2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

# Improved Predictability of Reactive Robot Control using Control Lyapunov Functions

Petter Ögren*
Department of Autonomous Systems
Swedish Defence Research Agency
SE-164 90 Stockholm, Sweden
petter.ogren@foi.se

*Abstract*— **Model based robot control approaches are often designed to allow the verification of certain system properties such as safety or goal convergence. However, designing such controllers is often very time consuming, and most of the time it is not possible to add additional control objectives without jeopardizing the previously proved system properties.**

**So-called reactive control approaches, such as the subsumption architecture, potential field methods and voting schemes offer an alternative to the model based approaches. These methods are inherently modular and thus straightforward to apply to problems when there are a number of control objectives that needs to be met at the same time, e.g., navigation, collision avoidance and formation maintenance. The basic idea is to design one controller for each objective, and then merge the different controller outputs. The problem is that the result of this merging of different outputs is hard to analyze and predict in detail.**

**This paper describes an attempt to narrow this gap between the modularity of reactive approaches and the predictability of model based methods using a concept similar to Control Lyapunov Functions (CLFs). By assigning a CLF-like scalar function to each control objective, signifying to what extent that objective is met, the predictability is improved in two ways. Firstly, the user can prioritize between not only different objectives, but also between different levels of satisfaction in different objectives. Secondly, the time derivatives of these functions can be used to find controls that maintain an important objective that is currently met, while striving to satisfy another, less important one that is not met.**

## I. Introduction

The field of Robotic control can loosely be divided into model based and reactive control approaches. In model based control, a mathematical model of the dynamic or kinematic behavior of the robot is used to synthesize a controller and analyze and prove attractive system properties such as safety or goal convergence.

Model-based approaches to robot control with multiple objectives include [7], [12], [8]. These approaches perform well, but are tailor made to the specific problems they address, and adding additional objectives to be met is often difficult or impossible without spoiling the structures being used in the proofs.

Non model based methods include [2], [13], [1], [4], [11], [6], [9], [10]. A nice overview can be found in [2], and a common idea is to design one controller for each control objective and then merge the outputs in some clever fashion. This merging can be either a selection of the most important controller at each time instant, or a fusion of a set of control signals into a single output. Examples of selection schemes are the subsumption architecture [4], and the TCA system, [11], while fusion methods include the potential field approach [6], voting schemes [9], fuzzy rules [10], and the dynamical systems approach, [1].

In this paper we propose a framework called Behavior Control Lyapunov Functions (BCLFs). These are scalar functions describing to what extent each control objective is met. They are furthermore chosen in such a way that there is always a choice of control signal such that they decrease. It is our hope to use these functions to combine the modularity of the reactive approaches with the predictability of the model based approaches.

This framework will enable two things. Firstly, the operator can prioritize not only between different objectives, but also between different levels of satisfaction, i.e. keeping a safety distance of $1m$ might be more important than reaching the goal on time, which in turn is more important than keeping the ideal safety distance of $2m$. Secondly, the robot controller can use the time derivatives of the BCLFs to choose controls preserving a more important objective that is already met, while aiming at achieving a less important objective.

The organization of this paper is as follows. In Section II we review the basic idea of Lyapunov theory and discuss the structure of reactive architectures. Section III then presents the BCLF framework and some analytical properties. Finally, the approach is illustrated by simulation examples in Section IV and conclusions are drawn in Section V.

## II. Background: Lyapunov Theory and Reactive Architectures

In this Section we will describe the two main ideas built upon in this paper.

### A. Control Lyapunov Functions

The ideas below were first presented in 1892 by the Russian mathematician Aleksandr Lyapunov, and they build upon a very natural observation: if the energy of a system decreases monotonically everywhere, except at a local energy

*The author was funded by the Swedish defence materiel administration (FMV) through the *Technologies for Autonomous and Intelligent Systems* (TAIS) program, 297316-LB704859.

minimum, then the system will end up in that minimum. The definition below is taken from a paper by Artstein [3].

