IAC-16,B3.4-B6.5,2

# Adapting Columbus Operations and Providing a Basis for Future Endeavours

**J. M. Bach*, D. Sabath*, G. Söllner*, F. Bender+**
* DLR, Oberpfaffenhofen, Germany
+ Technische Universität München, Lehrstuhl für Raumfahrttechnik

**67th International Astronautical Congress,**
**26 – 30 September 2016**
**Guadalajara, Mexico**

# IAC-16,B3.4-B6.5,2

# CONSOLIDATING COLUMBUS OPERATIONS AND LOOKING FOR NEW FRONTIERS

**Author**
J. M. Bach, DLR, Oberpfaffenhofen, Germany, marius.bach@dlr.de
**Co-Authors**
D. Sabath, DLR, Oberpfaffenhofen, Germany, Dieter.Sabath@dlr.de
G. Söllner, DLR, Oberpfaffenhofen, Germany, Gerd.Soellner@dlr.de
F. Bender, Technische Universität München, Lehrstuhl für Raumfahrttechnik, florian.bender@tum.de

## ABSTRACT

On 15th December 2015, Timothy Peake – the 4th ESA astronaut in 20 months – headed into orbit for a 6-month stay on the ISS. The British astronaut's “Principia” mission holds many interesting tasks, not only for Tim Peake himself (he performed an EVA on 15th January 2016) but also for the teams on the ground. One of the most exciting activities was the second session of the Airway Monitoring experiment, which again included an experiment run in the US airlock under coordination of the Columbus Control Centre (Col-CC). Besides that, there were many other experiments, such as EML, PK4, DOSIS and Meteron, and also the transition to new NASA tools (e.g. WebAD) was done in this period. Since the establishment of ESA's new setup in July 2015, Col-CC has been working together with all its partners to define the new interfaces, exploit new possibilities, and define in detail the tasks for the operations teams. Besides the ongoing work to monitor and command Columbus, support the ESA experiments on the ISS, as well as supporting the ESA astronaut himself, Col-CC is looking forward towards potential future tasks and challenges. Based on many years of experience in human space flight, an initial study was launched to investigate some of the challenges of human space flight activities beyond Earth orbit.

One of these challenges is the delay of communication transmissions experienced over long distances. Until now, all our human space flight operations have been based on (near) real-time communications to monitor and control the spacecraft. This paper describes the results of our study investigating the necessary changes to current operations in the case of long-distance communications. Example procedures are assessed on their reliance on real-time communications and thus how current operations would be impacted by transmission delays. Methods are proposed to make the procedures tolerant to delays, and enable operations to use these procedures for deep space missions.

## Introduction

The “Principia” mission of Timothy Peake started in December 2015 and was the final event in the row of ESA astronauts in the years 2014 to 2016. Starting with the German ESA astronaut, Alexander Gerst in May 2014, the almost continuous stay of ESA astronauts on-board ISS was continued by Samantha Cristoforetti in December 2014 and Andreas Mogensen in September 2015. While the “Principia” mission has just ended in June 2016, the next mission with Thomas Pesquet is already approaching and will start in December 2016.

Based on the long experience of DLR’s German Space Operations Centre(GSOC) in manned space operations, in 2005 the Eneide Mission and

---

in 2006 the support of the Astrolab mission with Thomas Reiter (see [1] and [2]) could be performed successfully. The Interim Utilization Phase was done in parallel to setting up Col-CC ([3] to [5]) for the later Columbus operations. Since February 2008, when Col-CC started its Columbus operations (see [6] to [14] and [13] to [14]), all further missions and increments have been prepared and supported successfully. Also, the new setup with split responsibilities among several services and new interfaces resulted in more efficient operations of Columbus and the payloads. With this experience, it will allow operating Columbus until at least 2024, assuming that the boundary conditions will not change (see [12]).

In parallel to the standard operations tasks, the Col-CC Flight Control Team (FCT) is looking towards the future and is conducting preliminary investigations on selected aspects of future deep space missions. In the study [15], modifications to operations have been analysed when boundary conditions change from (near) real-time to long-distance communications. The focus was put on quantifying the effect of the signal propagation delay on procedure execution.

## European Astronauts on ISS

Having the fourth ESA astronaut in a short timeframe on-board ISS was one of the main driver of Increment 45/46. Timothy Peake started his “PRINCIPIA” mission on 15 December 2015 launching from Baikonur in Kazakhstan together with Roscosmos cosmonaut Yuri Malenchenko and NASA astronaut Tim Kopra. After four orbits, the Soyuz capsule docked to the ISS and the three crewmates could enter the ISS on the same day (see Fig. 1).

