Supplementary Materials for the Simulation Experiment

Peng Lv [a], Zhangcong Xu [a], Yiding Ji [b], Shaoyuan Li [a], Xiang Yin [a]

[a] *Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China*

[b] *Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong 511458, China*

In this document, we provide complete details on constructing the DES model $G = G_0 \times G_1 \times T$ for the simulation experiment. Moreover, we also provide a detailed explanation of the optimal supervisor S^*.

DES Construction: Note that the DES G built by us is the product of G_0, G_1 and T. Specifically, even though the movement of the UGV and UAV is independent, which leads to their synchronization $G_0 \times G_1$ being a pure shuffle, to enforce synchronous movements, we further introduce a binary variable $k \in \{0, 1\}$ to encode the turn-based decision of two agents such that: if $k = 0$, then the UGV moves; otherwise, the UAV moves. Moreover, since properties of interest are on states in $G_0 \times G_1$ while properties of interest are on edges in the specification model T, states in $G_0 \times G_1$ needs to be synchronized with edges in T in order to obtain the overall DES $G = G_0 \times G_1 \times T$. We refer to [1] for details of this type of product. Therefore, in the DES G, each state is a four-tuple: (q, q', k, p), where q and q' represent the location of the UGV and UAV, respectively, k indicates who will move next step and p indicates the progress in completing specification T.

The resulting product system G contains more than 500 states. However, there are many states that are unrelated to the synthesis of the optimal strategy. For example, the sub-tasks captured by the edges from p_0 to p_5 in T can always be satisfied by a finite trajectory of the UGV without considering the uncontrollable behaviors of the UAV, which can be synthesized trivially [1], and these finite trajectories will not influence the average cost per visit to logistics, so removing the part associated with these sub-tasks from G will not influence the synthesis of optimal supervisor. Moreover, there are many states in G that are unreachable and we resort to some pruning methods to remove these redundant states. Finally, the resulting system for synthesis only has 40 states and 3 marked states with the controllable and uncontrollable events sets being $\Sigma_c = \{E, S, W, N\}$ and $\Sigma_{uc} = \{B, H, \overline{W}, \overline{E}\}$, respectively. The DES G after pruning redundant states is shown in Figure 1 with all the marked states being labeled by double circles and all the uncontrollable events being labeled by red arrows. Note that we further introduce a dead state $(*, *, *)$ and an uncontrollable event $*$ in Figure 1 to represent the case when the UGV and the UAV meet. Therefore, although G operates under the turn-based decision of two agents, it can still model the parallel moving mechanism between them.

Synthesis Result: We have implemented the proposed synthesis algorithm in `Python 3.9.8` with the optimal supervisor S^* being solved in 2.17 sec. Specifically, S^* works as follows:

- when M needs to move, then S^* enables all the defined actions at the corresponding state; for example, when we arrive at state $(q_6, q_7, 1, p_7)$, which means that M arrives at Lab 2, then S^* enables H, $\overline{E}$ and $\overline{W}$;
- when we arrive at states $(q_1, q_6|q_7|q_8, 0, p_6)$, which means that C arrives at lobby from Whs 1 and M can be anywhere, then S^* enables only E;
- when we arrive at states $(q_1, q_6|q_7|q_8, 0, p_7)$, which means that C arrives at lobby from logistics and M can be anywhere, then S^* enables only S;
- when we arrive at states $(q_2, q_6|q_7, 0, p_5)$, which means that C arrives at logistics from lobby or Lab 1 and M can be at Lab 1 or Lab 2, then S^* enables only W;
- when we arrive at state $(q_2, q_8, 0, p_5)$, which means that C arrives at logistics from lobby or Lab 1 and M can only be at Lab 3, then S^* enables both W and S;
- when we arrive at states $(q_5, q_6|q_7, 0, p_6)$, which means that C arrives at Whs 1 from lobby or Lab 1 and M can be at Lab 1 or Lab 2, then S^* enables only N;
- when we arrive at state $(q_5, q_8, 0, p_6)$, which means that C arrives at Whs 1 from lobby or Lab 1 and M can only be at Lab 3, then S^* enables both N and E;
- when we arrive at states $(q_6, q_7|q_8, 0, p_6)$, which means that C arrives at Lab 1 from Whs 1 and M can be at Lab 2 or Lab 3, then S^* enables only N;
- when we arrive at states $(q_6, q_7|q_8, 0, p_7)$, which means

---

⋆ This work was supported by the National Natural Science Foundation of China (62061136004, 61803259, 61833012). Corresponding author X. Yin.