*Definition 1 (Control Lyapunov Function):* A positive definite $C^1$ function $V(x) : \mathbb{R}^n \to \mathbb{R}^+$ is called a control Lyapunov function if,

$$\inf_u \left(\nabla V^T f(x,u)\right) < 0, \ \forall x \neq 0.$$

This definition is motivated by the fact that if $V$ is a control Lyapunov function, then the right choice of control $u(x)$ will make it decrease and conclusions about stability and convergence can be made.

*Remark 1:* In Section III we will make significantly stronger assumptions than the ones above, in order to prove finite time properties in a straightforward manner. So even though we do not use the above definition explicitly, the ideas of this paper rest heavily upon the Lyapunov framework.

We now move on to the area of reactive control architectures.

### B. Reactive Architectures

In a book by Arkin [2] a number of reactive control approaches are described in detail. Arkin summarizes the idea as follows. "Simply put, reactive control is a technique for tightly coupling perception and action, typically in the context of motor behaviors, to produce timely robotic response in dynamic and unstructured worlds."[2, p. 22]

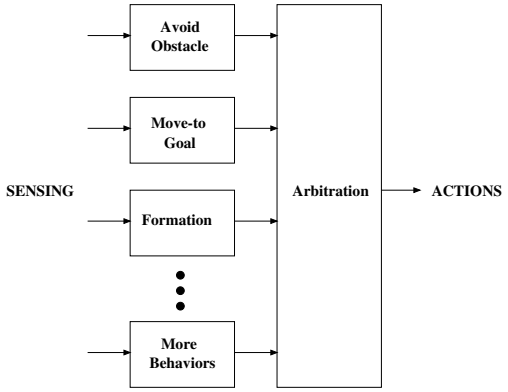


Fig. 1. A typical reactive architecture, from [2, p. 112-114]. Note how all individual controllers (behaviors) work in parallel.

A typical reactive architecture is seen in Figure 1. The different controllers are called behaviors and work in parallel, each describing its own complete sensor-to-actuator mapping. The desired actuator commands of the behaviors are then merged by an arbitrator. There are mainly four suggested ways of arbitration:

1) *Suppression*: A strict hierarchy of behaviors is defined, where the highest active one makes the call. For example, the `Avoid-obstacle` can override the `Move-to-goal` behavior if there is an obstacle close by. This is the arbitration mechanism suggested by Brooks in the subsumption architecture [5] .
2) *Selection*: A variable hierarchy where both the agent's goals and sensory information governs what behavior gets control.
3) *Voting*: All behaviors are allowed to vote on their preferred action. The choice with the most votes is then executed.
4) *Vector summation*: All behavior outputs are in the form of a vector, e.g. either desired velocity or acceleration. These vectors are then summed, and in some implementations normalized, to get the desired motor command.

Having reviewed the ideas of reactive robot control as well as some basic Lyapunov theory we are ready to describe what we mean by Behavior Control Lyapunov Functions and how they can be used to do arbitration in a more transparent and predictable way.

## III. The Behavior CLF Framework

In this section we will describe how to translate a set of mission objectives and priorities into Behavior CLFs and a priority table. We then characterize what control sets meet the different objectives. Finally, based on these sets a controller is proposed.

### A. Mission Objectives and Priority Tables

The first step in the controller design is to state a number of mission objectives and a corresponding set of scalar functions describing to what degree those objectives are met. The user then supplies a priority table with information about what objectives are more important than others.

In a ground robot setting the mission objectives might be as follows

- Arrive at the goal location within the given time.
- Avoid static obstacles with a given safety margin.
- Avoid collisions with other robot with a given safety margin.
- Move in some desired formation with other robots.

Ideally, the mission can be performed while accomplishing all these objectives. There are however situations where the objectives are contradicting, and a tradeoff decision has to be made. For example, when the robot is moving in formation with some other robots who have different goal locations. Then there is a tradeoff of staying in formation, or leaving the formation to reach the goal location. Below we propose to use scalar functions and a priority table to make sure that such tradeoffs are made in a transparent and predictable way.