Fig. 1: Tim Peake arrives at the ISS (Photo: NASA)

Already four weeks after his arrival at the ISS, ESA astronaut Tim Peake performed his first EVA and worked outside the ISS together with Tim Kopra (see Fig. 2). He performed the EVA on 15 January 2016 replacing a failed power regulator and installing some cabling. The EVA was shortened due to problems with Tim Kopra’s EVA suit.

Fig. 2: Tim Peake during his EVA
(Photo: NASA)

Among many other experiments, Tim Peake performed the Airway Monitoring experiment in the US Airlock under reduced pressure together with Tim Kopra. The scientific goal of ESA’s Airway Monitoring experiment is to investigate how space flight affects lung health by measuring exhaled nitric oxide (NO) levels as evidence of airway inflammation. Previous research indicates that humans in space are prone to airway inflammation due to the increased risk of inhalation of dust and other free-floating particles as described in [14].

After preparing the experimental setup in the US Airlock, Tim Peake and Tim Kopra performed the experiment under reduced atmospheric pressure on 25 February 2016 (see Fig. 3). Thanks to the lessons learned which have been implemented since the last session in 2015, the experiment could be performed very smoothly.

Fig. 3: Tim Peake performing the Airway Monitoring experiment in the Airlock (Photo: NASA)

Tim Peake returned to Earth with Soyuz 45S on 18 June 2016 with cosmonaut Yuri Malenchenko and astronaut Tim Kopra after working on-board the ISS during Increments 46 and 47.

## New Interfaces, new Tools and Standard Operations

In [13], the new operations setup implemented in January 2014 is described. Meanwhile this setup has been fully implemented and experience during three ESA long-term missions has been gained.

The infrastructure upgrade of the Col-CC Ground Segment has been finalized in the last year. The virtualization of server hardware was mostly done by the Col-CC Ground segment team and offers on the one side improvement of server handling and associated hardware and on the other side makes control room swaps easier than before. Details on the virtualization at Col-CC can be found in [16] and [17].

In Spring 2016, the first commissioning phase of the new on-board Solid State Drives (SSD) was performed. The first tests show good performance of the new subsystem which will improve the handling of large on-board recorded video files.

Also in 2014, NASA started to introduce a new planning suite called OPTIMIS, consisting of SCORE, WebAD and Viewer. The implementation of SCORE and Viewer is described in [14]. After the start of the operational use of Viewer, in September 2015 the change from the ESA tool OPDCS (Operations Planning Data Collection System) to the NASA tool WebAD (Web Activity Dictionary) took place. The advantage of the transition to WebAD is that the ESA team will use the whole NASA planning suite instead of an ESA stand-alone solution. This allows the use of a unified and integrated system with a strong backing in development. In [14], the planned transition schedule in the timeframe until September 2016 described the new system to be ready for operational usage in increment 49/50. Because of some problems with the OPDCS end of 2015, the transition schedule has been advanced by 6 months to December 2015. As the teams have been trained for WebAD until then, it was possible to substitute OPDCS by WebAD already at the start of Increment 47/48. Hence, a transition procedure was developed and some tests were performed in the next weeks to ensure a smooth transition in early 2016. In January 2016, the OPDCS database was frozen for some days and the whole database content of the OPDCS was transferred to WebAD and cross-checked for transfer flaws. After confirmation of a good transition of the data to the new tool, WebAD has been used in operations by the ESA team from the start of Increment 47/48 onwards. On the operations side, OPDCS was retired a few weeks afterwards, when the new tool was judged fully reliable for operations.

In addition to the scheduled updates, the Columbus FCT has to deal with unplanned on-board events. In July 2016, one of the CWSA (Columbus Water Separator Assembly) went off and a switch-over to the second unit had to be performed. Meanwhile, the first unit is up again but it is intended to replace it by a new unit, which will be delivered by the end of 2016.

## Investigations on Operations in Deep Space

One major challenge of manned deep-space flight is the communication signal delay experienced over long distances. Until today, all human space flight operations have been based on real-time communications to monitor and control spacecraft. In the future, missions well beyond Earth's vicinity are anticipated, which will result in the need for delay-tolerant operations.

So far, only robotic missions have been conducted in deep space with significant delays in communications, while all manned missions were close to Earth. For the communications delay, the One Way Light Time (OWLT) is an appropriate measure. Table 1 provides an exemplary overview of the approximate maximum OWLT of various missions during their farthest point from Earth.

Tab. 1: OWLT of various missions

| Destination | Mission | OWLT |
|---|---|---|
| LEO | ISS, via GSO relay satellites (manned) | < 0.3 s |
| Moon | Apollo 11 (manned) | 1.3 s |
| $L_2$ (Earth-Sun) | Gaia | 5.0 s |
| 2000 SG344 | *proposed* (manned) NEO mission [18] | 11.9 s |
| Mars | Mars Science Laboratory | 22.3 min [19] |
| Europa | Galileo | 52.1 min [20] |
| Pluto | New Horizons | 4.4 h |

Space operations heavily rely on validated ground and flight procedures. For Columbus and ISS in general, all activities and commanding are planned, prepared and coordinated days and weeks in advance; this also includes preparing for numerous contingency situations (for which analysis and recovery actions are pre-defined).

