SATISH DHAWAN SPACE CENTRE (SDSC)
INDIAN SPACE RESEARCH ORGANIZATION (ISRO)

CRYOGENIC SYSTEM DYNAMICS AND VEHICLE INTERFACE ANALYSIS FOR GSLV AND LVM3 VEHICLES

By

C DHIYA

B TECH CSE (AIML)

Under the guidance of

Mr. K. MURALI, Engineer – SF, LSSF /SDSC SHAR/ISRO

January - February 2026

INTRODUCTION

The **Cryo Control System** is a vital part of the launch infrastructure, managing Liquid Oxygen (**LOX**) and Liquid Hydrogen (**LH2**) supply to the launch vehicle.

Key Systems Analyzed:

- **Intelligent Control System (ICS):** Performance validation is essential for mission safety.
- **Accumulator Charging Systems:** Monitoring pressure cycles and valve timing.
- **Cryo Arm Retraction:** Analyzing boom movement dynamics and retraction angles.

This project introduces a **Python-based Automation Suite** designed to streamline the post-mission analysis of these systems. By leveraging libraries such as **Pandas** for data manipulation and **Matplotlib** for visualization, the tool processes high-frequency telemetry data from Excel logs. It implements complex logic to map PLC timestamps to real-world events, analyzes accumulator charging cycles, and visualizes boom retraction trajectories.

DATASETS

Mark-2 Datasets

Dataset 1:

Dataset 2:

Dataset 3:

Mark-3 Datasets

Dataset 1:

Dataset 2:

Dataset 3:

What datasets contain (common description):

