A Robust Model-Predictive Guidance System for Autonomous Vehicles in Cluttered Environments

Peter Travis Jardine, *Student Member, IEEE* and Sidney N. Givigi, *Senior Member, IEEE*

***Abstract*—This paper proposes a system for guidance of autonomous vehicles based on a generalized robust model-predictive control (R-MPC) technique. It offers two scenarios, in which R-MPC is used to provide optimized motion plans guaranteed to satisfy constraints and avoid obstacles in the presence of bounded uncertainty. The first scenario involves a military unmanned aerial vehicle flying over a target, while avoiding enemy defenses. The second scenario involves an assistive care robot safely navigating to waypoints throughout a cluttered home. In each case, obstacles are represented as a collection of linear inequality constraints that adapt to changes in the environment. Results show that when R-MPC is formulated using a series of positive-semidefinite relaxations on linear inequality constraints, safe optimum trajectories can be planned by solving a convex quadratic program. When formulated in terms of perturbations on feedback predictions, this solution is guaranteed to be robustly stable.**

***Index Terms*—Autonomous systems, navigation, optimization, path planning, predictive control, robust control.**

I. Introduction

An essential function of many autonomous vehicles is the ability to navigate in the presence of obstacles [1], [2]. For certain applications, such as military unmanned aerial vehicles (UAVs), these vehicles normally require a significant amount of human control, particularly for high-level decision making [3]. This presents a number of limitations, including the high communication bandwidth used for command and control and the additional time required to include humans in the decision loop [4]. Overcoming these limitations motivates research into autonomous vehicles capable of making onboard decisions. Optimal motion planning techniques should anticipate the future consequences of control actions while making the most efficient use of limited resources [5], [6].

Model-predictive control (MPC) has become a popular technique for control of vehicles, both manned [7] and unmanned [8], [9]. More recently, MPC has emerged as a popular framework for planning in complex environments because of its ability to produce optimized solutions, while guaranteeing adherence to state and input constraints [10], [11]. The main limitation of MPC is the complex computation required during optimization, which can stymie attempts at real-time implementation [12]. However, given certain assumptions, the MPC problem can be solved efficiently as a convex optimization [13]. Inequality constraints have been used to achieve obstacle avoidance for MPC-based motion planning. These techniques typically incorporate constraint softening to avoid feasibility issues caused by uncertain disturbances [14]. Therefore, they do not provide guarantees of stability and constraint adherence in the presence uncertainty. Such robust guarantees are possible if the worst-case scenario of bounded uncertain disturbances is used [15]. While these robust techniques have been thoroughly investigated from a theoretical perspective, it is desirable to consider their application in the context of a system of systems [16].

This paper investigates robust model-predictive control (R-MPC)-based guidance techniques for autonomous vehicles that must avoid moving obstacles while subjected to bounded uncertain disturbances. The approach is robust to uncertainty (noise) in the inputs, measurements, predictions, and modeling represented as noise in the states. A general problem is defined, which uses linear system dynamics and a quadratic objective function. This is applicable to any problem in which a vehicle moving in a plane must get to some position, while saving energy and satisfying constraints. It is then demonstrated how these constraints, such as obstacles, can be represented as a collection of constraints that adapt to changes in the environment. The problem is formulated in terms of perturbations on ideal feedback predictions and positive-semidefinite relaxations on linear inequality constraints. When formulated in the form of R-MPC, this results in a convex optimization problem. The solution is a sequence of optimized control inputs that are guaranteed to satisfy constraints and avoid obstacles in the presence of bounded uncertain disturbances. Furthermore, given certain conditions, the system is robustly stable.

In order to demonstrate that this technique can be used for a broad range of applications without any significant change in formulation, it is applied to two different scenarios. The first scenario involves a military UAV flying over a target while avoiding enemy defenses. The second scenario involves an assistive care robot safely navigating to waypoints throughout a cluttered home. The guidance and navigation system is shown to be modular and portable and able to be used in different applications with minimal changes.

The choice of scenarios is driven by situations in which unmanned vehicles are becoming more common. In the first scenario, for UAV attack scenarios, prolonged exposure to enemy defenses could result in mission failure [4], especially when

Manuscript received December 7, 2017; accepted February 11, 2018. Date of publication April 16, 2018; date of current version May 31, 2019. *(Corresponding author: Sidney N. Givigi.)*

The authors are with the Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, ON K7K 7B4, Canada (e-mail: peter.jardine@rmc.ca; sidney.givigi@rmc.ca).

Digital Object Identifier 10.1109/JSYST.2018.2806629

limited resources and/or sensory input are available. As for the second scenario, an increase in life expectancy throughout much of the developed world has led to a large number of elderly people with physical and cognitive limitations. In [17], the authors propose a mobile robot to assist with routine tasks around a home such as access to interactive media, reminders of scheduled medication dosages, and dispensing of personal items. End-user studies identify safety as a critical requirement for such systems. This necessarily includes guarantees of collision avoidance and the ability adapt to changes in the environment based on information collected using onboard sensors.

To summarize, the contributions of this paper are as follows.

1) It formulates a generalized convex R-MPC guidance technique to provide optimized control inputs, while guaranteeing constraint satisfaction and obstacle avoidance in the presence of bounded uncertain disturbances.
2) It demonstrates that this technique can be applied to two different applications (military UAV and assistive elderly-care robot) without any significant change in formulation.
3) It develops a novel approach for formulating large rectangular objects as linear inequality constraints in an R-MPC application.
4) It provides a proof of stability for the system.
5) It shows through simulations that the system can be ported to different applications with minimal changes.

This paper is organized as follows. Section II describes the notation used throughout this paper. Section III defines the problem. Section IV describes the generalized R-MPC solution. Section V provides a stability analysis. Section VI presents results of a simulated military UAV attack mission. Section VII presents results of an assistive elderly-care robot. Section VIII concludes this paper.

II. Notation

Throughout this paper, $\oplus_{i=1}^{b}$ is used to denote the diagonal concatenation of b matrices. For example:

$$\oplus_{i=1}^{3} a_i = \begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_2 & 0 \\ 0 & 0 & a_3 \end{bmatrix}. \tag{1}$$

Column vectors representing vehicle state error $e_k \in \mathbb{E}$, perturbations $g_k \in \mathbb{G}$, disturbances $w_k \in \mathbb{W}$, measured outputs $\bar{y}_k \in \mathbb{Y}$, input changes $\Delta u_k \in \mathbb{U}$, and states $\bar{\gamma}_k \in \mathbb{X}$ at timestep k are expanded over a finite prediction horizon N as $E \in \mathbb{E}^N$, $G \in \mathbb{G}^N$, $W \in \mathbb{W}^N$, $Y \in \mathbb{Y}^N$, $\Delta U \in \mathbb{U}^N$, and $\Gamma \in \mathbb{X}^N$, respectively. The bar over $\bar{\gamma}_k$ means this state has been extended to include inputs from the previous timestep $k-1$, as described in Section III-B.

Matrices $\bar{A}$, $\bar{B}$, and $\bar{C}$ describe the system dynamics in terms of $\bar{\gamma}_k$ and Δu_k. These dynamics are expanded over the prediction horizon as $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, and $\boldsymbol{\mathcal{C}}$. The notation $\tilde{A} \succeq 0$, $\tilde{A} \succ 0$, $\tilde{A} \preceq 0$, $\tilde{A} \prec 0$ describes matrix $\tilde{A}$ as either positive semidefinite, positive definite, negative semidefinite, or negative definite, respectively. Similarly, $\tilde{A} \geq 0$, $\tilde{A} > 0$, $\tilde{A} \leq 0$, and $\tilde{A} < 0$ describe elementwise inequalities. $|A|$ represents the elementwise absolute value. Matrix $\tilde{A}$ has transpose $\tilde{A}^T$ and inverse $\tilde{A}^{-1}$. Matrix $\tilde{A}^{a\times b}$ has a rows and b columns. Matrices $0^{a\times b}$ and $I^{a\times b}$ represent the zero and identity matrices with dimensions indicated. $\mathbf{1}^a$ is a column vector with a elements. Additional notation is described as required.

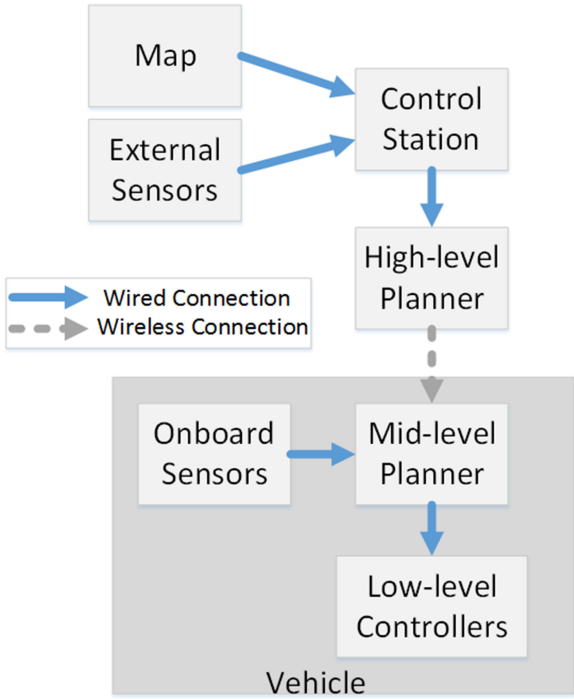

Fig. 1. System architecture for the guidance system.

