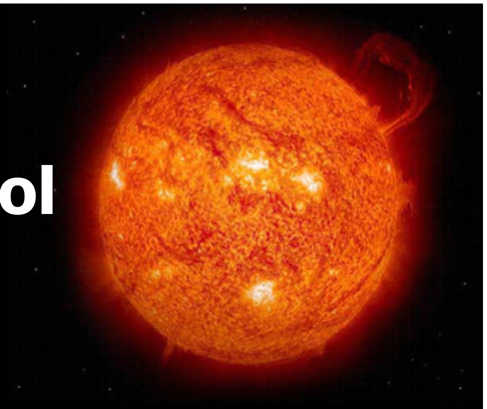
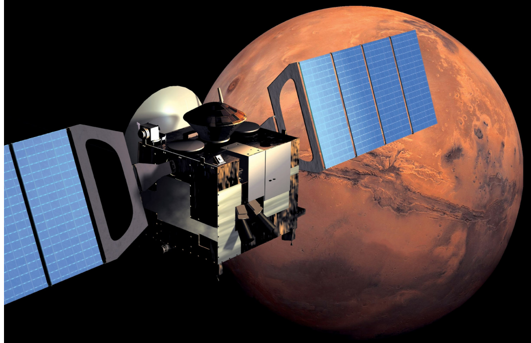
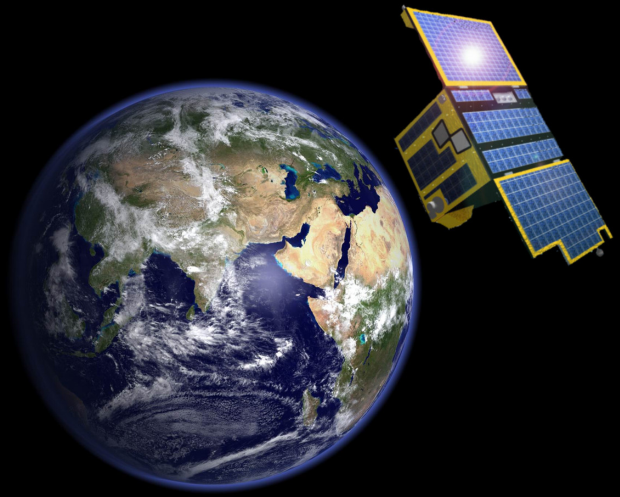
Computer-Aided Software Design for Spacecraft Guidance, Navigation and Control







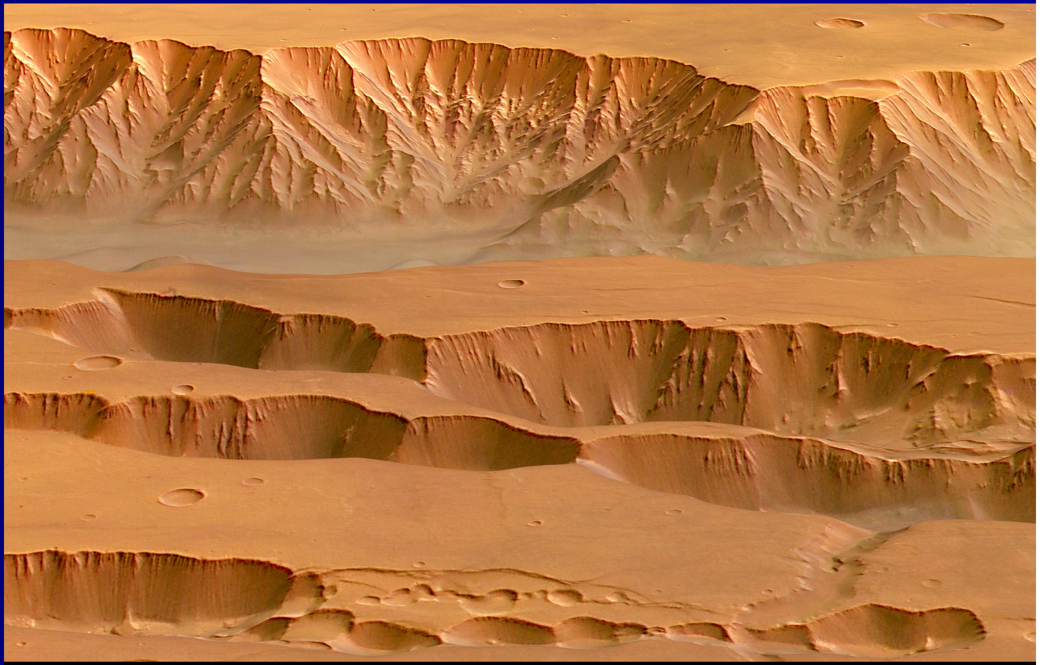
Jean de Lafontaine
President
NGC Aérospatiale Ltée
NGC Aerospace Ltd

2008 IEEE Multiconference on Systems and Control
2-5 September 2008, San Antonio, Texas, USA

Picture courtesy of ESA

BACKGROUND-1

- The European Space Agency (ESA) identified space autonomy as the next enabling technology for:
  - terrestrial missions (Earth observation: environment, security)
  - planetary exploration missions (Mars, Moon, asteroid, comet)

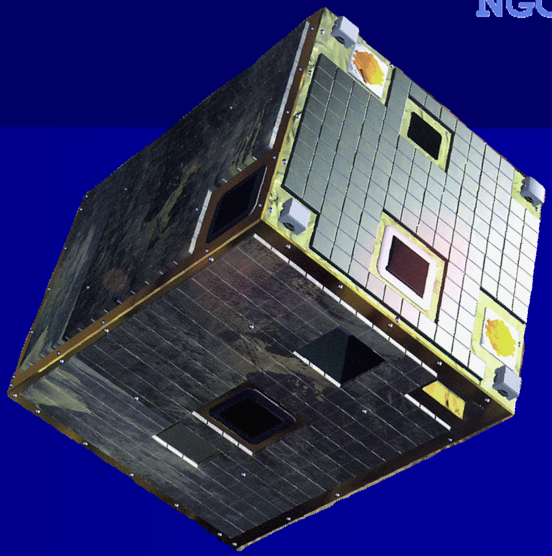


copyright : ESA/DLR/FU Berlin (G. Neukum)

BACKGROUND-2

- High resistance:
  - will not work, will lose control of the spacecraft
  - will increase development cost
- ESA upper management: Let try and see
- Initiated the PROBA programme in 1990's
  - PRoject for On-Board Autonomy ("Probare" = "let's try")
  - demonstrate the benefit of autonomy in space
  - demonstrate new technologies, new S/C development methods
  - launch the PROBA-1 spacecraft within 2 years after start of Phase B

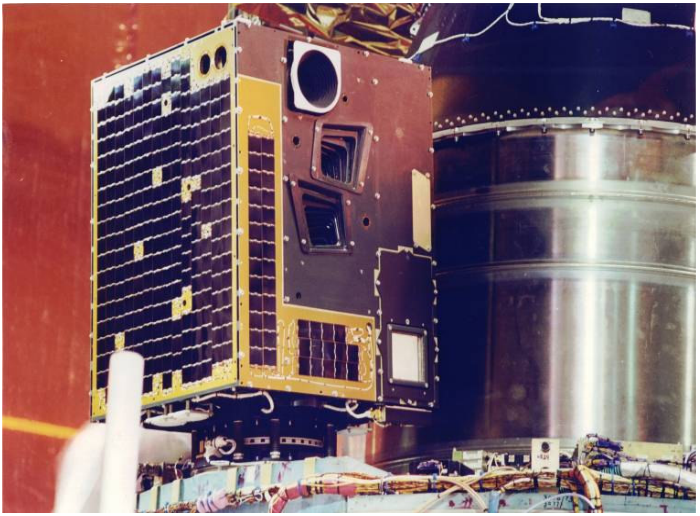
BACKGROUND-3



✧ PROBA-1: Earth-Observation Mission

- launched in October 2001
- 2-year mission
- still successfully operating after 7 years

- 1st fully autonomous ESA spacecraft
- 1st with automatic flight code generation
- 1st with variable-gain Kalman filter
- 1st with complete on-board guidance
- 1st with quaternion-based multivariable gyroless + sliding-mode controller for large-angle manoeuvres

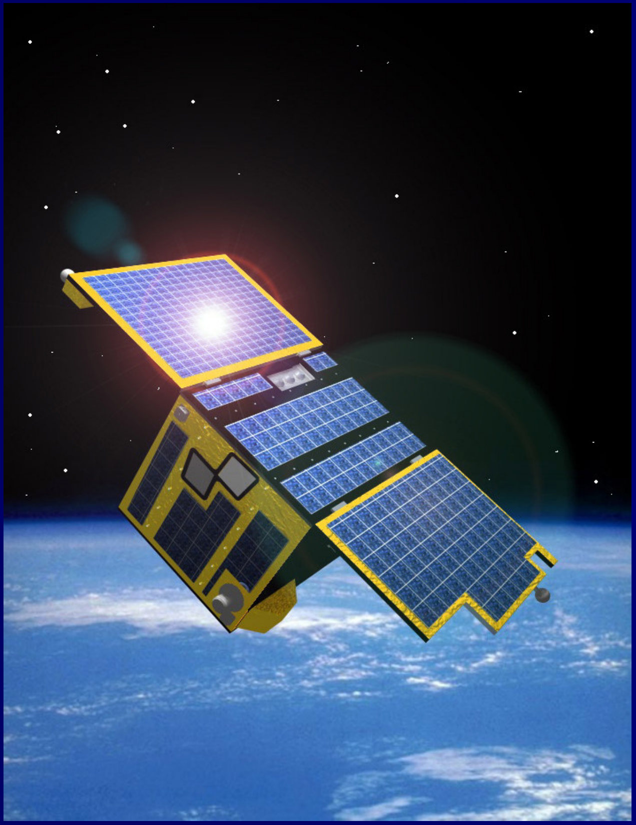


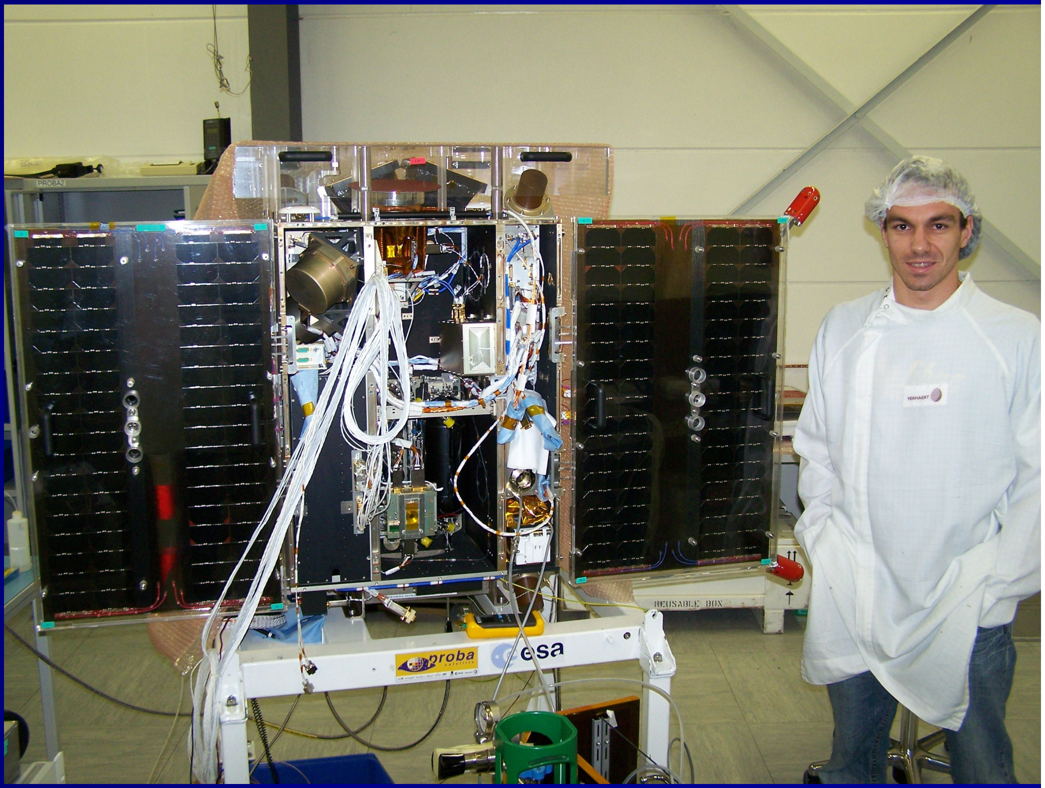
PROBA-1 on the PLSV Launcher

BACKGROUND-4

✧ PROBA-2: Sun-Observation Mission

- to be launched in April 2009
- same autonomy as in PROBA-1 + GNC technology experiment
- magnetic-based state estimation with unscented Kalman Filter





BACKGROUND-5

✧ PROBA-3: Formation-Flight Mission

- to be launched in 2013
- Coronagraph S/C and Occulter S/C on elliptical orbit
- high-accuracy position and attitude determination & control

BACKGROUND-6

- NGC Aerospace was or is currently the contractor for the development of the autonomous GNC system for:
    - PROBA-1
    - PROBA-2
    - PROBA-3 (in negotiation)

- Realisation of these complex on-board software would not have been possible without the use of computer-aided software development tools

OBJECTIVE & OUTLINE

NGC

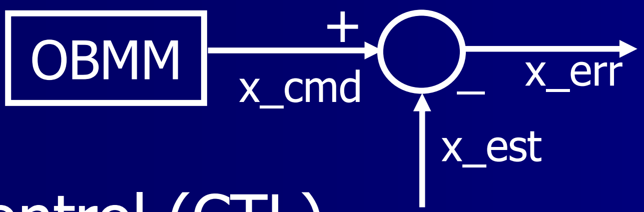
- OBJECTIVE

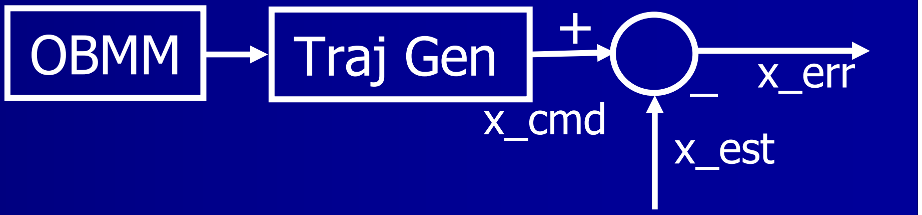
  To demonstrate the need for, and the characteristics of, computer-aided software design for flight-code generation via the particular case of the PROBA flight software

- OUTLINE
  - The Need: Trends in Spacecraft Control System Design
  - The Example: PROBA
  - The Process: The PROBA Software Development
  - The Lessons Learned and the Benefits
  - Conclusions

SOME DEFINITIONS-1

- Navigation (NAV)
  - the determination of the current dynamical state of the vehicle
  - by extension: the determination/calculation of environmental variables (Sun position, Earth attitude, Earth target position)
- Guidance (GDC)
  - the determination of the difference between the estimated state from NAV and the desired state from the Mission Manager
  - the computation of the time history of the desired state

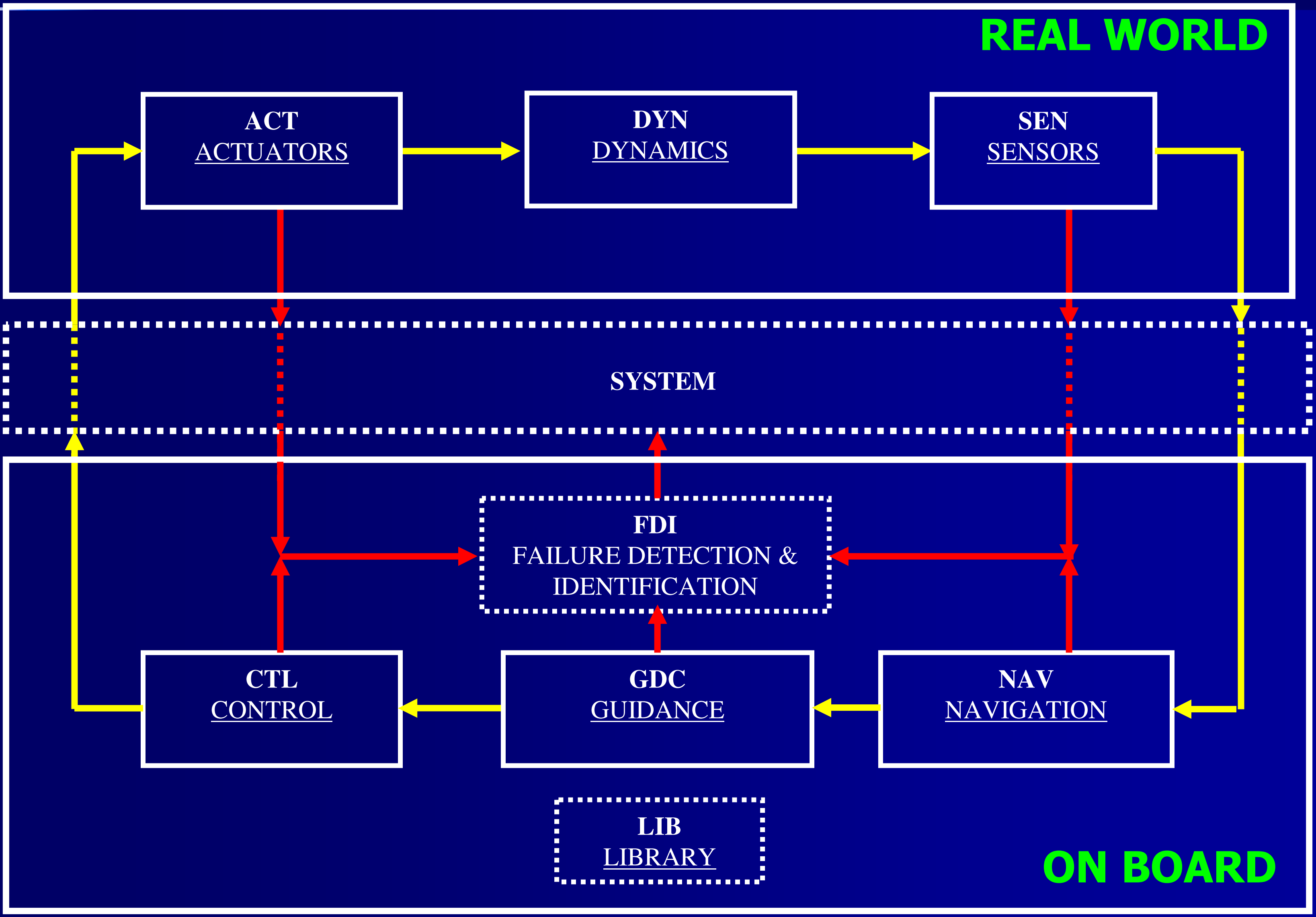




- Control (CTL)
  - the computation of the required actions that will bring the estimated state coincident with the desired state in a stable manner and compliant with performance specifications

