entropy

MDPI

Article

Rapidly Tuning the PID Controller Based on the Regional Surrogate Model Technique in the UAV Formation

Binglin Wang [1], Xiaojun Duan [1], Liang Yan [1,*], Juan Deng [1] and Jiangtao Chen [2]

[1] College of Liberal Arts and Sciences, National University of Defense Technology, Changsha 410073, China; wangbinglin14@nudt.edu.cn (B.W.); xjduan@nudt.edu.cn (X.D.); dengjuan@amss.ac.cn (J.D.)

[2] China Aerodynamics Research and Development Center, Mianyang 621000, China; chenjt@cardc.cn

* Correspondence: yanliang@nudt.edu.cn

Received: 3 April 2020; Accepted: 29 April 2020; Published: 6 May 2020

Abstract: The leader–follower structure is widely used in unmanned aerial vehicle formation. This paper adopts the proportional-integral-derivative (PID) and the linear quadratic regulator controllers to construct the leader–follower formation. Tuning the PID controllers is generally empirical; hence, various surrogate models have been introduced to identify more refined parameters with relatively lower cost. However, the construction of surrogate models faces the problem that the singular points may affect the accuracy, such that the global surrogate models may be invalid. Thus, to tune controllers quickly and accurately, the regional surrogate model technique (RSMT), based on analyzing the regional information entropy, is proposed. The proposed RSMT cooperates only with the successful samples to mitigate the effect of singular points along with a classifier screening failed samples. Implementing the RSMT with various kinds of surrogate models, this study evaluates the Pareto fronts of the original simulation model and the RSMT to compare their effectiveness. The results show that the RSMT can accurately reconstruct the simulation model. Compared with the global surrogate models, the RSMT reduces the run time of tuning PID controllers by one order of magnitude, and it improves the accuracy of surrogate models by dozens of orders of magnitude.

Keywords: surrogate model; proportional controller; UAV formation; classifier

1. Introduction

The cooperative control of the unmanned aerial vehicle (UAV) formation is a research hotspot because of its widespread use, such as in forest fire surveillance, field surveillance, and antipoaching efforts [1,2]. Tuning controllers with efficient optimization methods is of prime importance to maintaining robust formation. In practice, the classical proportional-integral-derivative (PID) controller and its variations, such as the proportional controller and the proportional-integral controller, occupy 90% of industrial control [3]. However, many engineers think that many PID control loops in practice are not in high performance [3]. It is notable that the PID controller is parameter sensitive; hence, a more refined optimization method is required. Our study focuses on developing a high-efficient method to tune PID controllers.

Many researchers attempted to improve the robustness of UAVs through designing controllers. Some researchers researched the robustness of a single UAV. López-Estrada et al. [4] designed a robust fault detection and tracking controller system. Guzmán-Rabasa et al. [5] designed the fault detection and diagnosis system when a UAV was under partial or total actuator fault. The robust control of UAV formations has also attracted the attention of researchers. To design a robust UAV formation, which was independent of the environment, Viktor et al. [6] proposed an onboard relative localization method based on ultraviolet light. The robustness of UAVs with specific tasks has also been studied. Guerrero-Sánchez et al. [7] controlled single UAV with a cable-suspended payload through a hierarchical scheme with controllers based on energy and the linear matrix inequality. Tuning controllers plays an important role in keeping robust UAV systems.

To accelerate the process of tuning controllers, surrogate models (SUMOs) have been introduced because theoretical tuning methods or empirical tuning methods may be cumbersome or inefficient [8,9]. Regarding a system as a black box, SUMOs mimic relationships between system inputs and outputs. Hence, SUMOs have good adaptability. There are some common types of SUMOs, such as Kriging [10], polynomial chaos expansions (PCE) [11], polynomial chaos Kriging (PCK) [12], the radial basis function neural network (RBFNN) [13], and the generalized regression neural network (GRNN)[14]. It is worth noting that SUMOs have been widely used to optimize UAVs [15–17]. Researchers have used SUMOs to tune different controllers in different systems successfully, such as the mixing process [18], the cruise control system [19], and the unmanned underwater vehicle [20]. Among these systems, through offline optimization, various controllers were tuned, including the fuzzy logic controller [20], the proportional-integral controller [18], and the PID controller [18,19]. Additional SUMO-related techniques have been introduced. Lü [21] performed online optimization on high-purity distillation processes via the RBFNN. To investigate large, multidimensional input spaces, Matinnejad et al. [22] reduced the dimensionality of SUMOs, including the linear regression, the exponential regression, and the polynomial regression. Pan and Das [23] adopted Kriging to optimize the fractional order PID controller. Guerrero et al. [24] proposed a surrogate-based optimization workflow. Faruq et al. [25] proposed a Pareto-based surrogate modelling algorithm for optimizing PID controllers.

The previous works usually construct global SUMOs for control systems. However, the global SUMOs may not be the best choice for tuning controllers because researchers only concern the successful part of control systems. The accuracy of global SUMOs may be affected by the failed control results. For example, in this study, the UAV formation may generate singularity points when the control fails. Singularity points are fatal to the accuracy of SUMOs. In previous studies, singularity points did not raise close attention because researchers have prior experience, and successful samples were easy to be found [18–20,23,25]. The reasons for singularity points occurrence in this work are summarized as follows: First, the error of closed-loop systems may be reinforced compared to open-loop systems. Second, the solver in the simulation program may get exceptionally large values when the control fails. In this study, without prior experience, it is hard to avoid the singularity points before sampling. Therefore, recklessly tuning controllers using global SUMOs is problematic. A novel SUMO technique is thus needed to filter singularity points.

The remainder of this paper is presented as follows. Section 2 constructs the UAV formation simulation model and defines performance measures. In Section 3, the regional surrogate model technique (RSMT) is proposed based on the regional information entropy. In Section 4, the RSMT is used by different SUMOs, i.e., Kriging, PCE, PCK, the RBFNN, and the GRNN. Then, the Pareto fronts of the original simulation model and the RSMT are evaluated to compare their effectiveness.

2. The UAV Formation Model

2.1. The Leader–Follower Structure

Following Xu [26], fixed-wing UAVs form the UAV formation, which adopts the leader–follower (L–F) architecture: one leader leads the group while followers are controlled to maintain clearance between followers and the leader. The earth-fixed reference frame is built, and the dynamic models of UAVs [27] are given by

$$\begin{cases} \dot{x}_{\mathrm{L}} = V_{\mathrm{L}} \cos\phi_{\mathrm{L}} \cos\theta_{\mathrm{L}} \\ \dot{y}_{\mathrm{L}} = V_{L} \sin\phi_{\mathrm{L}} \cos\theta_{\mathrm{L}} \\ \dot{z}_{\mathrm{L}} = V_{\mathrm{L}} \sin\theta_{\mathrm{L}} \\ \dot{x}_{\mathrm{F}} = V_{\mathrm{F}} \cos\phi_{\mathrm{F}} \cos\theta_{\mathrm{F}} \\ \dot{y}_{\mathrm{F}} = V_{\mathrm{F}} \sin\phi_{\mathrm{F}} \cos\theta_{\mathrm{F}} \\ \dot{z}_{\mathrm{F}} = V_{\mathrm{F}} \sin\theta_{\mathrm{F}} \end{cases}, \tag{1}$$

where the subscripts L and F denote the leader and follower, respectively; x, y, and z denote the position of UAVs on the x-axis, y-axis, and z-axis; V is the forward velocity; θ is the track angle of UAVs; ω is the heading angular rate of UAVs, $\dot{\phi} = \omega$. As the angle between the forward direction and x-axis, the heading angle ϕ [26] can be given by

$$\sin\phi = \frac{V_y}{\sqrt{V_x^2 + V_y^2}}, \tag{2}$$

where V_x and V_y are the components of V on the x-axis and y-axis. Because this paper focuses on fixed-wing UAVs which usually fly at the same height in a formation [26,28,29], we assume that UAVs do not change their height, i.e., $\theta_L = \theta_F = 0$. Because the method of controlling all followers is identical, and there is no connection between followers, we examine only one follower instead of multiple followers. Figure 1 shows the geometry of the L-F structure in the x, y plane as follows:

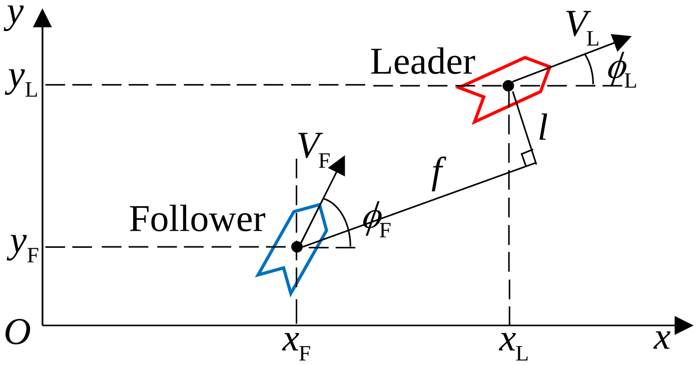

Figure 1. The leader–follower structure [29] in the x, y plane. One leader leads the group while the follower is controlled to maintain clearance between the follower and the leader.

The position relations between the leader and follower [26] are

$$\begin{cases} \Delta f = (x_{\mathrm{L}} - x_{\mathrm{F}}) \cos\phi_{\mathrm{L}} + (y_{\mathrm{L}} - y_{\mathrm{F}}) \sin\phi_{\mathrm{L}} - f_{\mathrm{d}} \\ \Delta l = -(x_{\mathrm{L}} - x_{\mathrm{F}}) \sin\phi_{\mathrm{L}} + (y_{\mathrm{L}} - y_{\mathrm{F}}) \cos\phi_{\mathrm{L}} - l_{\mathrm{d}} \end{cases}, \tag{3}$$

where f_{d} and l_{d} are the desired forward and lateral clearances; Δf and Δl are the clearance errors in the forward and lateral directions.

