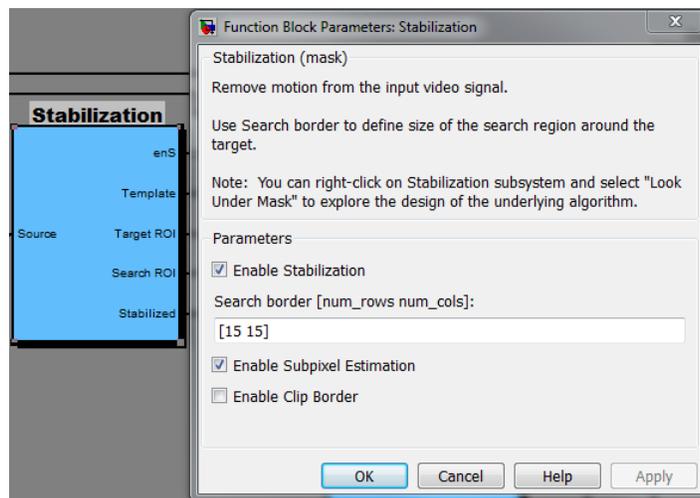
- Double click Motion Extraction and Tracking block for options:
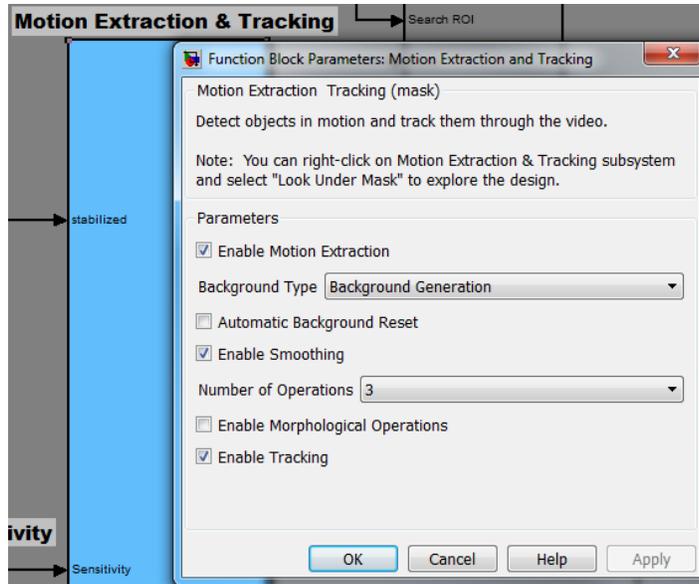
- Double click Select View to change what view you prefer:
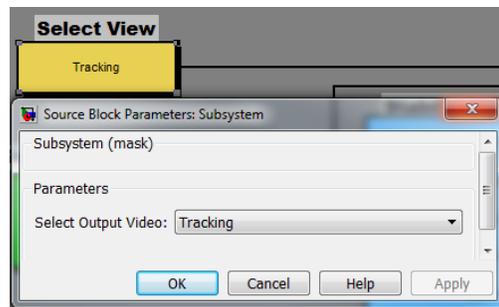
- Double click Select Data to Save block to set which data you wish to save:
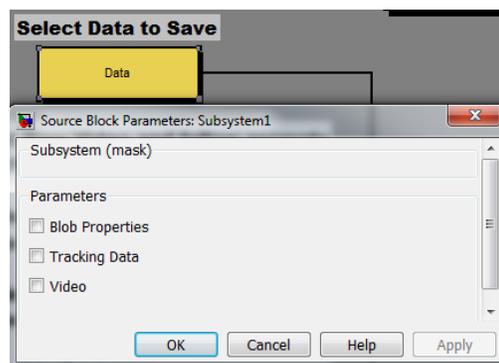
- Double click Select Sensitivity block to change the sensitivity of the motion extraction:
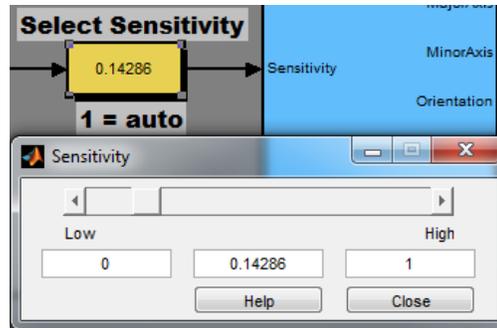
3. Start Simulation (the play button)

- can repeat step 2 during execution

  - changing views and parameters until satisfied.

4. Simulation will end when the video length is exhausted or by pressing stop

This concludes the tutorial; the next chapter discusses different uses for this tool as well as providing a range of analysis tools to use with the extracted data. Stress test results are provided with the purpose of determining the strengths and weaknesses of this tool.

3.3 Stress test

I downloaded numerous videos from multiple online sources [14–16, 28–32] with moving agents in which to test my tool to determine what it could and could not withstand. The following are the difficulties I discovered while extensively testing this tool.

3.3.1 Shaky videos

Most of the issues come from very shaky videos; more specifically videos with rotational shake. If the shaking is translational only, the stabilization handles this well as long as the template ROI remains within the video for all time. For translational shaky videos in which the original template ROI leaves the video, the user can turn

the stabilization off and wait until another stationary object falls within the ROI to turn stabilization back on; the background generation would have to be reset as well. To be able to handle rotational shake the stabilization would need to include an estimation of a geometric transform from one frame to the next. Only translational motion is estimated with the current stabilization technique and therefore cannot estimate the rotational motion. Simulink has a built in geometric transformation estimator that could be used if you include specific points of interest within the template ROI and the search ROI. However, automating the point generation is a tedious process as it is difficult to ensure the same points will be generated in each ROI.

3.3.2 Stable videos

With no surprise, videos that are already fairly stable produce better results than those that are somewhat shaky. Most of the videos extract motion fine with the AutoThreshold, however those videos with a low dynamic range between the object and the background must have a manual Threshold set with the sensitivity slider gain to produce good results. Another issue to keep in mind is objects that remain motionless for too long will become part of the background that is generated. This is observed when looking at videos that include traffic lights where vehicles must stop for a period of time. This can be overcome by first preforming the background generation for the entire length of the video, then subtracting the current frame from the background image like before. Objects that are very small (area < 10 pixels^2) are complex to track as the area fluctuations are relatively too extreme from frame to frame and the object is not detected. I ascertained objects with area around 30 pixels^2 had no issues with motion extraction and tracking. For very small objects it would be necessary to apply a couple of dilation operations on the figure to ensure the motion is captured, however, this will skew the area output to be larger than detected.

3.3.3 Optimal videos

This tool is optimized for videos not having very much rotational shake, in which a stationary object remains within the video frame for all time. It is optimized to extract objects of area greater than and including 30 pixels^2. However, with a little tweaking and adjustments using the Simulink environment, good results can be obtained from a wide variety of videos. This tool accepts the following video file name extensions: .avi, .mpeg, .mp2, .mpg, .mp4 (including H.264 encoded), .m4v (including H.264 encoded), .wmv, .asf, .asx, and any format supported by Microsoft DirectShow 9.0 or higher.

3.4 Other useful applications

Many of the examples shown are for analyzing self-organized flow of vehicles. However, there are many other useful applications in which to utilize the data obtained by this tool. For one, it could be used in the same manner to capture how objects move in an environment with the purpose of generating realistic dynamic obstacles for path planning problems. For example, think about the vehicles as obstacles and planning a path for a robot to navigate through this intersection via this visual means of determining obstacle location (Figure 10).

Figure 10: Example of usefulness in path planning
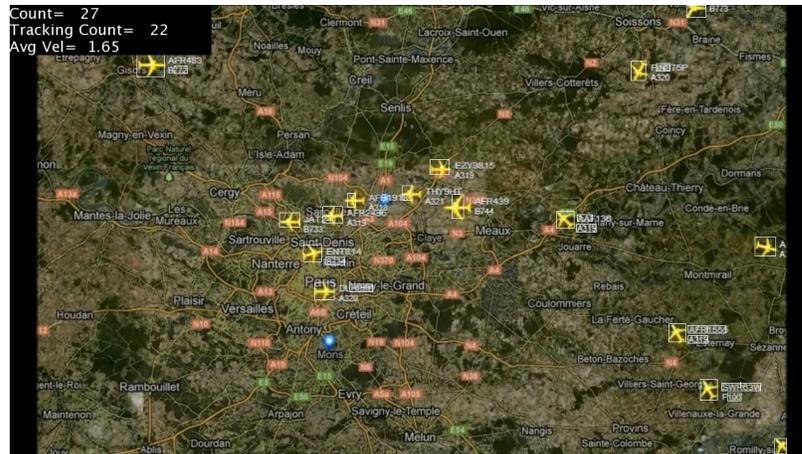
Figure 11: Example of use with air traffic video
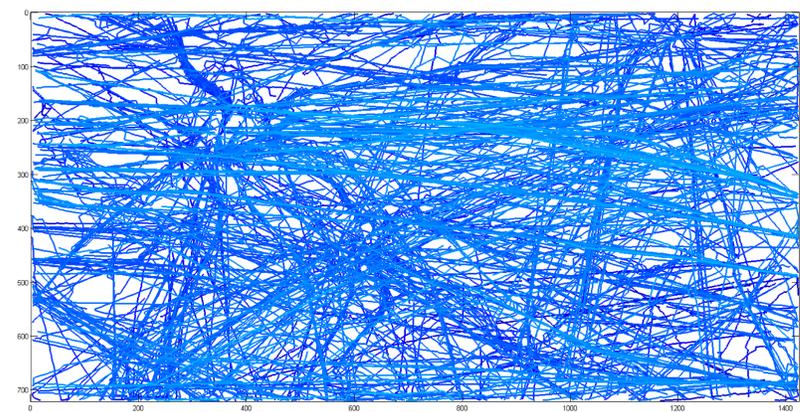
Figure 12: Extracting trajectories from FlightRadar24.com

One could also set up a counter with the motion extraction tool to automatically count how many agents pass through a specific location. For example this could be used on FlightRadar24.com video to determine how many aircraft take off and land at an airport. It could also show common routes taken to the airport or routes taken to avoid bad weather. Figure 11 shows the tracking view of aircraft from a video of FlightRadar24.com from the Simulink tool I created. Figure 12 is an example of the trajectories extracted from a video of flights over Cleavland.

There are many other different useful applications for this tool from analyzing a swing of the bat, racket, or golf club to studying the behavior of different animals.

27

3.5 *Data analysis*

I have also developed some scripts to analyze the data extracted from the tool. These include: heading distribution, trajectory plots, speed profiles, speed distribution, pixel to world coordinate transformation, tracking algorithm (world coordinates). I will show the analysis of a traffic intersection in Hanoi, Vietnam during the discussion.