SOME DEFINITIONS-2

NGC



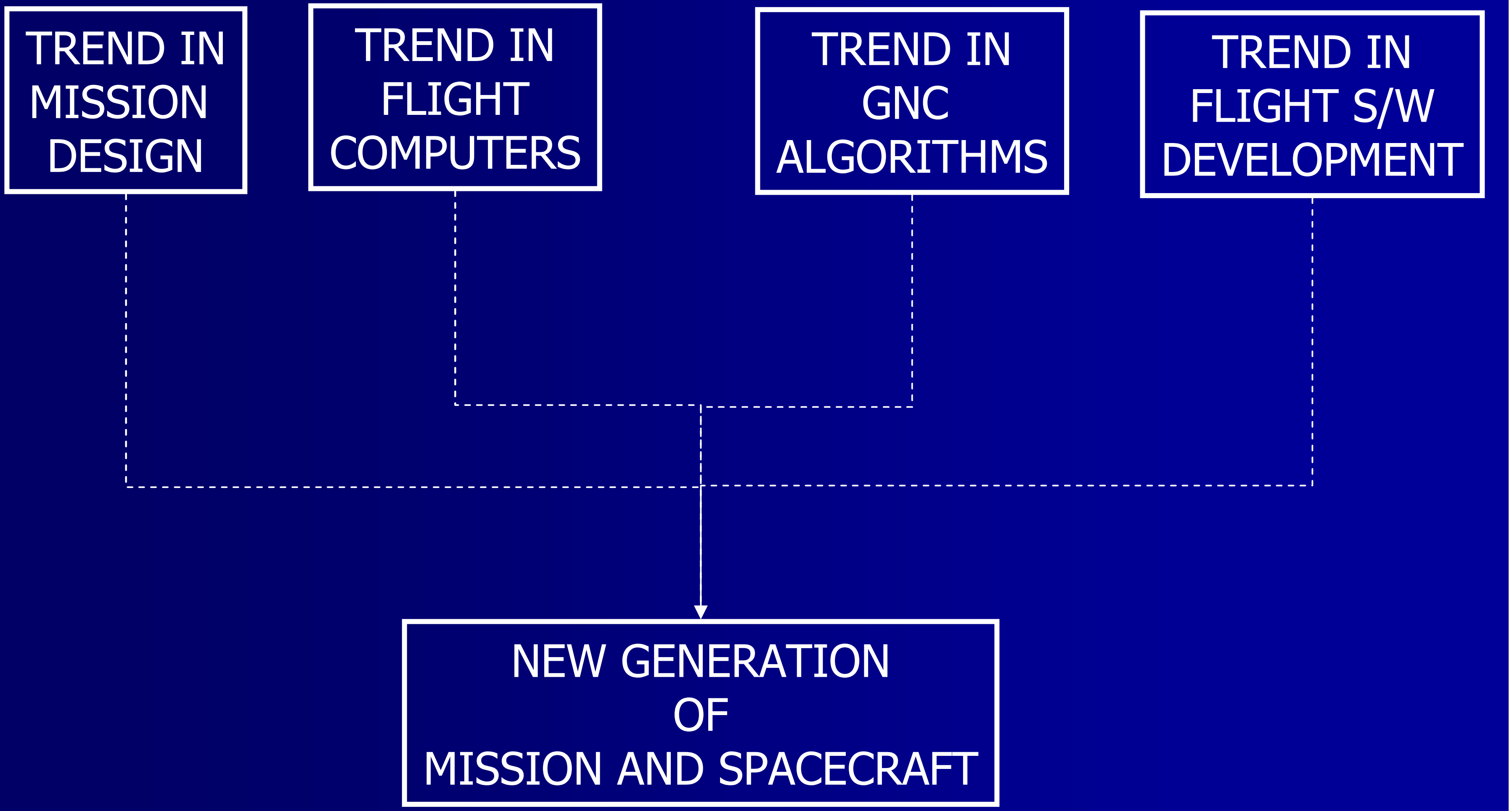
THE NEED AND THE TRENDS

The Need: Trends in S/C GNC Design

What the Users need

The tools we need

The methods we need



TREND IN MISSION DESIGN

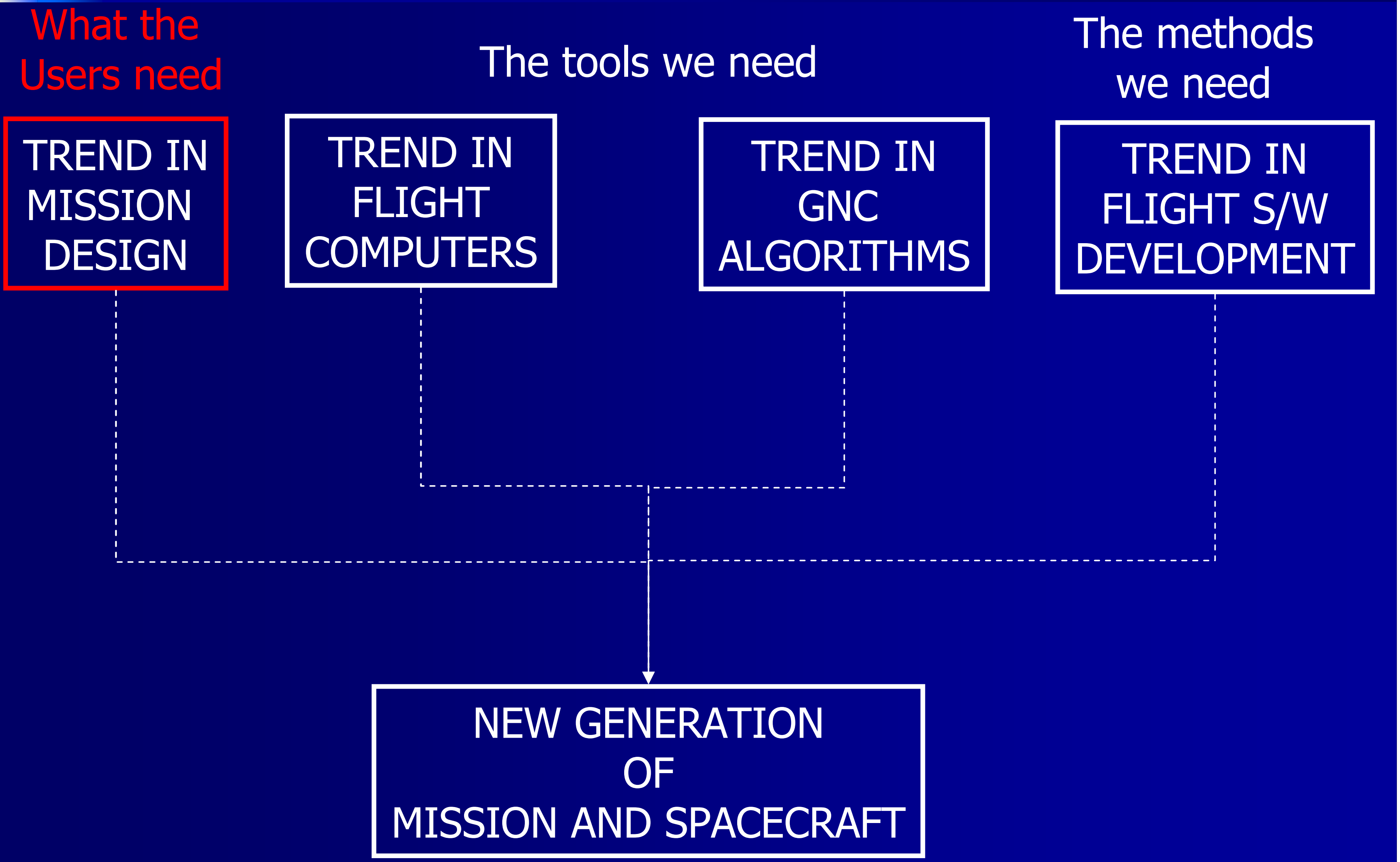
TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

NEW GENERATION OF MISSION AND SPACECRAFT

The Need: Trends in S/C GNC Design



What the Users need

The tools we need

The methods we need

TREND IN MISSION DESIGN

TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

NEW GENERATION OF MISSION AND SPACECRAFT

TRENDS IN SPACE MISSION DESIGN-1

✧ Smaller spacecraft on low-cost missions

- increased needs in environment monitoring and security
- formation flight of many S/C instead of large S/C
- availability of cheaper "piggy-back" launches
- smaller, cheaper sensors and actuators

⇒ reduction of development costs and operational costs

→ **autonomy**

TRENDS IN SPACE MISSION DESIGN-2

✧ Spacecraft autonomy ⇒ GNC autonomy

- cost: smaller staff at the ground station for operations
- efficiency: quick correction of in-flight anomalies
- accuracy: real-time state measurements vs predicted
- no choice: in some missions, the signal time-of-flight precludes closing the control loop via the Earth station (e.g. Mars landing)

⇒ ‘intelligent’ flight software ⇒ larger development costs

TRENDS IN SPACE MISSION DESIGN-3

✧ CONCLUSIONS

- lower operational costs ⇒ spacecraft autonomy
- lower development costs

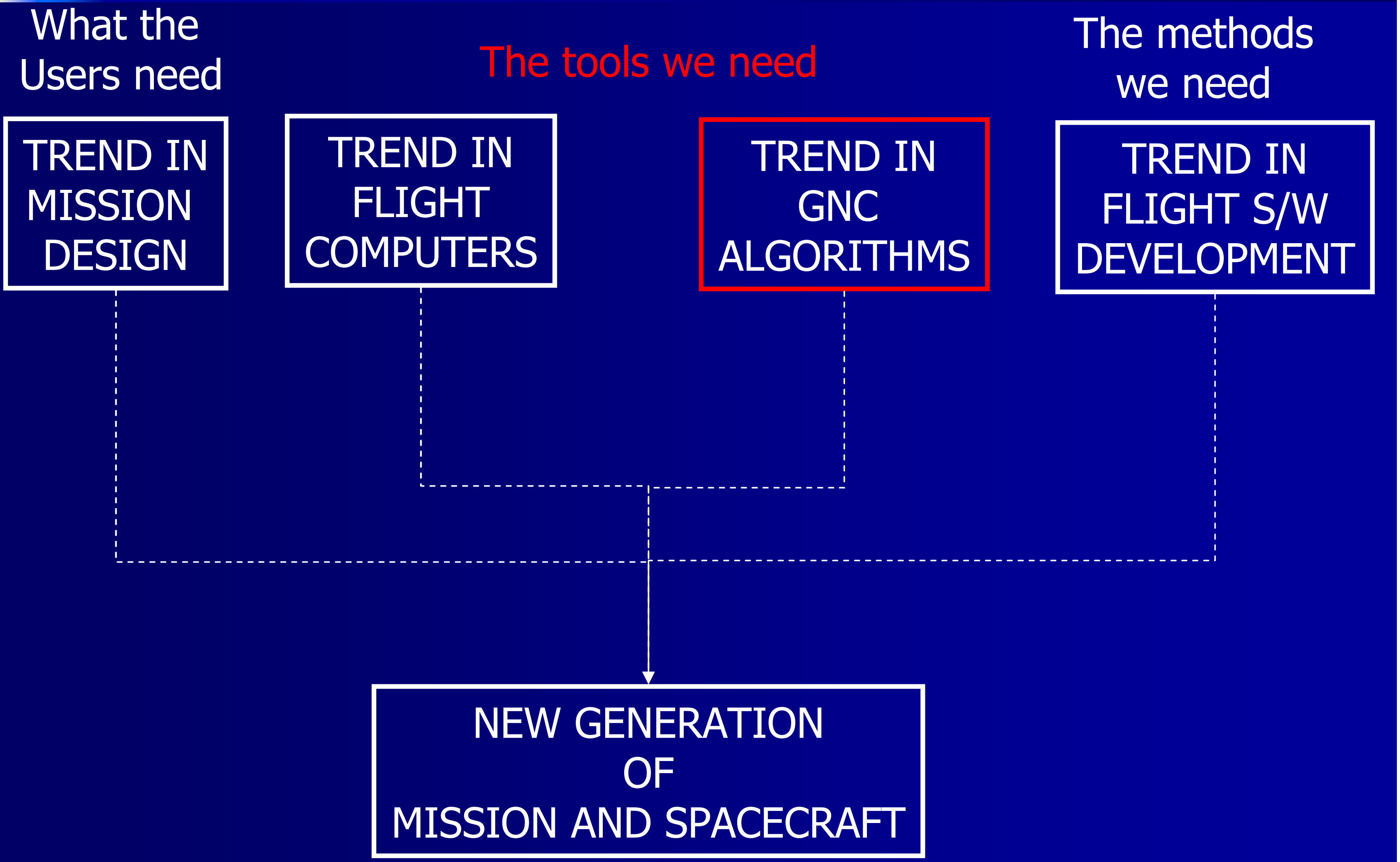
but…

next

- intelligent on-board software
- higher software development costs

$(one line of code in space) = 2 x $(same line of code on ground)

The Need: Trends in S/C GNC Design



What the Users need

The tools we need

The methods we need

TREND IN MISSION DESIGN

TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

NEW GENERATION OF MISSION AND SPACECRAFT

TRENDS IN S/C GNC ALGORITHMS-1

✧ NAVIGATION: the traditional

- sensor output feedback for attitude
- ground-based orbit determination

✧ NAVIGATION: the trend

- state feedback
- Extended Kalman Filter, Unscented Kalman Filter
- autonomous star sensor, GPS-based orbit determination

⇒ sensor delay recovery, sensor outage compensation, measurement interpolation, sensor fusion

⇒ more complex on-board software

TRENDS IN S/C GNC ALGORITHMS-2

✧ GUIDANCE: the traditional

- ground-based state-trajectory computation
- uplink of polynomial coefficients for reference attitude

✧ GUIDANCE: the trend

- on-board computation of reference attitude profile
- on-board computation of reference trajectory profile

⇒ autonomous on-board decision

⇒ more complex on-board software

TRENDS IN S/C GNC ALGORITHMS-3

✧ CONTROL: the traditional

- decoupling assumption: one controller per axis ⇒ SISO
- PID controller, lead-lag controllers, flexibility filters

✧ CONTROL: the trend

- multivariable control of coupled dynamics ⇒ MIMO
- LQG/LQR control, robust control, adaptive control, predictive control, nonlinear control, sliding-mode control
- nonlinear dynamic inversion, robust dynamic inversion

⇒ better performance of ‘intelligent’ algorithms
⇒ higher design complexity, higher controller complexity
⇒ more complex on-board software

TRENDS IN S/C GNC ALGORITHMS-4

NGC

✧ CONCLUSIONS

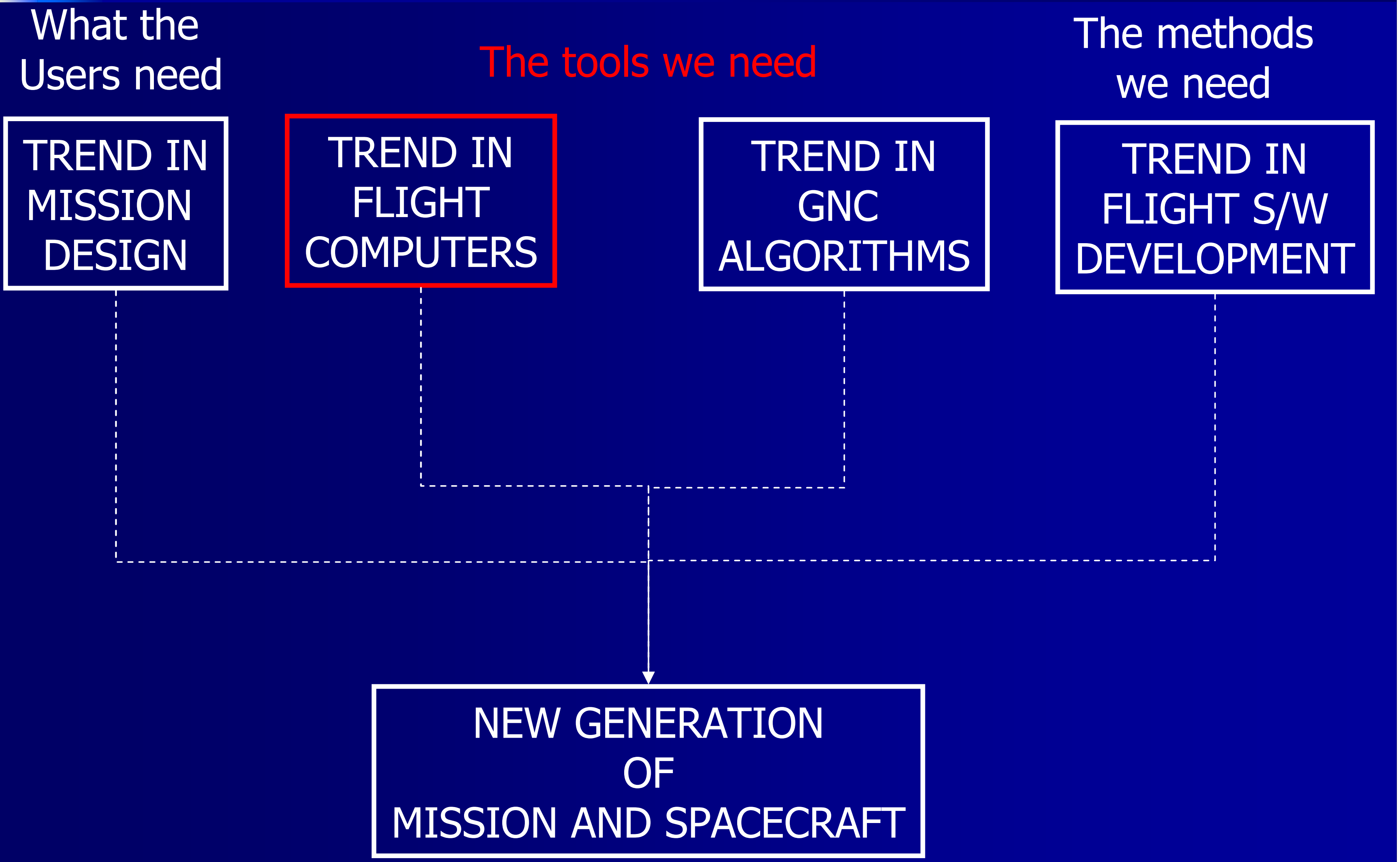
- intelligent GNC software with…
- better performance
- better autonomy

but…

- more complex on-board software
- higher software development costs
- more demanding on-board computer resources

→ **next**

The Need: Trends in S/C GNC Design



What the Users need

The tools we need

The methods we need

TREND IN MISSION DESIGN

TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

NEW GENERATION OF MISSION AND SPACECRAFT

TRENDS IN FLIGHT COMPUTERS-1

✧ Hardwired Control System

- analogue link between sensors and actuators
- analogue/hybrid computer for verification & validation
- no in-flight reprogramming
- limited to simple input-output relationships

✧ Microprocessor-based GNC System

- digital link between sensors and actuators
- digital computer for verification & validation
- in-flight reprogramming possible
- complexity of the software only limited by memory, computing power and ability to validate and verify the software before flight
- 10 MIPS (2001), 40 MIPS (2006), 100 MIPS, 500 MIPS

TRENDS IN FLIGHT COMPUTERS-2

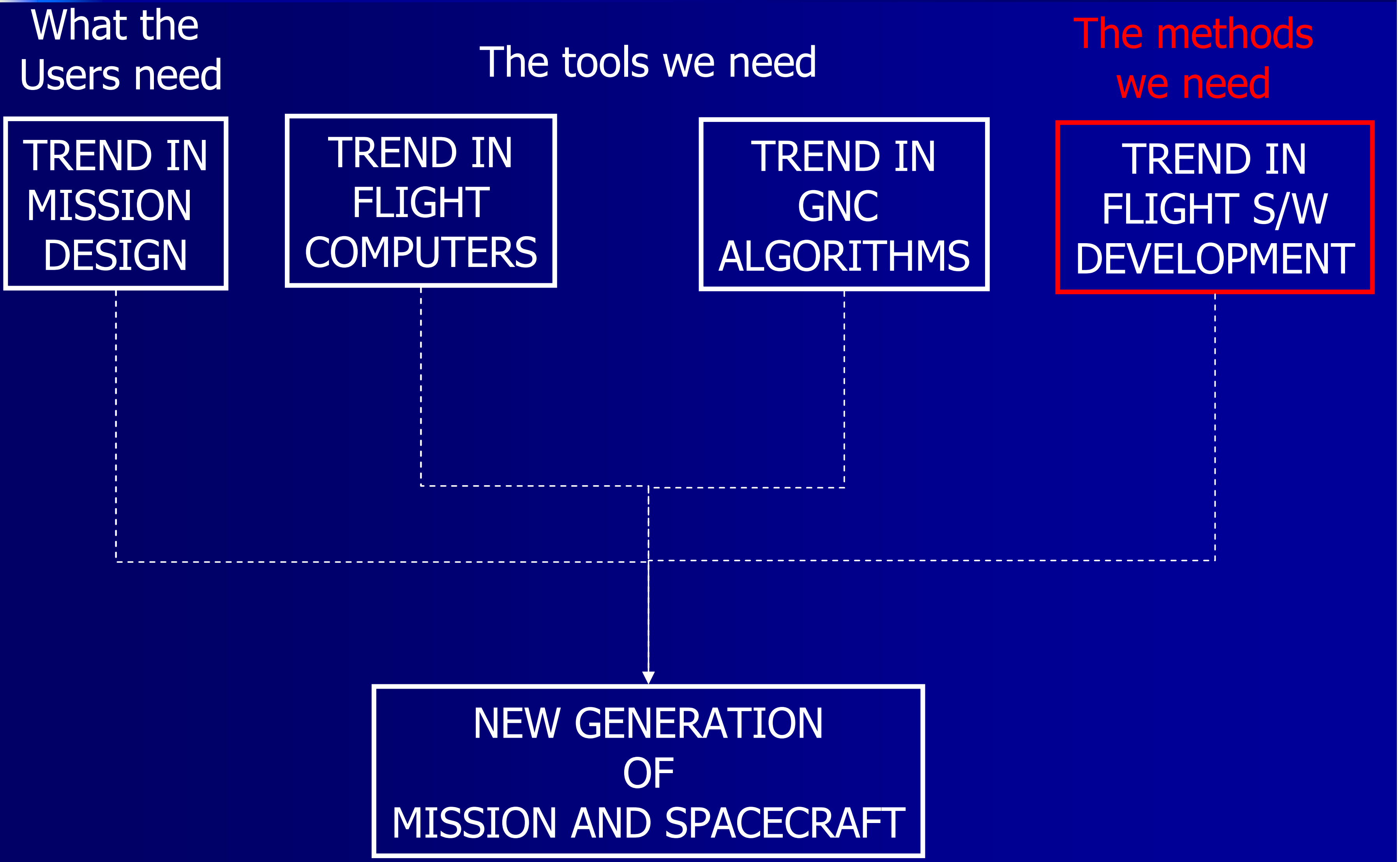
✧ CONCLUSIONS

- space-qualified computers are more powerful
- can cope with more complex GNC algorithms

but…

- more complex on-board software remains
- higher software development costs remain

The Need: Trends in S/C GNC Design



What the Users need

The tools we need

The methods we need

TREND IN MISSION DESIGN

TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

NEW GENERATION OF MISSION AND SPACECRAFT

TRENDS IN FLIGHT CODE DEVLOPMENT

Evolution roughly organised into 4 generations

- 1st generation:
  - paper design, home-made computer tools for validation
  - hand-coding in low-level language (Assembler)
  - limited flight-code validation with flight computer
- 2nd generation:
  - computer-aided tools for design/validation (CASE tools)
  - hand-coding at high level (C or ADA)
  - home-made simulator for flight-code validation

