

Article

Event-Triggered Intervention Framework for UAV-UGV Coordination Systems

Wu Wang ¹, Junyou Guo ^{1,2} , Guoqing Tian ^{1,2} , Yutao Chen ^{1,2}  and Jie Huang ^{1,2,*} 

¹ College of Electrical Engineering and Automation, Fuzhou University, Fuzhou 350108, China; wangwu@fzu.edu.cn (W.W.); N190127103@fzu.edu.cn (J.G.); N190127122@fzu.edu.cn (G.T.); yutao.chen@fzu.edu.cn (Y.C.)

² Key Laboratory of Industrial Automation Control Technology and Information Processing, Education Department of Fujian Province, Fuzhou 350108, China

* Correspondence: jie.huang@fzu.edu.cn

Abstract: Air-ground coordination systems are usually composed of unmanned aerial vehicles (UAV) and unmanned ground vehicles (UGV). In such a system, UAVs can utilize their much more perceptive information to plan the path for UGVs. However, the correctness and accuracy of the planned route are often not guaranteed, and the communication and computation burdens increase with more sophisticated algorithms. This paper proposes a new type of air-ground coordination framework to enable UAVs intervention into UGVs tasks. An event-triggered mechanism in the null space behavior control (NSBC) framework is proposed to decide if an intervention is necessary and the timing of the intervention. Then, the problem of whether to accept the intervention is formulated as an integer programming problem and is solved using model predictive control (MPC). Simulation results show that the UAV can intervene in UGVs accurately and on time, and the UGVs can effectively decide whether to accept the intervention to get rid of troubles, thereby improving the intelligence of the air-ground coordination system.

Keywords: air-ground coordination system; event-triggered; null space-based behavior control; model predictive control



Citation: Wang, W.; Guo, J.; Tian, G.; Chen, Y.; Huang, J. Event-Triggered Intervention Framework for UAV-UGV Coordination Systems. *Machines* **2021**, *9*, 371. <https://doi.org/10.3390/machines9120371>

Academic Editors: Pascual Campoy, Dariusz Horla, Wojciech Giernacki and Dan Zhang

Received: 31 October 2021
Accepted: 15 December 2021
Published: 20 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of science and technology, robot, as an advanced tool that integrates a number of advanced technologies, are having more and more impacts on human society. For complex and dynamic tasks and environments, a multi-robot system has the advantages of lower operating costs, fewer system requirements, stronger adaptability, and flexible scalability, when compared to a single robot [1].

Among all kinds of robots, unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) have been widely used in civilian and military fields. UAVs can provide a global and accurate view of the environment by making use of their fast moving speed and their being less prone to GPS signal interruption. UGVs have high load capacity and can endure long-term task requirements. Air-ground coordination systems, formed by combining the functional characteristics of UAVs and UGVs, can not only effectively avoid weaknesses such as the short flight time of UAVs and poor perception of UGVs, but also provide breakthrough ideas for multi-robot coordination systems with huge heterogeneity and functional complementarity [2,3]. Air-ground coordination systems have been widely used in many scenarios, such as agriculture [4], rescue [5], exploration [6], and surveillance [7].

UAV-UGV coordination systems can be classified into eight different settings, from the perspective of the functional roles that UGVs and UAVs have in a system [8]. The functional roles of UAVs and UGVs can be divided into: sensors, actuators, decision makers, and auxiliary facilities. UAV-UGV coordination systems can be expressed as $\langle X | Y \rangle$, where

X is the functional role of UAVs and Y is the functional role of UGVs. Most air-ground coordination systems can be classified as different types using different combinations of UAV and UGV roles [9–12]. One typical class is written as $\langle S, D | A \rangle$, where UAVs act as sensors and decision makers, and UGVs act as actuators. This class of systems is of particular interest because there are tasks that UGVs cannot complete by their own intelligence. For example, a UGV may fall into a local minimum when avoiding multiple obstacles (e.g., stops in the middle between two obstacles). This necessitates UAV intervention for UGV on-line in the process of task execution. In this case, UAVs perceive the environment from the air and make decisions for UGVs. This type of system traditionally collects information through the manual control of UAVs, which help path planning for UGVs offline [13]. Such a process has difficulty in coping with dynamic environments. Many researchers propose to achieve path planning and control on-line [14,15]. However, this is limited in cases when UGVs do not need UAVs to make perception decisions when performing their basic tasks. In addition, on-line decision making and path planning bring communication and computation burdens. Therefore, key problems arise as how to build a decision-making model for UAV intervention and how to solve the problem of task conflicts between external interventions and the UGVs own tasks.

To solve the potential decision-making/planning/control conflicts, the null space behavior control (NSBC) method is one of the potential solutions, where different basic tasks for UAVs and UGVs are defined as behaviors with assigned priority. This method ensures that, under the premise of the complete execution of high-priority tasks, the partial execution of secondary tasks is possible [16]. An air-ground cooperative formation method based on the NSBC has been proposed to ensure that the formation shape can be maintained when UGVs or UAVs encountered obstacles [17–19]. A new type of human decision-making behavior model has been proposed in the framework of the NSBC using an event-triggered mechanism [20]. The human intervention task, with the highest priority, is triggered only when defined decision variables reach a threshold. It reduces the burden of communication and computation by intervening robots in an event-triggered way and enables human assistance to complete tasks beyond robots' own capabilities. However, the existing methods cannot avoid risks brought on by bad or malignant interventions from either humans or UAVs.

Motivated by the above discussions, in this paper, we focus on how UAVs intervene with UGVs to improve the overall intelligence of the system, while reducing the burden of communication and computation. The contributions of this paper can be summarized as follows. First, a new type of air-ground coordination system is proposed, which can be written as $\langle S, D, A | S, D, A \rangle$, where both UAVs and UGVs have the ability to perceive, make decisions, and execute tasks. It extends the unidirectional intervention that existing works consider to bidirectional interventions. Second, the drift diffusion model (DDM) and model predictive control (MPC) are introduced into the NSBC framework to accurately determine the timing of intervention, to achieve an optimal trade-off between decision speed and accuracy, and to predict whether an intervention is correct and acceptable. The latter decision-making acceptance problem is formulated into an integer programming problem and solved using current state and future predictions.

The rest of this paper is organized as follows. In Section 2, we briefly introduce preliminaries of the NSBC; In Section 3, the UAV and UGV task design and decision making model based on event-triggered way is presented; In Section 4, the intervention task decision maker based on MPC is presented; In Section 5, simulation cases are studied and discussed; Section 6 concludes the paper.