The L-F structure aims to keep the desired clearance between the follower and the leader. The UAV formation is divided into the outer loop and the inner loop, which contain PID controllers and linear quadratic regulator (LQR) controllers, respectively. The outer loop controls the position dynamics to maintain the desired formation; the inner loop controls the UAV itself. The outer-loop controller generates commands into the inner-loop controller. The conceptual structure of the used UAV formation is shown in Figure 2. The reference generator gives the velocity and attitude of the leader [29]. Appendix A provides details of the inner-loop-controller design and the system matrices of a single UAV. Because the LQR controller belongs to optimum control, we only optimize the outer-loop controller, i.e., the PID controller, which is designed as follows.

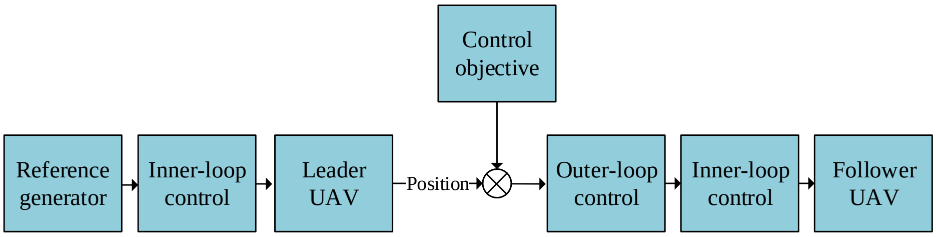

Figure 2. The conceptual block diagram of the leader–follower UAV formation.

2.2. Outer-Loop-Controller Design

It is assumed that f and l are the actual forward and lateral clearances from the leader reference frame [29]:

$$\begin{cases} f = (x_\mathrm{L} - x_\mathrm{F})\cos\phi_\mathrm{L} + (y_\mathrm{L} - y_\mathrm{F})\sin\phi_\mathrm{L} \\ l = -(x_\mathrm{L} - x_\mathrm{F})\sin\phi_\mathrm{L} + (y_\mathrm{L} - y_\mathrm{F})\cos\phi_\mathrm{L} \end{cases}. \tag{4}$$

Differentiate the formula Equation (3) with respect to time, through substituting Equations (1) and (4), the rates of error change [29] are

$$\begin{bmatrix} \Delta\dot{f} \\ \Delta\dot{l} \end{bmatrix} = \begin{bmatrix} V_\mathrm{L} - l\omega_\mathrm{L} \\ -f\omega_\mathrm{L} \end{bmatrix} + \begin{bmatrix} -\cos(\phi_\mathrm{F} - \phi_\mathrm{L}) \\ -\sin(\phi_\mathrm{F} - \phi_\mathrm{L}) \end{bmatrix} V_\mathrm{F}. \tag{5}$$

The outer-loop controllers aim to generate proper commands, which will be tracked by the inner-loop controllers. We adopt two PID controllers as the outer-loop controllers in the forward and lateral directions. The two PID controllers are represented as M_l and M_f, which are given as follows:

$$M_\mathrm{l}(\Delta l) = K_\mathrm{Pl}\Delta l + K_\mathrm{Il}\int \Delta l dt + K_\mathrm{Dl}\frac{d\Delta l}{dt}, \tag{6}$$

$$M_\mathrm{f}(\Delta f) = K_\mathrm{Pf}\Delta f + K_\mathrm{If}\int \Delta f dt + K_\mathrm{Df}\frac{d\Delta f}{dt}, \tag{7}$$

where subscripts P, I, D represent the proportional gain, integral gain, and derivative gain of PID controllers, respectively; subscripts f and l represent the forward and lateral directions of UAVs, respectively. It is assumed that $\boldsymbol{K} = \{K_\mathrm{Pl}, K_\mathrm{Il}, K_\mathrm{Dl}, K_\mathrm{Pf}, K_\mathrm{If}, K_\mathrm{Df}\}$, which are user-defined and the key of tuning PID controllers. Then, Equation (5) can be written as

$$\begin{bmatrix} \Delta\dot{f} \\ \Delta\dot{l} \end{bmatrix} = \begin{bmatrix} V_\mathrm{L} - l\omega_\mathrm{L} \\ -f\omega_\mathrm{L} \end{bmatrix} + \begin{bmatrix} -\cos(\phi_\mathrm{F} - \phi_\mathrm{L}) \\ -\sin(\phi_\mathrm{F} - \phi_\mathrm{L}) \end{bmatrix} V_\mathrm{F} = \begin{bmatrix} -M_f(\Delta f) \\ -M_l(\Delta l) \end{bmatrix}, \tag{8}$$

Then, rearranging Equation (8), the following equation is gotten:

$$\begin{bmatrix} \cos(\phi_\mathrm{F} - \phi_\mathrm{L}) \\ \sin(\phi_\mathrm{F} - \phi_\mathrm{L}) \end{bmatrix} V_\mathrm{F} = \begin{bmatrix} M_f(\Delta f) + V_\mathrm{L} - l\omega \\ M_l(\Delta l) - f\omega \end{bmatrix}. \tag{9}$$

Let $h_\mathrm{F1} = M_f(\Delta f) + V_\mathrm{L} - l\omega_\mathrm{L}$, $h_\mathrm{F2} = M_l(\Delta l) - f\omega_\mathrm{L}$. The reference commands for the follower [29] are

$$V_\mathrm{F}^\mathrm{r} = \sqrt{h_\mathrm{F1}^2 + h_\mathrm{F2}^2}, \tag{10}$$

$$\phi_\mathrm{F}^\mathrm{r} = \begin{cases} \phi_\mathrm{L} + \pi/2 & h_\mathrm{F1} = 0, h_\mathrm{F2} > 0 \\ \phi_\mathrm{L} - \pi/2 & h_\mathrm{F1} = 0, h_\mathrm{F2} < 0 \\ \phi_\mathrm{L} + \arctan(h_\mathrm{F2}/h_\mathrm{F1}) & h_\mathrm{F1} > 0 \\ \phi_\mathrm{L} + \arctan(h_\mathrm{F2}/h_\mathrm{F1}) - \pi & h_\mathrm{F1} < 0, h_\mathrm{F2} \leq 0 \\ \phi_\mathrm{L} + \arctan(h_\mathrm{F2}/h_\mathrm{F1}) + \pi & h_\mathrm{F2} < 0, h_\mathrm{F1} \geq 0 \end{cases}. \tag{11}$$

2.3. Performance Measures of the UAV Formation

The follower's trajectory generates response curves, whose horizontal axis is the time and whose vertical axis is the clearance to the leader in two directions. Response curves are evaluated via three kinds of commonly used measures, as follows:

- **Steady-state value** (y_{st}): the stable value of the response curve, which is the direct aim of the controller.
- **Overshoot** (σ): the maximum peak value of the response curve measured from the desired response, which is given by [30]

$$\sigma\% = \frac{y_{\max} - y_{\text{st}}}{y_{\text{st}}} \times 100\%, \tag{12}$$

  where $y_{\max}$ is the peak value of the response curve beyond y_{st}.
- **Accommodation time** (t_{a}): the time at which the response curve enters a specific interval around the desired response and no longer exceed the specific interval.

The lateral and forward motion are mutually independent and controlled by different controllers, so y_{st} and σ are divided into the lateral steady-state value l_{st}, the forward steady-state value f_{st}, the lateral overshoot σ_{l}, and the forward overshoot σ_{f}.

3. The Regional Surrogate Model Technique Based on the Regional Information Entropy

A change of systems, especially for actual physical systems, is usually a gradual process, which makes the response surface smooth, such as in computational fluid dynamics [31], aerology [32], and hydrology [33]. Thus, the global SUMO is adopted in most cases. However, in this study, singular points make the response surface rough, and the global SUMO is no longer effective. There are two reasons for this phenomenon: First, the UAV formation is a closed-loop system, which may reinforce errors. Second, the solvers in simulation fail to solve equations, which lead to the generation of singular points. Without prior experience for determining the selection of parameter space, singular points are unavoidable, and it is essential to mitigate the effect caused by singular points. Based on the regional information entropy, the RSMT is proposed as a means of reconstructing the UAV formation.

3.1. Regional Information Entropy Analysis

The SUMO can be viewed as a way to reconstruct the information of systems. Hence, a reasonable SUMO should fully display useful information and avoid interference from useless information, which in this study is mainly caused by singular points. Hence, analyzing the regional information entropy relationship can provide us with a decision basis for screening information. As a way of measuring the information content, information entropy S [34] is given by

$$S = -\int p\left(x\right) \ln p\left(x\right) dx, \tag{13}$$

where x is the output of the system and $p\left(x\right)$ is the probability distribution function (PDF) of x. There is a positive correlation between S and information content.