Timestamped telemetry values

LOX pressure measurements

LH2 pressure measurements

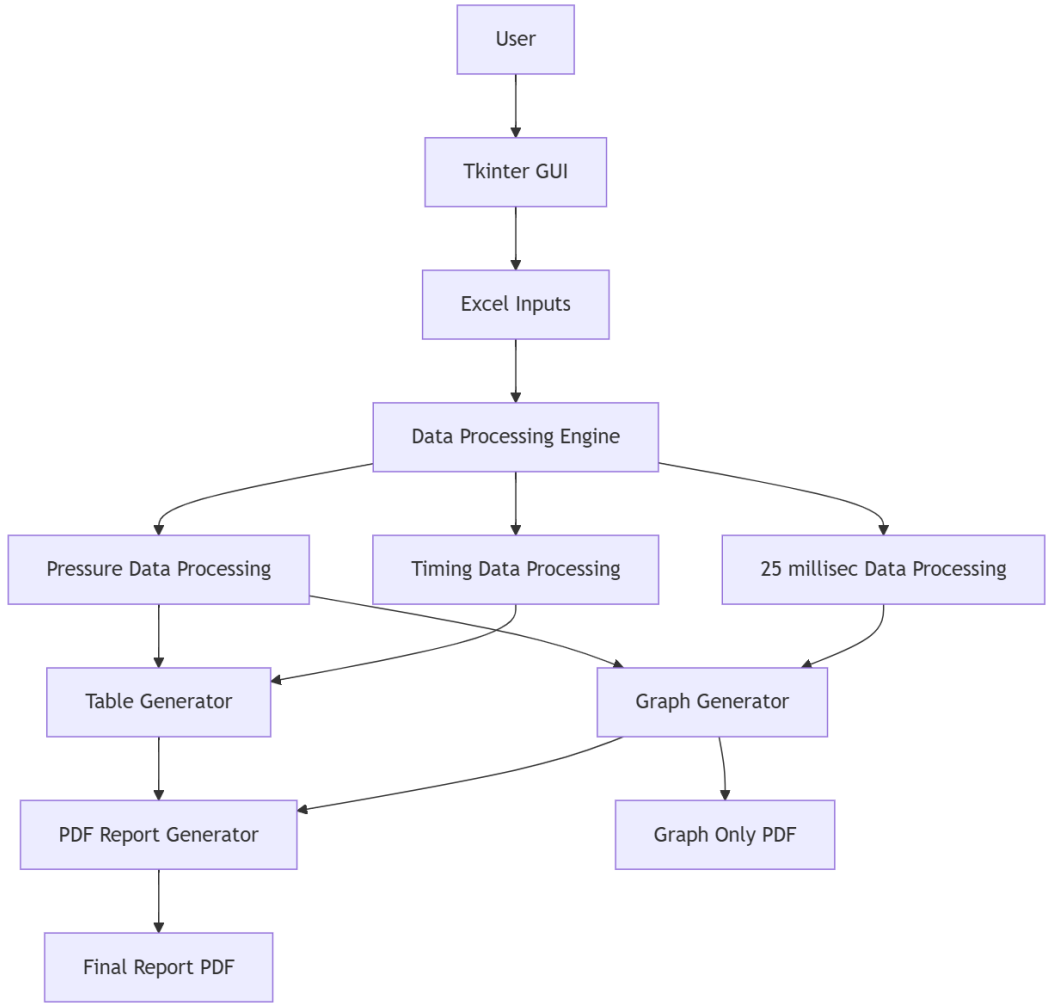
Accumulator charging/discharging values

Cryogenic arm boom position data

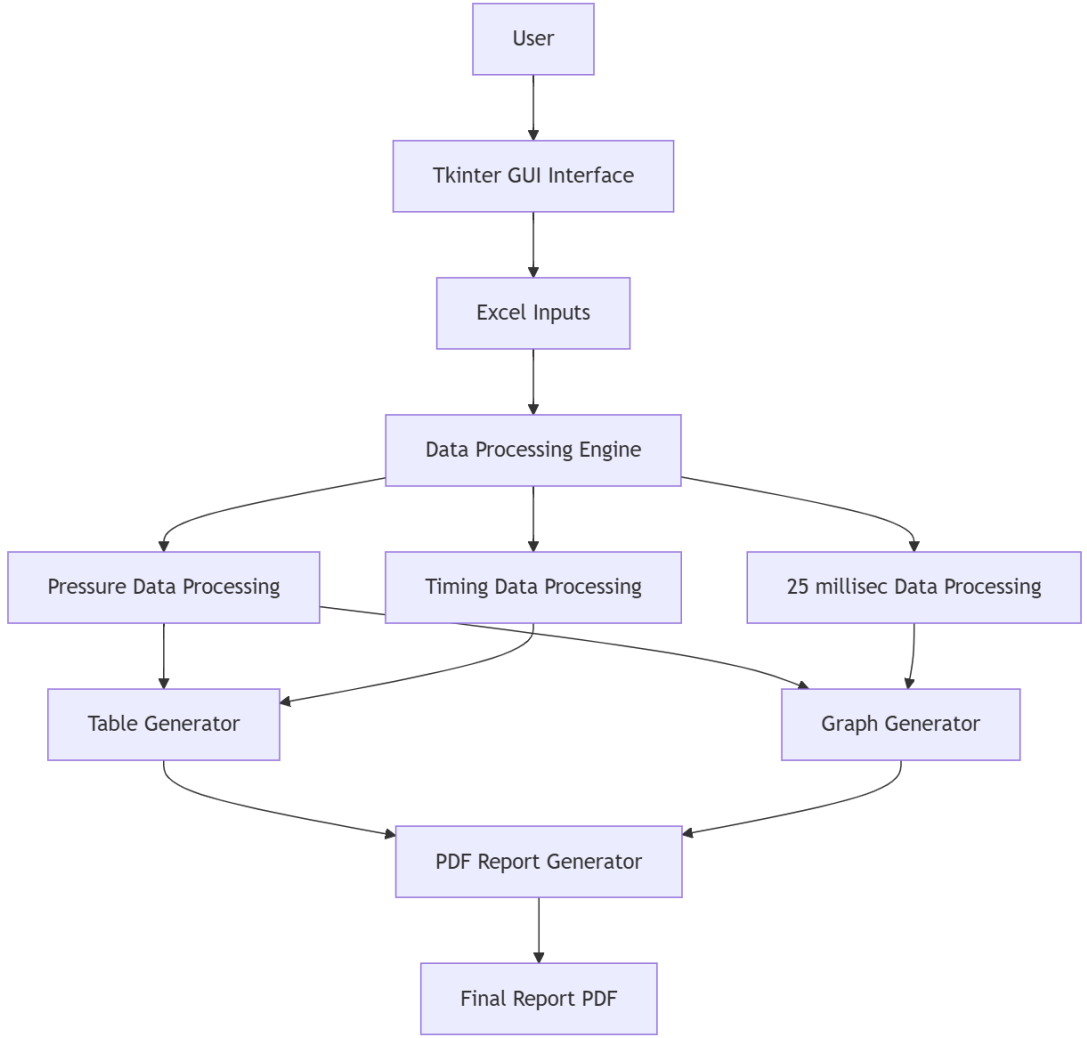
UCU separation event timings

SYSTEM ARCHITECTURE DIAGRAM

MARK II (GSLV)



MARK III (LVM3)



INPUT AND DATA PREPROCESSING STAGE

User Input Through GUI

System uses **Tkinter GUI** for uploading Excel files.

File selection handled using:

filedialog.askopenfilename()

User provides:

Event Date

Event Time

Inputs are passed to data processing functions.

Reading Excel Files

Excel files are loaded using **Pandas**:

```
pd.read_excel(file_path,engine=
"openpyxl", dtype=object)
```

Converts raw cryogenic data into structured dataframe.

Supports multiple Excel formats.

File Validation

System checks whether file is selected:

Displays error if file path is missing.

```
if not file_path:
    messagebox.showerror("Error",
"Please select Excel file")
```

Prevents processing without input data.

Column Validation

Column names standardized using normalization:

Removes special characters and converts to uppercase.

```
def normalize(col):
    return re.sub(r"[^A-Z0-9]", "", str(col).upper())
```

Required columns verified before processing.

Ensures correct parameters and handles inconsistent column names.

Date–Time Validation

