WORK DYNAMICS OF TASKWORK AND TEAMWORK IN FUNCTION ALLOCATION FOR MANNED SPACEFLIGHT OPERATIONS

Martijn IJtsma, Lanssie M. Ma, Amy R. Pritchett and Karen M. Feigh
Georgia Institute of Technology
Atlanta GA USA

This paper proposes a methodology for human-robot function allocation for future manned space exploration missions that uses fast-time computational simulation. Dynamics of taskwork and teamwork often result in emergent work patterns that are difficult to predict from static analysis of function allocations. We model the dynamics of taskwork and teamwork and demonstrate our approach through a case study that explores the function allocation design space for an on-orbit maintenance mission involving humans and various robots. The case study highlights the method's ability to predict possible concerns associated with limited availability of physical resources, action interdependencies, and communication requirements with possible time delays, and shows the influence of work dynamics on mission performance.

Communication delays associated with future manned exploration missions are on the order of tens of minutes. These delays no longer allow for ground-centered concepts of operation in which the locus of control lies with the mission control center; instead, there is a requirement to shift autonomy from the ground to astronaut crewmembers. In this context, crew autonomy refers to the crew's ability to make decisions and execute tasks independently from ground control. NASA envisions robots to play a large role in this higher crew autonomy.

Function allocation is the process of distributing tasks over multiple agents and comprises two dimensions: allocation of authority and responsibility. Authority is the notion of which agent is executing a task. Responsibility denotes which agent is accountable for the outcome of a task. Function allocation for manned space flight operations poses several unique challenges: (1) due to communication delays the mission crew can no longer rely on ground support; (2) there is limited availability of resources (i.e., consumables and tools); and (3) the environment is unfamiliar and extremely hazardous. Possible concerns associated with these challenges are, amongst others, long idle times for some agents as they wait for others to complete tasks, excessive communication required between agents, and high taskloads. Thus, there is a need for design methods that can objectively evaluate human-robot function allocations.

In the space domain, earlier work on evaluation of function allocations ranges from descriptions of a method for optimizing cost or reliability in human-robot teaming (Shah, Saleh, Hoffman, 2007) to 1-g full-scale testing of different function allocations in a space assembly task (Rehnmark, Currie, Ambrose, Culbert, 2004). One of the difficulties in evaluating function allocation is that the interplay of availability of resources, interdependencies of tasks, and communication requirements with possible time delays results in emergent work patterns that are difficult to predict from static analysis of function allocations. Therefore, function allocation evaluation methods should account for the dynamics of both the taskwork conducted by a team and the teamwork within the team.

This paper applies a fast-time computational simulation framework called Work Models that Compute (WMC) to evaluate human-robot function allocations for manned space flight operations. The focus of this paper is on the modeling and simulation of the dynamics of the team's taskwork and teamwork. A case study demonstrates simulation of the work dynamics for various function allocations for an on-orbit maintenance mission, highlighting the ability of this method to predict potential concerns.

Computational Simulation Framework

WMC is a computational simulation framework that can evaluate function allocation options. The framework has previously been used for analyzing and synthesizing function allocation in the air traffic management system (Pritchett, Bhattacharyya, & IJtsma, 2016) and between the pilot and the autoflight system in the flight deck (Pritchett, Feigh, Kim, 2014a; Pritchett, Feigh, Kim, 2014b). Work dynamics are modeled in WMC through three inter-acting types of models: agent models, resources, and actions.

Information and physical entities in the work environment are modeled as resources. Information resources, are represented computationally as variables. Physical resources, such as tools or spacecraft components, are modeled by computational structures defining attributes important to their use, including containing information on the resource's location, current "ownership" by an agent, and availability.

Actions are standalone descriptors of the work that is being performed. Each action has an attribute that defines the interaction with the environment through three types of relationships with resources: (1) an action *gets* an information resource, thereby retrieving information from the environment; (2) an action *sets* an information resource, corresponding to changing information in the environment; and (3) an action *uses* a physical resource, similar to using a tool in real-life.