Manual generation of the flight code

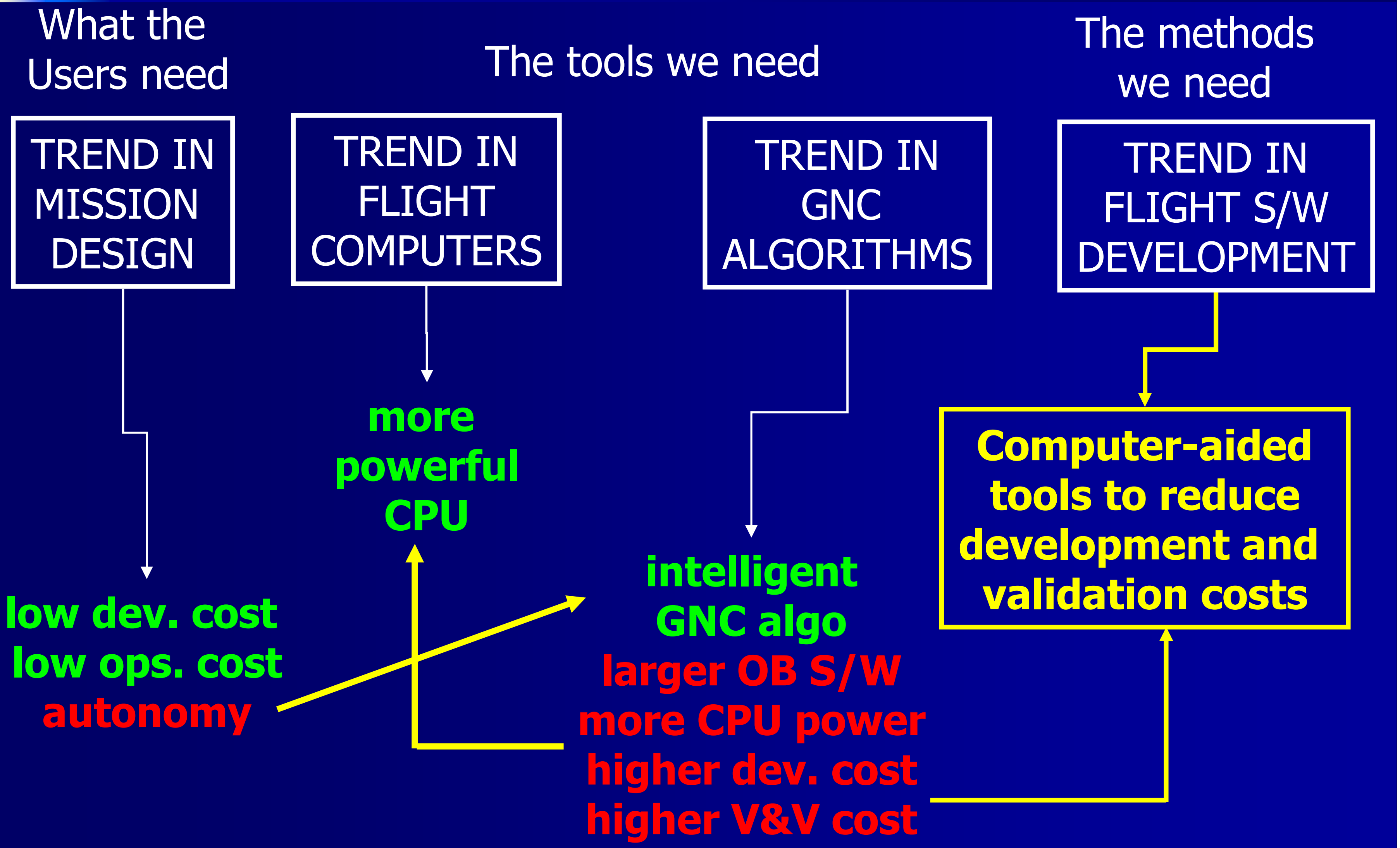
- 3rd generation:
  - CASE tools for design/validation
  - CASE tool for automatic flight-code generation
  - home-made simulator for flight-code validation

Automatic generation of the flight code

- 4th generation (the PROBA generation):
  - single CASE tool from conceptual design to flight-code validation

Automatic validation

The Need: CONCLUSIONS



What the Users need

The tools we need

The methods we need

TREND IN MISSION DESIGN

TREND IN FLIGHT COMPUTERS

TREND IN GNC ALGORITHMS

TREND IN FLIGHT S/W DEVELOPMENT

more powerful CPU

low dev. cost
low ops. cost
autonomy

intelligent GNC algo
larger OB S/W
more CPU power
higher dev. cost
higher V&V cost

Computer-aided tools to reduce development and validation costs

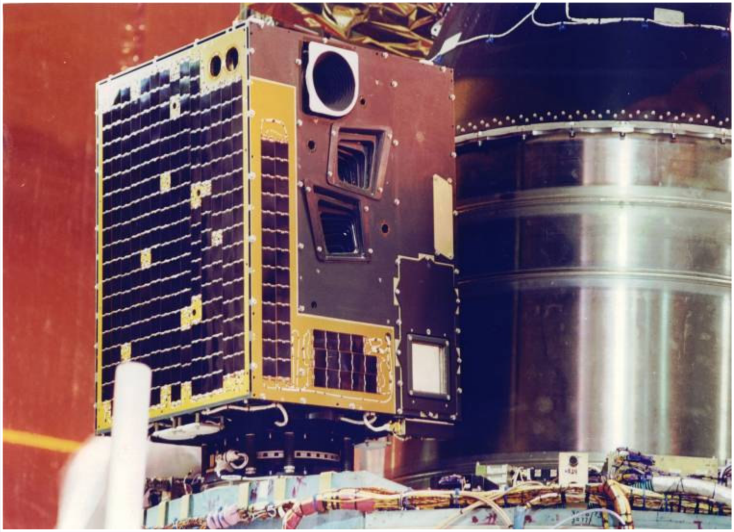
THE EXAMPLE

THE PROBA-1 MISSION

✧ The PROBA-1 Mission

- demonstration of autonomy in space
- Earth observation with two instruments
  - hyperspectral camera (color) @ 20m
  - high-resolution camera (black & white) @ 4m

PROBA-1 Launch



PROBA-1 on the PLSV Launcher

ORBIT

615 km altitude

SPACECRAFT

95 kg, 600 X 600 X 800 mm (*a big TV*)

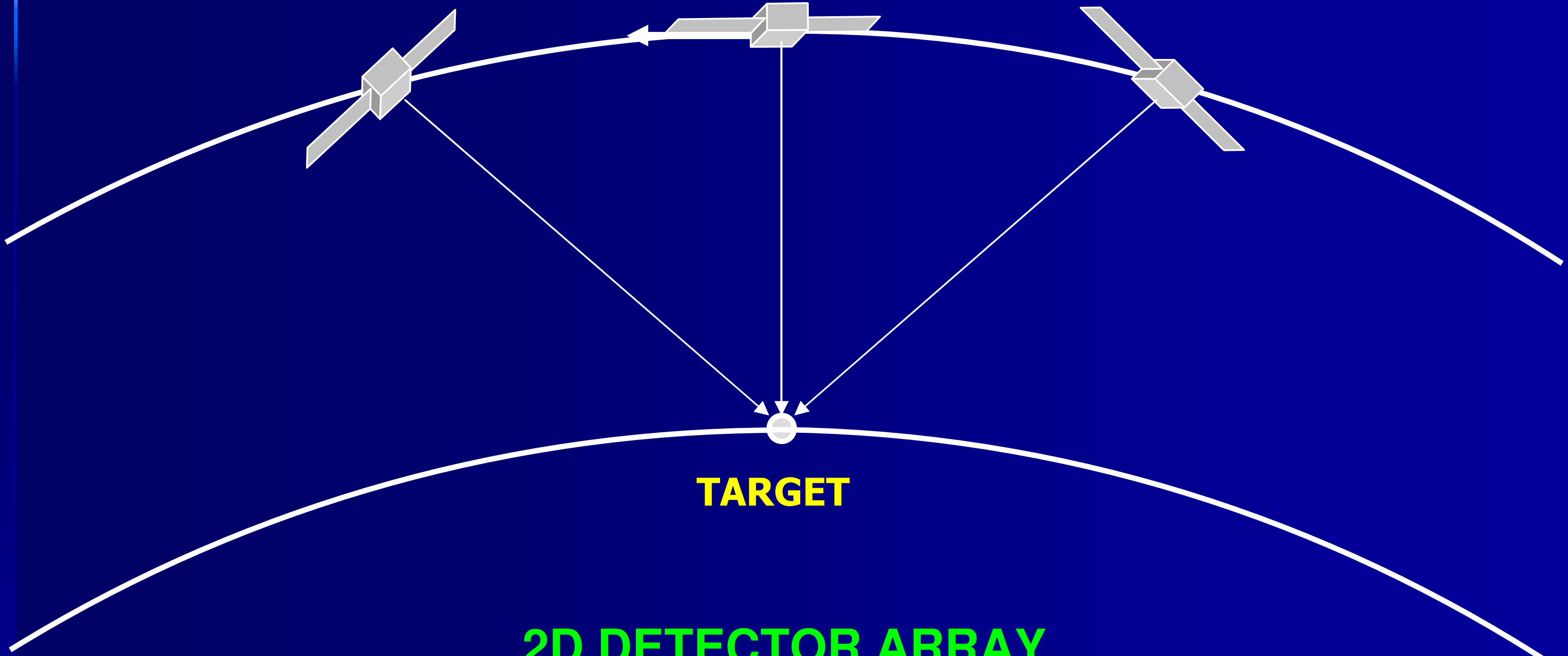
40 W average power (*a light bulb*))

LAUNCH & OPERATION

Launched 22 October 2001

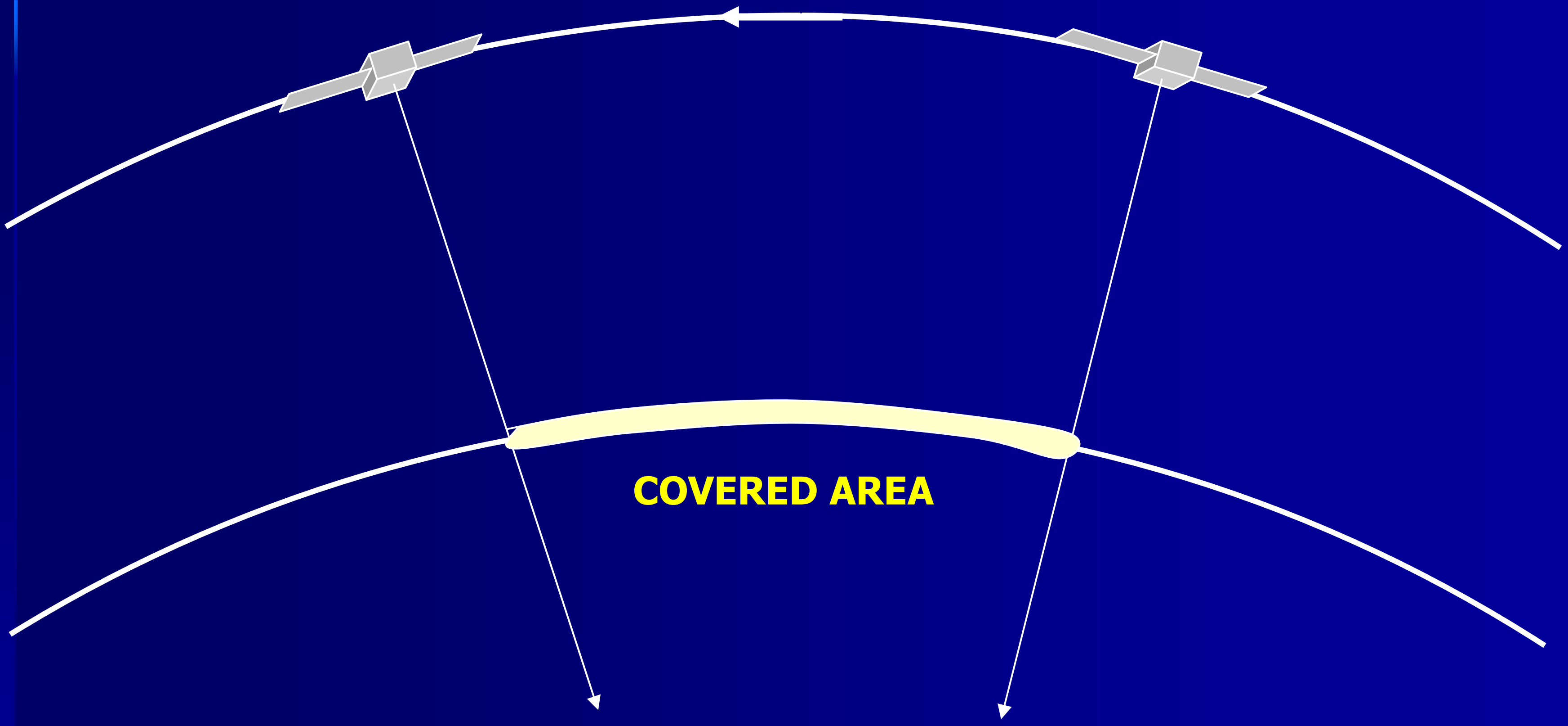
Still operating successfully

THE PROBA-1 MISSION



2D DETECTOR ARRAY
CCD-TYPE CAMERA
B&W CAMERA @ 4m

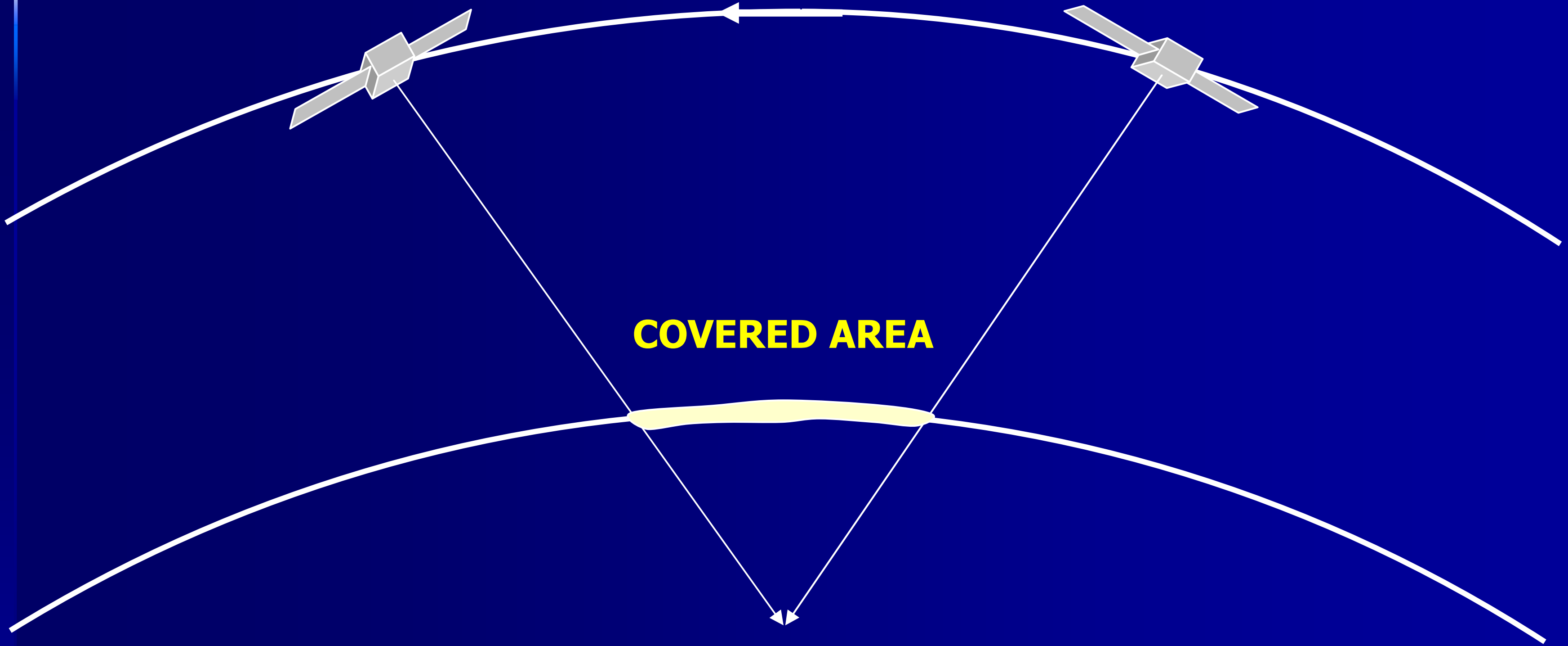
THE PROBA-1 MISSION



COVERED AREA

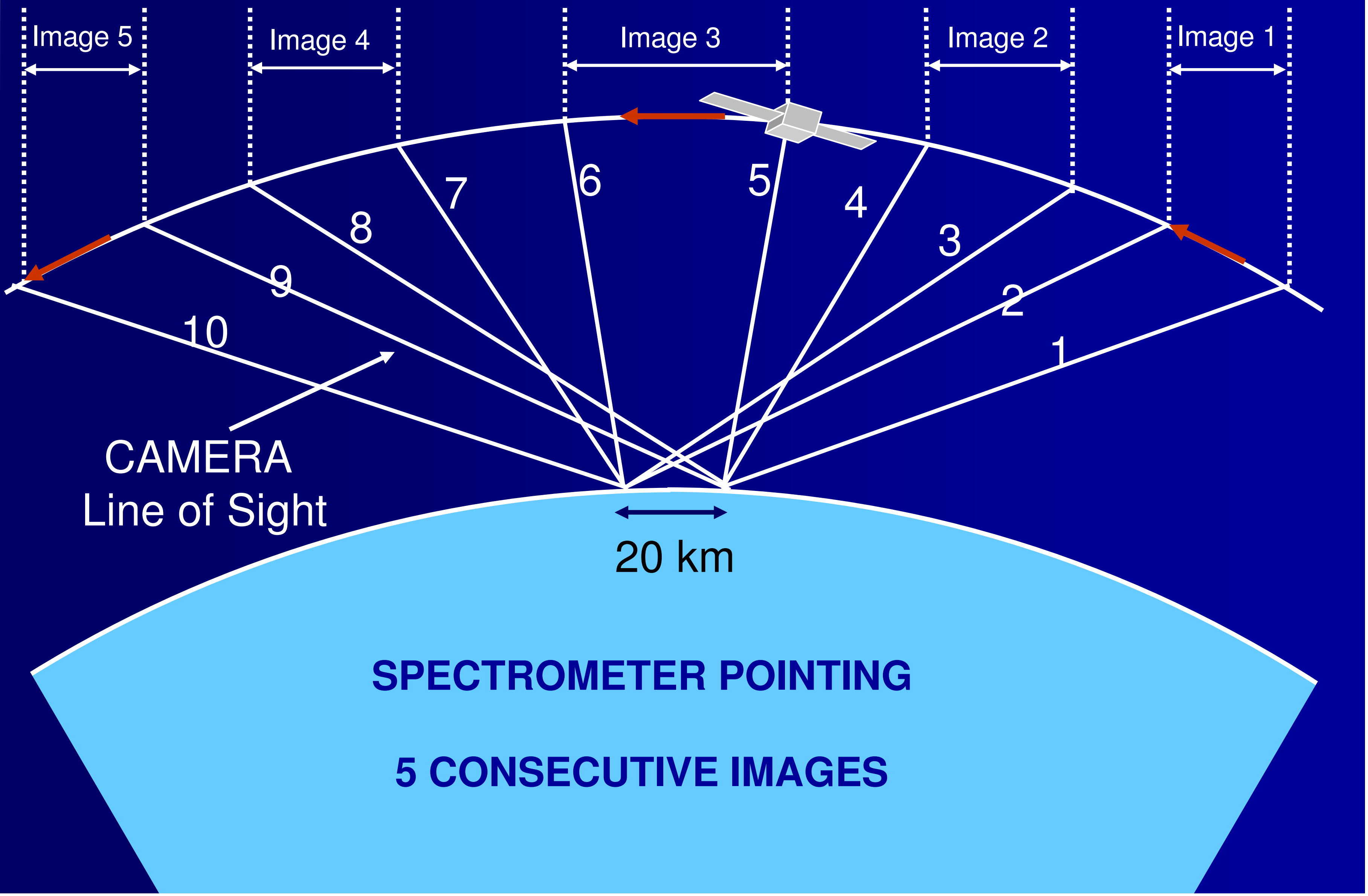
SINGLE-LINE DETECTOR ARRAY
PUSH-BROOM POINTING
(pointing to centre of the Earth)

