52. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

PROCCEDINGS | 10 - 13 September 2007

FACULTY OF COMPUTER SCIENCE AND AUTOMATION

COMPUTER SCIENCE MEETS AUTOMATION

VOLUME I

Session 1 - Systems Engineering and Intelligent Systems
Session 2 - Advances in Control Theory and Control Engineering
Session 3 - Optimisation and Management of Complex Systems and Networked Systems
Session 4 - Intelligent Vehicles and Mobile Systems
Session 5 - Robotics and Motion Systems

TECHNISCHE UNIVERSITÄT
ILMENAU

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der deutschen
Nationalbiografie; detaillierte bibliografische Daten sind im Internet über
http://dnb.ddb.de abrufbar.

ISBN 978-3-939473-17-6

Impressum

Herausgeber:
Der Rektor der Technischen Universität Ilmenau
Univ.-Prof. Dr. rer. nat. habil. Peter Scharff

Redaktion:
Referat Marketing und Studentische Angelegenheiten
Kongressorganisation
Andrea Schneider
Tel.: +49 3677 69-2520
Fax: +49 3677 69-1743
e-mail: kongressorganisation@tu-ilmenau.de

Redaktionsschluss: Juli 2007

Verlag:
Technische Universität Ilmenau/Universitätsbibliothek
Universitätsverlag Ilmenau
Postfach 10 05 65
98684 Ilmenau
www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung:
Verlagshaus Monsenstein und Vannerdat OHG
Am Hawerkamp 31
48155 Münster
www.mv-verlag.de

Layout Cover: www.cey-x.de

Bezugsmöglichkeiten:
Universitätsbibliothek der TU Ilmenau
Tel.: +49 3677 69-4615
Fax: +49 3677 69-4602

Preface

Dear Participants,

Confronted with the ever-increasing complexity of technical processes and the growing demands on their efficiency, security and flexibility, the scientific world needs to establish new methods of engineering design and new methods of systems operation. The factors likely to affect the design of the smart systems of the future will doubtless include the following:

- As computational costs decrease, it will be possible to apply more complex algorithms, even in real time. These algorithms will take into account system nonlinearities or provide online optimisation of the system's performance.
- New fields of application will be addressed. Interest is now being expressed, beyond that in "classical" technical systems and processes, in environmental systems or medical and bioengineering applications.
- The boundaries between software and hardware design are being eroded. New design methods will include co-design of software and hardware and even of sensor and actuator components.
- Automation will not only replace human operators but will assist, support and supervise humans so that their work is safe and even more effective.
- Networked systems or swarms will be crucial, requiring improvement of the communication within them and study of how their behaviour can be made globally consistent.
- The issues of security and safety, not only during the operation of systems but also in the course of their design, will continue to increase in importance.

The title "Computer Science meets Automation", borne by the 52nd International Scientific Colloquium (IWK) at the Technische Universität Ilmenau, Germany, expresses the desire of scientists and engineers to rise to these challenges, cooperating closely on innovative methods in the two disciplines of computer science and automation.

The IWK has a long tradition going back as far as 1953. In the years before 1989, a major function of the colloquium was to bring together scientists from both sides of the Iron Curtain. Naturally, bonds were also deepened between the countries from the East. Today, the objective of the colloquium is still to bring researchers together. They come from the eastern and western member states of the European Union, and, indeed, from all over the world. All who wish to share their ideas on the points where "Computer Science meets Automation" are addressed by this colloquium at the Technische Universität Ilmenau.

All the University's Faculties have joined forces to ensure that nothing is left out. Control engineering, information science, cybernetics, communication technology and systems engineering – for all of these and their applications (ranging from biological systems to heavy engineering), the issues are being covered.

Together with all the organizers I should like to thank you for your contributions to the conference, ensuring, as they do, a most interesting colloquium programme of an interdisciplinary nature.
I am looking forward to an inspiring colloquium. It promises to be a fine platform for you to present your research, to address new concepts and to meet colleagues in Ilmenau.