2. Preliminary

Let's briefly review the NSBC methods [21], by defining as $\sigma \in \mathbb{R}^m$ mathematical expressions for the behavior to be achieved (usually called tasks). $\rho \in \mathbb{R}^n$ is defined as a variable vector describing the system configuration. In general, they are related through the following models:

$$\sigma = f(\rho) \quad (1)$$

with the corresponding differential relationship:

$$\dot{\sigma} = \frac{\partial f(\rho)}{\partial \rho} \times v = J(\rho) \times v \quad (2)$$

where $J(\rho) \in \mathbb{R}^{m \times n}$ is the configuration-dependent task Jacobian matrix and $v \in \mathbb{R}^n$ is the system velocity. The reference velocity v_d is to act at the differential level by inverting the (locally linear) mapping and pursueminimum-norm velocity, leading to the least-squares solution:

$$v_d = J^\dagger \dot{\rho}_d = J^T (JJ^T)^{-1} \dot{\rho}_d \quad (3)$$

where ρ_d is the reference trajectory, J^\dagger is the pseudo-inverse of the Jacobian matrix. Since the discrete-time integration of the reference velocity will cause the numerical drift of the reconstructed position, the following closed loop inverse kinematics (CLIK) form is used to compensate for the drift:

$$v_d = J^\dagger (\dot{\rho}_d + \Lambda \tilde{\rho}_d) \quad (4)$$

where $\dot{\rho}_d$ is the derivative of the desired task function. Λ is a suitable constant positive definite matrix of gains. $\tilde{\rho}_d = \rho_d - \rho$ is the task error.

Consider there are N tasks, and each task is assigned a priority (expressed by subscript i , $i = 1$ means the highest priority). The NSB solution to the task combination can be formulated in an iterative way, defining the velocity vector as follows:

$$v^i = v_{d,i} + N_i \times v^{i+1}, i = 1, 2, \dots, N \quad (5)$$

where $v^{(n+1)} = 0$, $v^1 = v_d$, and the matrix $N_i = (I - J_i^\dagger J_i)$ is the null space projector matrix of Jacobian. In sum, corresponding to a lower priority task, are projected onto the null space of the immediately higher priority task; then, eventually, conflicting velocity components are cut off, before being added to higher priority task velocity components. The geometric model of its comprehensive velocity output is shown in Figure 1.

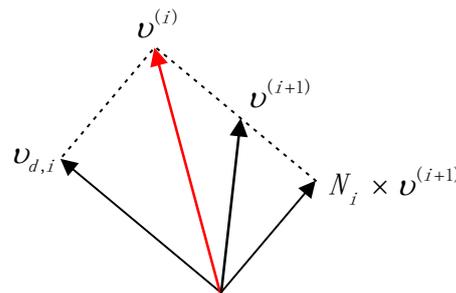


Figure 1. Task velocity composition in the NSBC framework. The velocity $v^{(i+1)}$ is projected into the null space of the higher priority task and added to $v_{d,i}$. Finally, $v_{d,i}$ and $N_i \times v^{(i+1)}$ get the composite task speed v^i through the vector sum.

3. Framework Design of Air-Ground Coordination Systems

This paper proposes an air-ground coordination system framework for UAVs to intervene UGVs through an event-triggered mechanism. As shown in Figure 2, The DDM is employed for simple decision-making modeling by accumulating decision variables. Bayes' risk criterion is used to achieve the optimal trade-off between decision speed and accuracy, which is in charge of setting the decision threshold. The combination of the two achieves an accurate judgment on the timing of intervention [22]. When the decision variable does not reach the threshold, intervention is triggered, hence reducing the communication resources needed. To solve task conflicts, DDM is embedded into the NSBC framework for UAVs

to determine whether and when to intervene in UGVs. When the decision variable has not reached the threshold, the UAVs and UGVs perform their own basic tasks. When the decision variable reaches the decision threshold, UGVs can no longer rely on their own intelligence, and the UAVs need to intervene to help make decisions.

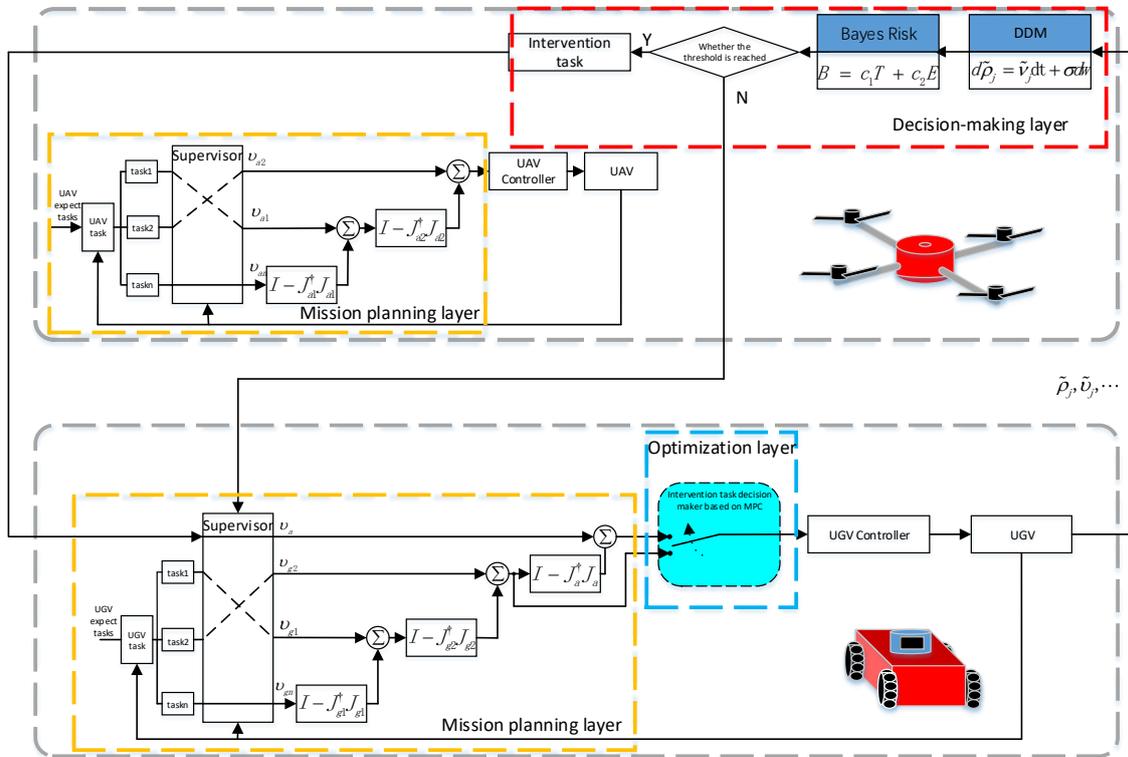


Figure 2. Framework design of air-ground cooperative system. It is mainly composed of three layers. (1) Task planning layer: is responsible for task design based on NSBC and resolution of task conflicts (2) Decision-making layer: is mainly responsible for judging the timing of intervention. (3) Optimization layer: optimizes the decision-acceptance problem for UGVs.

