Noname manuscript No.
(will be inserted by the editor)

Online Adaptation for Autonomous Unmanned Systems Driven by Requirements Satisfaction Model (Appendix)

Yixing Luo[1], Yuan Zhou[2], Zhi Jin[1], Haiyan Zhao[1], Tianwei Zhang[2], Yang Liu[2], Danny Barthaud[3], Yijun Yu[3]

[1] Key Lab. of High Confidence of Software Technologies (MoE), Department of Computer Science and Technology, School of EECS, Peking University, Beijing, China.
[2] School of Computer Science and Engineering, Nanyang Technological University, Singapore.
[3] School of Computing and Communications, The Open University, Milton Keynes, UK.

The date of receipt and acceptance will be inserted by the editor

1 UAV Delivery Scenario

1.1 Requirements Satisfaction Modeling

In this part, we will illustrate the detailed requirement satisfaction functions we used in the UAV delivery scenario based on the requirement listed in Table 1.

- Safety: The indicator to evaluate the safety requirement is the collision risk of UAV during the flight. Supposing that the obstacles detected by the UAV at time instant k is $\mathcal{O}_k$, while the current state of UAV is s_k. Thus, the QM of safety is $\mathcal{X}_{S_{o,k}} = \frac{\|\boldsymbol{x}_k-\boldsymbol{x}_o\|_2-r_a-r_o}{D_o}, \forall o \in \mathcal{O}_k$. Such that the average distance between UAV and the center of obstacle reflects the safety risk.

$$DS^2(\mathcal{X}_{S_{o,k}}) = \begin{cases} 1, & \mathcal{X}_{S_{o,k}} \geq 1 \\ 0, & \mathcal{X}_{S_{o,k}} < 0 \\ \mathcal{X}_{S_{o,k}}, & otherwise \end{cases}$$

- Timelines: The total traveling time from time instant i to j is denoted as $\xi_{ij} = \sum_{k=i}^{j-1} \frac{\|\boldsymbol{x}_{k+1}-\boldsymbol{x}_k\|_2}{\boldsymbol{v}_k}$. The indicator of timeliness is $\mathcal{X}_\xi = \xi_{0T}$, the degree of satisfaction of the timeliness requirement of the whole trajectory DS_ξ is:

$$DS^1(\mathcal{X}_\xi) = \begin{cases} 1, & \mathcal{X}_\xi \leq \Delta_t \\ \dfrac{\Delta - \mathcal{X}_\xi}{\Delta - \Delta_t}, & \Delta_t < \mathcal{X}_\xi \leq \Delta \\ 0, & \mathcal{X}_\xi > \Delta \end{cases}$$

- Accuracy: The average quality of the information collected during the mission is denoted as $\mathcal{X}_\varphi = \frac{1}{\xi_{0T}} \sum_{k=0}^{T-1} \|\boldsymbol{\omega}\| \tau$, the degree of satisfaction is DS_φ is:

$$DS^2(\mathcal{X}_\varphi) = \begin{cases} 1, & \mathcal{X}_\varphi \geq A_t \\ \dfrac{\mathcal{X}_\varphi - A}{A_t - A}, & A \leq \mathcal{X}_\varphi < A_t \\ 0, & \mathcal{X}_\varphi < A \end{cases}$$

- Energy-saving: The total energy consumption from time instant i to j is denoted as $e_{ij} = \sum_{k=i}^{j-1} \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2 + \eta_1 \cdot \|\boldsymbol{v}_{k+1} - \boldsymbol{v}_k\|_2 + \eta_2 \cdot \|\boldsymbol{\omega}_k\| \tau$. The indicator of energy consumption is $\mathcal{X}_e = e_{0T}$, the degree of satisfaction of energy requirement DS_e is:

$$DS^1(\mathcal{X}_e) = \begin{cases} 1, & \mathcal{X}_e \leq E_t \\ \dfrac{E - \mathcal{X}_e}{E - E_t}, & E_t < \mathcal{X}_e \leq E \\ 0, & \mathcal{X}_e > E \end{cases}$$

1.2 Experiment Results

We show the details of the experimental results in the UAV delivery scenarios discussed in the paper.