To make the four robot objectives above explicit in terms of functions and inequality constraints we first write

$$\begin{aligned} V_{1:TOA} &= t + \frac{||p - p_{goal}||}{v_{nom}} \\ V_{2:OA} &= -\min_{o \in \text{(obst)}} ||p - o|| \\ V_{3:CA} &= -\min_{q \in \text{(robots)}} ||p - q|| \\ V_{4:F} &= \Sigma_{q \in \text{(robots)}} \left|(||p - q|| - 30)\right| \end{aligned} \quad (1)$$

where $V_{1:TOA}$ is the estimated time of arrival (TOA), $V_{2:OA}$ is the negative distance to the closest obstacle, used for obstacle avoidance (OA), $V_{3:CA}$ is the negative distance to the closest robot, used for collision avoidance (CA), $V_{4:F}$ is the formation (F) fitness function, $p$ is the robot position, and the time $t$ is assumed to be part of the state $x$. The distances are measured in negative numbers in order for all objectives to be stated in the form $V_i \leq b_{ij}$.

After having defined the scalar functions we need a set of inequalities to decide when a mission objective is satisfied. Instead of just writing one inequality for each objective, the user supplies a priority table such as the one below. Column 0 is always satisfied for technical reasons explained below. Column 1 contains the first priority of the robots, in this case to keep $V_{2:OA} \leq -1$, i.e., having a safety margin of at least $1m$ to all obstacles. Column 2 then contains the second priority of the robots, having the same safety margin to all other robots, i.e. $V_{3:CA} \leq -1$. Column 3 adds the objective of reaching the goal before $t = 400$, i.e. $V_{1:TOA} \leq 400$ and so on. Note that Column 5 contains the desired safety margin of $V_{3:CA} \leq -15$, thus if the robot is able to arrive within $t = 250$, the safety margin of $15m$ is used. But if both of these can not be achieved, $t = 250$ is more important, and instead a safety margin of $1m$ is used, see Column 4.

The main idea of this approach is then to choose controls that decrease the functions, and thus satisfy the inequalities and the corresponding objectives in the order described in the table below. For this to work, the functions must have some particular properties defined below. The following three sections are slightly more technical, but the main idea is as simple as described above.

### B. Behavior Control Lyapunov Functions

We will now formally define what we mean by Behavior Control Lyapunov Functions (BCLF)

*Definition 2 (BCLF):* Given a system $\dot{x} = f(x, u)$, a bounded set of admissible controls $U_{adm}$, a piecewise $C^1$ function $V : \mathbb{R}^n \to \mathbb{R}$, and scalars $b$ and $\epsilon > 0$. Then $V$ is a BCLF for the bound $b$ and $\epsilon$ if

$$\min_{u \in U_{adm}} \nabla V^T f(x, u) \leq -\epsilon, \ \forall x : V(x) \geq b.$$

With this definition we can state the following Lemma.

*Lemma 1:* If $V$ is a BCLF, for $b, \epsilon$, then there exists a control sequence that achieves the objective $V(x) \leq b$ in finite time.

*Proof.* Assume we start at some $x = x_0$. Apply the control $u^* = \text{argmin}_{u \in U_{adm}} \nabla V^T f(x, u)$. Then $\dot{V} \leq -\epsilon$ and hence the bound $b$ will be reached in time $t \leq \frac{V(x_0)-b}{\epsilon}$. ∎

Note that we will use a kinematic robot model, $\dot{x} = u, \quad ||u|| \leq v_{max}$, throughout this paper. This is the case that is most similar to the reactive robot approaches since most of them use the desired robot velocity as output. However, the framework is open to using dynamic robot models, with more complex BCLFs. There are no general restrictions as to what kind of system $\dot{x} = f(x, u)$ can be.

| $b_{ij}$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $V_{1:TOA} \leq$ | $\infty$ | $\infty$ | $\infty$ | 400 | 250 | 250 | 250 | 0 |
| $V_{2:OA} \leq$ | $\infty$ | $-1$ | $-1$ | $-1$ | $-1$ | $-1$ | $-1$ | $-1$ |
| $V_{3:CA} \leq$ | $\infty$ | $\infty$ | $-1$ | $-1$ | $-1$ | $-15$ | $-15$ | $-15$ |
| $V_{4:F} \leq$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | 8 | 8 |

Given a priority table with $\{b_{ij}\}$ values, such as the one above, one can formally define the current priority level as follows.

