Formal Methods for Timing Verification
The 2015 FMTV Challenge

waters2015.inria.fr/challenge

proposed by Rafik HENIA and Laurent RIOUX
Thales Research & Technology France

Hosted by **WATERS 2015**

In conjunction with ECRTS 7th-10th July 2015 Lund, Sweden

- Motivation
- Industrial use-case
- Challenge 1
- Challenge 2

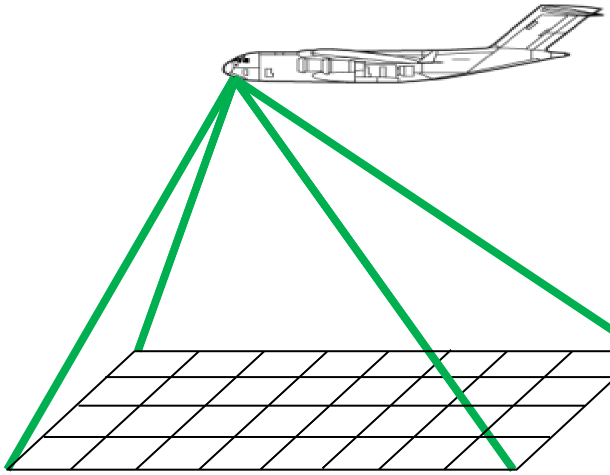
- Motivation
- Industrial use-case
- Challenge 1
- Challenge 2

- Evaluate the applicability of the different formal timing verification methods to a concrete industrial application and identify their strengths and weaknesses
- Challenge the various formal timing verification methods with scientific stakes issued from a real industrial use case
- Promote discussion , closer interactions, cross fertilization of ideas and synergies across the breadth of the real-time research community and the industry

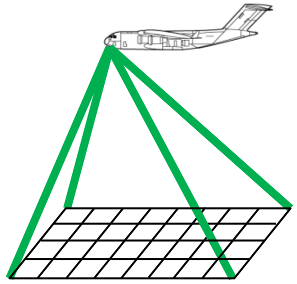
- Motivation
- Industrial use-case
- Challenge 1
- Challenge 2

Aerial video system to detect and track a moving object, e.g. a vehicle on a roadway

- Mission critical system
- Used in intelligence, surveillance, reconnaissance, tactical and security applications
- Characterized by strict and less strict constraints on timing

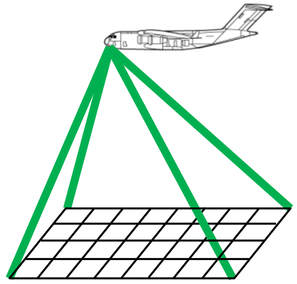


Aerial video tracking system – main tasks

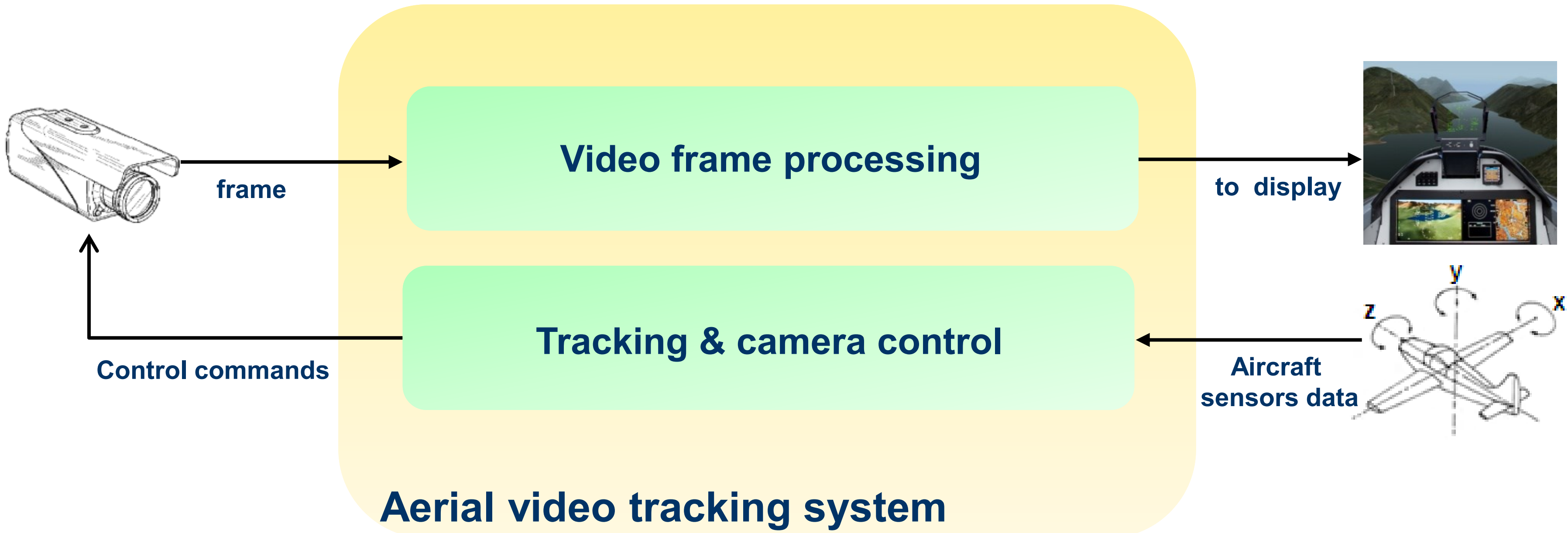


- Display a high quality video imagery to the user
- Detect patches of the image that may be moving differently from the background by combining image registration and motion estimation
- Track the corresponding object over longer time periods when such a patch persists for several frames
- Follow the tracked object even when it is temporarily hidden from view (e.g. the vehicle proceeds in and out of several tree obstructed areas) through motion prediction

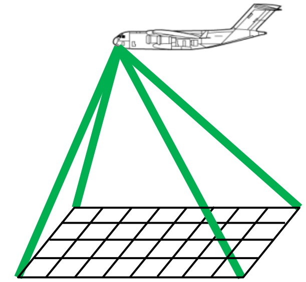
Consists of two subsystems:



- Video frame processing
- Tracking and camera control

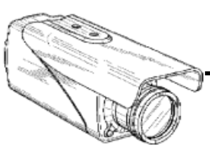


- Motivation
- Industrial use-case
- Challenge 1
- Challenge 2



Video frame processing – main tasks

- Process the video frames sent by the camera
- Embed tracking data into the video
- Convert the frames to the required format
- Run the video at 25 frames per second
- Display a high quality video imagery on the monitor

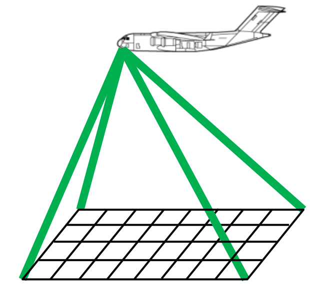


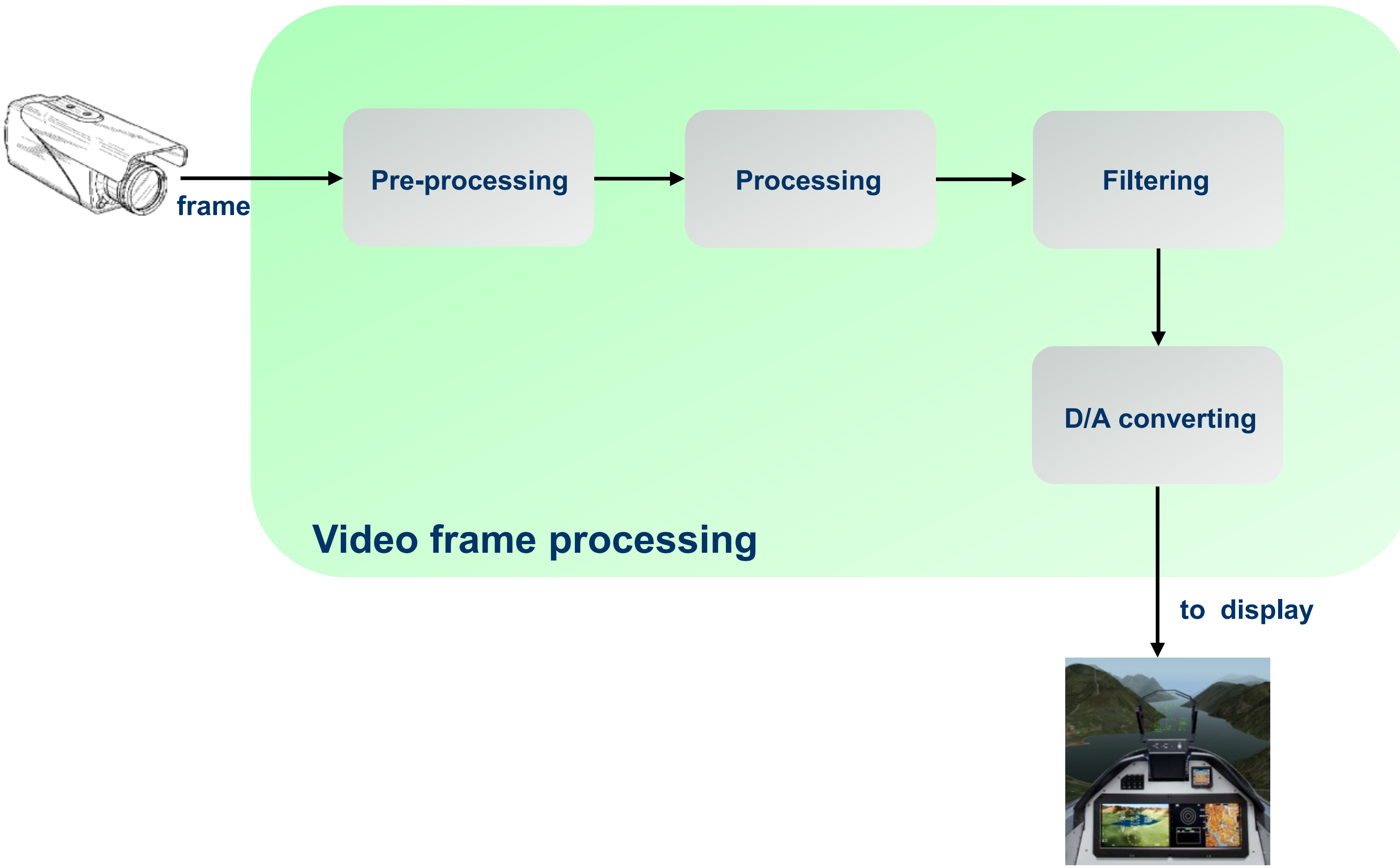
frame

Video frame processing

to display

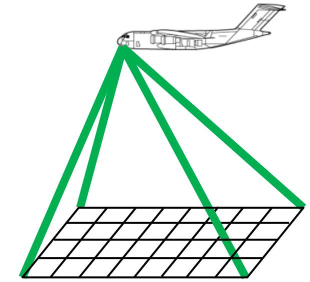
Video frame processing – functional view