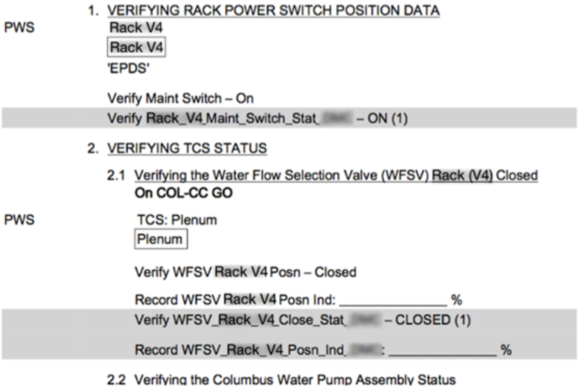
The study examines the impact of communication delays on the existing procedures that are being used for commanding by flight controllers in day-to-day Columbus operations.

## Procedures for Columbus Operations

The Columbus procedures, known as Operations Data File (ODFs), exist for a range of tasks and situations, including:

- Columbus activation & checkout as well as module safing & deactivation
- maintenance (both preventive and corrective)
- activation, operation, deactivation and reconfiguration of both systems and payloads
- contingency, emergency, off-nominal operation

Most commonly, checklist procedures are used for daily operations, which are textual and list step-by-step instruction sets. Each procedure step has a sequential number and a title and contains one or more atomic instructions in a standardized format. It may be divided into several smaller steps with each sub-step having a number and title, too (see Fig. 4). Procedure steps may contain additional notes, caution and warning blocks to provide further background information or indicate where special care is required. Off-nominal blocks prescribe pre-defined reactions for some contingency situations that may arise during execution of a procedure.



1. VERIFYING RACK POWER SWITCH POSITION DATA
PWS Rack V4
Rack V4
'EPDS'

Verify Maint Switch – On
Verify Rack_V4_Maint_Switch_Stat_[redacted] – ON (1)

2. VERIFYING TCS STATUS

2.1 Verifying the Water Flow Selection Valve (WFSV) Rack (V4) Closed
On COL-CC GO

PWS TCS: Plenum
Plenum

Verify WFSV Rack V4 Posn – Closed

Record WFSV Rack V4 Posn Ind: ______________ %
Verify WFSV_Rack_V4_Close_Stat_[redacted] – CLOSED (1)

Record WFSV_Rack_V4_Posn_Ind_[redacted]: ______________ %

2.2 Verifying the Columbus Water Pump Assembly Status

Fig. 4: (Redacted) first steps of a procedure used in Col‑CC ground operations.

As shown in Fig. 4, when a procedure step can be executed by both crew and ground, it typically lists the unshaded crew instructions followed by the equivalent ground instructions in the shaded block. Ground operators are familiar with both the crew instructions and the ground language and thus can follow along all instructions of a procedure.

Whenever possible, steps are performed by ground commanding to preserve crew time as much as possible.

There are, however, procedures that consist of combined crew and ground steps. An example is the installation of new hardware including its initial activation, where the Columbus power outlets are being commanded by ground to save the crew time required for handling of these steps. In these procedures, frequent synchronization points between crew and ground are driving the proceeding between steps. These points of synchronization are explicit voice call-outs by the ground resp. a crew member contacting the responsible control centre, who then has to wait for a reply by the ground before being allowed to continue with the execution of the procedure.

For crew call-outs, this is mostly codified as an «On COL CC GO» (read: ask COL CC if ready to proceed) or «✔COL-CC on […]» (read: check with COL CC that […]) instruction. These calls are generally used to synchronize execution of the current procedure with the ground, e.g. to make sure a set of ground-only instructions have been successfully completed, or to consult with the ground on how to handle off-nominal situations, e.g. above-threshold smoke detector sensor values.

Naturally, the on-board crew-only procedures and steps are not affected by an increased latency in communications. However, for the mentioned synchronization points as well as for ground-only procedures, delays pose a significant impact.

Reaching a critical system state during operations should be avoided at all times. Due to the way the systems on-board the station are designed, entering a transient critical state cannot be avoided during the execution of some procedures.

In general, a critical state is reached when systems are activated but the automatic, continuous checks for operational limits by the on-board Data Management System are not (yet).

