



HAL
open science

The Ingenuity Mars Helicopter Specified and Analyzed with the Real-time Mode-aware Dataflow Model

Guillaume Roumage, Selma Azaiez, Cyril Faure, Stéphane Louise

► **To cite this version:**

Guillaume Roumage, Selma Azaiez, Cyril Faure, Stéphane Louise. The Ingenuity Mars Helicopter Specified and Analyzed with the Real-time Mode-aware Dataflow Model. 2025. hal-04881272

HAL Id: hal-04881272

<https://hal.science/hal-04881272v1>

Preprint submitted on 11 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Ingenuity Mars Helicopter Specified and Analyzed with the Real-time Mode-aware Dataflow Model

Guillaume Roumage[†], Selma Azaiez[‡], Cyril Faure[‡], Stéphane Louise[‡]

[†]guillaume.roumage.research@proton.me

^{†‡}Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France

[‡]firstname.lastname@cea.fr

Abstract—Ingenuity is an autonomous Cyber-Physical System (CPS) that has successfully completed more than 70 flights over Mars between 2021 and 2024. Ensuring the safety of its mission is paramount, as any failure could result in catastrophic economic damage and significant financial losses. Dataflow Models of Computation and Communication (DF MoCCs) serve as a formal framework for specifying and analyzing the timing behavior of such CPSs. In particular, the Real-time Mode-aware Dataflow (RMDF) model is highly suitable to specify and analyze real-time and mode-dependent Cyber-Physical Systems (CPSs) like Ingenuity. This paper showcases the application of RMDF for the specification and analysis of Ingenuity. We propose a dataflow specification of Ingenuity, analyze its timing behavior, and provide a feasibility test. Finally, we proposed a plausible explanation of the timing anomaly that occurred during the sixth flight of Ingenuity.

Index Terms—Real-time Mode-aware Dataflow, Ingenuity Mars Helicopter, Dataflow Model, Real-time

I. INTRODUCTION

CPSs are reactive systems that detect environmental shifts through sensors, process this information using computational processes, and then use the output to control actuators. CPSs range from digital signal processing systems to embedded/cloud infrastructures, soft/hard real-time systems, and even a mix of all the above. These complex systems must operate reliably without threatening their internal processes. For example, a failure of the actuator in an autonomous car can lead to catastrophic consequences such as a car crash or a pedestrian accident.

The Ingenuity Mars helicopter is a real-time and autonomous CPS. It is a small coaxial helicopter designed by the National Aeronautics and Space Administration (NASA). It has completed 72 flights over Mars from April 2021 to January 2024. Ingenuity is composed of sensors - an altimeter, two Inertial Measurement Units (IMUs), and a camera - and actuators - eight motors. The sensors provide measurements to the navigation system, which estimates the state of Ingenuity. The control system then uses the state to keep Ingenuity as close as possible to a reference trajectory. A set of waypoints provides the reference trajectory. The control system sends commands to the motors to adjust the altitude, heading, and translation of Ingenuity.

The safety of Ingenuity is a critical issue. The safety of a CPS is the guarantee that the system will not endanger itself

or its environment or lead to substantial financial losses. As an illustration of this latter point, the Ingenuity Mars mission cost raised \$85 million dollars [1].

In this paper, we proposed to use Dataflow Model of Computation and Communications (DF MoCCs) formalisms [2] to analyze the safety of Ingenuity. These formalisms permit the specification of a system with an oriented graph where actors represent computational units and channels represent communication links. DF MoCCs also provide static analyses to ensure the safety of the system’s specification. For instance the Real-time Mode-aware Dataflow (RMDF) model is highly suitable to specify and analyze real-time and mode-dependent CPSs like Ingenuity. RMDF can be used to guarantee, prior to execution, the existence of a memory-bounded execution (consistency), the existence of a deadlock-free execution (liveness), and to derive the timing behavior of Ingenuity.

1) *Contribution*: This paper presents the specification of the Ingenuity Mars helicopter with the dataflow model RMDF [3]. Besides guaranteeing the consistency and liveness of the system, RMDF allows the derivation of the timing behavior of Ingenuity. A plausible explanation of the timing anomaly that occurred during the sixth flight of Ingenuity¹ is also given in this paper.

2) *Paper Organization*: This paper starts by presenting the background and terminology of the RMDF model in Section II. As RMDF extends POLYGRAPH [4], which itself extends Synchronous Dataflow (SDF) [5], those two dataflow models are also presented. The Ingenuity mars helicopter use case is presented in Section III with its overview in Section III-A, its RMDF specification in Section III-B, its timing behavior analysis in Section III-C, and its feasibility test in Section III-D. The timing anomaly of the sixth flight of Ingenuity is explained in Section IV. Finally, Section V concludes the paper.

II. BACKGROUND AND TERMINOLOGY

RMDF [3] is an extension of POLYGRAPH [4], which is itself a superset of the SDF model [5]. In this section, we

¹<https://science.nasa.gov/missions/mars-2020-perseverance/ingenuity-helicopter/surviving-an-in-flight-anomaly-what-happened-on-ingenuitys-sixth-flight/>

present SDF, and then we explain how the POLYGRAPH model expands it.

A. The SDF Model

An SDF specification is an oriented graph $G = (V, E)$ where V is a finite set of *actors* and E is a finite set of *channels*. An SDF specification is characterized by a *topology matrix* G_Γ of size $|E| \times |V|$. An *actor* is a computational unit that both produces and consumes data tokens every time it is executed. The atomic amount of data exchanged is known as a *token*. The entry (i, j) of G_Γ is the number of tokens produced or consumed by the actor v_j on the channel c_i each time the actor v_j is executed (this number is positive if the tokens are produced, and negative otherwise). An execution of an actor is called a *job*.