Professor Peter Scharff
Rector, TU Ilmenau

Professor Christoph Ament
Head of Organisation

Table of Contents

CONTENTS

Page

1 Systems Engineering and Intelligent Systems

2 Advances in Control Theory and Control Engineering

3 Optimisation and Management of Complex Systems and Networked Systems

4 Intelligent Vehicles and Mobile Systems

5 Robotics and Motion Systems

Paper submitted after copy deadline

2 Advances in Control Theory and Control Engineering

52^{nd} Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
10 – 13 September 2007

J. W. Vervoorst / T. Kopfstedt

Mission Planning for UAV Swarms

Introduction

Unmanned Aerial Vehicles (UAVs) play an ever increasing role in a wide variety of scenarios in both the civilian and military sector, carrying out tasks like traffic surveillance, firefighting, or reconnaissance missions. Furthermore, the use of groups of vehicles, or swarms, has been shown to accomplish certain objectives more efficiently and more effectively than a single vehicle can, for example in terrain mapping or search missions. In all these scenarios, the autonomous vehicles need to fly on trajectories that match their flight envelope, are as short as possible, and most importantly, avoid collisions with obstacles and other UAVs at all cost.

In order to achieve this, Mixed-Integer Linear Programming (MILP) is used as the optimization principle. MILP extends regular linear programming to include variables that are constrained to integer or binary values. Thus, MILP offers the possibility to add logical and decision making constraints into the optimization, such as obstacle and collision avoidance.

However, finding long-range minimum-time trajectories in environments with many obstacles is a complex optimization problem. In this paper, a Model Predictive Control (MPC) approach is chosen to decrease computational complexity and limit computation time, therefore making the algorithm capable of real-time calculations as well as handling unknown or dynamically changing environments. The presented algorithm is capable of calculating near-optimal flight trajectories to ensure that the UAV swarm carries out its mission in the minimum time.

Problem Formulation

In general, a mission for a UAV swarm can be described as visiting a number of waypoints spread out on a map. All waypoints have to be visited once during the mission by one UAV. This so called Task Assignment constitutes the first part of the mission planning algorithm. During the Task Assignment phase, waypoints are assigned such that each UAV will have to travel only a minimum distance, therefore also bringing the total mission time to a minimum. The problem is formulated as a multi-dimensional Traveling Salesman Problem (TSP), as implemented in [1].

In this paper, only planar motion is considered, meaning UAVs fly at constant altitude. The dynamics of the UAVs are modeled in the form of a simple point mass, bringing their discrete-time state space representation to:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_k + \begin{bmatrix} (\Delta t)^2/2 & 0 \\ 0 & (\Delta t)^2/2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \end{bmatrix}_k$$

UAV motion is constrained by a maximum velocity and turn rate, therefore giving us:

$$|v_x| \le v_{\max} \;,\; |v_y| \le v_{\max} \qquad |a_x| \le a_{\max} \;,\; |a_y| \le a_{\max}$$

Calculating the length of a vector is a nonlinear operation. In order to adhere to the Linear Programming problem formulation, we approximate vector length by checking a number $n_{v_{\max}}$ of unit vectors onto which the velocity/acceleration vector is projected [5].

$$v^T \cdot i_k \le v_{\max} \qquad k = 1,\dots,n_{v_{\max}} \qquad a^T \cdot i_k \le a_{\max} \qquad k = 1,\dots,n_{a_{\max}} \tag{1}$$

$$i_k = \begin{bmatrix} \cos\left(\dfrac{2\pi k}{n_{v_{\max}}}\right) \\ \sin\left(\dfrac{2\pi k}{n_{v_{\max}}}\right) \end{bmatrix} \qquad i_k = \begin{bmatrix} \cos\left(\dfrac{2\pi k}{n_{a_{\max}}}\right) \\ \sin\left(\dfrac{2\pi k}{n_{a_{\max}}}\right) \end{bmatrix} \tag{2}$$

MILP also allows an efficient way to declare obstacles by using binary variables. Throughout this paper we limit ourselves to rectangular obstacles. Obstacles are described by their lower left and upper right corner $[x_{ll}, y_{ll}, x_{ur}, y_{ur}]^T$. The variable b_{object} is a binary, and the number M is an arbitrary large positive value.

$$\begin{aligned} x_{jk} &\le (x_{ll}) + M \cdot b_{object_{ijk1}} \\ y_{jk} &\le (x_{ll}) + M \cdot b_{object_{ijk2}} \\ x_{jk} &\ge (x_{ur}) - M \cdot b_{object_{ijk3}} \\ y_{jk} &\ge (x_{ur}) - M \cdot b_{object_{ijk4}} \end{aligned} \qquad \begin{aligned} &\sum_{l=1}^{4} b_{object_{ijkl}} \le 3 \\ \\ &i = 1 \dots N_O \,,\quad j = 1 \dots N_V \\ &k = 1 \dots T \,,\quad l = 1 \dots 4 \end{aligned} \tag{3}$$

$[x_{jk}, y_{jk}]^T$ is the position of UAV j at time k. Index i describes the number of obstacles, j lists the number of UAVs in the swarm, k the number of time steps in the planning horizon, and l enumerates the four inequalities. As long as $\sum b_{object_{ijkl}} \le 3$ is fulfilled, the UAV is outside the obstacle.