THE PROBA-1 MISSION



SINGLE-LINE DETECTOR ARRAY
REDUCED-SPEED PUSH-BROOM

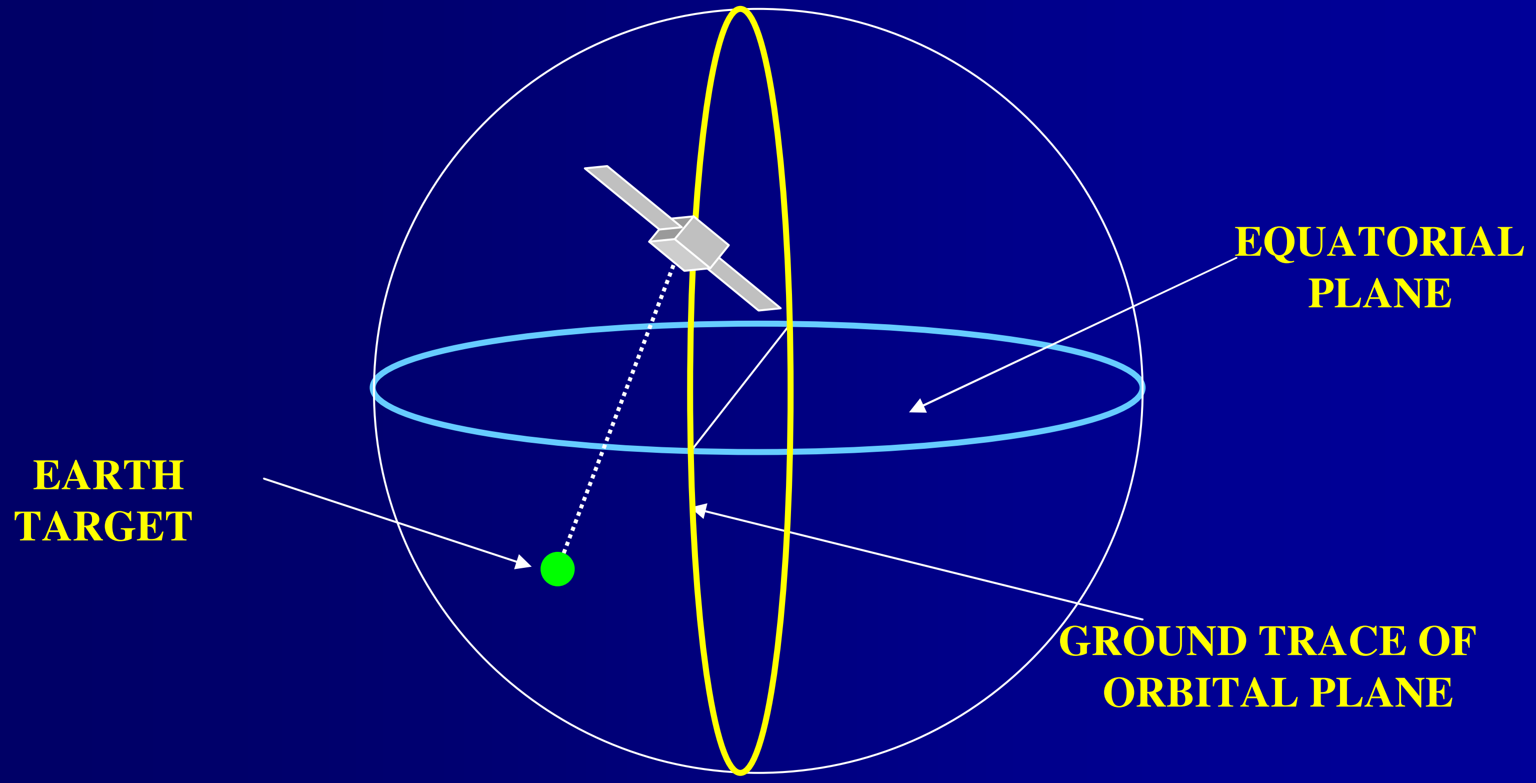
THE PROBA-1 MISSION



SPECTROMETER POINTING

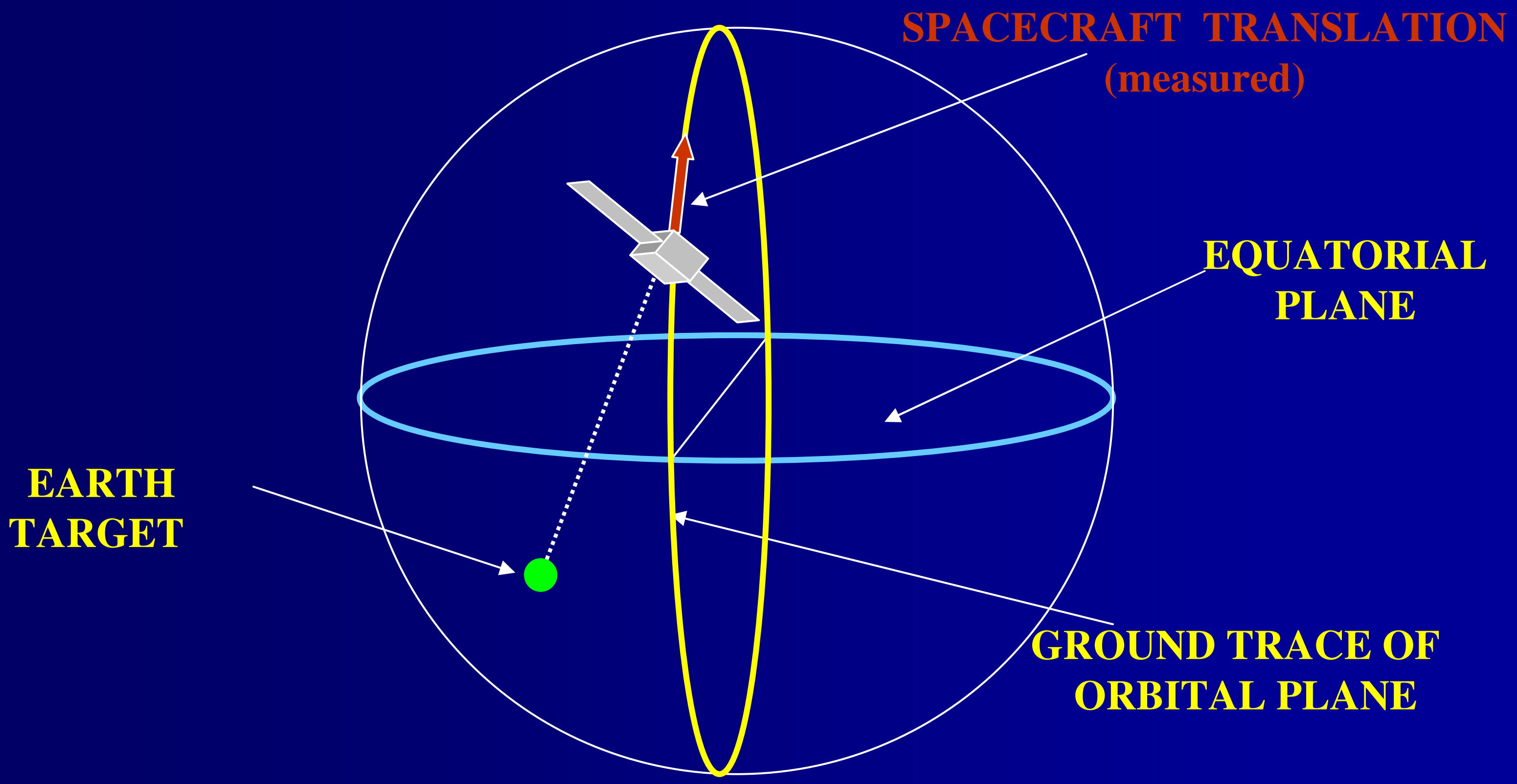
5 CONSECUTIVE IMAGES

THE FOUR MOTIONS TO TAKE INTO ACCOUNT

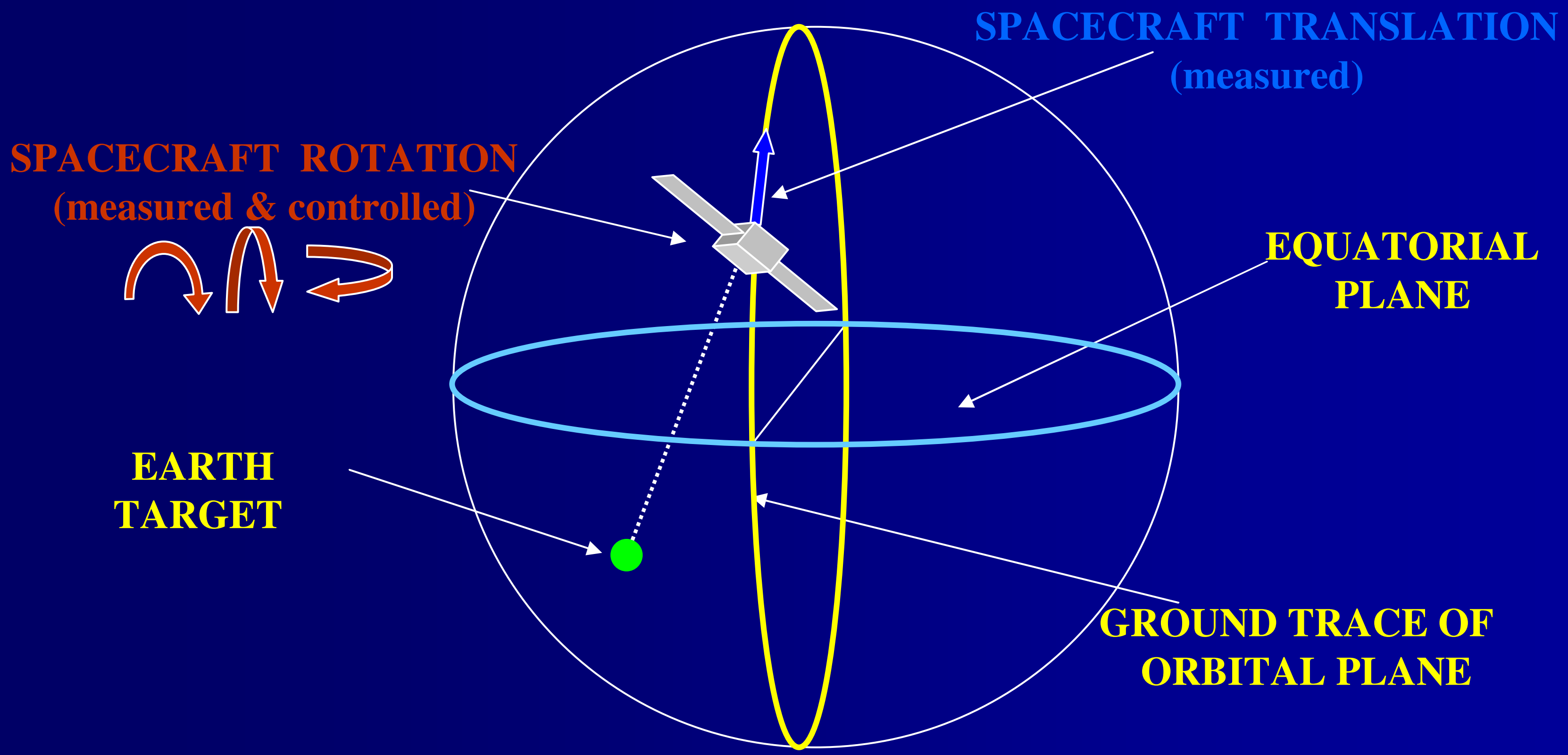


Presentation at RMC, Kingston, 8 October 2002

THE FOUR MOTIONS TO TAKE INTO ACCOUNT

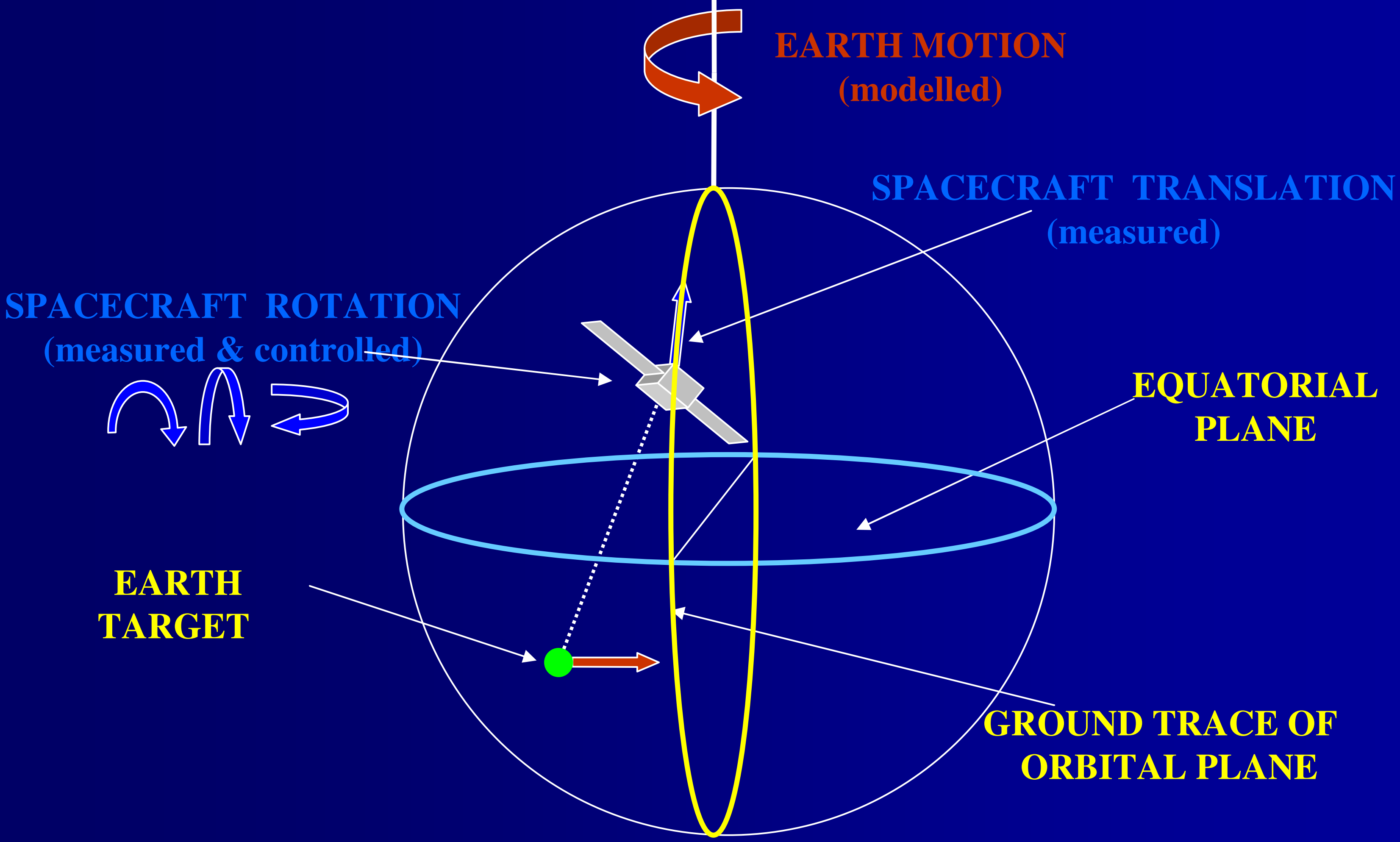


THE FOUR MOTIONS TO TAKE INTO ACCOUNT



Presentation at RMC, Kingston, 8 October 2002

THE FOUR MOTIONS TO TAKE INTO ACCOUNT



EARTH MOTION
(modelled)

SPACECRAFT TRANSLATION
(measured)

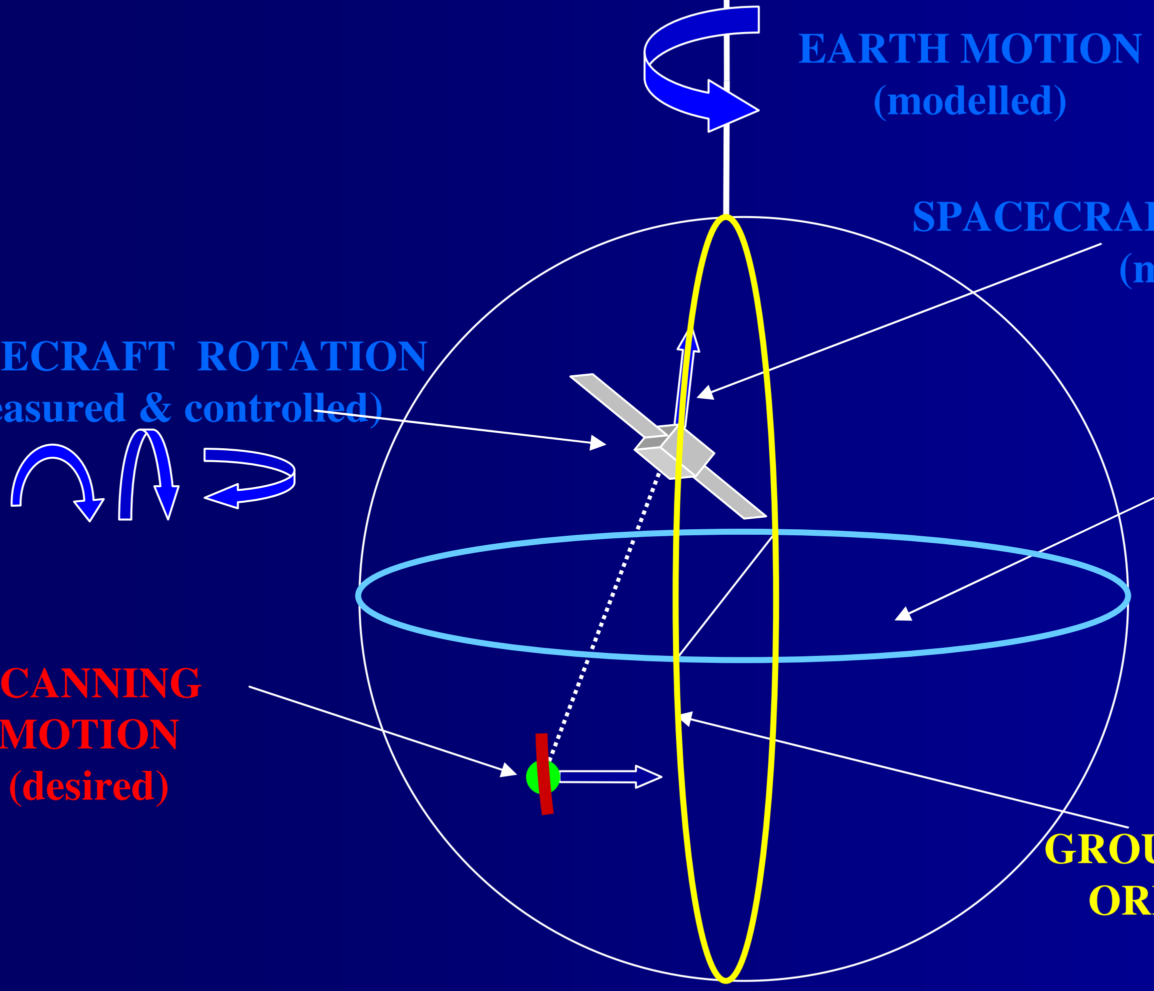
SPACECRAFT ROTATION
(measured & controlled)

EQUATORIAL
PLANE

EARTH
TARGET

GROUND TRACE OF
ORBITAL PLANE

THE FOUR MOTIONS TO TAKE INTO ACCOUNT



EARTH MOTION
(modelled)

SPACECRAFT TRANSLATION
(measured)

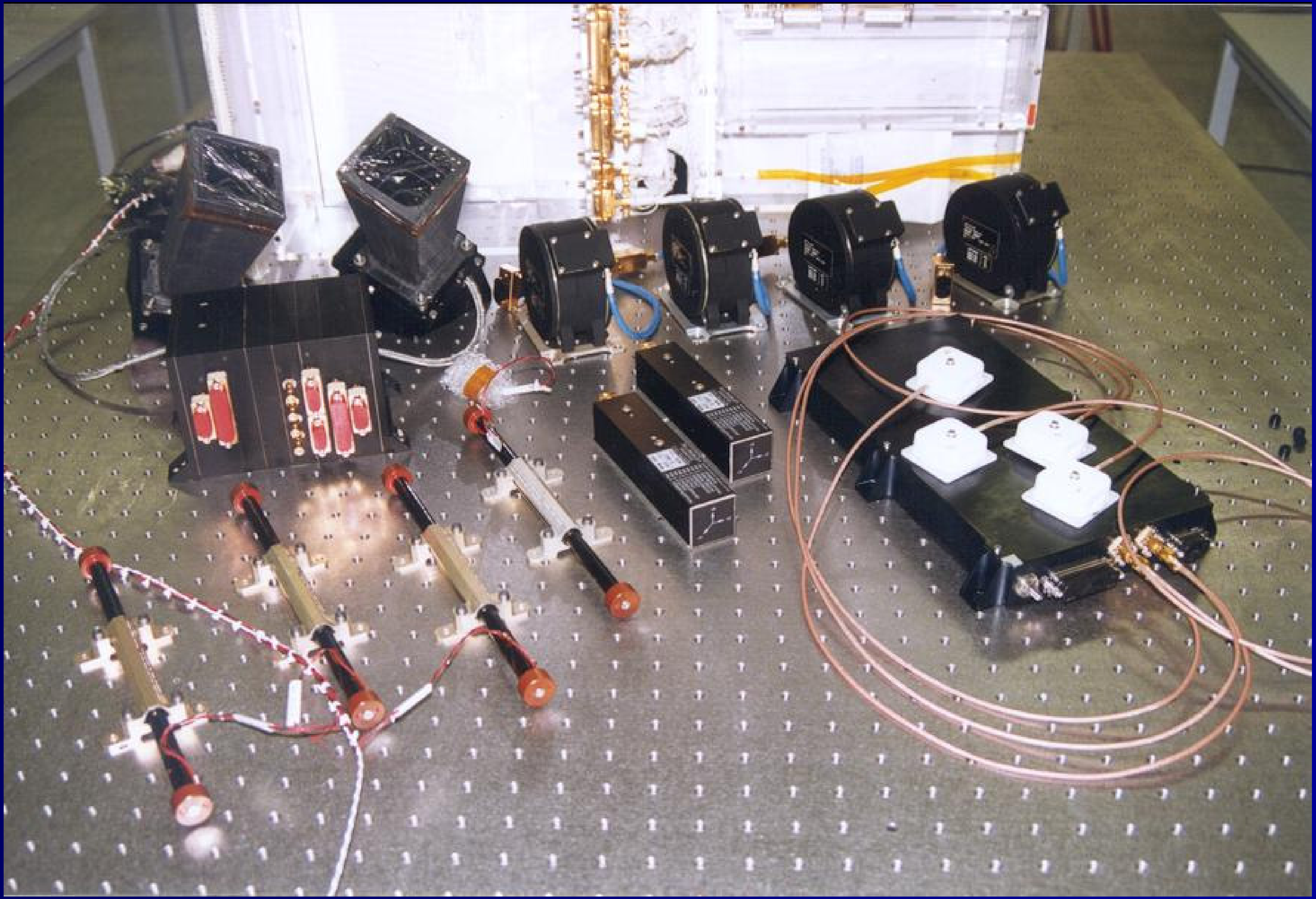
SPACECRAFT ROTATION
(measured & controlled)