3.1. Task Planning Layer Task Design

3.1.1. UGV Basic Task Function Design

UGV Move-to-Target Task Function Design

The movement of the UGV to the target point is defined as the task of moving to the target point. Once the target point is reached, the task is completed and the UGV stops moving. Define the location of the target point $\rho_g = [x_g \ y_g \ z_g]^T$, location of UAV. ρ as a controllable task variable $\sigma_m = \rho$. Define the position to reach the target point as the desired function $\sigma_{md} = \rho_g$. Then, the output of the UGV motion task is:

$$v_{mj} = J_{mj}^+(\dot{\sigma}_{mj} + B\tilde{\sigma}_{mj}) \tag{6}$$

where B is defined as the UGV motion task gain. J_{mj}^+ is the Jacobian pseudo-inverse matrix for the motion task. $\tilde{\sigma}_{mj} = \sigma_{md} - \sigma_m$ is the task errors.

UGV Obstacles-Avoidance Task Function Design

The UGV needs to avoid obstacles detected by the sensor when moving along the reference trajectory to the target point. In order to ensure the safety of the UGV during its movement, obstacle avoidance tasks can provide basic guarantees. Define the location of UGV is $\rho = [x \ y \ z]^T$. Obstacle location is $\rho_o = [x_o \ y_o \ z_o]^T$. The obstacle avoidance task function is:

$$\sigma_a = D = \sqrt{(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2} \quad (7)$$

Define the desired function of obstacle avoidance task as $\sigma_{ad} = d$, which d is the obstacle avoidance safety distance. Then, the output of the obstacle avoidance task function is:

$$v_a = J_{aj}^\dagger(\dot{\sigma}_{ad} + A\tilde{\sigma}_{aj}) \quad (8)$$

where A is defined as the UGV obstacles-avoidance task gain. J_{aj}^\dagger is the Jacobian pseudo-inverse matrix for the obstacles-avoidance task. $\tilde{\sigma}_{aj} = \sigma_{ad} - \sigma_a$ is the obstacles-avoidance task errors.

3.1.2. UAV Basic Task Function Design

UAV Formation Task Function Design

UAVs and UGVs maintain a formation in order to effectively sense the movement of UGV and prepare for subsequent UAV landings. Define the task variable that the UAV keeps at the center of the UGVs as

$$\sigma_c = \frac{1}{n}(\rho_1 + \rho_2 + \dots + \rho_n) \quad (9)$$

where $\rho_i, i = 1, 2, \dots, n$, represents the location of UGVs. Define the desired formation task function as σ_{cd} . UAV formation task output functions as follows.

$$v_c = J_{cj}^\dagger(\dot{\sigma}_{cd} + C\tilde{\sigma}_{cj}) \quad (10)$$

where C is defined as the UAV formation task gain. J_{cj}^\dagger is the Jacobian pseudo-inverse matrix for the formation task. $\tilde{\sigma}_{cj} = \sigma_{cd} - \sigma_c$ is the formation task errors.

UAV Obstacles-Avoidance Task Function Design

Although the working environment of the UGV is relatively simple for the ground environment, it is necessary to design the task of avoiding obstacles. The task design process is the same as the obstacle avoidance task design of the UGV. Therefore, the output of the UAV obstacle avoidance task is

$$v_{au} = J_{auj}^\dagger(\dot{\sigma}_{aud} + A\tilde{\sigma}_{auj}) \quad (11)$$

where A is defined as the UAV obstacles-avoidance task gain. J_{auj}^\dagger is the Jacobian pseudo-inverse matrix for the UAV obstacles-avoidance task. σ_{aud} is the UAV desired obstacle-avoidance task function, and σ_{au} is the UAV obstacles-avoidance task function. $\tilde{\sigma}_{auj} = \sigma_{aud} - \sigma_{au}$ is the UAV obstacles-avoidance task errors.

3.1.3. Composite Task Function Design

Composite task refers to the combination of multiple basic tasks according to task priority. Define $\sigma_b \in \mathbb{R}^{m_b}$ as the b_{th} task function, $\forall 1 \leq b \leq r$, where m_b is the dimension of the task space. Moreover, we further define the task hierarchy which follows the rules [23]:

- (1) Assume that $b = r$ is the lowest priority, and $b = 1$ is the top priority. Here, $m_b > m_a$ implies that m_b is the index of a lower priority than m_a ; a task of priority m_b may not disturb another task of priority m_a . The lower-priority tasks are executed in the null space of all higher priority tasks.
- (2) The mappings from the velocities to the task velocities are captured by the task Jacobian matrix $J_b \in \mathbb{R}^{m_b \times n}$, $\forall 1 \leq b \leq r$.
- (3) The dimension m_r of the lowest level task may be greater than $n - \sum_{b=1}^{r-1} m_b$ so that n is ensured to be greater than the total dimension of all tasks.

Following the aforementioned rules, the composite task velocity of UAV or UGV can be obtained by Equation (5).

3.2. UAV Intervention Task Design

In this paper, only the supervision behavior and intervention behavior of UAV to UGV are considered. The supervision behavior is defined by UAV monitoring the UGV task execution process without intervention, until failure is detected. The intervention behavior is defined by UAV intervening UGV. For the UAV supervision behavior, there is no task input to the UGV. Instead, for the UAV intervention behavior, task input to the UGV should be provided. Thus, the task corresponding to the UAV intervention behavior should be designed. The intervention task will be set as the highest priority task execution of the UGV, and the original task of the UGV will be projected onto the null space of the intervention task of the UAV. Define the desired UAV intervention task function as:

$$\sigma_{int} = f(\rho_{int}) \quad (12)$$

where ρ_{int} is the real-time position of the UGV that the UAV would intervene. The derivative of the UAV intervention task is given by

$$\dot{\sigma}_{int} = J_{int}v_{int} \quad (13)$$

where J_{int} is the intervention Jacobian matrix. Therefore, the output of the UAV intervention task is

$$v_{int} = J_{int}^{\dagger}(\dot{\sigma}_{int} + \Lambda_{int}\tilde{\sigma}_{it}) \quad (14)$$

where Λ_{int} is defined as the UAV intervention task gain. J_{int}^{\dagger} is the Jacobian pseudo-inverse matrix for the UAV intervention task. σ_{in} is the UAV intervention task function. $\tilde{\sigma}_{it} = \sigma_{int} - \sigma_{in}$ is the UAV intervention task error.

Assumption 1. UAVs can give intervention tasks v_{int} .

Remark 1. Intervention tasks can be given by fuzzy logic [24], reinforcement learning [25], neural network [26] method.

3.3. Decision-Making Layer Design

The decision-making layer consists of DDM and Bayes Risk criteria. The DDM is a cognitive process modeling method in the "two-choice forced decision-making problem", which is suitable for simple decision-making process modeling. This model accumulates decision information under external noise. When the accumulated decision information reaches any decision threshold, the choice corresponding to the decision threshold is selected as the final decision result. In this paper, DDM is used as the event-triggered way for UAV intervene in UGV, embedded in the NSBC framework. The formula for the DDM of UAV is as follows.

$$d\tilde{\rho}_j = \tilde{v}_j dt + \sigma_j dw_j(t) \quad (15)$$

The decision variables should be selected to reflect the progress of the robot to complete the task. In this paper, the task error of the UGV of the NSBC method is selected as the decision variable of the UAV. The task error is $\tilde{\rho}_j = \rho_{rdj} - \rho_j$, where ρ_{rdj} is the preset trajectory and ρ_j is the trajectory planned by NSBC. $\tilde{v}_j = v_{rdj} - v_j$ is the drift rate, which characterizes the amount of change of the decision variable per unit time. $\sigma_j dw_j(t)$ is white noise conforming to Gaussian distribution, which represents the influence of noise during the accumulation process of decision information.

