Autonomous UCAV Strike Missions using Behavior Control Lyapunov Functions

Petter Ögren*, Adam Backlund*, Tobias Harryson†, Lars Kristensson*, Patrik Stensson†

*Swedish Defence Research Agency (FOI), SE-164 90 Stockholm, Sweden

†Swedish Air Force, Air Combat School, Box 645, SE-751 27 Uppsala, Sweden

Abstract

An autonomous Unmanned Combat Aerial Vehicle (UCAV) carrying out a surveillance or strike mission must be able to handle situations where the different mission objectives are in conflict and a tradeoff must be made, e.g. when the time of arrival is in conflict with the prescribed safety distance to an enemy surface to air missile (SAM) site. This paper describes a framework called Behavior Control Lyapunov Functions (BCLF), to handle such tradeoffs. The framework combines the natural idea of different control behaviors for different mission objectives, suggested in the Behavior Based robotics approach, with the mathematical transparency of Control Lyapunov Functions (CLF) from control theory.

First, each behavior is represented by a scalar function with certain CLF-like properties, describing to what extent that mission objective is satisfied. The operator then edits a priority table reflecting the order of importance between different objectives, as well as different levels of satisfaction. Based on the table and the current levels of satisfaction the algorithm decides which objectives should be focused on right now, and which should currently be ignored. Finally, the current high priority objectives are transformed into recommended subsets of the available control choices, and passed to the controller. The paper is concluded with simulation examples illustrating the approach.

I. Introduction

The issues of autonomous UAV control have been given an increasing amount of attention in recent years.[1,5,7–9,11,13–15] The path planning part of an autonomous UAV is always important, however in dynamic environments, when the UAV must react to pop-up threats, as well as the motion of friendly and enemy aircraft, planning alone will not solve the problem.

A number of new approaches,[10,11] building upon ideas from model predictive control (MPC) have emerged trying to incorporate long term planning into short term control schemes. These short term schemes often use trajectory optimization formulations, an approach that is powerful, but of a very low degree of transparency, from an operator point of view.

In mixed initiative control applications, where an autonomous system is to cooperate with, and in some situations be controlled by, a human operator, situations when the operator is surprised by the autonomous system must be avoided. Therefore, from a mixed initiative point of view, frameworks such as the behavior based approaches[2,4] are very attractive, but unfortunately lack the mathematical formalism necessary to guarantee certain system properties.

In this paper we propose a framework called Behavior Control Lyapunov Functions (BCLF). It is our hope to combine the reactivity and operator transparency of the behavior based approach with the mathematical rigor of Lyapunov theory. This rigor will furthermore enable the operator to make the different mission objectives and their order of priority explicit in such a way that the autonomous system at all times can state which objectives are being fulfilled, which are focused on right now and what lower priorities are not addressed currently. Or conversely, what mission objective tradeoff, e.g. time of arrival vs. threat level, needs

*Corresponding author: `petter.ogren@foi.se`.

to be solved, and how the system intends to make that tradeoff. Finally, to avoid the drawbacks inherent in all reactive approaches, the framework accepts as one of the mission objectives to follow a given path, this path can be planned by a human operator, or produced by an on board planning algorithm, such as the one presented in.[15]

The organization of this paper is as follows. In Section II we review the main ideas of Lyapunov theory and discuss the structure of behavior based architectures. Section III then presents the BCLF framework and some analytical properties. Then, in Section IV, the approach is illustrated by a simulation example and conclusions are drawn in Section V.

II. Background: Lyapunov Theory and Behavior Based Architectures

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

A. Lyapunov Theory of Stability

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 minimum, then the system will end up in that minimum.

The definition below is taken from an excellent book by Sastry [12, p. 199].