An actor $v \in V$ has (optional) input and output ports connected to input and output channels of E . A channel $c_i = (v_j, v_k, n_{ij}, n_{ik}, init_i) \in E$ connects an output port of the actor $v_j \in V$ to an input port of the actor $v_k \in V$. This channel also has a production rate $n_{ij} \in \mathbb{N}^*$ (which is the entry (i, j) of G_Γ), a consumption rate $n_{ik} \in \mathbb{N}^*$ (the entry (i, k) of G_Γ), and a number of initial tokens $init_i \in \mathbb{N}$. An actor's job produces/consumes tokens to/from the buffer of its output/input channels according to the production/consumption rate. For the sake of simplicity in the rest of this paper, we denote $[c_i]$ the number of initial tokens of the channel c_i .

B. The POLYGRAPH Model

1) *Syntax and Semantic*: The POLYGRAPH model is a superset of the SDF model and expands it in two ways: the production and consumption rates become periodic sequences (as in CSDF [6]), and a subset of actors may have timing constraints. The terminology of the static analysis of an SDF specification, i.e., consistency, liveness, consistent and live execution, and hyperperiod [5], is extended to a POLYGRAPH specification. Consistency ensures the system can be executed within a bounded memory, while liveness guarantees deadlock-free execution. A hyperperiod is a partially ordered set of actors' jobs that returns the system to its initial state within a given time frame.

a) *Rational production and consumption rates*: Let $G = (V, E)$ be a POLYGRAPH specification and let $G_\Gamma = (\gamma_{ij}) \in \mathbb{Q}^{|E| \times |V|}$ be its topology matrix. The production and consumption rates are rational²: a channel $c_i \in E$ is a tuple $(v_j, v_k, \gamma_{ij}, \gamma_{ik}, [c_i])$ such that $\gamma_{ij} \in \mathbb{Q}^*$ and $\gamma_{ik} \in \mathbb{Q}^*$, and $[c_i] \in \mathbb{Q}$ ($\mathbb{Q}^* = \mathbb{Q} \setminus \{0\}$). Rational rates imply that the number of tokens produced/consumed can differ for each job. Specifically, a token is produced and stored in a channel if and only if sufficient fractional token parts are symbolically produced. Indeed, only an integer number of tokens may be produced or consumed. In other words, an actor with a rational production rate such as $1/n$ in the dataflow model actually produces data once every n jobs during runtime, thus

²A channel of a POLYGRAPH specification as defined in [4] has at least one integer rate. However, for the sake of simplicity, we consider in this paper that both rates are rational.

validating the precedence constraint toward the consumer once every n of its jobs.

The rational production and consumption rates are natural consequences of the periodicity of real-time actors (with specified and imposed frequencies) together with the consistency of the system's model imposed by the topology matrix. The rational rates are a compact notation for a CSDF equivalency [6].

b) *Timing constraints*: The actors of a POLYGRAPH specification may have a frequency constraint. Furthermore, they may also have a phase if they have a frequency constraint. The phase usually models the system's end-to-end latency. The frequency dictates the actor's execution at the specified rate, while the phase postpones its initial execution. Actors with frequency constraints are referred to as *timed actors*. An execution of a consistent and live POLYGRAPH specification is an infinite repetition of its first hyperperiod. Although the number of jobs is not bounded during an execution, their timing constraints are cyclic over a hyperperiod.

C. The RMDF model

1) *Mode-Dependent Execution*: The RMDF model is a superset of the POLYGRAPH model. Besides the ability to specify CPSs with relaxed real-time constraints, RMDF can specify CPSs with mode-dependent execution. A mode-dependent CPS has a set of conditional execution branches. The choice of the execution branch to execute is usually based on runtime data. To illustrate, the RMDF specification of Fig. 1 has three conditional executions branches composed of actors $C_1, \dots, C_{n_1}, D_1, \dots, D_{n_2}$, and E_1, \dots, E_{n_3} .

2) *Splitters and Joiners*: The RMDF model introduces two new actors: the *splitter* and the *joiner*. A *splitter* is an actor that consumes a data token from its input channel and produces it on its output channels in the lexicographic order. A predefined number of tokens is sent to the first channel in the lexicographic order of the output channels, then to the second channel, and so on. The number of tokens sent to an output channel is the numerator of the production rate of that channel. An execution of a splitter consumed and produced one token at a time. A similar behavior holds for a *joiner*, and the interested reader is referred to [3] for more details.

3) *Mode Deciders, Controlled Splitters and Control Joiners*: The RMDF model introduces three new actors: the *mode decider*, the *controlled splitter*, and the *controlled joiner*. The *mode decider* is responsible for deciding the execution branch to execute (actor B of Fig. 1 is a mode decider). A *controlled splitter* and a *controlled joiner* are used to route data tokens to/from the execution branches. The *mode decider* is connected to the *controlled splitter* and the *controlled joiner* through a *control channel* that carries a *control token*. This control token is used to decide from/to which channel tokens are consumed/produced by assigning a value to parametric rates. In Fig. 1, those parametric rates are m_1, m_2 and m_3 . In other words, the *controlled splitter* and the *controlled joiner* have an *switch-case* structure that decides how a data token is produced and consumed.

4) *Control areas*: The RMDF model introduces the concept of *control areas*. A control area is a set of actors that are conditioned by a *mode decider*. The control area of the mode decider B of Fig. 1 is composed of actors C_1, \dots, C_{n_1} , D_1, \dots, D_{n_2} , and E_1, \dots, E_{n_3} . In order to ensure the static analysis of an RMDF specification, restrictions are applied on control areas. Those restrictions are detailed in Property 1 of [3].

III. THE INGENUITY MARS HELICOPTER

The main contribution of this paper is the specification of the Ingenuity Mars helicopter with the RMDF model. Section III-A provides an overview of Ingenuity, Section III-B presents the RMDF specification of Ingenuity, Section III-C presents the timing behavior analysis of Ingenuity and Section III-D presents a feasibility test for Ingenuity.

