Planning Formation Changes in Multi-Vehicle Systems based on Discrete-Continuous Linear Optimization

Planung von Formationswechseln in Mehrfahrzeugsystemen durch diskret-kontinuierliche lineare Optimierung

Bachelor-Thesis von Stefanie Bartsch
November 2010

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Informatik
Fachgebiet Simulation, Systemoptimierung und Robotik
Prof. Dr. Oskar von Stryk

Planning Formation Changes in Multi-Vehicle Systems based on Discrete-Continuous Linear Optimization
Planung von Formationswechseln in Mehrfahrzeugsystemen durch diskret-kontinuierliche lineare Optimierung

Vorgelegte Bachelor-Thesis von Stefanie Bartsch

Prüfer: Prof. Dr. Oskar von Stryk
Betreuer: Dr. Christian Reinl

Tag der Einreichung:

Erklärung zur Bachelor-Thesis

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 30. November 2010

(Stefanie Bartsch)

Abstract

The first heterogeneous multi-vehicle path planning framework based on mixed-integer linear programming (MILP) including formation constraints is outlined with the possibility to model formation changes. Since MILP guarantee global optima, this approach suits for benchmarks against heuristic methods or to be embedded in model-predictive control approaches. Existing nonlinear path planning approaches including formation constraints are only locally optimal and therefore provide not necessarily the best solution. Proposals using hybrid system tools suffer from large computational efforts especially without good initial values. MILP solutions can then serve as initial estimates.

The need to maintain and change formations, for instance in surveying large land areas while keeping in range of communication, induces the proposed framework. Further, it is capable to deal with inter-vehicle collision avoidance. A fixed arrival time approach is considered to obtain fuel optimal paths. Formation topologies are modeled using a neighbor-referenced approach and are appended as constraints to the MILP.

Two distinct ways to model pairwise distances are presented. To study systematically their efficiency concerning calculation time and achieved optimum, three example formations are implemented. The *indirect distance determination* is characterized by a strong restriction of the search space. In contrast, there is no limitation of the search space using the *direct distance determination* . Consequently, calculation time increases substantially but the resulting paths require less fuel.

The developed framework, including formation constraints, serves to plan formation changes.

Kurzzusammenfassung

Das erste heterogene Mehrfahrzeug-Planungsproblem basierend auf gemischt-ganzzahliger linearer Optimierung (mixed-integer linear programming, MILP), welches Formationstopologien enthält und die Möglichkeit zur Modellierung von Formationswechseln bietet, wird vorgestellt. Da MILP globale Optima garantieren, eignet sich dieser Ansatz als Benchmark für andere Ansätze wie z.B. Heuristiken oder zur Einbettung in Verfahren der modell-prädikativen Regelung. Bestehende nichtlineare Ansätze zur Bahnplanung unter Einhaltung von Formationstopologien sind dagegen nur lokal optimal und liefern damit nicht notwendigerweise die beste Lösung. Unter Benutzung hybrider Automaten zur Modellierung von Planungsproblemen ist der Rechenaufwand sehr groß, speziell mit ungeeigneten Startwerten. MILP-Lösungen können hier zur Startwertschätzung herangezogen werden.

Die Nachfrage zur Bildung von Formationen und Formationswechseln, z.B. Einhaltung von Funkverbindungen während der kooperativen Landvermessung, führt zu dem hier vorgestellten Framework, welches außerdem Kollisionen zwischen Fahrzeugen behandelt. Dazu wird zunächst ein MILP-Modell mit fester Endzeit hergeleitet zur Berechnung energieoptimaler Bahnen. Formationsbedingungen werden mit Hilfe von paarweisen Distanzen benachbarter Fahrzeuge modelliert und gehen als Nebenbedingungen in das MILP ein. Zwei Möglichkeiten zur Modellierung paarweiser Distanzen werden hergeleitet und analysiert. Für eine systematische Effizienzuntersuchung bezüglich Rechenzeit und erreichtem Optimum werden exemplarisch drei Formationen implementiert. Die *indirekte Distanzvorgabe* ist durch eine starke Einschränkung des Suchraums gekennzeichnet. Dagegen ist der Suchraum der *direkten Distanzvorgabe* nicht limitiert. Dadurch steigt die Rechenzeit wesentlich, aber der Energieverbrauch sinkt.

Das entwickelte Framework mit Formationsnebenbedingungen kann zur Planung von Formationswechseln herangezogen werden.

Contents

1. Introduction **2**
1.1. Motivation . . . 2
1.2. Intention . . . 3
1.3. Outline . . . 4

2. State of research **5**
2.1. Foundations . . . 5
2.1.1. Mixed-integer linear programming . . . 5
2.1.2. Combinatorics . . . 5
2.2. Related work . . . 6
2.2.1. Overview of path planning techniques and optimal cooperative control . . . 6
2.2.2. Formation control . . . 7
2.2.3. Inter-vehicle collision avoidance . . . 8
2.3. Concept . . . 8

3. Mixed-integer modeling of vehicle dynamics and formation conditions **10**
3.1. Modeling of vehicle dynamics . . . 10
3.2. Appending formation conditions using binary variables . . . 11
3.2.1. Indirect distance compliance . . . 11
3.2.2. Direct distance compliance . . . 12

4. Simulation **14**
4.1. Inter-vehicle collision avoidance . . . 14
4.2. Implemented examples of formation changing vehicles . . . 15
4.2.1. Line formation . . . 16
4.2.2. Triangle formation . . . 19
4.2.3. Parallelogram formation . . . 22

5. Numerical results **25**
5.1. Comparison of the MILP-models . . . 25
5.1.1. Comparing indirect and direct distance determination . . . 25
5.1.2. Comparing two ways of holding a formation . . . 28
5.1.3. Example of formation changing vehicles . . . 30

6. Conclusion and outlook **31**

A. Appendix **32**
A.1. Calculation results . . . 32

1 Introduction

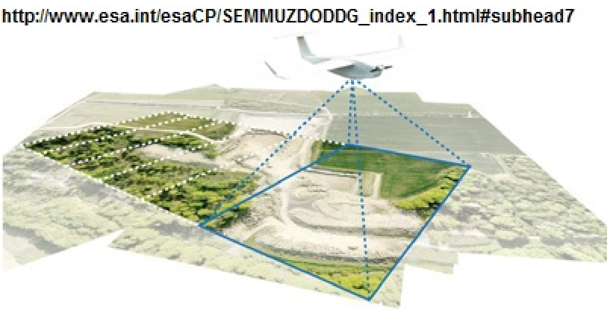
1.1 Motivation

Autonomous mobile vehicles are used in multiple applications, for instance in surveying inaccessible land areas as seen in Figure 1.1(c). Another example are rescue robots aiding rescue workers to reduce personnel requirements and fatigue and also to access unreachable areas.
Experience has shown that in many applications a team of vehicles can fulfill a task more efficiently than a single vehicle. That is why the interest in research and development in cooperative autonomous multi-vehicle systems has grown strongly within the last years. Examples range from applications in mobile sensor networks to transportation systems to terrain mapping [1] to object manipulation [2]. An example for cooperating autonomous helicopters with a ground sensor-actuator wireless network is shown in Figure 1.1(b). A broad overview of the control of cooperating multi-vehicle systems provides Murray [3].
As a motivation, consider the example seen in Figure 1.1(a). Robot soccer is an approved benchmark problem of cooperating autonomous mobile systems since a soccer robot has to navigate in a dynamic world, to act in real time on the basis of incomplete information and to react to unforeseen events. Among others, robot soccer requires sensors to identify objects, motor skills for ball shooting and moving, reactive behavior, self localization and path planning to reach a specific target position.
A second example considers unmanned aerial vehicles (UAVs). As shown in Figure 1.1(c), UAVs can be used to survey land areas such as development areas or disaster zones as well as for observing traffic. However, Richards et al. [4] described that many smaller vehicles are more powerful than a single monolithic vehicle due to the fact that they have a high degree of redundancy and reconfigurability for the case of a single vehicle failure. But more stringent requirements on vehicle coordination, a high-level mission management and fault detection are necessary. On account of this consider a collection of vehicles performing a cooperative task. To fulfill surveillance tasks, for instance, a collection of vehicles is used to maintain a centralized or decentralized description of the state of a land area. Centralized descriptions are based on the assumption that each vehicle of the multi-vehicle system has global knowledge of the coordination information of any other vehicle. The advantage of using this approach is the possibility to compute a global optimal cooperative behavior. However, centralized approaches fail either if communication links are noisy and not persistent or if the central decision component breaks down. Then, decentralized approaches are used instead. In decentralized approaches decisions are taken locally on the basis of local information. Advantages of this approach are the straightforward extensibility with additional vehicles and the compensation of vehicle failures. The passage between centralized and decentralized approaches is fluent, depending on the environment and the cooperative task, since in most cases a combination of local and global path planning is necessary.
In special cases such as formation flying of satellite clusters of UAVs or UAVs surveying larger land areas (see Figure 1.1(d)), it is useful to satisfy formation topologies, for instance directly if UAVs shall deduce an unknown area or indirectly through communication constraints. Corresponding further literature can be extracted from Reinl and Stryk [5]. The satisfaction of formation topologies of multi-vehicle systems leads to an optimization problem and can be modeled approximatively by a mathematical modeling language. In general, an optimization problem consists of an objective function and one or more constraints. The aim is to find an optimum such that the constraints are fulfilled and the objective function takes on a value as minimal as possible. The problem is converted to a Mixed Integer Linear Program (MILP) with vehicle constraints such as maximum velocity, formation constraints and collision avoidance constraints. Optimality of this MILP depends on the requirements in a certain scenario. It can either be optimal with respect to the overall time to fulfill a special task or optimal with respect to the required

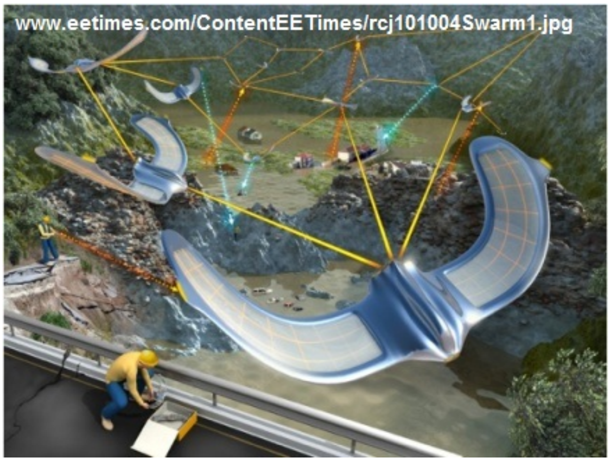
energy to fulfill the task. Further, optimality can be claimed through occuring of certain state changes. Mostly, a combination of these three types of optimality describes the desired system behavior. In this thesis optimality with respect to energy consumption is considered.
For several years, there have been very efficient MILP-solvers on the market. In this thesis, the resulting MILP is implemented in Matlab and solved via an interface by the CPLEX optimization software [6].

(a) Cooperating mobile robots playing soccer

(b) Cooperation of autonomous helicopters with a ground sensor-actuator wireless network



(c) Unmanned micro air vehicle used to survey land areas



(d) Future formation flying vehicles

Figure 1.1.: Applications of autonomous mobile vehicles

1.2 Intention

One approach to solve multi-vehicle cooperative control problems is the use of hybrid system tools as described in Glocker, Reinl and v. Stryk [7]. The advantage of this approach is due to the fact that optimal strategies for complex multi-vehicle problems are determined but the disadvantage is that the problem becomes computationally intensive for large problems. Therefore one needs to find faster techniques. One way to solve optimal control problems is to use MILPs. A key advantage of using MILPs is that there is highly optimized software available.
Within this thesis an existing path-planning algorithm using MILPs is extended by formation topologies.

The maintenance of formation topologies ensures that vehicles are in range of communication for instance.
Thereby the following questions are responded:

1. Which way pairwise distances can be modeled?
2. Out of it, how can formation conditions be deduced?
3. Is there a possibility to change the formation optimally?

For the first question two different approaches to satisfy pairwise distances are introduced. By means of geometrical considerations, formation conditions are deduced using these two approaches. Both approaches are compared with one another with respect to calculation time and the achieved optimum. In addition, two ways of maintaining a formation over several time steps are introduced and compared with one another with respect to calculation time. Then, since the vehicles are heterogeneous, the effect on calculation time due to the use of heterogeneous instead of homogeneous vehicles is studied.
The developed MILP-model including formation constraints can then be used to plan optimal formation changes which answers the last question.