Agent models do not contain descriptors of specific work activities. Thus, by modeling the actions outside of the agent models, new function allocations can be easily tested by differing assignments of actions to agents during run-time. The agent model then takes in the action it is assigned and executes it. WMC can use any type of agent model that is deemed appropriate for the analysis as long as it meets the computational interface standard of accepting calls from the simulation framework to execute the actions it is passed during run-time. Examples of agent models currently used in WMC are a perfect agent that can execute all tasks instantly and perfectly, and a more-extensive model that also adds elements of task management, delaying and interrupting actions when its assigned taskload reaches limits.

The interplay of the three types of models determines the work dynamics. Agents influence the work dynamics through how they manage them, such as methods for task management. The duration of an action can be contingent on the agent executing it (e.g., robots typically take longer to complete an action than a human). Physical resources can be used by only one action at a time: When an action requires a physical resource that is not available at the scheduled action time, WMC will delay the action until all the required resources are available. Finally, actions influence the work dynamics through their interdependencies, which are modeled through actions scheduling follow-up actions. Actions can be scheduled to occur immediately after one action has been completed, or can be scheduled at a later time. Additionally, the modeler can specify the action's own update cycle, through which the action can dynamically determine its own required next execution time.

This format allows simulation of both taskwork (i.e., the work inherently required to fulfill mission objectives, regardless of function allocation), and the teamwork that is implicitly or explicitly required to coordinate the taskwork for any given function allocation. When allocating the taskwork to agents in terms of authority and responsibility, WMC will automatically engender teamwork actions that are required by an authority-responsibility mismatches. This mismatch occurs when one agent is authorized to perform an action but a different agent is responsible for the outcome of that action (Woods, 1985). Mismatches are common when allocating authority to robotic agents, as they typically cannot be held accountable for action outcomes and a human is therefore required to verify whether the action has been completed to standard. This required teamwork is modeled in the form of monitoring and confirmation actions. Monitoring is a parallel action in which the responsible agent observes the authorized agent during the execution of the main action. Confirmation is a subsequent action in which the responsible agent confirms the successful execution of the main action after it has been executed. The authorized agent needs to wait for confirmation before it can continue with its next action. These monitoring and confirmation actions emerge as the work progresses, and themselves can impose significant taskload on agents.

WMC assesses several metrics of the function allocation. Logging of the action execution times and duration for each agent captures the total time required to perform the mission, as well as the taskload imposed on each agent. Likewise, logs of physical interaction capture when an agent performs an action that requires a physical resource that has last been used by another agent implying the need for an exchange of physical resources. Similarly, cognitive interaction is logged when one agent gets an information resource that has earlier been set by another agent, reflecting the transfer of information between the two agents.

Case Study

To demonstrate the methodology, we analyze several function allocation options for an on-orbit maintenance mission involving several robots and human astronauts. The goal of the on-orbit maintenance mission is to inspect and, whenever necessary, repair three exterior panels on the spacecraft. The agent included in the analysis are an extra-vehicular astronaut (EV), an intra-vehicular astronaut (IV), a Remote Manipulator System (RMS), two Robonaut robots, and a Mission Control Center (MCC) agent. The agents are modeled with a simple performance model that can only execute one action at a time, except for monitoring and confirmation actions for which there is no taskload limit. Physical resources in the scenario include, amongst others, toolsets to conduct the inspection and perform necessary repairs (usually two), five panels (three that need to be inspected and two backup panels that can be used for repair) and a Portable Life Support System (PLSS). Information resources include information on the panels that need to be repaired and information on confirmation by a responsible agent.