EQUATORIAL PLANE

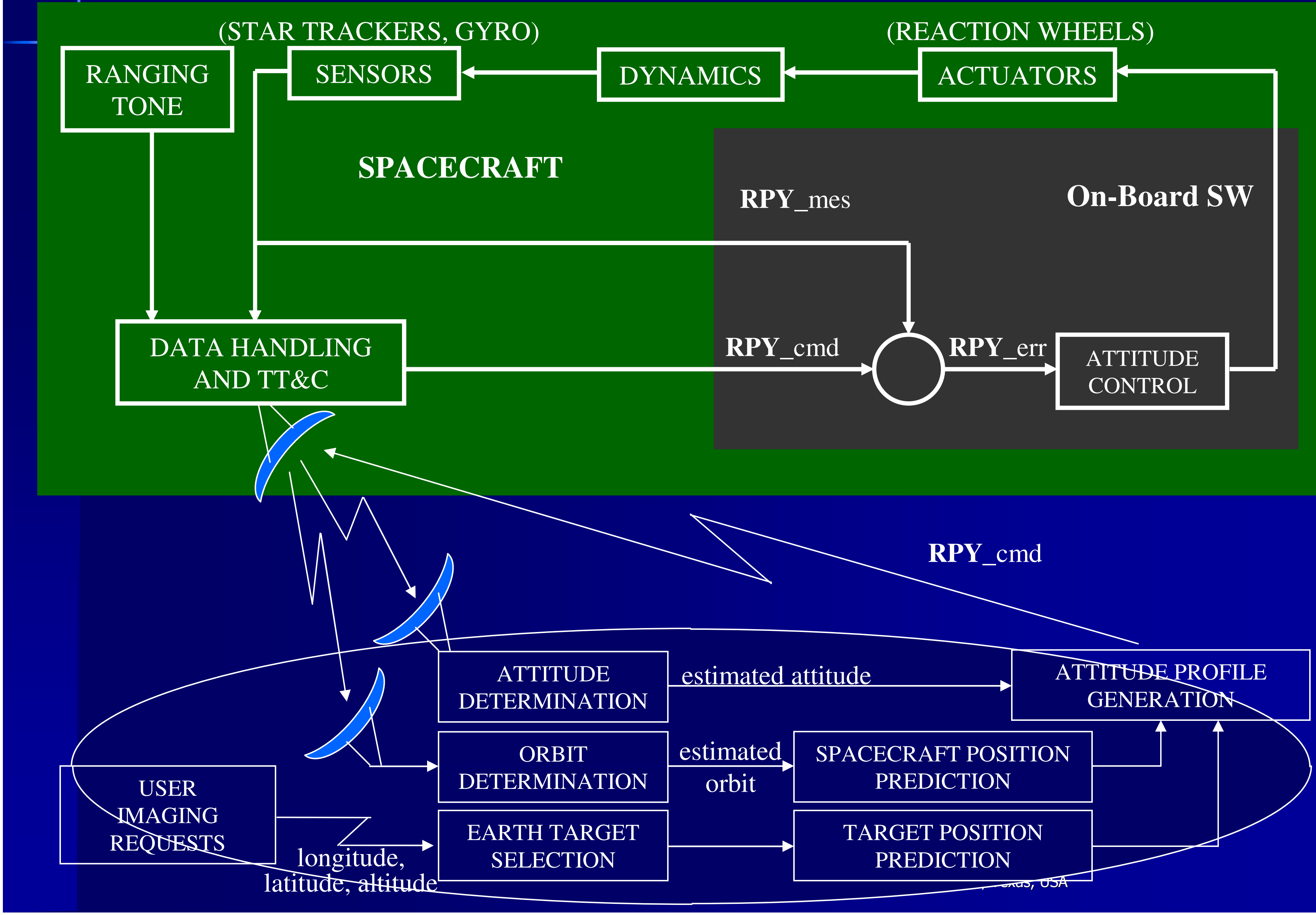
SCANNING MOTION
(desired)

GROUND TRACE OF ORBITAL PLANE

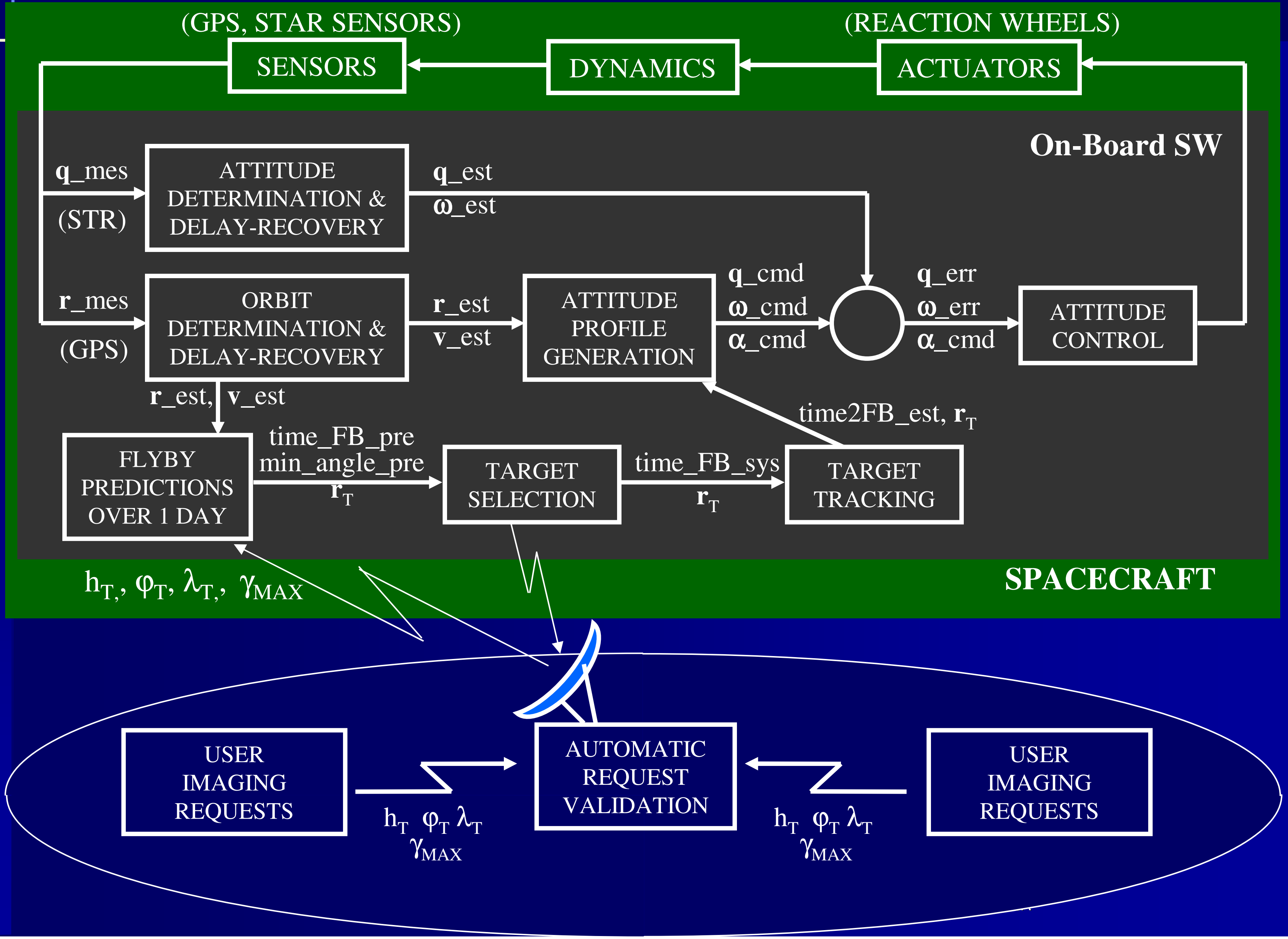
PROBA-1 SENSORS AND ACTUATORS



TYPICAL SPACECRAFT CONTROL



PROBA-1 SPACECRAFT CONTROL



(GPS, STAR SENSORS)

SENSORS

DYNAMICS

(REACTION WHEELS)

ACTUATORS

On-Board SW

q_mes

(STR)

ATTITUDE DETERMINATION & DELAY-RECOVERY

q_est

ω_est

r_mes

(GPS)

ORBIT DETERMINATION & DELAY-RECOVERY

r_est

v_est

ATTITUDE PROFILE GENERATION

q_cmd

ω_cmd

α_cmd

q_err

ω_err

α_cmd

ATTITUDE CONTROL

r_est, **v**_est

FLYBY PREDICTIONS OVER 1 DAY

time_FB_pre

min_angle_pre

$\mathbf{r}_T$

TARGET SELECTION

time_FB_sys

$\mathbf{r}_T$

TARGET TRACKING

time2FB_est, $\mathbf{r}_T$

$h_{T,}$, φ_T, $\lambda_{T,}$, γ_{MAX}

SPACECRAFT

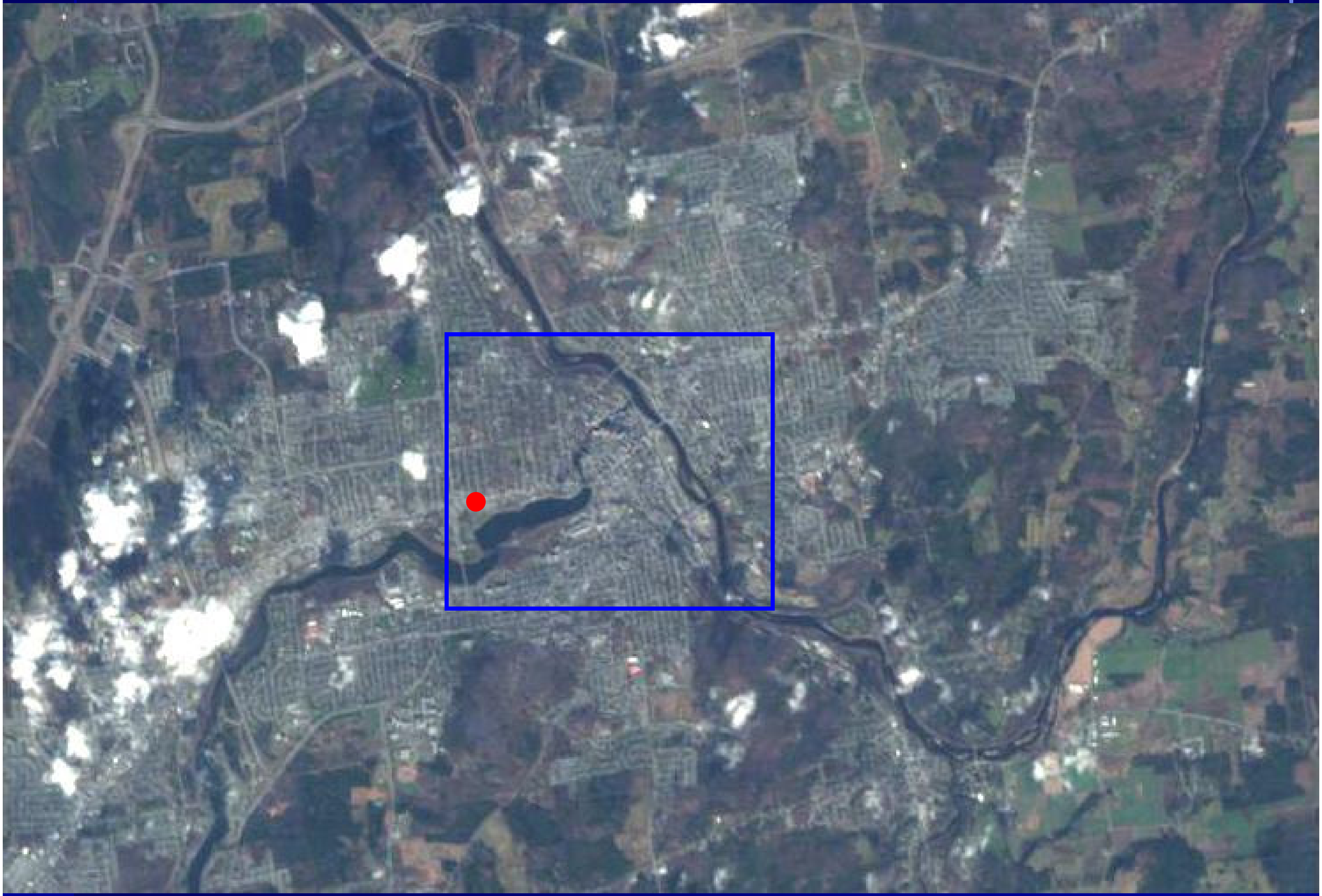
USER IMAGING REQUESTS

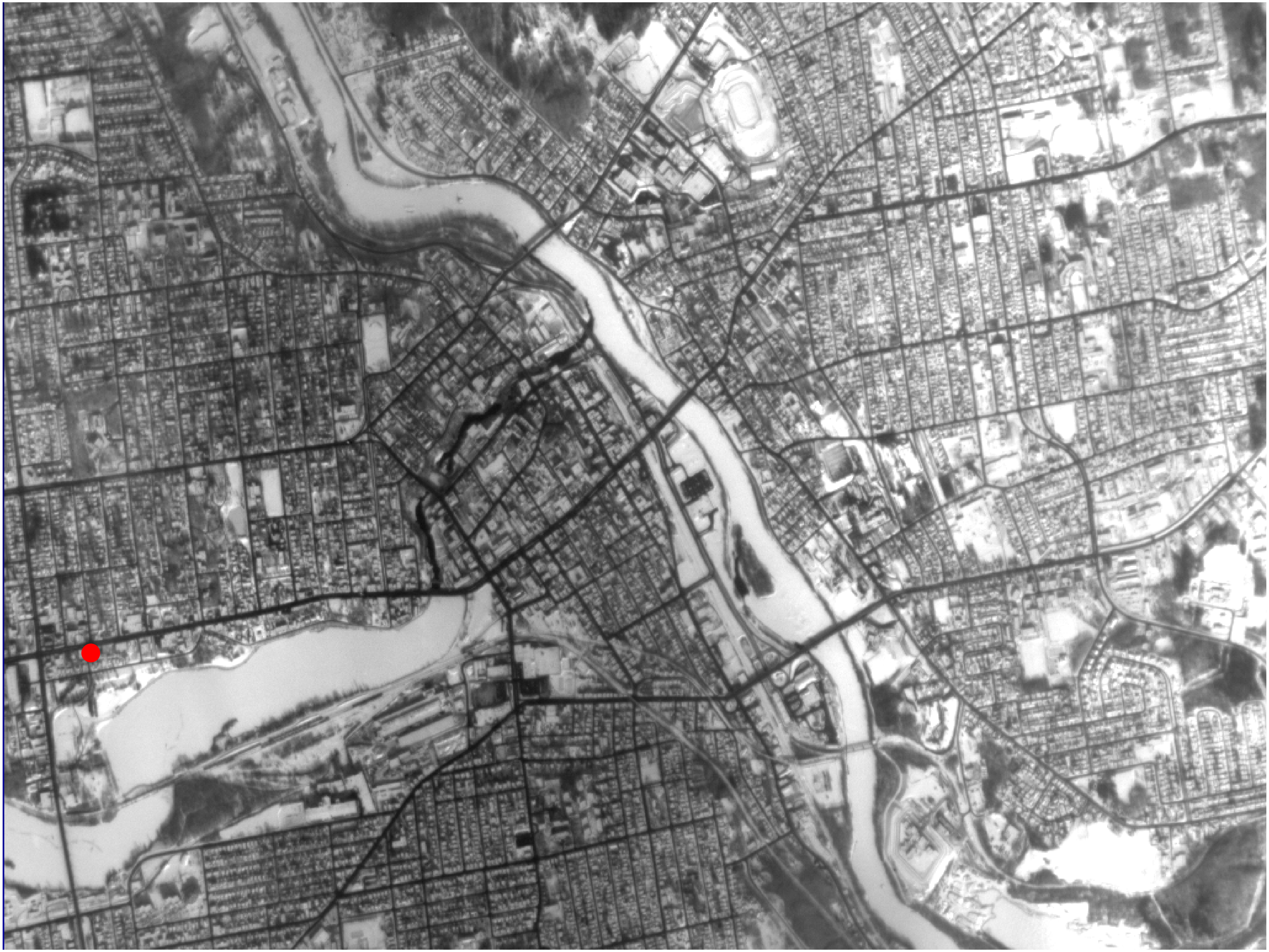
h_T φ_T λ_T γ_{MAX}

AUTOMATIC REQUEST VALIDATION

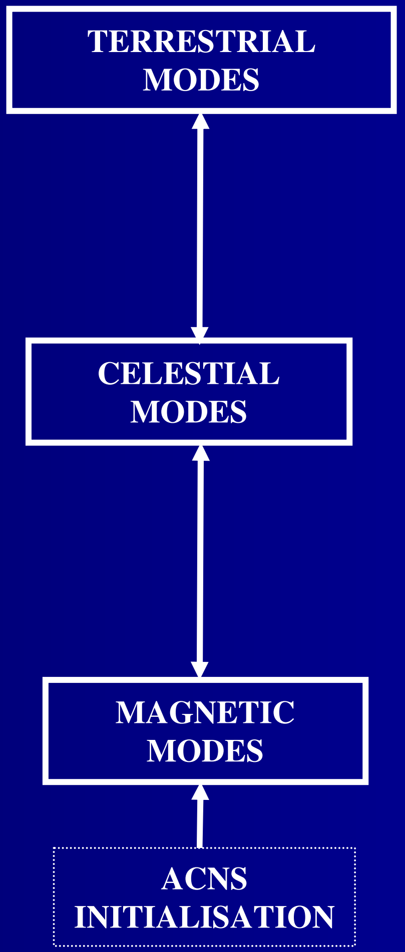
h_T φ_T λ_T γ_{MAX}

USER IMAGING REQUESTS





PROBA-1 MODES OF OPERATION



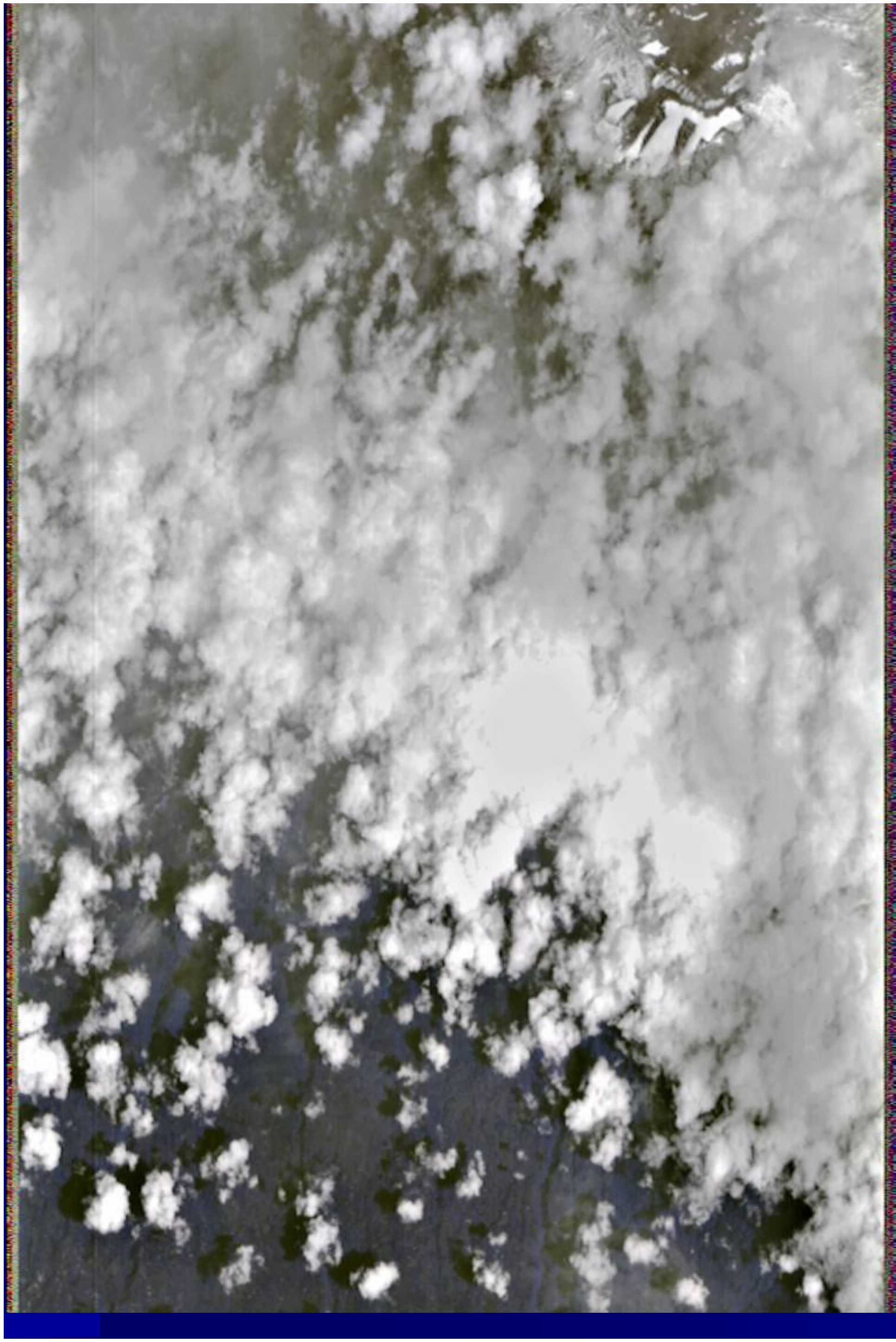
HARDWARE	SOFTWARE		SUB-MODES
GPS RECEIVER +	NAV_ORB NAV_EARTH NAV_TARGET NAV_EVENTS GDC_IMAGING GDC_RELORB +	**TERRESTRIAL MODES**	CHRIS IMAGING SCANS FIXED-EARTH POINTING NADIR POINTING
STAR SENSORS REACTION WHEELS +	NAV_ATT NAV_SUN NAV_ECLIPSE GDC_ATTERR GDC_SUN GDC_INERTIAL CTL_ATT CTL_ANGMOM +	**CELESTIAL MODES**	SUN POINTING COARSE POINTING FINE POINTING
MAGNETOMETER MAGNETIC TORQUER	NAV_MAG CTL_MAG	**MAGNETIC MODES**	DETUMBLING MAGNETIC ACQUISITION ATTITUDE HOLD
S /C SEPARATION →		**ACNS INITIALISATION**	