III. Problem Statement

In this section, the guidance system is described and the generalized vehicle planning in the form of linear–quadratic MPC is formulated.

A. System Description

The solution of the guidance problem involves the use of several subsystems that encompass localization, navigation, and control of the vehicles [18]. As illustrated in Fig. 1, the vehicles need a map of the environment in order to plan their trajectory. This map can take different forms.

In the UAV case discussed in Section VI, the map is only a geographic map of the region of operation and the fixed obstacles. Notice though that details of the position of the enemy batteries are not strictly necessary (although it could be included in the formulation without loss of generality). Furthermore, the external sensors in this case provide the position of the target. This can come from a satellite, for example. In the home-care example discussed in Section VII, the map is an occupancy grid and no external sensor is necessary.

The control station receives the map and data from sensors and creates a high-level plan. This plan is mission specific and depends on data entered by a user. Notice, however, that the user does not need to interact with the system at all times. He could,

for instance, create guidelines that would be used by the planner to generate desired behaviors.

The actual plan is generated iteratively by the mid-level planner. This is necessary in order to react to moving obstacles, for which the vehicles need to rely on onboard sensors (see Fig. 1). This planner uses the dynamics and the MPC strategy introduced in the next sections to come up with the detailed plan while taking into account the low-level controllers.

B. Vehicle Dynamics

This section develops a linear model that is applicable to any vehicle traveling in a 2-D plane. Consider a vehicle with position coordinates x, y, and θ, forward velocity v, angular velocity ω, and dynamics governed by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\cos\theta \\ v\sin\theta \\ \omega \end{bmatrix} \tag{2}$$

where $\dot{x}, \dot{y}$ and $\dot{\theta}$ are the time rate of change in position and orientation, respectively. The orientation θ is the heading of the autonomous vehicle in the xy plane. Angular velocity is defined as the rate at which the orientation changes with time; therefore, $\dot{\theta} = \omega$. Assuming $v \neq 0$ and letting the input vector $u = [u_1\ u_2]^T = [\ddot{x}\ \ddot{y}]^T$, we use an exact linearization through dynamic extension to obtain the following discrete linear state-space representation of the vehicle dynamics:

$$\underbrace{\begin{bmatrix} \gamma_{1,k+1} \\ \gamma_{2,k+1} \\ \gamma_{3,k+1} \\ \gamma_{4,k+1} \end{bmatrix}}_{\gamma_{k+1}} = \underbrace{\begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \gamma_{1,k} \\ \gamma_{2,k} \\ \gamma_{3,k} \\ \gamma_{4,k} \end{bmatrix}}_{\gamma_k} + \underbrace{\begin{bmatrix} \frac{dt^2}{2} & 0 \\ 0 & \frac{dt^2}{2} \\ dt & 0 \\ 0 & dt \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} u_{1,k} \\ u_{2,k} \end{bmatrix}}_{u_k} \tag{3}$$

where $\gamma_1 = x$, $\gamma_2 = y$, $\gamma_3 = \dot{x}$, and $\gamma_4 = \dot{y}$; A and B are the system and input matrices; and γ_k and u_k are the state and input vectors at timestep k. The vehicle position is measured directly according to the measurement model:

$$y_k = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{C} \gamma_k \tag{4}$$

where y_k is the measured output vector and C is the measurement matrix.

Notice that the system in (3) and (4) describes a double integrator. However, the dynamics could be expanded to encompass different dynamical behavior as required.

Given linear system dynamics, the inputs can be redefined in terms of input increments as $\Delta u_k = u_k - u_{k-1}$, and the standard linear state-space model extended to include the new variable $\gamma_{u,k} = u_{k-1}$:

$$\underbrace{\begin{bmatrix} \gamma_{k+1} \\ \gamma_{u,k+1} \end{bmatrix}}_{\bar{\gamma}_{k+1}} = \underbrace{\begin{bmatrix} A & B \\ 0 & I \end{bmatrix}}_{\bar{A}} \underbrace{\begin{bmatrix} \gamma_k \\ \gamma_{u,k} \end{bmatrix}}_{\bar{\gamma}_k} + \underbrace{\begin{bmatrix} B \\ I \end{bmatrix}}_{\bar{B}} \Delta u_k \tag{5}$$

where $\bar{\gamma}_k$ is the new state vector including γ_k and $\gamma_{u,k}$. The measurement model is also extended to include this additional state:

$$\bar{y}_k = \underbrace{\begin{bmatrix} C & 0 \end{bmatrix}}_{\bar{C}} \bar{\gamma}_k \tag{6}$$

where $\bar{y}_k$ indicates a measurement derived from the extended state vector $\bar{\gamma}_k$. In most of the autonomous vehicles applications, $\bar{y}_k$ would be the position of the vehicle measured with navigation sensors (global positioning system, for example) or simultaneous localization and mapping.

By expanding the system dynamics as a series of interconnected linear systems N timesteps into the future, we explicitly obtain the set of future measurements in terms of the initial states and inputs as follows [19]:

$$\underbrace{\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \vdots \\ \bar{y}_N \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} \bar{C} & 0 & \dots & 0 \\ 0 & \bar{C} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{C} \end{bmatrix}}_{\boldsymbol{\mathcal{C}}} \left(\underbrace{\begin{bmatrix} \bar{A} \\ \bar{A}^2 \\ \vdots \\ \bar{A}^N \end{bmatrix}}_{\boldsymbol{\mathcal{A}}} \bar{\gamma}_0 \underbrace{\begin{bmatrix} \bar{B} & 0 & \dots & 0 \\ \bar{A}\bar{B} & \bar{B} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}^{N-1}\bar{B} & \bar{A}^{N-2}\bar{B} & \dots & \bar{B} \end{bmatrix}}_{\boldsymbol{\mathcal{B}}} \underbrace{\begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix}}_{\Delta U} \right) \tag{7}$$

or, simply

$$Y = \boldsymbol{\mathcal{C}}\Gamma = \boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{A}}\bar{\gamma}_0 + \boldsymbol{\mathcal{B}}\Delta U). \tag{8}$$

C. Objective Function

Consider a reference measurement y_r representing the desired vehicle position, velocity, or orientation for the vehicle. The vector y_r (later stacked as Y_r) is in general the target location of the vehicle. This reference could also represent a desired track or the location of a stationary target. We define Y, ΔU, and Y_r as the stacked set of predicted output, input, and stationary reference/target vectors over the prediction horizon N. Using these parameters, we then define an objective function as [13]

$$J(\bar{\gamma}_0, \Delta U) = (Y - Y_r)^T \boldsymbol{\mathcal{Q}}(Y - Y_r) + \Delta U^T \boldsymbol{\mathcal{R}} \Delta U + \bar{\gamma}_N^T \bar{P} \bar{\gamma}_N \tag{9}$$

where $J(\bar{\gamma}_0, \Delta U)$ is the objective cost, $\boldsymbol{\mathcal{Q}} = \oplus_{i=1}^{N} \bar{Q}_i$ and $\boldsymbol{\mathcal{R}} = \oplus_{i=1}^{N} \bar{R}_i$ such that $\bar{Q}_i$ and $\bar{R}_i$ are the weighting matrices for the outputs and inputs at each step, and $\bar{P}$ is the weight of the terminal cost. The design parameters $\bar{Q}_i$ and $\bar{R}_i$ are typically selected through trial and error but can also be determined using more advanced techniques, such as reinforcement learning. The interested reader is referred to [20].

By minimizing (9), we minimize the difference between the vehicle and the reference $(Y - Y_r)$ as well as the energy expended through control inputs (ΔU) over N. Therefore, we define the optimum sequence of control inputs ΔU^*, which minimizes (9) as follows:

$$\Delta U^* = \arg\min_{\Delta U} \quad J(\bar{\gamma}_0, \Delta U). \tag{10}$$

Obstacle avoidance and system constraints are imposed by constraining (10). The formulation of these constraints is described in greater detail in Section IV-A.

IV. R-MPC Solution

Almost all practical applications of MPC for vehicle motion planning involve uncertainty. This section shows how the optimization (10) can be solved while ensuring that the constraints are satisfied when system is subjected to bounded uncertain disturbances in the form of system noise. This solution is based on the early work of [21] and [22], who defined the *min–max* MPC approach involving a maximization of bounded disturbances inside of the greater overall minimization. Several variable substitutions allow this complex dual optimization to be solved as a single convex minimization using semidefinite relaxations on linear inequality constraints [25].