3.5.1 Coordinate transformation

This coordinate transformation is based on the assumption that the user can provide some reference points in the image that correspond to latitude and longitude coordinates. This is done by finding feature point in the video and reference to world coordinates (i.e. use Google Earth to find same feature point and obtain the latitude and longitude of the feature point). Data should be saved in an M-by-4 matrix of M feature points arranged as [latitude, longitude, Xpixel, Ypixel] keeping in mind that the pixel coordinate system has a reversed y-axis (i.e. the origin is in the top left corner of the image and the positive y-direction is down). Figure 13 is given as an example of how this would work. Figure 13c illustrates a projective view from pixel coordinates to world coordinates. However, just a projective transformation will not be enough to get good estimates for speed. Therefore, a scattered data interpolant is generated.

The M feature points are used to create two scattered data interpolants; one for the latitude and one for the longitude. The Matlab function TriScatteredInterp is used with the pixel coordinates and the latitude and longitude to create the two different interpolants respectively. With these interpolants the latitude and longitude can be estimated for any point within the convex set of feature points. The natural neighbor interpolation method is used in the generation of the interpolants. I use this interpolation method because my reference points match well with the Voronoi tessellation method. This has advantages over simpler methods of interpolation,
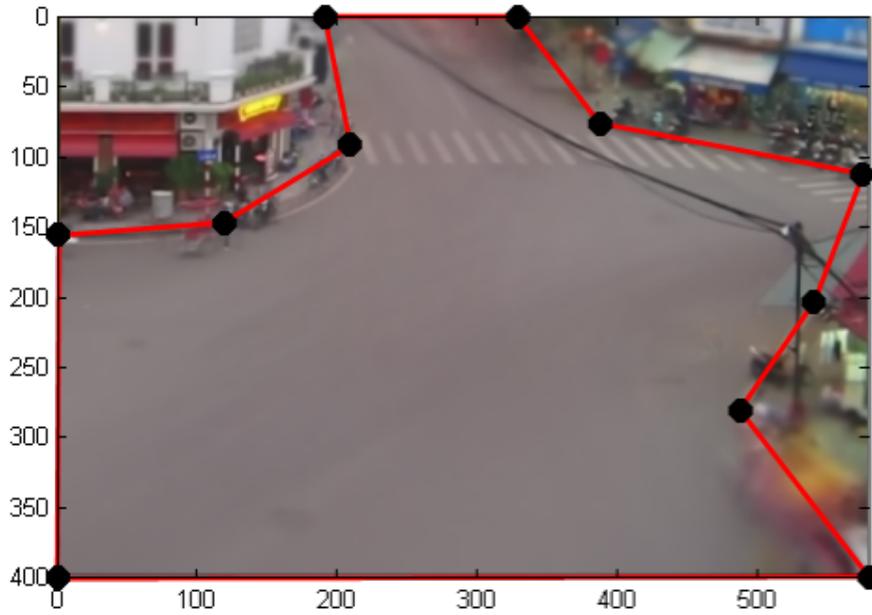
(a) Pixel coordinates

(b) World coordinates

(c) Pixel to World transformation

Figure 13: Reference points chosen comparing video image with Google Earth image

such as nearest neighbor, in that it provides a more smooth approximation to the underlying "true" function. This should result in world coordinates above the center of the image being further away from the center of the image than world coordinates below the center of the image. The interpolation is validated in Figure 14 by plotting the world distance from the center of the image.
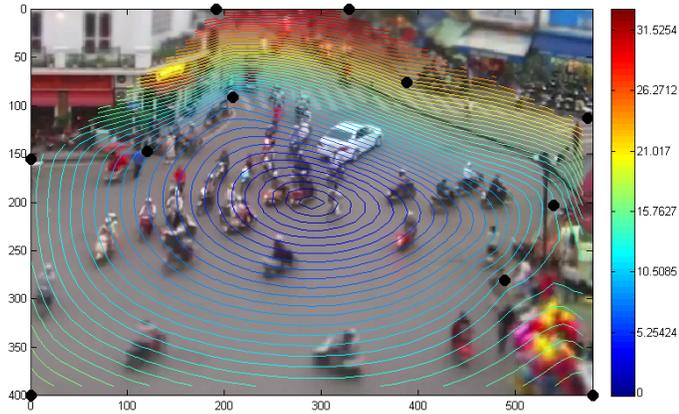
Figure 14: Verification of interpolation. Color represents distance from center of the image in World coordinates (meters)

3.5.2 Tracking algorithm

This tracking algorithm is applied to the extracted centroid data from the Simulink tool. This is of the same form as the one I embedded into the Simulink tool mentioned before, except for the fact that this one will use the world coordinate system to obtain the results. Therefore, use of depth of the video will be accounted for to select the best vehicle from frame to frame. The user must specify the centroid data to load; a start and end frame to complete the tracking over; as well as reference points necessary for coordinate transformation. After the tracking algorithm is complete the user has the option to generate trajectory plots, speed profiles, and heading distributions in the world or pixel coordinate frame.

3.5.3   Data visualization

The trajectories of individual tracked objects can be plotted using this script. The user must specify if they would like the position and speed to be given in world or pixel coordinates for plotting. If they choose pixel coordinates; the trajectories will be plotted on top of a frame of the video used in the Simulink tool. By choosing world coordinates this generates a birds-eye view of the video assuming reference points were supplied to the tracking algorithm. Figure 15 shows the different types of plots that can be obtained from the analysis. All of the subplots in Figure 15 are plotting all of the extracted trajectories for the given video, though you can specify which ones you would like to plot. Notice how the objects appear to be moving slower near the top of the image in Figure 15a, but are corrected in Figure 15b after the coordinate transformation. You can see in Figure 15c and 15d how the trajectories would look from a birds-eye view in both pixel speed (pixel/s) and world speed (m/s) respectively.

The heading distribution is another plot that can be generated to determine the direction of the main flows of objects. Figure 16 shows the heading distributions for the objects in pixel coordinates and in world coordinates. Notice, it is very difficult to distinguish two separate flows in the pixel coordinate system (Figure 16a), but very apparent in the world coordinate system (Figure 16b).

The images in Figure 17 show how the coordinate transformation can supply us with the necessary information to be able to make comparisons with the speed distribution and object density over time. From examining Figure 17c and 17d for the world coordinate frame, it is determine that the location of the highest density of objects over time corresponds with the area with the slower speed as expected. One thing to note is that the location is not the center of the intersection, but actually just before the intersection. In the pixel coordinate system (Figure 17a and 17b) this is not the case; because of the geometry of the camera, it seems as if there is a higher

(a) Pixel speed in Pixel coordinates

(b) World speed in Pixel coordinates

(c) Pixel speed in World coordinates

(d) World speed in World coordinates
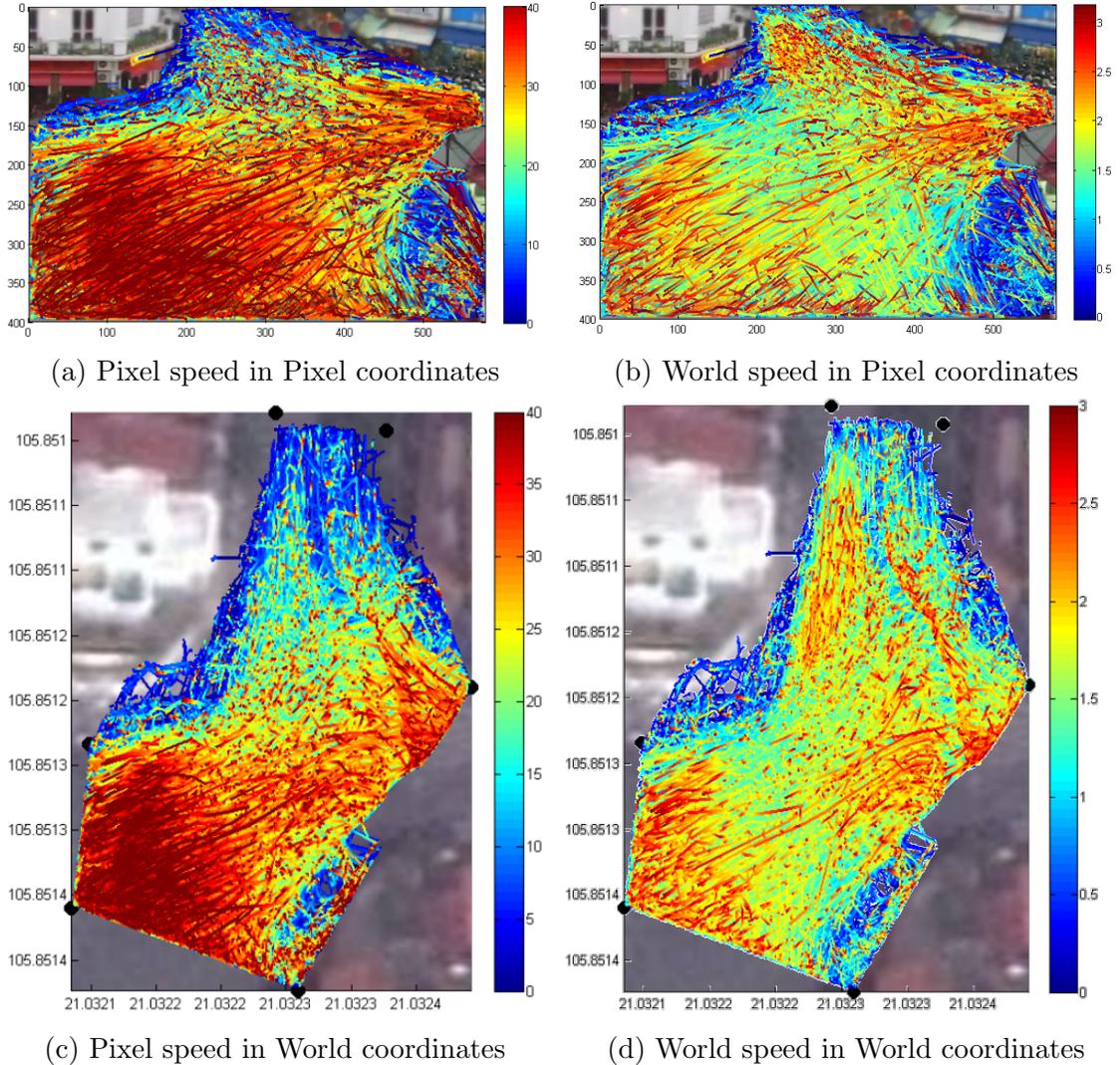
Figure 15: Trajectories of vehicles with colormap representing speed

density of objects in the upper portion of the video, and also the objects are traveling slower. The high density area in the bottom right of both the world (Figure 17d) and pixel (Figure 17b) coordinate plots are due to a vendor selling balloons and they are constantly floating around in motion.