The generation of UAV decision threshold requires the introduction of Bayes' risk criterion function. This function can minimize the decision risk and realize the trade-off

between decision speed and accuracy. The Bayes' risk criterion function is the weighted sum of decision time (T) and decision deviation (E). The formula is as follows:

$$B = c_1T + c_2E \tag{16}$$

where c_1 and c_2 are the correlation coefficients of decision time and decision deviation, respectively. The formulae for decision time T and decision deviation E are as follows.

$$T = \frac{\zeta_j}{\tilde{v}_j} \tanh\left(\frac{\zeta_j \tilde{v}_j}{\sigma^2}\right) + \left(\frac{2\zeta_j(1 - e^{-\frac{2\tilde{\rho}_{j,0}\tilde{v}_j^2}{\sigma^2}})}{\tilde{v}_j(e^{\frac{2\zeta_j\tilde{v}_j}{\sigma^2}} - e^{-\frac{2\zeta_j\tilde{v}_j}{\sigma^2}})} - \tilde{\rho}_{j,0}\right) \tag{17}$$

$$E = \frac{1}{1 + e^{\frac{2\zeta_j\tilde{v}_j}{\sigma^2}}} - \frac{(1 - e^{-\frac{2\tilde{\rho}_{j,0}\tilde{v}_j^2}{\sigma^2}})}{(e^{\frac{2\zeta_j\tilde{v}_j}{\sigma^2}} - e^{-\frac{2\zeta_j\tilde{v}_j}{\sigma^2}})} \tag{18}$$

where $\tilde{\rho}_{j,0}$ is the decision deviation at the initial moment. Since the decision deviation E decreases exponentially with the increase of the threshold ζ_j , and the decision time T increases with the increase of ζ_j , the Bayes' risk criterion function has a minimum value. By minimizing this function and solving for $\frac{\partial B}{\partial \zeta} = 0$, the decision threshold ζ_j is obtained.

$$\frac{\sigma^2(e^{\frac{2\zeta_j\tilde{v}_j}{\sigma^2}} - e^{-\frac{2\zeta_j\tilde{v}_j}{\sigma^2}})}{2\tilde{v}_j^2} + \frac{2\zeta_j}{\tilde{v}_j} = \frac{c_1}{c_2} \tag{19}$$

At this time, when the decision variable has not reached the decision threshold, the UGV maintains its original task output. When the decision variable reaches the decision threshold, it indicates that the UGV is currently unable to complete the task with its own intelligence, and the UAV needs to give an intervention task to intervene the UGV to get out of the predicament. The original task of the UGV is projected onto the null space of the intervention task of the UAV, ensuring that once the intervention task is given, it will be executed as the highest priority task, and the original task of the UGV will be partially executed. At this time, the output formula of the UGV task is as follows:

$$v_{ugv} = \begin{cases} v_{ugv}, \tilde{\rho}_j < \zeta_j \\ v_{int} + (I - J_{int}^+ J_{int})v_{ugv}, \tilde{\rho}_j \geq \zeta_j \end{cases} \tag{20}$$

4. Optimization Layer Design Based on MPC

4.1. Optimal Control Formulation

Taking into account that the UAV may be affected by visual occlusion, inaccurate sensor measurement, and disturbance of the fuselage itself during the execution of the task, an undesirable intervention task is given. As a result, UGV performs unnecessary intervention tasks, and even worse results. Therefore, it is necessary to introduce the intervention task decision maker based on MPC to determine whether the UGV accepts the intervention task. This boils down to the 01 integer programming problem [27].

This paper proposes an intervention task decision maker based on MPC. By establishing the optimization problem and adding related constraints to solve it, the overall task performance including UGV intervention task can be optimized.

Consider a group of n UGVs, Define the state of the UGV as:

$$\rho(t) = \rho_i(t), i = 1, 2, \dots, n \tag{21}$$

where each element represents the state of each UGV. Define a set of binary vectors:

$$\begin{cases} \omega_i^j(t) = [\omega_{i,1}^j(t)^T, \omega_{i,2}^j(t)^T] \in \{0, 1\} \\ i = 1, \dots, n \\ j = 1, 2 \end{cases} \quad (22)$$

where j Represents two possible choices of whether to accept the intervention task. The kinematics of each UGV can be written as a convex combination of kinematics in different modes:

$$\dot{\rho}_i(t) = \omega_{i,1}^j(t)[v_{int}(t) + (I - J_{int}^\dagger J_{int})v_{ugv}(t)] + \omega_{i,2}^j(t)v_{ugv}(t), i = 1, \dots, n \quad (23)$$

where $v_{int}(t)$ is the intervention task issued by the UAV at time t , and $v_{ugv}(t)$ is the original composite task of the UGV at time t . One of the important constraints on $\omega_i(t)$ is:

$$\begin{cases} \omega_{i,1}^j + \omega_{i,2}^j = 1 \\ \omega_{i,1}^j, \omega_{i,2}^j \in \{0, 1\} \end{cases} \quad (24)$$

Through this constraint, it is ensured that the UGV will either accept the intervention task of the UAV and be executed as the highest priority task, and the original composite task of the UGV will be projected to the null space of the intervention task of the UAV, or it does not accept the intervention task of the UAV and maintains the original composite task output. Define a cost function

$$L(x) = \int_0^T \sum_{i=1}^n (a_i \|\rho_{rdj}^i(t) - \rho_i(t)\|^2 + b_i \|u_i\|^2) dt, i = 1, \dots, n, \quad (25)$$

which consists of combined task errors of the NSBC from all UGVs plus slack variables, each weighted by positive parameters a_i and b_i . As a result, an OCP can be formulated as

$$\min_{\rho_i, w, u} L(\rho_i) \quad (26a)$$

$$s.t. \rho_i(0) = \hat{\rho}_{i0} \quad (26b)$$

$$\begin{cases} \omega_{i,1}^j + \omega_{i,2}^j = 1 \\ \omega_{i,1}^j, \omega_{i,2}^j \in \{0, 1\} \end{cases} \quad (26c)$$

$$\begin{cases} \|\rho_i(t) - \rho_o^i(t)\| \geq d_i - u_i(t) \\ i = 1, \dots, n \end{cases} \quad (26d)$$

$$u_i(t) \geq 0, \forall t \in [0, T] \quad (26e)$$

where $\hat{\rho}_{i0}$ is the initial state of the i -th UGV. $\rho_o^i(t)$ is the Obstacle position detected by the i -th UGV at time t . d_i is the corresponding safety distance. $u_i(t)$ is the slack vectors and T is the prediction horizon. slack variable u_i as a soft constraint for the optimization problem. The purpose of adding u_i to constraints is to effectively obtain optimized solutions.