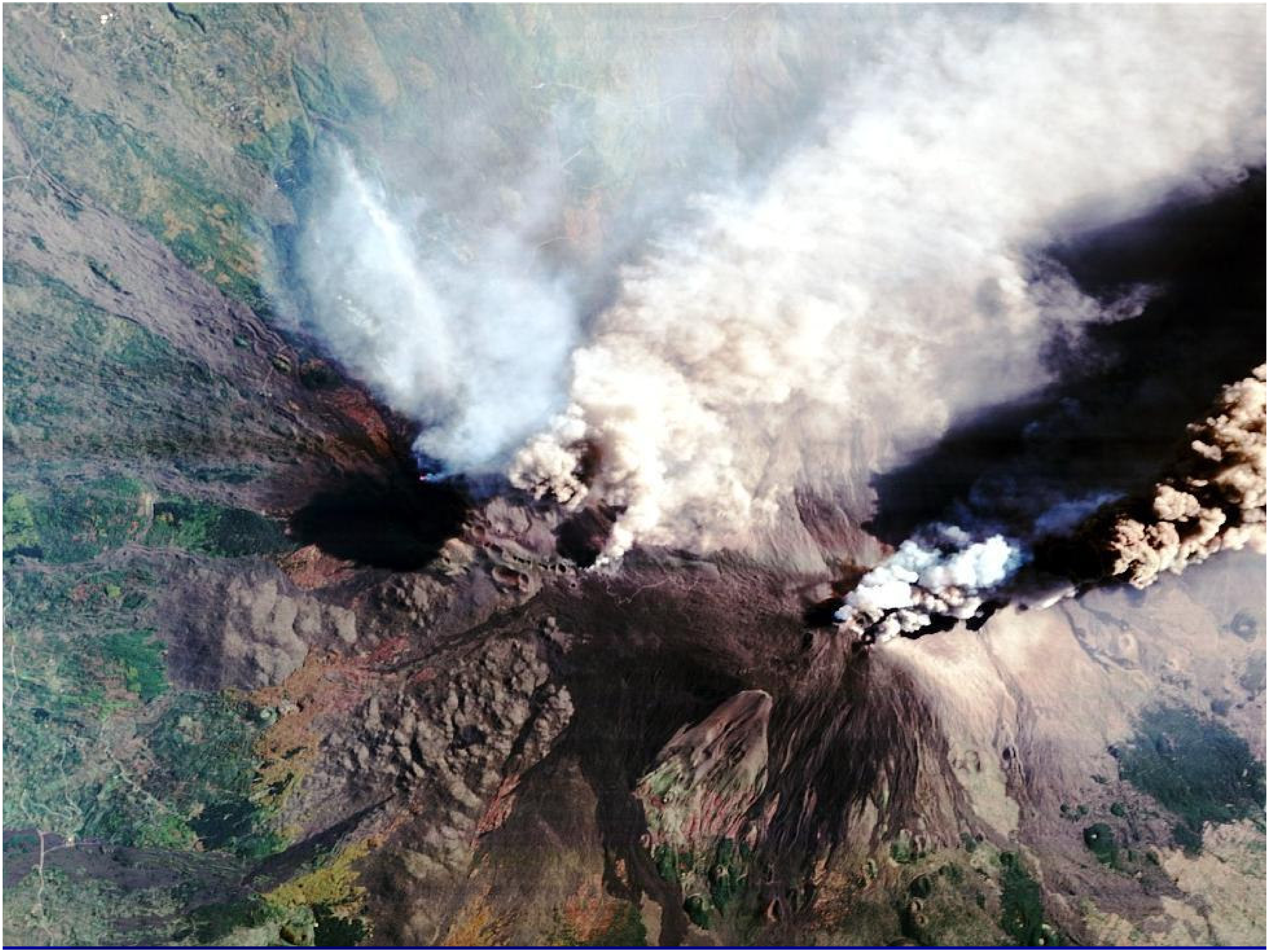
PROBA OPERATIONAL MODES

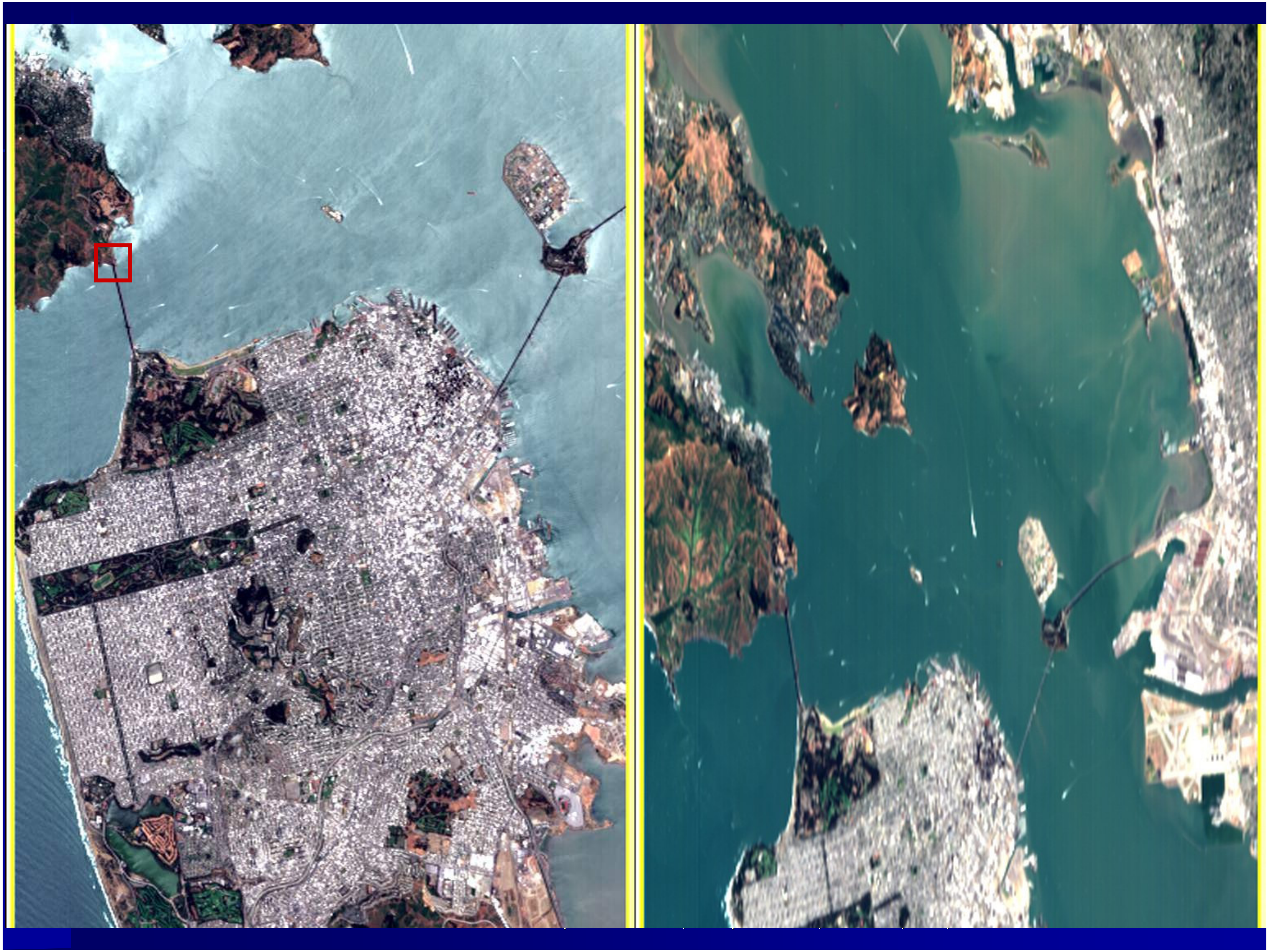
NGC

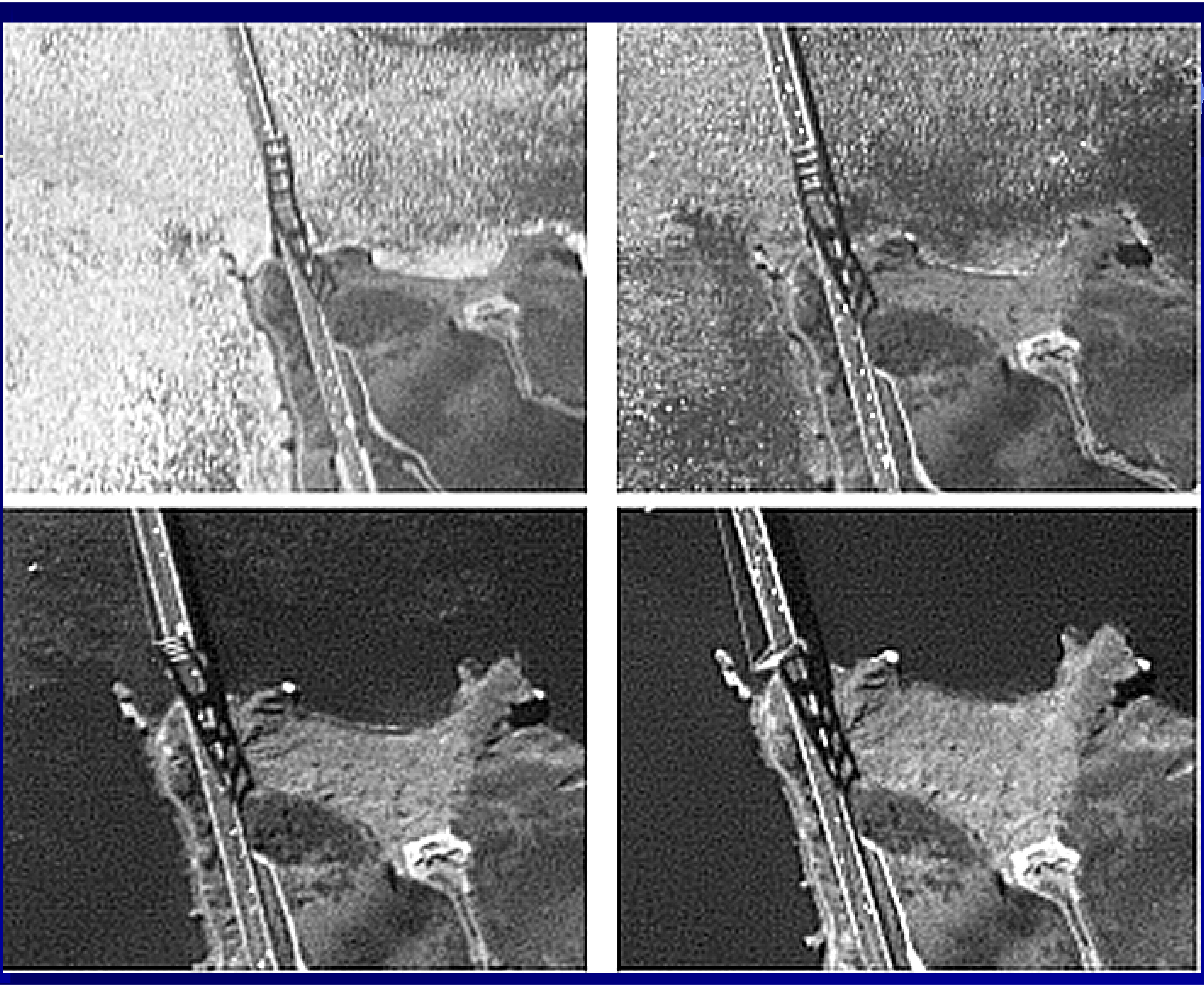
SOME PROBA-1 FLIGHT RESULTS

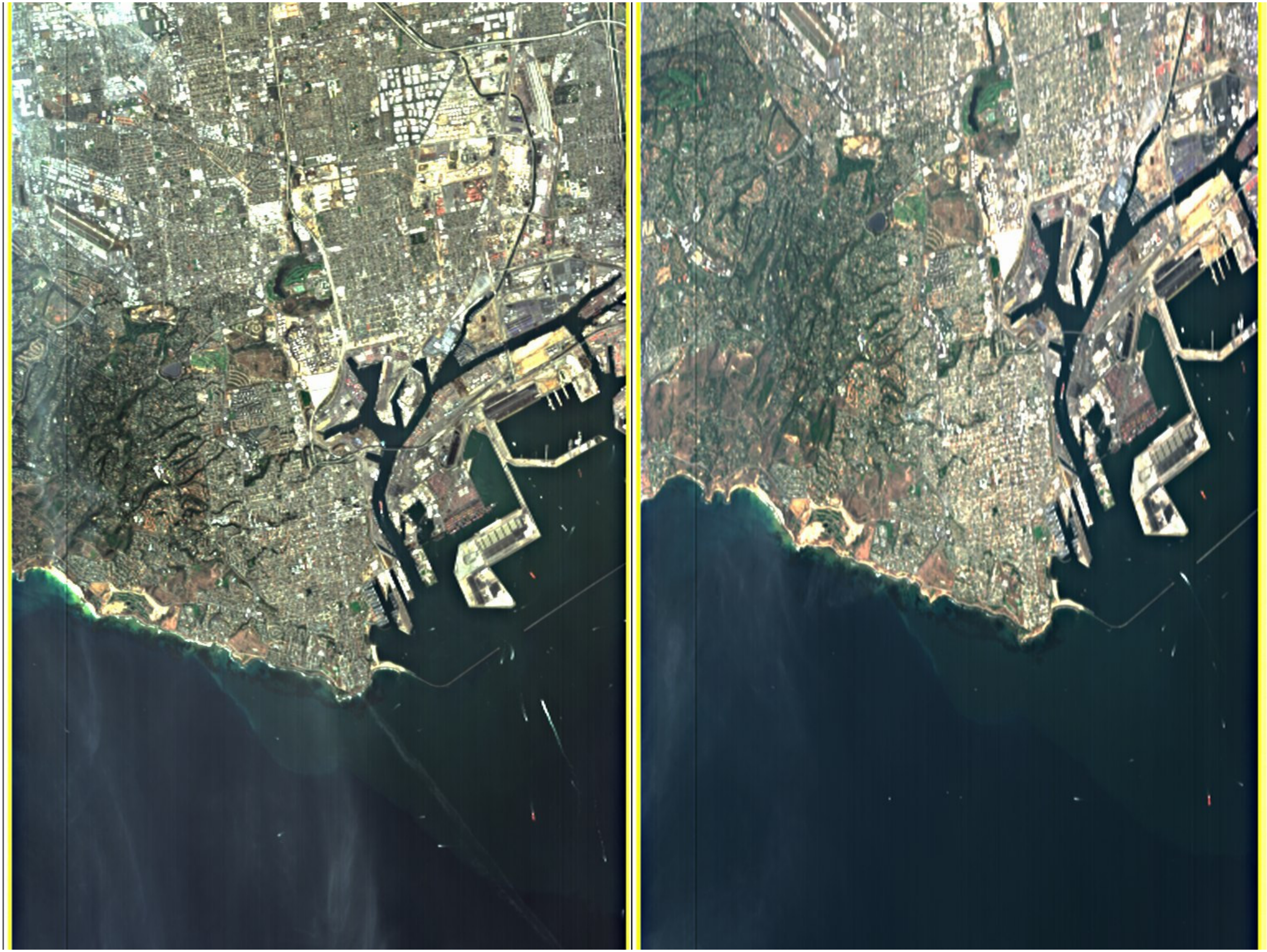
LAUNCH 22 OCTOBER 2001

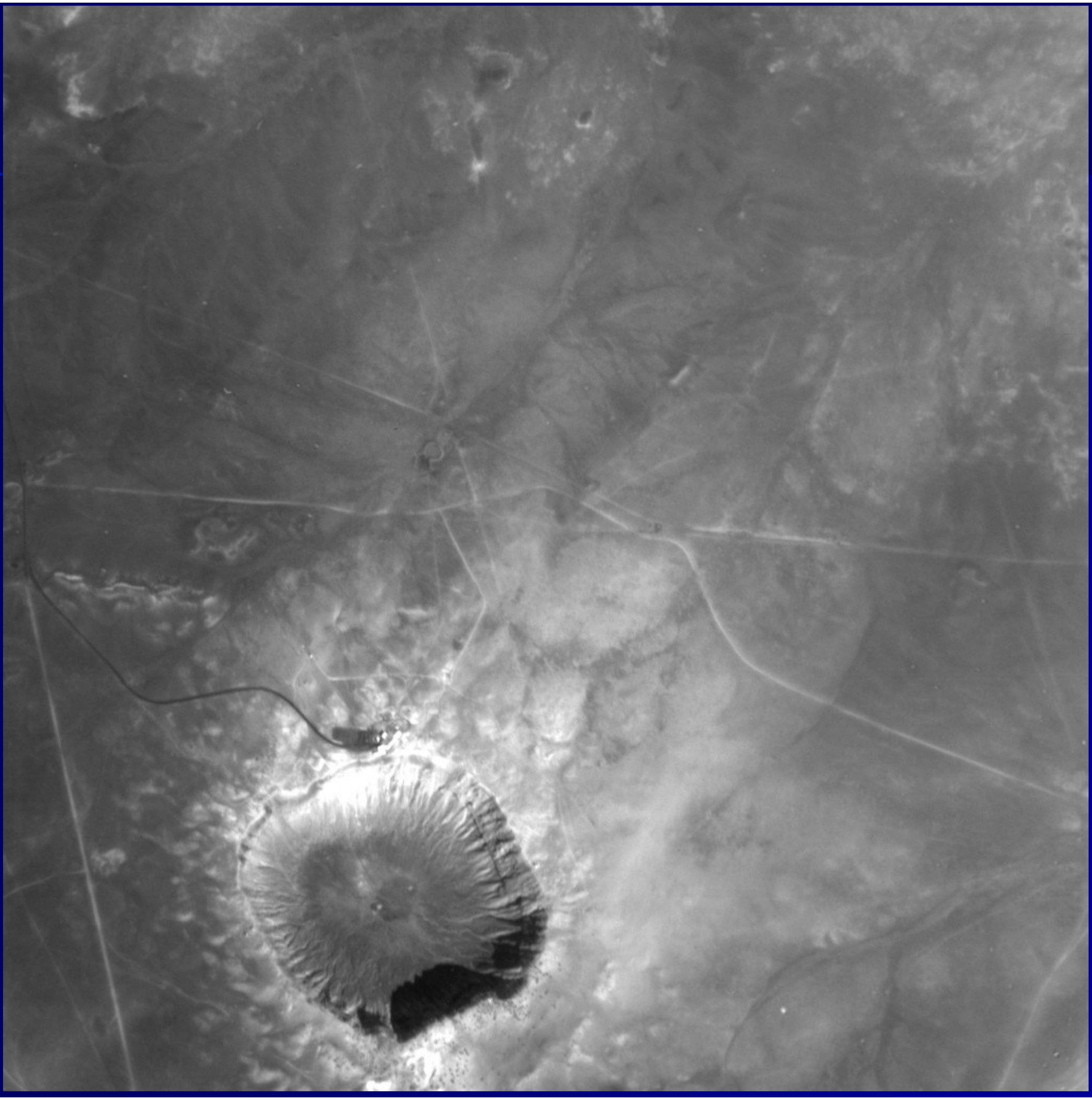




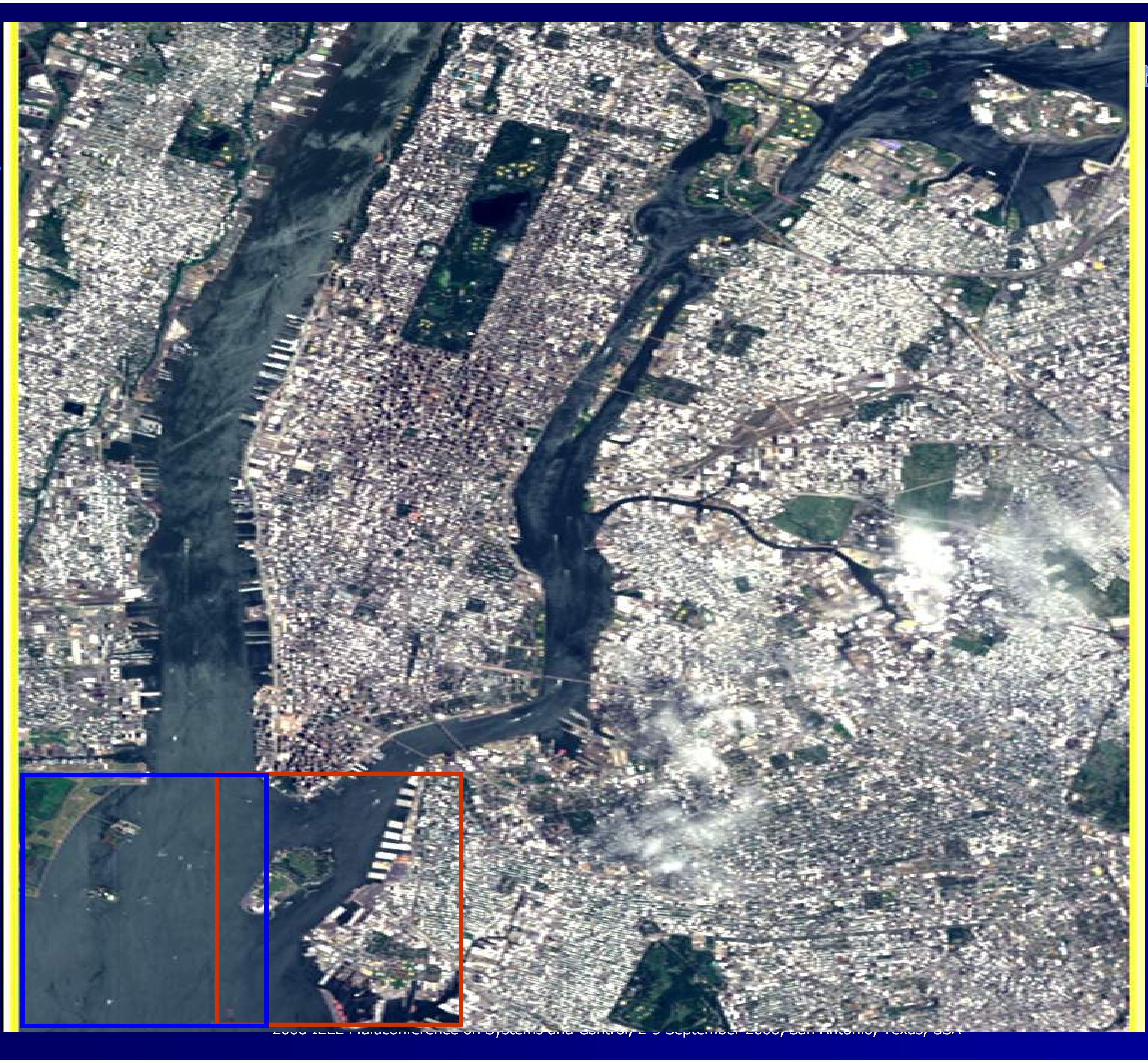






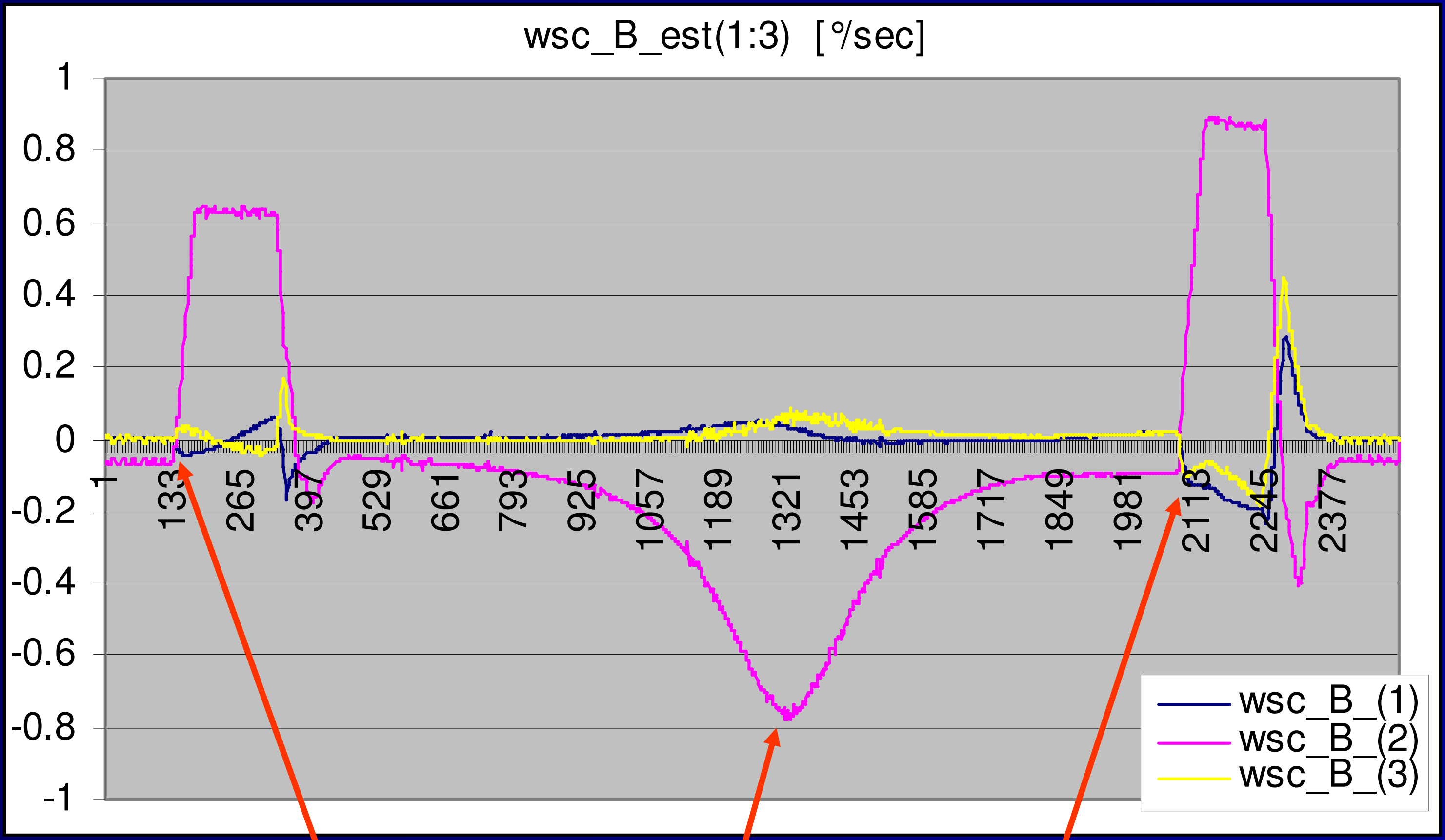


NGC



B&W CCD CAMERA OPERATION

Spacecraft angular velocity [°/sec]



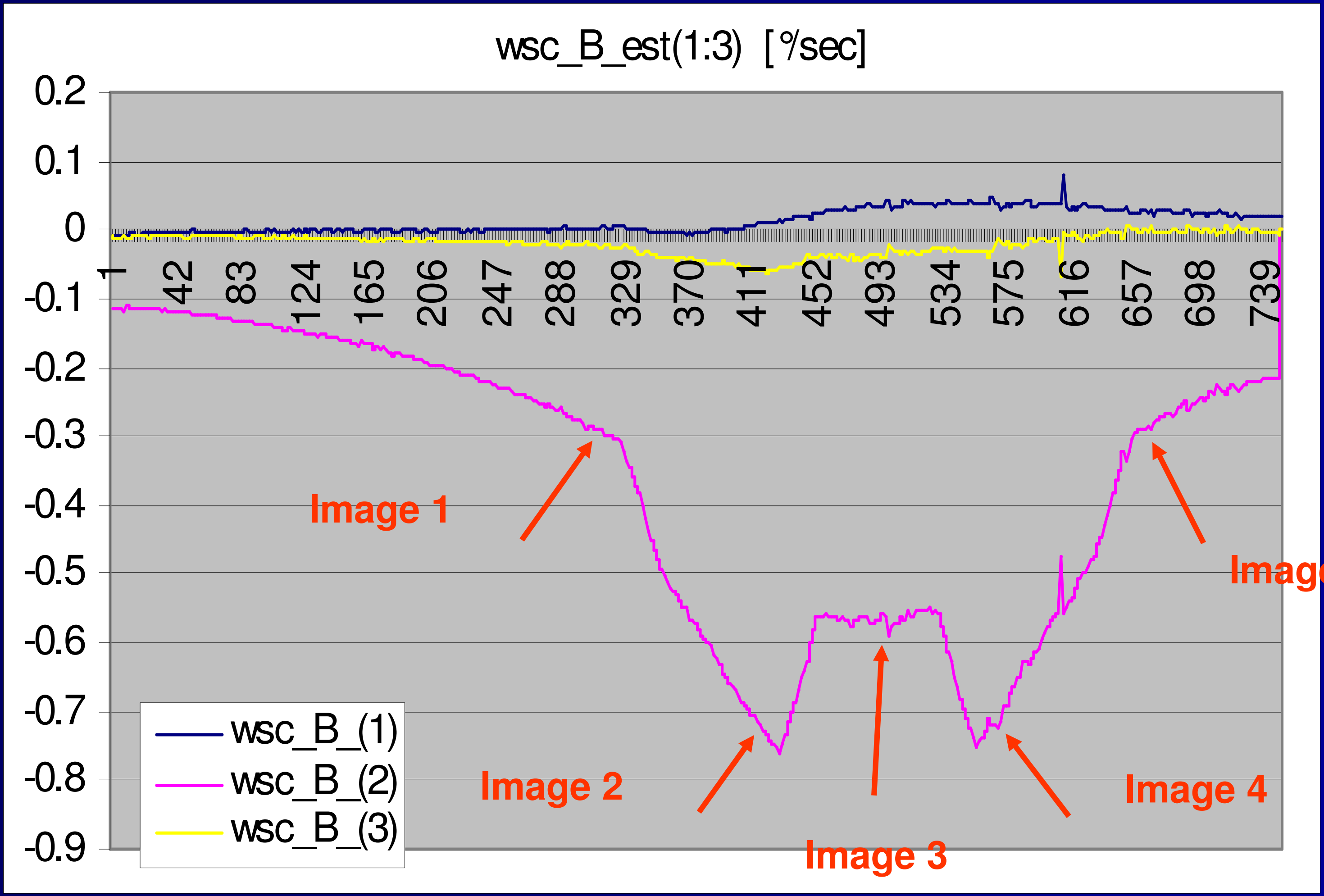
Fixed target pointing commanded

Target Flyby

Nadir pointing commanded

5-IMAGE SPECTROMETER OPERATION

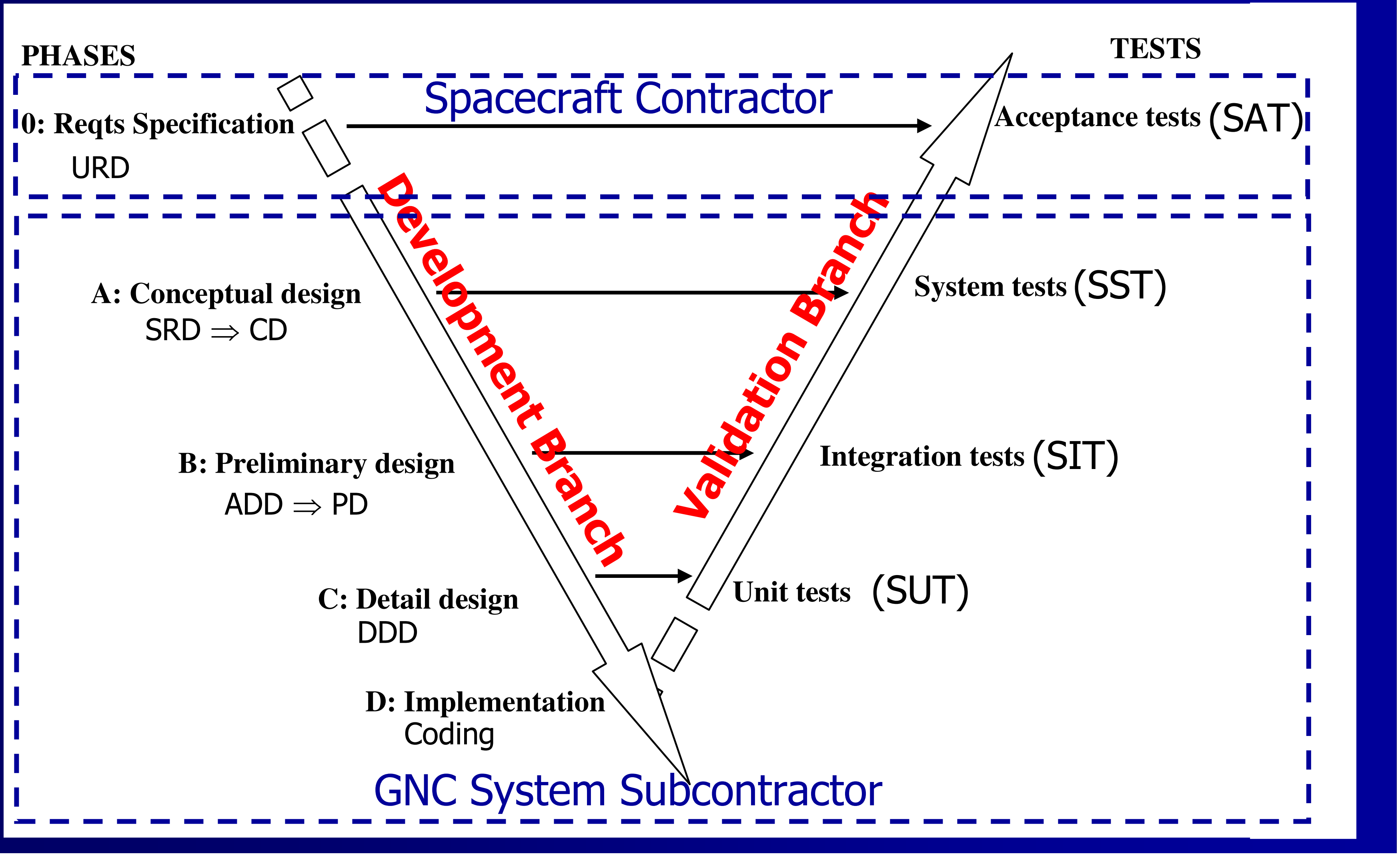
Spacecraft angular velocity [°/sec]