The first two columns of Table 1 show the decomposition of the required taskwork into functional blocks and actions. Columns 3-6 show four different function allocation (FA) options in terms of authority/responsibility; FA1 and FA2 are reasonably attainable with current robotic capabilities, and FA3 and FA4 may be possible with future robotic capabilities. To demonstrate the effect of resources on work dynamics we additionally simulate an altered version of option FA4 in which there is just one instead of two toolsets. Responsibility for actions that are executed by robots is usually allocated to the IV astronauts, engendering monitoring actions, although FA4-B examines the impact of giving responsibility to the MCC engendering confirmation actions, an allocation impacted by an assumed 10 second communication delay. The EV astronaut is responsible for his/her own actions.

Table 1. Candidate function allocations for the on-orbit maintenance scenario.

Functional blocks	Actions	Current capabilities		Future day capabilities		
		FA1	FA2	FA3	FA4-A	FA4-B
1. Exit dock	1.1 Prepare	EV/EV	EV/EV	EV/EV	Robo2/ IV	Robo2/MCC
	1.2 Leave dock	EV/EV	EV/EV	EV/EV	Robo2/ IV	Robo2/MCC
2. Traverse	2.1 Traverse	EV/EV	EV/EV	EV/EV	Robo2/ IV	Robo2/MCC
3. Inspect panel	3.1 Get inspection tools	EV/EV	EV/EV	Robo1/ IV	Robo2/ IV	Robo2/MCC
	3.2 Apply inspection tools	EV/EV	EV/EV	EV/EV	Robo2/ IV	Robo2/MCC
	3.3 Store inspection tools	EV/EV	EV/EV	Robo1/ IV	Robo2/ IV	Robo2/MCC
4. Repair panel	4.1 Get repair tools	EV/EV	EV/EV	Robo1/ IV	Robo1/ IV	Robo1/MCC
	4.2 Get new panel	EV/EV	RMS/IV	RMS/ IV	RMS/ IV	RMS/MCC
	4.3 Remove broken panel	EV/EV	EV/EV	EV/EV	Robo1/ IV	Robo1/MCC
	4.4 Emplace new panel	EV/EV	EV/EV	EV/EV	Robo1/ IV	Robo1/MCC
	4.5 Dispose of broken panel	EV/EV	RMS/ IV	RMS/ IV	RMS/ IV	RMS/MCC
	4.6 Store repair tools	EV/EV	EV/EV	Robo1/ IV	Robo1/ IV	Robo1/MCC
5. Enter dock	5.1 Enter dock	EV/EV	EV/EV	EV/EV	Robo2/ IV	Robo2/MCC

Results

Figure 1 shows the time trace of the actions for function allocation option FA1. With just one agent, the EV, performing all the work, this manifests as a linear sequence of the actions. From left to right, the agent first prepares and leaves the dock, then traverses to the first panel, inspects it and continues to the next panel. Inspection of the second panel shows it needs repair and the agent thus gets a new panel and swaps it with the damaged panel. After also repairing the third panel, the agent returns to the dock. This function allocation has zero idle time, no interaction between agents and no monitoring requirements.

Figure 2 shows the time trace for option FA2. Some actions can occur simultaneously; the RMS can get a backup panel while the EV prepares the tools and start removing the broken panel. Then, the RMS can dispose of the broken panel while the EV installs the backup panel. The parallel occurrence of the actions reduces the total mission time. However, because the two processes are not perfectly synchronized, the RMS frequently needs to wait for the EV, reflected in the increased idle time as compared to FA1. Also, divvying up the tasks in this way requires physical interaction between in the agents, in this case the backup panel is first being “used” by the RMS and then transferred to the EV. Similarly, the broken panel is transferred from the EV to the RMS. Finally, the IV is responsible for the RMS’s operation and thus needs to monitor the task progress, with idle times in between.

The simulation results for FA3 are shown in Figure 3. Off-loading the tasks from EV to Robonaut does not result in a notable decrease in mission time, mostly because the interdependencies between their actions do not allow for parallel execution. For example, inspection cannot start before the inspection tools have been prepared. Additionally, this FA requires notable physical interaction between the Robonaut and EV to interchange tools.