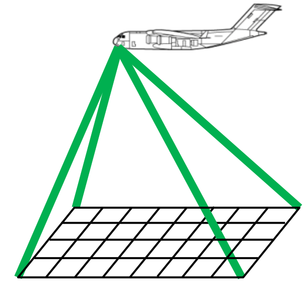
CHALLENGE 1 – VIDEO FRAME PROCESSING

Video frame processing – functional view



frame → Pre-processing → Processing → Filtering → D/A converting → to display

- Receives frames from the camera
- Removes reflections
- Normalizes the intensity of the individual particles images
- Sends the pre-processed frames to the Processing function



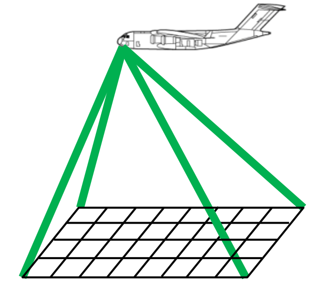
Video frame processing – functional view

frame → Pre-processing → Processing → Filtering → D/A converting → to display

- Receives frames from the Pre-processing function
- Embeds tracking information into the video frames
- Executes zoom in & zoom out instructions
- Sends the processed frames to the Filtering function

CHALLENGE 1 – VIDEO FRAME PROCESSING

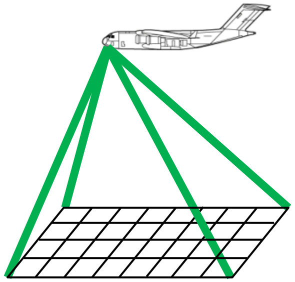
Video frame processing – functional view

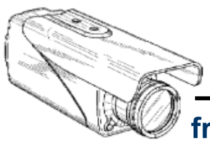


frame → Pre-processing → Processing → Filtering → D/A converting → to display

- Receives images from the Processing function
- Resizes the frames
- Removes the noise
- Sends the frames to the A/D converting function

Video frame processing – functional view

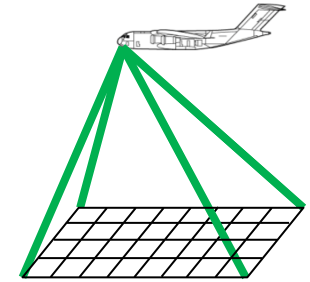


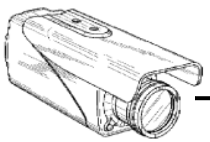


frame → Pre-processing → Processing → Filtering → D/A converting → to display

- Receives frames from the Filtering function
- Converts frames from digital to analog
- Sends the processed frames to the display

Video frame processing – functional deployment

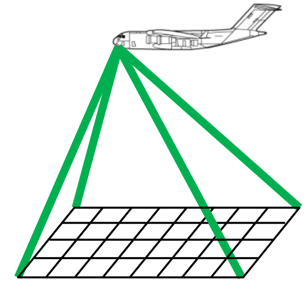


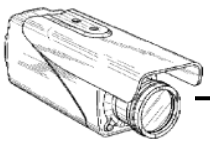


frame → **FPGA**: Pre-processing → **GPP1**: Processing → **GPU**: Filtering → **GPP2**: D/A converting → to display

- Each function is deployed on a different physical device (processor)

Video frame processing – architectural view

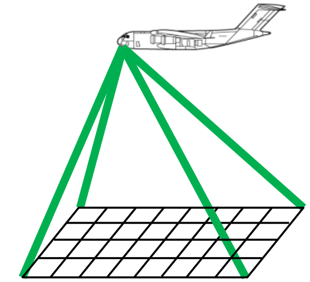




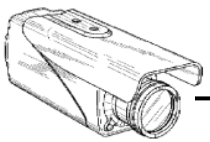
frame

FPGA: T1 (Pre-processing) → GPP1: T2 (Processing) → GPU: T3 (Filtering) → GPP2: T4 (D/A converting) → to display

- Each function is mapped to one single task



Video frame processing – timing behavior and characteristics



frame

FPGA

T1
(Pre-processing)

GPP1

T2
(Processing)

GPU

T3
(Filtering)

GPP2

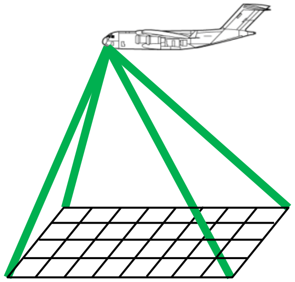
T4
(D/A converting)

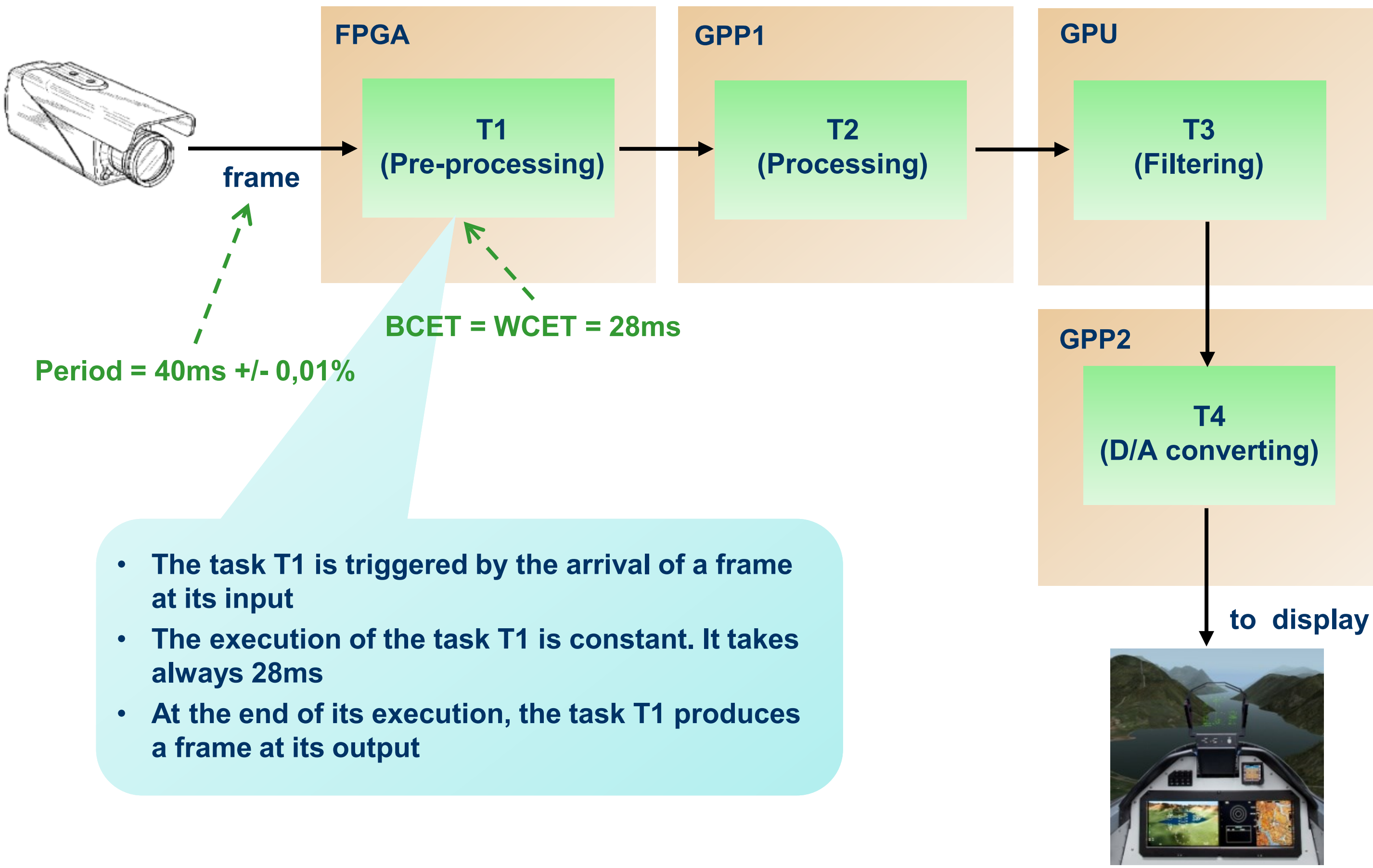
to display

Period = 40ms +/- 0,01%

- The frames are sent from the camera strictly periodically, i.e. the period value is constant
- Due to the hardware clock drift, the exact period value is unknown, however it ranges between 40ms + 0,01% and 40ms - 0,01%
- Each frame is assigned a unique index

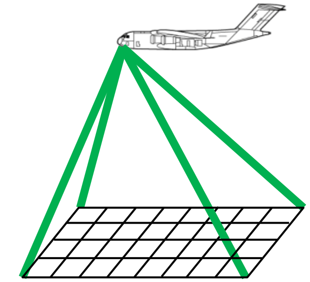
Video frame processing – timing behavior and characteristics

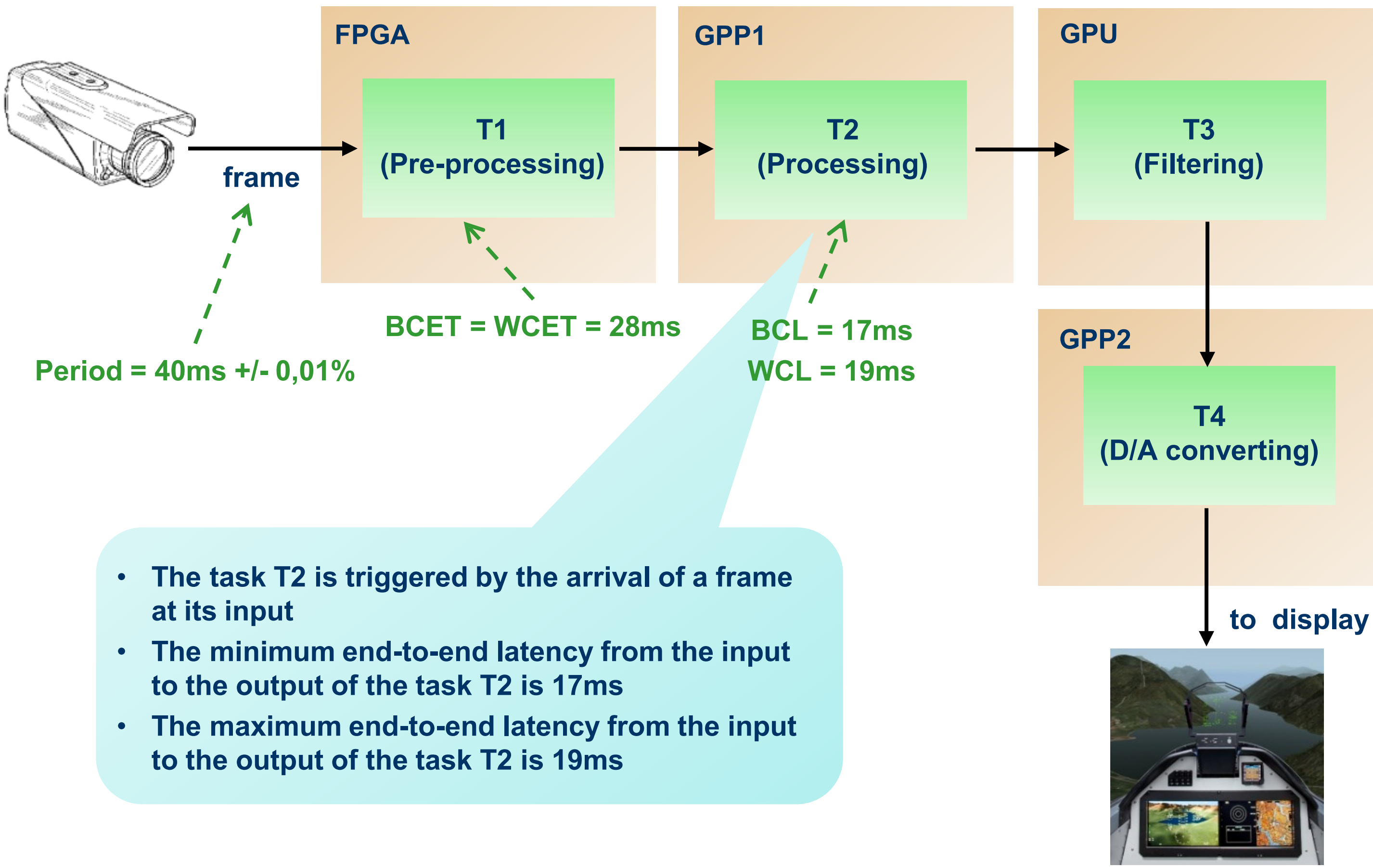




- The task T1 is triggered by the arrival of a frame at its input
- The execution of the task T1 is constant. It takes always 28ms
- At the end of its execution, the task T1 produces a frame at its output