4.2. Real-Time Model Predictive Control Algorithm

During the task execution process, problem (26) is difficult to solve because it contains integer variables. In this paper, we employ a real-time MPC algorithm with “first discretize then optimize” methodology and an outer-convexification with integer relaxation technique [28]. First, we relax the binary variable ω_i to real number in the range of 0 to 1, that is $\hat{\omega}(t) \in [0, 1]$. Then, through the multiple shooting method [29], problem (26) can be converted to a nonlinear programming problem:

$$\min_{\rho_i, \hat{\omega}, u} \sum_{k=0}^N L(x_{s|k}, u_{s|k}) \quad (27a)$$

$$s.t. x_{s|0} = \hat{x}_0 \tag{27b}$$

$$\begin{cases} \omega_{i,1,s|k}^j + \omega_{i,2,s|k}^j = 1 \\ \omega_{i,1,s|k}^j, \omega_{i,2,s|k}^j \in \{0, 1\} \end{cases} \tag{27c}$$

$$\|\rho_{i,s|k}(t) - \rho_o^i(t)\| \geq d_i - u_{i,s|k}(t) \tag{27d}$$

$$u_{i,s|k}(t) \geq 0, \tag{27e}$$

$$i = 1, 2, \dots, n \tag{27f}$$

$$k = 0, 1, \dots, N \tag{27g}$$

where N represents that the prediction time domain T is divided into N parts, that is each time interval is $\Delta t = \frac{T}{N}$. $s | k$ is the k th-step predicted at sampling time s . The NLP can be solved efficiently by standard NLP solvers using sequential quadratic programming or interior point methods. Finally, a sum-up-rounding (SUR) step can be employed to obtain the binary variable $\omega_i(t)$ from $\hat{\omega}(t)$ [28]. The SUR step reads as

$$h_{s|k}^j = \sum_{r=0}^k \hat{\omega}_{s|r}^j \Delta t - \sum_{r=0}^{k-1} \omega_{s|r}^j \Delta t \tag{28}$$

$$\omega_{s|k}^j = \begin{cases} 1 & \text{if } h_{s|k}^j \geq h_{s|k}^r \quad \forall r \neq j \& j < r \forall r : h_{s|k}^j = h_{s|k}^r \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

5. Simulation

In this part, consider an UAV and two UGVs moving in the x-y-z three dimensional space, where each robot is modeled as a first-order system. The goal of the entire air-ground coordination system is to move to the target point and avoid obstacles in an unknown environment. The UAV is equipped with cameras to sense UGVs ground environment; UGVs are equipped with sensors to sense surrounding obstacles. The values of parameters used in this simulation are shown in Table 1.

Table 1. Parameter values used in the simulation.

Parameter	Value
initial position	$[1 \ 0.5 \ 0]^T$
obstacles position	$[-1 \ 0.5 \ 0]^T$, $[0.2 \ 1 \ 2]^T$, $[1.7 \ 4 \ 0]^T$, $[4 \ 7 \ 0]^T$, $[-0.5 \ 5 \ 0]^T$
target position	$[3.3 \ 13 \ 0]^T$, $[-1 \ 13 \ 0]^T$, $[1.1 \ 13 \ 2]^T$
UGV1 preset trajectories	$[3.3 \ 1t \ 0]^T$
UGV2 preset trajectories	$[1.3 \ 1t \ 0]^T$
safe distance	2 m
task gain A,B,C	3,2,5,1.5
DDM parameter $c_1, c_2, \sigma_j, \zeta_j$	0.5, 1, 10, 3
MPC sampling frequency	20 Hz
MPC prediction horizon	3 s
MPC grid point number	60

First, this paper proposes three verification methods for the air-ground collaborative system framework. (a) The UGV encountered a local minima point, and the UAV did not take intervention measures. (b) The UGV encounters a local minima point, and the UAV takes effective intervention to prompt the UGV to get rid of the extreme point. (c) The UGV encounters a local minima point, the UAV gives a bad intervention, and the UGV chooses not to accept the intervention task.

Next, two case studies are given to better demonstrate the advantages of the proposed UAV-UGV coordination system’s event-triggered intervention framework.

5.1. Case A

In this case, the effective intervention of UAV is verified by comparing method (a) and method (b). In the process of UGV performing tasks, there may be situations in which they cannot rely on their own intelligence to solve problems. For example, when a UGV faces two obstacles and performs an obstacle avoidance task, the sum of the speed vectors of the two obstacle avoidance tasks will have the same magnitude as the speed vector moving to the target point, but the direction will be opposite, which makes it stuck in the local minima point. Trajectory diagram of the air-ground system with no intervention measures taken by UAV and effective intervention measures taken by UAV shown in Figure 3.

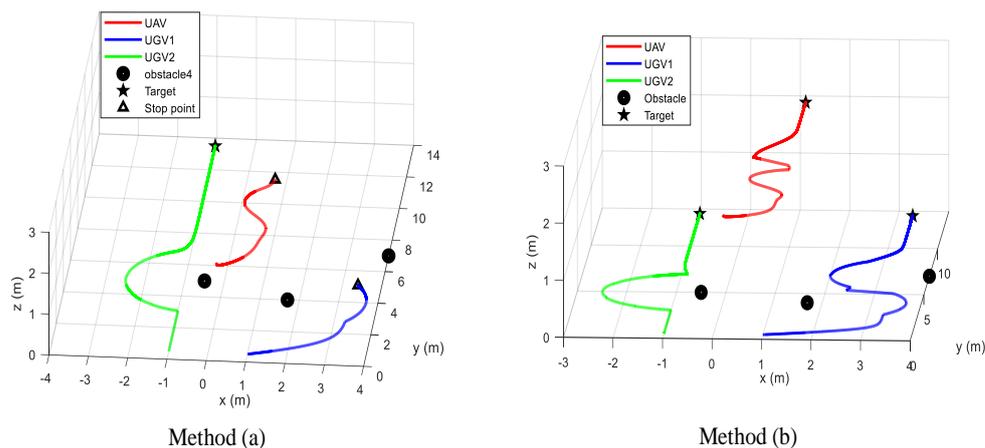


Figure 3. The figure show that the trajectory of method (a) and method (b).

In method (a), the UGV2 can complete the task by virtue of its own intelligence. Since the UAV did not effectively intervene in the UGV1, the UGV1 fell into a minimum value at (3.3 m, 5.1 m). The UAV’s decision variable for UGV1 keeps increasing until it exceeds the decision threshold, but the intervention task is not triggered. Through the effective intervention of UAV in method (b) at time 14.45 s, after the decision variable of UGV1 exceeds the threshold, it will fall back below the threshold after 1.7 s. It shows that, after the effective intervention of the UAV, the UGV1 has been able to complete the task with its own intelligence. The decision variables of the UAV are shown in Figure 4.