A. Overview of Ingenuity

Ingenuity operates independently due to communications delays of at least 5 minutes between Earth and Mars³. Hence, real-time navigation from Earth is impossible. Ingenuity uses a Kalman filter [7] to fuse measurements from three sensors (cf. Fig. 2, with a redundancy of the IMU) and regularly updates its *state*, i.e., its position, velocity, attitude, and its angular rate. Ingenuity has two IMUs for redundancy, which measure 3-axis acceleration at 1600 Hz and angular rates at 3200 Hz, and an inclinometer is used before flights to calibrate IMU biases. Additionally, an altimeter measures the distance to the ground at 50 Hz, and a downward-looking navigation camera provides images at 30 Hz [8]. Ingenuity’s actuators consist of eight motors - four for each rotor. One motor keeps the rotation speed constant, while the remaining motors control the blade pitch, affecting Ingenuity’s motion.

1) *Vision-Based Navigation*: Ingenuity uses a *dead reckoning* process to compute its current state: this navigation technique estimates the current vehicle’s state based on previous sensor measurements. However, the dead reckoning process is susceptible to drift. To counter this issue, Ingenuity uses a vision-based navigation system. Periodically, it captures images of the Martian ground, detects visual features, and tracks them from one image to the next. Some images are regularly classified as *base frames*, and the features in these images become the *pseudo landmarks*, while other images are classified as *search frames* whose features will be computed and compared to the pseudo landmarks. Therefore, Ingenuity can determine a visual shift and correct the drift in the dead reckoning process by comparing the current features with the most recent pseudo landmarks. A new base frame is generated if certain conditions are not met, including reaching the maximum number of search frames between two consecutive base frames or a drop in the number of tracked features (tracked features are features from a search frame that are successfully matched with pseudo landmarks) [9].

2) *Navigation Algorithm*: The IMU measurements undergo pre-processing with filters, followed by integration and down-sampling from 3200 Hz and 1600 Hz to 500 Hz. The down-sampled IMU measurements are corrected with IMU biases computed from the previous state estimation, then sent to the navigation filter component and buffered for future use by a state propagation process. The navigation filter operates as a Kalman filter and fuses the IMU data, altimeter measurements, and vision system results to estimate Ingenuity’s state. The resulting state is sent back to the state propagation process, which propagates the state, i.e., predicts the state for the next time steps, using the buffered IMU data. The control system of Ingenuity is responsible for keeping the helicopter as close as possible to a reference trajectory, given as a set of waypoints: each of them specifies an (x,y)-position, a ground-relative altitude, a translation heading, a dwell heading, and a dwell time. Ingenuity advances between waypoints by performing a sequence of maneuvers: (1) altitude adjustment, (2) turn to specified flight heading, (3) horizontal translation, (4) turn to dwell heading, (5) pausing for the specified dwell time [10].

B. An RMDF Specification of Ingenuity

We have translated the textual specifications of Ingenuity [8]–[13] to an RMDF specification depicted in Fig. 3. Actors *IMU*, *Altimeter*, and *Camera* are the three sensors of Ingenuity, with frequencies of 500 Hz, 50 Hz, and 30 Hz, respectively. Ingenuity downsamples its *IMU* data from 3200 Hz and 1600 Hz to 500 Hz, so we assume this downsampling is already done. The specification of Ingenuity of Fig. 3 is consistent and live, as verified by our internal scripts implementing the static analyses of RMDF specifications [3].

1) *Vision System*: The vision system is depicted in the top tier of Fig. 3. The actors from *Camera* to *Feature Match* belong to the vision system of Ingenuity. The *Laber Decider* is responsible for classifying a frame as a base or search frame. In the real system, a new base frame is triggered if at least one of the following conditions is not met [9]: (1) the number of tracked features is below a threshold, (2) the number of search frames since the last base frame is above a threshold, (3) visual features are not well-distributed. Note that those conditions cannot be predetermined at design time as they depend on the frame’s environment and quality. RMDF allows such dynamic behavior to be specified.

The decided mode is sent to the control port of the *Controlled splitter* and the *Controlled joiner* through *Duplicator 2*. *Controlled splitter* and *Controlled joiner* enclose an *if-then-else* structure that decides how a frame is processed. In the *then* branch, the features of a search frame are tracked by *Feature Tracking* before being filtered with a *Filtering Procedure*. In the *else* branch, the features of a base frame are used to update the *Pseudo Landmarks*.

There are two conditional branches: *Feature Tracking* → *Filtering Procedure* and *Pseudo Landmarks*. The parametric rates m_1 and m_2 on channels c_2, c_3, c_4 and c_5 determine whether or not the conditional branch is taken during an iteration. The *Controlled splitter* and *Controlled joiner* use

³<https://mars.nasa.gov/mars2020/spacecraft/rover/communications/>

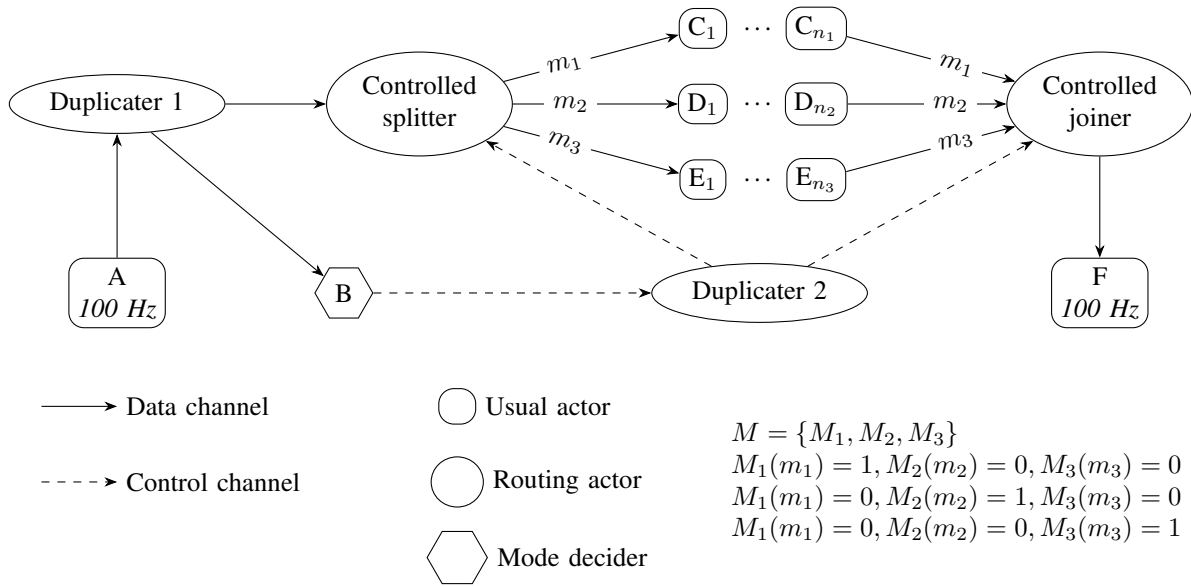


Fig. 1: An example of an RMDF specification. A usual actor is an actor which is neither (non-) controlled splitter or joiner, nor a mode decider.