1.2.1 Scalability To demonstrate the scalability of `Captain` with respect to different environments, we further simulated the UAV case on two selected real urban environments from the open building dataset of Portland in USA [1]. We used ArcGIS map to set up a 3D model based on the method in [2]. Their original spaces are $500 \times 500 \times 100 m^3$ and $10^3 \times 10^3 \times 100\ m^3$, and compressed into $50 \times 50 \times 10\ m^3$ and $100 \times 100 \times 10\ m^3$, respectively. In the dataset, we can also obtain the longitude and latitude of the center of each building, as well as its average height and building types (i.e., industrial and commercial buildings, houses and apartments for living). The buildings for industrial and commercial use are viewed as obstacles, while houses and apartments are

Correspondence to: Prof. Zhi Jin

Table 1: Adaptation results of `Captain` for different scales of environment

Scale		*Accuracy* [%]	*Traveling Time* [s]	*Energy Consumption* [unit]	*Safety Risk*	*Privacy Risk*	*Real-time Performance*
50	$\mathcal{S}$	90	60	100	0	0	Adaptation Rate
	$\mathcal{H}$	80	90	150	1	1	6/120
	$\mathcal{X}$	90	60.00	105.57	0	0	Overhead
	$\mathcal{DS}$	100%	100%	88.85%	100%	100%	Avg. 0.066s Std. 0.139s
100	$\mathcal{S}$	90	90	200	0	0	Adaptation Rate
	$\mathcal{H}$	80	150	300	1	1	3/203
	$\mathcal{X}$	90	101.50	210.94	0	0.1534	Overhead
	$\mathcal{DS}$	100%	80.83%	89.06%	100%	99.98%	Avg. 0.132s Std. 0.180s

viewed as private regions. We used ArcGIS map to set up a 3D model according to the method introduced in [2]. As shown in Fig. 1, the process of real urban scenarios modeling is explained. We set the range of longitude and latitude of working space on the ArcGIS map, and all the building are marked as blue (Fig. 1(a)). Then the ArcGIS map is turned into a binary map. Thus a 3D model of the urban scene is set up with the scale of $500m \times 500m \times 100m$ in Fig. 1(c).

In each case, the flight task of UAV is to travel from the position $[0, 0, 0]$ to the destination $[49, 49, 0]$ and $[99, 99, 0]$ respectively, within the budget of accuracy, time and energy, as well as minor safety and privacy risk. The trajectory, state transition and requirement achievement of the UAV at each time instant are shown in Figure 1(d), Figure 1(e) and Figure 1(f). So as to the scale of $1000m \times 1000m \times 100m$ in Figure 2. Table 1 summarizes the requirement adaptation results in these two settings. we find that the states generated from `Captain` and the PD controller are almost the same, indicating that the planning results of `Captain` can be translated by the PD controller and executed by the UAV effectively.

From our simulation results, in Setting 2, the flight task is completed in 60.07 s, consuming 106.27 units of energy without safety and privacy risk. In Setting 3, the task is finished in 101.5 s, consuming 210.94 units of energy. It does not have safety risk, but gets 4 points along the trajectory where the soft constraint of privacy-preserving is violated. This results in an average 99.98% requirements satisfaction along the path. `Captain` has high scalability as the two important steps *Requirements Satisfaction Checking* and *Requirements Satisfaction Optimization* are solved by SQP, which can handle large-scale optimization problems. In contrast, AMOCS-MA fails to compute a motion plan at real-time when the size of workspace increases.

2 UUV Oceanic Surveillance

This example originates from [3,4]. Rather than considering single-objective optimization for the UUV scenario in [3,4], we extended it to achieve multiple dynamic requirements under uncertainties and disturbances through `Captain`, while other configurations like sensors are kept the same as [3]. The requirements to achieve in this scenario are listed as follows:

- Scanning Distance (R_l): A segment of surface over a distance of $L_t = 100$ km is expected to be examined by the UUV within $\Delta = 10$ hours, while the threshold of surveillance distance is $L = 90$ km.
- Energy Consumption (R_e): A total amount of energy $E_t = 5.4$ MJ is expected to be consumed, while the maximum amount of energy is $E = 6$ MJ.
- Accuracy (R_φ): The accuracy of sensor measurements is targeted at $A_t = 90\%$, while the accuracy threshold is set as $A = 80\%$.