Figure 4 shows the result for option FA4-A in which a Robonaut performs panel inspection and a second Robonaut performs repair actions, in coordination with RMS. The actions for inspection and repair are fairly independent, and, thus, having them executed by different agents can notably decrease mission duration. Additionally, these actions being standalone and comparable in duration results in a reduction in the total idle time and the required number of physical resource exchanges. However, with multiple robotic operations occurring in parallel, the IV experiences a high monitoring load.

The constraints imposed by physical resources are clearly shown in the differences between FA4-A with two toolsets (Figure 4) versus FA4-A with one toolset (Figure 5). With two toolsets, repair and inspection can be performed in parallel, whereas sharing one toolset between multiple agents results in agents waiting for each other. The sharing of the toolset also increases the required number of physical resource exchanges.

Finally, Figure 6 shows the time trace for FA4-B, which has MCC responsible for all robotic operations. Here, the authority-responsibility mismatch was assumed to be addressed through confirmation actions, since the 10 second delay in the MCC's receipt of any portrayal of the execution of the action would obstruct real-time monitoring. From the time trace it is clear that the required confirmation actions together with the communication delay result in an inefficient function allocation with long idle times. It does, however, alleviate the monitoring load of the IV, who is now available for other tasks. High confirmation taskload for MCC may not be a problem as MCC can easily increase capacity through (comparably) unlimited human resources.

Conclusions & Future Work

This paper described the modeling and method for evaluating function allocation options for future manned space flight. We argue that the benefit of using computational simulation is its ability to identify emergent work patterns that might go unnoticed when applying static analysis methods. Insight in the emergent interplay between agents, the availability of resources and interdependencies of actions can help the designer make more informed trade-offs in the function allocation process. The case study highlights our approach to simulating the dynamics of taskwork and teamwork and the major influence it can have on the system's performance.

Beyond the initial evaluation provided here, the simulation framework can be applied throughout the process of designing function allocations. Identifying the emergent patterns and steering design decisions based on a good understanding of the implications of function allocation options will ease the design process in later stages. Although we intend to extend our case study with more accurate and elaborate task decompositions, we also believe there is a benefit in modeling coarser tasks and simulating them in WMC to gain higher-level insights before breaking down the tasks into smaller subtasks. WMC can be used in an iterative process in which identification of the required taskwork and teamwork is updated based on results from the computational simulation and vice versa.

In the future we plan to extend the WMC architecture to simulate the dynamics of taskwork and teamwork in greater detail and at higher levels of fidelity, supporting the later, more detailed evaluations of function allocations and the procedures and mechanisms supporting them. We will additionally consider new teamwork actions between robots and humans that are of interest to function allocation designers, particularly focusing on the differences in capabilities of robots and humans and the consequences thereof on the required teamwork.

Acknowledgements

This work is sponsored by the NASA Human Research Program with Jessica Marquez serving as Technical Monitor under grant number NNJ15ZSA001N. The authors would also like to thank the other WMC developers.

References

Pritchett, A.R., Bhattacharyya, R.P. & IJtsma, M. (2016). Computational Assessment of Authority and Responsibility in Human-Automation Function Allocation. *Journal of Air Transportation*, 93–101.

Pritchett, A.R., Kim, S.Y., Feigh, K.M. (2014a). Modeling Human-Automation Function Allocation. *Journal of Cognitive Engineering and Decision Making*, *8*(1), 33–51.

Pritchett, A.R., Kim, S.Y., Feigh, K.M. (2014b). Measuring Human-Automation Function Allocation. *Journal of Cognitive Engineering and Decision Making*, *8*(1), 52–77.

Rehnmark, F., Currie, N., Ambrose, R.O., & Culbert, C. (2004). *Human-Centric Teaming in a Multi-Agent EVA Assembly Task* (No. 2004-01-2485). SAE Technical Paper.