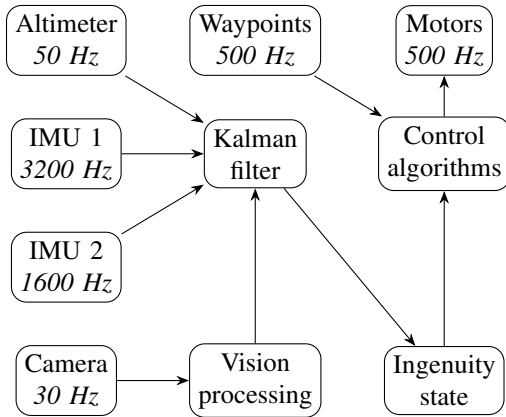


Fig. 2: An overview of the Ingenuity helicopter.

the control token to decide from/to which channel tokens are consumed/produced. Finally, the self-loop on *Feature Tracking* models the continuous tracking of features from one search frame to the next.

2) *Navigation System*: The actors from *IMU* and *Altimeter* to *State Propagation* belong to Ingenuity’s navigation system. The *Navigation Filter* inherits the frequency of 500 Hz from the *IMU*. Thus, it has to oversample the vision processing result with a consumption rate of 3/50 as the frequency of the *Camera* is 30 Hz. The same logic applies to the channel *Duplicater 4* \rightarrow *Navigation Filter* and *Duplicater 4* \rightarrow *State Propagation*.

The following paragraph details the origin of the initial token of 1/50 on the channel c_6 . *Motors* operates at a frequency of 500 Hz (i.e., 2 ms = 6 · (1/3) ms) with a phase of 1 ms. As a result of the liveness analysis, a tick in the symbolic execution of the Ingenuity specification is equivalent to a time advance

of 1/3 ms. Consequently, *Motors* is required to execute at tick τ if $\tau \equiv 3 \pmod{6}$. At tick 99, *Motors* must execute for the 17-th time, and *Navigation Filter* must also execute for the 17-th time due to data dependencies. Therefore, $17 \cdot (3/50)$ tokens are required on c_6 . However, at the 99-th tick, *Camera* has executed only once. Its second execution occurs later, at the 100-th tick. Thus, there is only one token on c_6 , which is not enough for the 17-th execution of *Navigation Filter* and a deadlock occur. The initial token of 1/50 on c_6 avoids this deadlock by enabling the 17-th execution of *Navigation Filter*.

The *Navigation Filter* is the core of the navigation system of Ingenuity: it fuses the measurements from the sensors to estimate the state of Ingenuity. Then, *State Propagation* is responsible for predicting the state of Ingenuity for the next time steps. Note that *IMU-Correction* corrects the *IMU* measurements with the biases computed from the previous state estimation. The integration of the *IMU* data is redundant with *NC - IMU - Integration* and *FC - IMU - Integration*. The integration is performed in two hardware units, denoted as the *Navigation Computer* and *Flight Computer* in Ingenuity’s specifications. We do not detail the difference between these two units here; the interested reader is referred to [10]. An initial token on the channel c_9 models the initial bias correction. An inclinometer initializes the *IMU* biases before the flight in the real system.

3) *Control System*: The actors *Control Altitude*, *Control Yaw 1*, *Control Translation*, and *Control Yaw 2* are responsible for triggering Ingenuity’s motors to keep it as close as possible to the reference trajectory. This latter is provided by a set of waypoints produced by *Waypoints*. The four control actors compare the current state of Ingenuity with the expected state and send the appropriate commands to the motors. Fig. 3b provides details on the production and consumption rate

specifications. They are chosen such that, during an execution, *Control Altitude* is first executed, then *Control Yaw 1* adjusts the heading, followed by the *Control Translation*, and finally, another heading adjustment with *Control Yaw 2*. Then, the cycle repeats.

4) *Fidelity of the Ingenuity Specification*: In the vision system, we have simplified the labeling process of frames. In the real system, a frame can be classified as both a base frame and a search frame, e.g., when the number of tracked features is below a threshold. In such cases, the current features replace the pseudo landmarks, causing a frame initially classified as a search frame to be also classified as a base frame. Our specification operates on the assumption that a frame can only be a base or search frame, but never both simultaneously. As a result, continuous tracking of features from a search frame to a base frame is not supported by our specification. As a future work, this could be improved by adding another mode that would allow a frame to be sent to multiple conditional branches.

The navigation system assumes that the IMU data are already downsampled to 500 Hz. However, in the real system, the IMU data are downsampled from 3200 Hz and 1600 Hz to 500 Hz. In addition, we simplify by assuming that the initial bias correction is already incorporated. In the real system, the inclinometer initializes the IMU biases before flight, and this initialization is represented by the initial token on c_9 in Fig. 3a.

In our specification, we implicitly operate under the assumption that movement during each maneuver (altitude, yaw, and translation) occurs almost instantaneously, although this does not accurately reflect real-world conditions. After performing the *Control Yaw 2* maneuver, Ingenuity also undergoes a specified wait time. Our model does not account for this waiting time. Furthermore, based on the specification, we know that the end-to-end latency of Ingenuity is 10 ms, which implies that *Motors* phase should be 10 ms. However, the RMDF model imposes a limitation where the phase must be strictly less than the period (as for the POLYGRAPH model). Due to this constraint, we have set the *Motors* phase to the maximum allowable value of 1 ms. It would be interesting to investigate how to relax this constraint in future work.