Video frame processing – timing behavior and characteristics

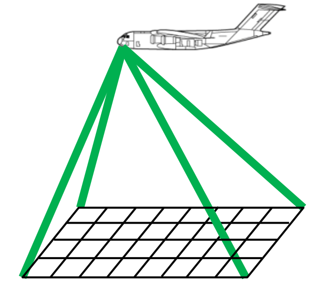


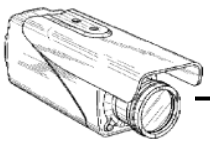


- The task T2 is triggered by the arrival of a frame at its input
- The minimum end-to-end latency from the input to the output of the task T2 is 17ms
- The maximum end-to-end latency from the input to the output of the task T2 is 19ms

CHALLENGE 1 – VIDEO FRAME PROCESSING

Video frame processing – timing behavior and characteristics





frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

GPP2

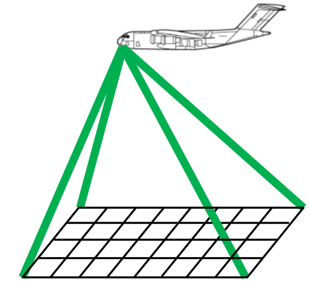
T4
(D/A converting)

to display

- A register is used for the communication between the tasks T2 and T3
- At the end of each execution, the task T2 overwrites the register content by writing a frame
- At each activation, the task T3 reads the current frame in the register
- For simplicity, we assume that there is no conflicts between the read and write accesses to the register

CHALLENGE 1 – VIDEO FRAME PROCESSING

Video frame processing – timing behavior and characteristics



frame

Period = 40ms +/- 0,01%

FPGA

T1 (Pre-processing)

BCET = WCET = 28ms

GPP1

T2 (Processing)

BCL = 17ms
WCL = 19ms

GPU

T3 (Filtering)

Period = 40/3ms +/- 0,05%

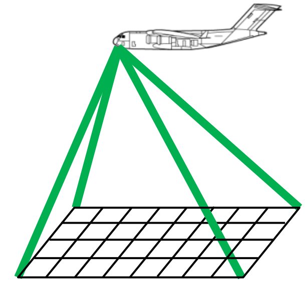
GPP2

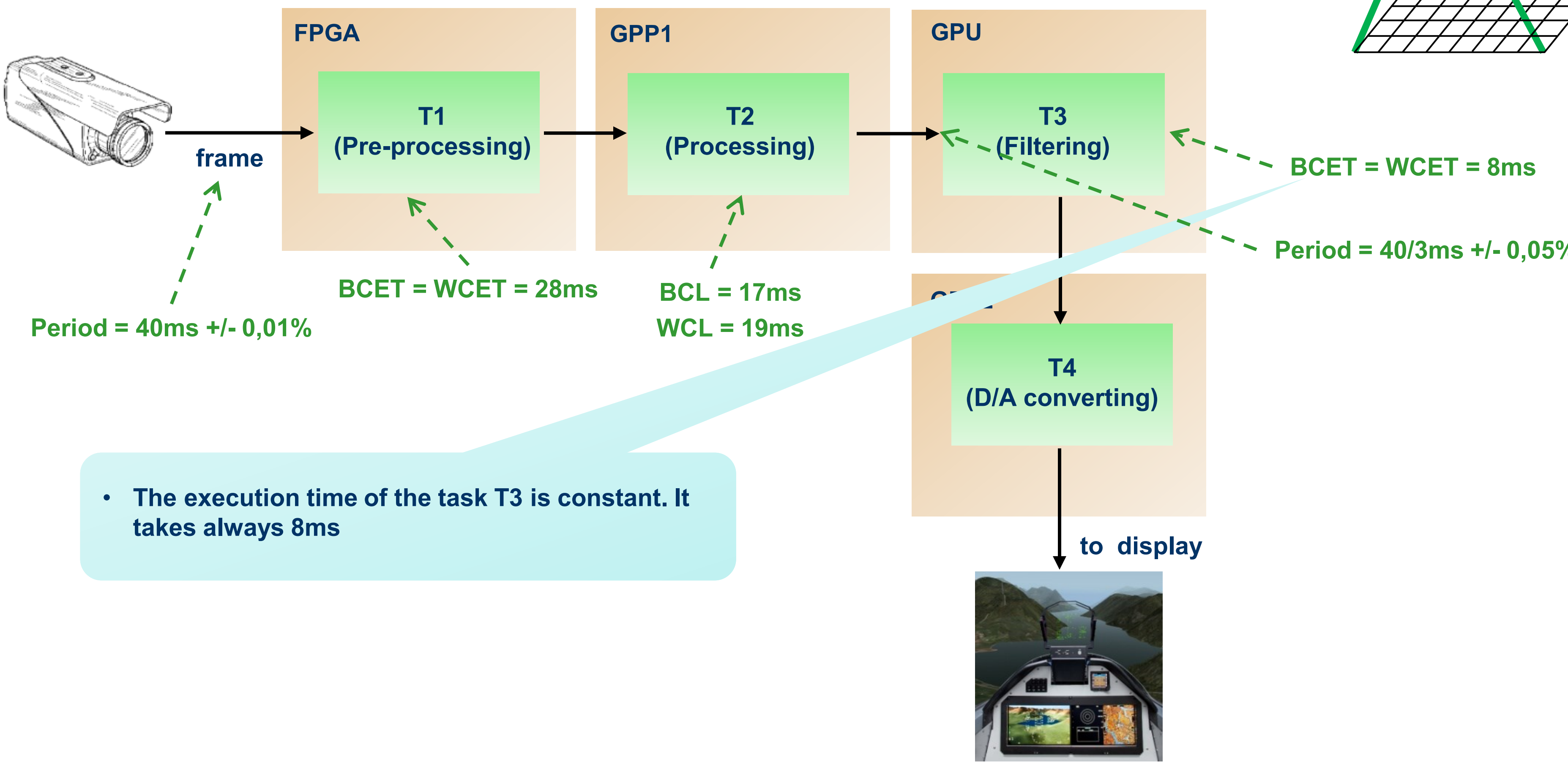
converting)

to display

- The task T3 is activated strictly periodically, i.e. the period value is constant
- Due to the clock drift, the exact period value is unknown, however it ranges between 40/3ms + 0,05% and 40/3ms - 0,05%
- Note that since the task T3 is activated more frequently than the task T2, it will process the same register content more than once

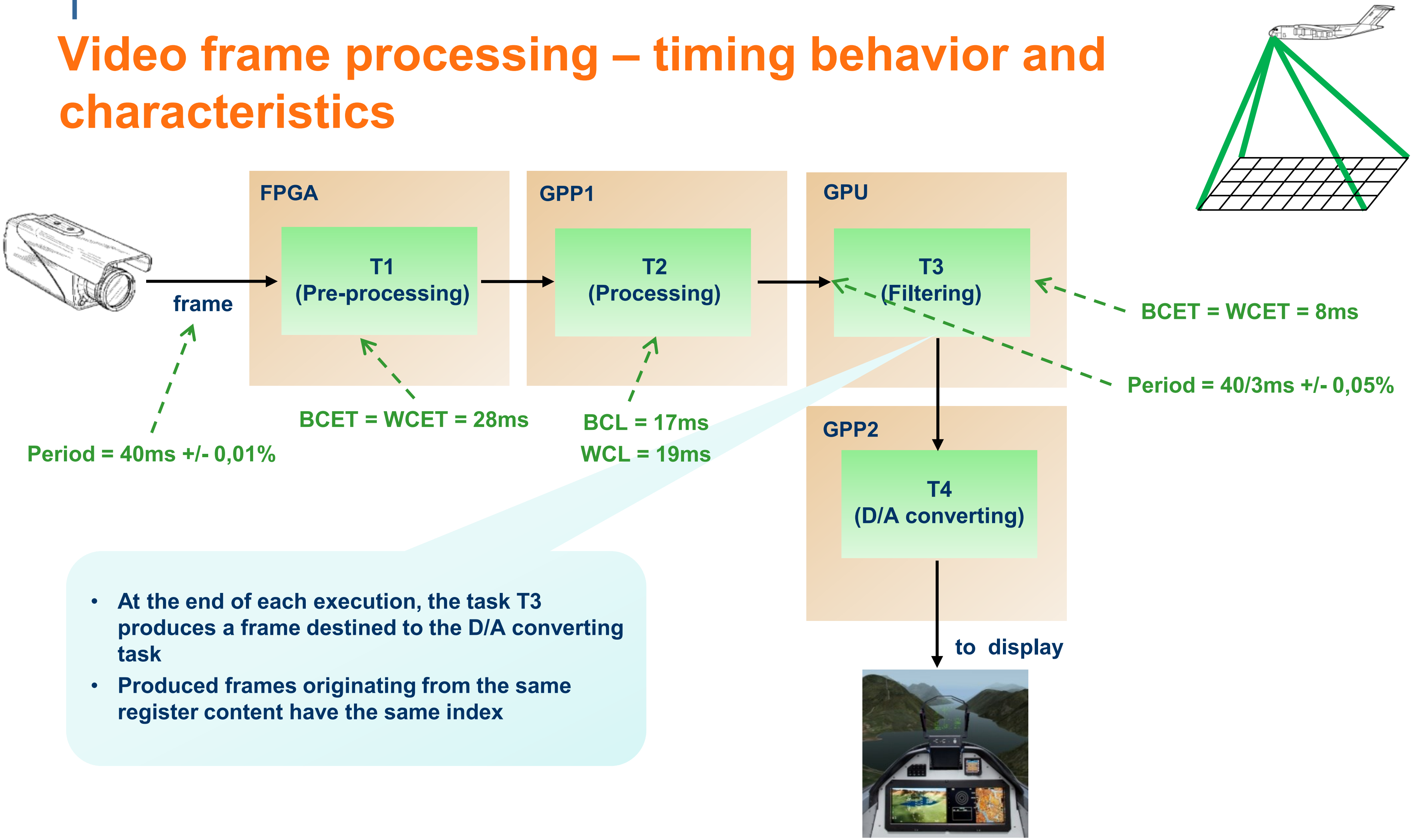
Video frame processing – timing behavior and characteristics