Shah, J.A., Saleh, J.H., & Hoffman, J.A. (2007). Review and synthesis of considerations in architecting heterogeneous teams of humans and robots for optimal space exploration. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, *37*(5), 779–793.

Woods, D.D. (1985). Cognitive Technologies: The Design of Joint Human-Machine Cognitive Systems. *AI Magazine*, *6*(4), 86–81.

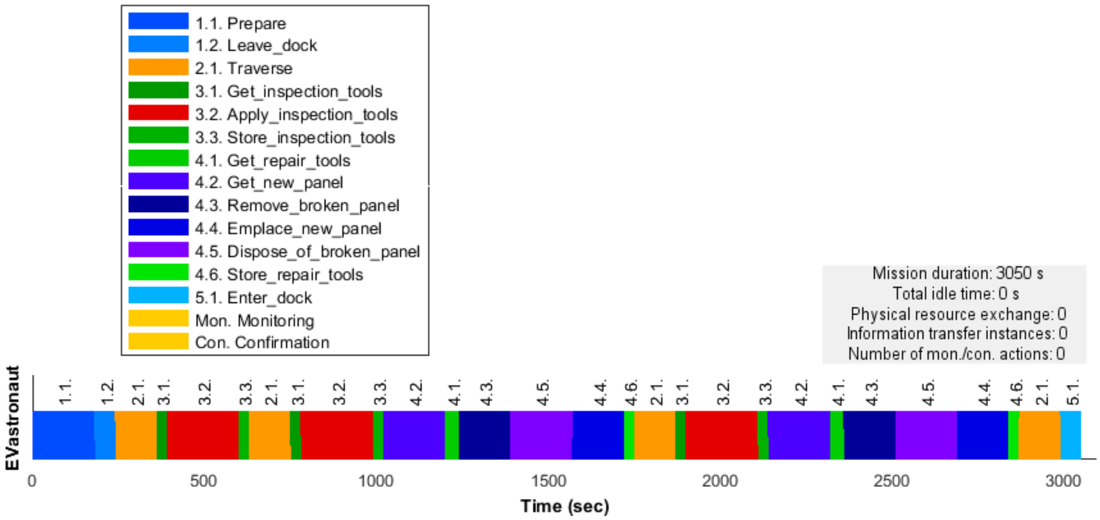

Figure 1. Function allocation option 1 (FA1) with EV astronaut performed all of the work.

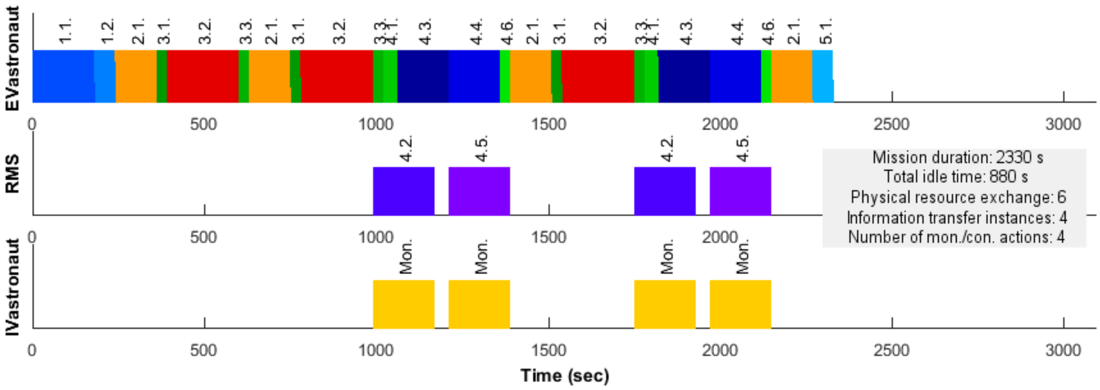

Figure 2. Function allocation option 2 (FA2) with RMS handling panels.