A. System Constraints and Obstacle Avoidance

This section formulates a set of linear inequalities to constrain the optimization in (10). These inequalities are chosen such that they ensure the optimum sequence of control inputs adheres to system constraints and avoid obstacles over the prediction horizon. Normally, imposing a constraint on the system states based on the location of obstacles creates a space of feasible solutions that is nonconvex [14]. Therefore, the optimization in Section III-C would become nonconvex and difficult to solve. As shown in Fig. 2, by defining a minimum *safe zone* around each obstacle, we can constrain the set of feasible solutions of (10) to a convex polytope through the combination of linear inequalities [14]. These inequalities are drawn tangent to specific points around the obstacles.

Given the position of the vehicle (x_v, y_v) and position of an obstacle (x_e, y_e), we define a circle representing the minimum *safe zone* of radius r_e around the obstacle. This *safe zone* is defined such that it considers the size of the obstacle and vehicle. We define the straight line with length l_e and slope m_e between the center of mass of the two objects. The point (x_p, y_p) lies on this line at a distance r_e from the obstacle. The linear inequality constraint is defined by the line with slope perpendicular to m_e, tangent to the circle drawn around the obstacle of radius r_e and intersecting (x_p, y_p).

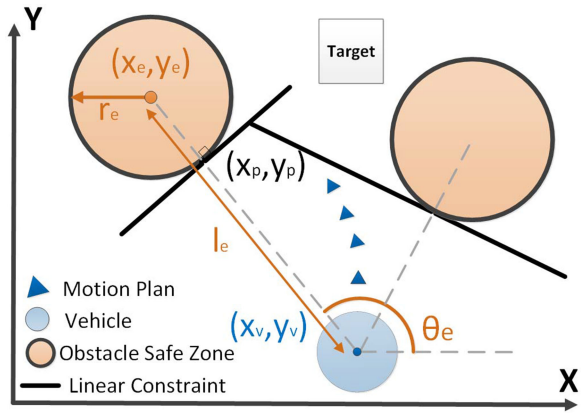

Fig. 2. Illustration of linear inequality constraints for obstacle avoidance.

Given slope m_e computed as follows:

$$m_e = \frac{y_e - y_v}{x_e - x_v} \tag{11}$$

then $m_e' = -1/m_e$ is the slope perpendicular to m_e. The angle θ_e between the vehicle and the obstacle with respect to the x-axis is computed as

$$\theta_e = \arctan(m_e). \tag{12}$$

Assuming that the distance between the two objects is l_e, we use θ_e to find (x_p, y_p) as follows:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} (l_e - r_e)\cos\theta_e + x_v \\ (l_e - r_e)\sin\theta_e + y_v \end{bmatrix}. \tag{13}$$

Therefore, for each obstacle at a given timestep, a linear inequality can be used to constrain the vehicle states to a region above or below the line with slope m' intersecting (x_p, y_p) as follows:

$$\begin{aligned} \begin{bmatrix} -m_e' & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \bar{\gamma} &\le -m_e{}' x_p + y_p, \quad \text{if } \pi \ge \theta_e > 0 \\ \underbrace{\begin{bmatrix} m_e' & -1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{M_{e,k}} \bar{\gamma} &\le \underbrace{m_e{}' x_p - y_p}_{f_{e,k}}, \quad \text{if } \pi < \theta_e \le 2\pi \end{aligned} \tag{14}$$

which is a relationship conditional on the value of θ_e. This condition on θ_e ensures that the vehicle is forced to the correct side of the linear constraint. The values of the constraint matrices $M_{e,k}$ and the constraint vectors $f_{e,k}$ will differ for each obstacle at timestep k. This linear inequality constraint is assumed constant for the entire prediction horizon. Once the first optimized input is executed, the vehicle (and possibly the obstacle) will have moved. An updated set of linear constraints is then developed and used for the next optimization. This process, often referred to as *receding horizon*, continues for each timestep until the target is reached. By updating the constraints at each timestep, the plan is able to adapt to changes in the environment.

Since we assume that the obstacle is static throughout the prediction horizon, we can use (14) to construct the constraint matrix M_e and constraint vector f_e for a single obstacle over

horizon N as

$$\underbrace{\oplus_{i=1}^{N} M_{e,i}}_{M_e}\,\Gamma \leq \underbrace{\begin{bmatrix} f_{e,1} & f_{e,2} & \dots & f_{e,N} \end{bmatrix}^T}_{f_e}. \tag{15}$$

Additional obstacles are incorporated by stacking the corresponding solutions to (15). Using the relationship in (8), these obstacle constraints can be combined with other constraints on the system states and inputs as follows:

$$\begin{bmatrix} M_\gamma \boldsymbol{\mathcal{B}} \\ M_e \boldsymbol{\mathcal{B}} \\ M_u \end{bmatrix} \Delta U \leq \begin{bmatrix} f_\gamma - M_\gamma \boldsymbol{\mathcal{A}} \gamma_0 \\ f_e - M_e \boldsymbol{\mathcal{A}} \gamma_0 \\ f_u \end{bmatrix} \tag{16}$$

where M_γ and f_γ define constraints on the system states, and M_u and f_u define constraints on the inputs.

B. Incorporating Feedback Predictions

This section defines an ideal unconstrained feedback control law to stabilize the system at each prediction step, while treating constraint handling as perturbations on this stabilizing law. These perturbations are then treated as an optimization variable, rather than the inputs themselves. This *closed-loop paradigm* technique introduces an underlying stability provided these perturbations (i.e., the effect of constraints) converge to zero over time and allow the unconstrained stabilizing law to take over [23], [24]. It is also a necessary condition for the formal robust stability analysis in Section V.

Let us define the state *error* term as the difference between the current vehicle state $\bar{\gamma}_k$ and the target state γ_r

$$e_k = \bar{\gamma}_k - \gamma_r. \tag{17}$$

If we redefine our coordinate system such that the target state is at the origin, then

$$e_k = \bar{\gamma}_k. \tag{18}$$

Notice that this condition is over conservative and may not be feasible in actual applications such as when fixed wing UAVs are used. However, as pointed out in [25], the condition is added in order to provide a way to prove convergence, even if it will not be used in practice. We define a nominal input using the following unconstrained stabilizing feedback law:

$$\Delta u_k = -K_{cl}(e_k) + g_k \tag{19}$$

where K_{cl} is the stabilizing gain and g_k is a perturbation on the law. We can now substitute (19) and (18) into (5) and letting $\Phi = \bar{A} - \bar{B}K_{cl}$ obtain the error dynamics as a function of the perturbations

$$e_{k+1} = \Phi e_k + \bar{B} g_k. \tag{20}$$

Similar to the derivation of (7), we obtain the error predictions over finite horizon N as

$$E = \boldsymbol{\mathcal{S}_e} G + \boldsymbol{\mathcal{T}_e} e_0 = \Gamma \tag{21}$$

where E and G are the set of error states and perturbations over N, and $\boldsymbol{\mathcal{S}_e}$ and $\boldsymbol{\mathcal{T}_e}$ are derived from the expansion of the error dynamics over N. Similarly, the inputs are expanded over N as

$$\Delta U = \boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0 \tag{22}$$

where $\boldsymbol{\mathcal{S}_u}$ and $\boldsymbol{\mathcal{T}_u}$ are derived from the expansion of the inputs as a function of G throughout the prediction horizon.

C. Min–Max Problem Formulation

Consider the system in (8) affected by disturbances

$$\begin{aligned} \bar{\gamma}_{k+1} &= \bar{A}\bar{\gamma}_k + \bar{B}\Delta u_k + D w_k \\ \bar{y}_k &= \bar{C}\bar{\gamma}_k \end{aligned} \tag{23}$$

where $w_k \in \mathbb{W}$ is an unknown but bounded external disturbance structured with dimensions identical to Δu. Considering (8) and (21), when these dynamics (including disturbances) are expanded over finite horizon N, we obtain the set of future measurements as

$$Y = \boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}_e} G + \boldsymbol{\mathcal{T}_e} e_0) + \boldsymbol{\mathcal{C}}\boldsymbol{\Omega} W \tag{24}$$

where W is the set of stacked disturbance vectors over the horizon and

$$\boldsymbol{\Omega} = \begin{bmatrix} D & 0 & \dots & 0 \\ \bar{A}D & D & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}^{N-1}D & \bar{A}^{N-2}D & \dots & D \end{bmatrix} \tag{25}$$