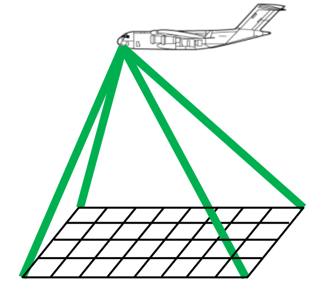
- The execution time of the task T3 is constant. It takes always 8ms

Video frame processing – timing behavior and characteristics



- At the end of each execution, the task T3 produces a frame destined to the D/A converting task
- Produced frames originating from the same register content have the same index

Video frame processing – timing behavior and characteristics



frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

BCET = WCET = 8ms

Period = 40/3ms +/- 0,05%

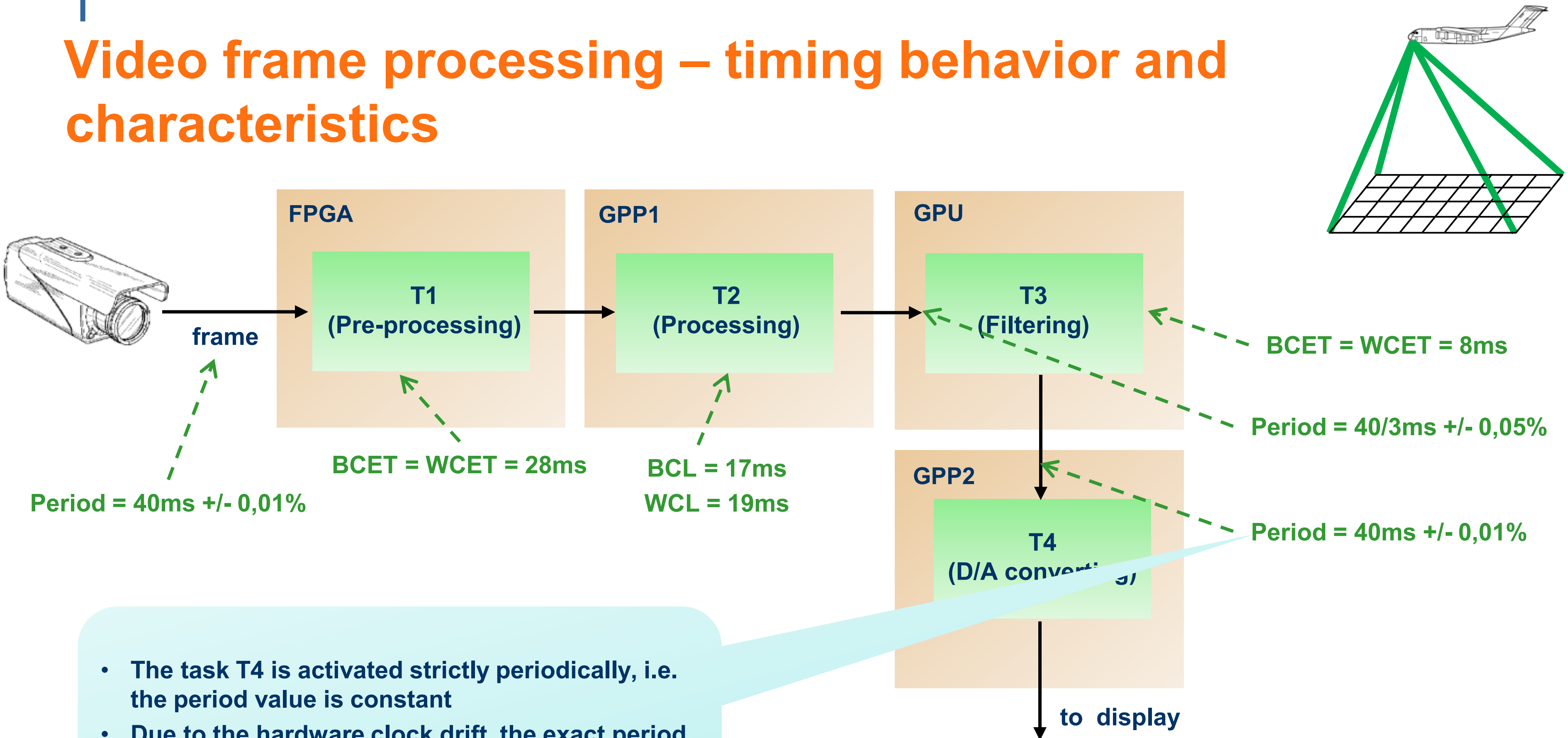
GPP2

T4
(D/A converting)

to display

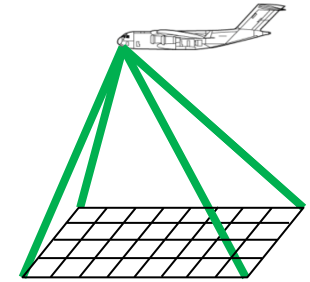
- There is a buffer at the input of task T4 to store selected frames sent by task T3
- The buffer size is n (n is an integer), i.e. the buffer can contain at most n frames at the same time
- If the buffer is full, the frame sent by the task T3 is discarded
- For each frame index value, only one single frame can be stored in the buffer. If the buffer has already stored a frame with a given index, any additional received frame with the same index is discarded
- The time required to discard a frame or to store it in the buffer can be ignored

Video frame processing – timing behavior and characteristics



- The task T4 is activated strictly periodically, i.e. the period value is constant
- Due to the hardware clock drift, the exact period value is unknown, however it ranges between 40ms + 0,01% and 40ms - 0,01%

Video frame processing – timing behavior and characteristics



frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

BCET = WCET = 8ms

Period = 40/3ms +/- 0,05%

GPP2

T4
(D/A converting)

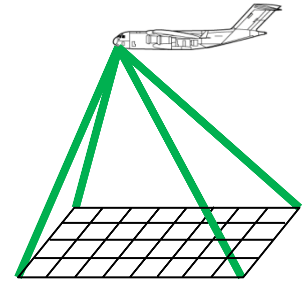
Period = 40ms +/- 0,01%

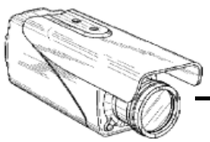
CET = 1ms or 10ms

- When activated, the execution of the task T4 takes 1ms if the buffer is empty
- If the buffer is not empty, the task T4 consumes a single frame from the buffer. In this case, the task execution takes 10ms
- At the end of its execution, if a frame has been processed, the task T4 produces a frame to be displayed on the monitor

to display

Video frame processing – timing behavior and characteristics





frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

BCET = WCET = 8ms

Period = 40/3ms +/- 0,05%

GPP2

T4
(D/A converting)

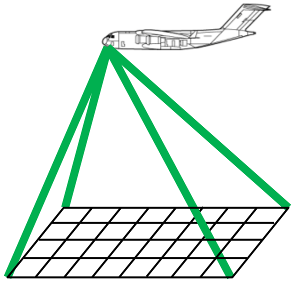
Period = 40ms +/- 0,01%

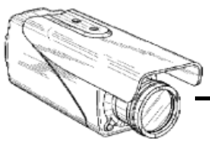
CET = 1ms or 10ms

to display

- The communication between all processors, the access to the register between GPP1 and GPU and the access to the buffer between GPU and GPP2 are considered to be timeless

Video frame processing – challenge 1A





frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

BCET = WCET = 8ms

Period = 40/3ms +/- 0,05%

GPP2

T4
(D/A converting)

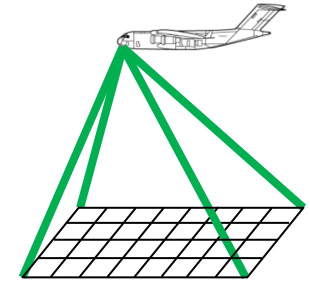
Period = 40ms +/- 0,01%

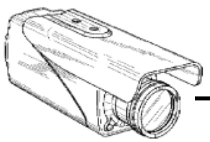
CET = 1ms or 10ms

to display

- **Compute:**
  1. the minimum and maximum latencies for a given frame from the camera output to the display input , for a buffer size n = 1
  2. Compute the minimum and maximum latencies for a given frame from the camera output to the display input , for a buffer size n = 3

Video frame processing – challenge 1B





frame

Period = 40ms +/- 0,01%

FPGA

T1
(Pre-processing)

BCET = WCET = 28ms

GPP1

T2
(Processing)

BCL = 17ms
WCL = 19ms

GPU

T3
(Filtering)

BCET = WCET = 8ms

Period = 40/3ms +/- 0,05%

GPP2

T4
(D/A converting)

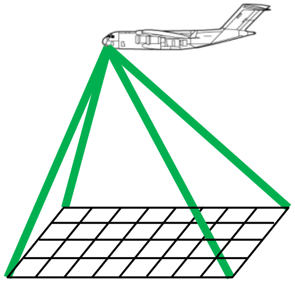
Period = 40ms +/- 0,01%

CET = 1ms or 10ms

to display

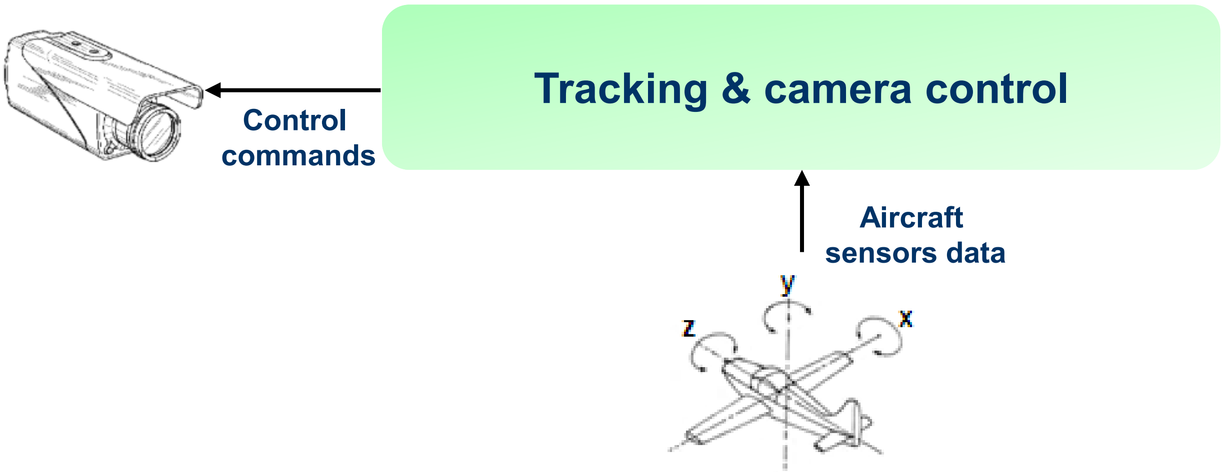
- Due to the different clock drifts, all frame with a same index may be discarded at the entrance of the buffer at the input of the task T4.
- Compute:
    1. the minimum time distance between two frames produced by the camera that will not reach the display, for a buffer size n = 1
    2. the minimum time distance between two frames produced by the camera that will not reach the display, for a buffer size n = 3