THE PROCESS

Software Development-1

- The typical V-shape software development/validation process



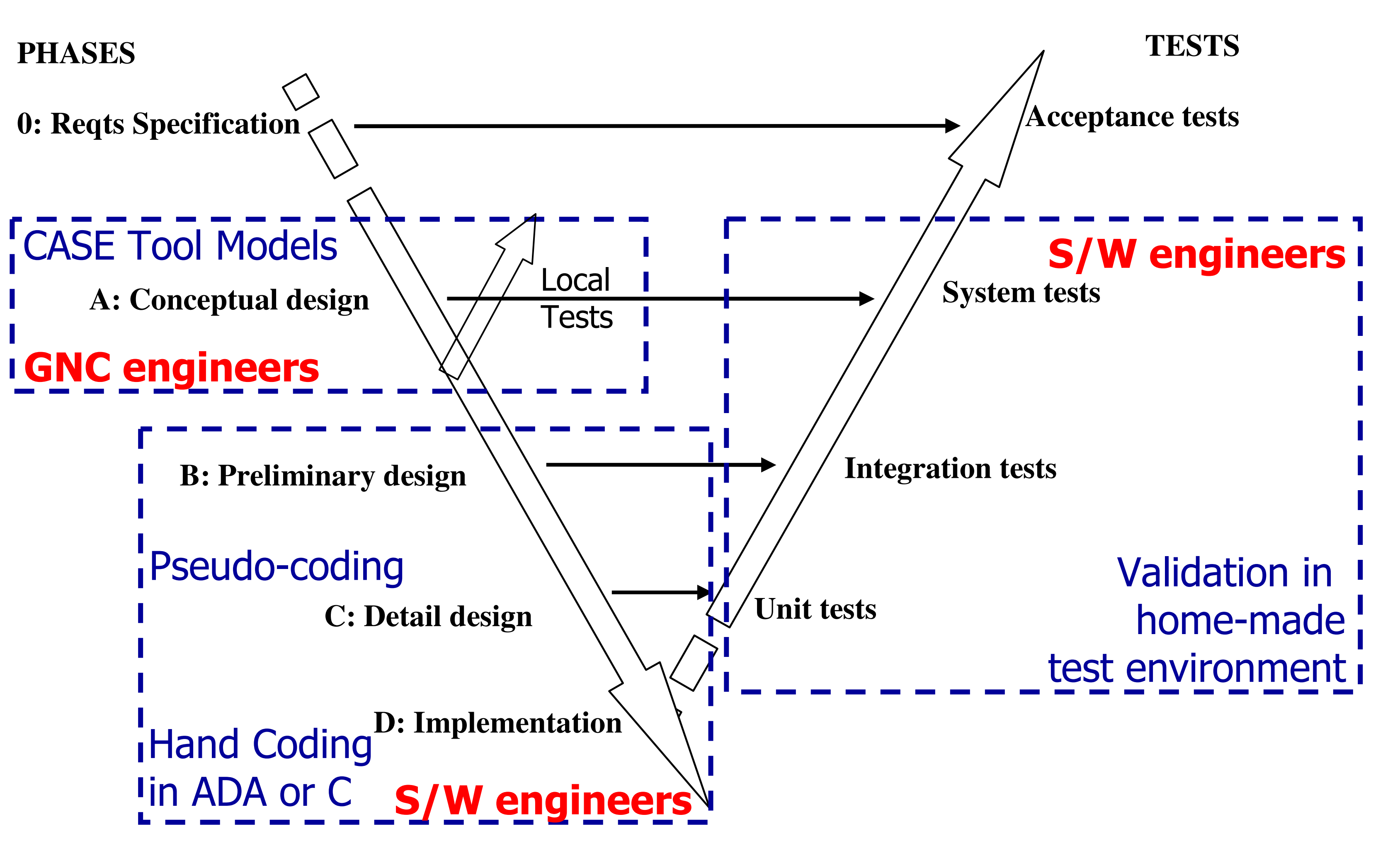
Software Development-4

- More Definitions
  - Algorithms:
    - mathematical description of a software function
    - at conceptual design level
  - Pseudo-Code
    - mathematical description of a software module and flow logic
    - at preliminary and detailed design levels
  - Models (in the context of CASE tools):
    - block-diagram description of algorithms and pseudo-code
    - e.g. Simulink™ models, SystemBuild™ models
  - Code:
    - description of algorithms and pseudo-code in high-level, readable, computer language (ADA, C, C++)