Finally, Ingenuity has different phases of flight, including *climb*, *waypoint tracking*, and *descent*, as mentioned in Table I. Each phase of flight involves turning on and off different sensors. Specifically, the *Camera* and *Altimeter* are disabled when Ingenuity is under 1-meter altitude to avoid erroneous measurements due to dust kickup from rotor downwash. Altitude thresholds initiate the transition from one phase to another. Ingenuity also has a fault response mode, a “land as soon as possible” mode. In this mode, the *Camera* is disabled, while the *IMU* and *Altimeter* remain enabled. The transition to the fault response mode is triggered when 50 consecutive packets containing vision processing results fail to arrive at *State Propagation* [10]. Our specification only considers the *Waypoints Tracking* phase, and there is no fault response mode.

TABLE I: Phases of flight of Ingenuity.

Phase of flight	Climb ¹	Waypoints Tracking	Descent ²	Fault Response Mode
Altitude threshold	< 1 meter	≥ 1 meter	< 1 meter	any
Camera	○	●	○	○
Altimeter	○	●	○	●
IMU	●	●	●	●

¹ *climb* contains also a *take off* phase where control of the helicopter is limited

² a monitoring of the vertical velocity is also enabled when altitude is less than 0.5 meters

C. Timing Behavior Analysis of the RMDF Specification of Ingenuity

The specification of Ingenuity presented in this paper is consistent and live, i.e., it has a memory-bounded and a deadlock-free execution. Those properties are verified using internal tools of the RMDF library. The timing propagation presented in [14] has been applied to the Ingenuity specification and result are presented in Table II, assuming Best-Case Execution Times (BCETs) and Worst-Case Execution Times (WCETs) are 0.12 and 0.20 ms, respectively, for all actors except *Controlled Splitter* and *Controlled Joiner* which have a constant execution time of 0 ms.

D. Feasibility Test of the RMDF Specification of Ingenuity

We have applied the feasibility test presented in [3] to the Ingenuity specification. This feasibility test asserts that it is necessary that the WCET of all actors are less than or equal to the minimum execution windows length presented in column *Exec. Windows* of Table II for Ingenuity to be feasible. Those maximum WCETs of Ingenuity are presented in Table III.

IV. A TIMING ANOMALY ON THE SIXTH FLIGHT OF INGENUITY

During the sixth flight of Ingenuity, an anomaly occurred resulting in a camera frame to be lost⁴ and Ingenuity tilting back and forth in an oscillating pattern. To the best of our knowledge, this “lost of camera frame” anomaly is the only public information available about this incident. With the help of the RMDF formalism, this section presents a plausible scenario and the causes that could have led to this incident.

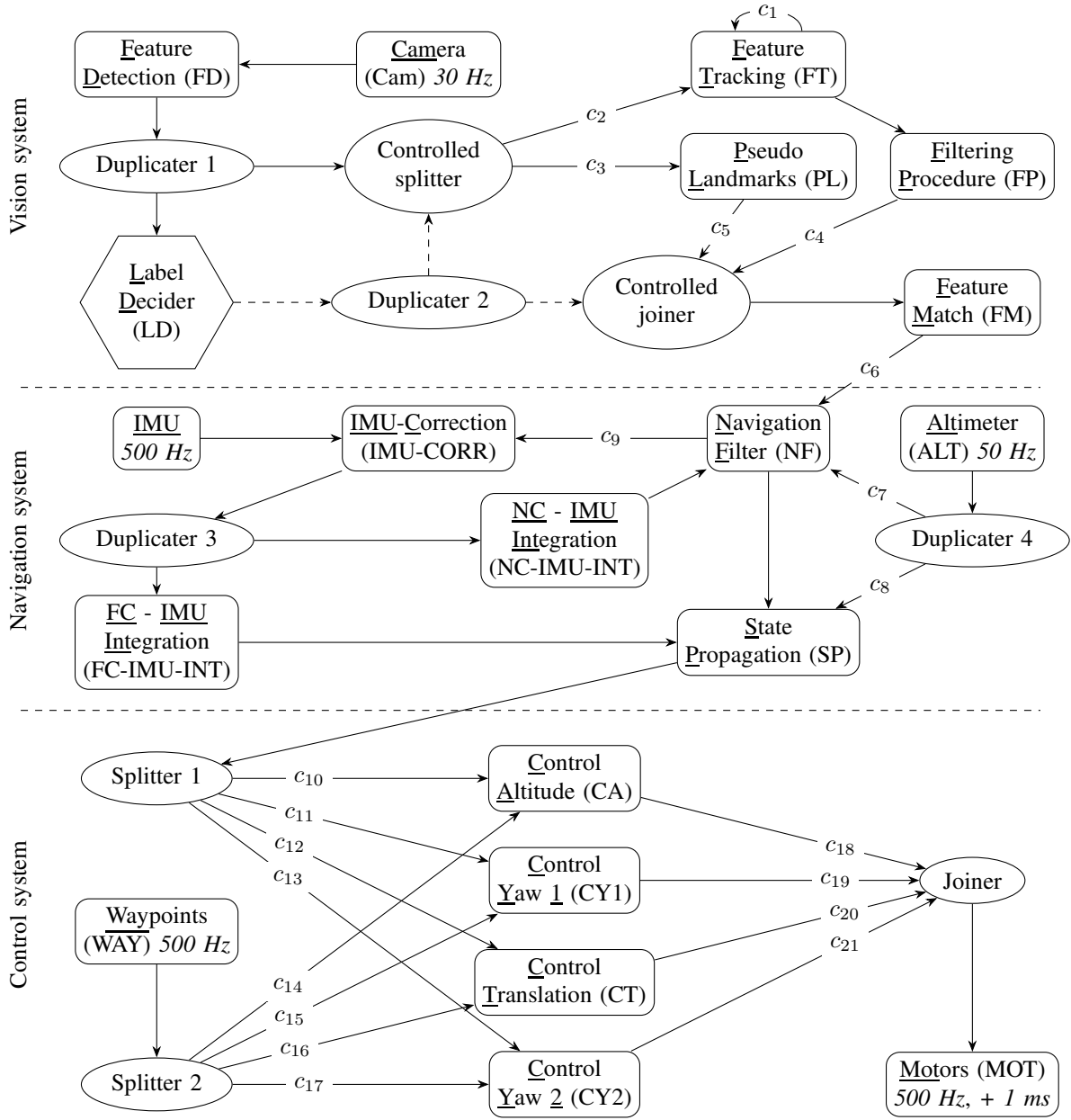
A. Additional Assumptions of Ingenuity’s Inner Functioning