For simplicity, we examine only one input with one output. The space of success (SOS) is the success interval I_{succ}, which is the set of successful outputs. I_{succ} needs to contain all potential optimal solutions. The space of failure (SOF), i.e., the failed interval I_{fail}, is the set of failed outputs, and $I_{\text{succ}} \cap I_{\text{fail}} = \varnothing$. P_{fail} and P_{succ} are the probabilities of outputs belonging to I_{fail} and I_{succ} respectively. Because success and failure are complementary events, $P_{\text{succ}} + P_{\text{fail}} = 1$. S_{succ} and S_{fail} are the information entropy of I_{succ} and I_{fail}, respectively. Containing useful and useless information, the entropy of the entire system is $S_{\text{succ}} + S_{\text{fail}}$, which is the whole information entropy of the global SUMO, i.e., the global SUMO completely reconstruct the entire system. The information entropy ratio of two kinds of information is W, $W = S_{\text{succ}}/S_{\text{fail}}$. It is assumed that I_{succ} and I_{fail} are both uniform distributions; then, the PDF of x is given by

$$p\left(x\right) = \begin{cases} P_{\text{succ}}/\left(b-a\right) & a \leq x \leq b \\ P_{\text{fail}}/\left(a - x_{\min} + x_{\max} - b\right) & x_{\min} \leq x < a \quad \text{or} \quad b < x \leq x_{\max} \end{cases}, \tag{14}$$

where a, b are the bounds of I_{succ}, $I_{\text{succ}} \in [a,b]$; $x_{\min}$, $x_{\max}$ are the lower limit and upper limit of x, $I_{\text{fail}} \in [x_{\min}, a) \cup (b, x_{\max}]$. S_{fail} and S_{succ} are given by

$$\begin{aligned} S_{\text{fail}} = & -\int_{x_{\min}}^{a} \frac{P_{\text{fail}}}{a - x_{\min} + x_{\max} - b} \ln \frac{P_{\text{fail}}}{a - x_{\min} + x_{\max} - b} dx \\ & -\int_{b}^{x_{\max}} \frac{P_{\text{fail}}}{a - x_{\min} + x_{\max} - b} \ln \frac{P_{\text{fail}}}{a - x_{\min} + x_{\max} - b} dx \\ & = -P_{\text{fail}} \ln \frac{P_{\text{fail}}}{a - x_{\min} + x_{\max} - b}, \end{aligned} \tag{15}$$

$$S_{\text{succ}} = -\int_{a}^{b} \frac{P_{\text{succ}}}{b-a} \ln \frac{P_{\text{succ}}}{b-a} dx = -P_{\text{succ}} \ln \frac{P_{\text{succ}}}{b-a}. \tag{16}$$

It is assumed that $a = -5$, $b = 5$, $x_{\min} = -1000$, and $x_{\max} = 1000$. According to Equations (15) and (16), Case 1 in Figure 3 shows the relationship between P_{fail} and W. If there is no prior experience in parameter selection, P_{succ} will be small, which makes $S_{\text{fail}} > S_{\text{succ}}$; in other words, useless information covers up useful information. Hence, the new SUMO technique should prevent useful information from being concealed by increasing W. In practice, we do not consider the output value and input parameter of failed results, which is the source of useless information. Hence, ignoring the difference within failed results is reasonable. Regarding failed results as one event, S_{fail} and S_{succ} can be given by

$$S_{\text{fail}} = -P_{\text{fail}} \ln\left(P_{\text{fail}}\right), \tag{17}$$

$$S_{\text{succ}} = -\int_{a}^{b} p_{\text{succ}}\left(x\right) \ln p_{\text{succ}}\left(x\right) dx, \tag{18}$$

where $p_{\text{succ}}\left(x\right)$ is the PDF of I_{succ}, $x \in I_{\text{succ}}$; $P_{\text{succ}} = \int_a^b p_{\text{succ}}\left(x\right) dx$. Assuming that I_{succ} is the uniform distribution, $p_{\text{succ}}\left(x\right)$ and W are given by

$$p_{\text{succ}}(x) = \begin{cases} 0 & x < a \quad \text{or} \quad x > b \\ P_{\text{succ}}/\left(b-a\right) & a \leq x \leq b \end{cases}, \tag{19}$$

$$W = \frac{S_{\text{succ}}}{S_{\text{fail}}} = \frac{-P_{\text{succ}} \ln\left(P_{\text{succ}}/\left(b-a\right)\right)}{-P_{\text{fail}} \ln P_{\text{fail}}} = \frac{\left(1-P_{\text{fail}}\right) \ln\left(\left(1-P_{\text{fail}}\right)/\left(b-a\right)\right)}{P_{\text{fail}} \ln P_{\text{fail}}}. \tag{20}$$

Case 2 in Figure 3 shows the relationship between P_{fail} and W. For Case 2 in Figure 3, W is always larger than 5, which means that the proportion of useless information is reduced, and useful information constitutes almost the entirety of the information. Ignoring the difference within I_{fail} effectively eliminates useless-information interference. Moreover, to verify the results shown Case 1 in Figure 3, Case 3 in Figure 3 shows the same relation when the distribution of x is Student's t-distribution. A detailed discussion is provided in Appendix B. The values of a, b, $x_{\min}$, and $x_{\max}$ impact W slightly; hence, the changes of these values do not affect the related conclusions.

In conclusion, constructing SUMOs needs to reduce the concealing of useful information. Different results should be differently treated according to the aim of constructing SUMOs. Based on the analysis presented above, the RSMT is proposed as a means of tuning PID controllers in the UAV formation.

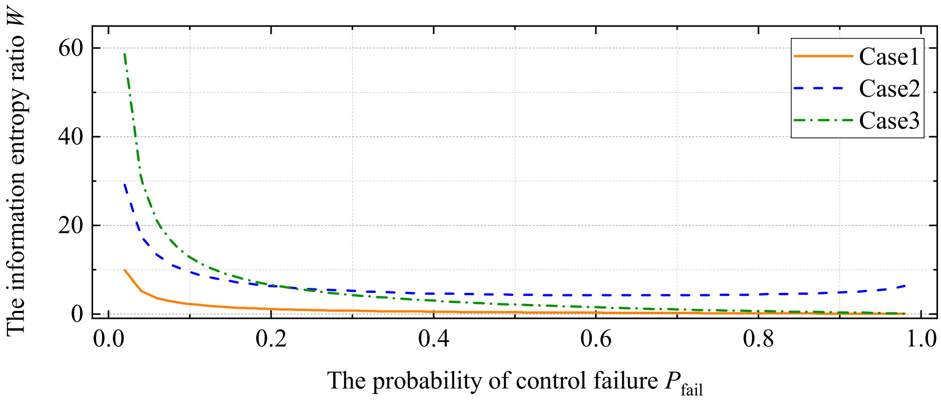

Figure 3. The relationship between P_{fail} and W. Case 1: all samples are used in the SUMO construction. I_{succ} and I_{fail} are both uniform distributions. Case 2: the failed results are viewed as one event, and I_{succ} is the uniform distribution. Case 3: I_{succ} is the uniform distribution and I_{fail} is the t-distribution. Filtering useless information is essential for preventing useful information from being submerged. Ignoring the difference within I_{fail} effectively eliminates useless-information interference.

3.2. The Regional Surrogate Model Technique

Section 3.1 discusses the relationship between S_{fail} and S_{succ}. To purify information, we propose the RSMT, which is shown in Algorithm 1 and Figure 4 (the source code can be obtained from the authors). Class 1 means that the sample belongs to the SOS, and class 0 is contrary to class 1. Whether the control is successful or failed is determined according to user-defined thresholds. As discussed in Section 3.1, we ignore the difference within I_{fail} and focus on I_{succ}. Instead of the global SUMO, the RSMT constructs the regional SUMO, which reconstructs the system only in the SOS. The RSMT can also be viewed as a weighted global SUMO: the weight of training samples belonging to the SOS is one; the weight of other samples is zero.

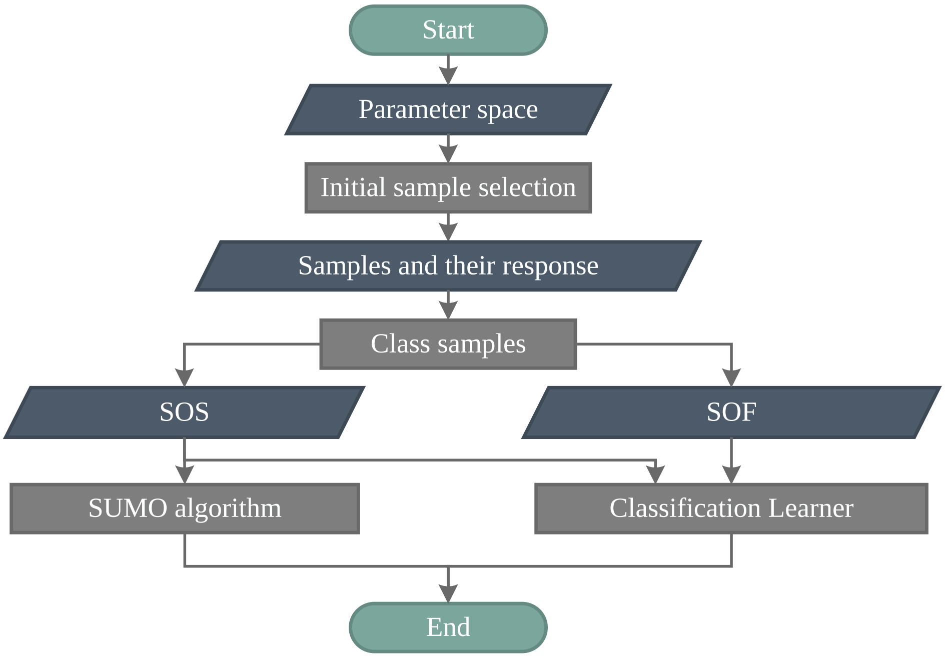

Figure 4. The regional surrogate model technique. The SUMO is constructed only in the SOS, whose boundary is found by a classification learner.

Algorithm 1 The regional surrogate model technique.

```
Input: the number of initial samples $N$; the parameter space $PS$; the criteria of the SOS.
Output: A classifier; a regional SUMO
    Definition: the selected training set for the SUMO $ST$; the training set for classifier $CT$
    Make the initial sample selection from the $PS$ and get $N$ samples
    Put selected samples into the simulation model to get their response
    for each sample and its response
      if $i$-th sample belongs to the SOS
        add $i$-th sample and its response into $ST$;
        classify $i$-th sample with class 1;
        add $i$-th sample and its class into $CT$;
      else
        classify $i$-th sample with class 0;
        add $i$-th sample and its class into $CT$.
      end if
    end for
    Train the SUMO by $ST$
    Train the classifier by $CT$
```

In the RSMT, distinguishing class $0/1$ requires user-defined thresholds, which should be more lenient than control objectives to avoid ignoring potential optimal solutions. After classing samples in accordance with thresholds, a classifier is trained by samples and their subordinate class to find the boundary of the SOS, which is difficult to describe analytically. Only samples belonging to the SOS are selected as the training set of SUMOs. Thus, it is limited to the use of trained SUMOs, whose use process is shown in Figure 5. When inputting new parameters, the first step is classifying the new parameters by the trained classifier. If it can lead to successful control, the outputs of the parameters are predicted by the trained SUMOs. Otherwise, these parameters are abandoned because they do not belong to potential optimal solutions.