- Motivation
- Industrial use-case
- Challenge 1
- Challenge 2

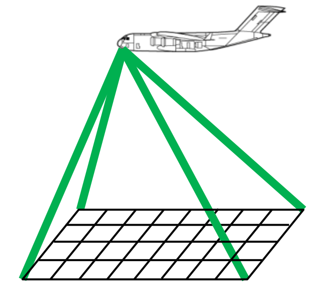


Tracking & camera control – main tasks

- Perform motion prediction for the tracked object
- Calculate new camera angle based on the aircraft sensors data (position, direction and speed, etc..) and the tracked object motion prediction
- Execute zoom-in and zoom-out instructions



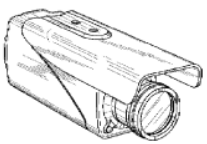
Tracking & camera control – functional view



Target position prediction

Tracking control

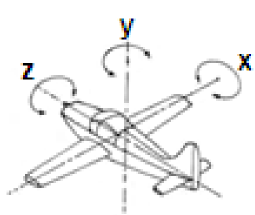
Camera control

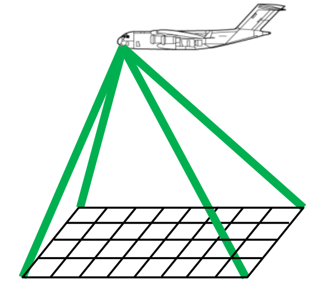


Control commands

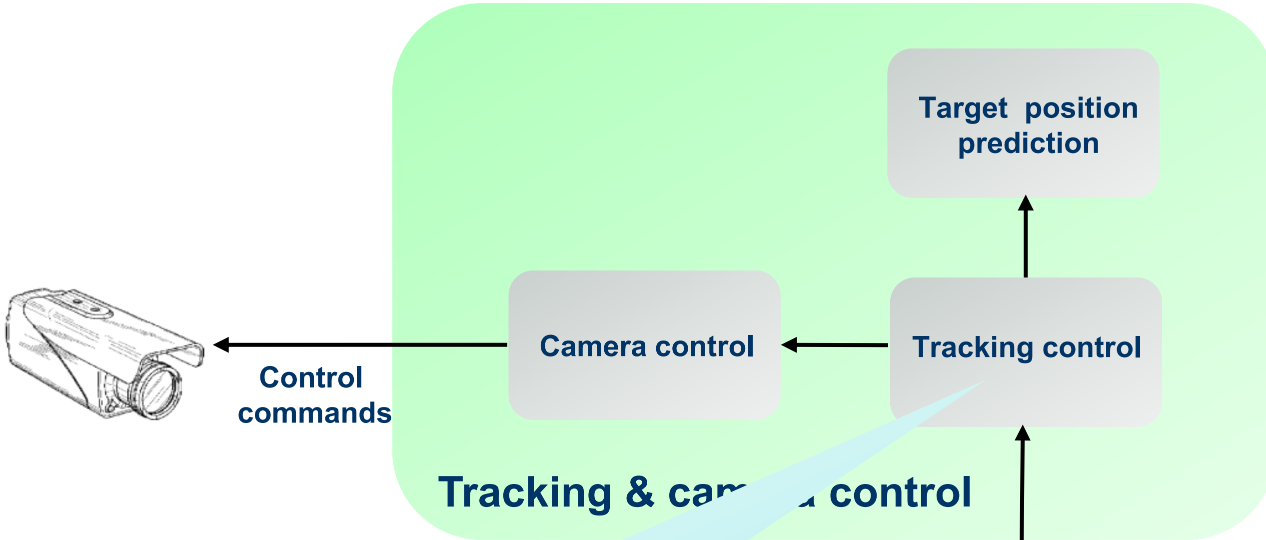
Tracking & camera control

Aircraft sensors data

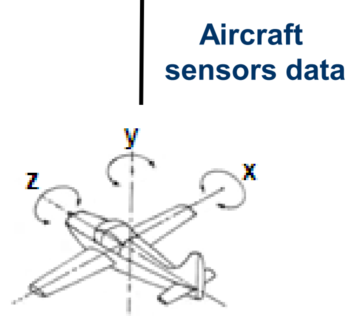




Tracking & camera control – functional view

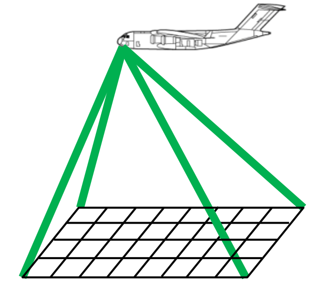


- Processes the aircraft sensors data (position, direction, speed, etc..)
- Controls the whole tracking process
- Generates alerts and various tracking data



CHALLENGE 2 – TRACKING & CAMERA CONTROL

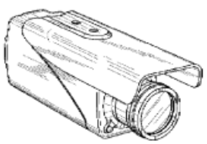
Tracking & camera control – functional view



Target position prediction

Camera control

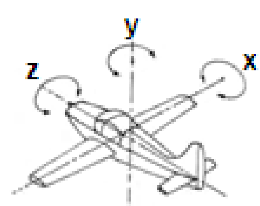
Tracking control



Control commands

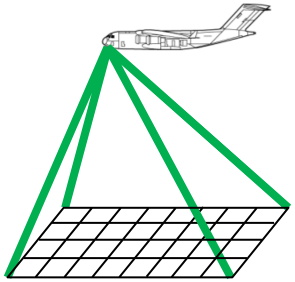
Trackin mera control

Aircraft sensors data



- Receives data about the aircraft speed, position and direction from the Tracking control function
- Performs motion prediction for the tracked object