For example, in a procedure for activating a payload rack (International Standard Payload Rack, ISPR), the power to the ISPR is switched on a couple of steps before the internal smoke detectors are activated. This leaves a timeframe within which a locally confined fire can stay undetected. Such a situation is especially critical when the execution of a procedure is halted between entering and exiting the critical state, i.e. in this example between activating the power and switching on the smoke detectors. This is one of the reasons why system safing and rolling back procedures step-by-step is important when their execution has been aborted.

### Impact of signal propagation delay

Real-time execution of activities for space systems is governed by the time a signal takes to travel from its source to a destination. The OWLT is a measure of distance normalized to the speed of light and quantifies this time in the optimal condition, i.e. when the signal travels at full light speed.

For example, a telemetry packet sent by a probe on the lunar surface takes about 1.2 to 1.3 seconds to reach a ground station on Earth, assuming a direct path of communication. To send a command to the probe and receive changing telemetry values as a result of this command thus takes at least 2.4 seconds.

Such a send-receive or write-read cycle is called a round-trip: A packet is sent from the ground station and arrives at the probe, then a resulting telemetry packet travels all the way back from the probe to the ground station. The duration of the round-trip is (at least) two times the OWLT.

The procedure steps used in Col-CC operations generally follow a read-write-read pattern, i.e. certain telemetry values are verified to be at an expected value resp. within an accepted range, then a command is sent and the resulting change of state verified via telemetry again. This three-step process is repeated within the procedure multiple times for all equipment of concern to the associated activity. The execution of procedures thus requires multiple full round-trips of communication, one for each write-read group of instructions that may be clustered into one or more procedure steps. However, procedures often contain multiple successive read-only steps, which do not require a round-trip to be executed one after another, and a varying number of steps with one or more commands.

It follows that the number of round-trips does not correlate with the number of steps in a procedure and they are therefore not a good indicator for the impact of the signal propagation time.

### Characteristics of procedure/time criticality

To assess the criticality of a procedure based on its steps, basic characteristics can be defined:

- $n_{steps,total}$ – total number of steps in a procedure

Rule: Add up the number of steps in the procedure. If a procedure step is split into sub-steps, count the number of sub-steps instead.

- $n_{steps,crit.}$ – number of steps while in critical state

Rule: Similar to the total number of steps, but just counting the steps between entering and leaving a critical system state, inclusive of the steps where this state is entered resp. left.

As outlined before, some steps (incl. sub-steps) in procedures contain multiple read-write groups of instructions, while many others only contain read instructions. As such, the steps as currently defined in procedures do not provide the necessary granularity to measure the amount of round-trips and must thus be decomposed into individual instructions. These instructions are then regrouped into sets of instructions that can be executed in one block without the need for a full round-trip.

Hence, the following mentions of “sets” refer to a group of continuous read instructions that may be followed by one or more write instructions, i.e. a set ends after a write instruction and before the following read instruction. Commonly, such an instruction set consists of several Verify and/or Record instructions followed by one command.

The general rule is that an instruction set contains consecutive instructions that are acted upon in quick succession without an intermediate waiting time induced by the data transmission to and from the spacecraft. Such a group of instructions is distinguished from a “procedure step”, which is a numbered and named step forming a group of logically or procedurally related instructions.

As this study focusses on the activities in the control centre, explicit call-outs by the on-board crew to the ground are ignored when re-grouping instructions of a procedure. For completeness, these call-outs are instead measured by a dedicated characteristic.

Characteristics of procedures that can be derived from decomposed procedures:

- $n_{sync,explicit}$ – number of explicit synchronization points

Rule: Add up the number of explicit call-outs to the station or ground, e.g. “ON Col-CC GO”.

- $n_{sync,implicit}$ – number of implicit synchronization points

Rule: Count the number of consecutive read-write instruction groups and subtract one, resp. increase this number by one any time a command instruction (or set thereof) is followed by a Verify or Record instruction (ignoring explicit call-outs and “Record Command Time Stamp” instructions).

- $n_{sync,total}$ – total number of synchronization points

Rule: Sum of $n_{sync,implicit}$ and $n_{sync,explicit}$. Equal to the number of round-trips plus the number of coinciding implicit and explicit sync points.

### Impact factors

Based on these characteristics, a number of impact factors can be derived. These are listed below and describe different kinds of procedure criticality: Either the share in critical steps or in (various types of) round-trips experienced during execution of a procedure. The latter factors enable measuring the impact the OWLT has on procedure execution in a quantitative way. All factors were chosen on the basis that a lower value indicates that a procedure carries less overall risk, thereby providing a simple and uniform way to indicate the quality of a procedure.

- $k_{crit.steps} = \frac{n_{steps,crit.}}{n_{steps,total}}$

Ratio of procedure steps while in critical state to total number of procedure steps, ranging from zero (best) to one (worst case, i.e. all procedure steps are critical).