Email addresses: `lv-peng@sjtu.edu.cn` (Peng Lv), `randomx200@sjtu.edu.cn` (Zhangcong Xu), `jiyiding@ust.hk` (Yiding Ji), `syli@sjtu.edu.cn` (Shaoyuan Li), `yinxiang@sjtu.edu.cn` (Xiang Yin).

that C arrives at Lab 1 from logistics and M can only be at Lab 2 or Lab 3, then S^* enables only W;
- when we arrive at states $(q_6, q_6, 0, p_6|p_7)$ or $(*, *, *)$, which means that C and M have met each other and the safety constraint has been violated, then S^* can only enable the dead action $*$,

where we use $q|q'$ and $p|p'$ to denote that M can be at q or q' and T can be at p or p', respectively.

References

[1] Bruno Lacerda, David Parker, and Nick Hawes. Optimal and dynamic planning for Markov decision processes with co-safe LTL specifications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1511–1516, 2014.

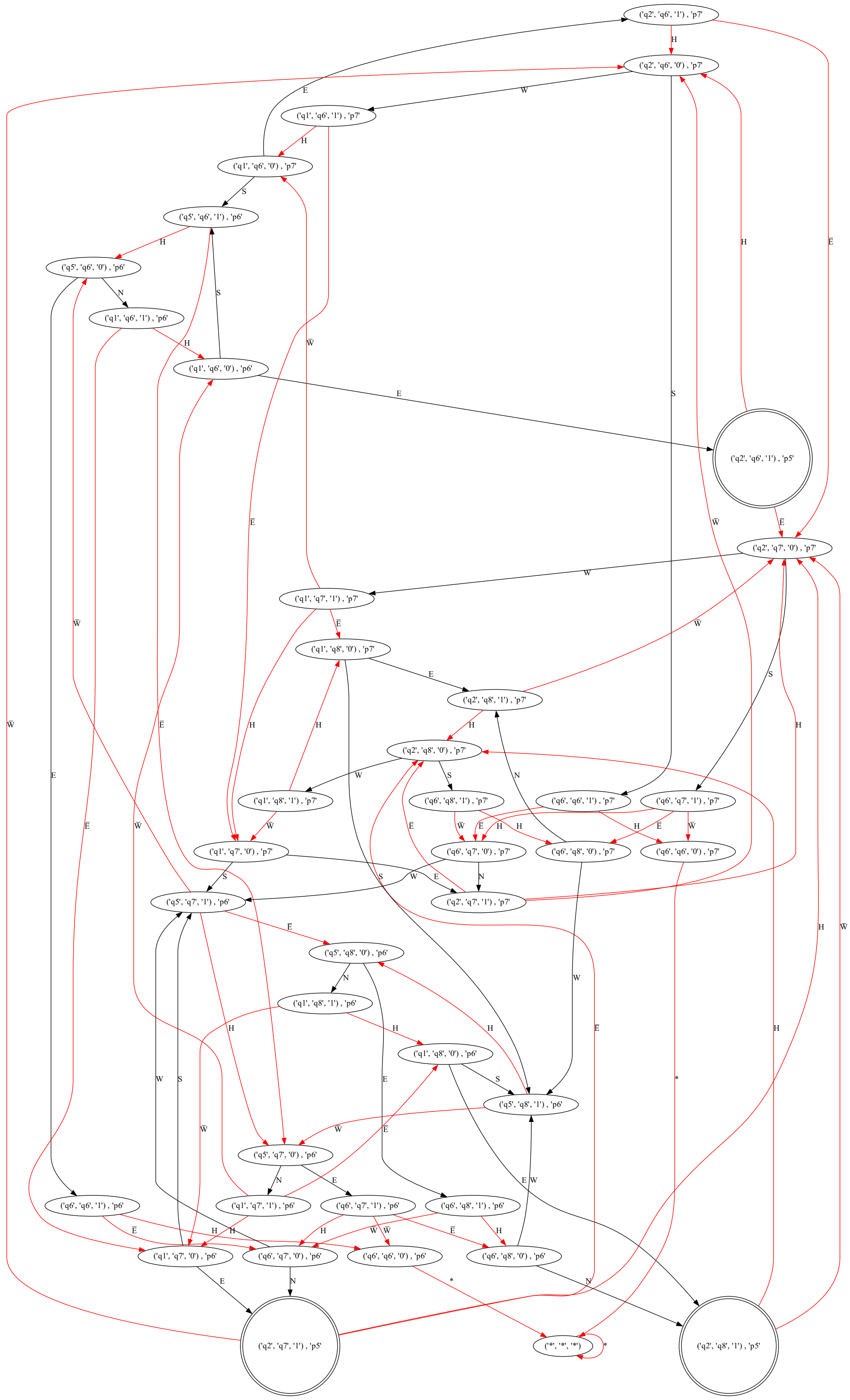

Fig. 1. The DES G after pruning redundant states.