Tracking & camera control – functional view



Control commands

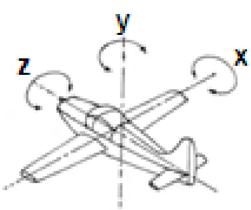
Camera control

Tracking control

Target position prediction

Track & camera control

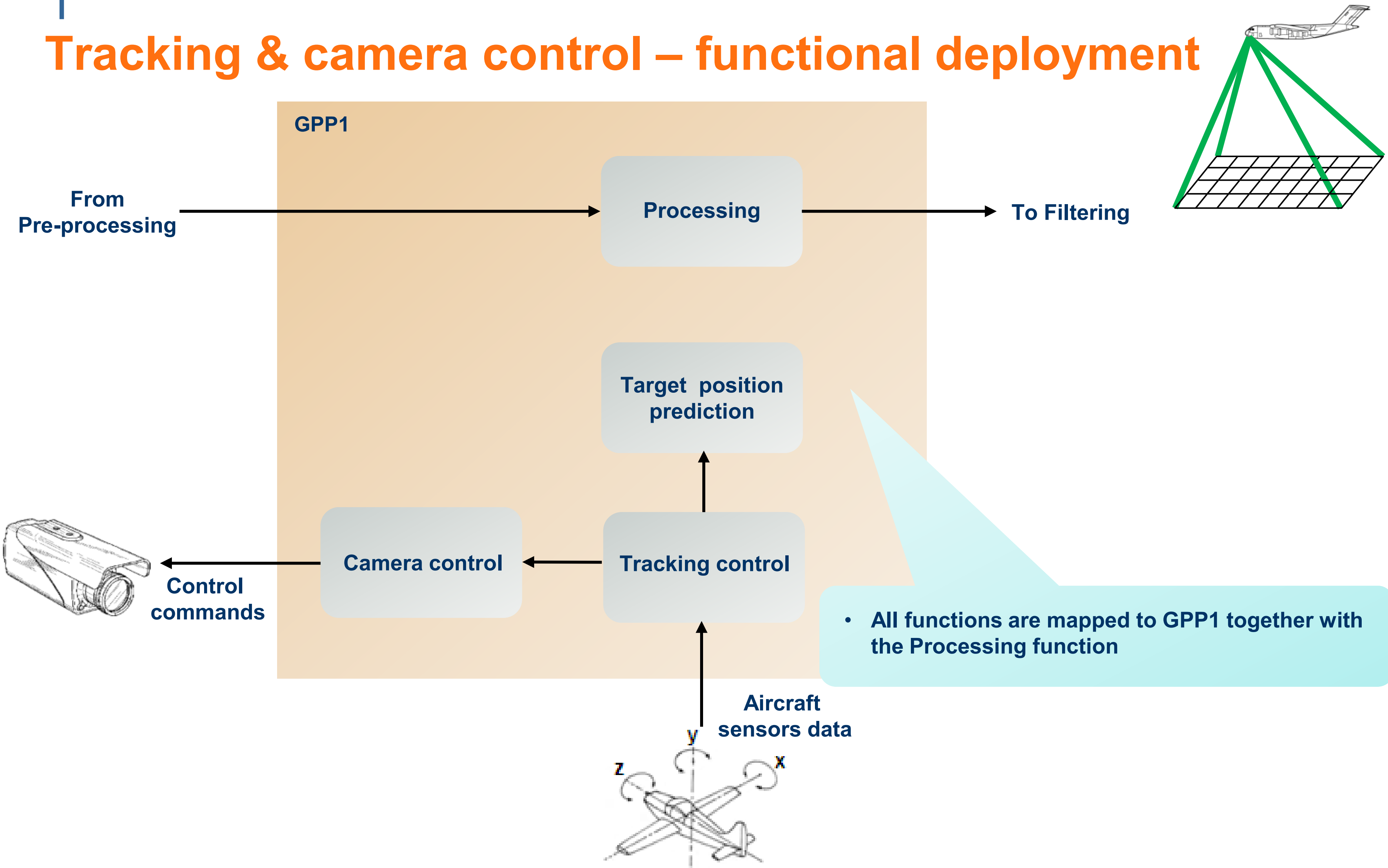
Aircraft sensors data



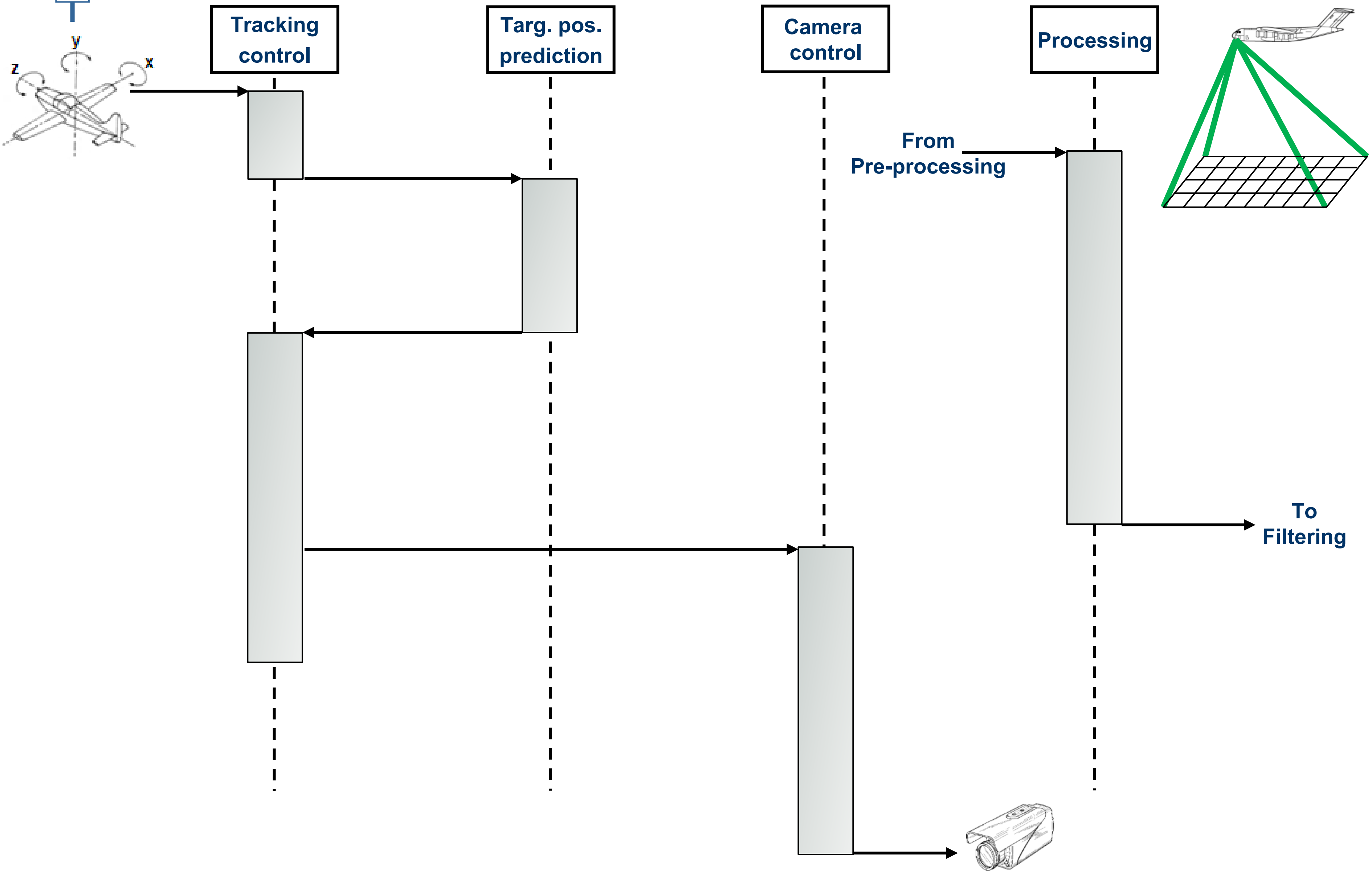
- Receives data about the position of the tracked object from the Tracking control function
- Calculates new angle for the camera based on the aircraft position, speed and direction and the tracked object motion prediction

CHALLENGE 2 – TRACKING & CAMERA CONTROL

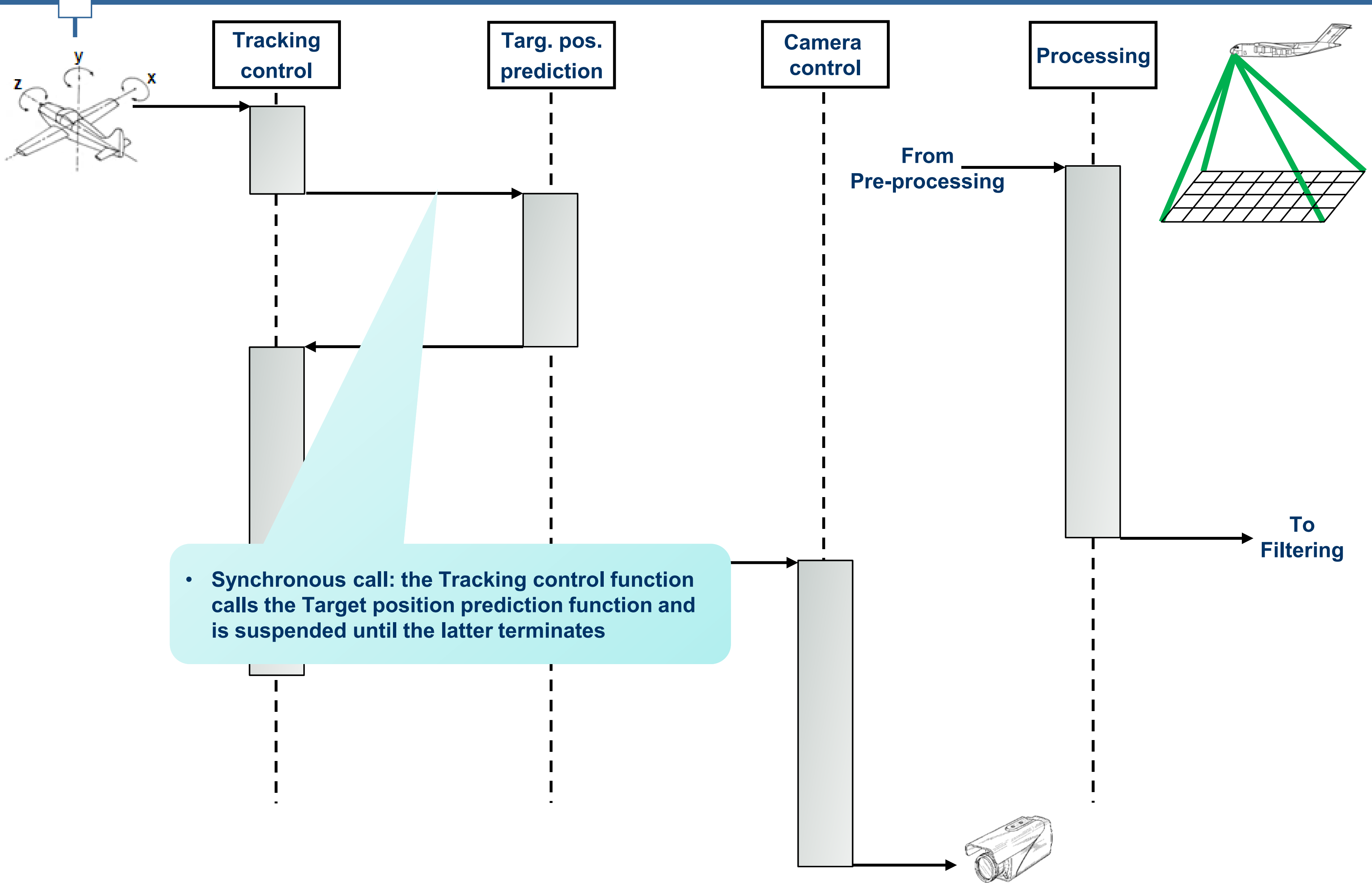
Tracking & camera control – functional deployment