3.5.4 Observations

The results I have previously shown are a self-organized traffic flow in Hanoi, Vietnam. The video was chosen because it involves flows of vehicles entering an intersection without any aid instructing them where or when to go. They are sending visual and

(a) Pixel coordinates   (b) World coordinates
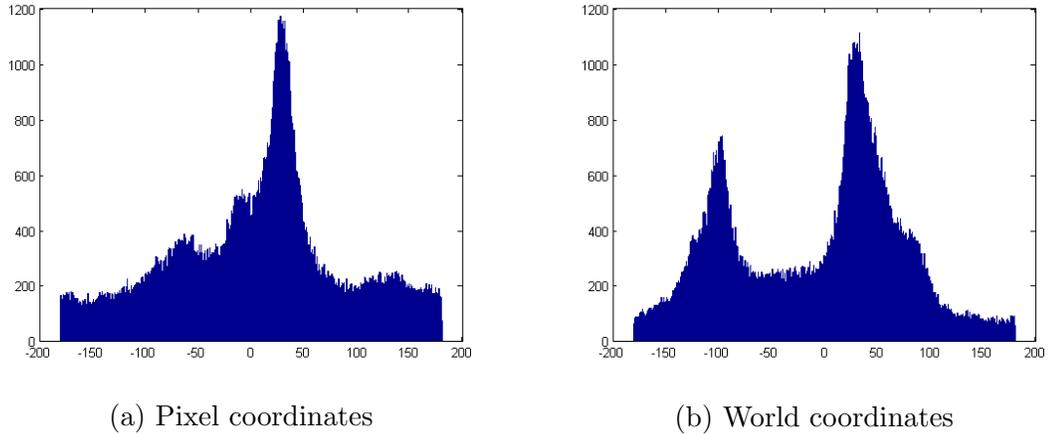
Figure 16: Heading distributions

non-visual cues to each other, and deciding the maneuver that will best suit their needs. Although there are no traffic signals or rules, traffic accidents and gridlock is infrequent. Observed in Figure 17 the area of the highest congestion and slowest average speed is not the center of the intersection but actually just before the center, similar to the effect seen in the probability distribution function of objects under a sequential conflict rule [27]. The system is very efficient for the large group of vehicles traveling through the intersection. With this in mind, my next goal is to expand previous research [19, 20] on sequential conflict resolution in intersecting aircraft flows to include flows of arbitrary thickness (i.e. to mimic the geometry of the self-organized flows that were analyzed). The next chapter discusses the aircraft flow model used followed by a chapter on the stability of these flows utilizing sequential control. More comparisons between the self-organized flows and the aircraft flows are made later on in the thesis.

(a) Pixel speed distribution

(b) Density in Pixel coordinates

(c) World speed distribution

(d) Density in World coordinates
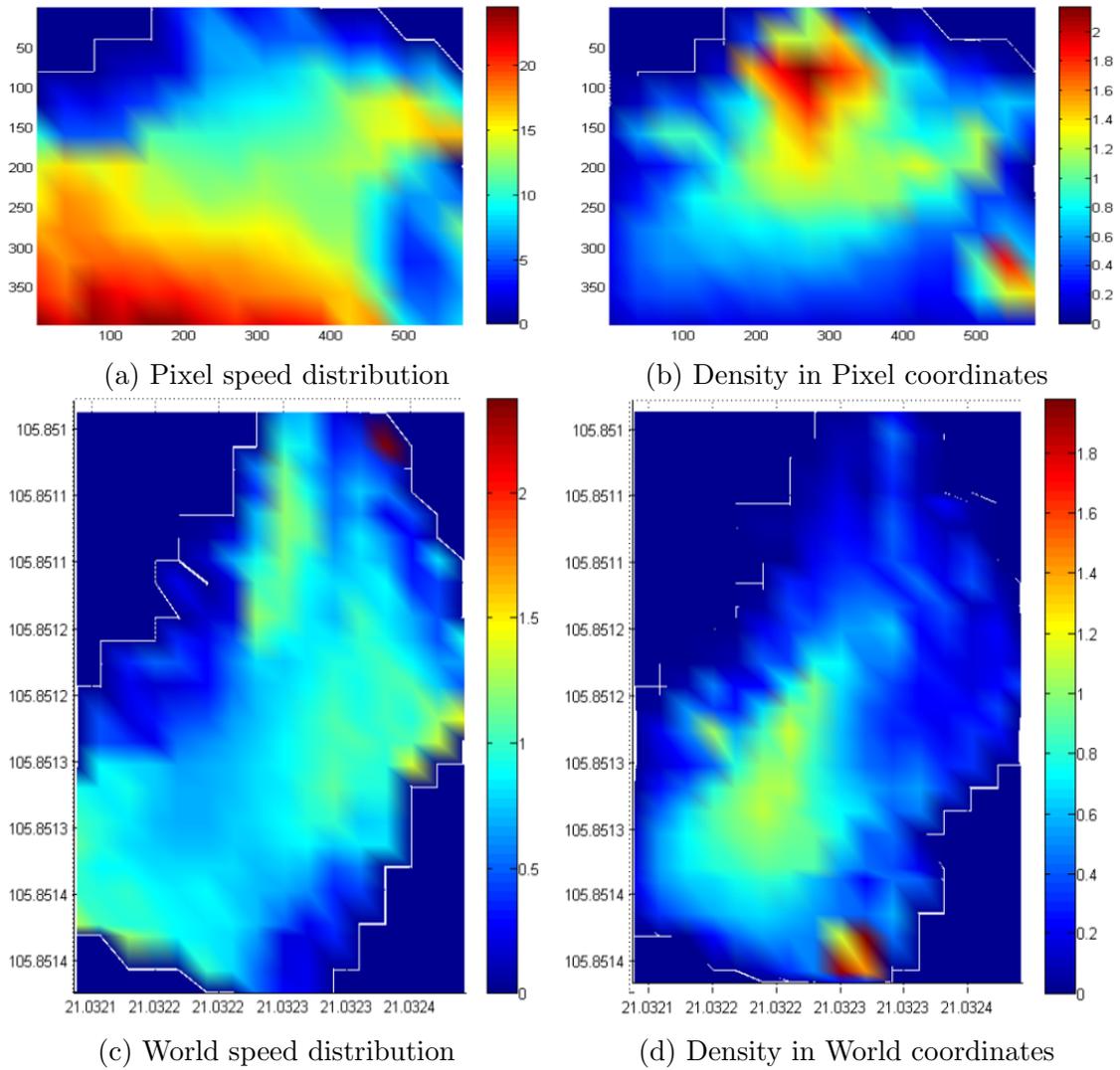
Figure 17: Speed and density distributions

CHAPTER IV

SEQUENTIAL CONTROL OF AIRCRAFT FLOWS
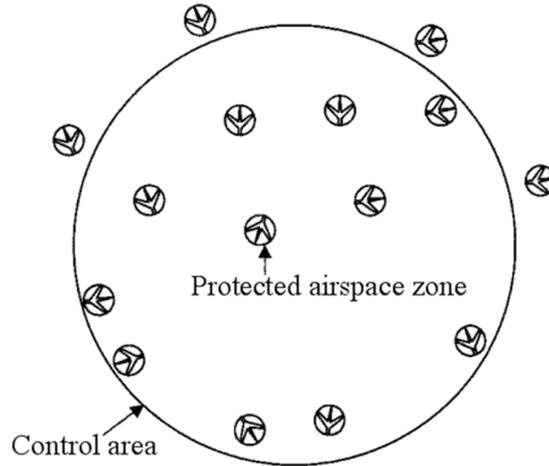
Figure 18: Aircraft flowing in and out of the "control area".

4.1 Aircraft flow model

The models presented in this thesis consist of a continuous flow of spatially distributed aircraft traveling sequentially through a finite airspace. For simplifying purposes, all aircraft are assumed to fly at the same altitude and at a constant velocity along straight paths. This may seem like a large constraint on the problem, however most flight trajectories will fly at constant flight levels for different legs of the flight. Moreover, many current air traffic controllers will arrange aircraft in their sector irregardless of their altitude/flight level. Therefore a control area is considered for the zone of conflicts, this can be thought of much like what an actual air traffic controller would see on their computer screen. Figure 18 illustrates aircraft flowing in and out of the control area. It is assumed that the position and velocity of all aircrafts within the
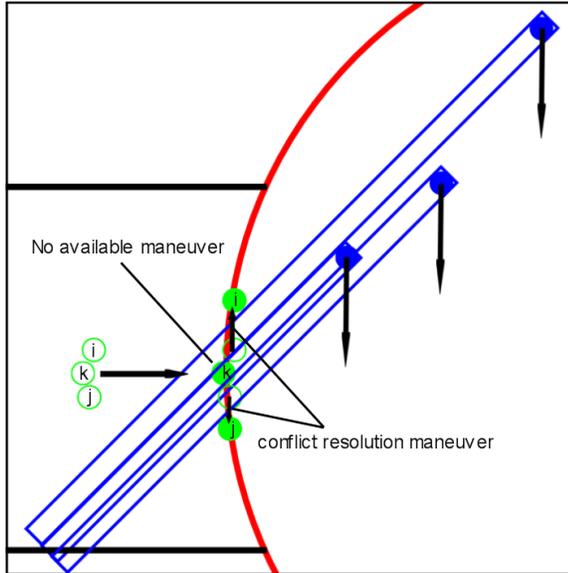
Figure 19: Conflict within the same flow resulting in no available lateral conflict resolution maneuver.

control area are known; this can be achieved using current systems or the future use of GPS in a context named ADS-B [6].

I concentrate on a continuous flow of aircraft flowing through a finite portion of the airspace to alleviate concerns about the "domino effect", whereby one conflict resolution maneuver creates new conflicts which in turn need to be solved etc. To avoid potential conflicts within the same flow, the aircraft are spaced a given miss distance apart. A simple example of same flow conflict is illustrated in Figure 19; aircraft i makes a conflict resolution maneuver to the left and aircraft j makes one to the right resulting in aircraft k unable to make a conflict resolution maneuver because of aircraft i and j (i.e. same flow conflict). All the models incorporate an optimized decentralized conflict resolution rule, which simply determines which conflict resolution maneuver (i.e. left or right) would result in the least displacement for the current aircraft. As an aircraft enters the conflict zone it determines one maneuver in which it will avoid all aircraft that have already entered the conflict zone resulting in the smallest deviation.

4.1.1 Aircraft maneuver models

While it is customary to consider three dimensions when designing and analyzing systems for aircraft conflict detection, this paper will only investigate air traffic evolving in two dimensions: The trajectories of all aircraft are assumed to develop in the horizontal plane. Although tactical conflict resolution has displayed that vertical maneuvers appear to be most efficient (such as in the case of TCAS (Traffic Alert and Collision Avoidance System)), horizontal maneuvers might be more appropriate for the "strategic" conflict resolution context considered in this thesis, because they induce less passenger discomfort and they do not require flight level changes and thus may not perturb the vertically stratified traffic structure as it exists today in the enroute airspace.

Three models were considered for the conflict resolution maneuver in this thesis as illustrated in Figure 20:

(1) Heading control model: Single heading changes are used to modify aircraft trajectories. Following the approach of Andrews [33], these changes are assumed to occur instantaneously when the aircraft makes a decision, This model is used for simulation purposes in [20].

(2) Double heading control model: Conflict resolution of an aircraft is realized by two successive instantaneous heading changes. A detailed discussion about this model can be found in [34].

(3) Offset model: This model consists of modeling aircraft trajectory changes via a single instantaneous lateral position change, the velocity and heading remaining the same before and after the position change. Although this appears to be less realistic, it is simpler to use for analysis purposes. In addition, the offset model can be treated as a close approximate of the heading control model and the double heading control model.