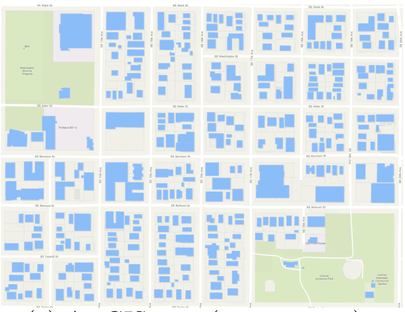
2.1 Requirements Satisfaction Modeling

The UUV is equipped with 5 sensors for ocean surveillance. The scanning time 10 hours is 360 time instance, $x_i, i \in [1, 5]$ is the portion of time the sensor i should be used during system operation in each instance. Acc_i is the accuracy of sensor i; E_i is the energy consumed by sensor; V_i is the scanning speed of sensor. q_i is portion of accuracy of sensor and p_i is for scanning speed respectively in decimals. The energy consumed is related with working accuracy and speed of sensor. The corresponding measures are listed as follows: $\mathcal{X}_l = \sum_{k=0}^{T}\sum_{i=0}^{N} x_i q_i V_i \tau$, $\mathcal{X}_e = \sum_{k=0}^{T}\sum_{i=0}^{N} x_i E_i \cdot \frac{e^{p_i+q_i}-1}{e^2-1}\tau$, and $\mathcal{X}_\varphi = \sum_{k=0}^{T}\sum_{i=0}^{N} x_i p_i Acc_i$, where $T = 360$, i.e., adaptations is performed every 100 surface measurements of the UUV state, and the time instance k incremented by $1 \sim 100$. The requirement satisfaction functions are listed as follows:

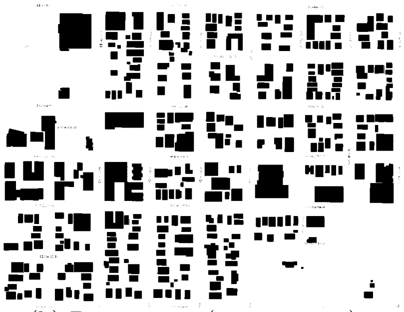
- Scanning distance: A segment of surface over a distance of $L_t = 100$ km is expected to be examined by the UUV within $\Delta = 10$ hours, while the distance threshold is $L = 90$ km.

$$\mathrm{DS}^2(\mathcal{X}_l) = \begin{cases} 1, & \mathcal{X}_l \geq L_t \\ \dfrac{\mathcal{X}_l - L}{L_t - L}, & L \leq \mathcal{X}_l < L_t \\ 0, & \mathcal{X}_l < L \end{cases}$$

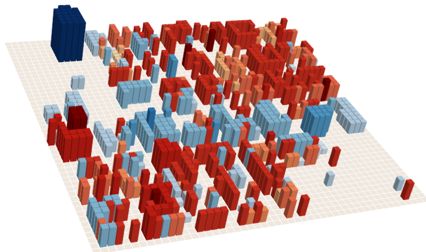
- Energy-saving: A total amount of energy $E_t = 5.4$ MJ is expected to be consumed, while the maximum



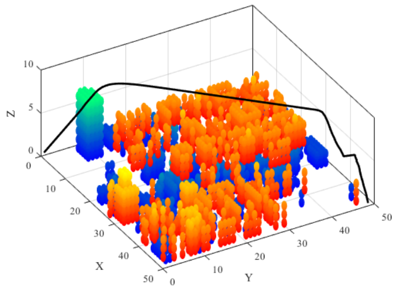
(a) ArcGIS map (50 × 50 × 10).



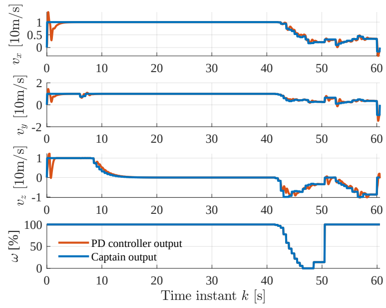
(b) Binary map (50 × 50 × 10).



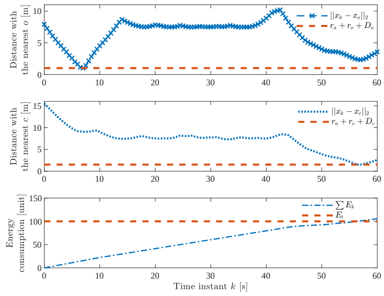
(c) Grid map (50 × 50 × 10).



(d) Flight Trajectory of UAV.

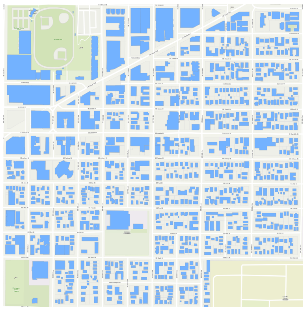


(e) State of UAV during the flight.



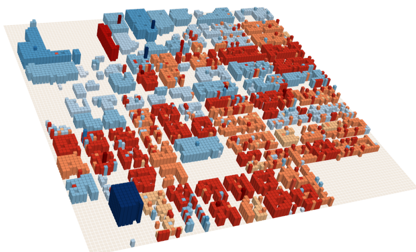
(f) Requirement achievement of UAV during the flight.