We need to make additional assumptions to specify this anomaly in Fig. 3a:

- 1) We know from the textual specification [9] that the vision system executes on a single core, so we may assume that the vision systems run on a preemptive kernel.
- 2) We may assume that the *Controlled Joiner* of Fig. 3a was, in fact, a *Joiner* with a naive behavior; that is,

⁴<https://science.nasa.gov/blogs/surviving-an-in-flight-anomaly-what-happened-on-ingenuitys-sixth-flight/>

(a) An RMDF specification of Ingenuity deployed after its sixth flight. Rates of 1 and initial tokens of 0 are omitted for clarity.



(b) Channels' specifications of the above RMDF specification.

chan.	cons.	prod.	init.	chan.	cons.	prod.	init.	chan.	cons.	prod.	init.
c_1	1	1	1	c_8	1	1/10	0	c_{15}	1/4	1	0
c_2	m_1	1	0	c_9	1	1	1	c_{16}	1/4	1	0
c_3	m_2	1	0	c_{10}	1/4	1	0	c_{17}	1/4	1	0
c_4	1	m_1	0	c_{11}	1/4	1	0	c_{18}	1	1/4	0
c_5	1	m_2	0	c_{12}	1/4	1	0	c_{19}	1	1/4	0
c_6	3/50	1	1/50	c_{13}	1/4	1	0	c_{20}	1	1/4	0
c_7	1	1/10	0	c_{14}	1/4	1	0	c_{21}	1	1/4	0

Fig. 3: An RMDF specification of Ingenuity deployed after its sixth flight and the specification of its channel characteristics.

TABLE II: Releases, deadlines and execution windows length of the n -th job for each actor of Ingenuity, assuming BCET and WCET of all actors are 0.12 and 0.20 ms, respectively, except for *Controlled Splitter* and *Controlled Joiner* which have a constant execution time of 0 ms.

Actor	Timing constraints of the n -th job		
	Release	Deadline	Exec. Windows
CAM	$(100 \cdot (n-1))/3$	$\begin{cases} 6/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 176/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 336/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 6/5 = 1.20 \text{ ms} \\ 25/15 \approx 1.87 \text{ ms} \\ 8/15 \approx 0.53 \text{ ms} \end{cases}$
FD	$(9 + 2500 \cdot (n-1))/75$	$\begin{cases} 7/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 177/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 337/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 32/25 = 1.96 \text{ ms} \\ 146/75 \approx 1.94 \text{ ms} \\ 46/75 \approx 0.61 \text{ ms} \end{cases}$
LD	$(18 + 2500 \cdot (n-1))/75$	$\begin{cases} 8/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 178/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 338/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 34/25 = 1.36 \text{ ms} \\ 152/75 \approx 2.03 \text{ ms} \\ 52/75 \approx 0.70 \text{ ms} \end{cases}$
CS	$(27 + 2500 \cdot (n-1))/75$	$\begin{cases} 8/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 178/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 338/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 31/25 = 1.24 \text{ ms} \\ 143/75 \approx 1.91 \text{ ms} \\ 43/75 \approx 0.57 \text{ ms} \end{cases}$
FT	$(27 + 2500 \cdot (n-1))/75$	$\begin{cases} 9/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 179/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 339/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 36/25 = 1.44 \text{ ms} \\ 158/75 \approx 2.11 \text{ ms} \\ 58/75 \approx 0.78 \text{ ms} \end{cases}$
FP	$(36 + 2500 \cdot (n-1))/75$	$\begin{cases} 2 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 36 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 68 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 38/25 = 1.52 \text{ ms} \\ 164/75 \approx 2.19 \text{ ms} \\ 64/75 \approx 0.85 \text{ ms} \end{cases}$
PL	$(27 + 2500 \cdot (n-1))/75$	$\begin{cases} 2 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 36 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 68 + \lfloor (100 \cdot (n-1))/3 \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 41/25 = 1.64 \text{ ms} \\ 173/75 \approx 2.31 \text{ ms} \\ 73/75 \approx 0.97 \text{ ms} \end{cases}$
CJ	$(9 + 500 \cdot (n-1))/15$	$\begin{cases} 2 + \lfloor (100 \cdot (n-1)) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 36 + \lfloor (100 \cdot (n-1)) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 68 + \lfloor (100 \cdot (n-1)) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 7/5 = 1.40 \text{ ms} \\ 31/15 \approx 2.07 \text{ ms} \\ 11/15 \approx 0.73 \text{ ms} \end{cases}$
FM	$(9 + 500 \cdot (n-1))/15$	$\begin{cases} 11/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 0 \\ 181/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 1 \\ 341/5 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 3 = 2 \end{cases}$	$\begin{cases} 8/5 = 1.60 \text{ ms} \\ 34/15 \approx 2.27 \text{ ms} \\ 14/15 \approx 0.93 \text{ ms} \end{cases}$
NF	$\begin{cases} 18/25 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 0 \\ 5054/75 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 33 \\ 9/25 + 2 \cdot (n-1) & \text{otherwise} \end{cases}$	$12/5 + 2 \cdot (n-1)$	$\begin{cases} 42/25 = 1.68 \text{ ms} \\ 76/75 \approx 1.01 \text{ ms} \\ 51/25 \approx 2.04 \text{ ms} \end{cases}$
IMU-CORR	$3/25 + 2 \cdot (n-1)$	$2 + 2 \cdot (n-1)$	$47/25 = 1.88 \text{ ms}$
NC-IMU-INT	$6/25 + 2 \cdot (n-1)$	$11/5 + 2 \cdot (n-1)$	$49/25 = 1.96 \text{ ms}$
FC-IMU-INT	$6/25 + 2 \cdot (n-1)$	$12/5 + 2 \cdot (n-1)$	$54/25 = 2.16 \text{ ms}$
IMU	$2 \cdot (n-1)$	$9/5 + 2 \cdot (n-1)$	$9/5 = 1.8 \text{ ms}$
ALT	$20 \cdot (n-1)$	$11/5 + 20 \cdot (n-1)$	$11/5 = 2.2 \text{ ms}$
SP	$\begin{cases} 21/25 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 0 \\ 5063/75 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 33 \\ 62/25 + 2 \cdot (n-1) & \text{otherwise} \end{cases}$	$13/5 + 2 \cdot (n-1)$	$\begin{cases} 44/25 = 1.76 \text{ ms} \\ 82/75 \approx 1.09 \text{ ms} \\ 53/25 = 2.12 \text{ ms} \end{cases}$
CA	$\begin{cases} 24/25 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 25 = 0 \\ 3/5 + 8 \cdot (n-1) & \text{otherwise} \end{cases}$	$14/5 + 8 \cdot (n-1)$	$\begin{cases} 46/25 = 1.84 \text{ ms} \\ 11/5 = 1.2 \text{ ms} \end{cases}$
CY1	$\begin{cases} 5072/75 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 25 = 8 \\ 13/5 + 8 \cdot (n-1) & \text{otherwise} \end{cases}$	$24/5 + 8 \cdot (n-1)$	$\begin{cases} 88/75 = 1.17 \text{ ms} \\ 11/5 = 1.2 \text{ ms} \end{cases}$
CT	$\begin{cases} 2524/25 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 25 = 12 \\ 23/5 + 8 \cdot (n-1) & \text{otherwise} \end{cases}$	$34/5 + 8 \cdot (n-1)$	$\begin{cases} 46/25 = 1.84 \text{ ms} \\ 11/5 = 1.2 \text{ ms} \end{cases}$
CY2	$\begin{cases} 12572/75 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 25 = 20 \\ 33/5 + 8 \cdot (n-1) & \text{otherwise} \end{cases}$	$44/5 + 8 \cdot (n-1)$	$\begin{cases} 88/75 \approx 1.17 \text{ ms} \\ 11/5 = 1.2 \text{ ms} \end{cases}$
WAY	$2 \cdot (n-1)$	$2 + 2 \cdot (n-1)$	2 ms
MOT	$\begin{cases} 27/25 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 0 \\ 5081/75 + \lfloor 100 \cdot (n-1) \rfloor & \text{if } (n-1) \bmod 50 = 33 \\ 1 + 2 \cdot (n-1) & \text{otherwise} \end{cases}$	$1 + 2 \cdot (n-1)$	$\begin{cases} 48/25 = 1.92 \text{ ms} \\ 94/75 \approx 1.25 \text{ ms} \\ 2 \text{ ms} \end{cases}$