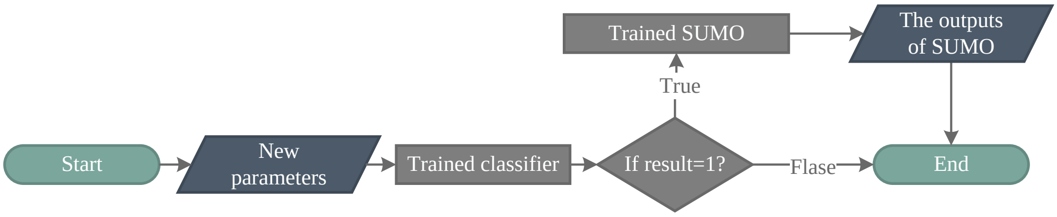

Figure 5. The use of trained SUMOs obtained by the RSMT. Inputs are judged by the classifier, and only the inputs belonging to the SOS are predicted by the trained SUMO. If the result of classifier equals to 1, it means that the new parameters belong to the SOS.

Instead of optimizing parameters of SUMOs, the RSMT focuses on selecting a more reasonable training set for constructing SUMOs in a specific region. In this paper, the classifier adopts a decision tree, which performs well in binary classification and is given in Algorithm 2, following [35]. In a sense, the global SUMO is the combination of multiple regional SUMOs, whose marginal values are the same. If the response surface is rough, it is difficult to mimic the dramatically changed response surface using only one SUMO. Because the RSMT constructs the regional SUMO in the SOS whose outputs are limited, the selected response surface will be smooth, as a result of which the regional SUMO has high accuracy without missing the potential optimal solution.

Algorithm 2 Generating decision tree.

```
Input: $D$: the training set; $C$: the attribute set.
Output: A decision tree
Function TreeGenerate $(D, C)$
Create a node $N$
if tuples in $D$ belong to only one class $C$ then
    label $N$ as a leaf node with class $C$; return
end if
if $C$ is empty OR the samples of $D$ are of the same class then
    set label $N$ as the leaf node with the most common class in $D$; return
end if
Find the best splitting criterion $c_*$ from $C$
for each $c_*$ do
    add a branch below $N$, corresponding to $c_* = c_*^v$
    $D_v$ is the subset of $D$ with $c_* = c_*^v$
        if $D_v$ is empty then
            label the branch node as the leaf node with the most common class in $D$; return
        else
            set TreeGenerate $(D_v, C \setminus \{c_*\})$ as the branch node
        end if
end for
```

4. Simulation and Results

4.1. Evaluation Results for SUMOs Based on the RSMT

In this study, we attempt to substitute the SUMO for the UAV formation in Section 2. As parameters to be optimized, inputs are $\boldsymbol{K} = \{K_{\mathrm{Pl}}, K_{\mathrm{Il}}, K_{\mathrm{Dl}}, K_{\mathrm{Pf}}, K_{\mathrm{If}}, K_{\mathrm{Df}}\}$. With no correlation between them, the six intervals of $\boldsymbol{K}$ form the entire parameter space. Outputs are five performance measures, i.e., l_{st}, f_{st}, σ_{l}, σ_{f}, and t_{a}. The trained SUMO is evaluated using the root mean squared error (RMSE) [36], which is given by

$$RMSE = \sqrt{\frac{1}{n_t}\sum_{i=1}^{n_t}(y_i - \hat{y}_i)^2}, \tag{21}$$

where n_t is the number of test points; $\hat{y}_i$ and y_i are the estimated value and exact value of the ith test point, respectively.

At first, the initial sample selection adopts Latin hypercube sampling. Table 1 shows the evolution results regarding whether or not SUMOs are constructed through the RSMT, and Appendix C provides a brief introduction to the applied SUMOs, including Kriging, PCE, PCK, the RBFNN, and the GRNN. If the RSMT is not adopted, global SUMOs are constructed. In the simulation, the initial positions of the leader and the follower are $(200\ \mathrm{m}, 200\ \mathrm{m})$ and $(0, 400\ \mathrm{m})$, which are the same as [29]. Control objectives are $f_{\mathrm{d}} = 100$ m and $l_{\mathrm{d}} = -100$ m, which are also the same as [29]. Optimization aims to find the optimal PID controllers that can maintain the formation with low l_{st}, f_{st}, σ_{l}, σ_{f}, and t_{a} under constraints that $|\Delta f| < 3\%\,|f_{\mathrm{d}}| = 3$ m and $|\Delta l| < 3\%\,|l_{\mathrm{d}}| = 3$ m. Minimizing σ_{l} and σ_{f} aims to reduce the risk of UAVs collisions, which are important in the UAV formation [37]. In this work, the thresholds for the SOS are $\pm f_{\mathrm{d}}$ and $\pm l_{\mathrm{d}}$, which mean that the regional SUMO will be constructed in the region $|\Delta f| < 100$ m and $|\Delta l| < 100$ m.

In Table 1, the corresponding values of each SUMO are the RMSE of f_{st}. "Time" is the total run time of constructing all SUMOs. It is assumed that the intervals of six inputs are the same. In Table 1, the intervals of $\boldsymbol{K}$ are the intervals of six inputs, i.e., K_{Pl}, K_{Il}, K_{Dl}, K_{Pf}, K_{If}, and K_{Df}. For an interval of $\boldsymbol{K}$, the first row shows results for regional SUMOs through the RSMT, and the second row shows results for global SUMOs. Regional SUMOs generate from our proposed method, i.e., the RSMT.

Meanwhile, global SUMOs are the traditional way to construct SUMOs, i.e., adopting all samples to construct SUMOs. Because the second row of each $\boldsymbol{K}$ constructs the global SUMO without the classifier, the result of "Classification accuracy" is "null". The calculation condition is MATLAB 2019a, Intel (R) Xeon (R) W-2145 CPU @ 3.70GHz, 32GB Memory, Windows 10.

Table 1. Comparison of regional and global SUMOs for f_{st}.

$\boldsymbol{K}$ [a]	**Classification Accuracy (%)** [b]	**Time (s)** [c]	**Kriging** [d]	**PCE** [d]	**PCK** [d]	**GRNN** [d]	**RBFNN** [d]
$[0, 0.1]$	85	350.7	8.3	12.3	7.7	21.0	14.9
	null	437.6	1.62×10^{6}	1.41×10^{6}	1.37×10^{6}	1.26×10^{6}	1.63×10^{7}
$[0, 0.2]$	85	317.9	11.4	16.2	16.3	20.8	26.6
	null	289.6	8.20×10^{14}	1.35×10^{14}	2.08×10^{14}	8.11×10^{13}	1.90×10^{15}
$[0, 0.3]$	84.6	318.3	8.6	9.2	8.9	14.2	29.4
	null	567.7	9.41×10^{25}	2.71×10^{25}	3.23×10^{26}	1.97×10^{25}	5.27×10^{26}
$[0, 0.4]$	81.8	307.4	7.4	9.4	8.1	11.5	16.5
	null	322.5	3.76×10^{37}	3.32×10^{36}	6.40×10^{37}	4.43×10^{36}	1.40×10^{38}
$[0, 0.5]$	79	111.5	7.7	8.6	7.4	12.2	12.3
	null	312.3	4.63×10^{51}	7.20×10^{50}	4.44×10^{50}	5.53×10^{50}	1.99×10^{52}
$[0, 0.6]$	80.2	88.5	7.4	8.7	8.5	12.1	11.4
	null	356.4	8.21×10^{59}	3.94×10^{59}	3.94×10^{59}	4.68×10^{59}	1.09×10^{61}

[a] Two PID controllers have six inputs which share the same interval, and inputs are $\boldsymbol{K} = \{K_{\mathrm{PI}}, K_{\mathrm{II}}, K_{\mathrm{DI}}, K_{\mathrm{Pf}}, K_{\mathrm{If}}, K_{\mathrm{Df}}\}$. For a value of $\boldsymbol{K}$, the first and second rows show results for regional SUMOs and global SUMOs, respectively. [b] Because the second row of each $\boldsymbol{K}$ constructs the global SUMO without the classifier, the result of "Classification accuracy" is "null". [c] "Time" is the total run time for constructing all SUMOs. [d] Values of SUMOs represent the RMSE of f_{st} for trained SUMOs. The RSMT can significantly reduce errors and save computation time.

According to Table 1, trained regional SUMOs are significantly better than trained global SUMOs. This phenomenon is in accord with the information relationship in Section 3.1. Adopting the same calculation method in Section 3.1, Table 2 shows the information entropy ratio W of corresponding $\boldsymbol{K}$ in Table 1. Because the sample size is limited, we use the frequency approximation as the probability. W_1 is the information entropy ratio with the RSMT and is corresponding to the first row of each $\boldsymbol{K}$ in Table 1. W_1 is large with the RSMT, which reserves useful information and avoids useless-information interference. At the same time, W_2 is the information entropy ratio without the RSMT and is corresponding to the second row of each $\boldsymbol{K}$ in Table 1. W_2 is relatively small and will decrease with the decrease of P_{succ}. It means that useful information will be covered by useless information with the decrease of P_{succ}. Moreover, W_1 is always larger than W_2. W_1/W_2 increases quickly with the expansion of $\boldsymbol{K}$. W_1/W_2 shows the change of the information entropy ratio with the RSMT or not. The RSMT effectively increases the proportion of useful information. Figure 6 shows the relationship between W_1/W_2 and the effects of the RSMT, which is represented by orders of magnitude change of Kriging's RMSE. The increase of W_1/W_2 brings the better effect of the RSMT, especially when W_1/W_2 is relatively small. The RSMT increases the proportion of useful information entropy, which leads to accurate regional SUMOs.