Theorem II.1 ***(Lyapunov Theorem - LaSalle's Principle)*** *Suppose we are given a system $\dot{x} = f(x)$, $x(t_0) = x_0$, an equilibrium point $x^* = 0$ and a positive definite function $V(x)$. If $-\dot{V} \geq 0$, and the set $S = \{x \in \mathbb{R}^n : \dot{V}(x) = 0\}$ contains no trajectories except $x \equiv 0$, then x^* is globally asymptotically stable.*

We call the function $V(\cdot)$ a *Lyapunov function*.

Remark II.2 *In Section III we will make significantly stronger assumptions than the ones above, in order to prove finite time properties in a straight forward manner. So even though we do not use the above theorem explicitly, the ideas of this paper rest heavily upon the framework presented above.*

The step to control Lyapunov functions is now not far, but was defined much later, 1983, in a paper by Artstein.[3]

Definition II.3 (Control Lyapunov Function) *A positive definite C^1 function $V(x) : \mathbb{R}^n \to \mathbb{R}^+$ is called a control Lyapunov function if,*

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

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

B. Behavior Based Architectures

In a book as late as 1998, by Georgia Tech researcher R. Arkin[2] an approach called behavior based control is 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]

A typical reactive architecture is seen in Figure 1. The behaviors 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 `go-to-goal` behavior if there is an obstacle close by. This is the arbitration mechanism suggested by Brooks in the subsumption architecture[6] .
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.

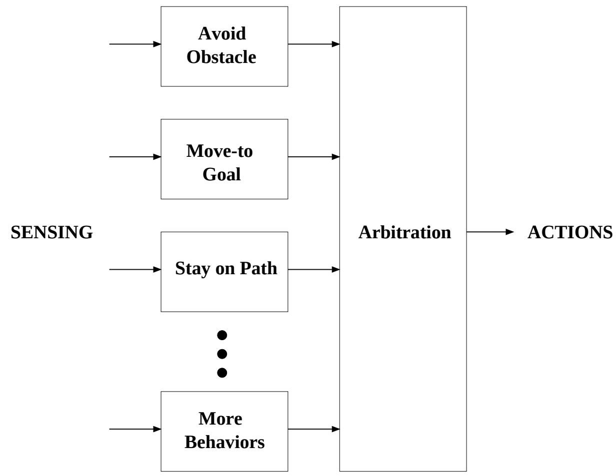

Figure 1. A Typical reactive architecture, from [2, p. 112-114].

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 Behavior Based robotics 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 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 as BCLFs

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. Examining our UCAV application, a set of reasonable objectives in a strike mission might be the following:

- Arrive at the target within the given time window and deliver the ordnance.
- Return to base without running out of fuel.
- Stay on, or close to, a given waypoint flight path.
- Stay outside of the range of SAM sites.
- Stay away from enemy fighters.

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 enemy fighters are patrolling the target area, or when a pop-up SAM is found to be blocking the way to the target, in such a way that there is a choice of either avoiding the threat and arriving too late on target, or arriving on time but enduring a higher risk of being shot down.

To make the five mission goals above explicit in terms of functions and inequality constraints we write

$$V_t = \left|(t_t - t) - \frac{\mathrm{dist}(p, p_t)}{v_{nom}}\right| \leq T_{\text{window}} \tag{1}$$

$$
\begin{aligned}
V_f &= -(x_f - \text{dist}(p, p_b)c) \leq 0 \\
V_p &= \{\text{distance from designated path}\} \leq 1km \\
V_s &= -\frac{||p - p_s||}{d_{range}} \leq -1 \\
V_e &= -||p - p_e|| \leq -50km
\end{aligned}
$$

where t_t is the desired time of arrival, p is the UCAV position, p_t is the target position, $\text{dist}(\cdot, \cdot)$ is the distance, following the waypoint path, between two positions, v_{nom} is the nominal speed of the UCAV, T_{window} is the size of the time window around t_t, c is fuel consumption, x_f is the amount of fuel left, p_b is the base position, p_e is the position of the closest enemy fighter, p_s is the position of the closest enemy SAM and d_{range} is the threat range of that particular SAM type. Note that we used $\leq$-constraints in all inequalities above, and that the right hand sides, called b_{ij} below, are all set by the operator before the mission. The main idea of this approach is then to choose controls that decrease the functions, and thus satisfy the inequalities and the corresponding mission objectives. For this to work, the functions must have some particular properties defined below.