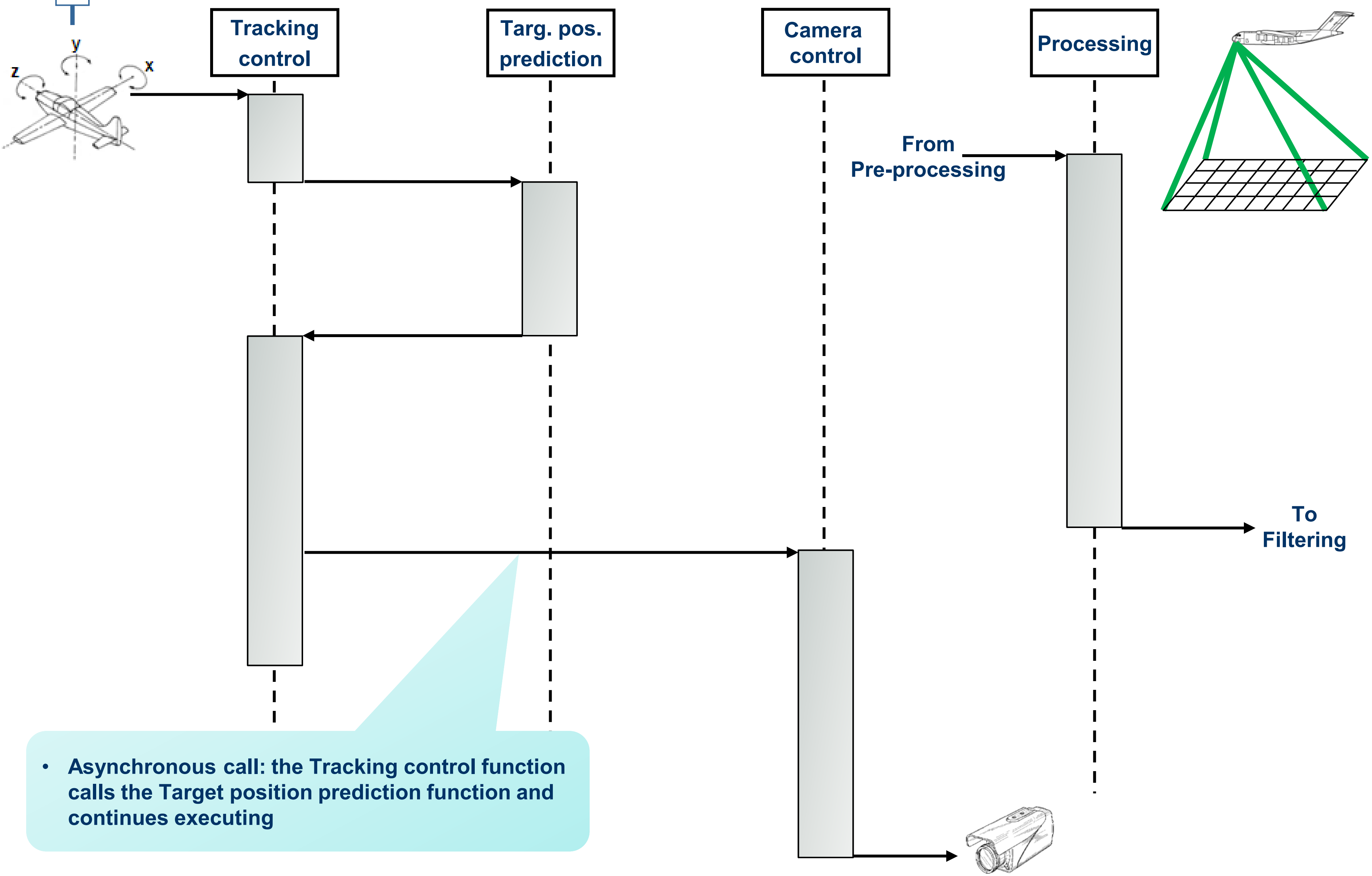
CHALLENGE 2 – TRACKING & CAMERA CONTROL



CHALLENGE 2 – TRACKING & CAMERA CONTROL

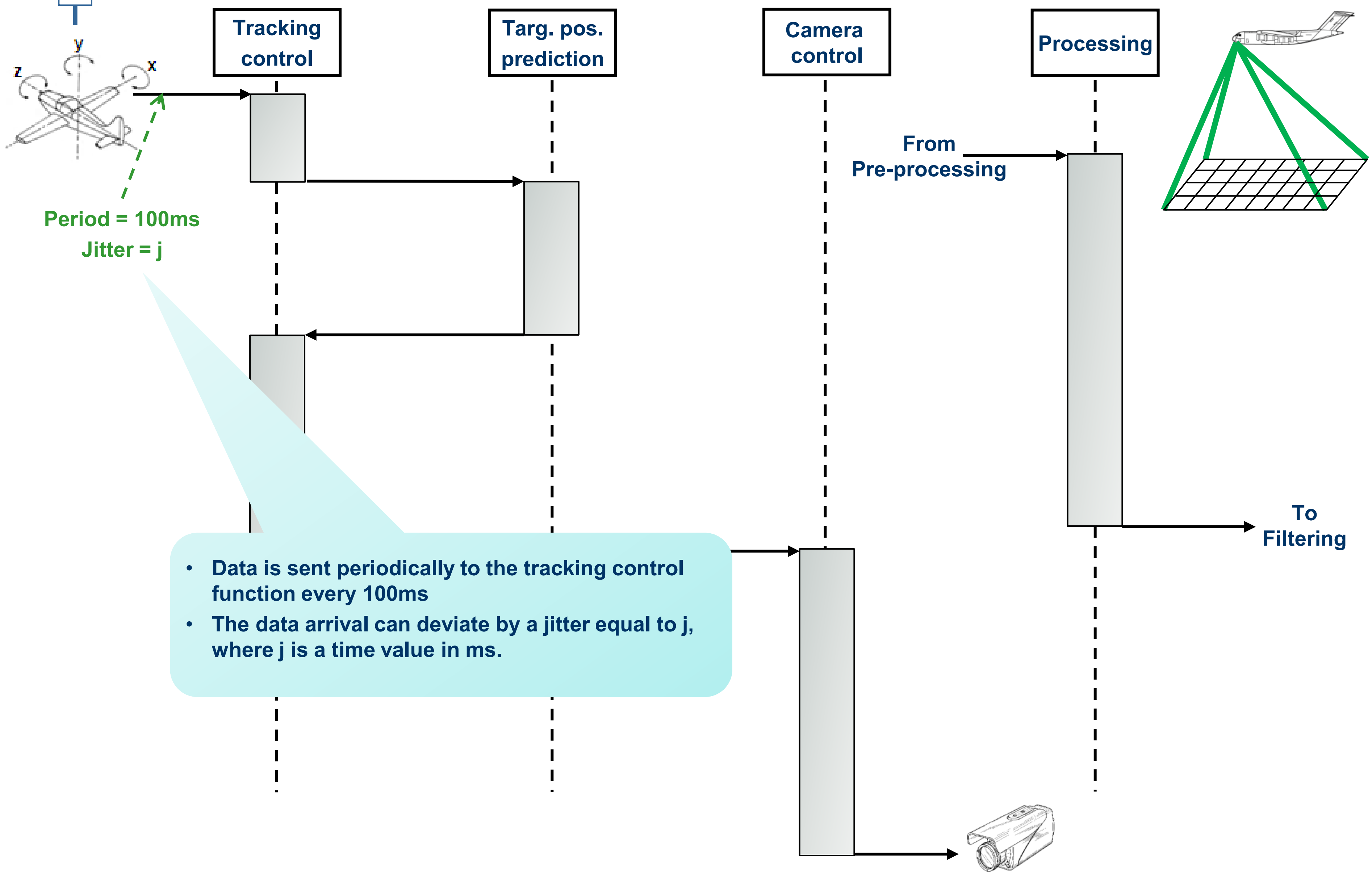


CHALLENGE 2 – TRACKING & CAMERA CONTROL



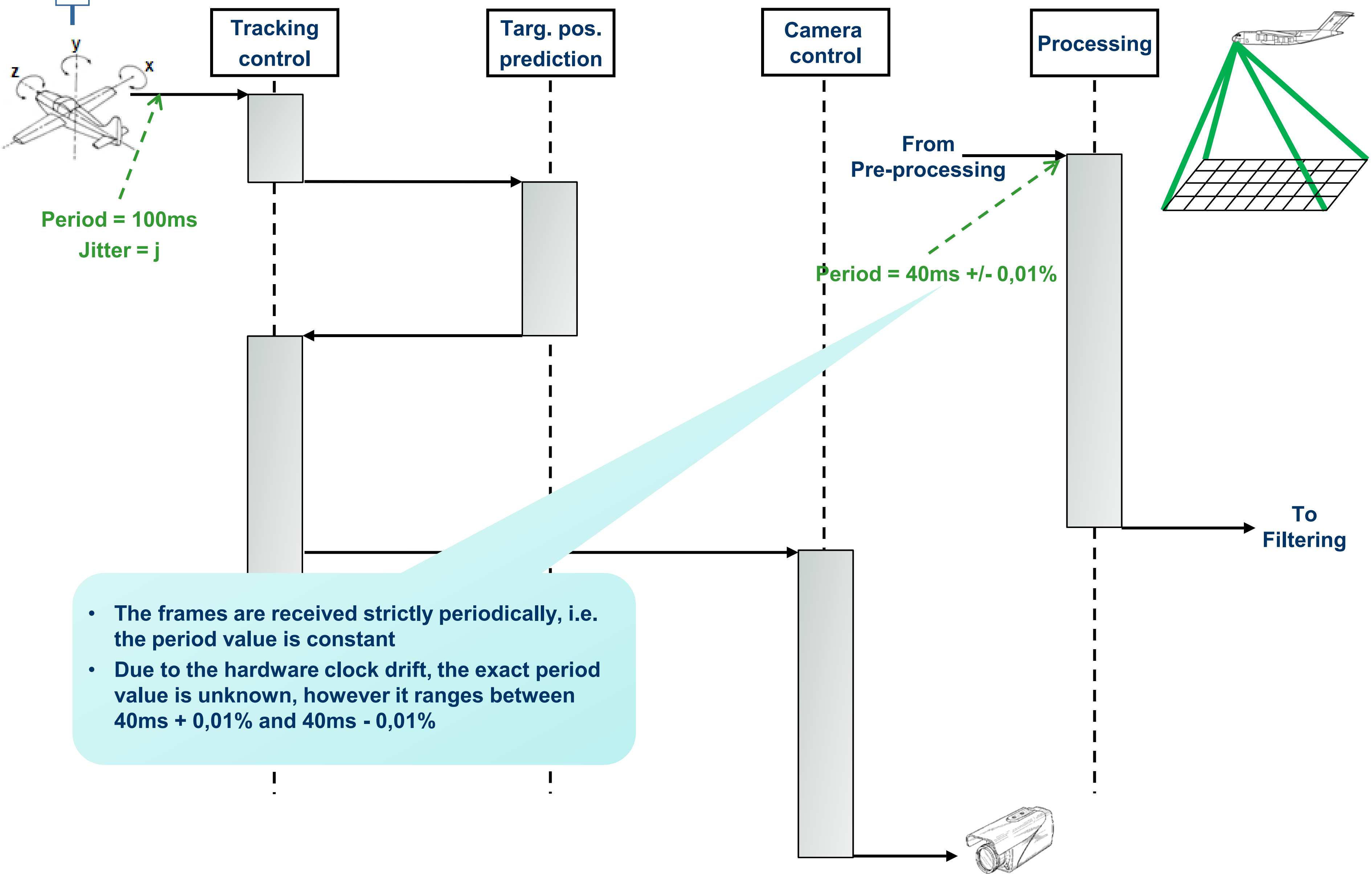
- Asynchronous call: the Tracking control function calls the Target position prediction function and continues executing

CHALLENGE 2 – TRACKING & CAMERA CONTROL



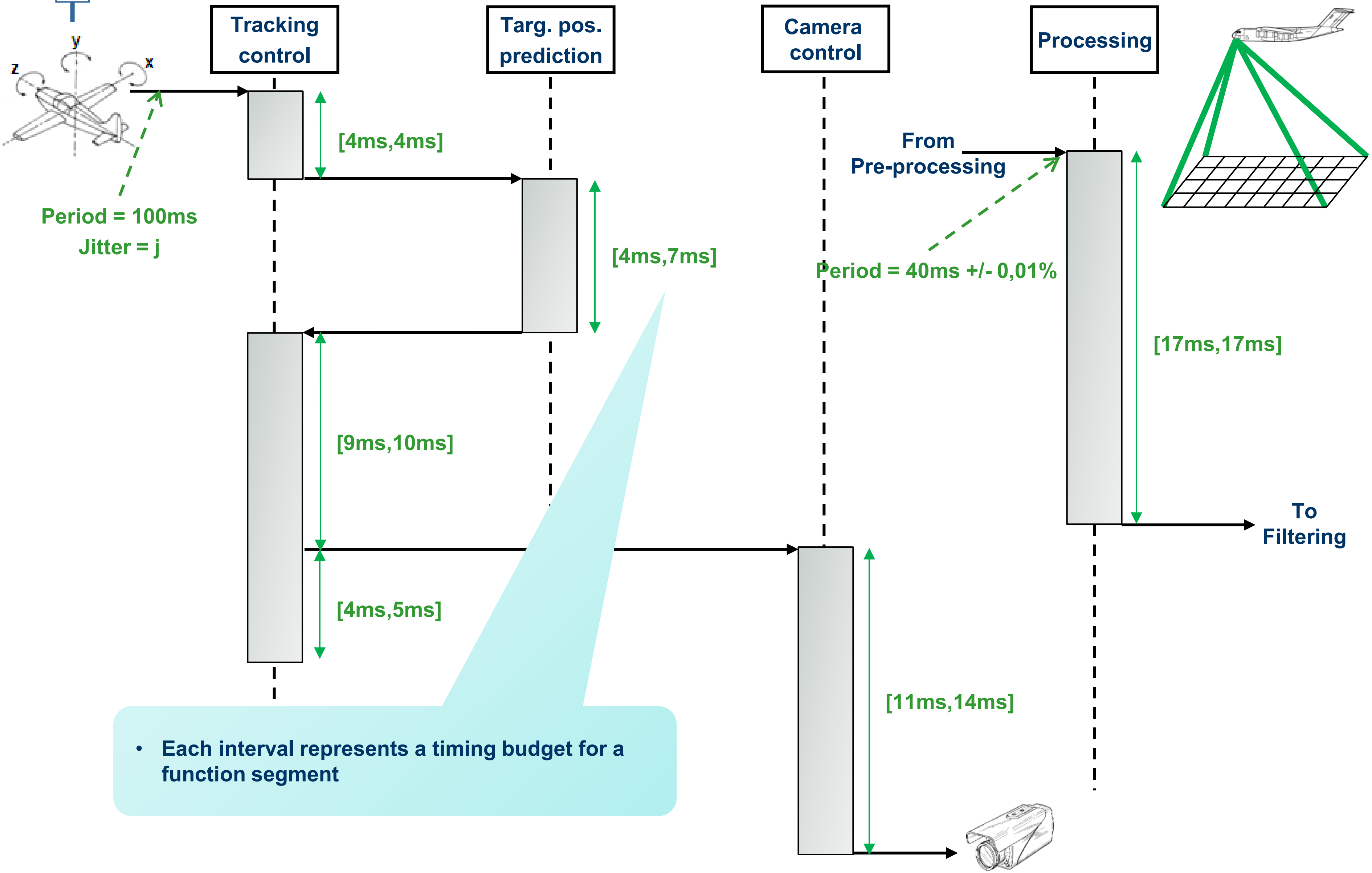
- Data is sent periodically to the tracking control function every 100ms
- The data arrival can deviate by a jitter equal to j, where j is a time value in ms.