While analyzing the models in Figure 20 it is determined that given the distance
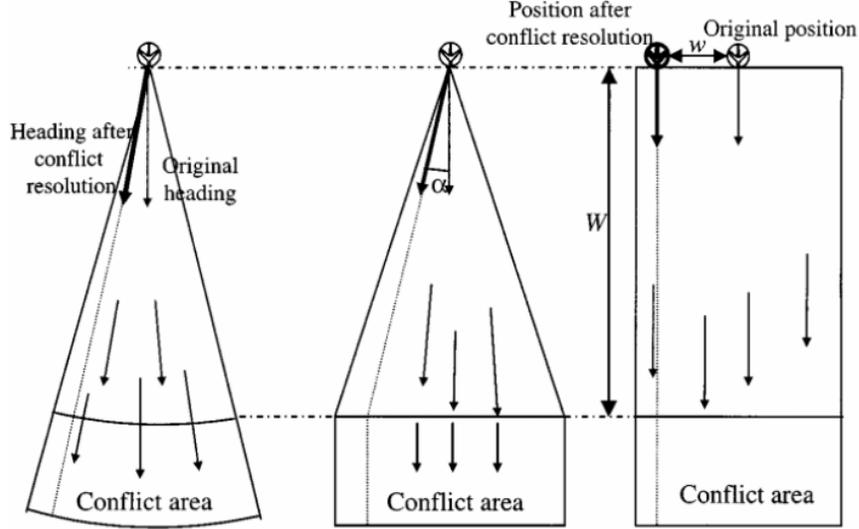
Figure 20: Heading control model *v.s.* offset model. Left: The aircraft maneuver is an instantaneous heading change. Middle: The aircraft maneuver is an instantaneous heading change followed by a second instantaneous heading change. Right: the aircraft maneuver is an instantaneous position change.

to conflict W, the lateral displacement w in the offset model is equivalent to a heading change of amplitude $\alpha = \tan{(w/W)}^{-1}$. If W is assumed much greater than w which is usually the case for strategic conflict resolution, the longitudinal displacement difference between the models is on the order of w^2/W, which is assumed to be small. Comparisons between the offset model and the other two models can be found in [20, 34]. In this thesis the offset model is used for the conflict resolution maneuver. However, the basic ideas of this work can be naturally generalized to the other models.

4.1.2 Aircraft flow arrival geometry

The basic aircraft flow model chosen in [20] is shown in Figure 21, originally introduced by Niedringhaus [35]. Two aircraft flows, oriented at a given angle θ ($\theta = 90^o$ in Figure 21) relative to each other, feed aircraft into a circular conflict area. The flows are organized in such a way that all aircraft within each flow are originally headed in the same direction. For simplicity of exposition only, it will also be assumed all
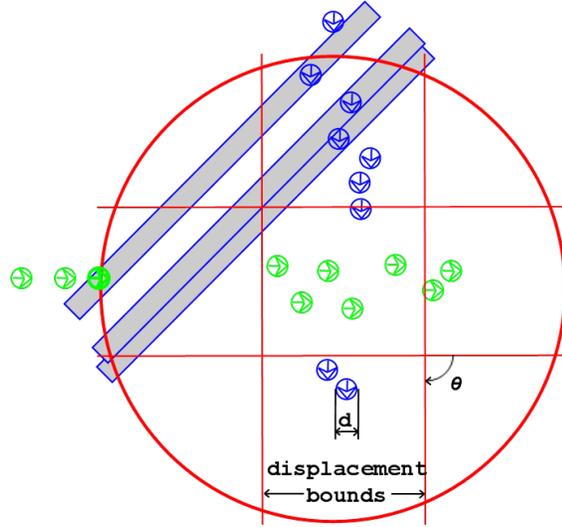
Figure 21: Aircraft flow configuration for stability analysis. The shaded areas are location where a conflict will occur.

aircraft are originally flying along the same track prior to entering the control volume. The spacing between each aircraft in each flow is arbitrary but no less than a given minimum safe distance d (in practice $d = 5nm$). Let $A_1, A_2, ..., A_i, ...$ be the set of aircraft entering the control volume, where aircraft are indexed according to the order they entered the control volume. When two aircraft enter the control volume at the same time, the eastbound aircraft is indexed first.

A review of several generalizations to the basic flow geometry will be discussed in this thesis, including the consideration of arbitrary encounter angles, different times to conflict, different aircraft speeds and multiple aircraft flows. And finally the flow geometry will be further generalized to include arbitrary flow thickness in an attempt to mimic what was observed in the analysis of the self-organized traffic flow.

4.1.3 Conflict resolution rules

Several centralized and decentralized conflict resolution rules are available (see for example [35–42]).

A conflict is declared whenever the projected straight path of any aircraft pair

leads them to a miss distance that is less than d. The decentralized conflict resolution scheme chosen in this paper follows a sequential approach, whereby aircraft solve potential conflicts one at a time. To simplify matters, it is assumed that the order in which aircraft perform their resolution maneuver is the same as the order they enter the circular conflict area, although this assumption could be relaxed. An aircraft solving a conflict considers all other aircraft that maneuvered before it as moving obstacles, but does not account for the aircraft which have not maneuvered yet. Thus each aircraft has knowledge of all aircraft that have already performed a maneuver (or decided that no maneuver was necessary). A reliable implementation of such sequential approaches is described in [43]. Each aircraft within the control area projects a linear, slab-shaped "corridor" of width d, centered around the aircraft and inclined 45° (denoted in light gray in Figure 21). For general encounter angles and aircraft speeds, this aisle is oriented along the relative velocity vectors. For an aircraft to avoid any conflict, it must maneuver so that the circle does not intersect any of the shaded areas. Failure to do so means that a conflict will occur.

4.1.4 Conflict resolution maneuver

Given those aircraft $A_1, ..., A_{i-1}$ which have already performed a resolution maneuver (and must therefore be considered as obstacles), the resolution maneuver for the next aircraft A_i scheduled for conflict resolution will be such that (i) no conflict exists between A_i and $A_1, ..., A_{i-1}$ after the resolution and (ii) the amplitude of the conflict avoidance maneuver is as small as possible. For the heading control model, the resolution maneuver will be to minimize the amplitude of the deviation from the nominal heading. Similarly, for the offset model, the resolution maneuver will attempt to minimize the lateral position change necessary for conflict resolution. In the computer simulations presented thereafter, the conflict resolution maneuver is generated by a line search away from the nominal heading and position in both directions (i.e.

displacement left and right), using a predefined step size. The maneuver resulting in the least displacement is used as the conflict resolution maneuver.

4.2   Aircraft flow stability

The system of two intersecting aircraft flows under the sequential conflict resolution rule described above, is stable. More precisely, the following question is asked: Assuming the system has been running "stable" in the past, will it keep running "stable" in the future? Indeed, it is possible to construct "initial conditions" for the system such that conflicts are unavoidable. I will now show that an incoming aircraft can always find a conflict resolution maneuver and proceed with a conflict-free trajectory.

4.2.1   Definition of stability

In this paper, the system will be stable if:

(1) After conflict detection, all conflicts are resolved without creating any new conflicts.

(2) The deviation of the aircraft trajectory from nominal, due to the requirement for conflict resolution, remains bounded within the control volume.

This definition summarizes the two most important requirements in all traffic control: Guaranteed safety and efficiency of traffic handling. I will now discuss the stability of the basic aircraft flow model and the generalizations to it.

4.2.2   Review of previous flow geometries

The flow geometries discussed in this section have been previously researched and proven to be stable with analytical solutions to the bounds of the conflict resolution maneuvers. I present them here as an introduction to my contribution which includes flows of arbitrary thickness. They also provide a means of verifying the simulation model.
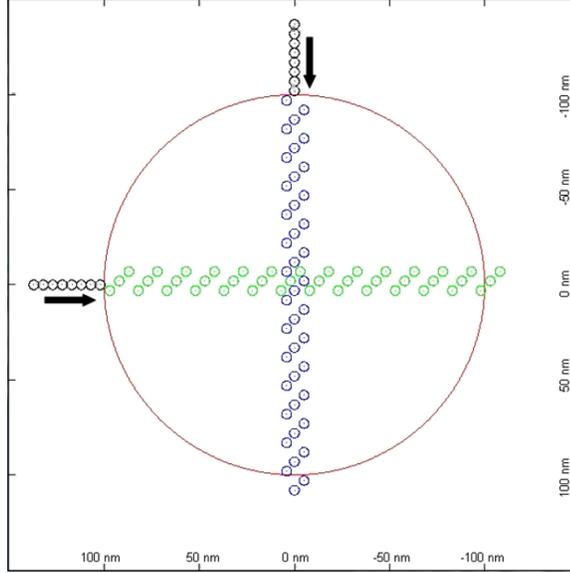
Figure 22: Orthogonal flow geometry simulation results

4.2.2.1   Orthogonal flow geometry

The orthogonal flow geometry is the basic aircraft flow model discussed above with aircraft spaced a distance $d = 5nm$ apart. This flow geometry consists of two continuous aircraft flows; one southbound and the other eastbound oriented 90° with respect to each other, as shown in Figure 22. The minimum miss distance is acceptable if and only if the small circles (of diameter $5nm$) centered around the aircraft do not overlap. Aircraft in each flow maintain the same velocity and follow along the same path; hence, aircraft in the same flow never intersect. Therefore the aircraft in one flow needs only to consider avoiding the aircraft in the orthogonal flow that are already within the control area and considered as moving obstacles. The control area considered for the zone of conflicts has a radius $R_{cz} = 100nm$.

It has been shown for the basic aircraft flow model that an aircraft entering the control area can always execute a lateral position change maneuver that results in a conflict-free trajectory. Further, it is proven that the lateral deviation of each aircraft is bounded above by $\sqrt{2}d$.