1.3 Outline

Chapter 2 State of research

Some basics are reviewed and path planning techniques are presented. In addition, collision avoidance techniques are compared with one another and the concept of this thesis is outlined.

Chapter 3 Derivation of a MILP - model

This chapter shows how to convert the nonlinear optimal control problem into a mixed integer linear program by means of introduced binary variables and the Big-M-method. Primarily the linearization of the objective is introduced and the dynamic of the vehicles is modeled. After that two different approaches to satisfy pairwise distances are presented - the indirect and direct distance determination.

Chapter 4 Simulation using Matlab

Formation conditions are formulated to gain line, triangle and parallelogram formations, whereas the choice of the vehicles can be either fixed or calculated by optimization. Furthermore, a method to avoid collisions between vehicles is presented.

Chapter 5 Numerical results

Test cases are created to compare the direct and indirect distance determination derived in Chapter 3 and to compare two ways of keeping a formation over several time steps. Moreover the effect on calculation time due to the use of heterogeneous vehicles is studied. An example of formation changing vehicles is presented.

Chapter 6 Conclusion and outlook

The most important results achieved in this thesis are summarized and conclusion out of this are drawn.

2 State of research

2.1 Foundations

2.1.1 Mixed-integer linear programming

A Mixed Integer Linear Program (MILP) [8] is a special case of a Linear Program (LP) in which some of the variables are restricted to take only integer values. It is an NP-complete problem [9], thus a solution in polynomial time is unlikely to exist. The calculation time increases with the number of binary variables, in the worst case exponentially. However, there are several successful solvers.
Given a vector $c \in \mathfrak{R}^n$, a matrix $A \in \mathfrak{R}^{m\times n}$, a vector $b \in \mathfrak{R}^m$ and a number $p \in \{0, 1, \ldots, n\}$ a MILP reads as follows:

$$\begin{aligned} \max\, & c^T \cdot x \\ & A \cdot x \leq b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned} \tag{2.1}$$

There are three special cases: If $p = 0$, eq. (2.1) reduces to an Linear Program (LP). LPs are a special case of convex optimization and are theoretically solvable in polynomial time using the ellipsoid method [10] or interior point methods [11]. Solving an LP using the simplex method requires exponential time in worst case. If $p = n$ one gets a pure integer program. If $p = n$ and $x \in \{0, 1\}^n$ the program is called binary linear program.
A MILP-problem can be solved using a branch-and-bound algorithm. Branch-and-bound is a search technique. In the branch-step the MILP is divided into subsets step-by-step. With suitable bounds suboptimal subsets are expected to be found early and to be eliminated. One expects that this limits the search space to a manageable number of solutions.

2.1.2 Combinatorics

	Variation (with regard to the order)	**Combination** (without regard to the order)	**Permutation**
With repetition	n^k	$\frac{(n+k-1)!}{k!\cdot(n-1)!} = \binom{n+k-1}{k}$	$\frac{n!}{l_1!\cdot l_2!\cdot\ldots\cdot l_k!}$
Without repetition	$\frac{n!}{(n-k)!} = \binom{n}{k} \cdot k!$	$\frac{n!}{k!\cdot(n-k)!} = \binom{n}{k}$	$n!$

Table 2.1.: Combinatorics

Combinatorics is a part of mathematics which deals with the determination of the numbers of possible arrangements or with choices of distinguishable or indistinguishable objects with or without regard to the order. Let n denote the number of possible arrangements and k the number of choices. The resulting possibilities of combinations are shown in Table 2.1.2.
Choosing with regard to the order is called variation, that means that $\{a, b\} \neq \{b, a\}$ for two elements a and b. In contrast, choosing without regard to the order (combination) $\{a, b\} = \{b, a\}$ for two elements a and b. A permutation is any possible arrangement of n elements in which all elements are used. First, consider n distinguishable elements with regard to the order. There are $n!$ possible arrangements. Now,

consider objects of multiple classes with regard to the order. For the number of possible arrangements of objects of multiple classes which are indistinguishable within a class, the number of permutations is the number of the possible arrangements of the distinguishable objects devided by the number of the possible arrangements of the indistinguishable objects.

2.2 Related work

2.2.1 Overview of path planning techniques and optimal cooperative control

A path is a continuous, spatial sequence of points, for example the path of a celestial body or the path of an endeffector [12]. A path that also considers time constraints, i.e. continuous position, velocity and acceleration values, is called trajectory.
Path planning for multi-vehicle systems is a challenging field of research since it is used in many areas including robotics and navigation. The basic path planning problem yields to determine a path from a certain dynamic state to a final state under constraints like moving without inter-vehicle collisions, avoiding stationary and moving obstacles and satisfying kinematic constraints of the vehicles. A more general problem also considers heterogeneous vehicles and multiple waypoint path planning while the order of the set of waypoints is chosen within the optimization.
Finding optimal solutions to path planning problems is intrinsically hard, since the set of feasible solutions is non-convex. Early path planning proposals are, for instance, graph searching or surface covering [13]. Later on, randomized algorithms like in Barraquand et al. [14] are presented. The use of potential functions for path planning is described, for example, in McQuade, Ward and McInnes [15]. However, computing potential functions free of local minima is very hard. Moreover, it can not be guaranteed that the resulting paths are collision-free.
One of the first proposals approximating a nonlinear path planning control problem by using a combination of linear and integer programming is the proposal of Schouwenaars et al. [16]. They considered a basic path planning problem including obstacle and inter-vehicle avoidance using a fixed arrival time approach and compared the fixed arrival time approach to receding horizon strategies.
To transform a multi-vehicle control problem into a MILP, the environment, the vehicle dynamics and constraints and the objective have to be modeled. Since MILPs are in the NP-hard computation class they may not be fast enough for real-time control. But MILP methods guarantee to find the globally-optimal solution to the control problem and therefore offers a benchmark against other approaches.
Receding horizon strategies deal with composing a sequence of locally optimal segments to a path. For this, consider a MILP solving N future time steps for a time i. These N time steps now serve as new input commands. The process is then repeated at time $i+1$. The key problems of receding horizon strategies are that the resulting path may not be globally optimal and the inherent computational complexity of the optimization problem.
The fixed arrival time approach discussed in the proposal of Schouwenaars et al. [16] belongs to the so called standard feedback solution. It provides the possibilities to minimize the overall time, the fuel consumption using a fixed arrival time or both. Schouwenaars et al. came to the conclusion that the path computed using receding horizon requires more fuel, since it is only locally optimal on every segment, than the path calculated using a fixed arrival time approach. However, dependent on the horizon length N, the solution using the receding horizon approach can be calculated much faster since the number of variables and constraints increases polynomially with the horizon length.
Instead of calculating fuel-optimal paths, Richards and How [17] as well as Richards, Bellingham, Tillerson and How [18] solved a more general problem minimizing the overall time. They presented a centralized MILP approach including multiple waypoint path planning of heterogeneous vehicles. This means, each vehicle is required to visit a number of waypoint, whereas the order of those visits is selected within the optimization to minimize the overall time. Assume that for all permutations of visits the overall time

could be calculated. Then it is simple to choose the minimum time path. However, this is intrinsically computational expensive and therefore not feasible when applying in practice. For solving this problem, Richards et al. presented two methods. The first one is a MILP approach, the second one is an approximative method which simplifies the coupling between the assignment and trajectory design problem. It relies on the fact that it is very often the case that the shortest vehicle path between two positions is a straight line. All in all, the approximative method offers much faster solution times than the MILP approach but results could only be sub-optimal.
Since MILP methods suffer computationally for large problems, Earl and D'Andrea [19] presented iterative MILP methods for obstacle avoidance and minimum time trajectory generation that require less binary variables than standard MILP methods.
Reinl and von Stryk [5] formulated a multi-vehicle optimal cooperative control system first as a nonlinear hybrid optimal control problem (HOCP) and then they transformed the HOCP to a MILP since MILPs can be solved much more efficiently. Furthermore HOCP delivers just a local optimum instead of a global optimum using MILP.

2.2.2 Formation control

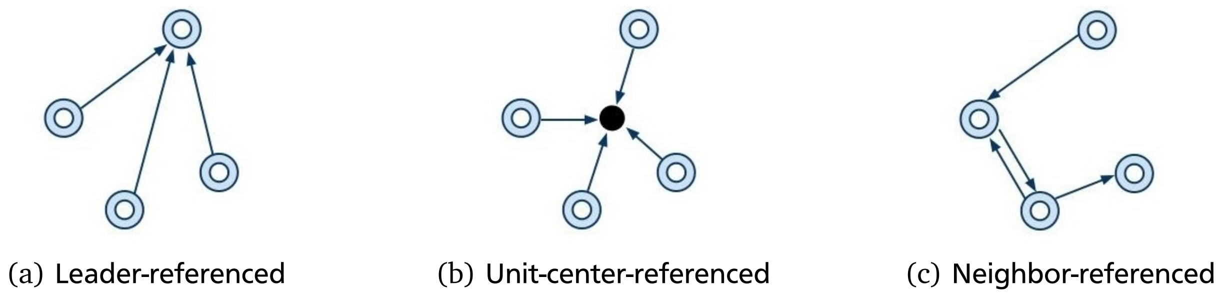


Figure 2.1.: Referencing techniques to determine a formation position

In formation control a group of vehicles set up and maintain predetermined geometric relationships (e.g. line, triangle, parallelogram) by controlling the relative formation positions between the involved vehicles while the group is moving. To maintain a formation, each vehicle in the formation has to determine its position based on the positions of the other vehicles. To this end, Balch and Arkin [20] identified three techniques to formation position determination illustrated in Figure 2.1:

1. *Leader-referenced*: Each vehicle determines its own formation position in relation to the leading vehicle. This means, the leading vehicle is not responsible to maintain formation but the other vehicles from the group.
2. *Unit-center-referenced*: Each vehicle determines its formation position in relation to the unit-center. The unit-center is computed independently by each vehicle by averaging the distances between all involved vehicles.
3. *Neighbor-referenced*: Each vehicle determines its formation position relative to its neighbor vehicle.

There have been numerous approaches proposed to implement formation control of a group of vehicles or robots. Wang [21] evolved a strategy for robot formations using the leader- and neighbor-reference. Chen and Luh [22] considered the generation of formations using distributed control. In this context it was demonstrated, how large groups of robots move cooperatively in various formations but without providing obstacle avoidance. The work of Balch and Arkin described geometric formations similarly but with providing obstacle avoidance. Moreover, they presented a behavior-based, decentralized control architecture to maintain a formation of a group of vehicles while using the unit-center-referenced and leader-referenced approach shown in Figure 2.1. One result of their work is that on the one hand

the unit-center approach provides better performance than the leader-referenced approach, on the other hand the unit-center-referenced approach requires a transmitter and receiver in each vehicle, whereas the leader-referenced approach only requires one transmitter for the leader and one receiver for each other vehicle. Other presented solutions which follow up with the leader-referenced approach are, for instance, the approaches of Parker [23] and Desai, Ostrowski and Kumar [24].
Egerstedt and Hu [25] proposed a model independent coordination strategy for multi-agent formation control. The main idea thereby is that the task is not specified with respect to a physical robot leader but to a virtual leader. Formations are then defined with respect to the real robots and the virtual leader. If then the virtual leader moves on a reference path to a non-physical predetermined point, the real robots should maintain a formation. The advantage of this approach is that robot dynamics can be ignored explicitly at the planning stage. A framework for planning and control of formations for multi-vehicle systems in a dynamic environment with respect to this idea provide Hao, Laxton, Agrawal, Lee and Benson [26]. To get a feasible shortest path for the virtual leader in a given map, standard graph search algorithms can be used. Here, Dijkstra's Algorithm is used to generate a set of waypoints the virtual leader has to traverse. A trajectory generator now produces a smooth trajectory for the virtual leader. The relationships between the group of vehicles and the virtual leader define the reference trajectory for each vehicle based on the reference trajectory of the virtual leader. In order to avoid obstacles or to pass constrictions the formation is allowed to dynamically change. To this end, they used a conservative approach: the entire formation is treated as a point, while the obstacles are enlarged. The main result of their work is that planning a trajectory via a virtual leader reduces the computational complexity significantly which allows online planning of formation paths.
Other approaches to perform a formation are using potential functions. In the presented solution of McQuade, Ward and McInnes [15] a formation is maintained if the potential function reaches a minimum. A proposed method to solve a formation stabilization problem by Dunbar and Murray [27], combines model predictive control with potential functions, but the resulting trajectories are not guaranteed to be collision-free.