- $k_{sync} = \frac{n_{sync,total}}{n_{steps,total}}$

Ratio of total number of synchronization points to total number of procedure steps (lower means less OWLT-dependence).

- $k_{sync,explicit} = \frac{n_{sync,explicit}}{n_{steps,total}}$

Ratio of number of explicit synchronization points to total number of procedure steps (a value of zero means no explicit calls to the ground are made).

- $k_{sync,implicit} = \frac{n_{sync,implicit}}{n_{steps,total}}$

Ratio of number of implicit synchronization points to total number of procedure steps (a value of zero means no commands are sent).

- $k_{crit.sync} = \frac{n_{sync,total}}{n_{steps,crit.}}$

Ratio of total number of synchronization points to procedure steps while in critical state (only applicable if $n_{steps,crit.} > 0$; cf. $k_{sync}$).

- $k_{crit.sync,explicit} = \frac{n_{sync,explicit}}{n_{steps,crit.}}$

Ratio of number of explicit synchronization points to procedure steps while in critical state (only applicable if $n_{steps,crit.} > 0$; cf. $k_{sync,explicit}$).

- $k_{crit.sync,implicit} = \frac{n_{sync,implicit}}{n_{steps,crit.}}$

Ratio of number of implicit synchronization points to procedure steps while in critical state (only applicable if $n_{steps,crit.} > 0$; cf. $k_{sync,implicit}$).

### Application to Columbus procedures

The characteristics and factors are applied to example procedures that are used in the day-to-day operations at Col-CC. This means analyzing and understanding the steps of the procedures, annotating them with status information, i.e. adding annotations that indicate at which instructions a critical state is entered and later left again, as well as decomposing the procedures, forming groups of logically related instructions.

While characterizing procedures, several rules must be followed that derive from the dynamic nature of manned space operations. During real-world operations, one major aspect concerning the execution of procedures is that the same procedure may be executed in a different way at different times. This may be due to a procedure being applicable to various distinct but mostly identical systems, or – more often than not – the execution depending on some system state and operational considerations.

This includes situations where e.g. only part of a procedure is scheduled to be executed or a sub-procedure is only called when a telemetry item is not in the expected value range.

These rules to be applied during characterization of procedures are:

1. Assume that every instruction that can be executed by both ground and crew is in fact executed by the ground (save crew work time).

2. Always include all steps of a procedure, even if they may not be applicable in some situations. If a step contains an instruction to end the execution of the procedure at this point, ignore this instruction and proceed with the execution to the end of the procedure.

3. For any conditional block, assume that the instructions within that block are part of the regular flow of instructions, i.e. any condition is assumed to be true. This does not apply to variants guarded by conditional blocks where only one condition can ever be true – in this case, count the block with the most read-write instruction combinations. As a general rule, always assume the worst case when scoring procedures. Exceptions to this general rule are conditional blocks with a repeat instruction, which may only be counted once.

4. Whenever another procedure is called out in a step, the steps and sub-steps of this sub-procedure must be considered as if they were part of the calling procedure. This rule does not apply to sub-procedures that effectively abort the procedure, e.g. contingency shutdowns of hardware due to certain error conditions. Likewise if a step in a sub-procedure contains an instruction to end the execution of the procedure at this point, ignore this instruction and only return to the calling procedure after the last step of the sub-procedure (cf. rule 2).

To derive the procedure impact characteristics and factors the following steps are followed:

1. Annotate the procedure with flags to indicate at which instruction a critical system state is entered and left again.

2. Count the number of procedure steps and critical steps per the rules outlined above.

3. List all executable instructions of the procedure regardless of their associated steps considering the rules described above. Form new groups of instructions that can be executed in sequence without requiring an intermediate round-trip.

4. Count and summarize the number of implicit and explicit synchronization points.

5. Calculate the impact factors as defined above.

As an example, these steps are applied to a type of procedure that is very common for Col-CC operations: Activation of a rack, namely the HRF-1 rack of NASA, which, due to a barter agreement, is located inside Columbus – this combination renders the procedure a bit more complex than comparable ones for ESA-owned racks. The procedure is registered as «1.305 HRF1 ISPR RACK ACTIVATION» (version FIN 8).

### Calculation Results

Results of using the impact quantification process described above are listed in Tab. 2. Results of procedures marked with an asterisk (*) only cover the procedure itself and do not include any sub-procedures for simplicity; instead, every call-out to a sub-procedure is counted as one additional implicit synchronization point.