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

Definition III.1 (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 ϵ 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 III.2 *If V is a BCLF, for b, ϵ, 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 $t \leq \frac{V(x_0) - b}{\epsilon}$. ∎

In our example we assume that we already have an autopilot accepting speed and heading commands. To capture the high level behavior and fuel consumption of the UCAV we choose the following model

$$
\begin{aligned}
\dot{x}_1 &= v \cos \phi, \\
\dot{x}_2 &= v \sin \phi, \\
\dot{x}_f &= -k_0 - k_1 v - k_2 v^2, \\
\dot{t} &= 1,
\end{aligned}
$$

where $(x_1, x_2)^T = p$ is position, x_f is the amount of fuel, ϕ is commanded heading and v is commanded speed, with the bound $v_{min} \leq v \leq v_{max}$. We write $x = (x_1, x_2, x_f, t)^T$ and $u = (v, \phi)^T$. We have included time t into the state to avoid explicit time dependence in $V(x)$. Finally, using a general second degree polynamial model in v for the fuel consumption, with constants $k_i > 0$, enables us to capture the effect of speed dependent fuel efficiency. Note that we use a planar UCAV model throughout this paper. This is just for convenience, and there are no general restrictions as to what kind of system $\dot{x} = f(x, u)$ can be.

Letting $\text{dist}(p, p_b) = ||(x_1, x_2) - p_b||$, we can now calculate e.g. $\dot{V}_s$ and $\dot{V}_f$.

$$
\begin{aligned}
\nabla V_f^T &= \left(\frac{c(x_1 - p_{b1})}{||(x_1, x_2) - p_b||}, \frac{c(x_2 - p_{b2})}{||(x_1, x_2) - p_b||}, -1, 0\right) \\
\nabla V_f^T f &= ((x_1, x_2) - p_b^T) \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} \frac{vc}{||(x_1, x_2) - p_b||} + k_0 + k_1 v + k_2 v^2 \\
\nabla V_s^T &= \left(\frac{(p_s - p)^T}{d_{range}||(p_s - p)||}, 0, 0\right) \\
\nabla V_s^T f &= \frac{-v}{d_{range}||(p_s - p)||} (p_s - p)^T \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix}
\end{aligned}
$$

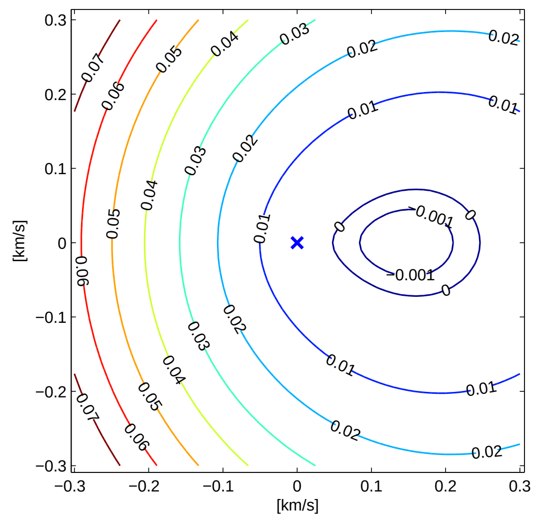

Figure 2. Level curves for $\nabla V_f^T f$, as a function of velocities $(v_1, v_2) = (v\cos\phi, v\sin\phi)$, with the goal direction to the right in the figure. Choosing a velocity towards the goal, within the 0-curve, makes the predicted fuel level when landing increase. Flying in the opposite direction, i.e. left in the figure, makes the predicted level decrease rapidly. The fuel consumption constants were set to $k_0 = 0.002, k_1 = 0.05, k_2 = 0.17, c = 0.1$.