it waits on all input channels and sends on its output channel the first token receive.

- 3) We may assume that *Feature Match* expects the frame to arrive in the correct order, i.e., the n -th frame produced by the camera is the n -th frame received by the *Feature Match*. Otherwise, it is considered as an anomaly, and the frame is discarded.

TABLE III: Maximum WCETs of the actors of the Ingenuity system that constitute a necessary condition for its feasibility.

Actors	WCET	Actors	WCET
Camera	0.53 ms	NC - IMU - Integration	1.96 ms
Feature Detection	0.61 ms	FC - IMU - Integration	2.16 ms
Label Decider	0.70 ms	IMU	1.8 ms
Controlled Splitter	0.57 ms	Altimeter	2.2 ms
Feature Tracking	0.78 ms	State Propagation	1.09 ms
Filtering Procedure	0.85 ms	Control Altitude	1.2 ms
Pseudo Landmarks	0.97 ms	Control Yaw 1	1.17 ms
Controlled Joiner	0.73 ms	Control Translation	1.2 ms
Feature Match	0.93 ms	Control Yaw 2	1.17 ms
Navigation Filter	1.01 ms	Waypoints	2 ms
IMU-Correction	1.88 ms	Motors	1.25 ms

B. A Plausible Explicability

The anomaly of the sixth flight is caused by a frame loss. This frame lost could result from a frame inversion followed by a frame discard. Let us assume that during the sixth flight, a job of the actor *Filtering Procedure* takes a longer execution time than its static WCET. The schedule of Fig. 4 presents this situation. A longer execution time of the second job of *Filtering Procedure* implies its preemption (assumption from Item 1), and it implies a frame inversion between frames number 2 and 3, leading to the discard of frame 2. This inversion is because the Joiner waits on all its input channels and sends the first token received (assumption of Item 2). According to Fig. 4, *Pseudo Landmarks* produces two frames before *Filtering Procedure* produces one. As frame number 2 arrives after frame number 3, frame number 2 is discarded (assumption from Item 3). According to our assumptions, this frame inversion followed by a frame discard may be the anomaly that occurred during the sixth flight of Ingenuity.

V. CONCLUSION AND FUTURE WORKS

We have presented the translation from the textual specification of the Ingenuity Mars helicopter to an RMDF specification. RMDF is a formalism presented in [3] which is well-suited to specify and analyze mode-dependent cyber-physical systems under relaxed real-time constraints such as Ingenuity. A discussion on the fidelity of the Ingenuity specification was presented. We have also presented a plausible scenario and the causes that could have led to the anomaly of the sixth flight of Ingenuity. As future work, we want to focus on the runtime verification of Ingenuity. We will use the results Table II to monitor the timing behavior of Ingenuity during its execution.