Regarding the computational cost, according to Table 1, training regional SUMOs is more time-saving with the expansion of $\boldsymbol{K}$ because the number of successful parameters will be fewer when the whole parameter space is larger. There are five different types of SUMOs, and the RSMT performs well in each of them, which shows that the RSMT has good generalization ability and is essential to various SUMOs.

In conclusion, the RSMT successfully filters singular points and maintains the high accuracy and low computational cost of regional SUMOs. In the next subsection, the optimal parameters of PID controllers are found through the RSMT.

Table 2. The information entropy ratio W in actual computation.

K	$f_{\mathrm{st},max}$	$f_{\mathrm{st},min}$	P_{fail}	P_{succ}	W_1 [a]	W_2 [b]	W_1/W_2
$[0, 0.1]$	3.14×10^{2}	-6.81×10^{8}	0.41	0.59	9.36	0.39	24.06
$[0, 0.2]$	1.18×10^{12}	-5.35×10^{16}	0.44	0.56	9.13	0.19	47.66
$[0, 0.3]$	3.26×10^{23}	-1.35×10^{28}	0.45	0.55	9.04	0.11	81.68
$[0, 0.4]$	1.26×10^{25}	-1.62×10^{39}	0.51	0.49	8.59	0.06	134.23
$[0, 0.5]$	1.76×10^{26}	-3.46×10^{53}	0.59	0.41	8.15	0.03	234.94
$[0, 0.6]$	4.03×10^{36}	-1.68×10^{62}	0.76	0.24	7.74	0.01	533.58

[a] W_1 is the information entropy ratio with the RSMT and is corresponding to the first row of each $\boldsymbol{K}$ in Table 1.
[b] W_2 is the information entropy ratio without the RSMT and is corresponding to the second row of each $\boldsymbol{K}$ in Table 1. W_1/W_2 increases quickly with the expansion of $\boldsymbol{K}$. Without the RSMT, useless information will cover up useful information. The RSMT effectively increases the proportion of useful information.

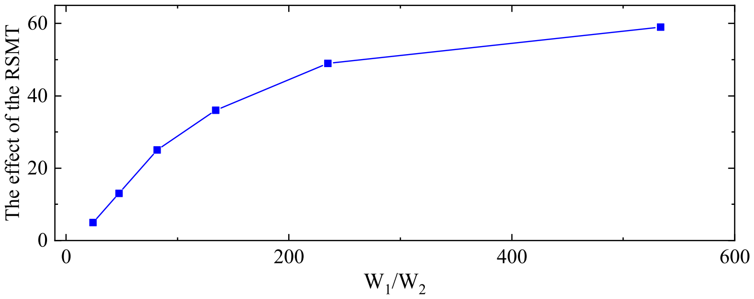

Figure 6. The relationship between W_1/W_2 and the effects of the RSMT. W_1/W_2 shows the change of the information entropy ratio with the RSMT or not. Effects of the RSMT are represented by orders of magnitude change of Kriging's RMSE in Table 1. The increase of W_1/W_2 brings the better effect of the RSMT, especially when W_1/W_2 is limited.

4.2. Tuning PID Controllers Through the RSMT

We try to tune two PID controllers whose six inputs all belong to $[0, 0.3]$. In the simulation, white noise is added to the lateral and forward positions of the leader, and the noise energy of white noise is 1×10^{-2}. As mentioned above, class 1 indicates that corresponding samples belong to the SOS, and the meaning of class 0 is reversed: class 0 indicates that corresponding samples belong to the SOF. The global Sobol sensitivity analysis is adopted to analyze the relationships between inputs and class 0/1, whose results are shown in Figure 7. According to Figure 7, K_{PI} and K_{II} are the most important inputs. Figure 8 shows prediction results of trained classifier in the K_{PI}, K_{II} plane. In Figure 8, blue symbols mean that the corresponding sample belongs to class 0, i.e., the SOF, and red symbols mean that the corresponding sample belongs to class 1, i.e., the SOS. "Correct" and "Incorrect" are the correctness of the classifier's prediction. The classification accuracy of the trained decision tree is 84.0%. Figure 9 shows the receiver operating characteristic (ROC) curve of the trained classifier. The area under the ROC curve equals 0.88. The false positive rate is 14%, and the true positive rate is 84%. Hence, the trained classifier is accurate and reliable.

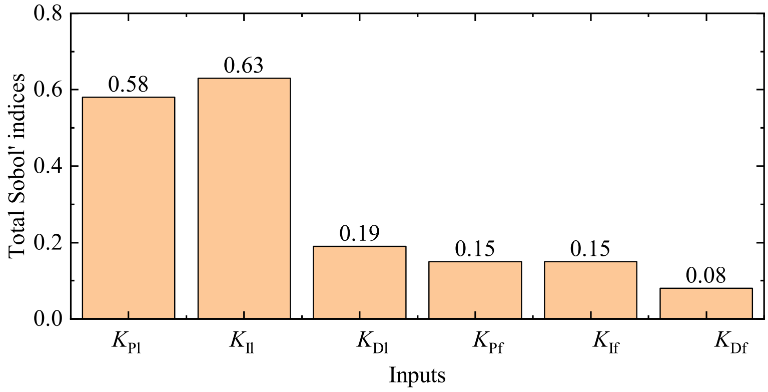

Figure 7. Total Sobol′ indices. K_{Pl} and K_{Il} are the most important inputs which effect classification results of inputs.

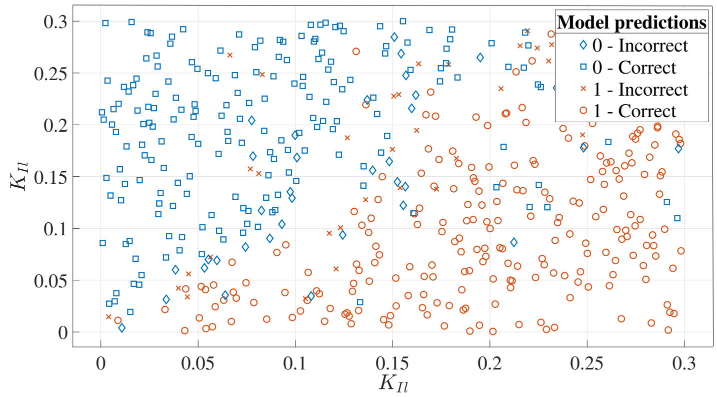

Figure 8. The prediction results of trained classifier in the K_{Pl}, K_{Il} plane. Blue symbols mean that the corresponding sample belongs to class 0, and red symbols mean that the corresponding sample belongs to class 1. Class 1 indicates that corresponding samples belongs to the SOS, and the meaning of class 0 is reversed. The classification accuracy of the trained classifier is 84%.

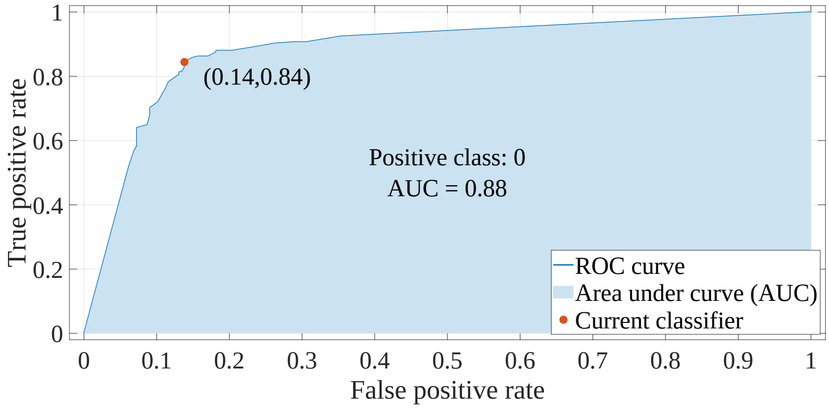

Figure 9. The receiver operating characteristic (ROC) curve of the trained classifier. The area under curve equals to 0.88. The false positive rate is 14% and the true positive rate is 84%. The trained classifier is accurate and reliable.

Table 3 presents the RMSEs of five performance measures by trained regional SUMOs through the RSMT. During the MATLAB/SIMULINK simulation, the solver is ode1 (Euler), and fixed-step size is 1×10^{-3}, so t_{a} is represented by the step number. Kriging, PCE, and PCK perform better than the RBFNN and GRNN. Adopting Kriging to tune PID controllers, the MATLAB function “paretosearch” and “gamultiobj” are used to find Pareto fronts. The run times of the simulation model and regional Kriging are denoted by “Simulation model time” and “Regional Kriging time”, respectively, in Table 4. The process of optimization is greatly accelerated. The cost of searching by Kriging is substantially lower than that of searching by the simulation model.

Table 3. The accuracy of regional SUMOs through the RSMT in $K \in [0, 0.3]$.

	$\sigma_f{}^a$	$\sigma_l{}^a$	$f_{st}{}^a$	$l_{st}{}^a$	$t_a{}^a$
Kriging	5.41	19.82	16.80	26.59	1.63×10^4
PCE	5.65	21.52	17.28	32.23	2.11×10^4
PCK	5.84	17.77	19.51	31.25	1.73×10^4
GRNN	15.15	65.43	15.74	28.88	3.26×10^4
RBFNN	18.84	37.34	37.33	155.45	6.36×10^4

[a] σ_f, σ_l, f_{st}, l_{st} and t_a denote the RMSE of them getting from Kriging, PCE, PCK, the RBFNN, and the GRNN, respectively. Every SUMO is accurate, but Kriging, PCE, and PCK perform better than the RBFNN and GRNN.

Table 4. Run time comparison of optimization by the actual model and by Kriging.

Function Name	**Gamultiobj**	**Paretosearch**
Number of solutions	70	60
Regional Kriging time (s)	8.62×10^3	7.03×10^2
Simulation model time (s)	1.59×10^5	3.44×10^4

Regional Kriging effectively shortens optimization time.