Level curves for $\nabla V_f^T f$ and $\nabla V_s^T f$ are depicted in Figure 2 and Figure 4. Note that choosing e.g. $\epsilon = 0.001$ makes V_f a BCLF in the sense of Definition III.1 above.

So far we have only discussed one mission objective and one BCLF. As we saw in Section II, a key point of behavior based robotics is how to combine a set of different control behaviors. Our hope is to propose a method that enables the human operator to give very clear priorities between the different mission objectives.

Coming back to the application examples, it is conceivable that the highest mission priority might be to deliver the weapon on target, and the second highest priority might be to get the UCAV back to base with just enough fuel to land. When that can be achieved, the third priority might be to avoid known SAM sites, and equally important increase the fuel safety margin to say 10%. Mission objectives might thus re-appear on different priority levels with different values. In detail, a mission priority table might look like Table 1 and 2. Each column represents a set of bounds to the functions on the left hand side of (1). Note that the bound "$\leq \infty$" trivially holds and thus PL 0 is always satisfied, while the highest level, PL 6 in this case, is the hardest to achieve.

A high risk example is shown in Table 1. As can be seen, the highest priority level, PL 1, corresponds to arriving at the target location on time, PL 2 corresponds to reaching the target on time, as well as arriving back to the base without running out of fuel, and so on. Similarly, a low risk example is shown in 2, where fuel and threats are more important that time on target.

b_{ij}	PL 0	1	2	3	4	5	6
$V_1 = V_t \leq$	∞	5min	5min	5min	5min	5min	5min
$V_2 = V_f \leq$	∞	∞	0	-10%	-15%	-15%	-15%
$V_3 = V_p \leq$	∞	∞	∞	∞	∞	∞	$5km$
$V_4 = V_s \leq$	∞	∞	∞	-1	-1.2	-1.2	-1.2
$V_5 = V_e \leq$	∞	∞	∞	∞	∞	$-50km$	$-50km$

Table 1. High risk mode. Note e.g. that arriving on target, (PL 1), is more important than avoiding SAMs, (PL 3).

Given a priority table with $\{b_{ij}\}$ values, such as Table 1 and 2, one can formally define the current priority level as

b_{ij}	PL 0	1	2	3	4	5	6
$V_1 = V_t \leq$	∞	∞	∞	∞	$5min$	$5min$	$5min$
$V_2 = V_f \leq$	∞	-10%	-10%	-15%	-15%	-15%	-15%
$V_3 = V_p \leq$	∞	∞	∞	∞	∞	∞	$5km$
$V_4 = V_s \leq$	∞	∞	-1	-1	-1	-1	-1
$V_5 = V_e \leq$	∞	∞	∞	∞	∞	$-50km$	$-50km$

Table 2. Low risk mode. Note that coming back with a 15% fuel margin and avoiding SAMs, (PL 3), is more important than reaching the target area, (PL 4).

Definition III.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 UCAV can now be stated in terms of maximizing the CPL, i.e. move the system as far down the priority level list as possible.

B. 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 on the next PL, while not violating the constraints of the current PL. Below we will see how this can be made more precise.

Definition III.4 (Control Sets) *Fix $k > 0$ and let $\epsilon > 0$ be the same as in Definition III.1. 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.*

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

Lemma III.5 *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 III.2. ∎

Remark III.6 *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.*

Remark III.7 *Note that in cases where a tradeoff must be made, e.g. when a target point is located within a SAM threat area, the sets $U_{sat}(x)$ and $U_{inc}(x)$ 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.*

Before going into details about controller design we look at one example of what U_{sat} and U_{inc} might look like.