We use this extended definition of Y, (21), and (22) to express the objective function (9) as follows:

$$\begin{aligned} J(e_0, G) = {} & G^T [\boldsymbol{\mathcal{S}_e}^T \boldsymbol{\mathcal{C}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{C}} \boldsymbol{\mathcal{S}_e} + \boldsymbol{\mathcal{S}_u}^T \boldsymbol{\mathcal{R}} \boldsymbol{\mathcal{S}_u}] G \\ & + 2G^T [\boldsymbol{\mathcal{S}_e}^T \boldsymbol{\mathcal{C}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{C}} \boldsymbol{\mathcal{T}_e} + \boldsymbol{\mathcal{S}_u}^T \boldsymbol{\mathcal{R}} \boldsymbol{\mathcal{T}_u}] e_0 \\ & + 2G^T [\boldsymbol{\mathcal{S}_e}^T \boldsymbol{\mathcal{C}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{C}} \boldsymbol{\Omega}]\, W \\ & + 2W^T [\boldsymbol{\Omega}^T \boldsymbol{\mathcal{C}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{C}} \boldsymbol{\mathcal{T}_e}]\, e_0 \\ & + W^T [\boldsymbol{\Omega}^T \boldsymbol{\mathcal{C}}^T \boldsymbol{\mathcal{Q}} \boldsymbol{\mathcal{C}} \boldsymbol{\Omega}] W + {e_N}^T \bar{P}_e e_N \end{aligned} \tag{26}$$

subject to

$$\begin{bmatrix} M_\gamma \boldsymbol{\mathcal{B}} \\ M_e \boldsymbol{\mathcal{B}} \\ M_u \end{bmatrix} \boldsymbol{\mathcal{S}_u}\, G \leq \begin{bmatrix} f_\gamma - M_\gamma(\boldsymbol{\mathcal{A}}\gamma_0 + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{T}_u} e_0 + \boldsymbol{\Omega} W) \\ f_e - M_e(\boldsymbol{\mathcal{A}}\gamma_0 + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{T}_u} e_0 + \boldsymbol{\Omega} W) \\ f_u - M_u \boldsymbol{\mathcal{T}_u} e_0 \end{bmatrix} \quad \forall W \in \mathbb{W}^N \tag{27}$$

where $\mathbb{W}^N$ is the set of possible disturbances over N. We define the optimum sequence G^* as that which minimizes (26) given the worst-case disturbances as follows:

$$G^* = \arg \min_G \max_W \quad J(e_0, G). \tag{28}$$

Notice that the constraints are extended to include the bounded uncertainty. Clearly, assuming the worst-case disturbances would yield an overly conservative path. The challenge now is to represent this problem as a single minimization that can be solved efficiently.

D. Reduction to Single Minimization

This section shows how the dual optimization in (28) can be expressed as a single minimization. This requires several

variable substitutions and a number of theorems related to linear inequalities, which are summarized in [25]. Notation has been adapted for the purpose of this paper.

First, we must make an assumption about the structure of the uncertainty. Let us assume that the uncertainty W is represented by a disturbance bounded in a unit box (i.e., $|W| \leq 1$). Then, the following theorem is used to produce the maximization of a linear function $\tilde{\omega}$ subject to these disturbances. A full proof is provided in [25].

Theorem 1 (Maximization of a Disturbed Linear Function): Given the uncertainty set W bounded in the box defined by $|W| \leq 1$ and vector $\tilde{\omega}$ with m number of elements $\tilde{\omega}_i$, then

$$\max_{|W|\leq 1} \tilde{\omega}^T W = \sum_{i=1}^{m} |\tilde{\omega}_i| = |\tilde{\omega}^T| \mathbf{1} \tag{29}$$

where $\mathbf{1}$ is a column vector of ones of appropriate length.

Let us now use the objective function as described in (9) to define a new scalar parameter, t which acts on the total objective cost as follows:

$$(Y - Y_r)^T \boldsymbol{\mathcal{Q}} (Y - Y_r) + \Delta U^T \boldsymbol{\mathcal{R}} \Delta U - t \leq 0. \tag{30}$$

The following theorem shows how a quadratic inequality can be transformed into a linear inequality and vice versa. The result is often referred to as the Schur Complement and is a very useful tool for control systems problems involving linear inequalities. Theoretical proofs and various applications of the Schur Complement for convex optimization are available in [26] and [27].

Theorem 2 (Objective Function as Linear Inequality): The inequality described by (30) is equivalent to the following linear inequality:

$$\begin{bmatrix} t & (Y - Y_r)^T & \Delta U^T \\ (Y - Y_r) & \boldsymbol{\mathcal{Q}}^{-1} & 0 \\ \Delta U & 0 & \boldsymbol{\mathcal{R}}^{-1} \end{bmatrix} \succeq 0. \tag{31}$$

Proof: Since $\boldsymbol{\mathcal{Q}}$ and $\boldsymbol{\mathcal{R}}$ are symmetric, square, and positive definite ($\boldsymbol{\mathcal{Q}} \succ 0$ and $\boldsymbol{\mathcal{R}} \succ 0$) and t is symmetric and square, then it follows from [25, Th. 5.2] that applying the Schur Complement of (30) transforms the quadratic uncertainty into the linear inequality described by (31). ■

By substituting (24) and (22) into (31) and separating the disturbances, we obtain the following rather large inequality:

$$\begin{bmatrix} t & \Delta Y_{cl}^T & (\boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0)^T \\ \Delta Y_{cl} & \boldsymbol{\mathcal{Q}}^{-1} & 0 \\ \boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0 & 0 & \boldsymbol{\mathcal{R}}^{-1} \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{\mathcal{C}\Omega} \\ 0 \end{bmatrix} W \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T + \left(\begin{bmatrix} 0 \\ \boldsymbol{\mathcal{C}\Omega} \\ 0 \end{bmatrix} W \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}^T \right)^T \succeq 0 \tag{32}$$

where $\Delta Y_{cl} = (\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{S}_e} G + \boldsymbol{\mathcal{T}_e} e_0) - Y_r)$.

Next, we develop a linear inequality that ensures the robust satisfaction of constraints while assuming worst case disturbances. First, we restate our problem in terms of four matrices: F, L, H, and Λ.

Recall that we defined the vector W as the set of stacked disturbance vectors over N, or $W = [w_1\, w_2 \ldots w_N]^T$. If the dimension of w_k is r, then the dimension of W is Nr. Let each of these uncertain elements in W be W_i. It follows from (23) that vector ΔU also has dimension Nr. We use W_i to develop the diagonal matrix Λ composed of the uncertain elements as follows:

$$W = \oplus_{i=1}^{Nr} W_i \mathbf{1}^{Nr} = \Lambda^T \mathbf{1}^{Nr} \tag{33}$$

where $\mathbf{1}^{Nr}$ is a column vector of ones. Then, we define the following matrices:

$$F = \begin{bmatrix} t & \Delta Y_{cl}^T & (\boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0)^T \\ \Delta Y_{cl} & \boldsymbol{\mathcal{Q}}^{-1} & 0 \\ \boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0 & 0 & \boldsymbol{\mathcal{R}}^{-1} \end{bmatrix} \tag{34}$$

$$H = \begin{bmatrix} 0 & (\boldsymbol{\mathcal{C}\Omega})^T & 0 \end{bmatrix} \tag{35}$$

$$L = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} (\mathbf{1}^{Nr})^T. \tag{36}$$

These new variables allow us to express (32) as

$$F + L\Lambda H + H^T \Lambda^T L^T \succeq 0. \tag{37}$$

Theorem 3 (Robust Satisfaction of Linear Inequality): The linear inequality (37) holds for all $\Lambda \in \mathbb{R}^{Nr \times Nr}$ if there exists vector $\tau \in \mathbb{R}^{Nr}$ such that $\tau = \oplus_{i=1}^{Nr} \tau_i$ and $\tau_i > 0$, matrix $T \in \mathbb{R}^{Nr \times Nr}$ such that $T = \oplus_{i=1}^{m} \tau_i$ and matrix $S \in \mathbb{R}^{Nr \times Nr}$ such that $S = \oplus_{i=1}^{Nr} \tau_i$ such that the following linear inequality:

$$\begin{bmatrix} F - LSL^T & H^T \\ H & T \end{bmatrix} \succeq 0 \tag{38}$$

is sufficient and necessary for $Nr = 1$ and sufficient for $Nr > 1$.

Proof: Since F, H, and L are real matrices and Λ is defined in (33) using W bounded by a unit box ($|W| \leq 1$), then it follows from [25, Th. 3.4] that linear inequalities (38) is sufficient and necessary for (37) to hold when $Nr = 1$ and sufficient for (37) to hold $Nr > 1$ ■

For simplicity, it can be shown algebraically that [25]

$$LSL^T = \begin{bmatrix} \sum_{j=1}^{N} \tau_j & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{39}$$

Finally, using the definitions of F, H, T, and Theorem 3, we see that (32) can be rewritten as

$$\begin{bmatrix} t - \sum_{j=1}^{N} \tau_j & \Delta Y_{cl}^T & (\boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0)^T & 0 \\ \Delta Y_{cl} & \boldsymbol{\mathcal{Q}}^{-1} & 0 & \boldsymbol{\mathcal{C}\Omega} \\ (\boldsymbol{\mathcal{S}_u} G + \boldsymbol{\mathcal{T}_u} e_0) & 0 & \boldsymbol{\mathcal{R}}^{-1} & 0 \\ 0 & \boldsymbol{\mathcal{C}\Omega}^T & 0 & T \end{bmatrix} \succeq 0. \tag{40}$$

This linear inequality allows us to define a new optimization

$$\min_{G,T,\tau} \quad t \tag{41}$$

subject to constraints (40) and

$$M_\gamma \Gamma \leq f_\gamma \quad \forall W \in \mathbb{W}^N$$
$$M_u \Delta U \leq f_u \quad \forall W \in \mathbb{W}^N. \tag{42}$$

We have now expressed the dual optimization as a single minimization of t constrained by a linear inequalities relating t to the objective function. The next step is to explicitly define our state and input constraints in (42).