Tab. 2: Impact factors of various procedures

| Procedure | 1.305 | 9.102 | 2.101* | 9.401* | 9.414* |
|---|---|---|---|---|---|
| $n_{steps,total}$ | 15 | 19 | 35 | 58 | 57 |
| $n_{sync,explicit}$ | 7 | 0 | 6 | - | - |
| $n_{sync,implicit}$ | 9 | 16 | 29 | 105 | 54 |
| $n_{sync,total}$ | 16 | 16 | 35 | 105 | 54 |
| $k_{sync}$ | 1.067 | 0.842 | 1 | 1.810 | 0.947 |
| $k_{sync,explicit}$ | 0.467 | 0 | 0.171 | - | - |

| | | | | | |
|---|---|---|---|---|---|
| $k_{sync,implicit}$ | 0.6 | 0.842 | 0.829 | 1.810 | 0.947 |

For comparison, the table also includes results for further example procedures. The procedures were selected based on the following considerations:

- 9.102 CHECKOUT OF BACK-UP WATER LOOP (WPA2): A simple but extensive ground-only (i.e. cannot be performed by crew) procedure, which is executed weekly to ensure that a critical back-up system behaves nominally.
- 2.101 TCS LOOP WPA1 TO WPA2 SWITCHOVER – AUTO: A highly complex and potentially long-running procedure that concerns critical equipment.
- 9.401 TCS LOOP WPA1 TO WPA2 SWITCHOVER – MANUAL: Same as 2.101 but with less on-board automation, i.e. more manual commanding by ground.
- 9.414 WATER LOOP FDIR SWITCHOVER: Similar to 2.101 resp. 9.401 but manually initiating the failover procedure that is triggered by the autonomous on-board monitoring system in certain contingency situations.

## Analysis and Discussion of Operations

Analysing the impact of the signal propagation time is a very complex and manifold topic. Even the simple characteristics and rules to quantify this impact on procedures as proposed can only hint to the real-world impact on operations as the highly dynamic nature and complex interactions of manned space operations are hard to limit to a restricted scope. Nonetheless, analysing a restricted aspect of these operations proofs to be useful and helps in painting the bigger picture. The scope chosen for this study limits the observations to the ground operations at Col-CC and the execution of its procedures.

ISS operations are heavily reliant on near-real-time and near-continuous communication between the space and ground segments. On the other hand, critical systems aboard the station are constantly monitored by on-board computers, which are able to react to a limited set of off-nominal situations and automatically recover critical functionality resp. switch to redundancies. But even then the operators on the ground will have to react to such contingencies quickly, as a safe mode of operations cannot be sustained for some hardware without further manual commanding. And while operations on-board the station will not come to an instant halt for longer and unexpected drop-outs in communication (e.g. one which is comparable to the longest round-trip times encountered during Mars missions), they will certainly switch to contingency mode sooner than later, i.e. disabling non-essential functionality. This is especially true when a sudden off-nominal behavior is detected and the ground is unable to investigate in a timely manner.

Investigating and recovering from such contingency operations is an arduous and lengthy process, which the on-board crew is unable to conduct autonomously in most cases due to the limited interfaces available to the crew. Incidentally, the multi-national architecture of the space station – in combination with the experience of the ISS astronauts – provides enough redundancies for the main communications stack to soften the effect of a temporary data transmission outage. This is however not fully applicable to inter-planetary missions with considerable signal propagation times.

It follows that the design for interplanetary spacecraft must allow for a significant increase in on-board autonomy. More specifically, the autonomous systems monitoring and control shall not only ensure the safety of crew and vehicle, but also the continuous operation in nominal mode as much as possible. The Data Management System of Columbus and its layered architecture provide a good stepping stone to achieve this autonomy.

The crew interfaces available in Columbus are however insufficient as they do not provide nearly the same capabilities the ground has. For missions with a considerable OWLT, the trade-off made for the design of Columbus between usability and cognitive strain on the one hand side and extensive on-board autonomy and complexity on the other one is not acceptable.

Current ISS operations focus on maximizing crew time for research and minimizing their time spent with maintaining and operating station equipment. The same principle is foreseeable for inter-planetary missions, at least in their primary mission phase, during which all systems shall operate in nominal mode. Any failure to do so will incur major cost by keeping the crew from fulfilling their mission. All on-board systems must thus be highly reliable and well understood in order to minimize the time spent analyzing a fault and fixing the system.

Further essential knowledge for continuous nominal operations is the mutual influence of system components and hence the global system state, which has to be verified against the expected state especially during commanding. For Columbus systems, this is done manually by the flight controllers, who then need to actively reconfigure the automatic on-board monitoring systems during the execution of procedures. This currently used approach entails operational risk, as the system can only ensure the nominal and safe functionality for equipment it is currently tasked to monitor.

Scenarios where system failures may remain undetected for too long have been mentioned above: E.g. during rack activation, there is a significant time gap, including a significant number of round-trips, between power activation and enabling of smoke detectors. Only the operator knows that the system moves to a critical state and that this state is only left when the appropriate monitoring is armed manually.