Receding Horizon Controller

The constraints above can now be used to formulate a MILP optimization problem. Traditionally, trajectory optimization is done over a fixed horizon. This means that the complete trajectory is calculated from beginning to end point, forming a large and complicated optimization that does not take into account changing environments. When obstacles are added or subtracted, the precalculated optimal solution essentially becomes worthless and a recalculation has to be performed.

To solve this problem, a Model Predictive Control setup is chosen. [2] lists the properties of Model Predictive Control, also called Receding Horizon Control:

- At time i and initial state x_i the optimization is performed for only the next N_P time steps. N_P is the so called planning horizon.
- The first N_E values of the optimal solution are used as inputs to the system. N_E is the execution horizon. N_E is usually set to 1.
- In the new initial state x_{i+N_E} the optimization is performed again, repeating the process, thus taking into account possible changes in the environment.

Similar implementations of MPC can be found in [3] and [4].

To be able to use this approach, we need to overcome a problem. As seen in Fig. 1, the trajectory is only optimized within a small area around the current state, the planning horizon. But the total trajectory consists of the optimized piece plus the remaining pieces from the planning horizon to the goal point. However, that piece is unknown and the total distance cannot be calculated. Therefore, the remaining trajectory is approximated with straight line segments connecting the planning horizon to the goal point.

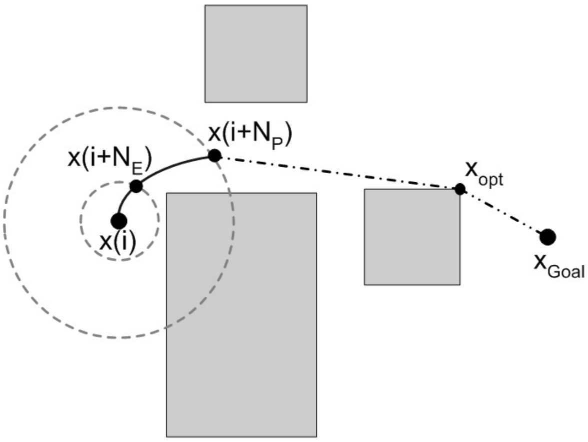

Fig. 1: Details of MPC

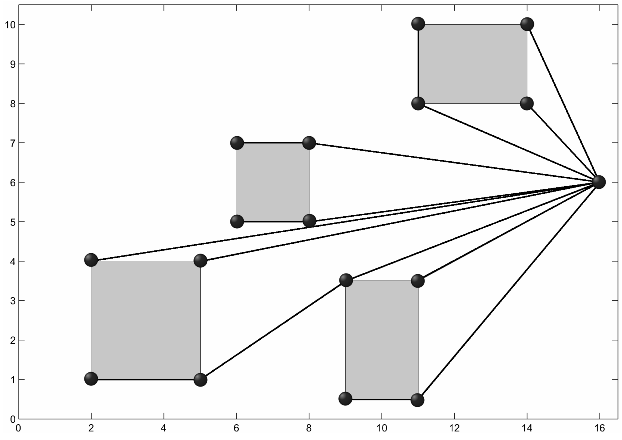

Fig. 2: Calculation of the cost map

In order to find the shortest distance to the goal point, a cost map of the complete environment is calculated, calculating and storing the distances of points to the goal point (Fig. 2). [6] shows that the shortest distance in an environment with convex obstacles is a combination of line segments between the points and the corners of the obstacles. Thus, the corners of the obstacles are cost points in our cost map. The visibility graph between all obstacles and the goal point is calculated and by using the Dijkstra algorithm, as outlined in [7], the shortest connections between all the points and the goal point are calculated and stored as the cost of each point.