E. Robust Constraints

Based on the work in [15], this section uses Theorem 1 to explicitly define the linear inequalities constraints in (42) assuming worst-case disturbances. Let us first assume that we only wish to examine how the state constraints are affected by disturbances.

Given that Γ represents the set of states over the prediction horizon, by substituting (23) into the state constraint portion of (42), we obtain

$$M_\gamma(\boldsymbol{\mathcal{A}}\gamma_0 + \boldsymbol{\mathcal{B}}\Delta U) + M_\gamma(\boldsymbol{\Omega} W) \leq f_\gamma \quad \forall W \in \mathbb{W}^N. \tag{43}$$

Now, we would like to separate the individual rows in $M_\gamma \boldsymbol{\Omega}$ and define the new variable ω_i as the row components

$$M_\gamma \boldsymbol{\Omega} = \begin{bmatrix} \omega_1^T & {\omega_2}^T & \dots & {\omega_q}^T \end{bmatrix}^T \tag{44}$$

where q depends on the structure of the uncertainty. Given this definition of ω_i, it follows that

$$(M_\gamma \boldsymbol{\Omega} W)_i = {\omega_i}^T W. \tag{45}$$

Since the values of W are uncertain, we introduce the vector $\zeta_{\gamma,i}$ to represent the maximum possible solution of (45), or

$$\zeta_{\gamma,i} = \max_{W \in \mathbb{W}^N} {\omega_i}^T W. \tag{46}$$

Therefore, it follows from Theorem 1 that

$$\zeta_{\gamma,i} = |{\omega_i}^T| \mathbf{1}. \tag{47}$$

We define ζ_γ as the column vector composed of elements $\zeta_{\gamma,i}$ for $i = 1, 2, ..., m$

$$\zeta_\gamma = \begin{bmatrix} \zeta_{\gamma,1} & \zeta_{\gamma,2} & \dots & \zeta_{\gamma,m} \end{bmatrix}^T. \tag{48}$$

Similarly, given the obstacle constraint matrix (M_e) from (27), we use the same procedure to derive a separate vector ζ_e for the obstacle constraints

$$\zeta_e = \begin{bmatrix} \zeta_{e,1} & \zeta_{e,2} & \dots & \zeta_{e,m} \end{bmatrix}^T. \tag{49}$$

This allows us to express the full set of state and input constraints as

$$\begin{bmatrix} M_\gamma \boldsymbol{\mathcal{B}} \\ M_e \boldsymbol{\mathcal{B}} \\ M_u \end{bmatrix} \boldsymbol{\mathcal{S}_u}\, G \leq \begin{bmatrix} f_\gamma - M_\gamma(\boldsymbol{\mathcal{A}}\gamma_0 + \boldsymbol{\mathcal{B}}\,\boldsymbol{\mathcal{T}_u} e_0) - \zeta_\gamma \\ f_e - M_e(\boldsymbol{\mathcal{A}}\gamma_0 + \boldsymbol{\mathcal{B}}\,\boldsymbol{\mathcal{T}_u} e_0) - \zeta_e \\ f_u - M_u\, \boldsymbol{\mathcal{T}_u} e_0 \end{bmatrix}. \tag{50}$$

V. Stability Analysis

This section investigates the stability characteristics of the proposed technique.

A. Robust Stability

Here, we present a theorem that guarantees stability of the min–max MPC in the presence of uncertainty. First, let us review the definition of an invariant set. Consider a system with states $\tilde{e}$ governed by linear dynamics $\tilde{\Phi}$

$$\tilde{e}_{k+1} = \tilde{\Phi}\tilde{e}_k. \tag{51}$$

Then, set $\tilde{\mathbb{E}}$ is said to be invariant if

$$\tilde{e}_k \in \tilde{\mathbb{E}} \Rightarrow \tilde{\Phi}\tilde{e}_k \in \tilde{\mathbb{E}}. \tag{52}$$

That is to say that once a state enters $\tilde{\mathbb{E}}$, it can no longer leave if governed by the dynamics in (51). The following theorem is based on the work of [28], in which stability is achieved by steering the vehicle states to an invariant set from which the stabilizing feedback law (19) can take over. Notation has been modified to suit this paper. We introduce a new terminal set $\mathbb{E}_T = \{e_k \in \mathbb{E} \,|\, K_{cl} e_k \in \mathbb{U}, e_{k+1} \in \mathbb{E}_\mathbb{T} \,\forall\, w_k \in \mathbb{W}\}$. Since we want the steady-state error to converge to a region around zero, we also ensure the terminal set contains the origin, $0 \in \mathbb{E}_T$. Recall $e_k = \bar{\gamma}_k - \gamma_r$, where γ_r is the target state.

Theorem 4 (Stability Given Bounded Disturbances): Considering the objective function in (26), linear model (23) with states $\bar{\gamma}_k \in \mathbb{X}$, error terms $e_k \in \mathbb{E}$, and inputs $\Delta u_k \in \mathbb{U}$, subject to disturbances $w_k \in \mathbb{W}$ and constraints (27), stabilizing feedback law (19) and $e_N \in \mathbb{E}_T$, when formulated as an MPC, the system will be asymptotically stable.

Proof: By construction, $\mathbb{E}_T$ is selected such that it contains the origin ($0 \in \mathbb{E}_T$) and satisfies the stated constraints ($\mathbb{E}_T \in \mathbb{E}$), and all inputs resulting from the stabilizing feedback law while inside the terminal set satisfy the input constraints ($K_{cl} e_k \in \mathbb{U} \,\forall\, e_k \in \mathbb{E}_T$). Furthermore, as described in [29], if the stabilizing feedback K_{cl} and the terminal cost $\bar{P}_e$ are chosen using the stabilizing gain and positive-definite solution of the discrete algebraic Riccati equation, then $e_k^T \bar{P}_e e_k$ is a Lyapunov function and the terminal set is positively invariant under (19). Therefore, given these conditions, it follows from the proof in [28, Sec. 4.6] that the system is asymptotically stable $\forall\, w_k \in \mathbb{W}$. ■

B. Extension to Positive-Semidefinite Relaxations

In Section IV, we presented an R-MPC technique for the robust satisfaction of constraints based on semidefinite relaxations. This section extends the stability analysis in Section V-A to include these semidefinite relaxations.

Let us consider the same variables and dynamics from Theorem 4 but now using the objective function in (41) and constraints (40) and (50). We also define a new terminal set $\mathbb{E}_\Pi = \{e_k \in \mathbb{E} \,|\, e_k^T \Pi e_k \leq 1\}$, where $\Pi \succ 0$. The positive-definite matrix Π is chosen such that the new stabilizing feedback K_{cl} is chosen for the predictions. Observe that the choice of Π will define the size of the invariant region and the actual value of the feedback prediction matrix K_{cl}. Furthermore, recall that perturbations G serve as the control variable in the closed-loop paradigm.

As with Theorem 4, we select the stabilizing feedback K_{cl} and the terminal cost $\bar{P}_e$ using the solution of the discrete algebraic Riccati equation. Therefore, it follows from [29] that $e_k^T \bar{P}_e e_k$ is a Lyapunov function and the terminal set is positively invariant under (19).

Theorem 5 (Robust Stability Using Semidefinite Relaxations): Given a semidefinite program with objective function (41) and constraints (40), (50), and linear stabilizing feedback law (19), with the additional constraint that the prediction $\hat{e}_{k+1}$ is invariant, i.e., $\hat{e}_{k+1} \in \mathbb{E}_{\mathbb{P}}$, the semidefinite minimax program guarantees stability if the problem is initially feasible at e_0.

Proof: This theorem follows through induction [25]. Let us start by assuming that the problem is feasible for the problem starting at step (e_{k-1}), i.e., there is a solution that will lead the system to the terminal set. Therefore, by the definition of the terminal set, $e_k \in \mathbb{E}_{\mathbb{P}}$. Furthermore, if $e_k \in \mathbb{E}_{\Pi}$, we know that $G = 0$ is a feasible solution at step k, which gives $e_{k+j+1} = (\bar{A} - \bar{B}K_{cl})e_{k+j} + Dw_k$, where the values of j are dependent on the prediction horizon. Since we have selected K_{cl} and P_e for invariance, we know that these predictions are also contained in $\mathbb{E}_{\mathbb{P}}$. This guarantees feasibility of state and input constraints. By construction, $e_{k+1} \in \mathbb{E}_{\mathbb{P}}$ holds for $\Delta u = -K_{cl}e_k$. Therefore, the problem is feasible at step k. ■

From this result, provided that a terminal set and linear state feedback matrix are selected that satisfy the assumptions listed, we can guarantee robust stability in the presence of bounded disturbances, while also adhering to constraints. Also, the prediction horizon must be sufficiently large. Practically, this translates to the goal being within the range of the vehicle's sensors. The recursive nature of MPC in these types of scenarios allows us respond to unpredictable changes in the environment (i.e., the movement of obstacles). At each timestep, the constraints are redefined and a new optimized control law is produced, which drives the system toward the goal.