2.2.3 Inter-vehicle collision avoidance

While multiple vehicles are moving to different waypoints, collision avoidance between the vehicles is necessary. In conservative approaches, vehicles are treated as points while obstacles are enlarged. Another common approach is the use of potential functions to avoid collisions as described in Latombe [28] but calculate a potential function free of local minima is computationally very hard. Other approaches like in Barraquand et. al [14] developed randomized algorithms to avoid collision. In [16] a method is proposed in which every pair of vehicles p and q is a minimum distance apart from each other in the x- or y- coordinate.

2.3 Concept

In this thesis a planning problem for heterogenous vehicles is extended with formation constraints with the aim to cope with formation changes. Since the originally nonlinear optimal control problem becomes computationally hard for large problems, the optimal control problem is converted into a MILP. Although MILPs belong to the class of $NP-$complete problems, they offer a lot of advantages.
A nonlinear objective function describing the minimization of fuel consumption is piecewise linearized. This means the arrival time is fixed. Nonlinear conditions to model formation topologies are appended as constraints to the optimal control problem. To model formation topologies a neighbor-referenced approach is used. Two approaches to model pairwise distances are derived and analyzed. The first one, the *indirect distance determination*, is dependent on the position of the coordinate systems but using

this one to model formations requires less computation time than the second approach. The second approach, the *direct distance determination*, is independent on the position of the coordinate sytems, but much more binary variables are required to model a formation.
For reformulating Boolean terms arosen from the modeling process, binary variables are introduced. The arising logical relations are reformulated to linear inequalities using the Big-M-method. Then, three example formations are implemented to study systematically their efficiency concerning calculation time and achieved optimum. The developed framework, including formation constraints, can be used to plan formation changes.

3 Mixed-integer modeling of vehicle dynamics and formation conditions

3.1 Modeling of vehicle dynamics

The basic path planning problem considered in this thesis deals with calculating an energy-optimal path between a certain dynamic state and a final state under formation and inter-vehicle collision constraints. In this section a simplified linear motion model is introduced. The vehicles are modeled as point masses with unit mass moving in 2-D. Furthermore a discrete time grid is used with fixed end time. That implies that the considered area is restricted. A vehicle can move from one grid point to an adjacent grid point, as long as that grid point is feasible.
The aim is to find a trajectory with a minimum amount of energy used. That means that the objective functional $J[u]$ is the sum over all time steps $k \in (1, ..., N)$ with a fixed number N of the absolute value of the acceleration:

$$\min J[u] = \min \sum_{k=1}^{N} \left\{ \left| u_k \right| \right\} \tag{3.1}$$

This non linear objective function can be linearized by introducing the variable q such that holds:

$$\min \sum_{k=1}^{N} q_k \tag{3.2}$$

$$-u_k \leq q_k, \;\; u_k \leq q_k \tag{3.3}$$

In addition, initial conditions for the positions and the velocities of the vehicles and end conditions are added since the vehicles should have velocity zero in the given end position. These conditions are simple bounds. t_f denotes the final time.

$$x_0 = x0 \quad x_{t_f} = xt_f \tag{3.4}$$

$$v_0 = v0 \quad v_{t_f} = vt_f \tag{3.5}$$

There are physical constraints of the vehicles like a minimum and maximum possible velocity and a minimum and maximum possible acceleration at each time step. The velocity and acceleration at each time step can also be constraint to fulfill safety regulations. The reachable area of the vehicles is also restricted. All these constraints are considered as simple box constraints. ub denotes the upper bound, lb the lower bound.

$$x_{lb} \leq x \leq x_{ub} \tag{3.6}$$

$$v_{lb} \leq v \leq v_{ub} \tag{3.7}$$

$$u_{lb} \leq u \leq u_{ub} \tag{3.8}$$

The dynamic is numerically approximated by using the Euler method. T_s denotes the step size:

$$\ddot{x} = u \quad \Longleftrightarrow \quad x_{k+1} = x_k + T_s \cdot v_k \tag{3.9}$$

$$v_{k+1} = v_k + T_s \cdot u_k \tag{3.10}$$

In the next section this model will be augmented by formation conditions.

3.2 Appending formation conditions using binary variables

In this section geometric formations will be modeled by pairwise distance dependent structures. This will be done by adding distance dependent constraints for each vehicle, which relate the position i of one vehicle with position j of another vehicle. These distance relations are nonlinear but nonlinear distances can be linearized and then transformed to linear constraints by using the Big-M-method.
Now two methods for modeling distances between adjacent vehicles and then making a formation change will be introduced. In the first method, the given distance between pairs of vehicles will be observed indirectly. The second method uses the given distance directly.

3.2.1 Indirect distance compliance

This method deals with splitting the given distance into $x-$ and $y-$direction (denoted by $dist_x$ and $dist_y$). Figure 3.1 illustrates this proceeding.

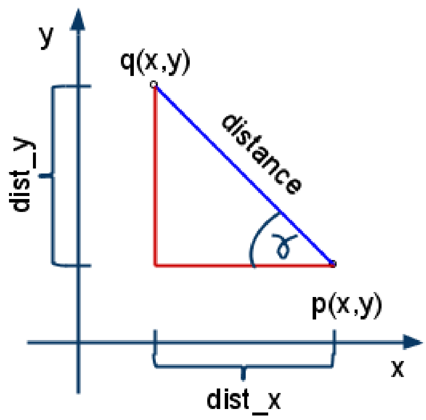


Figure 3.1.: Indirect distance compliance

If the user would like to have a special distance between the vehicles p and q, he has to compute the legs of the right-angled triangle. Since there are many ways to do so, the user has to choose one pair for $dist_x$ and $dist_y$. But then the user also fixes the angle γ between the two vehicles. As a result, there are four possible positions for vehicle q if the position of p is fixed and if $dist_x$ and $dist_y$ are given. The next vehicle has also four possible positions, so that for three vehicles, there are 16 different possibilities for determination of the distance. All in all, if one has n vehicles, there are 4^{n-1} possibilities for the positions of the vehicles satisfying the given distances relative to each other. Thus, the following equations hold:

$$\left[\left|p^x - q^x\right| = dist_x \wedge \left|p^y - q^y\right| = dist_y\right]$$

These equations are non-linear. An equation of this form can be linearized:

$$|p-q| = dist \iff p-q = dist \vee q-p = dist \tag{3.11}$$

To model the disjunction in equality 3.11 two binary variables are introduced. The number m is a small number smaller than $\min(p-q)$. The number M is greater than $\max(p-q)$ for suitable values for p and q. The constant $dist$ describes the distance between the vehicles p and q. Then the resulting system can be transformed into linear constraints using the Big-M-method:

$$\begin{aligned} b_1 = 1 \Rightarrow p-q-dist = 0 \quad &\iff \quad (1-b_1)\cdot m \leq p-q-dist \leq (1-b_1)\cdot M \\ b_2 = 1 \Rightarrow q-p-dist = 0 \quad &\iff \quad (1-b_2)\cdot m \leq q-p-dist \leq (1-b_2)\cdot M \end{aligned}$$

Moreover, $b_1 + b_2 = 1$ is forced, since the considered two vehicles cannot have the same position so that only one condition can be fulfilled. All in all one requires four inequalities, one equality, two binary variables and two constants in each direction to transform the distance of the vehicles p and q to linear conditions with this method.

Likewise the condition that the two vehicles have at least the distance $dist$ can be linearized:

$$\left[\left|p-q\right| \geq dist \quad \Leftrightarrow \quad p-q \geq dist \vee q-p \geq dist\right]$$

Applying the Big-M-method yields:

$$\begin{aligned} b_1 = 1 \Rightarrow p-q-dist \geq 0 \quad &\Leftrightarrow \quad (1-b_1)\cdot m \leq p-q-dist \\ b_2 = 1 \Rightarrow q-p-dist \geq 0 \quad &\Leftrightarrow \quad (1-b_2)\cdot m \leq q-p-dist \end{aligned}$$

Here again $b_1 + b_2 = 1$ is required. Hence, one requires just two inequalities, but also one equality, two binary variables and two constants in each direction to transform the distance of the vehicles p and q to linear conditions with this method. Analogously, the condition that the two vehicles have at most the distance $dist$ can be linearized:

$$\left[\left|p-q\right| \leq dist \quad \Leftrightarrow \quad p-q \leq dist \vee q-p \leq dist\right]$$

Applying again the Big-M-method yields:

$$\begin{aligned} b_1 = 1 \Rightarrow p-q-dist \leq 0 \quad &\Leftrightarrow \quad p-q-dist \leq (1-b_1)\cdot M \\ b_2 = 1 \Rightarrow q-p-dist \leq 0 \quad &\Leftrightarrow \quad q-p-dist \leq (1-b_2)\cdot M \end{aligned}$$

Here again $b_1 + b_2 = 1$ is required. The number of the required variables equals the number of variables in the above case.
In each of the transformations into linear conditions the equality $b_1 + b_2 = 1$ appears. The indirect determination of the distance using this equality is abbreviated with *Ma* or *M1* (*Method a* in line formation or *Method 1* in other formations).
The equality implies that $b_2 = 1 - b_1$. Therefore, all formulas can be expressed with only one binary variable, the other variable will be replaced and the equality is redundant. Since, if $b_1 = 1$ then $b_2 = 0$ and if $b_2 = 1$ then $b_1 = 0$, so the redundant equality always holds. This method is abbreviated with *Mb* (*Method b*) from now on.

3.2.2 Direct distance compliance

Here, the distance is used directly like shown in Figure 3.2. This means, the distance is the Euclidean norm of the positions of the vehicles p and q. Since no angle is determined by the user, there are infinitely many possibilities for vehicle q to have a certain distance from vehicle p. Since the norm is a non linear function, a linear approximation of the norm is introduced like shown in Figure 3.3. The figure shows an approximation with four and eight straight lines. The more straight lines are taken the better is the resulting approximation. Formally, the following holds:

$$\left\| \begin{pmatrix} p^x - q^x \\ p^y - q^y \end{pmatrix} \right\|_2 = dist \tag{3.12}$$

This equation can be approximated by n_γ straight lines $\gamma = (1, ..., n_\gamma)$. Applying the Big-M-method yields:

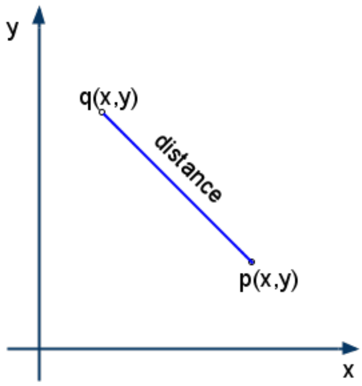


Figure 3.2.: Direct distance compliance

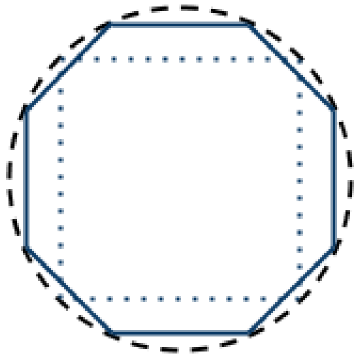


Figure 3.3.: Norm approximation

$$b_i = 1 \ \Rightarrow \ \sin\left(\frac{2\pi\gamma}{n_\gamma}\right) \cdot (p^x - q^x) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right) \cdot (p^y - q^y) = dist \tag{3.13}$$

$$\Leftrightarrow \quad (1 - b_i) \cdot m \leq \sin\left(\frac{2\pi\gamma}{n_\gamma}\right) \cdot (p^x - q^x) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right) \cdot (p^y - q^y) \leq (1 - b_i) \cdot M \tag{3.14}$$

It also has to hold the equation $\sum_{i=1}^{n_g} b_i \geq 1$ therewith the adjacent vehicle is at least on one straight line. The transformation of eq. 3.12 into linear contraints uses two appropiate constants m and M and needs one equality for the binary variables. Furthermore for each straight line, two inequalities and another binary variable are required. The method described above is abbreviated with *M2* (*Method 2*).