Example III.8 *Consider the situation depicted in Figure 3, where the UCAV is heading east towards the base when a threat in terms of a SAM-site appears 30km to the north-east. The amount of remaining fuel is $x_f = 16\%$, with a nominal fuel consumption $c = 0.1\%/km$, the SAM distance is $||p - p_s|| = 30km$ and the nominal threat range (dotted circle) is $d_{range} = 24km$. Furthermore, the priorities of Table 2 where used and $k = 50$.*

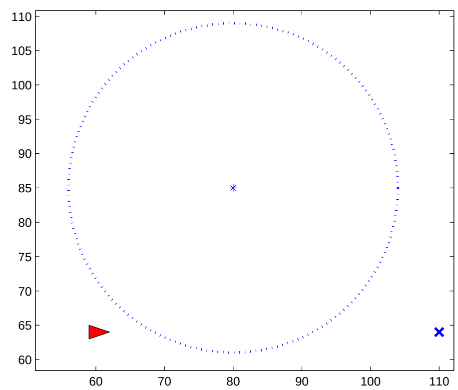

Figure 3. The UCAV (triangle) at $(60, 64)$ is heading towards the base (x) at $(110, 64)$ when a SAM (*) pops up at $(80, 85)$.

We start with calculating the BCLFs:

$$\begin{aligned} V_f &= -(x_f - dist(p, p_b)c) = -(16 - 50 \cdot 0.1) = -11 \\ V_s &= -\frac{||p - p_s||}{d_{range}} = \frac{30}{24} = -1.25 \end{aligned}$$

There are no known fighters airborne, making $V_F = -\infty$. The UCAV is on the path, $V_p = 0$, and since the target is already passed, V_t is defaulted to $V_t = 0$.

Looking at table 2, we have CPL=2, since the bound $V_2 = V_f = -11 \leq b_{23} = -15$ does not hold. Calculating U_{sat} we get

$$\begin{aligned} U_{sat} = \{u : \quad & \dot{V}_1(x,u) \leq \frac{1}{50}(\infty - 0) = \infty, \\ & \dot{V}_2(x,u) \leq \frac{1}{50}(-10 - (-11)) = 0.02, \\ & \dot{V}_3(x,u) \leq \frac{1}{50}(\infty - 0) = \infty, \\ & \dot{V}_4(x,u) \leq \frac{1}{50}(-1 - (-1.25)) = 0.005, \\ & \dot{V}_5(x,u) \leq \frac{1}{50}(\infty - (-\infty)) = \infty\} \end{aligned}$$