As will be shown in some of the scenarios that follow, there may be times when the target is not reachable due to the obstacle constraints. In such cases, the target is not in the set of admissible states and the theorems described above are not satisfied. A common technique for addressing these feasibility issues is to *soften* the constraints by enlarging the terminal set as described in [16]. Since the softening is only applied to the terminal set, this has the effect of driving the vehicle to the location that minimizes the objective function while adhering to constraints. Since the technique is executed as a receding horizon, this location moves closer to the target at each step. This still lends itself to the possibility of falling into a local minimum in certain scenarios, which is the motivation for the hybrid approach described in Section VII, where the MPC is guided by a higher level planner.

VI. Application to UAV Attack Missions

The R-MPC solution was used to guide a simulated military UAV in a number of ground attack missions involving enemy antiaircraft defenses. The UAV dynamics were assumed to be governed by the dynamics described in Section III-B, and the objective function in Section III-C was used to track a stationary target. The problem was formulated as described in Section IV to include a minimization of (41) subject to constraints (40) and (50).

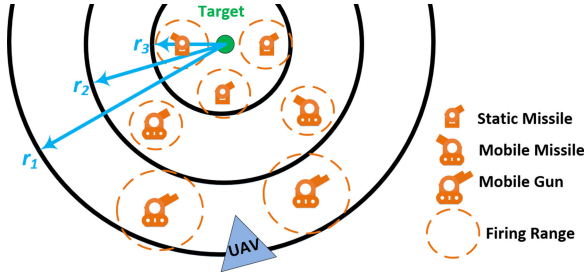

Fig. 3. Battlefield scenario with enemy antiaircraft weapons defending a high-value target.

Based on the battlefield scenarios investigated in [4], Fig. 3 illustrates the ground attack scenario considered in this paper. The mission of the UAV was to pass through all layers of the enemy defenses and fly over the target. When the UAV flew within radius r_1 of the target, the first layer of defense was activated. At this point, the mobile gun platforms operating within this layer of defense could start pursuing the UAV begin firing when in range. Successive layers (r_2 and r_3) were activated as the UAV closed in on the target. Air defense is an active area of research in system engineering [30].

A. Simulation Setup

Positions in the environment are expressed in Cartesian xy coordinates. Seven antiaircraft weapons were placed uniformly between the initial UAV position at $(1000, 0)$ and the target at $(3000, 3000)$, as shown in the tracking results presented in Section VI-C. Each scenario involved three defense layers of radii 3000, 2000, and 700 m centered on the target. The outer defense layer consisted of two mobile gun platforms, each with an effective range of 400 m and a maximum velocity of 20 m/s. Effective range is defined as the range at which the antiaircraft could use its weapons to destroy the UAV. The UAV was considered successful if it flew over the target while staying outside of this range.

The middle defense layer consisted of two mobile missile platforms, each with an effective range of 250 m and a maximum velocity of 20 m/s. The inner defense layer consisted of three static missile platforms, each with a range of 250 m. The UAV was given a maximum velocity of 60 m/s. These parameters were based on the approximate capabilities of shoulder-mounted rocket-propelled grenade launchers, small arms, and the RQ-1 Predator UAV described in [31]. Mobile enemy antiaircraft weapons were given the ability to pursue the UAV at a maximum velocity of 20 m/s when it was inside their respective defense layer.

Using the vehicle model shown in (2), we see that the constraints on velocity cannot be expressed as exact linear inequalities. However, the xy velocity components required in the linearized (3) can be approximated using the vehicle orientation at the time the planning is initialized. When executed as a receding

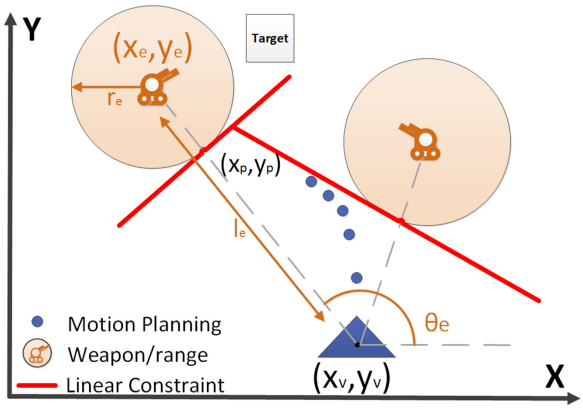

Fig. 4. Illustration of linear inequality constraints for enemy antiaircraft avoidance.

horizon, this has the effect of enforcing the velocity constraints throughout the mission.

All optimization solutions were computed using *CVX: Matlab Software for Disciplined Convex Programming (Version 2.1)* [32] on an Intel Xeon X5670 processor running at 2.93 GHz.

B. Enemy Avoidance via Linear Inequality Constraints

Avoidance of antiaircraft weapons was accomplished using linear inequalities constraints as described in Section IV-A, where the effective firing range of each antiaircraft platform was treated as an obstacle. These constraints are illustrated in Fig. 4.

The mobile platforms were assumed static throughout the prediction horizon. At each timestep, an updated set of linear constraints was developed, which allowed the plane to adapt to the changes in the environment. Also, in order to partially account for the dynamics of the mobile platforms, small buffer zones were created around each obstacle by increasing the radius of the *safe zone*. The values of the buffer zones were based on the distance an enemy platform could travel in the given sample time. For example, given a maximum velocity of 20 m/s and a sample time of 0.5 s, it is feasible that the enemy platform could be 10 m closer than expected after one timestep. Increasing this buffer zone increased the margin of safety, but created a more constrained environment for the UAV. Given a buffer zone of 10 m, the new radius would be computed as $(r'_e = r_e + 10\,\mathrm{m})$.

C. Results

This section presents the simulation results for the UAV attack mission as follows: Section VI-C1 presents the tracking and obstacle avoidance results for one simulation and Section VI-C2 analyzes the relationship between the computation time and prediction horizon size.

1) Tracking With Robust Guarantees: In this simulation, a prediction horizon of 10 and a sample time of 0.2 s were used. The weighting matrices Q and R were given a ratio of 1:10. Random process noise was introduced into the system (as a function of the inputs) with a standard deviation of 1 m/s^2.

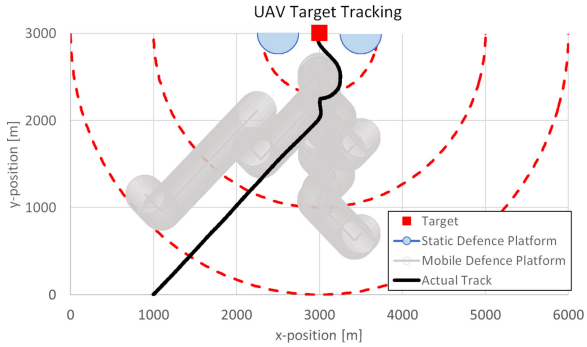

Fig. 5. UAV successfully reaches the target.

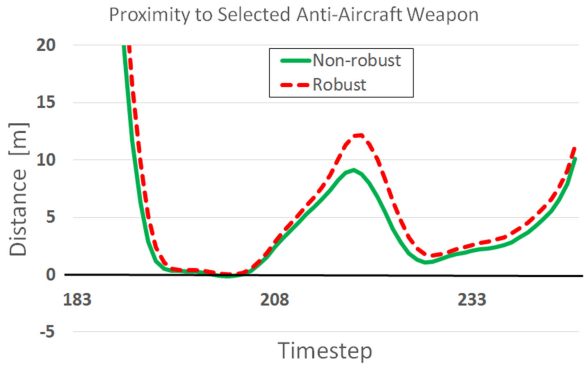

Fig. 6. Clearance from the firing range of selected antiaircraft weapon.

Several figures are presented, which illustrate the behavior of the motion planning algorithm.

In Fig. 5, we see that the simulated UAV successfully navigates to the target at position $(3000, 3000)$.

The benefits of this robust MPC technique are best illustrated when passing near the range of antiaircraft weapons. Fig. 6 illustrates one such situation between timesteps 183 and 233. The results of the R-MPC are compared to a nonrobust MPC under identical conditions. The nonrobust MPC was formulated as described in Section IV-C, except there was no incorporation of bounded disturbances and, hence, no maximization step. Here, we see that, generally, the R-MPC provided more conservative separation from the antiaircraft weapon.

A closer look near timestep 203 in Fig. 7 shows that the nonrobust MPC actually passed within the firing range of the selected antiaircraft weapons. This can be attributed to the process noise introduced into the system. Notice the R-MPC, since formulated to assume the worst-case scenario of noise, provides adequate separation in the presence of identical noise.