4 Simulation

In this simulation vehicles are required to reach different waypoints. It is known from the field of robotics [12] that it is more efficient with respect to the travel time of an industrial robot to use 'fly-by'-zones, that means approaching the waypoint approximatively, instead of approaching the waypoint exactly. In addition the robot arm is then not required to completely decelerate through the control system to satisfy a high positioning accuracy in the waypoint.

In this thesis, the 'fly-by'-zones are realised by creating an $\epsilon-$ neighborhood around the waypoint. Let $\left(x_p, y_p\right)$ denote the position of a vehicle p and $\left(x_w, y_w\right)$ denote the position of a waypoint w . p is in the neighborhood of w in the $x-$ or $y-$ coordinate if the following holds:

$$\begin{aligned} & x_p \leq x_w + \epsilon \\ and \quad & x_p \geq x_w - \epsilon \\ and \quad & y_p \leq y_w + \epsilon \\ and \quad & y_p \geq y_w - \epsilon \end{aligned} \tag{4.1}$$

Transforming these conditions into mixed integer/linear constraints requires introducing a binary variable b. For an appropiate M this results in:

$$\begin{aligned} x_p + M \cdot b &= M + x_w + \epsilon \\ -x_p + M \cdot b &= M - x_w + \epsilon \\ y_p + M \cdot b &= M + y_w + \epsilon \\ -y_p + M \cdot b &= M - y_w + \epsilon \end{aligned} \tag{4.2}$$

For $b = 1$ eq. 4.1 is fulfilled.

4.1 Inter-vehicle collision avoidance

Schouwenaars et al. proposed a collision avoidance method in which every pair of vehicles p and q is a minimum distance apart from each other in the $x-$ and $y-$coordinate. The disadvantage of this technique is founded on the fact that the approximation of the minimum valid distance between two vehicles is broadly defined. Thus, the case can occur that vehicles are supposed to reach a waypoint within an $\epsilon-$neighborhood but yet the $\epsilon-$neighborhood cannot hold all vehicles as shown in Figure 4.1. In this section an approach is proposed which approximates the minimum distance between vehicles using the direct distance determination described in Section 3.2.2. Thereby the approximation of the minimum valid distance between two vehicles is defined more precisely.
Let p_i denote the position of the $i-$th vehicle and p_j the position of the $j-$th vehicle at the $t-$th time step. There are n vehicles and T time steps. Further, *safeDist* denotes the minimum distance required to avoid collisions. Then, the constraints can be formulated as follows:

$$\forall i \in \left[1, \ldots, n-1\right], \forall j : j = i+1, \forall t \in [1, \ldots, T] : \left\| p_i - p_j \right) \right\| \geq safeDist \tag{4.3}$$

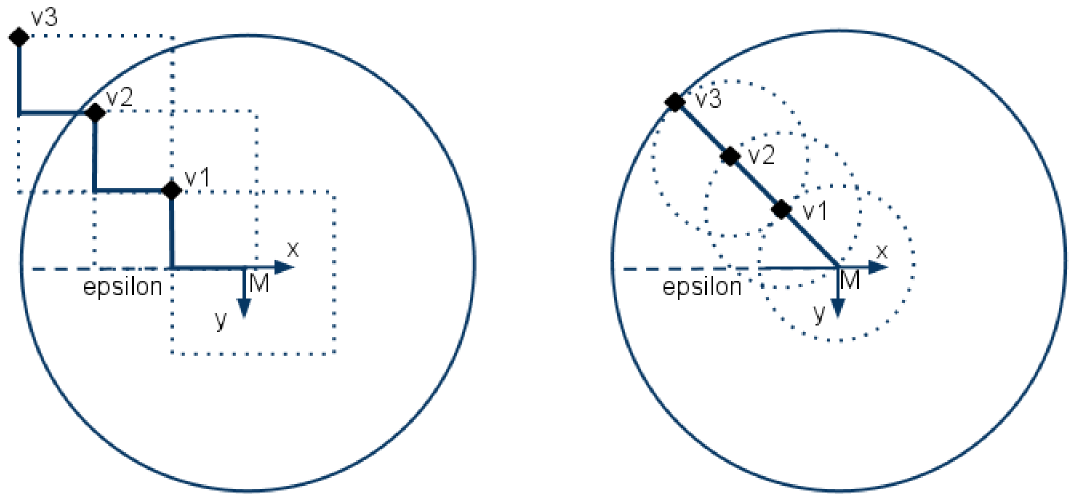


Figure 4.1.: Difference between approximation (left) and exact (right) minimum distance between two vehicles

The norm is approximated using n_γ straight lines:

$$\begin{aligned}&\forall t \in [1,\ldots,T], \forall \gamma \in \left[1,\ldots,n_\gamma\right], \forall i \in [1,\ldots,n-1], \forall j : j = i+1 :\\ &\sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\cdot\left(p_i^x - p_j^x\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\cdot\left(p_i^y - p_j^y\right) \geq safeDist\end{aligned} \tag{4.4}$$

Introducing binary variables b_γ leads to the following mixed integer/linear constraint:

$$\begin{aligned}&\forall t \in [1,\ldots,T], \forall \gamma \in \left[1,\ldots,n_\gamma\right], \forall i \in [1,\ldots,n-1], \forall j : j = i+1 :\\ &\sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\cdot\left(p_i^x - p_j^x\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\cdot\left(p_i^y - p_j^y\right) \geq safeDist - M\cdot b_\gamma\end{aligned} \tag{4.5}$$

To guarantee that at least one b_γ equals zero, the following sum has to hold:

$$\sum_{\gamma=1}^{n_\gamma} b_\gamma \leq n_\gamma - 1 \tag{4.6}$$

From eq. 4.6 it follows that eq. 4.4 holds as well as eq. 4.3.

4.2 Implemented examples of formation changing vehicles

In the following subsections let $o \in [1,\cdots,O]$ denote the row-counter of p and $i \in [1,\cdots,k]$ the column-counter of p with k denoting the number of vehicles participating in the formation. Let further $t \in [1,\cdots,T]$ denote the current time step.

4.2.1 Line formation

To create a line out of k vehicles, the first step is to consider the possibilities to take k vehicles out of n. Since the order of the vehicles is relevant and since every vehicle is unique, it follows that there are $k! \cdot \binom{n}{k}$ possibilities.
Since a neighbor-referenced approach is chosen, the distances between two neighboring vehicles have to be modeled and, to get a straight line, the distance between the first and the last vehicle has to be modeled.

Indirect distance determination

Using the indirect distance determination, the rotation of a straight line is implicitly induced by the choice of the distances in x- and y-direction. Consider at first the case without permutation of the vehicles. Then, the following holds:

$$\left| p_i^x - p_{i+1}^x \right| = dist_x \tag{4.7}$$
$$\left| p_i^y - p_{i+1}^y \right| = dist_y \tag{4.8}$$

One approach to linearize eq. 4.7 requires four binary variables:

$$b_1 = 1 \Leftrightarrow p_i^x - p_{i+1}^x = dist_x \tag{4.9}$$
$$b_2 = 1 \Leftrightarrow p_{i+1}^x - p_i^x = dist_x \tag{4.10}$$
$$b_3 = 1 \Leftrightarrow p_i^y - p_{i+1}^y = dist_y \tag{4.11}$$
$$b_4 = 1 \Leftrightarrow p_{i+1}^y - p_i^y = dist_y \tag{4.12}$$

This approach is abbreviated with *LMa* (*Line Method a*). The other approach requires only two binary variables if $b_1 = b_3$ and $b_2 = b_4$ is set and is abbreviated with *LMb* (*Line Method b*). To build up a line formation without permuting the participating vehicles using these two approaches is left to the reader.
Now consider the possibility to permute vehicles. For this, *LMa* is used. The constraints to model the distances can be formulated as follows:

$$\forall o, \forall i, \forall t: \quad and \begin{cases} \left| p_{o,i}^x - p_{o,i+1}^x \right| = dist_x \\ \left| p_{o,i}^y - p_{o,i+1}^y \right| = dist_y \end{cases} \tag{4.13}$$

These nonlinear constraints are linearized by eliminating the absolute values. This implies that:

$$\forall o, \forall i, \forall t: \quad and \begin{cases} or \begin{cases} p_{o,i}^x - p_{o,i+1}^x = dist_x \\ p_{o,i+1}^x - p_{o,i}^x = dist_x \end{cases} \\ or \begin{cases} p_{o,i}^y - p_{o,i+1}^y = dist_y \\ p_{o,i+1}^y - p_{o,i}^y = dist_y \end{cases} \end{cases} \tag{4.14}$$

To formulate these constraints as mixed integer/linear constraints, binary variables $b^1_{o,i,i+1,t,k}$ are introduced. It now appears, that the following holds for an appropiate M:

$$\forall o, \forall i, \forall t: \quad and \begin{cases} or \begin{cases} and \begin{cases} p^x_{o,i} - p^x_{o,i+1} + M \cdot b^1_{o,i,i+1,t,1} \leq M + dist_x \\ p^x_{o,i+1} - p^x_{o,i} + M \cdot b^1_{o,i,i+1,t,1} \leq M - dist_x \end{cases} \\ and \begin{cases} p^x_{o,i+1} - p^x_{o,i} + M \cdot b^1_{o,i,i+1,t,2} \leq M + dist_x \\ p^x_{o,i} - p^x_{o,i+1} + M \cdot b^1_{o,i,i+1,t,2} \leq M - dist_x \end{cases} \end{cases} \\ or \begin{cases} and \begin{cases} p^y_{o,i} - p^y_{o,i+1} + M \cdot b^1_{o,i,i+1,t,3} \leq M + dist_y \\ p^y_{o,i+1} - p^y_{o,i} + M \cdot b^1_{o,i,i+1,t,3} \leq M - dist_y \end{cases} \\ and \begin{cases} p^y_{o,i+1} - p^y_{o,i} + M \cdot b^1_{o,i,i+1,t,4} \leq M + dist_y \\ p^y_{o,i} - p^y_{o,i+1} + M \cdot b^1_{o,i,i+1,t,4} \leq M - dist_y \end{cases} \end{cases} \end{cases} \tag{4.15}$$

As a second step, the distance between the first and the last vehicle is modeled to receive a straight line in the following way:

$$\forall o, \forall t: \quad and \begin{cases} \left| p^x_{o,1} - p^x_{o,K} \right| = (K-1) \cdot dist_x \\ \left| p^y_{o,1} - p^y_{o,K} \right| = (K-1) \cdot dist_y \end{cases} \tag{4.16}$$

Elimination of the absolute values leads to linearized constraints:

$$\forall o, \forall t: \quad and \begin{cases} or \begin{cases} p^x_{o,1} - p^x_{o,K} = (K-1) \cdot dist_x \\ p^x_{o,K} - p^x_{o,1} = (K-1) \cdot dist_x \end{cases} \\ or \begin{cases} p^y_{o,1} - p^y_{o,K} = (K-1) \cdot dist_y \\ p^y_{o,K} - p^y_{o,1} = (K-1) \cdot dist_y \end{cases} \end{cases} \tag{4.17}$$

To formulate these constraints as mixed integer/linear constraints, binary variables $b^2_{o,i,i+1,t,l}$ are introduced. Hence, it follows for an appropiate M:

$$\forall o, \forall t: \quad and \begin{cases} or \begin{cases} and \begin{cases} p^x_{o,1} - p^x_{o,K} + M \cdot b^2_{o,1,K,t,1} \leq M + (K-1) \cdot dist_x \\ p^x_{o,K} - p^x_{o,1} + M \cdot b^2_{o,1,K,t,1} \leq M - (K-1) \cdot dist_x \end{cases} \\ and \begin{cases} p^x_{o,K} - p^x_{o,1} + M \cdot b^2_{o,1,K,t,2} \leq M + (K-1) \cdot dist_x \\ p^x_{o,i} - p^x_{o,K} + M \cdot b^2_{o,1,K,t,2} \leq M - (K-1) \cdot dist_x \end{cases} \end{cases} \\ or \begin{cases} and \begin{cases} p^y_{o,1} - p^y_{o,K} + M \cdot b^2_{o,1,K,t,3} \leq M + (K-1) \cdot dist_y \\ p^y_{o,K} - p^y_{o,1} + M \cdot b^2_{o,1,K,t,3} \leq M - (K-1) \cdot dist_y \end{cases} \\ and \begin{cases} p^y_{o,K} - p^y_{o,1} + M \cdot b^2_{o,1,K,t,4} \leq M + (K-1) \cdot dist_y \\ p^y_{o,1} - p^y_{o,K} + M \cdot b^2_{o,1,K,t,4} \leq M - (K-1) \cdot dist_y \end{cases} \end{cases} \end{cases} \tag{4.18}$$