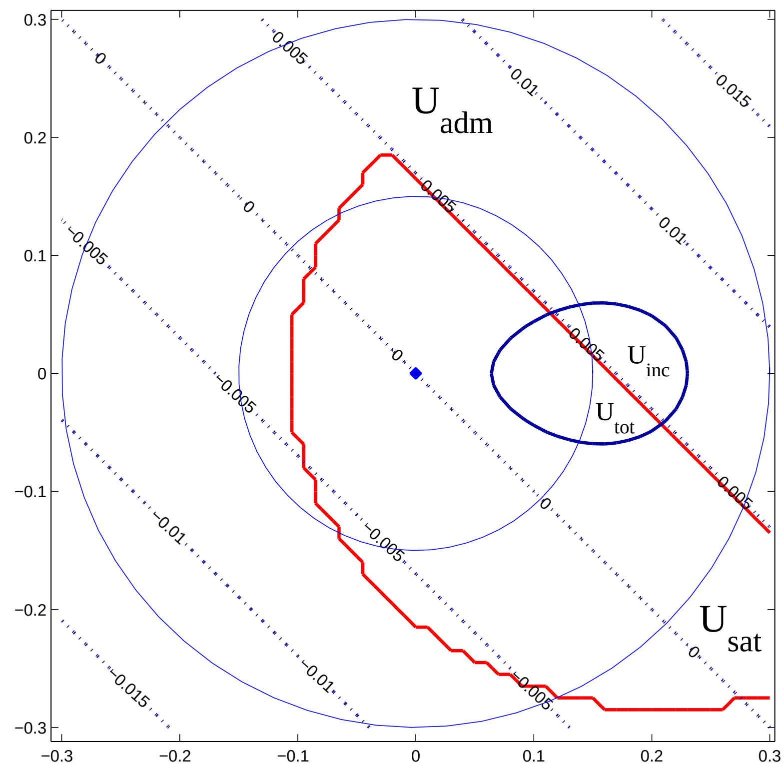

Figure 4. The important control sets of the example shown in Figure 3. Note that the recommended control set $U_{tot} = U_{adm} \cap U_{sat} \cap U_{inc}$ corresponds to heading east-south-east at a fuel efficient speed, which is reasonable, looking at Figure 3. Level curves of $\nabla V_s^T f$, dotted, are also shown in the figure.

i.e. the set shown in Figure 4, which is the intersection of the set within the 0.02 *bound in Figure 2, and below the* 0.005 *line in Figure 4.*

Going on to calculate $I_{next}(x)$ we get

$$I_{next}(x) = \{i : V_i(x) \geq b_{i3}\} = \{2\},$$

since the fuel bound of 15% *is the only objective not satisfied at PL 3. This makes*

$$U_{inc}(x) = \{u : \dot{V}_2(x, u) \leq -0.001\},$$

the corresponding level curve can be found in Figure 2. The sets U_{sat} and U_{inc} are both depicted in Figure 4, together with $U_{adm} = \{(v, \phi) : 150m/s < v < 300m/s\}$, representing the flight envelope of the UCAV, i.e. the set of physically achievable velocities.

To summarize, to avoid violating the CPL=2, i.e. to retain a distance > 24km to the SAM and a predicted fuel margin of 10% *, the UCAV must apply a control in the set $U_{sat} \cap U_{adm}$ i.e. either move slowly north, or chose an arbitrary speed heading south-east. In order to improve the chances of reaching CPL=3, i.e. increase the estimated fuel margin to* 15%*, a control can be chosen in $U_{tot} = U_{sat} \cap U_{adm} \cap U_{inc}$, corresponding to flying east-south-east with a fuel efficient speed.*

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

C. Controller Synthesis

As seen above, choosing a control in $U_{sat} \cap U_{inc}$ is attractive, but as noted in Remark III.7, one or both of these sets might be empty.

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 III.9 *Let k be the same as in Definition III.4. Let furthermore*

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

and

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

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

Lemma III.10 *The set $U_{dec}(x)$ is never empty.*

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

With these sets we are ready to define the control set to be used in the controller design.

Definition III.11 *Define the set U_{tot} as follows:*
If $U_{inc} \cap U_{sat} \cap U_{adm} \neq \emptyset$ then $U_{tot} = U_{inc} \cap U_{sat} \cap U_{adm}$,
else if $U_{sat} \cap U_{adm} \neq \emptyset$ then $U_{tot} = U_{sat} \cap U_{adm}$,
else $U_{tot} = U_{dec}$.

Note the properties of U_{tot}, as stated in Lemma III.5 and III.10.

The remaining question is what control to choose in U_{tot}. It is clear that the objectives in I_{next} should be focused on. We suggest the following:

$$u^* = \text{argmin}_{u \in U_{tot}} \Sigma_{i \in I_{next}} k_i \dot{V}_i(x,u), \tag{2}$$

where

$$k_i = \frac{V_i(x) - b_{i(j_{CPL}(x)+1)}}{||\nabla V_i(x)||},$$

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

We can now compare this control structure to the different arbitration alternatives in Behavior Based Robotics, 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 ideas from Behavior Based Robotics, using a mathematic framework inspired by CLFs.

In the next section we illustrate the approach with an example mission.

IV. Simulation Examples

Consider the mission setup and trajectories depicted in Figure 5. Two UCAVs are simulated using the priorities of Table 3, one with desired time of arrival $t_T = 840$ and one with $t_T = 720$.