*Definition 3 (CPL):* Given a set of BCLFs, $V_i$, and bounds, $b_{ij}$, such that $b_{i0} = \infty, \forall i$. Let the Current Priority Level (CPL) or $j_{CPL}(x)$, be given by

$$j_{CPL}(x) = \max\{j : V_i(x) \leq b_{ij}, \ \forall i\}$$

i.e. $j_{CPL}(x)$ is the rightmost column of $\{b_{ij}\}$, where all constraints are satisfied.

Note that demanding $b_{i0} = \infty$ for all $i$ guarantees that the CPL is always defined. The control objective of the robot can now be stated in terms of maximizing the CPL, i.e. move the system as far down the priority table as possible.

### C. Control Sets that Satisfy Different Mission Objectives

We will now make explicit the relationship between controls and CPLs.

To increase the CPL the controller must strive to satisfy the constraints in the next column of the table in section III, while not violating the constraints of the current column. Below we will see how this can be made more precise.

*Definition 4 (Control Sets):* Fix $k > 0$ and let $\epsilon > 0$ be the same as in Definition 2. Given a set of BCLFs, $V_i$, with corresponding bounds, $b_{ij}$. Let

$$U_{sat}(x) = \{u : \dot{V}_i(x, u) \leq \frac{1}{k}(b_{ij} - V_i(x)), \ j = j_{CPL}(x), \forall i\},$$

the set of controls *satisfying* the bounds of the CPL. Let furthermore

$$I_{next}(x) = \{i : V_i(x) \geq b_{i(j+1)}, \ j = j_{CPL}(x)\},$$

the set of objectives to be focused on and

$$U_{inc}(x) = \{u : \dot{V}_i(x, u) \leq -\epsilon, \forall i \in I_{next}(x)\},$$

the set of controls aiming to *increase* the CPL.

*Remark 2:* The constant $k$ governs how fast a satisfied bound $b_{ij}$ can be approached. If the worst possible $u \in U_{sat}$ is chosen, then we have equality in the constraints, $\dot{V}_i(x, u) = \frac{1}{k}(b_{ij} - V_i(x))$, and $V_i$ approaches $b_{ij}$ exponentially, with the time constant $k$.

We will now characterize the sets that guarantee the satisfaction of the CPL and the future increase in PL.

*Lemma 2:* If a system starts at $x(t_0) = x_0$, and the chosen controls $u$ satisfy

$$u \in U_{sat}(x),$$

then $j_{CPL}(x_0) \leq j_{CPL}(x(t)), \ \forall t > t_0$, i.e. the CPL will not decrease. If furthermore

$$u \in U_{sat}(x) \cap U_{inc}(x),$$

then $j_{CPL}(x_0) < j_{CPL}(x(t))$ will be satisfied in finite time, i.e. the CPL will increase.

*Proof.* Assume that there was a decrease in CPL at some state $\hat{x}$. Then $b_{ij} = V_i(\hat{x})$ and $\dot{V}_i(\hat{x}, u) > 0$ for some $i = \hat{i}$.

But since $u \in U_{sat}(x)$ we have $\dot{V}_{\hat{i}}(x, u) \leq k(b_{\hat{i}j} - V_{\hat{i}}) = 0$, which contradicts the assumption. This proves the first part of the Lemma. For the second part we note that $j_{CPL}(x_0) < j_{CPL}(x(t))$ in time $t < \max_{i \in I_{next}(x)} \frac{V_i - b_{i(j+1)}}{\epsilon}$, similarly to Lemma 1. ∎

*Remark 3:* Note that in cases where a tradeoff must be made, e.g., two robots are traveling in formation but have goal locations in different places. Then there are no way for both robots to stay in formation and arrive at their goal positions in time. Thus the sets $U_{sat}(x)$ and $U_{sat}(x) \cap U_{inc}(x)$ might be empty, depending on the current priority level. The sets are however a clear representation of the controls that guarantee kept or increased CPLs.

The question of what control to choose in the recommended set is the topic of the next section.