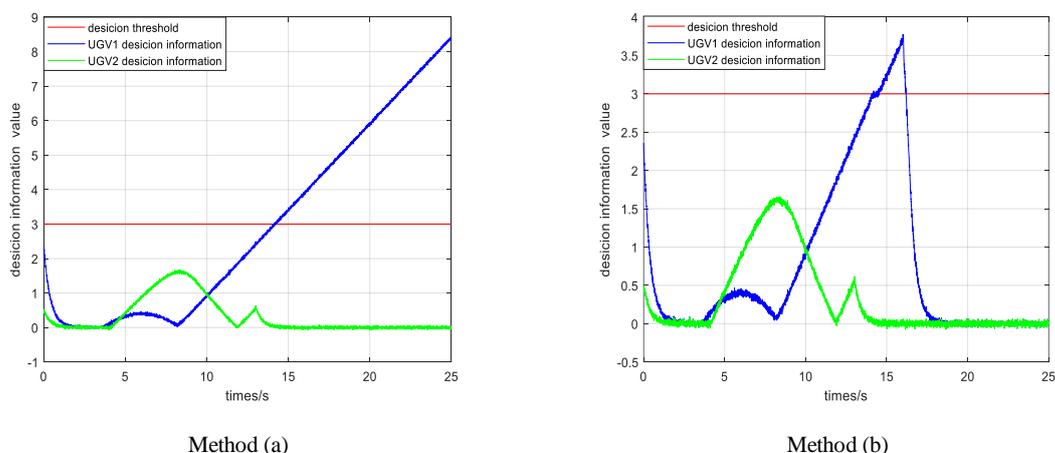


Figure 4. The figure shows the decision variables of method (a) and method (b).

Figure 5 shows the distances between the UGVs and their nearby obstacles. Once the UGV is moving into the range of the obstacle or another UGV, the obstacle avoidance task is activated with a higher priority to avoid collision. Method (a) shows that the UGV1 is stuck in the extreme point and cannot move, and the distance from the obstacle is constant

at 2 m. Method (b) shows that the UGV1 can get rid of the extreme point by effective intervention.

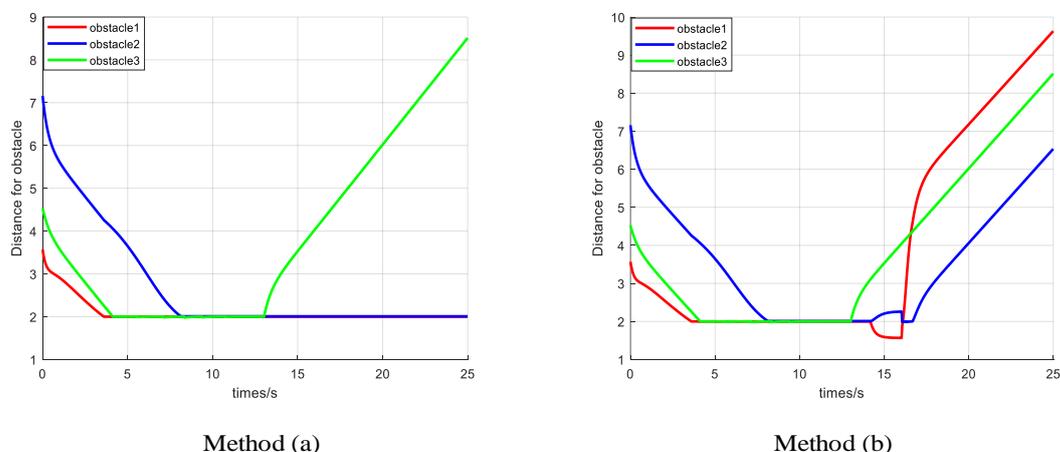


Figure 5. The figure shows the distance between the UGV and the obstacle in method (a) and method (b).

Figure 6 shows that in method (b), only when the decision variable reaches the decision threshold, will the UAV launch an intervention task. After the MPC intervention task decision maker, the UGV chooses to accept the intervention task. Method (a): Since it does not involve event triggering, it is not shown here.

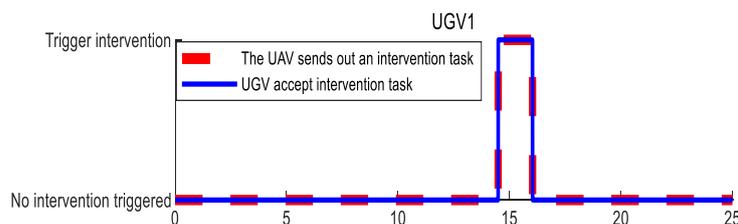


Figure 6. The figure shows method (b) event triggering mechanism.

5.2. Case B

In this case, by comparing method (b) and method (c), it is verified whether the UGV accepts the undesirable intervention task given by the UAV, so as to ensure its own safety. In method (b), due to factors such as disturbance or visual occlusion, the UAV gives the wrong intervention task, which causes UGV to collide with the sudden obstacle and cause danger. In method (c), since the MPC intervention task decider optimizes the intervention task and the UGV’s original composite task, when the wrong intervention task is given, the UGV chooses not to accept the intervention task. Until an effective intervention task is given, the UGV chooses to perform the intervention task. Please note that, at this time, due to the formation task of the UAV, the UAV’s trajectory is also shifted. The trajectory of method (c) and method (b) are shown in Figure 7.

In method (b), the wrong intervention task causes the UGV to crash into sudden obstacles (5.5 m, 5.8 m), which causes the UAV to continuously increase the decision variables of the UGV1 due to the failure of the UGV1. In method (c), although the UAV’s decision variables will continue to increase due to the maintenance of its original task until UAV effectively intervenes task to make the UGV get rid of the extreme point, at least the safety of the UGV is protected. As shown in Figure 8.

It can be seen from Figure 9 that the wrong intervention task of method (b) causes the distance between the UGV and the sudden obstacle to be zero. Method (c) effectively avoids this situation, even if a little time is sacrificed.

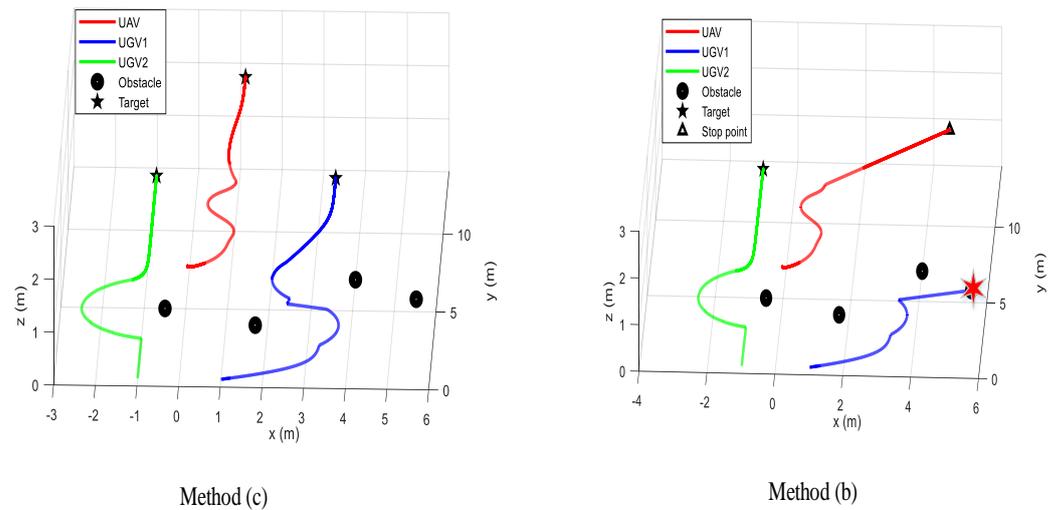


Figure 7. The picture shows the air-ground system trajectory of method (c) and method (b). In method (c), the UAV first sends out the wrong intervention task (UGV not received), and then sends out effective intervention task (UGV received). The UGV finally reaches the target point. In method (b), the UGV directly receives the wrong trajectory of the intervention task. The UGV encounters obstacles.