42
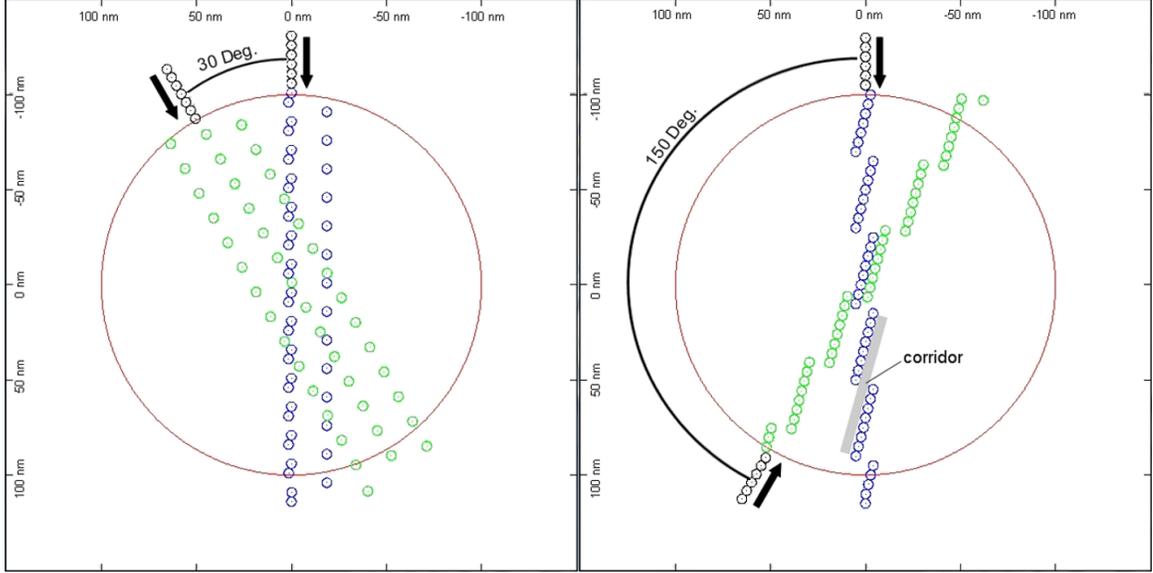
(a) 30° encounter angle   (b) 150° encounter angle

Figure 23: Arbitrary encounter angle simulation results

4.2.2.2   Arbitrary encounter angles

This generalization was achieved by keeping the orientation of one flow constant and angling the other flow away from the first. Let θ be the encounter angle measured counter-clockwise between the fixed flow to the other flow: $\theta = 0^o$ corresponds to the case when the two flows are parallel and $\theta = 180^o$ corresponds to the two flows going opposite directions. The control area and miss distance are the same as those of the orthogonal flow geometry.

The maximum deviation d_{max} created by the conflict resolution maneuver is bounded above by

$$\frac{d}{\left|\sin\left(\frac{\theta}{2}\right)\right|} \tag{1}$$

The next section provides a proof for a generalized version of (1).

Two simulations with different encounter angles of the two aircraft flows are shown in Figure 23. The largest lateral displacements of aircraft in the two examples are 19.3 nm and 5.2 nm respectively; the exact bounds obtained by 1. Note that some aircraft take advantage of a corridor that a previous aircraft created. This corridor is

43

the band created by propagating the safety circle of one aircraft along the constant relative velocity of the aircraft in the two flows. In Figure 23 notice when a small encounter angle is utilized, the offset distance the aircraft must travel for conflict resolution is greater than when the encounter angle is larger. Also the maximum number of aircraft involved with the same conflict increases as the encounter angle increases.

4.2.2.3   Aircraft maneuvering at different times to conflict

The *time to conflict* is defined as the period of time that is needed by an aircraft to fly, at the original velocity before resolution, from the point where the aircraft enters the control volume to the conflicting point, *i.e.*, the intersection point of the aircraft nominal paths. In the above definition, the distance between the point where an aircraft enters the control volume and the conflicting point is called the *distance to conflict.*

In the previous discussions, the control volume is assumed to be circular. However, our proof of stability has no specific requirement on the shape of the control volume. The control volume only needs to satisfy that each aircraft entering the control volume maneuvers at the same time to conflict.

Now I discuss the situation where the aircraft maneuver at different times to conflict. Consider two crossing aircraft flows, one eastbound and the other southbound. All aircraft fly at the same speed. Let d_1 and d_2 be the distances to conflict for the eastbound aircraft and southbound aircraft, respectively. If $d_1 \neq d_2$, then the eastbound aircraft and southbound aircraft maneuver at different times to conflict. Without loss of generality, assume $d_1 \leq d_2$.

In the conflict resolution for the above scenario, the lateral deviation of each eastbound aircraft is bounded by

$$\sqrt{2}d + |d_1 - d_2|, \tag{2}$$
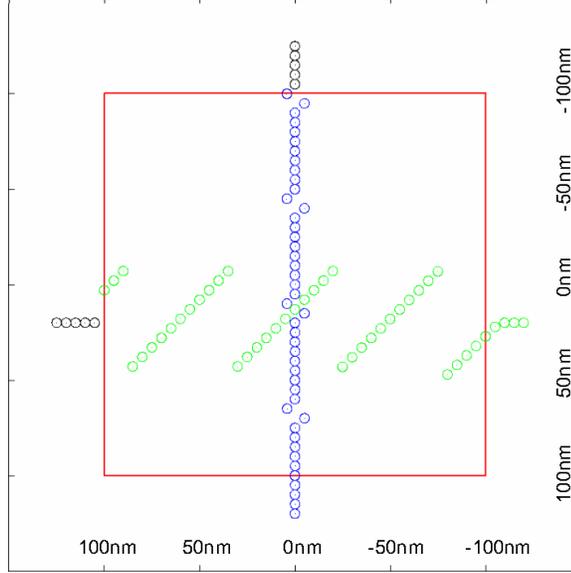
Figure 24: Two aircraft flows maneuvering at different times to conflict.

and the lateral displacement of each southbound aircraft is bounded by

$$\sqrt{2}d. \tag{3}$$

This conclusion can also be treated as a corollary of the result in the next section, where the proof is given.

Figure 24 shows a test example, in which the control volume is a square with $d_1 = 100nm$ and $d_2 = 115nm$. The aircraft flows in this simulation have fixed initial separation distance, $d = 5nm$. The largest lateral displacement is $22.1nm$ for the eastbound aircraft, and is $5nm$ for the southbound aircraft, both of which are within the bounds obtained by (2) and (3). Also it is noticed that the aircraft responding later to a conflict will on average experience larger lateral deviation to resolve the conflict.

4.2.2.4   Arbitrary flow velocity

The assumption of "same aircraft speed" is further relaxed to "different aircraft speeds". Let v_1 and v_2 be the speeds of aircraft in the two flows, respectively, and

45
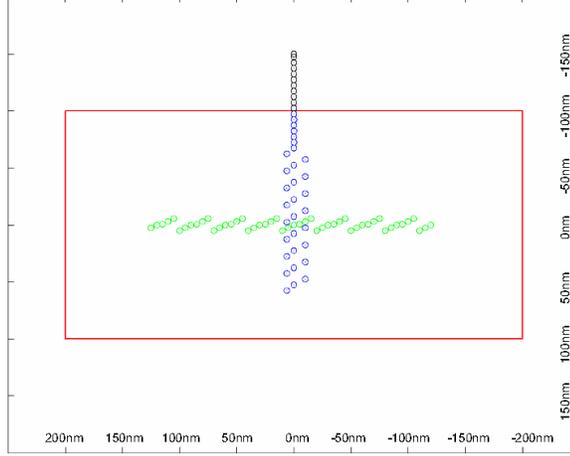
Figure 25: Two aircraft flows with different speeds (ratio 2:1), but maneuvering at the same time to conflict

denote $\mu = v_2/v_1$. For simplicity the encounter angle between the two aircraft flows is assumed to be 90 deg, and in order to be fair to the aircraft in different flows, each aircraft is assumed to maneuver at the same time to conflict. (These two assumptions will be relaxed in the next section.) Therefore $d_2 = \mu d_1$, were d_i is the distance to conflict for aircraft with speed v_i, $i = 1$ or 2.

Denote θ_i as the angle between the aisle and the flying direction of the aircraft with speed v_i, $i = 1$ or 2. It can be tested that $\theta_1 = \cos^{-1}((1+\mu^2)^{-\frac{1}{2}})$ and $\theta_2 = \cos^{-1}(\mu(1+\mu^2)^{-\frac{1}{2}})$. Then it is derived that, in the conflict resolution for the above scenario, the lateral deviation of aircraft with speed v_1 is bounded by

$$d\sqrt{1+\mu^2}, \tag{4}$$

and the lateral deviation of aircraft with speed v_2 is bounded by

$$\frac{d\sqrt{1+\mu^2}}{\mu}. \tag{5}$$

Figure. 25 illustrates an example for two flows of aircraft with speed ratio $2 : 1$.

46
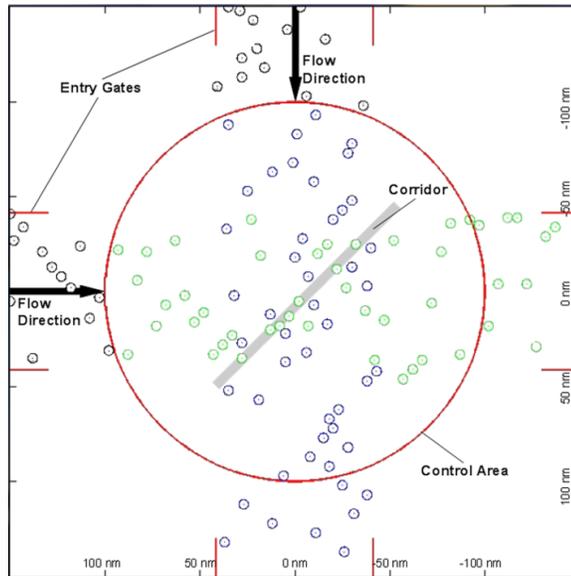
Figure 26: Arbitrary flow thickness geometry simulation

4.2.3   Flow geometries of arbitrary thickness

To generalize the flow geometry yet further, an arbitrary flow thickness is considered for the intersecting flows. For simplicity, the aircraft have constant speed and heading. The aircraft, however, enter the airspace at random positions within a specified starting area or "entry gates". This generalization remains close to the orthogonal flow geometry while adding some positive width to the aircraft flows as well as a bit of randomness to the potential conflicts. I can approach the geometry of the self-organized flows observed in the video analysis.

The control area is shown in Figure 26 as the red circle; and the initial flow bounds are shown as red gates located at the original flow's entrance and exit. The flow thickness is simply the distance from one entry gate to the other for a given flow. The aircraft enter sequentially but at random positions within the entry gates. Since the heading and speed is constant there are no conflicts within the same flow.

Just as in the previous models, the aircraft in the arbitrary flow thickness geometry maneuver to fall into a corridor that a previous aircraft creates to decrease
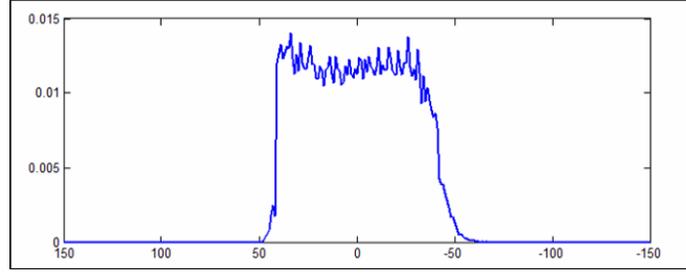
Figure 27: Exiting aircraft distribution after conflict resolution

the required displacement for conflict resolution (Figure 26). Also, as seen in Figure 26, it is difficult to examine the distribution of the aircraft in any flow, and the displacement bounds are therefore harder to determine graphically. This flow model was simulated for a large number of aircraft in each flow to supply sufficient data to create a probability distribution of the aircraft as they exited the airspace. Aircraft enter the airspace with a uniform distribution from gate to gate ($40nm$ to $-40nm$) and Figure. 27 shows the distribution of aircraft after conflict resolution maneuvers.