Software Development-6

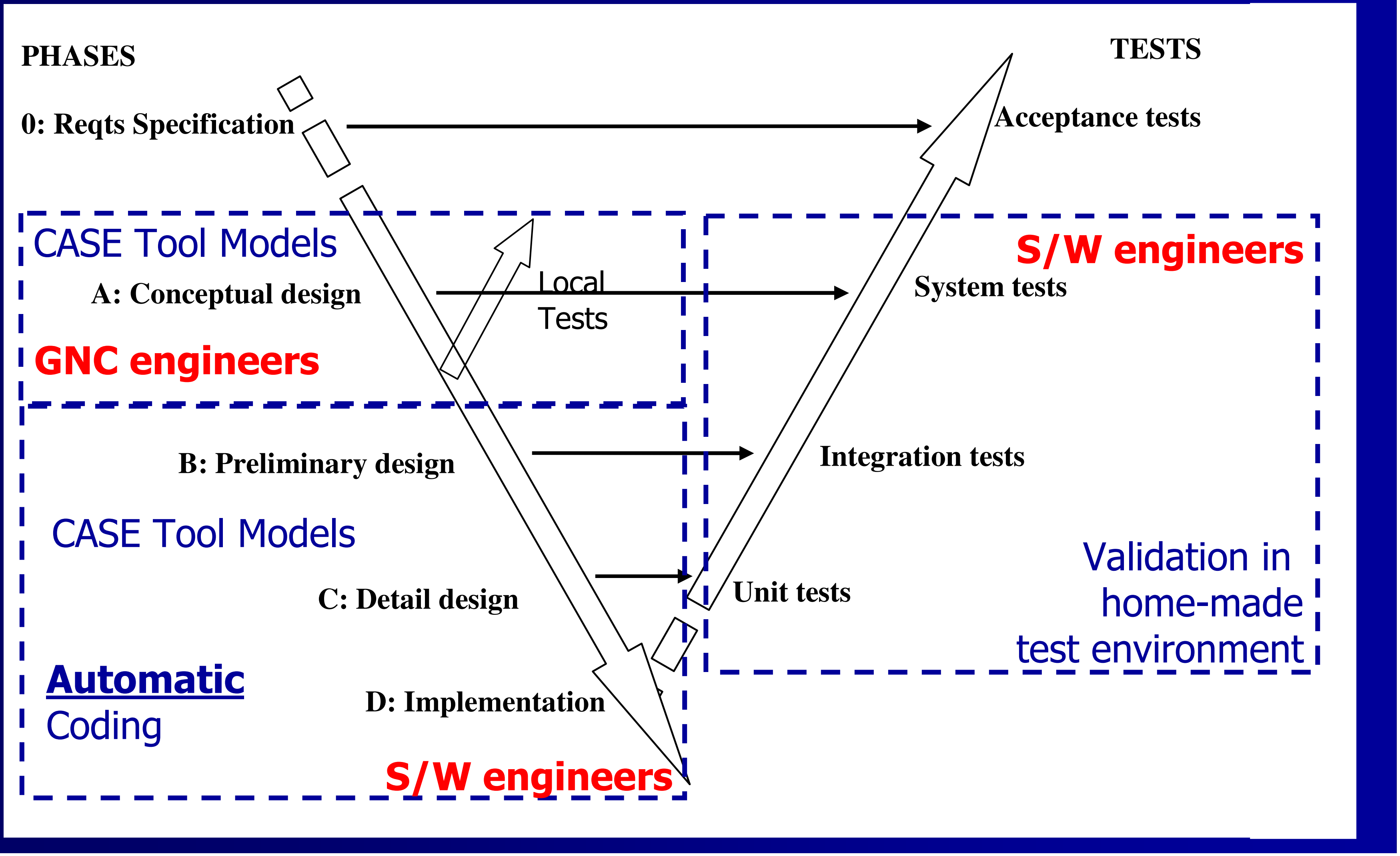
NGC

- Typical 2nd-generation S/W development/validation process



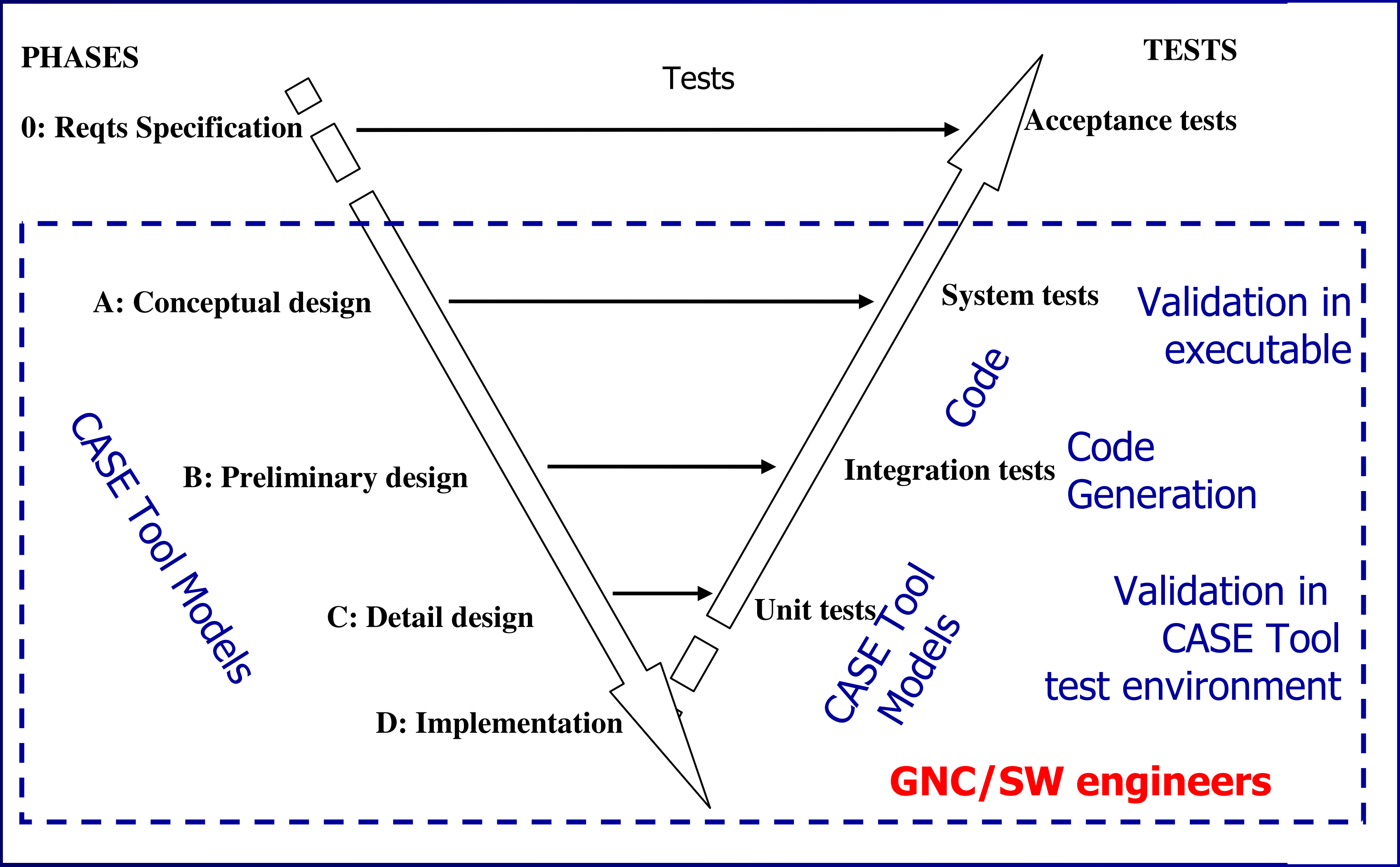
Software Development-8

- Typical 3rd-generation S/W development/validation process



Software Development-9

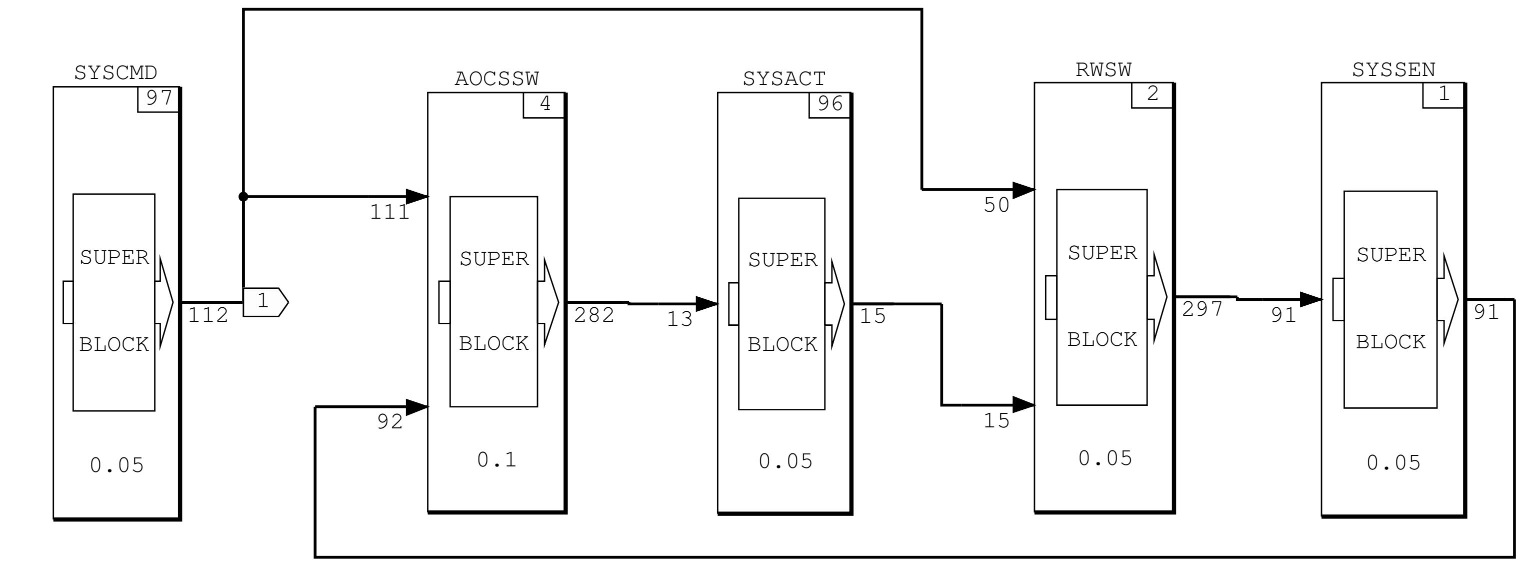
- Typical 4th-generation S/W development/validation process



PROBA-2 S/W DEVELOPMENT

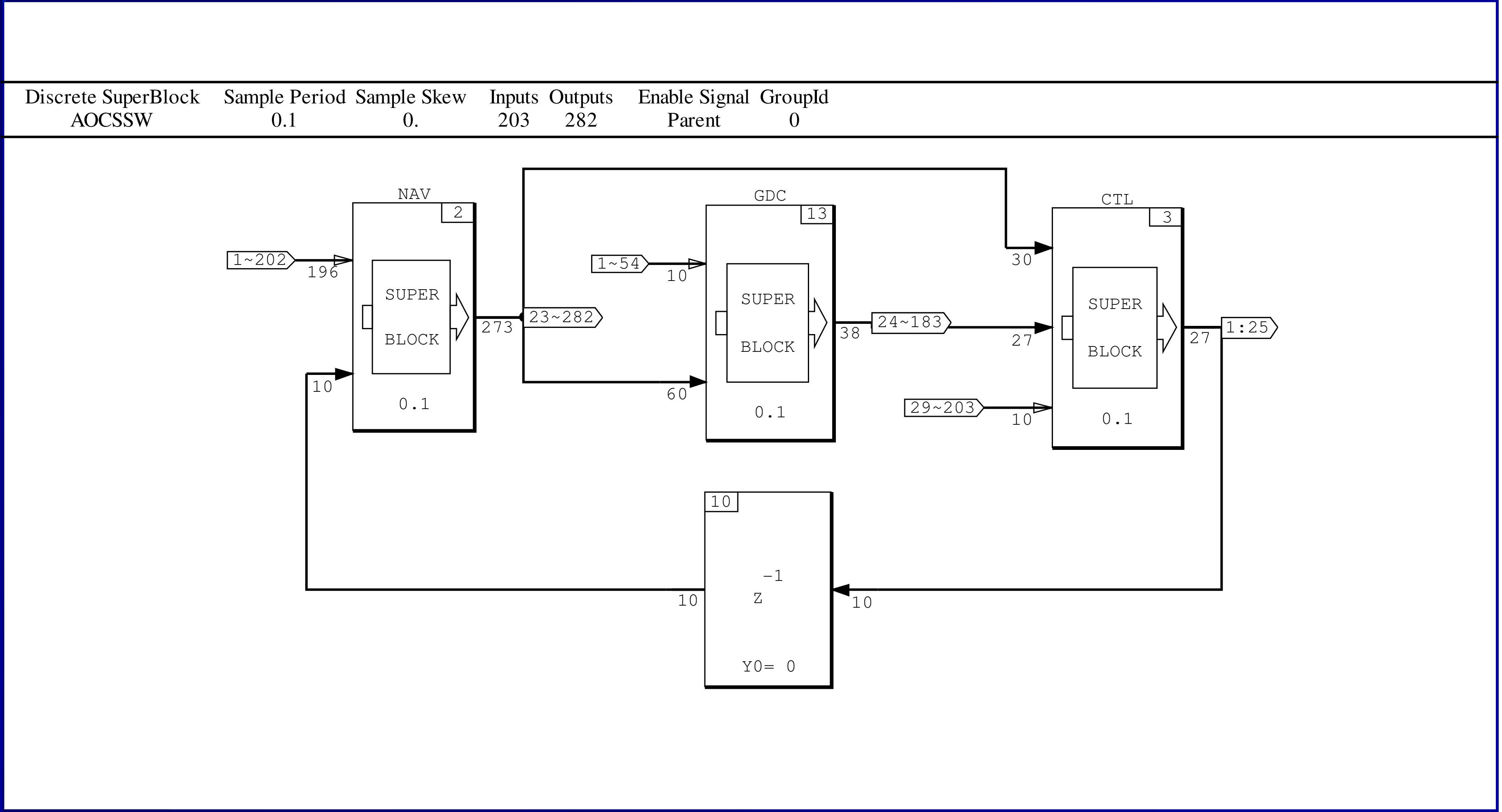
✧ PROBA-2 TOP LEVEL ARCHITECTURE

Discrete SuperBlock	Sample Period	Sample Skew	Inputs	Outputs	Enable Signal	GroupId
P2shell	0.05	0.	0	1	Parent	1



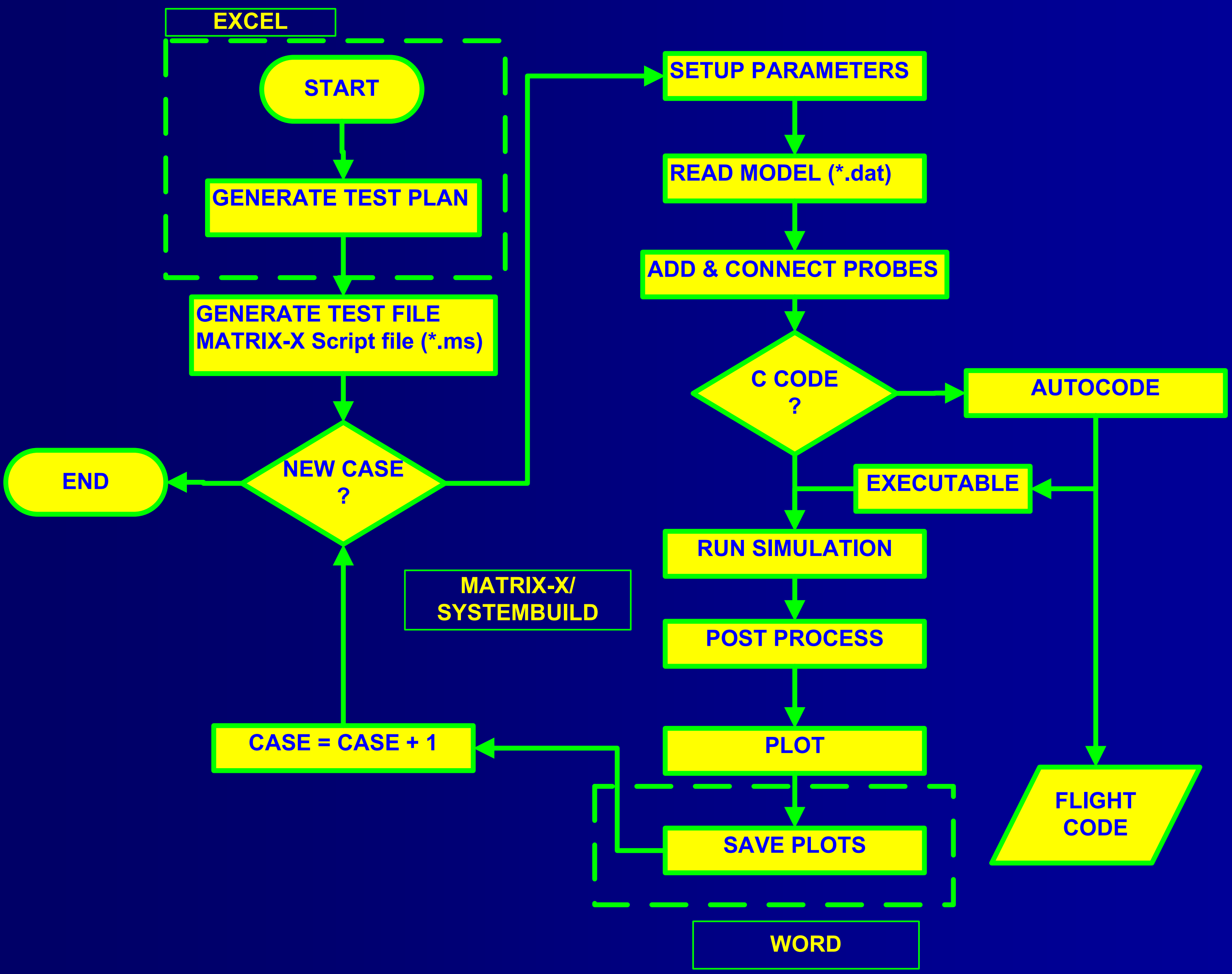
PROBA-2 S/W DEVELOPMENT

✧ PROBA-2 GNC MODULES



Discrete SuperBlock	Sample Period	Sample Skew	Inputs	Outputs	Enable Signal	GroupId
AOCSSW	0.1	0.	203	282	Parent	0

PROBA-2 S/W VALIDATION



THE LESSONS AND THE BENEFITS

PROBA-1 STATISTICS

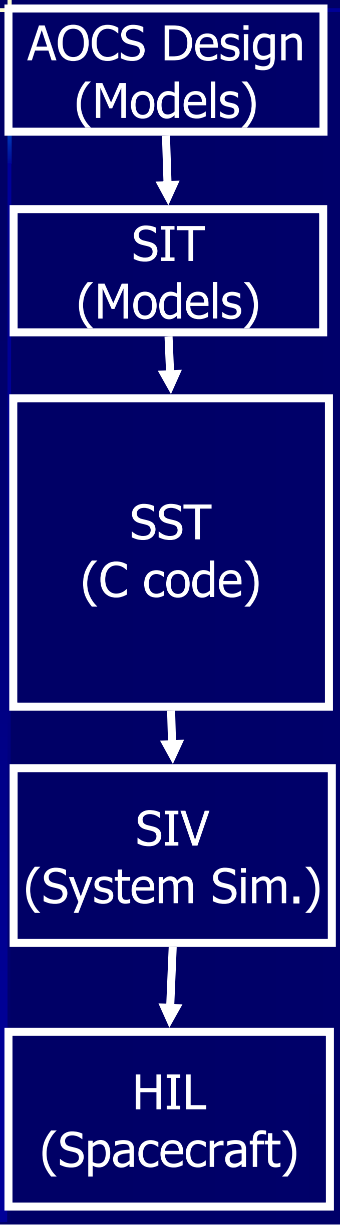
✧ **MatrixX models :**

- 1401 instances of 355 superblocks, 548 parameters in total
- The onboard GNC module has 128 inputs, 983 outputs
- The environment module has 33 inputs, 190 outputs

✧ **AutoCode generated software :**

- The onboard GNC module has 57217/27181 lines, 1016 global variables and 249 functions.
- The environment module has 18220/9563 lines, 734 global variables and 86 functions.
- The code is very readable.
- Traditional coding and validation **alone** would have taken **15 persons-years** (ESA estimation)
- With AutoCoding, PROBA spent **<9 persons-years** including requirements phase, algorithms definition and design, architecture specification, code production and validation.

PROBA-2 STATISTICS



AOCS Design (Models)

SIT (Models)

SST (C code)

SIV (System Sim.)

HIL (Spacecraft)

- Matrix/SystemBuild Environment
- Completed by NGC in 2006

- Software Integration Tests (SIT)
- Completed by NGC in 2006

- Software System-level Tests (SST)
- Completed by NGC in 2007
- Automatic generation of C-code
- Automatic compilation and building of executable
- Automatic generation of test reports

3 NGC Engineers at NGC (<5 PY)

- Software Independent Validation (SIV)
- Completed by Verhaert & NGC in 2007-08
- Perform Software Acceptance Tests (SAT)

- Hardware-In-the-Loop (HIL) Tests
- On-going at Verhaert
- Perform HW-SW integration tests & SAT

1 NGC Eng
1 VE Eng
1 ESA Eng
at S/C Contractor
(<1 PY)

LESSONS LEARNED

- Reduction in the number of documents:
  - automatic generation of document
  - models act as Architectural and Detail Design Documents
- Reduction of human interface from models to on-board code
  - reduction in verification process
  - minimisation of human errors, discrepancies, etc.
- Better visibility/understanding/organisation of the algorithms
  - easier to find sources of bugs
  - easier to add/delete modules
  - non-expert can easily understand
- Simpler/faster transfer of knowledge
  - easy and quick transfer of knowledge to software engineer
  - easy to add new engineers to the project

LESSONS LEARNED

NGC

+ Automated generation of test results
  - Hundreds of cases can be automatically generated over night
  - Turn-around time from bug correction to validation is shorter
+ Dramatic reduction in level of effort required
  - PROBA-1 took less than half the LOE compared to typical mission
  - PROBA-2 took even less

– The size of the on-board C code is not as optimal as if it had been written by humans

– The computational efficiency is not as optimal

– Some common algorithms (e.g. for-loop, while) available in native code (C code) are more complicated to implement in model-based form.

✧ One needs to learn how the code generator works in order to optimize the models for code generation

CONCLUSION

COMPUTED-AIDED SOFTWARE ENGINEERING

IS THE WAY FORWARD

THANKS TO...

- Jimmy Côté*
- Aymeric Kron*
- Steve Ulrich

NGC Aérospatiale Ltée
NGC Aerospace Ltd

- Frédéric Teston
- Pierrik Vuilleumier*
- Stefano Santandrea

esa

- Pieter VanDer Braembussche*
- Joris Naudet*
- Dirk Bernaerts

VERHAERT SPACE
A QinetiQ Company

- and thanks for your attention