### D. Controller Synthesis

As seen above, choosing a control in $U_{sat}$ and $U_{sat} \cap U_{inc}$ is attractive, but as noted in Remark 3, one or both of these sets might be empty. Below we define the set $U_{sat*}$ that is never empty and identical to $U_{sat}$ whenever $U_{sat} \neq \emptyset$.

In cases when $U_{sat}$ is empty the CPL can probably not be sustained and is about to drop one or more levels. In these situations we propose to search the priority table to find the highest PL that *can* be satisfied in the near future. The controls are then chosen to satisfy this level. To formalize this we make the following definition.

*Definition 5:* Let $k$ be the same as in Definition 4. Let furthermore

$$\begin{aligned} j_{CPL*}(x) \quad = \quad & \max\{j : V_i(x) \leq b_{ij}, \\ & \exists u \in U_{adm} : \\ & \dot{V}_i(x, u) \leq \frac{1}{k}(b_{ij} - V_i(x)),\ \forall i\} \end{aligned} \tag{2}$$

and

$$\begin{aligned} U_{sat*}(x) \quad = \quad & \{u \in U_{adm} : \\ & \dot{V}_i(x, u) \leq \frac{1}{k}(b_{ij} - V_i(x)), \\ & j = j_{CPL*}(x), \forall i\}, \end{aligned} \tag{3}$$

i.e. $j_{CPL*}(x)$ is the rightmost column of $\{b_{ij}\}$, where all constraints are satisfied, and there is a nonempty set, $U_{sat*}$, of controls that does not immediately violate them.

*Lemma 3:* The set $U_{sat*}(x)$ is never empty.

*Proof.* The proof follows directly from the definition of $j_{CPL*}(x)$. ∎

With these sets we are ready to address the controller design issue.

It is clear that the objectives in $I_{next}$ should be focused on. We suggest the following:

$$u^* = \text{argmin}_{u \in U_{sat*}} \Sigma_{i \in I_{next}} \dot{V}_i(x, u), \tag{4}$$

or $u^* = \text{argmin}_{u \in U_{sat*}} \Sigma_i \dot{V}_i(x, u)$ if CPL is at maximum.

In the next section we will see how the proposed controller performs in three different settings. However, we will first comment on how this control structure compares to the other arbitration alternatives that were reviewed in Section II. The priority table and $U_{sat}$, $U_{inc}$ clearly borrows a lot from the arbitration methods *Selection* and *Suppression*. Using set intersection of $U_{sat}$ and $U_{inc}$ is similar to *Voting*, in that options that are acceptable relative to many mission objectives are favored. Finally, minimizing a sum of $\dot{V}_i(x, u)$ resembles the arbitration method *Vector summation*. Thus the BCLF framework borrows and merges a set of different ideas from reactive robot control using a mathematic framework inspired by CLFs.

It is now time to illustrate the approach with a set of simulation examples.

## IV. Simulation Examples

Before going into details about the simulations we would like to emphasize two things, the first is that the proposed scheme is a variation of reactive robot control and thus the plots will look *qualitatively* very much like those of any other reactive control scheme. However, the outcomes are more predictable in the *quantitative* sense. For instance, Figure 8 shows how the given priorities in terms of e.g., time of arrival, are taken into account in a way that is not possible with the other reactive approaches. The second thing we would like to emphasize is that the choice of circular obstacles was made for simplicity. The proposed scheme works perfectly well with nonconvex obstacles with the modification of using a Navigation function such as the one in [7] instead of the norm to measure goal distance in Equation (1).

Three robots are simulated using the priorities of the table in section III. In the first two simulations we do not use the formation objective, $V_{4:F}$ of equation (1). Running the simulation we get the results of Figure 2, 3 and 4. Figure 2 shows a top view, Figure 3 shows the evolution of the BCLFs in equation (1) over time, and the left part of Figure 4 shows the sets $U_{sat*}$ and $U_{inc}$ of the rightmost robot at the beginning. Throughout this section we will use the term rightmost robot to refer to the robot starting at (100,0) etc.