A detailed inspection of Figure 27 indicates that the distribution of southbound aircraft after conflict resolution has a broader support than prior to the conflict resolution. Moreover, this distribution is not symmetric, with larger possible deviations to the east than to the west. Aircraft also show a higher probability of making a maneuver toward the opposing flow, matching the observations from the authors of [27] with their probability density function.

As observed in the simulations, the displacement bounds for the arbitrary flow thickness geometry are asymmetric; however, I am able to show that each individual aircraft within the flow has its own displacement bounds, which are symmetric like before in the flows of zero thickness. Each aircraft's displacement bounds are a function of its starting position. The bounds of the entire flow of aircraft is simply the superposition of the displacement bounds for the individual aircrafts. With this I can prove the following two equations giving a maximum displacement to the right
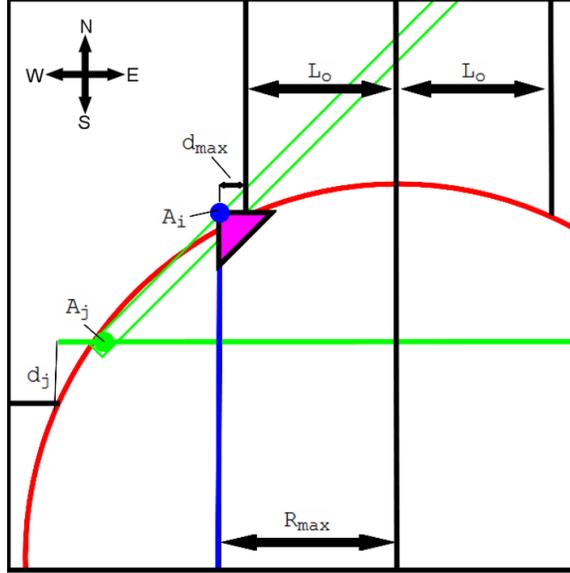
Figure 28: Arbitrary flow thickness geometry

and left of the center of the aircraft flow from the aircraft point of view:

$$R_{max} = L_0 + \sqrt{2}d \tag{6}$$

$$L_{max} = 3L_0 + \sqrt{2}d \tag{7}$$

The fact that there are bounds on the conflict resolution maneuvers tell us that the sequential conflict resolution rule with an arbitrary flow thickness geometry is stable as long as the bounds remain within the control area. Recall that a conflict resolution rule is considered stable if the conflict resolution maneuver is bounded and all conflicts are resolved without creating any new conflicts. A generalized proof for (6) and (7) is given in the following section.

4.2.4   The generalized two aircraft flows

I will now take into account all the generalizations that were made previously: The two aircraft flows considered here may be of arbitrary thickness, have arbitrary encounter angles, different times to conflict, and different speeds.

Consider two aircraft flows, flow 1 and flow 2. Let v_1 and v_2 be the speeds of aircraft in flow 1 and flow 2 respectively, and denote $\mu = v_2/v_1$. Let θ be the encounter angle between the two aircraft flows. Denote d_i as the distance to conflict for the aircraft in flow i, and let θ_i be the angle between the aisle (or the relative velocity vector) and the flying direction of the aircraft with speed v_i, $i = 1$ or 2.

It is proven in [19] for flows of zero thickness that θ_i is determined by

$$\cos^{-1}\left(\frac{1 - \mu \cos\theta}{\sqrt{1 + \mu^2 - 2\mu \cos\theta}}\right). \tag{8}$$

Also, if $\mu = 1/\cos\theta$ or $\mu = \cos\theta$, the aircraft from flow 1 or flow 2 can not resolve potential conflicts via lateral position changes only. If $\mu \neq \frac{1}{\cos\theta}$ and $\mu \neq \cos\theta$, the aircraft with speed v_1 can always execute a lateral displacement maneuver that results in a conflict-free trajectory. Further, for flows of zero thickness the lateral displacement is proven in [19] to be bounded above by

$$d_{max}' = \frac{d}{|\cos\theta_1|} + \max\{d_2 \sin\theta + d_2 \cos\theta \tan\theta_1 - d_1 \tan\theta_1, 0\} \tag{9}$$

Theorem 1. *For flows of arbitrary thickness $2L_0$, the conflict resolution maneuvers are bounded to the right and left by:*

$$R_{max} = L_0 + d_{max}' \tag{10}$$

$$L_{max} = 3L_0 + d_{max}' \tag{11}$$

Proof. Please refer to [19] for the proof of (8) and (9). I will start with the proof of R_{max} (10) first followed by the proof of L_{max} (11).

$R_{\max}$: Consider a southbound aircraft A_i entering the control area at its far west position, L_0 away from the center of the flow. A hypothesis is now made that states

there is no maneuver of amplitude less than or equal to d_{max}', as given in (9), in which aircraft A_i is conflict free. This hypothesis implies two things: First, according to the hypothesis, the southbound aircraft A_i cannot travel in a corridor created by a previous southbound aircraft whose own avoidance maneuver is small enough, because this would result in a conflict free trajectory, contradicting the hypotheses. Geometrically, no southbound aircraft can be in the pink triangular region in Figure 28 at the time A_i makes its conflict resolution maneuver. Second, for all possible lateral deviations of A_i with amplitude less than or equal to d_{max}', the southbound aircraft A_i is intersecting the shadow (i.e. conflict) of an eastbound aircraft, particularly A_j which has already made a maneuver of amplitude d_j, with $d_j > d_{max}'$ as shown in Figure 28. A contradiction is however reached, because there are no southbound aircraft within the small pink triangular region. Therefore, the eastbound aircraft A_j which was supposed to make the optimal maneuver should have had a displacement smaller than d_j and both aircraft would have been conflict free. Therefore the boundary for the aircraft's displacement to the right from the center is simply the far west initial position plus the displacement d_{max}'. Equation 10 shows the solution to the maximum displacement to the right(west) from the center of the aircraft flow.

$\mathbf{L_{max}}$: Because all aircraft must make an optimal maneuver, the bounds for each individual aircraft is symmetric, therefore the displacement an aircraft makes to the east must be less than or equal to the maximum displacement to the west. The displacement bounds of all other aircrafts in the flow can be determined rather easily now, since the right displacement bound is known. The displacement bound of any given aircraft d_{max}, can be expressed in terms of R_{max} as:

$$d_{max}(i) = R_{max}(i) - P_s(i) \tag{12}$$

Where $P_s(i)$ is the starting position of aircraft (i), the signed distance from the center of the flow (with west being the positive direction).

By examining (12) it is easy to see that an aircraft at the far east starting position

would generate the largest d_{max} and therefore the largest displacement to the west. Because the aircraft must make an optimal maneuver, the largest displacement to the east is also d_{max}. The bound for the aircraft's displacement to the left from the center of the flow is $2P_s - R_{max}$. Thus the largest displacement to the left, computed from the center on the flow, is obtained by setting $P_s = -L_0$, thus yielding (11) the maximum displacement to the left (east) from the center of the aircraft flow. □

The conclusions of the previous section can easily be verified as special cases to Theorem 1. For example, the case of $L_0 = 0$, $v_1 = v_2$, and $d_1 = d_2$ results in both (10) and (11) becoming (1), the arbitrary encounter angle geometry. For the case of $L_0 = 0$, $d_1/v_1 = d_2/v_2$, and $\theta = 90^o$, Theorem 1 obtains the same results as (4) and (5). Also For the case of $v_1 = v_2$, $d_1 = d_2$, and $\theta = 90^o$, (10) and (11) become (6) and (7) respectively.

4.2.4.1   Probability of maximum displacement

To better understand these limits I can look at the circumstances in which an aircraft will reach them; or in other words, when does the worst case scenario occur? The worst case scenario can be defined as the scenario in which an aircraft must make the largest maneuver to avoid conflict. Since the largest displacement bound is determined to occur for an aircraft starting at the far east position for the southbound flow; it follows that the worst case scenario will occur for an aircraft starting at the same position. A solution for the probability of the worst case scenario occurring is determined by setting the southbound aircraft to enter at the far east position and the eastbound aircraft to enter at the far north position. As seen in Figure 29 the aircraft deviate toward the displacement bound and then aircraft A_i maneuvers to the far west boundary resulting in the worst case scenario. Aircraft A_i's maneuver to the west boundary instead of the east boundary is arbitrary because the distance to each boundary is equivalent, therefore, either maneuver results in the least deviation
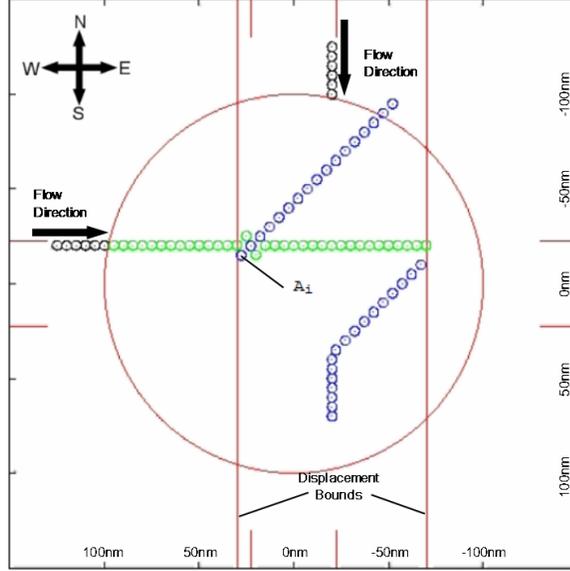
Figure 29: Worst case scenario

to achieve conflict resolution (i.e. an optimal maneuver).

If I consider the aircraft to enter the airspace randomly with a uniform distribution at $1nm$ increments from gate to gate I can determine the probability of this worst case occurring. From Figure 29 it is determined that the number of aircraft that must enter at the north most position for the eastbound flow, and the east most position for the southbound flow is $n = ceil[(R_{max} + L_{max})/d]$. The probability of an aircraft entering at any given location in the entry gates is $1/(2L_0 + 1)$. Therefore the probability that the aircraft would enter at the same location n times would be $(1/(2L_0 + 1))^n$. Since the southbound flow must have n aircraft enter at the far east position as well, the total probability of the worst case occurring in a continuous aircraft flow is given by:

$$P_{wc} = \left(\frac{1}{(2L_0 + 1)} \right)^{2n} \tag{13}$$

By plugging in n using Eqn. 10 and Eqn. 11 into Eqn. 13, the following equation is given for the probability of the worst case maneuver occurring as a function of the

53

thickness of the flow $2L_0$, the bound of a single maneuver d_{max}', and the size of the safety zone d around the aircraft:

$$P_{wc} = \left(\frac{1}{(2L_0 + 1)} \right)^{2(4L_0 + 2d_{max}')/d} \tag{14}$$

For $L_0 > 0$, the quantity $1/(2L_0 + 1)$ is always less than 1; therefore, for flows of large thickness $2L_0$, the probability of the worst case as determined by (14) can be assumed to be very small. On the other hand, for $L_0 = 0$ the worst case is guaranteed to occur. Also please keep in mind that the compactness in the flow geometry is to guarantee bounds exist while alleviating concerns of the "domino effect" and do not represent anticipated traffic flows.