We examine solutions of “paretosearch” which meets the constraint conditions of optimization mentioned in Section 4.1, i.e., $|\Delta f_{st}| < 3m$ and $|\Delta l_{st}| < 3m$. Selected Pareto solutions are evaluated using the technique for order of preference by similarity to ideal solution (TOPSIS) [38]. The solutions of Kriging are also entered into the simulation model to obtain results for evaluation. Then, simulation results from different sources are evaluated by TOPSIS. Table 5 shows the Pareto solutions of different sources. According to Table 5, the scores of the two sources are similar to each other. It means that optimal solutions of regional Kriging are also able to be optimal solutions of the simulation model. Adopting the solution with highest score, i.e., $[0.300, 0.0001, 0.300, 0.291, 0.164, 0.145]$, Figure 10 shows the trajectories of UAVs formation when the heading angle of the leader ϕ_L changes according to the sine function. When the heading angle of the leader UAV changes, the follower UAV can timely adjust to maintain the formation.

Table 5. The TOPSIS score of selected Pareto solutions.

K_{Pl}	K_{Il}	K_{Dl}	K_{Pf}	K_{If}	K_{Df}	**Score (10^{-1})**	**Source** [a]
0.300	0.0001	0.300	0.291	0.164	0.145	2.295	regional Kriging
0.191	0.0001	0.300	0.290	0.0001	0.300	2.286	regional Kriging
0.211	0.042	0.173	0.089	0.136	0.286	1.324	simulation model
0.019	0.0001	0.131	0.122	0.0009	0.131	1.286	simulation model
0.300	0.132	0.263	0.254	0.0009	0.263	1.121	simulation model
0.299	0.014	0.070	0.117	0.300	0.145	0.898	regional Kriging
0.299	0.014	0.300	0.117	0.300	0.145	0.789	regional Kriging

[a] Regional Kriging: regional Kriging gets the solution; simulation model: the simulation model gets the solution. The Pareto solutions of regional Kriging are reliable, and regional Kriging successfully replaces the simulation model.

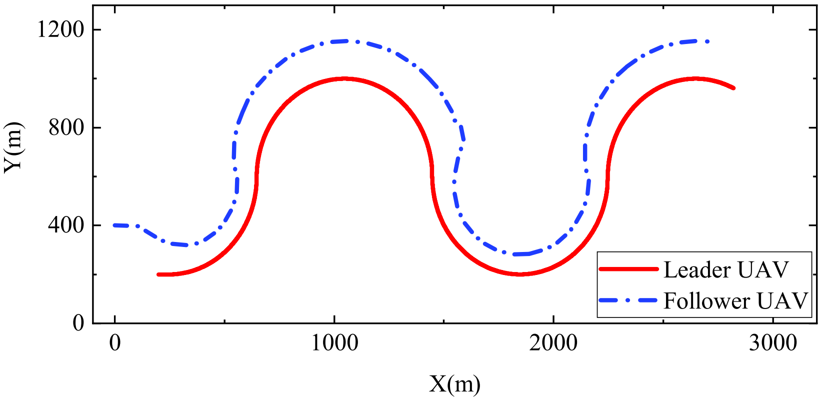

Figure 10. Trajectories of UAVs formation. The heading angle of the leader ϕ_L changes according to the sine function. PID controllers adopt the solution with the highest score. i.e., $\boldsymbol{K} = [0.300, 0.0001, 0.300, 0.291, 0.164, 0.145]$

With the RSMT, regional Kriging accurately replaces the simulation model to find optimal solutions. Abandoning useless information does not affect searching optimal parameters. The RSMT can accelerate the optimization process with high accuracy and low computational time simultaneously.

5. Conclusion and Discussion

To accelerate the process of tuning PID controllers, this work proposes the RSMT based on analyzing the regional information entropy relationship. The RSMT discards redundant information to construct the regional SUMO. A classifier is introduced to define the boundary of the regional SUMO. According to calculation results, the RSMT significantly improves the accuracy of SUMOs and reduces computational expense. The results verify the theoretical analysis of the regional information entropy relationship. To corroborate the reliability of the RSMT, the Pareto fronts are searched by regional SUMOs and the simulation model, respectively. It is found that different Pareto fronts are similar to each other. The RSMT reduces the run time of parameter optimization by one order of magnitude, and it gets reliable optimization results.

The RSMT can tune PID controllers with high efficiency and accuracy, and be available for various types of SUMOs. In the process of tuning PID controllers, the RSMT significantly reduces the singular-point interference, improves the accuracy of SUMOs, and reduces computational expense. Not only limited optimization of the UAV formation, but the RSMT can also be extended for tuning PID controllers in various systems because SUMOs only concern inputs and outputs of systems. In future research, we prone to solve the application problem of the RSMT in high-dimensional situations, which may be solved by combining sequential sampling and dimensionality reduction technology.

Author Contributions: conceptualization, B.W. and Y.L.; data curation, B.W. and Y.L.; formal analysis, B.W., X.D., and L.Y.; funding acquisition, X.D., and L.Y.; investigation: B.W., D.X., and L.Y.; methodology, B.W., X.D., and L.Y.; project administration, B.W., X.D., and L.Y.; resources, B.W., X.D., and L.Y.; software, B.W., X.D., and L.Y.; supervision, X.D. and J.C.; validation, J.C.; visualization, B.W., D.J., andL.Y.; writing–original draft, B.W., X.D., and L.Y.; writing–review & editing, Y.L., D.J., and J.C.; All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Numerical Wind Tunnel Project (NNW2019ZT7-B23) and the National Natural Science Foundation of China (No. 11771450).

Acknowledgments: The authors are grateful to Mingze Qi, Peng Li and Qing Xu for their help with this paper.

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

Abbreciations

GRNN	generalized regression neural network
L-F	leader–follower
LQR	linear quadratic regulator
SUMO	surrogate model
PCE	polynomial chaos expansions
PCK	polynomial chaos Kriging
PDF	probability distribution function
TOPSIS	technique for order of preference by similarity to ideal solution
PID	proportional-integral-derivative
RMSE	root mean squared error
SOF	space of failure
SOS	space of success
RSMT	regional surrogate model technique
RBFNN	radial basis function neural network
UAV	unmanned aerial vehicle

Appendix A. The Design of Single UAV

Appendix A.1. Inner-Loop Controller Design

Following [39,40], the inner-loop controller is designed as follows. The linearized model of UAV is

$$\begin{cases} \dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{u} \\ \boldsymbol{y} = C\boldsymbol{x} \end{cases}, \tag{A1}$$

where $\boldsymbol{x}$ is the state vector and $\boldsymbol{x} = [V, \eta, \tau, \epsilon, \beta, p, r, \zeta]^T$ represent forward velocity, attack angle, pitch rate, pitch angle, side-slip angle, roll rate, yaw rate, and yaw angle, respectively. $\boldsymbol{y}$ is the output vector, $\boldsymbol{y} = [V, \zeta - \beta]^T$; $\boldsymbol{u}$ is the control input vector, $\boldsymbol{u} = [\delta_e, \delta_T, \delta_a, \delta_r]^T$, which represent the deflections of elevator, throttle, aileron, and rudder, respectively. A, B, and C are the system matrix, the input matrix, and the output matrix, respectively.

The aim of inner-loop control is minimizing the difference between the UAV state and commands. The difference is represented by the cost function J [39], which is given by

$$J = \frac{1}{2}\int_0^{\infty} (\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{u}^T R \boldsymbol{u}) d\boldsymbol{x}, \tag{A2}$$

where Q, R are the weighting matrices. $\boldsymbol{u}$ is the output feedback, $\boldsymbol{u} = D\boldsymbol{y}$, where D is the feedback gain matrix. The UAV state equation can be written as $\dot{\boldsymbol{x}} = (A + BDC)\boldsymbol{x}$. D is obtained by $D = R^{-1}B^T P$, where R is user-defined and P is obtained by solving the algebraic Riccati equation [40]:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \tag{A3}$$

In this paper, R is defined as a planar unit matrix.

Appendix A.2. The System Matrices of a Single UAV

According to [26], the fixed-wing UAV model is given as follows:

$$A_{\mathrm{f}} = \begin{bmatrix} -0.334 & -2.9770 & 0 & -9.81 \\ -0.0016 & -4.1330 & 0.9800 & 0 \\ 0.0077 & -140.20 & -4.435 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}, A_{\mathrm{l}} = \begin{bmatrix} -0.7320 & 0.0143 & -0.9960 & -0.0706 \\ -893.00 & -9.0590 & 2.0440 & 0 \\ 101.673 & 0.0186 & -1.2830 & 0 \\ 0 & 0 & 1.0000 & 0 \end{bmatrix}, \tag{A4}$$

$$B_{\mathrm{f}} = \begin{bmatrix} -1.0750 & 0.2453 \\ 0.3470 & -4.1330 \\ -140.22 & 0 \\ 0 & 0 \end{bmatrix}, B_{\mathrm{l}} = \begin{bmatrix} 0 & 0.2440 \\ 328.653 & -308.498 \\ 47.528 & 102.891 \\ 0 & 0 \end{bmatrix}, \tag{A5}$$

$$A = \begin{bmatrix} A_{\mathrm{f}} & 0 \\ 0 & A_{\mathrm{l}} \end{bmatrix}, B = \begin{bmatrix} B_{\mathrm{f}} & 0 \\ 0 & B_{\mathrm{l}} \end{bmatrix}, \tag{A6}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \end{bmatrix}. \tag{A7}$$

Appendix B. Regional Information Entropy Relationship in the Case of the *t*-Distribution

Because I_{Suss} accounts for only a small part of the simulation output, it is assumed that there is a *t*-distribution in I_{fail} and uniform distribution in I_{succ}. The PDF of the *t*-distribution [41] is given by

$$p_{\mathrm{t}}(x) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad x \in (-\infty, +\infty), \tag{A8}$$

where $\Gamma(\cdot)$ is the gamma function and ν is the number of degrees of freedom. The entropy of the *t*-distribution [41] is

$$S_{\mathrm{t}} = \frac{\nu+1}{2}\left[\psi\left(\frac{1+\nu}{2}\right) - \psi\left(\frac{\nu}{2}\right)\right] + \ln\left[\sqrt{\nu} Beta\left(\frac{\nu}{2}, \frac{1}{2}\right)\right], \tag{A9}$$

where $Beta(\cdot)$ is the beta function and $\psi(\cdot)$ is the digamma function. Figure A1 shows the relationship between ν and S_{t}. When ν is larger than 40, S_{t} tends to be stable.