2) Computation Time: This section addresses one of the limitations of R-MPC, high computational requirements. Where traditional nonrobust MPC techniques require the optimization of a single variable, the robust design used in this paper requires the optimization of three variables (G, T, and τ). The computation required to handle these additional degrees of freedom grows with the prediction horizon. In order to demonstrate this growth in computation requirements, the times required for over 10 000 individual optimizations were recorded while varying prediction horizon (from $N = 1$ to $N = 20$) and cost

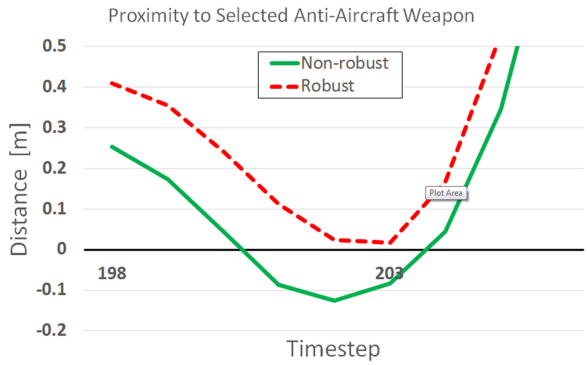

Fig. 7. R-MPC prevents UAV from entering the range of selected enemy defense platform in the presence of uncertainty.

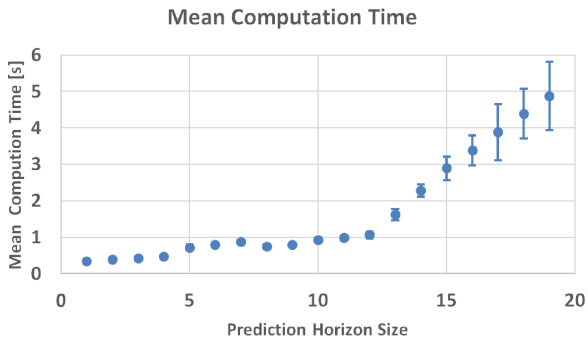

Fig. 8. Exponential growth in computation time at larger prediction horizons.

function (for $Q : R$ ratios of 1:2, 1:3, 1:4, and 1:5). The mean computation time for each prediction horizon size is presented in Fig. 8. The error bars represent three standard deviations.

From this figure, we clearly see the dramatic increase in the computation time at larger prediction horizons. Furthermore, the complexity of solving the optimization grows with the prediction horizon. This is demonstrated by the larger error bars in Fig. 8.

Computation time can be an important factor when considering robust guidance solutions, particularly for real-time applications. This is especially true for MPC-based motion planning, where the greatest benefits are seen when using large prediction horizons. Ultimately, deciding whether this exponential growth in computation time is acceptable depends on real-time constraints and is a potential topic for further research.

VII. Application to the Elderly-Care Scenario

The R-MPC solution was used to guide a simulated assistive elder-care robot scenario. Fig. 9 illustrates a typical home, in which an assistive elderly-care robot might operate [17].

The scenario described in [17] was simulated in MATLAB. The robot dynamics were assumed to be governed by the dynamics described in Section III-B, and the objective function in Section III-C was used to reach waypoints throughout the home. It was assumed that these waypoints were generated by a higher level planner, such as A*. For this application, the R-MPC solution was used to convergence on these waypoints, while guaranteeing constraint satisfaction and obstacle avoidance in the

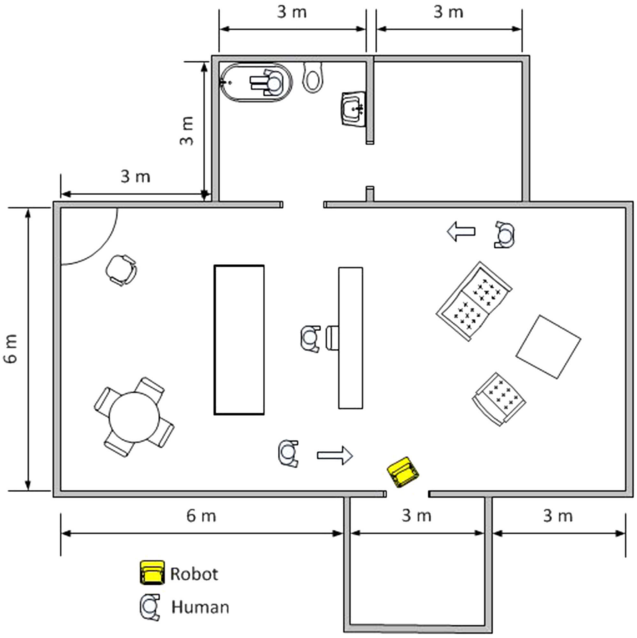

Fig. 9. Illustration of an assistive elderly-care environment.

presence of bounded uncertain disturbances. Typically, an A* path plan would have to be recalculated in its entirety to respond to the changes in the environment. When combined in this hybrid approach, the A* path plan can be used as a fixed global plan, while the R-MPC deals with these changes.

A. Simulation Setup

Five pieces of small furniture were placed in a 12 m by 6 m room and treated as circular objects with varying radii, as shown in the results present in Section VII-C. A large bookshelf and desk were represented by two rectangular obstacles. Three humans were simulated using three circular obstacles, each with a diameter of 0.7 m. One human sat stationary at a desk, while the other two crossed the room at a velocity of 4 km/h. The robot was given a maximum speed of 2 km/h. The positions in the environment are expressed in Cartesian xy coordinates. The robot started at an initial position $(3, -8.15)$ with the successive waypoints placed at $(5, -5.5)$, $(1, -2.5)$, and $(0, 0)$. The robot was required to pass through a narrow doorway to reach the third waypoint. A prediction horizon of 5 and a sample time of 0.5 s were used. The weighting matrices Q and R were given a ratio of 5:1.

B. Rectangular Obstacles

Obstacle avoidance was accomplished using linear inequalities constraints, as described in Section IV-A. One characteristic of this scenario is the presence of large rectangular objects such as desks and shelving. Fig. 10 illustrates how these large rectangular objects can be avoided using linear inequality constraints. At first, it may seem obvious that each edge of the rectangular object could be treated as a linear constraint. However, this would needlessly restrict the vehicle's motion. For example, if

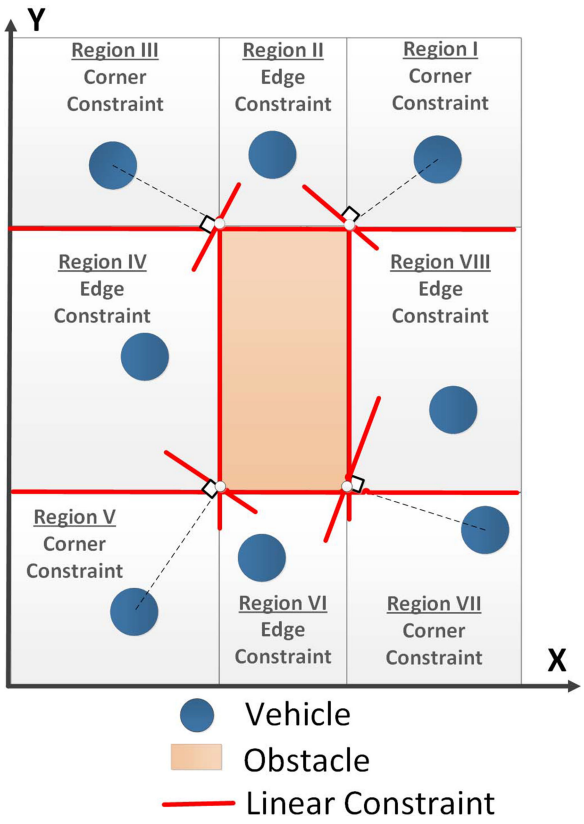

Fig. 10. Illustration of linear inequality constraints for large rectangular obstacles.

the vehicle was located in Region V from Fig. 10, it would not be permitted to leave that region. Also, for larger objects, it may be impractical to encompass the object in a circular *safe zone*. For this simulation, each corner of the rectangle was treated as an obstacle, as described in Section IV-A. Since the robust MPC was implemented using a receding horizon approach, the vehicle was guided around the obstacle to one of the adjacent edge regions. The edge constraints were then only enforced when in Regions II, IV, VI, and VIII, as defined in Fig. 10. The edges of the narrow doorway were also treated in this manner.

C. Results

Fig. 11 shows that the robot successfully converged on the three waypoints, while avoiding all obstacles. We observe that the robot successfully navigates around the corner of the large rectangular object toward to reach the second waypoint and then passes through the doorway to the final waypoint. This demonstrates that the technique described in Section VII-B can be used to avoid rectangular objects. The results also demonstrate that the robot is able to robustly navigate in the dynamic environment, while avoiding moving obstacles.