CHALLENGE 2 – TRACKING & CAMERA CONTROL



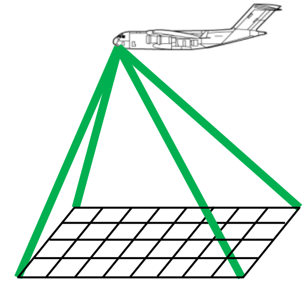
- The frames are received strictly periodically, i.e. the period value is constant
- Due to the hardware clock drift, the exact period value is unknown, however it ranges between 40ms + 0,01% and 40ms - 0,01%

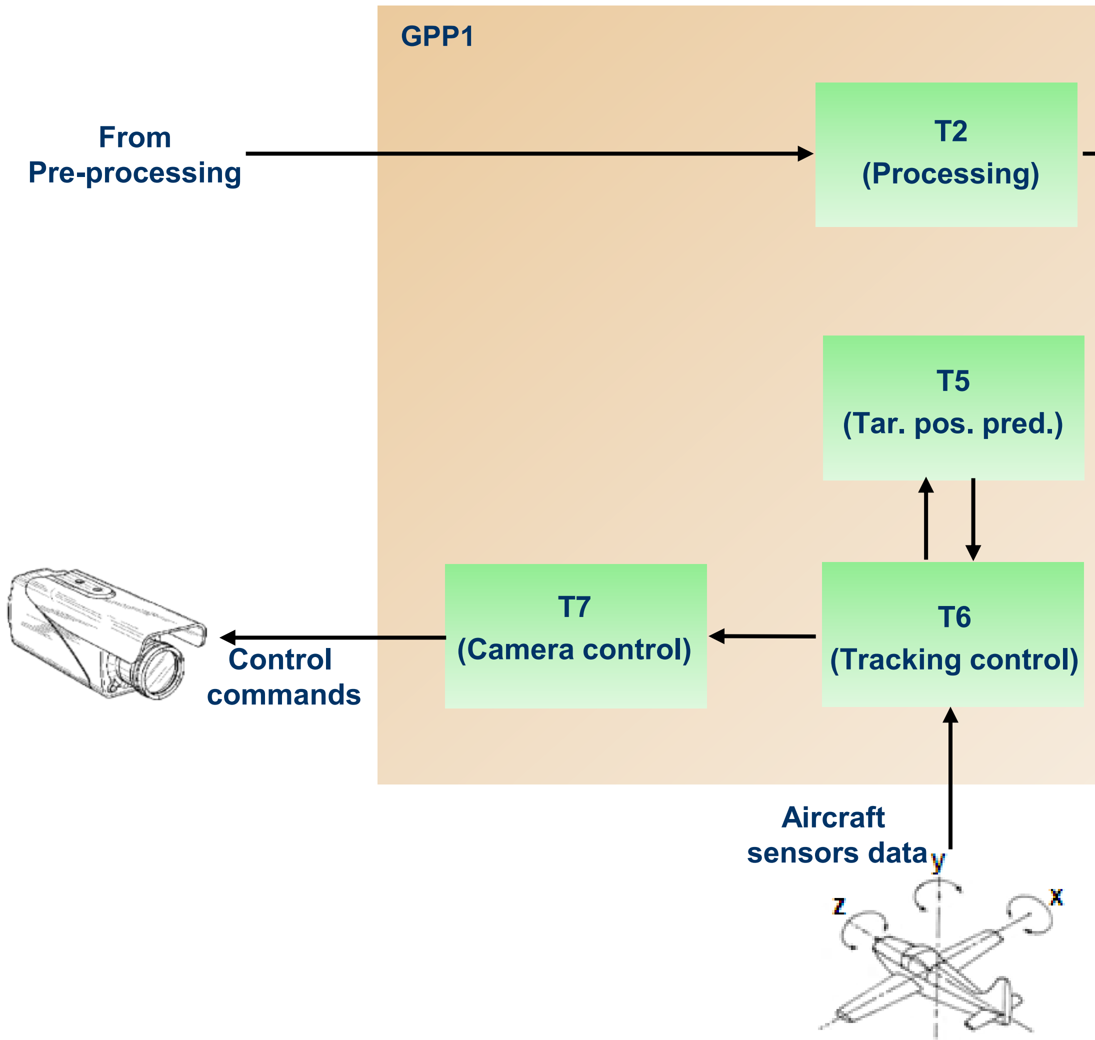
CHALLENGE 2 – TRACKING & CAMERA CONTROL



- Each interval represents a timing budget for a function segment

Tracking & camera control – architectural view

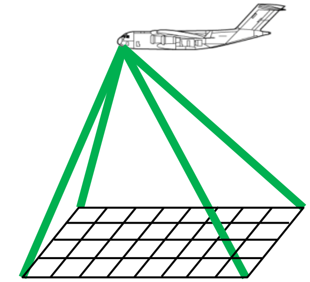




- Each function is mapped to one task
- We assume fixed priority preemptive scheduling on the processor with the following priority order: T2 > T6 > T5 > T7
- All tasks are triggered by the arrival of data at their inputs

CHALLENGE 2 – TRACKING & CAMERA CONTROL

Tracking & camera control – challenge 2A



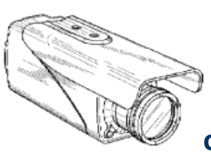
GPP1

From Pre-processing → T2 (Processing) → To Filtering

T5 (Tar. pos. pred.)

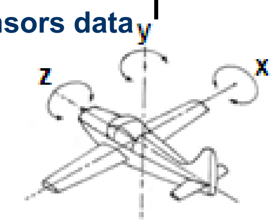
T6 (Tracking control)

T7 (Camera control)



Control commands

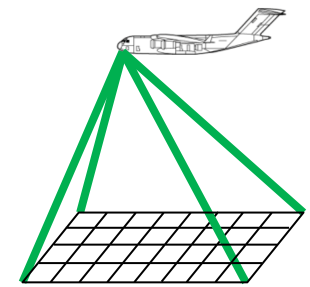
Aircraft sensors data

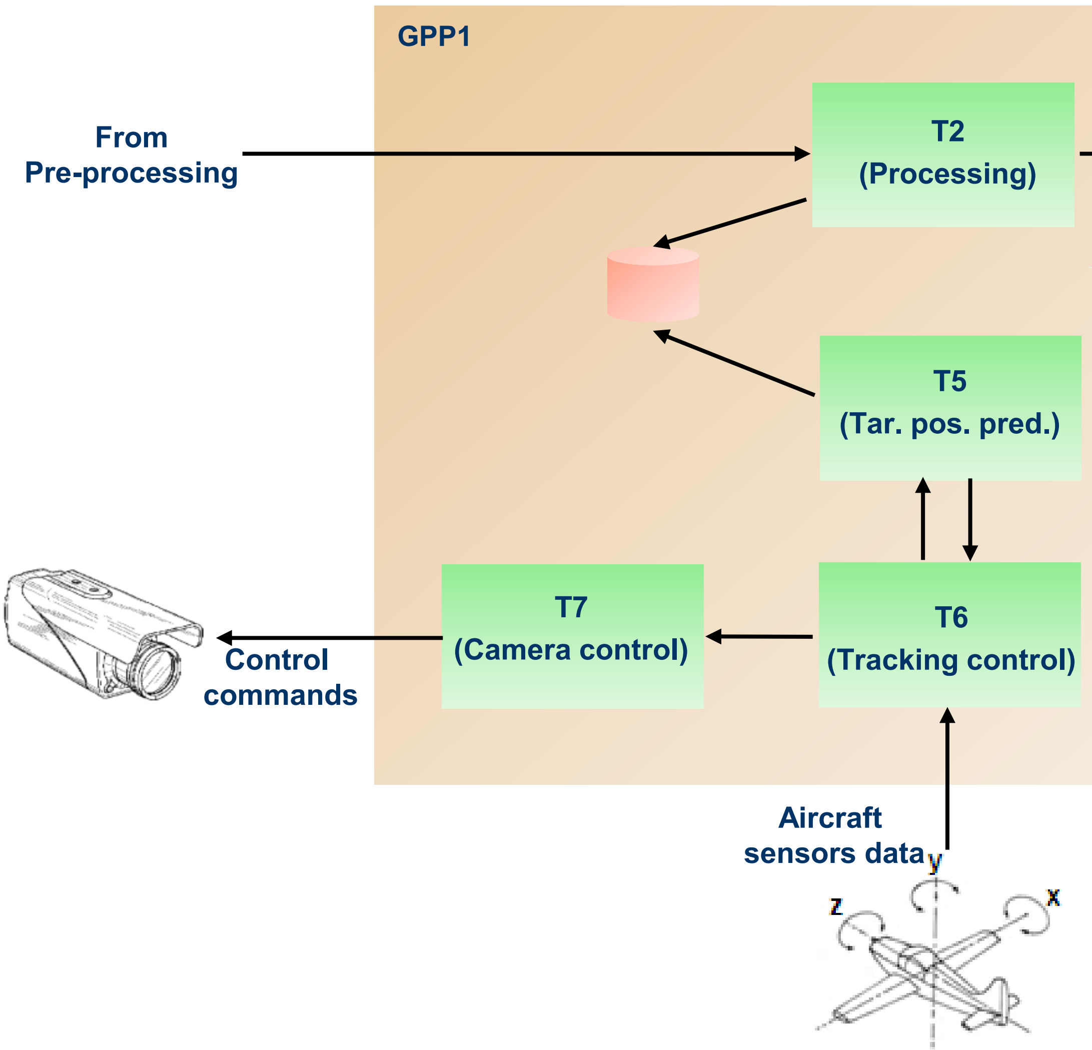


- **Compute:**
  1. the best-case and worst-case end-to-end latencies from activation of T6 to termination of T7 for a jitter value j = 0ms
  2. the best-case and worst-case end-to-end latencies from activation of T6 to termination of T7 for a jitter value j = 20ms

CHALLENGE 2 – TRACKING & CAMERA CONTROL

Tracking & camera control – challenge 2B





- The tasks T2 and T5 have access to a shared resource (prediction requires information from image)
- The resource is mutually exclusive and is protected by a priority ceiling protocol
- The access to the shared resource takes 2ms for both tasks
- Compute:
  1. the best-case and worst-case end-to-end latencies from activation of T6 to termination of T7 for a jitter value j = 0ms
  2. the best-case and worst-case end-to-end latencies from activation of T6 to termination of T7 for a jitter value j = 20ms
  3. The optimum priority assignment minimizing the worst-case latency for a jitter value j = 0ms and j = 20ms

For additional information about the challenge, the submission procedure etc. please visit:

waters2015.inria.fr/challenge