At the start near A, both UCAVs are in CPL 4, and $I_{next} = \{3\}$, since the distance to the planned path is larger than 10km. This makes both UCAVs move straight towards the closest part of the path, to increase the CPL.

Moving towards B, the distance to the enemy aircraft is close to the desired minimum range of 50km. Note that the small triangles represent three different snapshots of the UCAVs and enemy aircraft, at the first two snapshots the two UCAVs have not parted yet.

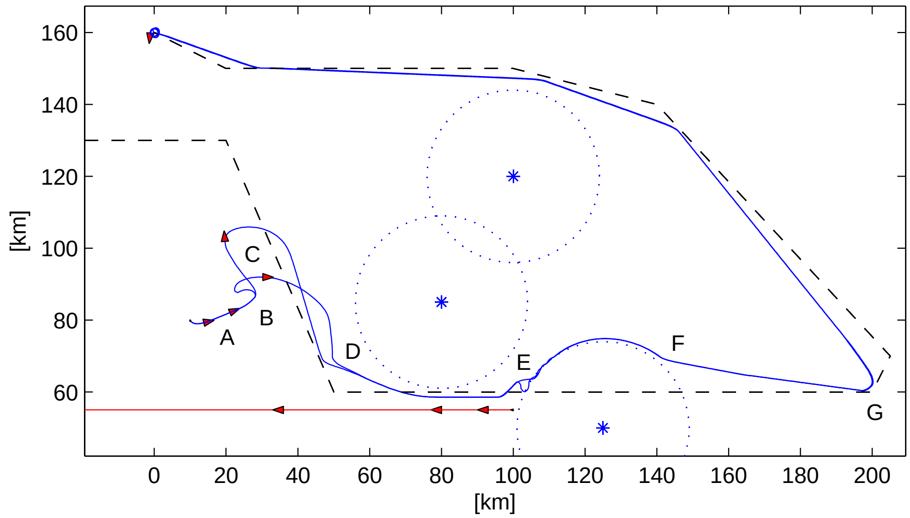

Figure 5. Two different UCAV mission trajectories. Both UCAVs start at $(10, 80)$, an enemy aircraft starts at $100, 50$, the dashed line is the planned waypoint trajectory, passing the two dotted SAM threats in the middle. The third SAM threat is a pop-up, discovered during the mission.

b_{ij}	PL 0	1	2	3	4	5	6
$V_1 = V_t \leq$	∞	$120s$	$120s$	$120s$	$120s$	$120s$	$120s$
$V_2 = V_f \leq$	∞	∞	-10%	-10%	-10%	-10%	-100%
$V_3 = V_p \leq$	∞	∞	∞	∞	∞	$10km$	$10km$
$V_4 = V_s \leq$	∞	∞	∞	-1	-1	-1	-1
$V_5 = V_e \leq$	∞	∞	∞	∞	$-50km$	$-50km$	$-50km$

Table 3. The priority table of the example.

At B both UCAVs have to turn to avoid getting too close to the fighter. The UCAV with a later desired time of arrival flies away from the threat. For the other UCAV, V_t is close to the bound. U_{sat} is empty, there is no way to achieve the time of arrival and the threat distance at the same time. This makes $j_{CPL-d} = 3$ and U_{dec} does not contain the $50km$ constraint since arrival time is of higher priority. However $I_{next} = \{5\}$ and a heading is chosen to maximize the threat distance, while not violating the time constraint, thus the UCAV is not heading straight for the next waypoint.

At C, the upper UCAV still maintains the safety distance of 50km while the lower is roughly 37km away from the threat. After the threat has passed both UCAVs proceed to the next waypoint.

At D, both UCAVs switch waypoints. Since they are within 10km of the waypoint and the next path segment, switching does not decrease the CPL. Note that the BCLF framework can thus also be used to decide when to switch waypoints, this is however not the main topic of the paper.