4.3 Concerns with sequential control

As mentioned previously there are concerns when considering sequential control for conflict resolution. Some of these have already been presented and certain steps were taken in consideration of these concerns (i.e. domino effect, same flow conflict). There are certain continuous flow geometries in which there is not a guaranteed conflict resolution maneuver available. For example, it is proven in [19] that for three intersecting aircraft flows that the lateral displacement of the aircraft will linearly increase as the aircraft number increases. Another example would be to consider the arbitrary flow thickness geometry considered previously and relax the assumption of constant heading within each flow.

4.3.1 Arbitrary heading within the same flow

For this generalization, to preserve the idea of a flow thickness, there must be restrictions on the heading of each aircraft in order to stay within the original flow thickness bounds (Figure 30). Consider an aircraft i starting at position $P_s(i)$, west of the center of the flow of thickness $2L_0$. For the aircraft's original trajectory to
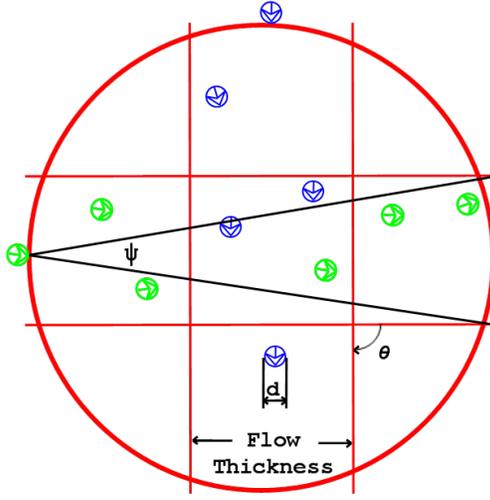
Figure 30: Arbitrary heading within same flow of arbitrary thickness

remain within the flow thickness, the heading of each aircraft must be bounded. The bounds of aircraft i's heading is determined by the following:

$$\arctan\left(\frac{-L_0 - P_s(i)}{2R_{cz}}\right) \leq \psi_l(i) \leq \arctan\left(\frac{L_0 - P_s(i)}{2R_{cz}}\right) \qquad (15)$$

If the same spacing of aircraft was used as in previous flow geometries, one immediate consequence would be same flow conflict, which cannot be resolved with displacement or heading maneuvers alone. If I further space the aircraft entering the airspace at larger increments, the problem can be alleviated. The spacing of aircraft in previous flow geometries was determined by the minimum safe distance d alone. Now, I allow for arbitrary headings ψ in the same flow of aircraft within a control area of radius R_{cz} and an encounter angle θ. The maximum heading allowed in each flow is $\psi_{max} = \arctan(L_0/R_{cz})$, determined from (15). The maximum heading must also be less than $\theta/2$, where θ is the encounter angle, to avoid aircraft from one flow having the same heading as an aircraft from the other flow. With these constraints on the heading, the spacing of aircraft required to remedy the concern of same flow conflicts is determined by the following:

$$Spacing = d + 2R_{cz}\left(\frac{1}{\cos(\psi_{max})} - 1\right) \tag{16}$$

Substituting in results for ψ_{max} into Eqn. 16 leads to the following equation in terms of the size of the control area R_{cz} and the flow thickness L_0:

$$Spacing = d + 2R_{cz}\left(\sqrt{1 + L_0{}^2/R_{cz}{}^2} - 1\right) \tag{17}$$

A question is now posed: Do the conflict resolution maneuvers for this flow geometry become unbounded? As this problem is much more difficult to determine the stability characteristics, it can be deduced from the definition of stability as the bounds on the conflict resolution maneuvers increase, the size of the control area R_{cz} must increase to ensure stability. Also from (17) as L_0 increases, the spacing between aircraft in each flow must be larger to ensure there is no conflict within the same flow. Therefore, there is a maximum size of the control area in which the flow of aircraft would become too low (i.e. the *Spacing* would become too large). This alone hints that a heading change or displacement maneuver is not enough to ensure conflict resolution for flows of large heading change ψ_{max} under a sequential conflict resolution scheme alone.

4.4   *Comparisons made with self-organized flows*

Now, with enough information about the generalized flow of intersecting aircraft, I can determine if they exhibit any self-organizing behavior when sequential control is utilized.

4.4.1   Existence of conflict resolution bounds

The proof of bounded conflict resolution maneuvers (10,11) provides evidence that all conflicts are resolved, which is obviously the case with self-organizing flows. Although, (10) and (11) show that the conflict resolution bounds have a broader support than

the original flow thickness, the probability of the maximum displacement occurring (14) is evident when conflict resolution bounds stay fairly close to the original flow thickness. This is visualized in Figure 27 of the aircraft flow and by examining the trajectories of the vehicles in the self-organized flow (Figure 15d).

4.4.2 Existence of "corridors"

As previously stated, the aircraft in geometries with arbitrary flow thickness maneuver to fall into a corridor that a previous aircraft creates to decrease the required displacement for conflict resolution. The orientation of the corridor is defined relative to the flow by (8), given the encounter angle θ, and the speed ratio μ. Therefore, through examination of Figure 16b, I can extract the heading of the main flows in the self-organized environment. The encounter angle between the two flows is $\theta \approx 130^o$ in the world coordinate system. The average velocity within each flow is calculated and the speed ratio μ of flow 1 ($Heading \approx 30^o$) and flow 2 ($Heading \approx -100^o$) is found to be $\mu \approx 1.51$. Using these values, the angle of the corridor relative to flow 1 is calculated as $\theta_1 \approx 30^o$ using (8). This data is overlapped on the Google maps photo in Figure 31a to show the orientation of the two main flows, and the corridor from a birds eye perspective, ensuring that the extracted data produced expected results. When translated back to the pixel coordinates as shown in Figure 31b, the same corridor defined in the sequential conflict resolution rules is observed as being utilized by the vehicles undergoing self-organization. Therefore, another aspect of self-organized flows is shown to exist when utilizing sequential control on intersecting aircraft flows.

4.4.3 Maneuver tendencies

Along with the corridors, it can be shown that the vehicles of self-organized flows tend to move towards the other flow. This is also seen in the simulation of aircraft under sequential control (Figure 27), as well as the probability density function of [27]. The
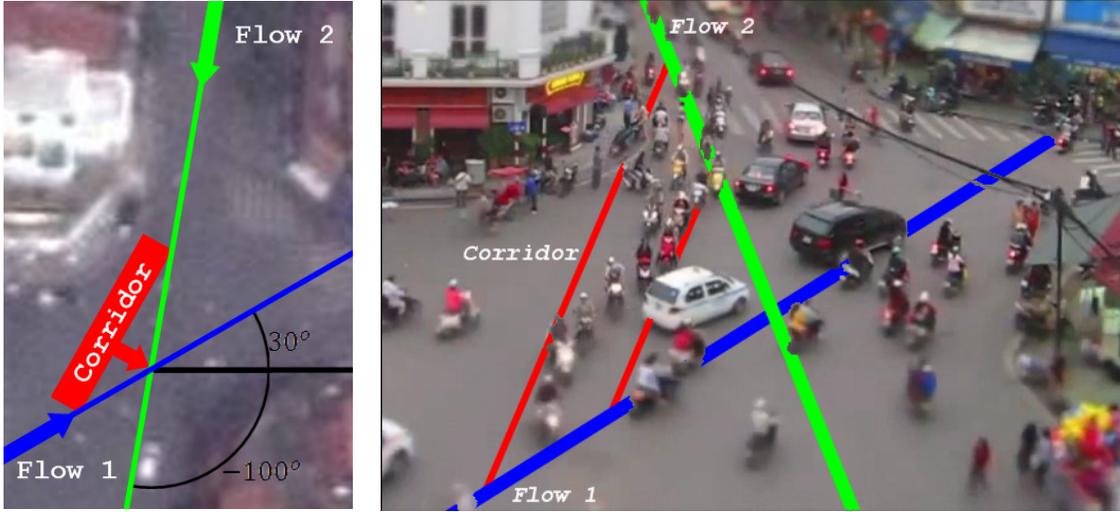
(a) World coordinates   (b) Pixel coordinates

Figure 31: Overlay of extracted heading: flow geometry and corridor existence

flow geometry (Figure 31a) information and the density data (Figure 17d) extracted extracted from the video is used to determine the location of highest density relative to each flow, hence determining what the most common maneuver would be. Figure 32 depicts the highest density towards the other main flow, excluding the balloon vendor outliers. Therefore, the sequential control scheme and the self-organized flows both have the tendency to maneuver toward the other flow.

As far as speed change maneuvers are concerned; the speed profile through the intersection can be generated by using the trajectory information for each object extracted from the self-organized flow (Figure 15d). It could be beneficial to assess the price (in terms of fuel consumption) of this maneuver. In fact, when aircraft are not flying at their optimal cruise speed fuel consumption is increased [44]. To alleviate the issue and to assess the impact of speed change, analysis of the specific fuel consumption (SFC) - $Mach$ number graphs of the engine and the aircraft would be necessary. For a jet engine: $C = TSFC * Thrust$ (Thrust SFC). Unfortunately, speed changes were not considered in the maneuver model of the simulation of intersecting aircraft flows under sequential control and these (SFC) - $Mach$ number graphics are
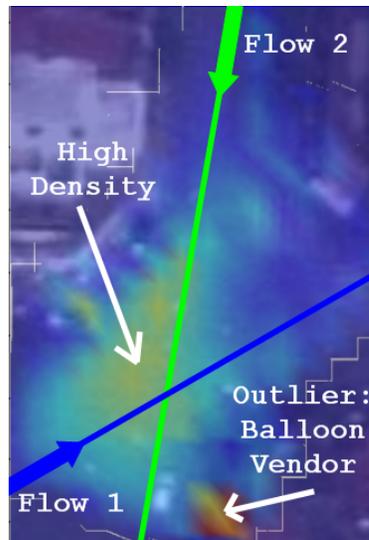
Figure 32: Overlay of flow geometry on position density

not easily attainable. With that said, one could compute the fuel consumption of one aircraft by following this process:

- Define an air speed for the aircraft

- Define the required thrust and angle of attack

- Compute the specific fuel consumption

- Compute the fuel consumption

CHAPTER V

CONCLUSION

5.1 Thesis summary

With the dramatic increase in air traffic demand by 2025 and the desire to curb the price of ATC, new solutions to air traffic control must be considered. As automation is a key element in achieving this, analytical guarantees on those elements must be made available to guarantee system safety. There should also be some understanding of the behavior of this system with the end goal of obtaining the concept of "free flight". The option discussed in this thesis was to use sequential conflict resolution rules in a controlled airspace. Some questions were raised: Can analytical guarantees on the sequential conflict resolution schemes be proven? Do vehicles under a sequential conflict resolution schemes exhibit self-organizing qualities? If not, how can they achieve behavior fitted to self-organization?