REFERENCES

- [1] NASA, “Mars 2020 Perseverance Launch Press Kit,” 2020.
- [2] G. Roumage, S. Azaiez, and S. Louise, “A Survey of Main Dataflow MoCCs for CPS Design and Verification,” in *2022 IEEE 15th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*, 2022, pp. 1–9.
- [3] G. Roumage, S. Azaiez, C. Faure, and S. Louise, “Real-time Mode-Aware Dataflow: A Dataflow Model to Specify and Analyze Mode-dependent CPSs under Relaxed Timing Constraints,” 2025. <https://hal.science/hal-04879612>.
- [4] P. Dubrulle, N. Kosmatov, C. Gaston, and A. Lapitre, “PolyGraph: A Data Flow Model with Frequency Arithmetic,” *International Journal on Software Tools for Technology Transfer*, vol. 23, no. 3, pp. 489–517, 2021.
- [5] E. A. Lee and D. Messerschmitt, “Synchronous Data Flow,” *Proceedings of the IEEE*, vol. 75, no. 9, pp. 1235–1245, 1987.
- [6] G. Bilsen, M. Engels, R. Lauwereins, and J. A. Peperstraete, “Cyclo-Static Dataflow,” *IEEE Transactions on Signal Processing*, vol. 44, no. 2, pp. 397–408, 1996.
- [7] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Transactions of the ASME - Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [8] B. Balaram, T. Canham, C. Duncan, H. F. Grip, W. Johnson, J. Maki, A. Quon, R. Stern, and D. Zhu, “Mars Helicopter Technology Demonstrator,” in *2018 AIAA Atmospheric Flight Mechanics Conference*. American Institute of Aeronautics and Astronautics, 2018.
- [9] D. S. Bayard, D. T. Conway, R. Brockers, J. H. Delaune, L. H. Matthies, H. F. Grip, G. B. Merewether, T. L. Brown, and A. M. San Martin, “Vision-Based Navigation for the NASA Mars Helicopter,” in *AIAA Scitech 2019 Forum*, 2019.
- [10] H. Grip, J. N. Lam, D. Bayard, D. T. Conway, G. Singh, R. Brockers, J. Delaune, L. Matthies, C. Malpica, T. Brown, A. Jain, M. San Martin, and G. Merewether, “Flight Control System for Nasa’s Mars Helicopter,” in *AIAA Scitech*, 2019.
- [11] M. A. San Martin, D. S. Bayard, D. T. Conway, M. Mandic, and E. S. Bailey, “A Minimal State Augmentation Algorithm for Vision-Based Navigation Without Using Mapped Landmarks,” 2017.
- [12] H. Grip, D. Scharf, C. Malpica, W. Johnson, M. Mandic, G. Singh, and L. Young, “Guidance and Control for a Mars Helicopter,” in *2018 AIAA Guidance, Navigation, and Control Conference*, 2018.
- [13] H. F. Grip, D. Conway, J. Lam, N. Williams, M. Golombek, R. Brockers, M. Mischna, and M. R. Cacan, “Flying a Helicopter on Mars: How Ingenuity’s Flights Were Planned, Executed, and Analyzed,” in *2022 IEEE Aerospace Conference (AERO)*, 2022.
- [14] G. Roumage, S. Azaiez, C. Faure, and S. Louise, “Static Timing Analysis of Cyber-Physical Systems with Relaxed Real-Time Constraints,” in *SEAA 2024: 50th Euromicro Conference Series on Software Engineering and Advanced Applications*, 2024.

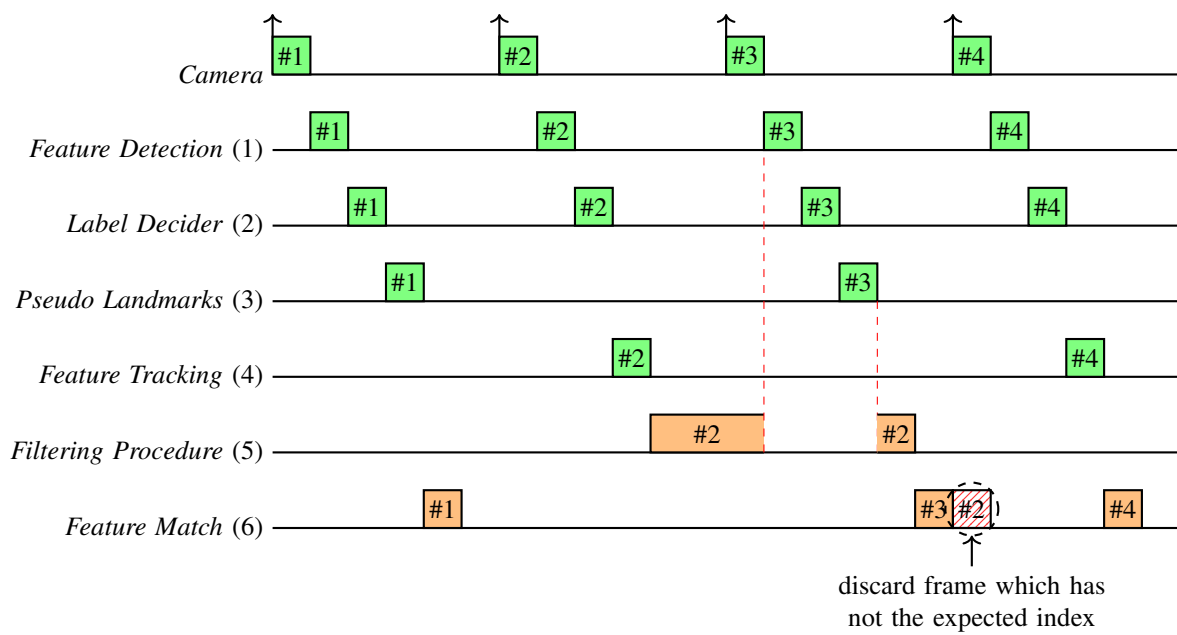


Fig. 4: Schedule of the actors of the vision processing leading to the anomaly of its sixth flight.