The length of the trajectory outside the planning horizon can now be determined. As detailed in Fig. 1, it is:

- The distance from x_{i+N_P} ,the edge of the planning horizon, to a known cost point x_{opt} that is visible from that point.
- The distance from that cost point to the goal point. This value has already been calculated and is stored in the cost map.

The visibility constraint between x_{i+N_P} and x_{opt} is very important. Because of it, local optimization within the planning horizon does indeed have a global influence on the whole trajectory, therefore also minimizing the total trajectory length.

Now the planning problem is expressed in MILP form. Each UAV can select only one cost point during each optimization:

$$\sum_{i=1}^{N_{CP}} \sum_{j=1}^{N_{goal}} b_{CP_{ijk}} = 1 \ , \quad k = 1,\ldots,N_V$$

with N_{CP} = number of cost points, N_V = number of UAVs, N_{goal} =2, the next two waypoints.

Equations (4) and (5) set the cost value and coordinates of the chosen cost point.

$$c_{opt_k} = \sum_{i=1}^{N_{CP}} \sum_{j=1}^{N_{goal}} c_i \cdot b_{CP_{ijk}} \ , \quad k = 1,\ldots,N_V \qquad (4)$$

$$\begin{bmatrix} x_{opt_k} \\ y_{opt_k} \end{bmatrix} = \sum_{i=1}^{N_{CP}} \sum_{j=1}^{N_{goal}} \begin{bmatrix} x_{CP_k} \\ y_{CP_k} \end{bmatrix} \cdot b_{CP_{ijk}} \ , \quad k = 1,\ldots,N_V \qquad (5)$$

The line connecting the edge of the planning horizon and the cost point is described by

$$\begin{bmatrix} x_{line_k} \\ y_{line_k} \end{bmatrix} = \begin{bmatrix} x_{opt_k} \\ y_{opt_k} \end{bmatrix} - \begin{bmatrix} \left(x_{i+N_P}\right)_k \\ \left(y_{i+N_P}\right)_k \end{bmatrix}$$

In order to test visibility, this connection is divided into N_{test} parts:

$$\begin{bmatrix} x_{test_{km}} \\ y_{test_{km}} \end{bmatrix} = \begin{bmatrix} \left(x_{i+N_P}\right)_k \\ \left(y_{i+N_P}\right)_k \end{bmatrix} + \frac{m}{N_{test}} \cdot \begin{bmatrix} x_{line_k} \\ y_{line_k} \end{bmatrix} \qquad m = 1 \ldots N_{test}$$

Each part is tested for interference with obstacles using a binary variable $b_{opt_{ikmn}}$, very much like in (3).

Equations (6) and (7) handle which goal point to use in the optimization. If a goal point is reached in the last step of the planning horizon or not at all, then only this point will be part of the optimization. If a goal point is reached within the planning horizon, then goal points are switched and the optimization directs the trajectory to the next goal point.

$$\sum_{i=1}^{N_{CP}} b_{CP_{i1k}} = \sum_{i=T}^{T+1} b_{goal_{ij}} \tag{6}$$

$$\sum_{i=1}^{N_{CP}} b_{CP_{i2k}} = \sum_{i=1}^{T-1} b_{goal_{ij}} \tag{7}$$

Cost Function

The cost function to be minimized is the total trajectory lengths of the UAVs, consisting of three parts:

- The part within the planning horizon, from x_i to x_{i+N_P}
- The line between x_{i+N_P} and the selected cost point
- The distance between selected cost point and goal point.

The first part is described by the first part of Equation (8), which represents the number of time steps to the goal point times the distance traveled per time step.

$$Z = \sum_{j=1}^{N_V} \left(\left(v_{j_{\max}} \cdot \Delta t \right) \cdot \sum_{k}^{T+1} k \cdot b_{goal_{kj}} + l_{line} + c_{opt_j} \right) \tag{8}$$

l_{line} is the distance between x_{i+N_P} and the selected cost point. Since it constitutes the length of a vector, it is calculated just as in Equations (1) and (2).

The third part is simply the stored value of the cost point, meaning the distance to the goal point.

Simulation

Two scenarios are presented that show the capabilities of the presented MCD algorithm, especially in changing environments. Fig. 3 shows a trajectory replanning after a UAV is lost during a mission.