For combining the binary variables $b^1_{o,i,i+1,t,k}, k \in [1,\dots,4]$ and $b^2_{o,i,i+1,t,l}, l \in [1,\dots,4]$ the binary variables $b^3_{o,m}, m \in [1,\dots,4]$ and b^4_o are introduced and it follows for an appropiate M:

$$
\begin{aligned}
\forall o, \forall i, \forall t: \quad & -b^1_{o,i,i+1,t,1} - b^1_{o,i,i+1,t,2} + M \cdot b^3_{o,1} \leq M - 1 \\
& b^1_{o,i,i+1,t,1} + b^1_{o,i,i+1,t,2} + M \cdot b^3_{o,1} \leq M + 1 \\
& -b^1_{o,i,i+1,t,3} - b^1_{o,i,i+1,t,4} + M \cdot b^3_{o,2} \leq M - 1 \\
& b^1_{o,i,i+1,t,3} + b^1_{o,i,i+1,t,4} + M \cdot b^3_{o,2} \leq M + 1
\end{aligned}
\tag{4.19}
$$

$$
\begin{aligned}
\forall o, \forall t: \quad & -b^2_{o,1,K,t,1} - b^2_{o,1,K,t,2} + M \cdot b^3_{o,3} \leq M - 1 \\
& b^2_{o,1,K,t,1} + b^2_{o,1,K,t,2} + M \cdot b^3_{o,3} \leq M + 1 \\
& -b^2_{o,1,K,t,3} - b^2_{o,1,K,t,4} + M \cdot b^3_{o,4} \leq M - 1 \\
& b^2_{o,1,K,t,3} + b^2_{o,1,K,t,4} + M \cdot b^3_{o,4} \leq M + 1
\end{aligned}
\tag{4.20}
$$

$$
\begin{aligned}
\forall o: \quad & -b^3_{o,1} - b^3_{o,2} - b^3_{o,3} - b^3_{o,4} + M \cdot b^4_o \leq M - 4 \\
& b^3_{o,1} + b^3_{o,2} + b^3_{o,3} + b^3_{o,4} + M \cdot b^4_o \leq M + 4
\end{aligned}
\tag{4.21}
$$

$$
\sum_{o=1}^{O} b^4_o = 1 \tag{4.22}
$$

The sum in equality 4.22 guarantees that at least one b^4_o, $o \in [1,\dots,O]$ equals one. This implies, that $\exists o$ so that $\sum_{m=1}^{4} b^3_{o,m} = 4$. From this it follows that the $or-$constraints in the equalities 4.15 and 4.18 are fulfilled. The $and-$constraints are implicitly fulfilled by the choice of the binary variables $b^1_{o,i,i+1,t,k}$ and $b^2_{o,1,K,t,l}$.

Direct distance determination

The case without permuation of the vehicles is left to the reader. With permutation of the vehicles results in the following equations, the first one for neighboring vehicles and the second one for the distance between the first and last vehicle:

$$
\begin{aligned}
\forall o, \forall \gamma, \forall i, \forall t: \quad & \left\| \left(p^x_{o,i} - p^x_{o,j}\right), \left(p^y_{o,i} - p^y_{o,j}\right) \right\|_2 \\
& = \sqrt{\left(p^x_{o,i} - p^x_{o,j}\right)^2 + \left(p^y_{o,i} - p^y_{o,j}\right)^2} \\
& \overset{!}{=} dist
\end{aligned}
\tag{4.23}
$$

$$
\begin{aligned}
\forall o, \forall \gamma, \forall t: \quad & \left\| \left(p^x_{o,1} - p^x_{o,K}\right), \left(p^y_{o,1} - p^y_{o,K}\right) \right\|_2 \\
& = \sqrt{\left(p^x_{o,1} - p^x_{o,K}\right)^2 + \left(p^y_{o,1} - p^y_{o,K}\right)^2} \\
& \overset{!}{=} (K-1) \cdot dist
\end{aligned}
\tag{4.24}
$$

To linearize these equations, the norm is approximated using n_γ straight lines:

$$\forall o, \forall \gamma, \forall i, \forall t: \quad and \begin{cases} - \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) - \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M - dist \\ \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M + dist \end{cases} \tag{4.25}$$

$$\forall o, \forall \gamma, \forall t: \quad and \begin{cases} - \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,1} - p^x_{o,K}\right) - \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,1} - p^y_{o,K}\right) + M \cdot b^2_{o,\gamma,t} \leq M - (K-1) \cdot dist \\ \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,1} - p^x_{o,K}\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,1} - p^y_{o,K}\right) + M \cdot b^2_{o,\gamma,t} \leq M + (K-1) \cdot dist \end{cases} \tag{4.26}$$

For combining the binary variables $b^1_{o,\gamma,t}$ and $b^2_{o,\gamma,t}$ the binary variable b^3_o is introduced and it follows with an appropiate M:

$$\forall o, \forall i, \forall t: \quad M \cdot b^3_o - \sum_{\gamma=1}^{n_\gamma} b^1_{o,\gamma,i,j,t} \leq M - 1 \tag{4.27}$$

$$\forall o, \forall t: \quad M \cdot b^3_o - \sum_{\gamma=1}^{n_\gamma} b^2_{o,\gamma,t} \leq M - 1 \tag{4.28}$$

$$\sum_{o=1}^{O} b^3_o = 1 \tag{4.29}$$

The sum in eq. 4.29 guarantees that at least one b^3_o equals one. This implies that $\exists o$ such that eq. 4.25 and eq. 4.26 hold. The line formation using the direct distance determination is abbreviated with *LM2*.

4.2.2 Triangle formation

Like in the last subsection first of all the number of possibilities to choose three vehicles out of n to generate equilateral triangles is considered. In the case of triangles the order of the vehicles is irrelevant, but every vehicle is unique. Therefore, it follows that there are $\binom{n}{3}$ possibilities. In the first method, the rotation of the triangle is implicitly induced by the choice of the distances in x- and y-direction while in the second method the rotation of the triangle is determined by the optimization.

Indirect distance determination

To build up a triangle using the indirect distance determination, appropiate distances in the x- and y-coordinate are required. Let them denote as $dx_i,\ i \in [1,\dots,3]$ and $dy_i,\ i \in [1,\dots,3]$. Now, consider the distance between neighboring vehicles. Modeling triangles without permutation of the vehicles is left to the reader. The constraints to model this distance with permutation of the vehicles can be formulated in the following way:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad and \begin{cases} \left| p^x_{o,i} - p^x_{o,j} \right| = dx_i \\ \left| p^y_{o,i} - p^y_{o,j} \right| = dy_i \end{cases} \tag{4.30}$$

By eliminating the absolute values these nonlinear constraints are linearized. This results in:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad and \begin{cases} or \begin{cases} p^x_{o,i} - p^x_{o,j} = dx_i \\ p^x_{o,j} - p^x_{o,i} = dx_i \end{cases} \\ or \begin{cases} p^y_{o,i} - p^y_{o,j} = dy_i \\ p^y_{o,j} - p^y_{o,i} = dy_i \end{cases} \end{cases} \tag{4.31}$$

To formulate these constraints as mixed integer/linear constraints, binary variables $b^1_{o,i,j,t,k}$ are introduced. Hence it follows for an appropiate M:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad and \begin{cases} or \begin{cases} and \begin{cases} p^x_{o,i} - p^x_{o,j} + M \cdot b^1_{o,i,j,t,1} \leq M + dx_i \\ p^x_{o,j} - p^x_{o,i} + M \cdot b^1_{o,i,j,t,1} \leq M - dx_i \end{cases} \\ and \begin{cases} p^x_{o,j} - p^x_{o,i} + M \cdot b^1_{o,i,j,t,2} \leq M + dx_i \\ p^x_{o,i} - p^x_{o,j} + M \cdot b^1_{o,i,j,t,2} \leq M - dx_i \end{cases} \end{cases} \\ or \begin{cases} and \begin{cases} p^y_{o,i} - p^y_{o,j} + M \cdot b^1_{o,i,j,t,3} \leq M + dy_i \\ p^y_{o,j} - p^y_{o,i} + M \cdot b^1_{o,i,j,t,3} \leq M - dy_i \end{cases} \\ and \begin{cases} p^y_{o,j} - p^y_{o,i} + M \cdot b^1_{o,i,j,t,4} \leq M + dy_i \\ p^y_{o,i} - p^y_{o,j} + M \cdot b^1_{o,i,j,t,4} \leq M - dy_i \end{cases} \end{cases} \end{cases} \tag{4.32}$$

Combining the binary variables $b^1_{o,i,j,t,k}, k \in [1,\dots,4]$ requires new binary variables $b^2_{o,m}, m \in [1,2]$ and b^3_o. Then it follows with an appropiate M:

$$\begin{aligned} \forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad -b^1_{o,i,j,t,1} - b^1_{o,i,j,t,2} + M \cdot b^2_{o,1} &\leq M - 1 \\ b^1_{o,i,j,t,1} + b^1_{o,i,j,t,2} + M \cdot b^2_{o,1} &\leq M + 1 \\ -b^1_{o,i,j,t,3} - b^1_{o,i,j,t,4} + M \cdot b^2_{o,2} &\leq M - 1 \\ b^1_{o,i,j,t,3} + b^1_{o,i,j,t,4} + M \cdot b^2_{o,2} &\leq M + 1 \end{aligned} \tag{4.33}$$

$$\forall o: \quad \begin{aligned} -b^2_{o,1} - b^2_{o,2} + M \cdot b^3_o &\leq M - 2 \\ b^2_{o,1} + b^2_{o,2} + M \cdot b^3_o &\leq M + 2 \end{aligned} \tag{4.34}$$

$$\sum_{o=1}^{O} b^3_o = 1 \tag{4.35}$$

The sum in equality 4.35 guarantees that at least one b^3_o, $o \in [1, \ldots, O]$ equals one. This implies, that $\exists o$ so that $b^2_{o,1} + b^2_{o,2} = 2$. From this it follows that the $or-$constraints in equality 4.32 are fulfilled. The $and-$constraints are implicitly fulfilled by the choice of the binary variables $b^1_{o,i,j,t,k}$. Triangles using the indirect distance determination are denoted by *TM1*.

Direct distance determination

Triangles using the direct distance determination are denoted by *TM2*. Again, the case without permutation of the vehicles is left to the reader. With permutation of the vehicles results in the following equation:

$$\begin{aligned} \forall o, \forall \gamma, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad & \left\| \left(p^x_{o,i} - p^x_{o,j}\right), \left(p^y_{o,i} - p^y_{o,j}\right) \right\|_2 \\ & = \sqrt{\left(p^x_{o,i} - p^x_{o,j}\right)^2 + \left(p^y_{o,i} - p^y_{o,j}\right)^2} \\ & \stackrel{!}{=} dist \end{aligned} \tag{4.36}$$

To linearize this equation, the norm is approximated using n_γ straight lines:

$$\begin{aligned} & \forall o, \forall \gamma, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \\ & and \begin{cases} - & \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) - \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M - dist \\ & \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M + dist \end{cases} \end{aligned} \tag{4.37}$$

To get one triangle the binary variable b^2_o is introduced and it follows with an appropiate M:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad M \cdot b^2_o - \sum_{\gamma=1}^{n_\gamma} b^1_{o,\gamma,i,j,t} \leq M - 1 \tag{4.38}$$

$$\sum_{o=1}^{O} b^2_o = 1 \tag{4.39}$$

Eq. 4.39 ensures that the triangle formation holds.

4.2.3 Parallelogram formation

To create a parallelogram out of four vehicles, the first step is to consider the possibilities to take four vehicles out of n. Since the order of the vehicles is relevant but the first vehicle is connected with the last vehicle and since every vehicle is unique, it follows that there are $3 \cdot \binom{n}{k}$ possibilities.