For future missions with a high demand for system autonomy, the system shall know when a critical system state is entered and left again, so it can effectively monitor these important state transitions. To provide even more situational awareness to the on-board monitoring systems, it may be helpful to convey the intended final outcome of a string of commands at the beginning of a procedure, so the system can monitor the procedure execution itself. This applies to remotely controlled operations in particular, as the time gap between ground monitoring and sending commands increases with the OWLT.

Another aspect of system reliability is complexity. While flight controllers work hard to minimize operational complexity of system procedures, it is not always possible to do so. Furthermore, there are no objective measures to identify the complexity of a procedure and track it during procedure development and revision. The procedure characteristics and impact factors proposed in this study may provide such metrics and aid in optimizing existing procedures.

The optimization goal is to reduce the impact factor values as much as possible while not increasing the total number of steps, i.e. reducing the number of synchronization points. This can be achieved in numerous ways, e.g. by grouping sets of write-only instructions and thereby separating them from read-only instructions, which ultimately leads to less round-trips being made.

Another way is to move some or all status checks into more complex on-board programs that also include the commands as specified in the procedure. The effect of such a measurement can be seen in the vastly improved impact factors (especially $k_{sync,implicit}$) of procedure 2.101 compared to 9.401 as listed in Tab. 2, with the former providing more system autonomy.

However, rewriting procedures to reduce impact factors may also result in an increased operational risk. This is especially true for the naïve approach of grouping all commands into one block and moving all post-command verification steps to the end of the procedure: “Blindingly” executing commands in an unknown system state may result in undefined behavior and is thus associated with higher risk than executing a series of combined command-verify steps one-by-one. This risk is not covered by the impact factors proposed herein.

Rewriting procedures may also require modifications to on-board systems. At the very least, this means adding software facilities to automatically roll-back commands if certain status checks fail and provide the ground with detailed execution analysis. Some systems may even require major hardware modifications to be able to track system state by themselves and always operate in a safe configuration in between a group of multiple consecutive commands.

The impact factors proposed in this study provide a basis for assessing different aspects of procedure quality. First and foremost, this is the OWLT-dependence of procedures. The factors and characteristics are limited – nevertheless, their concept can easily be extended on and further measurements added.

Their limitations are however not restricted to the number of measurements but also the rules that must be applied as described. It is very difficult to represent complex cases like multiple sub-procedure end points (i.e. instructions in sub-procedures to abort a procedure execution or jump back to the calling procedure) or dynamic procedure execution (e.g. limiting to a selection of procedure steps). Especially difficult to represent are procedures that rely on procedure or environmental parameters that change the way of the instruction flow, e.g. procedures that skip over steps depending on the result of a command execution.

The characterization steps and rules are intentionally simplistic so they can easily be implemented in software and thus allow automatic characterization of procedures. This may enable procedure maintainers to automatically track changes in quality whenever procedures are updated without incurring procedural overhead in the change workflow. Additionally, automating the scoring of procedures allows relaxing some of the characterization rules, e.g. by adapting the impact factors dynamically depending on the system state and planned activity.

## Conclusion

In the last two years Col-CC proved that it could handle a number of high activity phases in a row with 4 ESA astronauts on ISS in a short timeframe. This phase was also used to install new challenging payloads on-board Columbus such as PK-4 and EML and to take over new tasks outside the Columbus module (see [14]). In addition to the on-board activities, updates in the ground facilities like server virtualization and transition to the new NASA planning software was supported and implemented, respectively.

Despite some initial difficulties to fulfil all requirements and the need of some adaptation to the timeline and restriction to operations, almost all objectives of the increments could be achieved and both increments were successfully performed.

Furthermore, Col-CC is looking into the future and is preparing itself for possible future endeavours. In the study, the current setup of low-latency operations was applied to case studies of deep space missions with significant delays in communications.

Currently, the Columbus systems architecture, its operations concept, and the procedures indicate a substantial reliance on real-time operations, which is not surprising given the ISS mission design. However, Columbus does include some autonomy features to increase independence from ground: The module is trying to react to system failures in the most graceful way possible. In case of failure of switching to a safe state, ultimately immediate crew evacuation from the module is required – which may not be an option in future long-distance space missions.

This study introduces a simple method to quantify the impact of the OWLT on the execution of procedures. Measuring the characteristics of procedures commonly used in human spaceflight by the ground as proposed herein enables operators to easily determine if these procedures are suitable for potential future mission with a significant communications delay. Furthermore, the same characteristics permit procedure authors and maintainers to reliantly track the effect changes to procedures have on their delay tolerance.

Additionally, the findings are also applicable to today's low-latency ISS operations. The characteristics described, which are not measured today, allow Col-CC to quantify some key aspects of complexity in procedures.