The simulations of aircraft flows demonstrated how a sequential air traffic control scheme would effect the aircraft flow and showed aircraft following the corridor created by the previous aircraft to minimize lateral displacement necessary for conflict resolution. It is proven that sequential conflict resolution schemes are stable for two intersecting aircraft flows of arbitrary geometries and arbitrary flow thickness. The stability of the aircraft flow models presented, provides evidence that a sequential conflict resolution scheme could be used to decrease the total number of conflicts within a sector (i.e. those that fall into the flow geometries discussed). This would lead to a decrease in the workload of air traffic controllers, allowing them to manage more aircraft. Moreover, from the video analysis conducted, it is concluded the sequential conflict resolution rules discussed are moving toward a self-organized traffic

system. Using this analysis, it is possible to determine how certain flow geometries will transform after conflict resolution. This leads to the idea of funneling aircraft into specific flow geometries and thereby anticipating necessary bounds on the flows after conflict.

Analysis of self-organized traffic flows was pertinent to the understanding of the aircraft flows under sequential control. A new tool was developed to extract the motion of agents in the video and provide analysis and visualization of the data. The tool allowed comparisons which determined the similar behaviors between sequential control and self-organized flows (i.e. coming closer to the concept of "free flight"). With that said, there are many uses for the motion extraction and visualization tool. The tool was created to operate on various video types and extract motion while providing visual feedback on the performance. Also, it has been explicitly created with the idea of modification in mind; creating a user friendly environment whereby new methods can be introduced and new techniques applied. Therefore, allows a broader support of functionality for a multitude of applications.

5.2   Future work

Currently, path planning research is utilizing this tool to extract real world data of vehicle flows. The binary image for each frame of video is being saved for use as moving obstacles. The image is then transformed into world coordinates and a mask is used to set the stationary objects (i.e. buildings) as obstacles as seen in Figure 33. Now, the binary image represents all obstacles as $1's$ and everything else as $0's$. The goal is for a robot to traverse across the intersection without hitting an obstacle. Therefore, providing a test bed to apply a multitude of path planning techniques to real world environments. An advancement could be imagined by use of a quadrocopter hovering and recording the intersection while sending the measurement data extracted from the video to a ground based robot. Henceforth, enabling the robot to determine
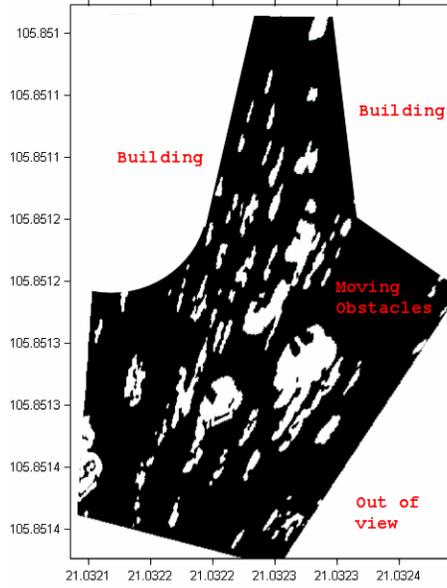
Figure 33: Binary image representing obstacles: white=obstacle

the best dynamic path to cross the intersection safely.

For the sequential control of intersecting aircraft flows, the next logical step of this research is to include a double heading maneuver model; representing a more realistic maneuver in an air traffic situation. Including speed changes in the maneuver model would be useful as well. A fuel burn analysis of the various maneuvers needed to ensure conflict resolution would be necessary to determine the feasibility of the sequential control used. Generating a more realistic traffic flow would be beneficial in understanding the impact of the control utilized. The next natural extension would be to move from the 2-D intersection to a 3-D environment.

REFERENCES

[1] NEXTOR, "Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the united states," Federal Aviation Administration, Technical Report, 2010.

[2] S. R. Wolfe, "Supporting air traffic flow management with agents," in <u>AAAI Spring Symposium: Interaction Challenges for Intelligent Assistants</u>, Stanford University, CA, USA, Mar. 2007.

[3] "RTCA task force 3: Free flight implementation," Final Report, RTCA, 1995.

[4] NASA, "Nasa and the next generation air transportation system (nextgen)," National Aeronautics and Space Administration, Washington D.C., USA, Technical Report, 2007.

[5] A. E. Jensen, "Working document on airport capacity and groundhandling: Towards a more efficiency policy," Committee on Transport and Tourism of the European Parliament, May 1993.

[6] I. Hwang and C. E. Seah, "Intent-based probabilistic conflict detection for the next generation air transportation system," <u>Proceedings of the IEEE</u>, vol. 96, no. 12, pp. 2040–2059, Dec. 2008.

[7] P. Ky and B. Miaillier, "SESAR: towards the new generation of air traffic management systems in Europe," <u>Journal of Air Traffic Control</u>, vol. 48, no. 1, pp. 11–14, Jan.-Mar. 2006.

[8] M. Gariel, A. Srivastava, and E. Feron, "Trajectory clustering and an application to airspace monitoring," Georgia Institute of Technology, Atlanta, Ga, Tech. Rep., 2009.

[9] A. Sipe and J. Moor, "Air traffic functions in the NextGen and SESAR airspace," in IEEE/AIAA 28th Digital Avionics Systems Conference, Orlando, FL, USA, Oct. 2009, pp. 2.A.6–1–2.A.6–7.

[10] H. Erzberger and R. A. Paielli, "Concept for next generation air traffic control system," Air Traffic Control Quarterly, vol. 10, no. 4, pp. 355–378, 2002.

[11] T. Vanderbilt, The traffic guru, July 2008. [Online]. Available: http://www.wilsonquarterly.com/article.cfm?AID=1234

[12] D. P., "Roundabouts 101: introduction to roundabouts," in TRB international roundabout conference, 2011.

[13] "Insurance institute of highway safety (iihs)," Roundabouts Status Report, vol. 35, no. 5, 2000.

[14] T. Prebble. (2011, Oct.) Hanoi intersection. [Online]. Available: https://www.youtube.com/watch?v=uAsKE78CF-s

[15] BBC. Oxford circus 'x-crossing' opens. [Online]. Available: http://news.bbc.co.uk/2/hi/8337341.stm

[16] YouTube. (2007) Saturday afternoon at shibuya crossing, tokyo. [Online]. Available: http://www.youtube.com/watch?v=QXt0dSgf6Ic

[17] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," Physical Review E, vol. 51, no. 5, pp. 4282–4286, May 1995.

[18] D. Helbing, L. Buzna, A. Johansson, and T. Werner, "Self-organized pedestrian crowd dynamics: experimens, simulations, and design solutions," Transportation Science, vol. 39, no. 1, pp. 1–24, Feb. 2005.

[19] Z.-H. Mao and E. Feron, "Stability and performance of intersecting aircraft flows under sequential conflict resolution," in Proc. American Control Conf., vol. 2, Arlington, VA, USA, Jun. 2001, pp. 722–729.

[20] Z.-H. Mao, E. Feron, and K. Bilimoria, "Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules," IEEE Transactions on Intelligent Transportation Systems, vol. 2, no. 2, pp. 101–109, Jun. 2001.

[21] M. Moussad, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," PLoS ONE, vol. 5, p. e10047, Apr. 2010.

[22] A. Johansson, A.-A. D. Helbing. H, and S. Al-Bosta, "From crowd dynamics to crowd safety: a video-based analysis," Advances in Complex Systems, vol. 11, no. 4, pp. 497–527, 2008.

[23] G. A. Carlson, DataPoint (Win32 Version 0.62) [Computer program], Xannah Applied Science and Engineering, Saint Peters, MO, 2003. [Online]. Available: http://www.xannah.org/datapoint

[24] Wikibooks, "Swistrack — wikibooks, the free textbook project," 2011, [Online; accessed 12-April-2013]. [Online]. Available: http://en.wikibooks.org/w/index.php?title=SwisTrack&oldid=2153660

[25] F. Bathge. (2011) object tracking using particle filters. [Online]. Available: studios.tinytall.de/projects/objecttracker

[26] A. Marzuoli, C. Hurter, and E. Feron, "Data visualization techniques for airspace flow modeling," in Intelligent Data Understanding (CIDU), 2012 Conference on. IEEE, 2012, pp. 79–86.

[27] E. Salaun, A. E. Vela, E. Feron, J.-P. Clarke, and S. Solak, "A simplified approach to determine airspace complexity maps under automated conflict resolution," in 28th Digital Avionics Systems Conference, Orlando, FL, USA, Oct. 2009.

[28] YouTube. (2011) Intersection time lapse. [Online]. Available: http://www.youtube.com/watch?v=th6iRkjwIqg

[29] ——. (2008) Time lapse atlanta intersection. [Online]. Available: http://www.youtube.com/watch?v=mDctl1s0OS8

[30] ——. (2012) Incredible indian traffic - isn't it crazy?! [Online]. Available: http://www.youtube.com/watch?v=KnPiP9PkLAs

[31] ——. (2010) Rush hour traffic in ho chi minh city, viet nam. [Online]. Available: http://www.youtube.com/watch?v=4phFYiMGCIY

[32] ——. (2007) Hanoi traffic. [Online; accessed 12-February-2012]. [Online]. Available: http://www.youtube.com/watch?v=GsXBjUuB9YM

[33] J. Andrews, "A relative motion analysis of horizontal avoidance," MIT Lincoln Lab, MA, USA, Technical Report, 1978.

[34] D. Dugail, Z.-H. Mao, and E. Feron, "Stability of intersecting aircraft flows under centralized and decentralized conflict avoidance rules," in GN&C Concepts in Air Traffic Control Systems Conference, 2001.

[35] W. Niedringhaus, "Stream option manager: automated integration of aircraft separation, merging, stream management, and other air traffic control functions," IEEE Trans. Systems, Man, and Cybernetics, vol. 25, no. 9, pp. 1269–1280, 1995.

[36] N. Durand, J.-M. Alliot, and O. Chansou, "An optimizing conflict solver for air traffic control," Air Traffic Control Quarterly, October 1995.

[37] E. Frazzoli, Z.-H. Mao, J.-H. Oh, , and E. Feron, "Resolution of conflicts involving many aircraft via semidefinite programming," Guidance, Control and Dynamics, vol. 24, no. 1, pp. 79–86, 2001.

[38] R. Ghosh and C. Tomlin, "Maneuver design for multiple aircraft conflict resolution," in American Control Conference, June 2000.

[39] J. Krozel, T. Mueller, and G. Hunter, "Free flight conflict detection and resolution analysis," 1996.

[40] J. Krozel, M. Peters, and K. Bilimoria, "A decentralized control strategy for distributed air/ground traffic separation," in AIAA Guidance, Navigation, and Control Conference, August 2000.

[41] P. Menon, G. Sweriduk, and B. Sridhar, "Optimal strategies for free- flight air traffic conflict resolution," Guidance, Control, and Dynamics, pp. 203–211, 1999.

[42] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A study in multiagent hybrid systems," IEEE Transaction on automatic control, vol. 43, no. 4, pp. 509–521, 1998.

[43] J.-M. Alliot, N. Durand, and G. Granger, "Faces: a free flight autonomous and coordinated embarked solver,," Air Traffic Control Quarterly, vol. 8, no. 2, pp. 109–130, 2000.

[44] É. Roux, "Pour une approche analytique de la dynamique du vol," These, SUPAERO-ONERA, 2005.