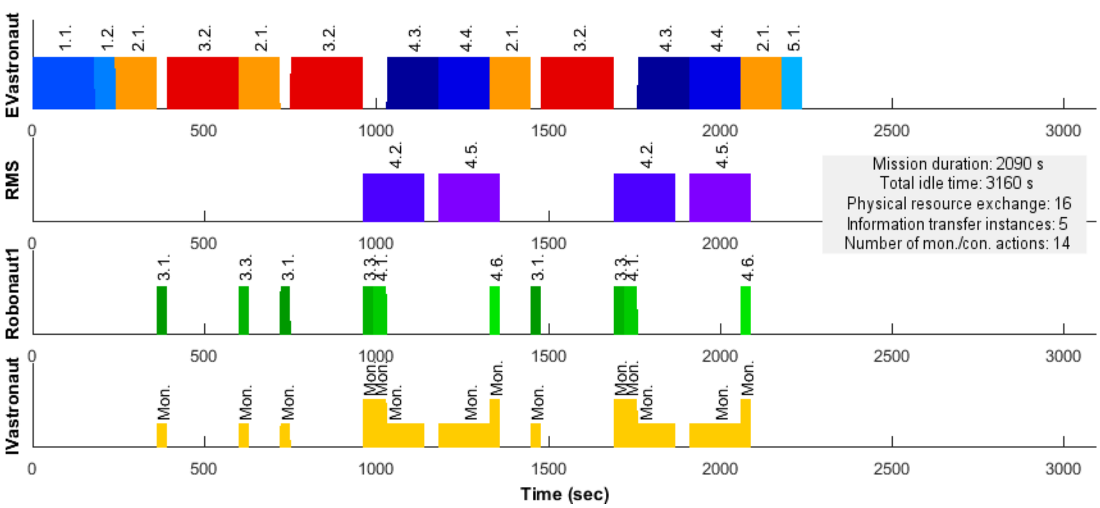

Figure 3. Function allocation option 3 (FA3) with RMS handling panels and Robonaut managing tools.

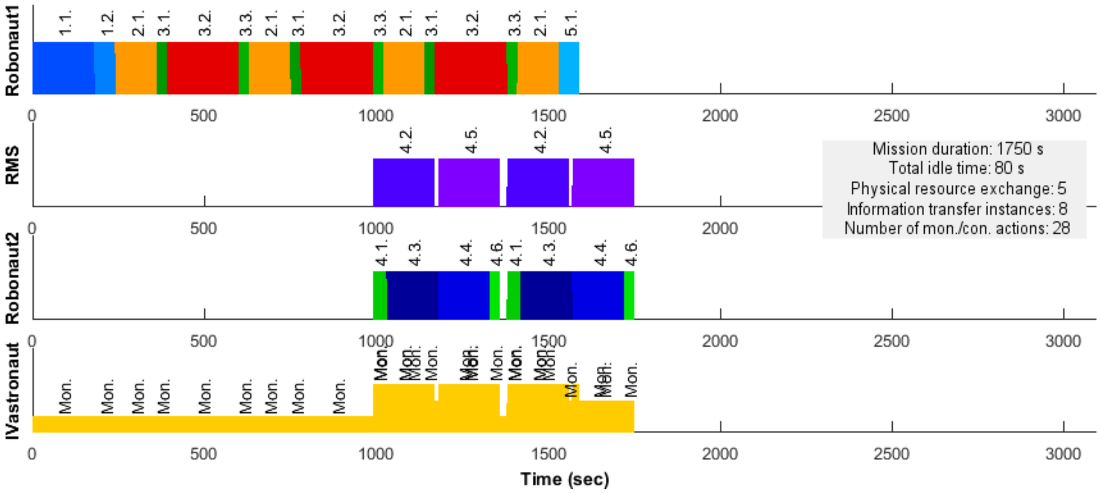

Figure 4. Function allocation option 4 (FA4-A) with Robonaut1 doing inspection, Robonaut2 & RMS doing repair.