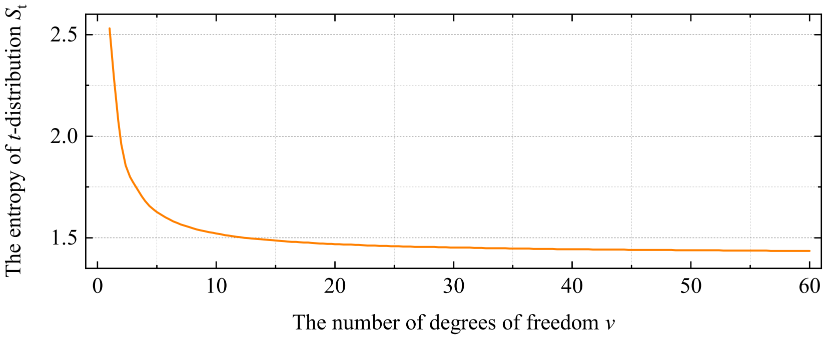

Figure A1. The relationship between ν and S_{t} for the *t*-distribution. When ν is larger than 40, S_{t} tends to be stable.

For ease of presentation, it is assumed that $x > 0$. The PDF of x is given by

$$p(x) = \begin{cases} P_{\mathrm{succ}}/(b-a) & 0 \le x \le b \\ \frac{P_{\mathrm{fail}}\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} & b < x \le x_{max} \end{cases}. \tag{A10}$$

S_{succ} and S_{fail} are given by

$$S_{\mathrm{succ}} = -P_{\mathrm{succ}} \ln \frac{P_{\mathrm{succ}}}{b}, \tag{A11}$$

$$\begin{aligned} S_{\text{fail}} &= -\int_{b}^{x_{\max}} P_{\text{fail}} p\left(x\right) \ln(P_{\text{fail}} p\left(x\right)) dx \\ &= -P_{\text{fail}} \left(\int_{b}^{x_{\max}} p\left(x\right) \ln P_{\text{fail}} dx + \int_{b}^{x_{\max}} p\left(x\right) \ln p\left(x\right) dx \right) \\ &= -P_{\text{fail}} \left(P_{\text{fail}} \ln P_{\text{fail}} + S_{\text{t}}\right). \end{aligned} \tag{A12}$$

Setting $b = 5$, $x_{\max} = 1000$, and $\nu = 50$, the relationship between W and P_{fail} is given by Case 3 in Figure 3. The relationship is similar to that in Case 1. With the rise of P_{fail}, W quickly decreases and the useful information is concealed. The type of distribution does not change the information entropy relationship between different intervals.

Appendix C. Brief Introduction to Kriging, PCE, PCK, the RBFNN, and the GRNN

Appendix C provides a brief introduction to Kriging, PCE, PCK, the RBFNN, and the GRNN. Kriging, PCE, and PCK are calculated using the UQlab toolbox[42]. $\boldsymbol{g}$ is the input vector, $\boldsymbol{g} \in \mathbb{R}^{M\times 1}$; $\mathcal{M}$ is the output variable.

Appendix C.1. Kriging

Kriging (also known as Gaussian process modelling) performs well in the local system, which is given by [42]

$$\mathcal{M}\left(\boldsymbol{g}\right) = \boldsymbol{\beta}^{T} \boldsymbol{f}\left(\boldsymbol{g}\right) + \sigma^{2} Z\left(\boldsymbol{g}, h\right). \tag{A13}$$

The first term $\boldsymbol{\beta}^{T} \boldsymbol{f}\left(\boldsymbol{g}\right)$ is the average of the Gaussian process, where $\boldsymbol{f}\left(\boldsymbol{g}\right)$ denotes arbitrary functions, and their corresponding coefficients are $\boldsymbol{\beta}$. σ^2 is the variance of the Gaussian process; $Z\left(\boldsymbol{g}, h\right)$ is the Gaussian process where h is the underlying probability space.

Appendix C.2. PCE

$\boldsymbol{G} \in \mathbb{R}^{M}$ is a random vector with independent components, and the PCE [42] of $M(\boldsymbol{G})$ is given by

$$\mathcal{M}(\boldsymbol{G}) = \sum_{\alpha \in \mathbb{N}^{M}} z_{\alpha} \Psi_{\alpha}(\boldsymbol{G}), \tag{A14}$$

where $\Psi_{\alpha}(\boldsymbol{G})$ is multivariate polynomials orthonormal; $\alpha \in \mathbb{N}^{M}$ is a multi-index identifying the components of Ψ_{α}; and $z_{\alpha} \in \mathbb{R}$ is the corresponding coefficients of Ψ_{α}. The truncated PCE, which is given by [42], is adopted in practice.

$$\mathcal{M}(\boldsymbol{G}) \approx \mathcal{M}^{PC}(\boldsymbol{G}) = \sum_{\alpha \in \mathcal{A}} z_{\alpha} \Psi_{\alpha}(\boldsymbol{G}), \tag{A15}$$

where $\mathcal{A} \subset \mathbb{N}^{M}$ is the set of selected multi-indices of multivariate polynomials. PCE is computationally superior to traditional methods in most cases, such as Monte-Carlo-based methods.

Appendix C.3. PCK

Because PCE and Kriging perform well in the global and local system, respectively, PCK is proposed as a combination of universal Kriging and PCE. PCK [42] is given by

$$\mathcal{M}(\boldsymbol{g}) \approx \mathcal{M}^{\text{PCK}}(\boldsymbol{g}) = \sum_{\alpha \in \mathcal{A}} z_{\alpha} \Psi_{\alpha}(\boldsymbol{G}) + \sigma^{2} Z(\boldsymbol{g}, h). \tag{A16}$$

The first term is the trend of PCK; the second term is the same as Kriging. PCK is more efficient than PCE and Kriging.

Appendix C.4. The RBFNN

The RBFNN overcomes the local minimum problem and has a good nonlinear fitting ability [43]. There are an input layer, a hidden layer, and an output layer in the RBFNN. If there is only one output, the RBFNN [43] can be given as

$$\mathcal{M}(\boldsymbol{g}, w) = \sum_{k=1}^{S1} w_{1k} BF\left(\|\boldsymbol{g} - \boldsymbol{c_k}\|_2\right), \tag{A17}$$

where $\boldsymbol{c_k} \in \mathbb{R}^{M \times 1}$ is the RBF centers in the input layer. $BF(\cdot)$ is the basis function that exists in the hidden layer. $\| \cdot \|_2$ denotes the Euclidean norm between $\boldsymbol{g}$ and $\boldsymbol{c_k}$; $S1$ is the number of neurons (and centers). w_{1k} devotes the weights of the output in the output layer.

Appendix C.5. The GRNN

As a one-pass algorithm, the GRNN contains four layers: an input layer, a pattern layer, a summation layer, and an output layer. The training set is $\{(\boldsymbol{g}_i, \mathcal{M}_i) | i = 1, \cdots, N\}$. $f(\boldsymbol{g}, \mathcal{M})$ is the joint PDF of $\boldsymbol{g}$ and $\mathcal{M}$. The conditional mean of $\mathcal{M}$ on $\boldsymbol{g}$ [44] is

$$E[\mathcal{M}|\boldsymbol{g}] = \frac{\int_{-\infty}^{\infty} \mathcal{M} f(\boldsymbol{g}, \mathcal{M}) d\mathcal{M}}{\int_{-\infty}^{\infty} f(\boldsymbol{g}, \mathcal{M}) d\mathcal{M}}. \tag{A18}$$

The estimated joint PDF can be written as [44]

$$\hat{f}(\boldsymbol{g}, \mathcal{M}) = \frac{1}{N(2\pi)^{(M+1)/2} \sigma^{M+1}} \times \sum_{i=1}^{N} \exp\left[-\frac{(\boldsymbol{g} - \boldsymbol{g}_i)(\boldsymbol{g} - \boldsymbol{g}_i)^T}{2\sigma^2}\right] \exp\left[-\frac{(\mathcal{M} - \mathcal{M}_i)^2}{2q^2}\right], \tag{A19}$$

where q is a user-defined smoothness parameter. Combining Equations (A18) and (A19), the conditional mean of $\mathcal{M}$, which is thought to be equal to $\mathcal{M}$, can be given by [44]

$$\mathcal{M} = E[\mathcal{M}|\boldsymbol{g}] = \frac{\sum_{i=1}^{N} \mathcal{M}_i \exp\left[-\frac{(\boldsymbol{g}-\boldsymbol{g}_i)(\boldsymbol{g}-\boldsymbol{g}_i)^T}{2\sigma^2}\right]}{\sum_{i=1}^{N} \exp\left[-\frac{(\boldsymbol{g}-\boldsymbol{g}_i)(\boldsymbol{g}-\boldsymbol{g}_i)^T}{2\sigma^2}\right]}. \tag{A20}$$

As a kind of RBFNN, the GRNN has a better approximation capability and learning rate than the traditional RBFNN. Moreover, the GRNN performs well in dealing with small samples and unstable data.