As can be seen in Figure 3, $V_{1:TOA} \leq 200$ for all times and thus all the $b_{ij}$ values in the first row of the table in section III are satisfied except the one in column 7, an estimated arrival time of $t = 0$. The second row contains the obstacle avoidance margins. Note that the numbers are negative due to the fact that all inequalities are '$\leq$'. It can be seen from Figure 3 that $V_{2:OA} \leq -1$ for all times and thus the obstacle margin of 1 is always satisfied. Finally, the third row contains inter robot distance margins and again it can be seen in Figure 3 that $V_{3:CA} \leq -15$ thus satisfying the margins of 1 as well as 15 for all times. The fourth row is not used in this scenario, and we conclude that all three robots remain at $j_{CPL*} = 6$ throughout the simulation. At this CPL, $I_{next}(x) = 1$ and the controllers focus on decreasing $V_{1:TOA}$, i.e. moving towards the goal, while $U_{sat*}$ makes

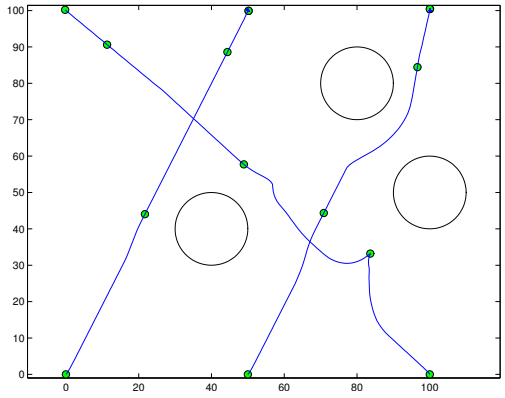

Fig. 2. Three robots starting at (0,0), (50,0) and (100,0) with corresponding goal locations at (50,100), (100,100) and (0,100). The large circles are obstacles while the small filled circles are snapshots of robot locations every 50 s. The robot trajectories are shown as solid curves. The $b_{ij}$ values used are found in the table of section III. Note how the rightmost robot first turns right to keep the 15m safety distance to the middle robot, but then chooses to stop and pass on the left instead of heading towards the obstacle. The $V_i$ functions are plotted in Figure 3.

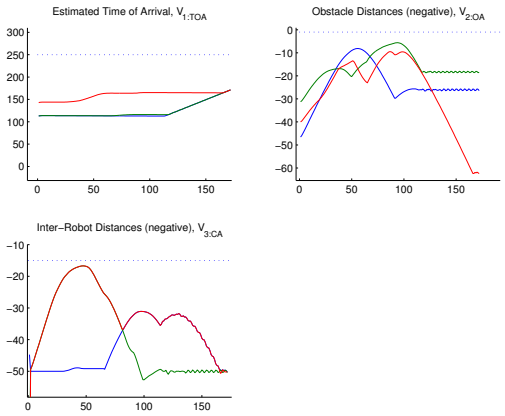


Fig. 3. The $V_i$ functions of the scenario in Figure 2. Note how the inter robot distances peak around $t = 50$ without violating the $V_{3:CA} \leq -15$ constraint. Note also that all estimated times of arrival are well within the bound of 250 and obstacle distances within the $-1$ constraint.

sure that they do so without jeopardizing the objectives already satisfied at $j_{CPL*} = 6$. In the beginning, as shown in the left part of Figure 4, $U_{sat*} = U_{adm}$, i.e. none of the currently satisfied objectives are close to not being met. As can be seen in the right part of Figure 4, this is however not the case for the next simulation.

The results of the second simulation run are shown in Figures 5, 6 and 4. This simulation is identical to the first one, with the one exception that the internal clock of the rightmost robot is started at $t = 100$ instead of $t = 0$, as the other robots. This fact makes the time of arrival objective, $t = 250$, very close to not being met, as can be seen in Figure 6.

The third simulation includes the formation objective, $V_{4:F}$ and is shown in Figures 7, 8 and 9.