Especially with further procedure annotations, this may lead to interesting applications in the near future, both for manual and automated quantification: E.g. procedure maintainers can track procedure complexity (including no. of steps between entering and exiting critical system states) and try to reduce it; planners may be notified about high complexity of newly introduced procedures and allocate a more suitable time slot for their execution (for automated quantification: auto-optimize time slot based on subset of steps in execution notes); and operators may be better informed about system state or even assisted in choosing responses to system behavior (annotations allow operator or state machine to easily track critical system state).

In summary, these findings might help quantifying and reducing the operations risk at Col-CC.

## References

[1] Kuch, T.; Sabath, D.; Fein J.: Columbus-CC – *A German Contribution to the European ISS Operations*, IAC-05-B4.2.08, 56th International Astronautical Congress, Fukuoka, Japan, 2005

[2] Sabath, D.; Kuch, T.; Fein J.: *Das Columbus-Kontrollzentrum in Oberpfaffenhofen*, DGLR-2005-153, DGLR Jahrestagung 2005, Friedrichshafen, Germany, 2005

[3] Sabath, D.; Nitsch, A.; Hadler, H.: *Columbus Operations – Joint undertaking between DLR and Industry*, SpaceOps 2006 Conference, AIAA 2006-5807, Roma, 2006

[4] Kuch, T., Sabath, D.: *The Columbus-CC—Operating the European laboratory at ISS,* 58th International Astronautical Congress, Hyderabad, 2007

[5] Sabath, D.; Hadler, H.: *Management and shift planning of the COL-CC Flight Control Team for continuous Columbus Operations*, SpaceOps 2008 Conference, AIAA 2008-3395, Heidelberg, 2008

[6] Sabath, D.; Schulze-Varnholt, D.: *First Experience with Real-Time Operations of the Columbus Module*, 59th International Astronautical Congress, IAC-08-B3.3.3, Glasgow, 2008

[7] Sabath, D.; Schulze-Varnholt, D.: *One Year of Columbus Operations and First Experience with 6 Persons Crew*, 60th International Astronautical Congress, IAC-09.B3.3.1, Daejon, 2009

[8] Sabath, D.; Nitsch, A.; Schulze-Varnholt, D.: *Highlights in Columbus Operations and Preparation for Assembly Complete Operations Phase*, 61st International Astronautical Congress, IAC-10.B6.1.5, Prague, 2010

[9] Sabath, D.; Nitsch, A.; Schulze-Varnholt, D.: *Changes in Columbus Operations and Outlook to Long-term Operation Phase*, 62nd International Astronautical Congress, IAC-11.B3.4.-B6.6.2, Cape Town, 2011

[10] Sabath, D.; Soellner, G.; Schulze-Varnholt, D.: *Development and Implementation of a New Columbus Operations Setup*, 63rd International Astronautical Congress, IAC-12.B3.4-B6.5.1, Naples, 2012

[11] Sabath, D.; Söllner, G.; Schulze-Varnholt, D.: *First Experience with New Col-CC Console Setup*, 64th International Astronautical Congress, IAC-13.B3.4-B6.5.3, Beijing, 2013

[12] Sabath, D.; Kuch, T.; Söllner, G.; Müller, T.: *The Future of Columbus Operations*, SpaceOps 2014 Conference, AIAA 2014-1618, Pasadena, 2014

[13] Baklanenko, M.; Sabath, D.; Söllner, G.: *New Col-CC Operations Concept and New Challenges*, 65th International Astronautical Congress, IAC-14,B3,4-B6.5.4, Toronto, 2014

[14] Leuoth, K.; Sabath, D.; Söllner, G.: *Consolidating Columbus Operations and Looking for New Frontiers*, 66th International Astronautical Congress, IAC-15,B3,4-B6.5.3, Jerusalem, 2015

[15] Bender, F.: *Impact of Communications Delay on Col-CC Manned Space Operations*, LRT-SA 2016/02, München, 2016

[16] Trebbin, N.: *Virtualization of the Columbus Control Room Infrastructure*, 14th International Conference on Space Operations, AIAA 2016-2578, Daejon, 2016

[17] Singer, T.: *Implementation of the True Multi-Mission Control Room at GSOC*, 14th International Conference on Space Operations, AIAA 2016-2454, Daejon, 2016

[18] *A Piloted Flight to a Near-Earth Object: A Feasibility Study*, Houston, TX, 15th June 2007.

[19] D. R. Dr. Williams, *Mars Fact Sheet*, 29th February 2016. [Online]. Available: http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html. [Accessed: 18-Apr-2016].

[20] *Europa (moon) distance to Earth in 16th Dec 1995 in lightminutes - Wolfram|Alpha* [Online]. Available: http://www.wolframalpha.com/input/?i=europa+(moon)+distance+to+Earth+in+16th+Dec+1995+in+lightminutes. [Accessed: 01-Jun-2016].