Indirect distance determination

To build up a parallelogram, appropiate distances in the x- and y- coordinate are required. Let them denote as $dx_i,\ i \in [1,\dots,4]$ and $dy_i,\ i \in [1,\dots,4]$. Now, consider the distance between neighboring vehicles. Modeling parallelograms without permutation of the vehicles is again left to the reader. The constraints to model this distance with permutation of the vehicles can be formulated in the following way:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 4 : \quad and \begin{cases} \left| p^x_{o,i} - p^x_{o,j} \right| = dx_i \\ \left| p^y_{o,i} - p^y_{o,j} \right| = dy_i \end{cases} \tag{4.40}$$

By eliminating the absolute values these nonlinear constraints are linearized. This results in:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 4 : \quad and \begin{cases} or \begin{cases} p^x_{o,i} - p^x_{o,j} = dx_i \\ p^x_{o,j} - p^x_{o,i} = dx_i \end{cases} \\ or \begin{cases} p^y_{o,i} - p^y_{o,j} = dy_i \\ p^y_{o,j} - p^y_{o,i} = dy_i \end{cases} \end{cases} \tag{4.41}$$

To formulate these constraints as mixed integer/linear constraints, binary variables $b^1_{o,i,j,t,k}$ are introduced. Hence it follows for an appropiate M:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad and \begin{cases} or \begin{cases} and \begin{cases} p^x_{o,i} - p^x_{o,j} + M \cdot b^1_{o,i,j,t,1} \leq M + dx_i \\ p^x_{o,j} - p^x_{o,i} + M \cdot b^1_{o,i,j,t,1} \leq M - dx_i \end{cases} \\ and \begin{cases} p^x_{o,j} - p^x_{o,i} + M \cdot b^1_{o,i,j,t,2} \leq M + dx_i \\ p^x_{o,i} - p^x_{o,j} + M \cdot b^1_{o,i,j,t,2} \leq M - dx_i \end{cases} \end{cases} \\ or \begin{cases} and \begin{cases} p^y_{o,i} - p^y_{o,j} + M \cdot b^1_{o,i,j,t,3} \leq M + dy_i \\ p^y_{o,j} - p^y_{o,i} + M \cdot b^1_{o,i,j,t,3} \leq M - dy_i \end{cases} \\ and \begin{cases} p^y_{o,j} - p^y_{o,i} + M \cdot b^1_{o,i,j,t,4} \leq M + dy_i \\ p^y_{o,i} - p^y_{o,j} + M \cdot b^1_{o,i,j,t,4} \leq M - dy_i \end{cases} \end{cases} \end{cases} \tag{4.42}$$

Combining the binary variables $b^1_{o,i,j,t,k}, k \in [1,\dots,4]$ requires new binary variables $b^2_{o,m}, m \in [1,2]$ and b^3_o. Then it follows with an appropiate M:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 3 : \quad \begin{aligned} -b^1_{o,i,i+1,t,1} - b^1_{o,i,i+1,t,2} + M \cdot b^2_{o,1} &\le M - 1 \\ b^1_{o,i,i+1,t,1} + b^1_{o,i,i+1,t,2} + M \cdot b^2_{o,1} &\le M + 1 \\ -b^1_{o,i,i+1,t,3} - b^1_{o,i,i+1,t,4} + M \cdot b^2_{o,2} &\le M - 1 \\ b^1_{o,i,i+1,t,3} + b^1_{o,i,i+1,t,4} + M \cdot b^2_{o,2} &\le M + 1 \end{aligned} \tag{4.43}$$

$$\forall o : \quad \begin{aligned} -b^2_{o,1} - b^2_{o,2} + M \cdot b^3_o &\le M - 2 \\ b^2_{o,1} + b^2_{o,2} + M \cdot b^3_o &\le M + 2 \end{aligned} \tag{4.44}$$

$$\sum_{o=1}^{O} b^3_o = 1 \tag{4.45}$$

The sum in equality 4.45 guarantees that at least one b^3_o, $o \in [1,\dots,O]$ equals one. This implies, that $\exists o$ so that $b^2_{o,1} + b^2_{o,2} = 2$. From this it follows that the $or-$constraints in equality 4.42 are fulfilled. The $and-$constraints are implicitly fulfilled by the choice of the binary variables $b^1_{o,i,j,t,k}$. Parallelograms using the indirect distance determination are denoted by *PM1*.

Direct distance determination

Parallelograms using the direct distance determination are denoted by *PM2*. Again, the case without permutation of the vehicles is left to the reader. With permutation of the vehicles results in the following equation:

$$\begin{aligned} \forall o, \forall \gamma, \forall i, \forall t : \forall j : j = i + 1 mod 4 : \quad & \left\| \left(p^x_{o,i} - p^x_{o,j}\right), \left(p^y_{o,i} - p^y_{o,j}\right) \right\|_2 \\ &= \sqrt{\left(p^x_{o,i} - p^x_{o,j}\right)^2 + \left(p^y_{o,i} - p^y_{o,j}\right)^2} \\ &\stackrel{!}{=} dist \end{aligned} \tag{4.46}$$

Since vehicles facing each other have to be at least $\frac{dist}{2}$ apart, the following has to hold:

$$\begin{aligned} \forall o, \forall \gamma, \forall i : 1 \le i \le 2, \forall t : \quad & \left\| \left(p^x_{o,i} - p^x_{o,i+2}\right), \left(p^y_{o,i} - p^y_{o,i+2}\right) \right\|_2 \\ &= \sqrt{\left(p^x_{o,i} - p^x_{o,i+2}\right)^2 + \left(p^y_{o,i} - p^y_{o,i+2}\right)^2} \\ &\stackrel{!}{\ge} \frac{dist}{2} \end{aligned} \tag{4.47}$$

To linearize the equation and the inequality, the norm is approximated using n_γ straight lines whereby two binary variables $b^1_{o,\gamma,i,j,t}$ and $b^2_{o,\gamma,t}$ are introduced:

$$\forall o, \forall \gamma, \forall i, \forall t : \forall j : j = i + 1 mod 4 :$$
$$and \begin{cases} - \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) - \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M - dist \\ \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) + \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^1_{o,\gamma,i,j,t} \leq M + dist \end{cases} \quad (4.48)$$

$$\forall o, \forall \gamma, \forall i : 1 \leq i \leq 2, \forall t :$$
$$- \sin\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^x_{o,i} - p^x_{o,j}\right) - \cos\left(\frac{2\pi\gamma}{n_\gamma}\right)\left(p^y_{o,i} - p^y_{o,j}\right) + M \cdot b^2_{o,\gamma,t} \leq M - \frac{dist}{2} \quad (4.49)$$

To obtain one parallelogram the binary variable b^3_o is introduced and it follows with an appropiate M:

$$\forall o, \forall i, \forall t : \forall j : j = i + 1 mod 4 : \quad M \cdot b^3_o - \sum_{\gamma=1}^{n_\gamma} b^1_{o,\gamma,i,j,t} \leq M - 1 \quad (4.50)$$

$$\forall o, \forall t : \quad M \cdot b^3_o - \sum_{\gamma=1}^{n_\gamma} b^2_{o,\gamma,t} \leq M - 1 \quad (4.51)$$

$$\sum_{o=1}^{O} b^3_o = 1 \quad (4.52)$$

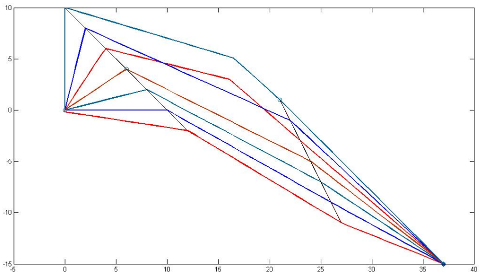
Eq. 4.52 ensures that the parallelogram formation holds.

5 Numerical results

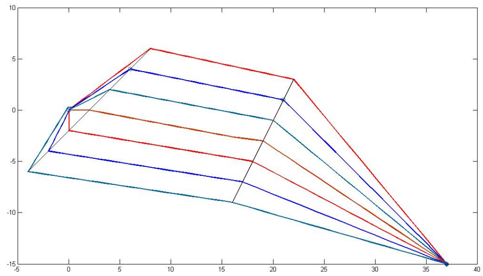
In this chapter the average case computational complexity and average optima of the MILP algorithm is performed by solving randomly generated problem instances using a uniform distribution. For this purpose, the MILP-solver implemented in the *CPLEX* software package [6] is used via *CPLEXINT*. This is a *Matlab*-interface for the *CPLEX* solver. *Matlab*, short for matrix laboratory, is a commercial software package from *The MathWorks*. It provides an environment for high-level programming language. Its power lies in numerical calculations, especially with vectors and matrices.

5.1 Comparison of the MILP-models

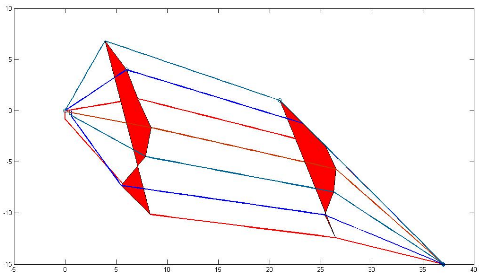
5.1.1 Comparing indirect and direct distance determination



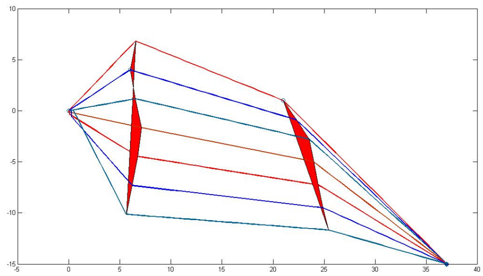
(a) LMa



(b) LMb



(c) LM2 with 4 straight lines



(d) LM2 with 16 straight lines

Figure 5.1.: Comparing the introduced approaches for modeling pairwise distances in reference to the line formation

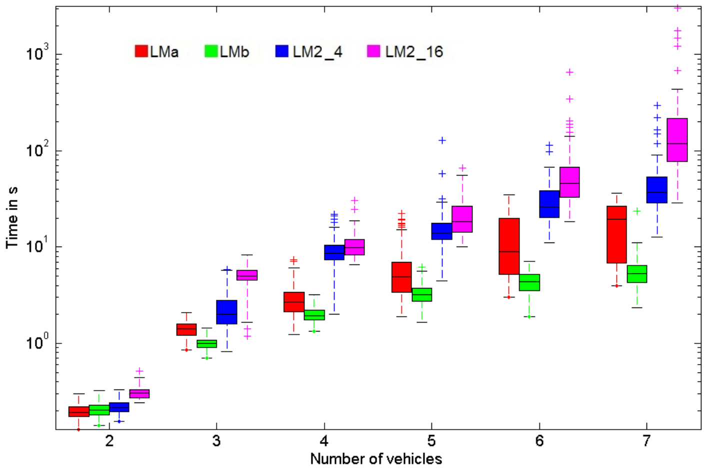
Test arrangement

In this section the *direct* and *indirect distance determination* is evaluated. For this purpose consider the two different approaches *LMa* and *LMb* for modeling pairwise distances in the case of the *indirect distance determination* for the line formation discussed in Subsection 4.2.1 on Page 16 and the approach for the *direct distance determination*. The last one is evaluated with four (*LM2_4*) and 16 (*LM2_16*) straight lines. In all of the four cases consider homogeneous vehicles without permutation of the vehicles.
In this test, as one can see in Figure 5.1, all vehicles are starting in the origin. Then they are supposed

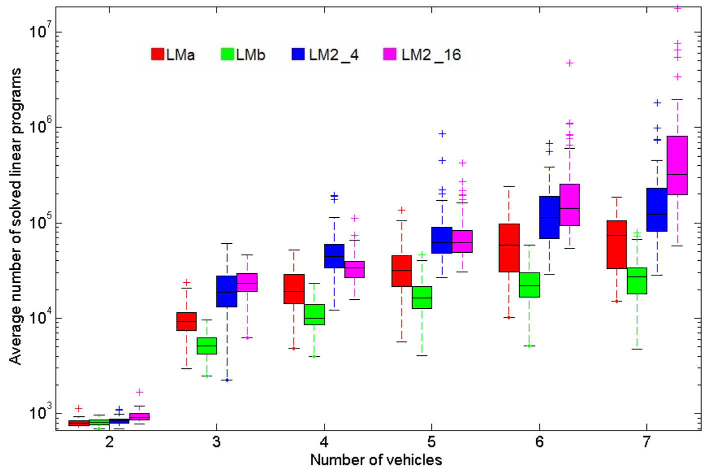
to reach a waypoint after 13 time steps while holding a line formation. Afterwards they shall reach a second waypoint after 37 time steps while again holding a line formation. In the end they are expected to reach a final destination after 50 time steps.
To get statistical relevant assertions the following is performed 100 times: In the first step choose a random waypoint sufficiently near the starting point. Then choose a second random waypoint sufficiently near the first one. The final destination is also choosen randomly and has to be sufficiently near the second waypoint. Then, in the first case, perform a line formation with approach a) from subsection 4.2.1. Afterwords, with the same way- and end points, perform a line formation with approach b) from subsection 4.2.1. In the third case the *direct distance determination* with four straight lines is used to model the pairwise distances in the line formation. In the last case there are 16 straight lines used instead.
In order to compare not only the average time and the number of solved linear programs in this evaluation, the random waypoints are fixed for one iteration. Furthermore the distances between pairwise vehicles have to be equal. Therefore in this evaluation the first line formation has distance $dist = \sqrt{8}$ between adjacent vehicles. This is directly the distance used in the *direct distance determination*. Using the Pythagorean theorem this distance is among others equal to $\sqrt{8} = \sqrt{2^2 + 1^2}$. Therefore the indirect distances are $dist_x = 2$ and $dist_y = 2$. The second line formation is expected to have the distance $dist = \sqrt{5}$. From this it follows that $dist_x = 1$ and $dist_y = 2$.
Now, also the calculated optima of the four considered line formations can be compared. All in all, this test is carried out for two to seven vehicles.