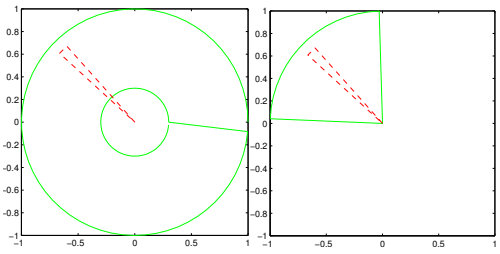

Fig. 4. The control constraints of the rightmost robot at the start of the two scenarios of Figure 2 (left plot) and Figure 5 (right plot). The left plot illustrates the set $U_{sat*}$ as two solid circles, meaning that high velocity as well as low velocity controls in all directions are allowed without violating the current constraints. The dashed triangle illustrates the set $U_{inc}$ and thus the direction towards goal in this case. The right figure illustrates that only high velocity commands in the north-west directions are allowed to satisfy the current constraints which include the time of arrival constraint that is close to being violated. Again, the dashed triangle denotes the set $U_{inc}$.

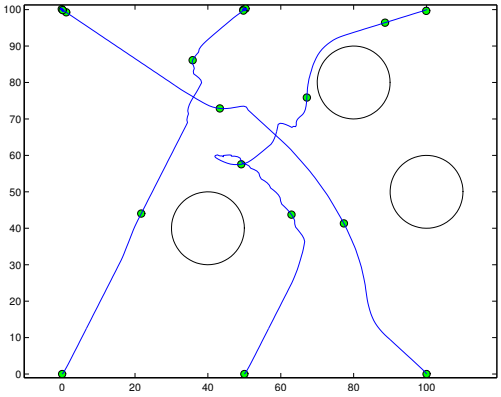

Fig. 5. The scenario of Figure 2 with the clock of the rightmost robot set $100s$ ahead, resulting in a time shortage to meet the time limit of $t = 250$. Note how the middle robot has to make a sharp left turn while the rightmost one only deviates slightly from its course.

As can be seen in Figure 8, all robots start with $V_{4:F} > 8$, i.e. they do not satisfy the formation objective and thus end up at $j_{CPL*} = 5$. This makes $I_{next}(x) = 4$ and the robots move towards eachother while initially ignoring their goal locations. After establishing a formation the robots reach $j_{CPL*} = 6$ as the only objective not being met is the ideal time of arrival at $t = 0$. Thus the robots move towards their individual goal locations making the formation rotate slightly to avoid obstacles and account for the positions of the different destinations. The controller constraints of the rightmost robot at two separate time instants are shown in Figure 9.

## V. Conclusions

Reactive approaches are modular, and easy to understand from a user point of view. However, they often lack the mathematical rigor and predictability of model based control schemes. In this paper we have tried to narrow this gap by proposing an approach that combines the modularity of reactive robot control methods with tools from Lyapunov theory.

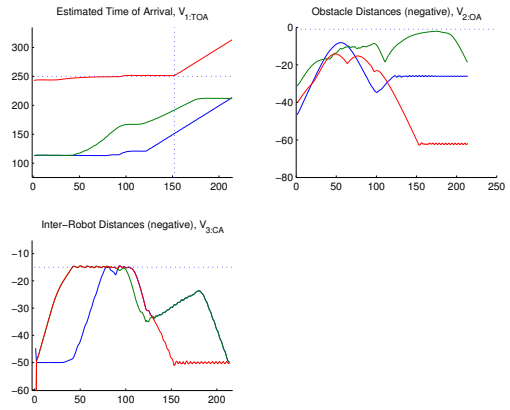


Fig. 6. The $V_i$ functions of the scenario in Figure 5. Note how the rightmost robot starts out close to the constraint of the $t = 250$ arrival time. Note also how all three robots are very close to the $15m$ safety distance constraint from time $t = 75$ to $t = 100$. The unmodelled robot dynamics actually makes the robot with time shortage fall just above the constraint resulting in a new safety limit of 1m and thus a straight line trajectory, in Figure 5, towards the goal from time $t = 100$.