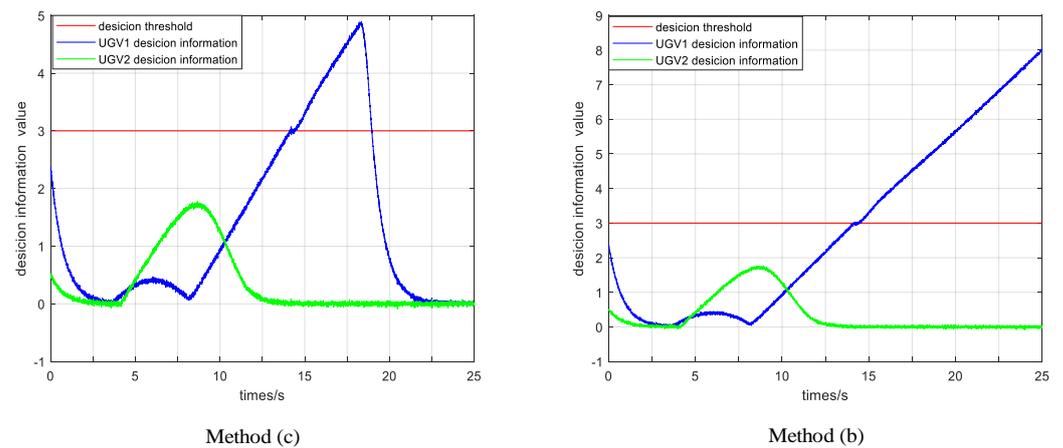


Figure 8. The figure shows the decision variables of method (c) with the wrong intervention first, then the correct intervention and method (b) with wrong intervention.

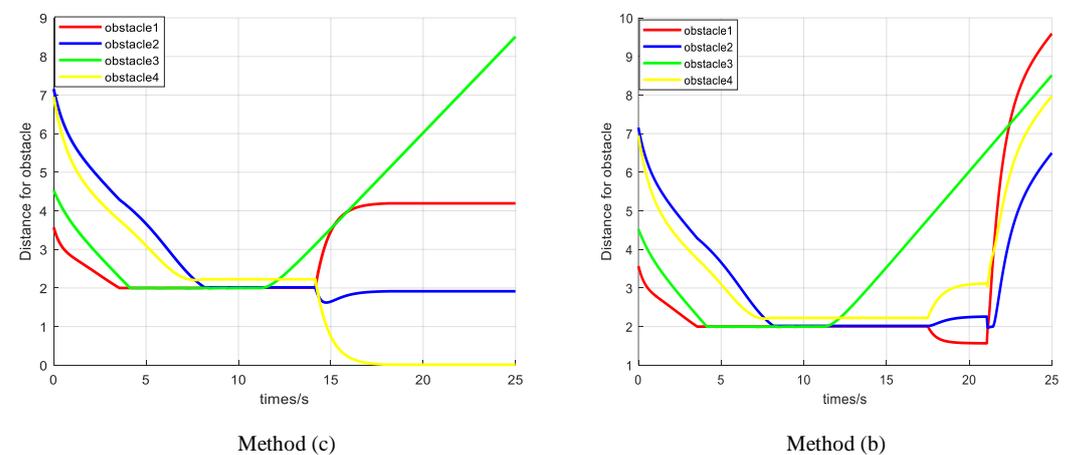


Figure 9. The figure shows the distance between the UGV and the obstacle in method (c) and method (b).

Figure 10 shows that Method (b) suggests that UGV continue to receive the wrong intervention tasks after the event is triggered. Method (c) shows that after MPC optimization, even if the UAV sends out an intervention task, the UGV chooses not to accept it until an effective intervention task is given.

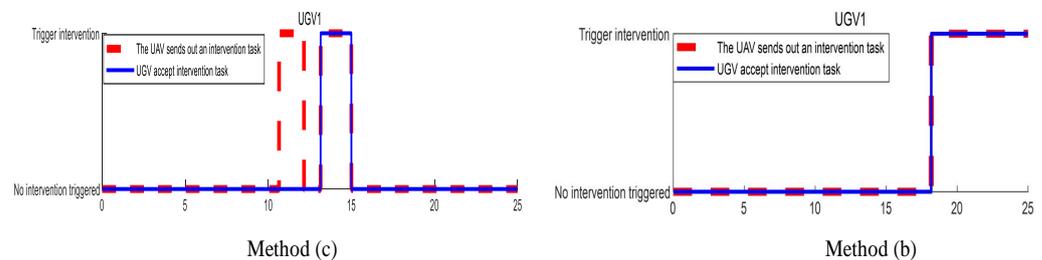


Figure 10. The figure shows the event-triggered performance of the method (c) and method (b), respectively. For method (c), After 18.1s, the UAV continues to send wrong intervention tasks, and the UGV continues to accept the wrong tasks. For method (b), the UAV sends the wrong intervention tasks at 10.6 s, which is refused by the decision maker based on MPC of the UGV. At 13.1 s, the UAV sends a correct intervention task, which is accepted by the UGV.

6. Conclusions and Future Work

This paper has proposed a new type of event-triggered mechanism-based air-ground coordination system for UAVs intervening in UGVs. The DDM has been embedded into NSBC to reduce the communication burden, and the intervention task decision maker based on MPC is designed to ensure that UGVs only accept safe and effective intervention tasks. At the end of this paper, two cases are studied to compare the performance of the system in situations without intervention, with direct intervention (correct or wrong), and with optimal intervention selected from the MPC. The results show that the UAV intervention task is triggered only when the decision variable reaches the decision threshold. The UGV can identify and reject wrong intervention tasks, ensuring its own safety.

In this paper, it is assumed that UAVs can give proper intervention tasks. However, no detailed description of how to design the intervention tasks is given. In future research, we will discuss how drones can select appropriate intervention tasks from the behavior database. At the same time, although DDM is used as a perceptual decision-making model for simulating humans, there remains a question that the intervention frequency may be infinite, i.e., Zeno behavior. We will consider this as the next step in research.