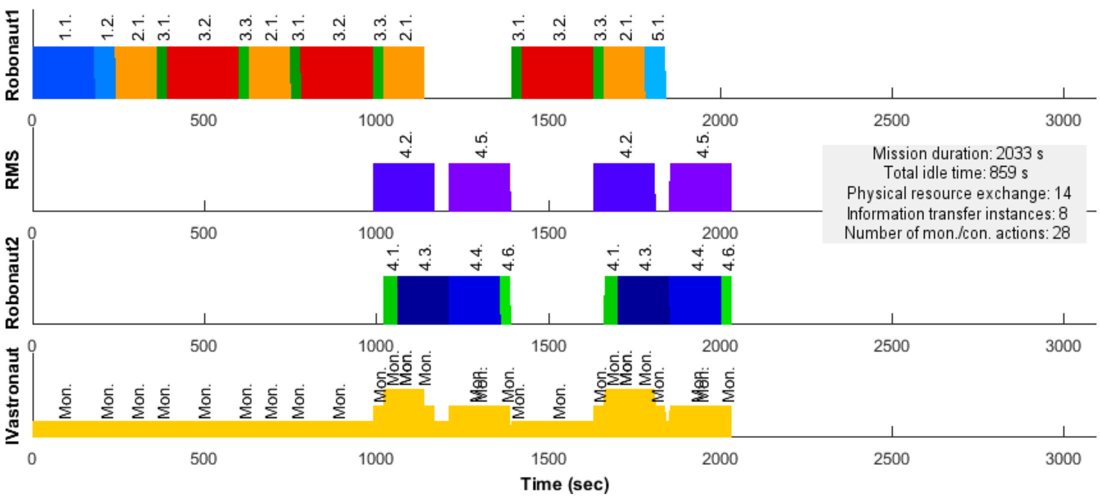

Figure 5. Function allocation option 4 (FA4-A) with only one toolset, shared by the agents, for inspection and repair.

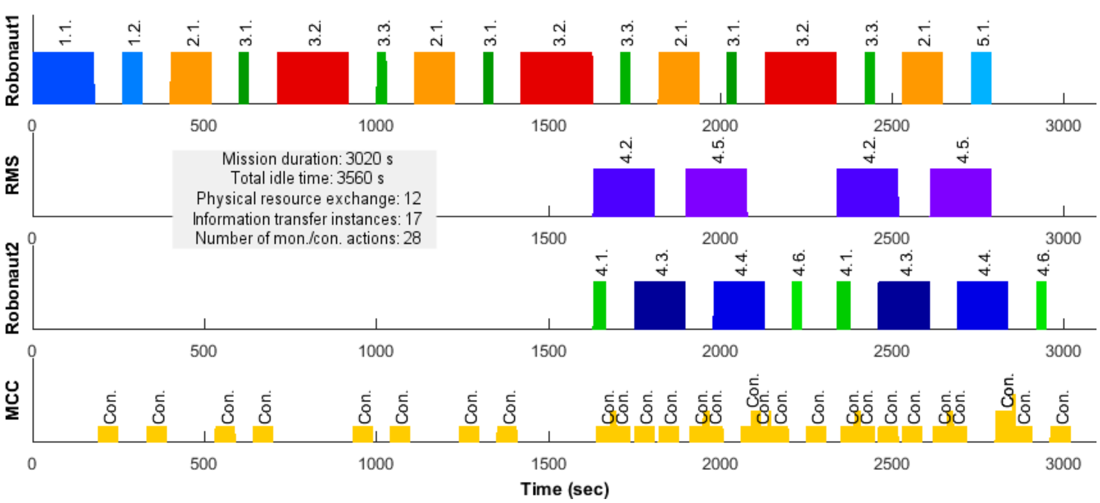

Figure 6. Function allocation option 4 (FA4-B) with MCC being responsible for robotic operations.