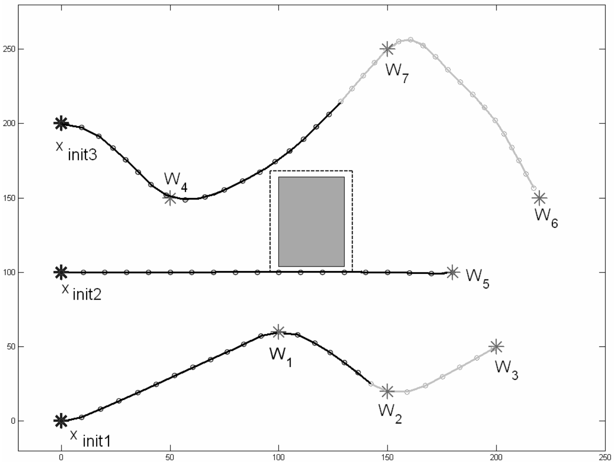

Fig. 3: Replanning after UAV loss

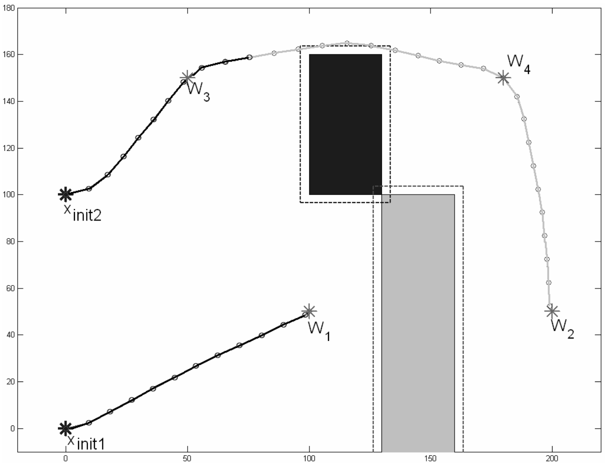

Fig. 4: Replanning due to unknown obstacles

The regularly planned trajectories are shown in black. UAV #2 is supposed to visit waypoints W_5 and W_6. However, UAV #2 is lost at W_5. Immediately, a replanning occurs and UAV #3 takes over W_6. The newly planned trajectories are shown in gray.

In Fig. 4, UAV #1 is scheduled to visit W_1 and W_2. However, it encounters an unknown obstacles, shown in gray, that was not part of the previous planning. After a replanning, it is determined that UAV #2 can reach W_2 faster and it takes over for #1.

Conclusion

The iterative MPC algorithm presented here has been shown to effectively handle various changes in the mission environment, decreasing computational complexity and still being able to calculate near-optimal trajectories for groups of cooperating UAVs.

References:

[1] SCHUMACHER, Corey; CHANDLER, Phillip; PACHTER, Meir: *Constrained Optimization for UAV Task Assignment.* AIAA Guidance, Navigation, and Control Conference, Providence, RI, USA, 16-19 August 2004

[2] MACIEJOWSKI, Jan M.: *Predictive Control with Constraints.* London: Prentice Hall, 2002

[3] RICHARDS, Arthur; HOW, Jonathan P.: *Model Predictive Control of Vehicle Maneuvers with Guaranteed Completion Time and Robust Feasibility.* American Control Conference, Denver, CO, USA, 4-6 June 2003

[4] FREW, Eric W.: *Receding Horizon Control Using Random Search for UAV Navigation with Passive, Non-cooperative Sensing.* AIAA Guidance, Navigation, and Control Conference, San Francisco, CA, USA, August 2005

[5] RICHARDS, Arthur; HOW, Jonathan P.: *Mixed-integer Programming for Control.* American Control Conference, pp. 2676-2683, June 8-10, 2005

[6] LATOMBE, J.-C.: *Robot Motion Planning.* Boston, MA, USA: Kluwer Academic Publishers, 1991, pp. 153-200

[7] CORMEN, Thomas H.: *Introduction to Algorithms.* Cambridge, MA, USA: The MIT Press, 2001, pp. 595-601, 629-635

Authors:
Dipl.-Ing. Jan Willem Vervoorst
Dipl.-Ing. Thomas Kopfstedt
Diehl BGT Defence GmbH & Co. KG, Alte Nußdorfer Str. 13
D-88662 Überlingen, Germany
Phone: +49 7551 89-2449
Fax: +49 7551 89-2351
E-mail: jan.vervoorst@diehl-bgt-defence.de