Interpretation

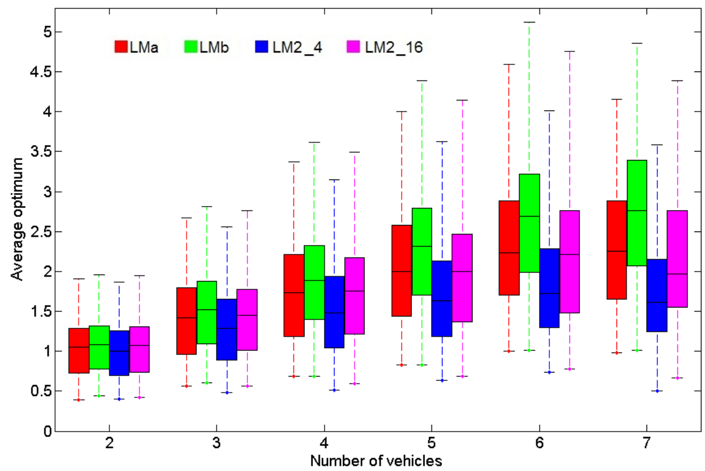
All of the four considered methods are generating line formations with the same distance between adjacent vehicles, but the difference regarding the computation time is tremendous. Since the search space in the case of *LMb* is limited strongest, the computation time, shown in figure 5.2(a), is the lowest compared to the other three methods. As shown in subsection 4.2.1, the search space of *LMa* is also very restricted, but less than for *LMb*. Therefore, the average computation time is higher. Since the search space of the line formation using the *direct distance determination* is the largest, both *LM2* with four and 16 lines lack the longest average computation time.
Except the last value of *LM2_4* in figure 5.2(b), the average time and the average number of solved linear programs increase potentially in the number of vehicles. As shown in Figure 5.3 the calculation time is linear dependent from the number of solved linear programs. Applying a regression line leads to the assumption, that the algorithm executing a line formation is in $O(n^2)$ using the indirect distance determination and in $O(n^3)$ using the direct distance determination.
All in all it is shown in this computational complexity study that solving this multi-vehicle control problem is computationally intensive. This was expected since the problem is $NP-$hard.
Comparing the average optima, shown in figure 5.2(c), it now follows that with increasing number of vehicles the optima of *LMb* are diminishing. That is because of the very restricted search space. Since the search space of *LMa* is greater, the average optimum compared to *LMb* is lower. Due to the fact that the search space for the *direct distance determination* is the largest, both *LM2* with four and 16 lines have the lowest average optima.
If it is necessary to save as many fuel or time as possible, the *direct distance determination* might be the better approach since on the one hand the search space is greater and on the other hand formations are rotational invariant, that means the formations are independet of the position of the coordinate system. Furthermore the distance can be applied directly so no indirect distances have to be calculated in advance.



(a) Average calculation time versus number of vehicles



(b) Average solved linear programs versus number of vehicles



(c) Average optimum versus number of vehicles

Figure 5.2.: Calculation results comparing the introduced approaches for modeling pairwise distances in reference to the line formations LMa, LMb, LM2 with 4 and 16 straight lines

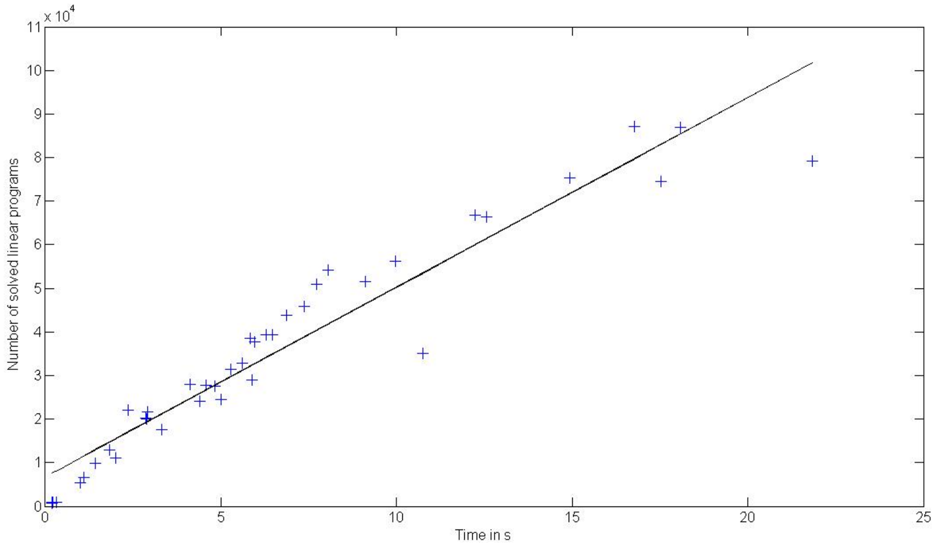
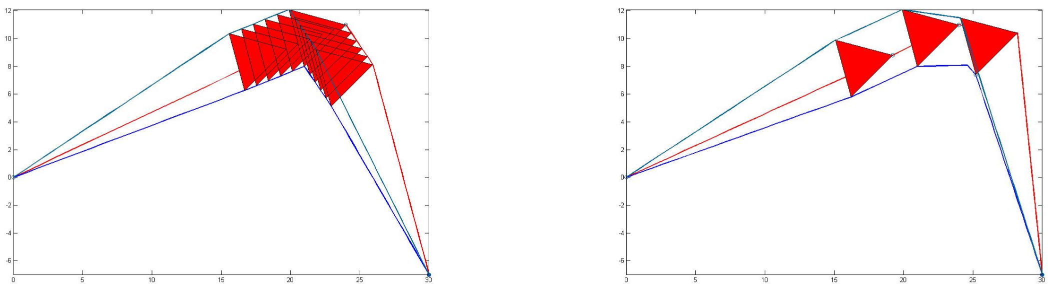


Figure 5.3.: Number of solved linear programs versus time



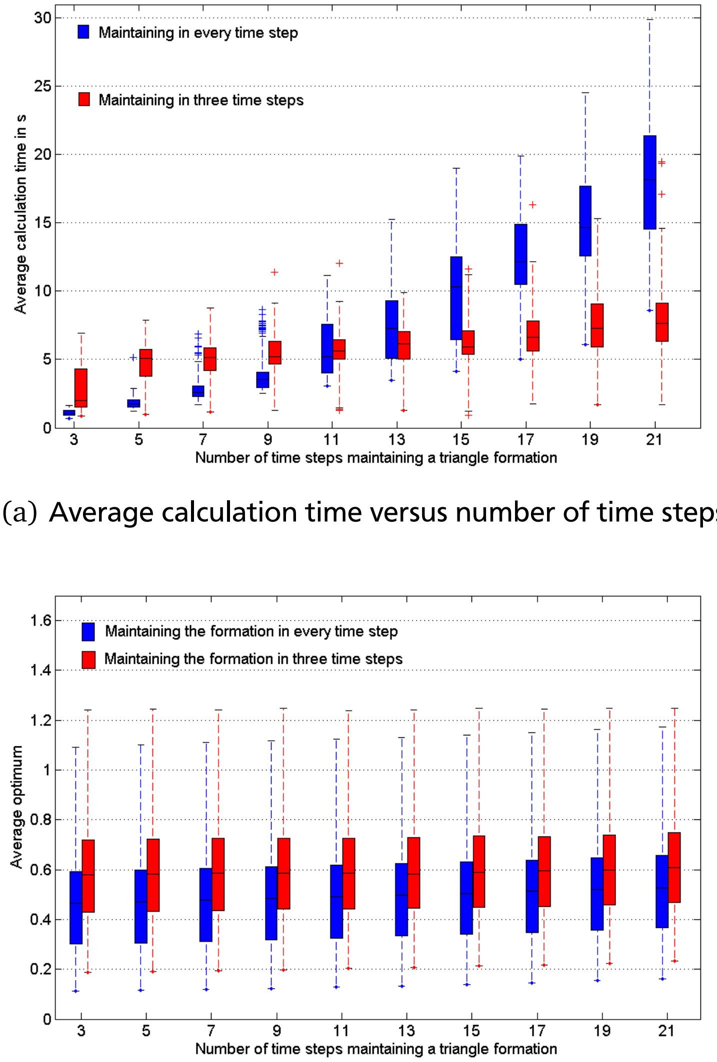
(a) Holding the formation at each timestep i (b) Holding the formation in three appropiate waypoints

Figure 5.4.: Comparing two ways of holding a formation over 11 time steps

5.1.2 Comparing two ways of holding a formation

Test arrangement

Consider now three homogeneous vehicles performing a triangle formation while holding this formation over i timesteps, $i \in \{3, 5, 7, 9, 11, 13, 15, 17, 19, 21\}$. Launching in a starting point they are expected to reach a random but attainable waypoint while holding the formation over i time steps and then they are supposed to reach an attainable random final destination after 50 time steps.
To hold a formation over several time steps two possibilities are presented. The first one deals with holding the formation at every of the i time steps. The second possibility is to introduce two more waypoints, one between the starting point and the initially waypoint and one between the initially waypoint and the final destination. Then, the formation is only supposed to hold in the initial and the two additional waypoints. The position of the additional waypoints is calculated using a convex combination between the waypoints before and afterwards with an appropiate $\lambda \in [0, 1]$.



(a) Average calculation time versus number of time steps

(b) Average optimum versus number of time steps

Figure 5.5.: Calculation results comparing two ways of maintaining a formation over several time steps in reference to the triangle formation TM1

Interpretation

As shown in Figure 5.5(a) the average calculation time maintaining the formation in every time step is lower than maintaining the formation in three time steps, however, only to eleven time steps maintaining a triangle formation. Then, since the calculation of maintaining the formation in every time steps increases stronger, it is advantageous to use only three time steps although two additional waypoints have to be calculated. However, the average optima using additional waypoints is higher than the average optima maintaining the formation in every time step (see Figure 5.5(b))

5.1.3 Example of formation changing vehicles

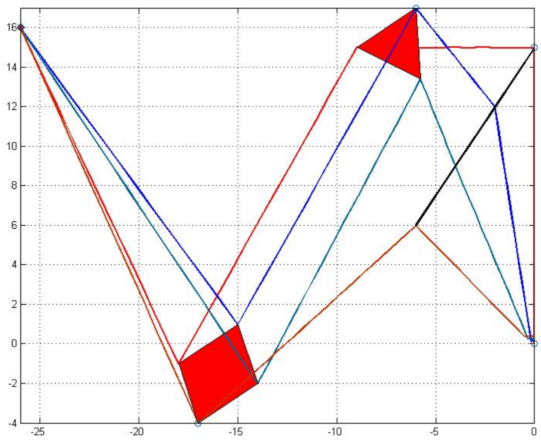


Figure 5.6.: Example showing a formation change: 1. line formation, 2. triangle formation, 3. parallelogram formation

Figure 5.6 shows an example of formation changing homogeneous vehicles. After thirteen time steps a line formation is obtained. Then, after 25 time steps the vehicles maintain a triangle formation. Here, the first three vehicles are supposed to maintain the triangle formation while the fourth vehicle prepares to maintain parallelogram formation after 38 with the first three vehicles. After 50 time steps all vehicles are required to reach a final destination. This combination of formations is performed 100 times with random waypoints. The average calculation time was 39.72145 seconds.