References

1. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
2. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tut.* **2015**, *18*, 1123–1152. [CrossRef]
3. Ang, K.H.; Chong, G.; Li, Y. PID control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **2005**, *13*, 559–576.
4. López-Estrada, F.R.; Ponsart, J.C.; Theilliol, D.; Zhang, Y.; Astorga-Zaragoza, C.M. LPV model-based tracking control and robust sensor fault diagnosis for a quadrotor UAV. *J. Intell. Robot. Syst.* **2016**, *84*, 163–177. [CrossRef]
5. Guzmán-Rabasa, J.A.; López-Estrada, F.R.; González-Contreras, B.M.; Valencia-Palomo, G.; Chadli, M.; Pérez-Patricio, M. Actuator fault detection and isolation on a quadrotor unmanned aerial vehicle modeled as a linear parameter-varying system. *Meas. Control* **2019**, *52*, 1228–1239. [CrossRef]
6. Walter, V.; Staub, N.; Franchi, A.; Saska, M. Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2637–2644. [CrossRef]

7. Guerrero-Sánchez, M.E.; Hernández-González, O.; Lozano, R.; García-Beltrán, C.D.; Valencia-Palomo, G.; López-Estrada, F.R. Energy-Based Control and LMI-Based Control for a Quadrotor Transporting a Payload. *Mathematics* **2019**, *7*, 1090. [CrossRef]
8. Song, Y.; Cheng, Q.S.; Koziel, S. Multi-Fidelity Local Surrogate Model for Computationally Efficient Microwave Component Design Optimization. *Sensors* **2019**, *19*, 3023. [CrossRef]
9. Kudinov, Y.; Kolesnikov, V.; Pashchenko, F.; Pashchenko, A.; Papic, L. Optimization of fuzzy PID controller's parameters. *Procedia Comput. Sci.* **2017**, *103*, 618–622. [CrossRef]
10. Kleijnen, J.P.C. Regression and Kriging metamodels with their experimental designs in simulation: A review. *Eur. J. Oper. Res.* **2017**, *256*, 1–16. [CrossRef]
11. Sudret, B. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **2008**, *93*, 964–979. [CrossRef]
12. Kersaudy, P.; Sudret, B.; Varsier, N.; Picon, O.; Wiart, J. A new surrogate modeling technique combining Kriging and polynomial chaos expansions—Application to uncertainty analysis in computational dosimetry. *J. Comput. Phys.* **2015**, *286*, 103–117. [CrossRef]
13. Li, X.; Gong, C.; Gu, L.; Gao, W.; Jing, Z.; Su, H. A sequential surrogate method for reliability analysis based on radial basis function. *Struct. Saf.* **2018**, *73*, 42–53. [CrossRef]
14. Park, J.; Kim, K. Meta-modeling using generalized regression neural network and particle swarm optimization. *Appl. Soft Comput.* **2017**, *51*, 354–369. [CrossRef]
15. Son, S.H.; Choi, B.L.; Jin, W.J.; Lee, Y.G.; Kim, C.W.; Choi, D.H. Wing design optimization for a long-endurance UAV using FSI analysis and the Kriging method. *Int. J. Aeronaut. Space Sci.* **2016**, *17*, 423–431. [CrossRef]
16. Joo, H.; Hwang, H.Y. Surrogate Aerodynamic Model for Initial Sizing of Solar High-Altitude Long-Endurance UAV. *J. Aerosp. Eng.* **2017**, *30*, 04017064. [CrossRef]
17. Zhe, Z.; Guo, H.; Ma, J. Aerodynamic layout optimization design of a barrel-launched UAV wing considering control capability of multiple control surfaces. *Aerosp. Sci. Technol.* **2019**, *93*, 105297. [CrossRef]
18. Ali, M.M.; Abdullah, S.; Osman, D. Controllers optimization for a fluid mixing system using metamodeling approach. *Int. J. Simul. Model* **2009**, *8*, 48–59.
19. Ab Malek, M.; Ali, M. Evolutionary tuning method for PID controller parameters of a cruise control system using metamodeling. *Model. Simul. Eng.* **2009**, *2009*, 234529. [CrossRef]
20. Faruq, A.; Abdullah, S.S.B.; Shah, M.F.N. Optimization of an intelligent controller for an unmanned underwater vehicle. *Telkomnika* **2011**, *9*, 245. [CrossRef]
21. Lü, W.; Zhu, Y.; Huang, D.; Jiang, Y.; Jin, Y. A new strategy of integrated control and on-line optimization on high-purity distillation process. *Chin. J. Chem. Eng.* **2010**, *18*, 66–79. [CrossRef]
22. Matinnejad, R.; Nejati, S.; Briand, L.; Brcukmann, T. MiL testing of highly configurable continuous controllers: Scalable search using surrogate models. In Proceedings of the 29th ACM/IEEE international conference on Automated software engineering, Västerås, Sweden, 15–19 September 2014; pp. 163–174.
23. Pan, I.; Das, S. Kriging based surrogate modeling for fractional order control of microgrids. *IEEE Trans. Smart Grid* **2014**, *6*, 36–44. [CrossRef]
24. Guerrero, J.; Cominetti, A.; Pralits, J.; Villa, D. Surrogate-based optimization using an open-source framework: the bulbous bow shape optimization case. *Math. Comput. Appl.* **2018**, *23*, 60. [CrossRef]
25. Faruq, A.; Shah, M.F.N.; Abdullah, S.S. Multi-objective optimization of PID controller using pareto-based surrogate modeling algorithm for MIMO evaporator system. *Int. J. Electr. Comput. Eng.* **2018**, *8*, 556–565. [CrossRef]
26. Xu, Q.; Yang, H.; Jiang, B.; Zhou, D.; Zhang, Y. Fault tolerant formations control of UAVs subject to permanent and intermittent faults. *Intell. Robot. Syst.* **2014**, *73*, 589–602. [CrossRef]
27. Zhang, B.; Sun, X.; Liu, S.; Deng, X. Adaptive Differential Evolution-based Receding Horizon Control Design for Multi-UAV Formation Reconfiguration. *Int. J. Control Autom. Syst.* **2019**, *17*, 3009–3020. [CrossRef]
28. Park, C.; Cho, N.; Lee, K.; Kim, Y. Formation flight of multiple uavs via onboard sensor information sharing. *Sensors* **2015**, *15*, 17397–17419. [CrossRef]
29. Li, P.; Yu, X.; Peng, X.; Zheng, Z.; Zhang, Y. Fault-tolerant cooperative control for multiple UAVs based on sliding mode techniques. *Sci. China-Inf. Sci.* **2017**, *60*, 070204. [CrossRef]
30. Shamsuzzoha, M.; Skogestad, S. The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning. *J. Process Control* **2010**, *20*, 1220–1234. [CrossRef]

31. Pan, I.; Goncalves, G.; Batchvarov, A.; Liu, Y.; Liu, Y.; Sathasivam, V.; Yiakoumi, N.; Mason, L.; Matar, O. Active learning methodologies for surrogate model development in CFD applications. *Bull. Am. Phys. Soc.* **2019**, *64*. [CrossRef]
32. Kim, S.W.; Melby, J.A.; Nadal-Caraballo, N.C.; Ratcliff, J. A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling. *Nat. Hazards* **2015**, *76*, 565–585. [CrossRef]
33. Zischg, A.P.; Felder, G.; Mosimann, M.; Röthlisberger, V.; Weingartner, R. Extending coupled hydrological-hydraulic model chains with a surrogate model for the estimation of flood losses. *Environ. Modell. Softw.* **2018**, *108*, 174–185. [CrossRef]
34. Oladyshkin, S.; Nowak, W. The Connection between Bayesian Inference and Information Theory for Model Selection, Information Gain and Experimental Design. *Entropy* **2019**, *21*, 1081. [CrossRef]
35. Hu, Y.J.; Ku, T.H.; Jan, R.H.; Wang, K.; Tseng, Y.C.; Yang, S.F. Decision tree-based learning to predict patient controlled analgesia consumption and readjustment. *BMC Med. Inform. Decis. Mak.* **2012**, *12*, 131. [CrossRef] [PubMed]
36. Zhang, J.; Chowdhury, S.; Messac, A. An adaptive hybrid surrogate model. *Struct. Multidiscip. Optim.* **2012**, *46*, 223–238. [CrossRef]
37. Thanh, H.L.N.N.; Hong, S.K. Completion of collision avoidance control algorithm for multicopters based on geometrical constraints. *IEEE Access* **2018**, *6*, 27111–27126. [CrossRef]
38. Behzadian, M.; Otaghsara, S.K.; Yazdani, M.; Ignatius, J. A state-of-the-art survey of TOPSIS applications. *Expert Syst. Appl.* **2012**, *39*, 13051–13069. [CrossRef]
39. Li, Y.; Chen, C.; Chen, W. Research on longitudinal control algorithm for flying wing UAV based on LQR technology. *Int. J. Smart Sens. Intell. Syst.* **2013**, *6*, 2155–2181. [CrossRef]
40. Rahimi, M.R.; Hajighasemi, S.; Sanaei, D. Designing and simulation for vertical moving control of UAV system using PID, LQR and Fuzzy Logic. *Int. J. Elec. Comput. Eng.* **2013**, *3*, 651.
41. Student’s t-Distribution. Available online: https://wikivisually.com/wiki/Student%27s_t-distribution (accessed on 4 May 2020).
42. Marelli, S.; Sudret, B. UQLab: A Framework for Uncertainty Quantification in Matlab. In *Vulnerability, Uncertainty, and Risk*; American Society of Civil Engineers: Reston, VA, USA, 2014; pp. 2554–2563.
43. Nguyen, N.P.; Hong, S.K. Fault-tolerant control of quadcopter UAVs using robust adaptive sliding mode approach. *Energies* **2019**, *12*, 95. [CrossRef]
44. Wang, Y.; Yin, D.Q.; Yang, S.; Sun, G. Global and local surrogate-assisted differential evolution for expensive constrained optimization problems with inequality constraints. *IEEE Trans. Cybern.* **2018**, *49*, 1642–1656. [CrossRef] [PubMed]

© 2020 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 (http://creativecommons.org/licenses/by/4.0/).