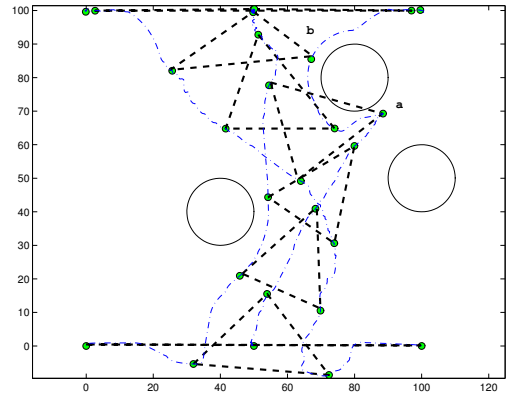


Fig. 7. The same scenario as in Figures 2 and 5, with the formation maintenance objective added, $V_{4:F}$. The snapshots are plotted every $35s$ and dashed triangles show concurrent positions of the robots. Note how the robots as before starts on the line y=0, but quickly move towards the middle, to form a triangle formation. Once this is established, they move north towards their respective goals. The obstacles, as well as the positions of the different goal locations make the formation rotate clockwise and finally dissolve in order for the robots to meet their arrival time constraints.

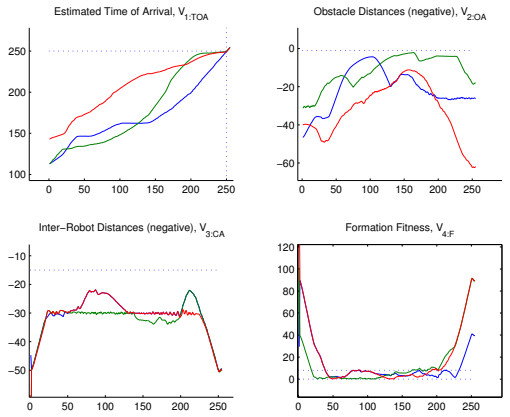


Fig. 8. The $V_i$ functions of the scenario in Figure 7. Note how all robots initially focus on meeting the formation constraint, which is satisfied around $t = 50$. Then the formation moves north while respecting obstacle distances. Finally, the formation constraint is violated in order to just meet the time of arrival constraint at $t = 250$.

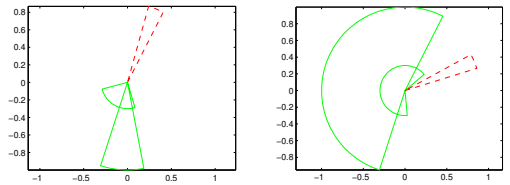

Fig. 9. The controller constraints of the middle robot of the scenario in Figure 7 at positions **a** (left plot) and **b** (right plot). At **a** the formation and obstacle avoidance constraints jointly excludes all control options but high velocity south, or low velocity south or south west. These are almost opposite to the desired goal direction of north east. At **b** on the other hand, only the obstacle avoidance constraint is active, enabling all high- and low velocity commands away from the obstacle, as well as a few low velocity commands moving slightly closer. Again, $U_{inc}$, in these cases pointing towards the goal, is shown as a dashed triangle.

## REFERENCES


[1] P. Althaus and H.I. Christensen. Behavior coordination in structured environments. *Brill Academic Publishers: Advanced Robotics*, 17(7):657–674, 2003.

[2] R.C. Arkin. *Behavior-Based Robotics*. MIT Press, Cambridge, MA, 1998.

[3] Z. Artstein. Stabilization with relaxed controls. *Nonlinear Anal.*, 7(11):1163–1173, 1983.

[4] R. Brooks. A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of [legacy, pre-1988]*, 2(1):14–23, 1986.

[5] Rodney Brooks. Elephants don't play chess. *Robotics and Autonomous Systems (6)*, pages 3–15, 1990.

[6] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, 5(1):90, 1986.

[7] P. Ogren and NE Leonard. Obstacle avoidance in formation. *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'03.*, 2, 2003.

[8] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[9] P. Pirjanian, H.I. Christensen, and J.A. Fayman. Application of voting to fusion of purposive modules: an experimental investigation. *Robotics and Autonomous Systems*, 23(4):253–266, 1998.

[10] A. Saffiotti, K. Konolige, and E.H. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.

[11] RG Simmons. Structured control for autonomous robots. *IEEE Transactions on Robotics and Automation*, 10(1):34–43, 1994.

[12] HG Tanner. Flocking with obstacle avoidance in switching networks of interconnected vehicles. *IEEE International Conference on Robotics and Automation, ICRA'04*, 3, 2004.

[13] B.B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110(2):293–320, 1999.