Fig. 1: Environment modeling with the scale of $500m \times 500m \times 100m$.

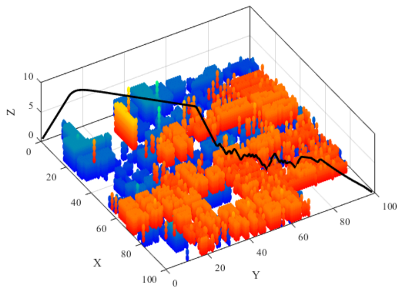


(a) ArcGIS map (100 × 100 × 10).

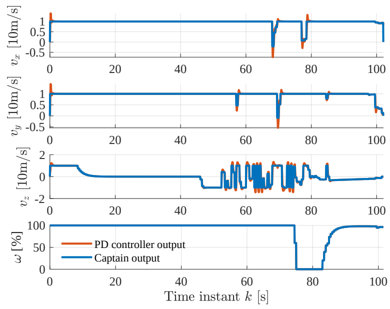
(b) Binary map (100 × 100 × 10).



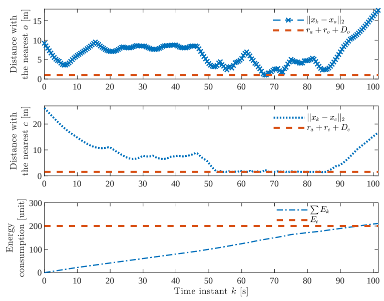
(c) Grid map (100 × 100 × 10).



(d) Flight Trajectory of UAV.

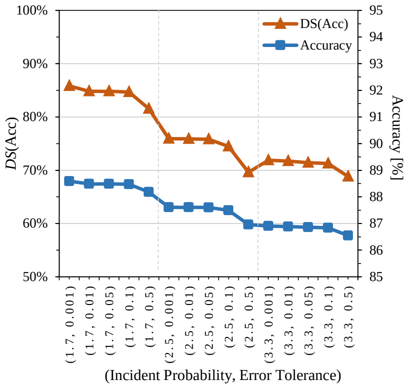


(e) State of UAV during the flight.

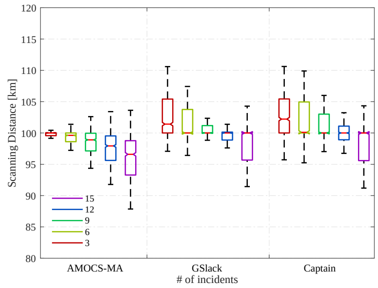


(f) Requirement achievement of UAV during the flight.

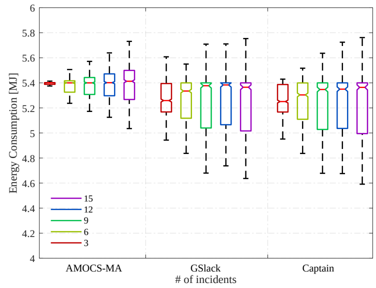
Fig. 2: Environment modeling with the scale of $1000m \times 1000m \times 100m$.



(a) Scanning accuracy with various incidents and ϵ. (`Captain`)



(b) Variance of scanning distance.



(c) Variance of energy consumption.

Fig. 3: Scanning accuracy, distance and energy consumption with various probability of incidents and violation tolerance.

amount of energy is $E = 6$ MJ.

$$\mathrm{DS}^1(\mathcal{X}_e) = \begin{cases} 1, & \mathcal{X}_e \leq E_t \\ \dfrac{E - \mathcal{X}_e}{E - E_t}, & E_t < \mathcal{X}_e \leq E \\ 0, & \mathcal{X}_e > E \end{cases}$$

– Accuracy: The accuracy of sensor measurements is targeted at $A_t = 90\%$, while the accuracy threshold is set as $A = 80\%$.

$$\mathrm{DS}^2(\mathcal{X}_\varphi) = \begin{cases} 1, & \mathcal{X}_\varphi \geq A_t \\ \dfrac{\mathcal{X}_\varphi - A}{A_t - A}, & A \leq \mathcal{X}_{\varphi_k} < A_t \\ 0, & \mathcal{X}_\varphi < A \end{cases}$$

2.2 Experiment Results

To demonstrate the generality of `Captain`, we applied it to a UUV case described in [3]. The requirement satisfaction models in this scenario are $\mathrm{DS}^1(\mathcal{X}_l)$ (scanning distance), $\mathrm{DS}^2(\mathcal{X}_e)$ (energy consumption) and $\mathrm{DS}^2(\mathcal{X}_\varphi)$ (accuracy). There are trade-offs between these requirements, e.g., when sensors (e.g., sensor 1) with a higher quality of surveillance is chosen, more energy is consumed, resulting in less distance scanned.