6 Conclusion and outlook

A new heterogeneous multi-vehicle path planning framework is extended with formation topologies to model formation changes. Formation topologies are required either by the cooperative task itself or indirectly by wireless communication for example. The presented framework is based on MILP. MILP is in the NP-complete class and therefore involves a high effort in calculation time. Further, its modeling flexibility is restricted. Nethertheless, MILP fulfills important requirements to plan formation changes. Optima are guaranteed to be globally and there are highly optimized software available to solve them. They offer a high modeling capability. Therefore, MILPs are well suited for benchmarks against heuristic methods, initial guesses for nonlinear methods, or to be embedded in model-predictive control approaches.
Two approaches to model pairwise distances are presented. The *indirect distance determination* is characterized by a strong restriction of the search space. Hence, calculation time shrinks. In contrast, there is no limitation of the search space using the *direct distance determination* . Consequently, calculation time increases substantially but the resulting paths require less fuel. Three formation examples are implemented to systematically study these approaches due to calculation time and achieved optimum. Formations are able to be kept over several time steps and formation changes are possible.
The framework may be extended to multiple waypoint path planning while the order of the waypoints is chosen within the optimization. It is extensible with regard to further formations. Because of the strong combinatorial character and the special discrete structure of this planning problem, it is of interest to integrate several MILP-solver for the purpose of a solver comparison.
Although this framework is not suitable for real-time applications it can be embedded in model-predictive control approaches. These are online-capable and part of current research. The framework is a centralized approach. Approaches to decentralize the planning problem are also part of current research since decentralized approaches deliver a high failure security and they are easier to expand.

A Appendix

A.1 Calculation results

Comparing the introduced approaches for modeling pairwise distances in reference to the line formation

# vehicles	2	3	4	5	6	7
LMa0	0.198823	1.4323883	2.903327	6.455976	12.575403	17.527354
LMb0	0.2012829	1.0039725	1.99739	3.319667	4.396258	5.887198
LM20_4	0.2192002	2.3687099	9.112464	16.770627	31.49236	47.963
LM20_16	0.3064192	5.011116	10.738461	21.84448	66.13506	223.27193

Table A.1.: Average calculation time in s versus number of vehicles

# vehicles	2	3	4	5	6	7
LMa0	1.0628898	1.4520072	1.7855189	2.0828318	2.36057	2.3339814
LMb0	1.0916299	1.5305183	1.9305469	2.3056255	2.665258	2.780128
LM20_4	1.0207498	1.3325757	1.5627693	1.7344384	1.8645715	1.729502
LM20_16	1.0744301	1.458978	1.7720223	2.0281307	2.2463877	2.1225108

Table A.2.: Average optimum versus number of vehicles

# vehicles	2	3	4	5	6	7
LMa0	803.33	9791.34	21648.15	39268.64	66308.46	74605.53
LMb0	823.2	5364.73	11110.61	17588.34	24072.01	28904.3
LM20_4	849.37	21982.56	51591.32	87136.02	147174.98	19099.2
LM20_16	948.87	24579.2	35094.08	79179.86	263971.68	888771.54

Table A.3.: Average number of solved lp's time versus number of vehicles

Comparing two ways of holding a formation over several time steps in reference to the triangle formation

# time steps maintaining the formation	average optimum maintaining in every time step	average optimum maintaining in three time steps	average time in s maintaining in every time step	average time in s maintaining in three time steps
3	0.4839099	0.6070895	1.0952798	2.8877567
5	0.4908842	0.6106762	1.829513	4.5671553
7	0.4975321	0.6110926	2.872281	4.818281
9	0.5040453	0.6124361	4.112244	5.268619
11	0.5107838	0.6139099	5.842642	5.604841
13	0.5176987	0.6164065	7.377963	5.963703
15	0.525014	0.6205456	9.961658	6.2965081
17	0.5331149	0.6251807	12.241133	6.858006
19	0.5420076	0.6307242	14.924364	7.727124
21	0.5517484	0.6374059	18.088163	8.063335

Table A.4.: Average calculation time in s and average optimum versus number of time steps maintaining a triangle formation

List of Figures

1.1. Applications of autonomous mobile vehicles 3

2.1. Referencing techniques to determine a formation position 7

3.1. Indirect distance compliance 11
3.2. Direct distance compliance 13
3.3. Norm approximation 13

4.1. Difference between approximation (left) and exact (right) minimum distance between two vehicles 15

5.1. Comparing the introduced approaches for modeling pairwise distances in reference to the line formation 25
5.2. Calculation results comparing the introduced approaches for modeling pairwise distances in reference to the line formations LMa, LMb, LM2 with 4 and 16 straight lines 27
5.3. Number of solved linear programs versus time 28
5.4. Comparing two ways of holding a formation over 11 time steps 28
5.5. Calculation results comparing two ways of maintaining a formation over several time steps in reference to the triangle formation TM1 29
5.6. Example showing a formation change: 1. line formation, 2. triangle formation, 3. parallelogram formation 30

List of Tables

2.1. Combinatorics 5

A.1. Average calculation time in s versus number of vehicles 32
A.2. Average optimum versus number of vehicles 32
A.3. Average number of solved lp's time versus number of vehicles 32
A.4. Average calculation time in s and average optimum versus number of time steps maintaining a triangle formation 33

Nomenclature

ϵ	ϵ-neighborhood of a waypoint
dist	Distance between vehicles
I	Number of columns of p
i	Counter for the vehicles participating the formation; current column of p
K	Number of vehicles participating the formation
LM11	Line formation using the indirect distance determination with permutation
LM20	Line formation using the direct distance determination without permutation
LM21	Line formation using the direct distance determination with permutation
LMa0	Line formation using the indirect distance determination (approach a)) without permutation
LMb0	Line formation using the indirect distance determination (approach b)) without permutation
LP	Linear Program
M	Large number for logical constraints
MILP	Mixed Integer Linear Program
N	Number of vehicles
O	Number of rows of p
o	Counter for the arrangements of the vehicles; current row of p
p	Matrix which contains line by line the possible and reasonable arrangements of the vehicles
PM10	Parallelogram formation using the indirect distance determination without permutation
PM11	Parallelogram formation using the indirect distance determination with permutation
PM20	Parallelogram formation using the direct distance determination without permutation
PM21	Parallelogram formation using the direct distance determination with permutation
safeDist	Safety distance between vehicles
T	Number of time steps
t	Counter for time steps
TM10	Triangle formation using the indirect distance determination without permutation
TM11	Triangle formation using the indirect distance determination with permutation
TM20	Triangle formation using the direct distance determination without permutation
TM21	Triangle formation using the direct distance determination with permutation
u	Control input
x,y	Vehicle state in x- and y-direction

Bibliography

[1] Marco, M. D. ; Garulli, A. ; Giannitrapani, A. ; Vicino, A.: Simultaneous Localization and Map Building for a Team of Cooperating Robots: A Set Membership Approach. In: *IEEE: Transactions on Robotics and Automation* 19(2) (2003), S. 238–249

[2] Sugar, T. G. ; Kumar, V.: Control of Cooperating Mobile Manipulators. In: *IEEE: Transactions on Robotics and Automation* 18(1) (2002), S. 94–103

[3] Murray, R. M.: Recent Research in Cooperative Control of Multi-Vehicle Systems. In: *ASME Journal of Dynamic Systems, Measurement and Control* (2006), September

[4] Richards, A. ; Schouwenaars, T. ; How, J. P. ; Feron, E.: Spacecraft Trajectory Planning with Avoidance Constraints Using Mixed-Integer Linear Programming. In: *Journal of Guidance, Control and Dynamics* 25 (2002), S. 755–764

[5] Reinl, C. ; Stryk, O. v.: Optimal control of multi-vehicle-systems under communication constraints using mixed-integer linear programming. In: *RoboComm '07: Proceedings of the 1st international conference on Robot communication and coordination*. Piscataway, NJ, USA : IEEE Press, 2007. – ISBN 978–963–9799–08–0, S. 1–8

[6] ILOG: *ILOG CPLEX 11.0, User's Manual*. 2007

[7] Glocker, M. ; Reinl, C. ; Stryk, O. v.: Optimal task allocation and dynamic trajectory planning for multi-vehicle systems using nonlinear hybrid optimal control. In: *Proc. 1st IFAC-Symposium on Multivehicle Systems*. Salvador, Brazil : Proc. 1st IFAC-Symposium on Multivehicle Systems, October 2-3 2006, S. 38–43

[8] Martin, A.: *Diskrete Optimierung*. 2006

[9] Garey, M. R. ; Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979

[10] Dür, M. ; Martin, A. ; Ulbrich, S.: *Optimierung I*. 2006

[11] Ulbrich, S.: *Innere-Punkte-Verfahren der konvexen Optimierung*. 2010

[12] Stryk, O. v.: *Robotik II*. 2009

[13] Lozano-Perez, T.: *Spatial planning with polyhedral models*, MIT, Diss., 1980

[14] Barraquand, J. ; Kavraki, L. E. ; Latombe, J. C. ; Li, T. Y. ; Motwani, R. ; Raghavan, P.: A random sampling scheme for path planning. In: *International Journal of Robotics Research* 16(6) (1997), S. 759–774

[15] McQuade, F. ; Ward, F. ; McInnes, C.: The autonomous configuration of satellite formations using generic potential functions. In: *Proceedings of the International Symposium Formation Flying* (2002)

[16] Schouwenaars, T. ; Moor, B. de ; Feron, E. ; How, J. P.: Mixed Integer Programming for Multi-Vehicle Path Planning. In: *Proceedings of the European Control Conference* (2001)

[17] Richards, A. ; How, J. P.: Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming. In: *Proceedings of the American Control Conference* (2002), S. 1936–1941

[18] Richards, A. ; Bellingham, J. ; Tillerson, M. ; How, J. P.: Co-ordination and Control of Multiple UAVs. In: *AIAA Guidance, Navigation and Control Conference and Exhibit* (2002)

[19] Earl, M. G. ; D'Andrea, R.: Iterative MILP Methods for Vehicle Control Problems. In: *CoRR* abs/cs/0505042 (2005)

[20] Balch, T. ; Arkin, R. C.: Behavior-Based Formation Control for Multi-Robot Teams. In: *IEEE Transactions on Robotics and Automation* 14 (1999), S. 926–939

[21] Wang, P. K. C.: Navigation stategies for multiple autonomous robots moving in formation. In: *Journal of Robotics Systems* 8 (1991), S. 177–195

[22] Chen, Q. ; Luh, J. Y. S.: Coordination and Control of a Group of Small Mobile Robots. In: *IEEE International Conference on Robotics and Automation* (1994), S. 2315–2320

[23] Parker, L. E.: *Heterogeneous Multi-Robot Cooperation*, Mass. Inst. of Technology, Cambridge, Diss., 1994

[24] Desai, J. ; Ostrowski, J. ; Kumar, V.: Control of Formations for Multiple Robots. In: *IEEE International Conference on Robotics and Automation* (1998)

[25] Egerstedt, M. ; Hu, X.: Formation Constrained Multi-Agent Control. In: *ICRA*, 2001, S. 3961–3966

[26] Hao, Y. ; Laxton, B. ; Agrawal, S. K. ; Lee, E. ; Benson, E.: Planning and Control of UGV Formations in a Dynamic Environment: A Practical Framework with Experiments. In: *IEEE: International Conference on Robotics and Automation* (2003)

[27] Dunbar, W. ; Murray, R.: Model predictive control of coordinated multi-vehicle formations. In: *Proceedings of the Conference on Decision and Control* (2002)

[28] Latombe, J. C.: *Robot Motion Planning*. Kluwer Academic Publishers, 1991