VIII. Conclusion

This paper presented a general technique for providing optimized robust motion plans for autonomous vehicles using convex R-MPC. A benefit of this technique is that it can be used for

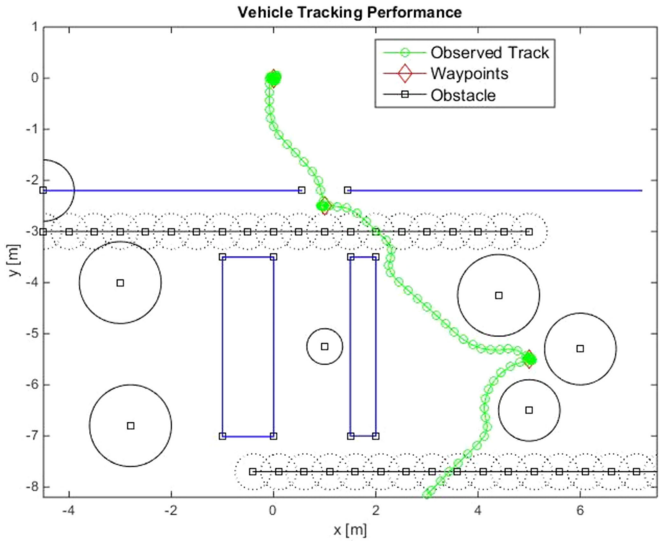

Fig. 11. Tracking for elderly-care simulation.

many different applications without any significant change in formulation. This flexibility was demonstrated through application to two scenarios. In each case, obstacles were represented as especially constructed linear inequality constraints that adapted to changes in the environment. The first scenario involved a military UAV flying over a target, while avoiding ground-based antiaircraft weapons. In this case, the firing range of each antiaircraft weapon was treated as an obstacle. The R-MPC provided safe trajectories for reaching the target by assuming the worst case of bounded uncertain disturbances. The second scenario involved an assistive care robot safely navigating to waypoints throughout a cluttered home. These waypoints were defined by a higher level path planner. The R-MPC successfully guided the robot to each waypoint, while ensuring obstacle avoidance. When combined this hybrid approach, a higher level path plan can be used as a fixed global plan, while the R-MPC deals with changes in the environment.

In order to deal with large rectangular objects in the assistive care robot scenario, a new technique was developed that successfully guided the robot around corners while maintaining a convex optimization. When formulated in terms of perturbations on feedback predictions, the system was proven robustly stable. One limitation to this technique was an exponential growth in mean computation time at increasing prediction horizons. Future work will focus on managing this growth in computation time for use with real-time constraints.

References

[1] F. Belkhouche, "Modeling and calculating the collision risk for air vehicles," *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 2031–2041, Jun. 2013.
[2] H. H. Chiang, Y. L. Chen, B. F. Wu, and T. T. Lee, "Embedded driver-assistance system using multiple sensors for safe overtaking maneuver," *IEEE Syst. J.*, vol. 8, no. 3, pp. 681–698, Sep. 2014.
[3] S. Giese, D. Carr, and J. Chahl, "Implications for unmanned systems research of military UAV mishap statistics," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2013, pp. 1191–1196.
[4] M. Suresh and D. Ghose, "UAV grouping and coordination tactics for ground attack missions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 673–692, Jan. 2012.

[5] H. Choset *et al.*, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT Press, Jun. 2005.
[6] T. Howard, M. Pivtoraiko, R. Knepper, and A. Kelly, "Model-predictive motion planning: Several key developments for autonomous mobile robots," *IEEE Robot. Autom. Mag.*, vol. 21, no. 1, pp. 64–73, Mar. 2014.
[7] J. Zhao and J. Wang, "Integrated model predictive control of hybrid electric vehicle coupled with aftertreatment systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1199–1211, Mar. 2016.
[8] M. Choi and S. Choi, "Model predictive control for vehicle yaw stability with practical concerns," *IEEE Trans. Veh. Technol.*, vol. 63, no. 8, pp. 3539–3548, Oct. 2014.
[9] F. Mohseni, A. Doustmohammadi, and M. B. Menhaj, "Distributed receding horizon coverage control for multiple mobile robots," *IEEE Syst. J.*, vol. 10, no. 1, pp. 198–207, Mar. 2016.
[10] N. Du Toit and J. Burdick, "Robot motion planning in dynamic, uncertain environments," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 101–115, Feb. 2012.
[11] P. Kou, D. Liang, and L. Gao, "Stochastic energy scheduling in microgrids considering the uncertainties in both supply and demand," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2589–2600, Sep. 2018.
[12] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.
[13] A. Bemporad and C. Filippi, "Suboptimal explicit MPC via approximate multiparametric quadratic programming," in *Proc. 40th IEEE Conf. Decision Control*, Dec. 2001, vol. 5, pp. 4851–4856.
[14] M. Mousavi, Z. Heshmati, and B. Moshiri, "LTV-MPC based path planning of an autonomous vehicle via convex optimization," in *Proc. 21st Iranian Conf. Elect. Eng.*, May 2013, pp. 1–7.
[15] J. Löfberg, "Minimax MPC for systems with uncertain gain," in *Proc. 15th IFAC World Congr.*, Jul. 2002, pp. 614–614.
[16] M. N. Zeilinger, M. Morari, and C. N. Jones, "Soft constrained model predictive control with robust stability guarantees," *IEEE Trans. Automat. Control*, vol. 59, no. 5, pp. 1190–1202, May 2014.
[17] M. Merten, A. Bley, C. Schroeter, and H.-M. Gross, "A mobile robot platform for socially assistive home-care applications," in *Proc. 7th German Conf. Robot.*, May 2012, pp. 1–6.
[18] P. T. Jardine and S. Givigi, "A predictive motion planner for guidance of autonomous UAV systems," in *Proc. Annu. IEEE Syst. Conf.*, Apr. 2016, pp. 1–6.
[19] P. T. Jardine, S. Givigi, and A. Noureldin, "Incorporating feedback predictions for optimized UAV attack mission planning," in *Proc. 23th Mediterranean Conf. Control Autom.*, Jun. 2015, pp. 740–746.
[20] P. T. Jardine, S. N. Givigi, and S. Yousefi, "Experimental results for autonomous model-predictive trajectory planning tuned with machine learning," in *Proc. Annu. IEEE Int. Syst. Conf.*, Apr. 2017, pp. 1–7.
[21] P. J. Campo and M. Morari, "Robust model predictive control," in *Proc. Amer. Control Conf.*, Jun. 1987, pp. 1021–1026.
[22] M. V. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361–1379, 1996.
[23] B. Kouvaritakis, J. Rossiter, and A. Chang, "Stable generalised predictive control: An algorithm with guaranteed stability," *IEEE Proc. D—Control Theory Appl.*, vol. 139, no. 4, pp. 349–362, Jul. 1992.
[24] J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach*, 1st ed., R. H. Bishop, Ed. Boca Raton, FL, USA: CRC Press, 2004.
[25] J. Löfberg, "Minimax approaches to robust model predictive control," Ph.D. dissertation, Dept. Elect. Eng., Linköping Univ., Linköping, Sweden, 2003.
[26] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory* (ser. Studies in Applied Mathematics), vol. 15. Philadelphia, PA, USA: SIAM, Jun. 1994.
[27] F. Zhang, *Matrix Theory: Basic Results and Techniques* (ser. Universitext). New York, NY, USA: Springer, 1999. [Online]. Available: http://opac.inria.fr/record=b1099896
[28] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, Jun. 2000.
[29] L. Zhang, J. Lam, and Q. Zhang, "Lyapunov and Riccati equations of discrete-time descriptor systems," *IEEE Trans. Automat. Control*, vol. 44, no. 11, pp. 2134–2139, Nov. 1999.
[30] R. Pryluk, T. Shima, and O. M. Golan, "Shoot-shoot-look for an air defense system," *IEEE Syst. J.*, vol. 10, no. 1, pp. 151–161, Mar. 2016.
[31] J. Pike, Global security database, Aug. 2014. [Online]. Available: http://www.globalsecurity.org/org/index.html
[32] M. Grant and S. Boyd, Cvx: Matlab software for disciplined convex programming, Jun. 2014. [Online]. Available: http://cvxr.com/cvx/

Peter Travis Jardine (S'17) received the B.Eng. degree in mechanical engineering and the M.A.Sc. degree in electrical and computer engineering from the Royal Military College, Kingston, ON, Canada, in 2006 and 2015, respectively. He is currently working toward the Ph.D. degree in electrical and computer engineering with Queen's University, Kingston.

His research focuses on autonomous robotics, where he specializes in the application of learning techniques for predictive control of autonomous aerial vehicles.

Sidney N. Givigi (SM'14) received the Ph.D. degree in electrical and computer engineering from Carleton University, Ottawa, ON, Canada, in 2009.

In 2009, he joined the Department of Electrical and Computer Engineering, Royal Military College of Canada, Kingston, ON, where he is currently an Associate Professor. His research interests include autonomous systems, especially the decentralized control of multiple vehicles, learning and adaptation of autonomous robots, and modeling of complex systems with game theory.