At E, the radar warning receiver of both UCAVs detect an a priori unknown SAM, and there is a conflict between the path, the time of arrival and the SAM threat distance constraints. The UCAV with earliest time of arrival is still close to violating it. Thus U_{sat} is again empty, the path constraint is removed in U_{dec} and the UCAV proceeds to pass the SAM. The other UCAV makes a couple of turns, to stay at CPL=5 as long as possible. This is where a replanning functionality would be used to supply a new path without a inherent path/threat conflict. In this simulation however, replanning is not integrated, no new plan is given, and once the time of arrival objective is in danger U_{dec} again removes the path constraint.

At F, $I_{next} = \{3\}$ makes the UCAV follow the threat circle until the path constraint is again satisfied. Then a straight heading towards the next waypoint is chosen.

At G, where the target is, the UCAVs do not switch waypoints at 10km, but proceed to deliver the ordnance. The first UCAV was 83s after its desired time on target while the second one was 59s late. Both times well inside the 120s constraint of the priority table. Having made the drop, both UCAVs then proceed to the home base.

Apart from the low fidelity simulations presented above, high fidelity, man-in-the-loop simulations were also carried out at the Swedish Air Force Air Combat Simulation Center (FLSC). These simulations are part of an ongoing effort to explore the advantages and disadvantages of unmanned air combat systems.

V. Conclusions

Reactive approaches are fast, and easy to understand and predict from an operator point of view. However, they often lack the mathematical rigor and preciseness of model based control schemes. In this paper we have proposed an approach that combines the transparency of behavior based robotics with tools from Lyapunov theory to enable an operator to state mission objective preferences in a clear and detailed way. Further more, we specify the control sets that keep and/or improve the current level of mission objective satisfaction. A simulation example was presented to illustrate the approach.

References

[1]Vitaly Ablavsky, Daniel Stouch, and Magnús Snorrason. Search path optmization for uavs using stochastic sampling with abstract pattern descriptors. In *AIAA Guidance, Navigation, and Control Conference and Exhibit 11 - 14 August 2003, Austin, Texas*, 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]T. Balch and R. Arkin. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, pages 926–939, Dec 1998.

[5]Randal W. Beard, Timothy W. McLain, Michael A. Goodrich, and Erik P. Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation, Vol. 18, NO. 6*, December 2002.

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

[7]Derek S. Caveney and J. Karl Hedrick. Path planning for targets in close proximity with a bounded turn-rate aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, California, 2005.

[8]Atilla Dogan. Probabilistic path planning for uavs. In *AIAA Unmanned Unlimited Systems, Technologies, and Operations*, San Diego, California, 2003.

[9]Yeonju Eun and Hyochoong Bang. Cooperative control of multiple ucavs for suppression of enemy air defense. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit 20 - 23 September 2004, Chicago, Illinois.*

[10]Bernard Mettler. Combining on- and offine optimization techniques for effcient autonomous vehicle's trajectory planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit 15 - 18 August 2005, San Francisco, California.*

[11]Ryan L. Pettit and Mark L. Homer. An autonomous threat evasion response algorithm for unmanned air vehicles during low altitude flight. In *AIAA 1st Intelligent Systems Technical Conference 20 - 22 September 2004, Chicago, Illinois.*

[12]S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer-Verlag, New York, 1999.

[13]K. Savla, F. Bullo, and E.Frazzoli. On traveling salesperson problems for Dubins' vehicle: stochastic and dynamic environments ,. In *IEEE CDC-ECC, Seville, Spain, 2005.*

[14]Ugur Zengin and Atilla Dogan. Dynamic target pursuit by uavs in probabilistic threat exposure map. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit 20 - 23 September 2004, Chicago, Illinois.*

[15]P. Ögren and M. Winstrand. Combining path planning and target assignment to minimize risk in a sead mission. In *AIAA Guidance, Navigation, and Control Conference and Exhibit 15 - 18 August 2005, San Francisco, California.*