2.2.1 Robustness In this scenario, the performance of the three methods are compared while adding random failures to parameters of sensors, i.e., sensor accuracy, scanning speed and energy consumption. For each method, we simulated UUV motion by adding different frequencies of random disturbances at different time instants. For each frequency of disturbances, we simulated 1000 rounds and computed the average accuracy, scanning distance and energy consumption. We can see that under `Captain`, the UUV can scan a longer distance with higher accuracy and less energy consumption than AMOCS-MA and GSlack. Moreover, the motion generated by AMOCS-MA can hardly satisfy all the three requirements, while the motion from `Captain` and GSlack can satisfy the timeliness and energy requirements, only slightly violating the accuracy requirement. The reason behind this is that at the *Requirements Satisfaction Analysis* stage, the accuracy is selected as the requirement for adaptation while the other two remain as their original soft constraints. Besides, `Captain` outperforms GSlack, especially in situations with very frequent disturbances. The variance of scanning distance and energy consumption of three strategies are compared in Figure 3(b) and Figure 3(c).

Table 2: Statistics on Overhead Data.

Cases	*Approaches*	*Average* [s]	*Standard Deviation* [s]
UAV	AMOCS-MA	0.2115	0.0879
	GSlack	0.0544	0.0637
	`Captain`	***0.0811***	***0.0821***
UUV	AMOCS-MA	0.0282	0.0072
	GSlack	0.0062	0.0023
	`Captain`	***0.0081***	***0.0058***

2.2.2 Real-time Performance Finally, we analyze the computation overhead of `Captain` in generating an optimal self-adaptive plan. Table 2 shows the empirical distribution of the computation time for 10000 executions of each method in either simulation, where the red dotted lines represent the average computation time. We find that `Captain` outperforms AMOCS-MA in terms of the average overhead. Specifically, the difference in average computation time between AMOCS-MA and our approach is about 130 ms and 20 ms in the UUV case.

2.2.3 Choice of violation tolerance As discussed in the paper, the choice violation tolerance ϵ determine which requirement needs relaxation, while the larger the value is, the less number of requirements in the *Unsatisfied* requirement set, along with the underlying higher risk of no feasible solution. Thus the value of ϵ should be determined based on experimental data. The choice of different violation tolerance ϵ and the performance scanning accuracy achievement are illustrated in Figure 3(a). As the number of incidents increases, the impact of ϵ on requirement satisfaction is more obvious. Figure 4 illustrates adaptation rate of each soft requirements in UUV case, as the increase of violation tolerance. In the UUV case, we choose $\epsilon_{\varphi,e,l} = \{10^{-3}, 0, 0\}$, when the adaptation rate of each requirement tends to be gentle.
More details and videos of the experimental results can be found on the website[1].

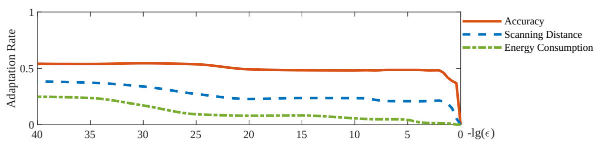


Fig. 4: Unsatisfied requirements reporting rate. (UUV case, # of incidents = 9)

References

1. Steven J Burian, Srinivas Pradeep Velugubantla, K Chittineni, Sri Ram Kumar Maddula, and Michael J Brown. Morphological analyses using 3D building databases: Portland, Oregon. Technical report, Utah. LA-UR, Los Alamos National Laboratory, Los Alamos, NM, 2002.
2. Alejandro Hidalgo-Panagua, Miguel A Vega-Rodríguez, Joaquín Ferruz, and Nieves Pavón. Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. *Soft Computing*, 21(4):949–964, 2017.
3. Stepan Shevtsov and Danny Weyns. Keep it simplex: Satisfying multiple goals with guarantees in control-based self-adaptive systems. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, pages 229–241. ACM, 2016.
4. Stepan Shevtsov, Danny Weyns, and Martina Maggio. SimCA*: A control-theoretic approach to handle uncertainty in self-adaptive systems with guarantees. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 13(4):17, 2019.

[1] https://yixingluo.github.io/Captain.github.io/