Author Contributions: Methodology: J.G. and Y.C.; software, J.G. and G.T.; investigation, W.W., J.G. and Y.C.; writing—original draft preparation, W.W., J.G. and Y.C.; writing—review and editing, W.W., Y.C. and J.H.; supervision, W.W. and J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China under Grants 61603094.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [[CrossRef](#)]
2. Lacroix, S.; Le Besnerais, G. Issues in cooperative air/ground robotic systems. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 421–432.
3. Chen, J.; Zhang, X.; Xin, B.; Fang, H. Coordination between unmanned aerial and ground vehicles: A taxonomy and optimization perspective. *IEEE Trans. Cybern.* **2015**, *46*, 959–972. [[CrossRef](#)] [[PubMed](#)]

4. Qin, H.; Meng, Z.; Meng, W.; Chen, X.; Sun, H.; Lin, F.; Ang, M.H. Autonomous exploration and mapping system using heterogeneous UAVs and UGVs in GPS-denied environments. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1339–1350. [[CrossRef](#)]
5. Liu, Y.; Luo, Z.; Liu, Z.; Shi, J.; Cheng, G. Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions. *IEEE Access* **2019**, *7*, 63504–63518. [[CrossRef](#)]
6. Li, J.; Deng, G.; Luo, C.; Lin, Q.; Yan, Q.; Ming, Z. A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9585–9596. [[CrossRef](#)]
7. Tokekar, P.; Vander Hook, J.; Mulla, D.; Isler, V. Sensor planning for a symbiotic UAV and UGV system for precision agriculture. *IEEE Trans. Robot.* **2016**, *32*, 1498–1511. [[CrossRef](#)]
8. Ding, Y.; Xin, B.; Chen, J. A Review of Recent Advances in Coordination Between Unmanned Aerial and Ground Vehicles. *Unmanned Syst.* **2021**, *9*, 97–117. [[CrossRef](#)]
9. Rodriguez-Ramos, A.; Sampedro, C.; Bavlle, H.; De La Puente, P.; Campoy, P. A deep reinforcement learning strategy for UAV autonomous landing on a moving platform. *J. Intell. Robot. Syst.* **2019**, *93*, 351–366. [[CrossRef](#)]
10. Jung, S.; Cho, H.; Kim, D.; Kim, K.; Han, J.I.; Myung, H. Development of algal bloom removal system using unmanned aerial vehicle and surface vehicle. *IEEE Access* **2017**, *5*, 22166–22176. [[CrossRef](#)]
11. Aranda, M.; López-Nicolás, G.; Sagüés, C.; Mezouar, Y. Formation control of mobile robots using multiple aerial cameras. *IEEE Trans. Robot.* **2015**, *31*, 1064–1071. [[CrossRef](#)]
12. Santana, L.V.; Brandão, A.S.; Sarcinelli-Filho, M. Heterogeneous leader-follower formation based on kinematic models. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 342–346.
13. Stentz, T.; Kelly, A.; Herman, H.; Rander, P.; Amidi, O.; Mandelbaum, R. Integrated Air/Ground Vehicle System for Semi-Autonomous Off-Road Navigation. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.
14. Peterson, J.; Chaudhry, H.; Abdelatty, K.; Bird, J.; Kochersberger, K. Online aerial terrain mapping for ground robot navigation. *Sensors* **2018**, *18*, 630. [[CrossRef](#)] [[PubMed](#)]
15. Mathews, N.; Christensen, A.L.; Stranieri, A.; Scheidler, A.; Dorigo, M. Supervised morphogenesis: Exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots. *Robot. Auton. Syst.* **2019**, *112*, 154–167. [[CrossRef](#)]
16. Marino, A. A Null-Space-based Behavioral Approach to Multi-Robot Patrolling. Ph.D. Thesis, Università degli Studi della Basilicata, Potenza, Italy, 2004.
17. Yao, P.; Wei, Y.; Zhao, Z. Null-space-based modulated reference trajectory generator for multi-robots formation in obstacle environment. *ISA Trans.* **2021**. [[CrossRef](#)] [[PubMed](#)]
18. Moreira, M.S.M.; Brandão, A.S.; Sarcinelli-Filho, M. Null space based formation control for a uav landing on a ugv. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 1389–1397.
19. Bacheti, V.P.; Brandão, A.S.; Sarcinelli-Filho, M. Path-following by a UGV-UAV Formation Based on Null Space. In Proceedings of the 2021 14th IEEE International Conference on Industry Applications (INDUSCON), São Paulo, Brazil, 15–18 August 2021; pp. 1266–1273.
20. Huang, J.; Wu, W.; Zhang, Z.; Chen, Y. A Human Decision-Making Behavior Model for Human-Robot Interaction in Multi-Robot Systems. *IEEE Access* **2020**, *8*, 197853–197862. [[CrossRef](#)]
21. Arrichiello, F. Coordination Control of Multiple Mobile Robots. Ph.D. Thesis, Università Degli Studi Di Cassino, Cassino, Italy, 2006; Volume 218.
22. Bogacz, R.; Brown, E.; Moehlis, J.; Holmes, P.; Cohen, J.D. The physics of optimal decision making: a formal analysis of models of performance in two-alternative forced-choice tasks. *Psychol. Rev.* **2006**, *113*, 700. [[CrossRef](#)] [[PubMed](#)]
23. Huang, J.; Zhou, N.; Cao, M. Adaptive fuzzy behavioral control of second-order autonomous agents with prioritized missions: Theory and experiments. *IEEE Trans. Ind. Electron.* **2019**, *66*, 9612–9622. [[CrossRef](#)]
24. Nirawana, I.W.S.; Aryanto, K.Y.E.; Indrawan, G. Mobile Robot Based Autonomous Selection of Fuzzy-PID Behavior and Visual Odometry for Navigation and Avoiding Barriers in the Plant Environment. In Proceedings of the 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 26–27 November 2018; pp. 234–239.
25. Chen, L.; Wu, M.; Zhou, M.; She, J.; Dong, F.; Hirota, K. Information-driven multirobot behavior adaptation to emotional intention in human–robot interaction. *IEEE Trans. Cogn. Dev. Syst.* **2017**, *10*, 647–658. [[CrossRef](#)]
26. Nie, M.; Luo, D.; Liu, T.; Wu, X. Action Selection Based on Prediction for Robot Planning. In Proceedings of the 2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), Oslo, Norway, 19–22 August 2019; pp. 201–206.
27. Chen, Y.; Zhang, Z.; Huang, J. Dynamic task priority planning for null-space behavioral control of multi-agent systems. *IEEE Access* **2020**, *8*, 149643–149651. [[CrossRef](#)]
28. Sager, S. Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. *J. Process. Control.* **2009**, *19*, 1238–1247. [[CrossRef](#)]
29. Bock, H.G.; Plitt, K.J. A multiple shooting algorithm for direct solution of optimal control problems. *IFAC Proc. Vol.* **1984**, *17*, 1603–1608. [[CrossRef](#)]