User date and time converted to standard format:

```
user_datetime = datetime.strptime(
    f"{user_date} {user_time}",
    "%d-%m-%Y %H:%M:%S"
)
```

Invalid format triggers error message.

TABLE 1: ACCUMULATOR CHARGING

Logic Overview: This table captures the state of the system at specific countdown milestones (e.g., T-20 min, T-10 min).

Time Calculation: It takes the user-provided "Launch Date & Time" and subtracts specific time deltas (Offsets) to find the target time.

Nearest Neighbor Search: It searches the Excel data for the row with a timestamp closest to (but not exceeding) that target time.

Data Extraction: It pulls specific pressure values (Header & Pivot pressures) for that timestamp.

Report Column	Excel Column Source	Logic
Ref. Time	Calculated (Launch Time - Offset)	e.g., T-20 min, T-10 min
Header Pressure	LOX_HDR1, LH2_HDR1	Value at nearest timestamp
Pivot 1 Pressure	LOX_PVT1, LH2_PVT1	Value at nearest timestamp
Pivot 2 Pressure	LOX_PVT2, LH2_PVT2	Value at nearest timestamp

TABLE 2: ACCUMULATOR CHARGING TIMINGS

Logic Overview: This table maps the cyclic charging and discharging events to a predefined sequence of pressure steps (e.g., 0-230 bar, 0-160 bar)

Data Retrieval: Reads columns like LOX_CHG_TIME and LOX_DISCHG_TIME into lists.

Step Mapping: Defines a fixed list of steps (steps array).

Sequential Filling: It iterates through the step list. If the step implies charging ("180-230"), it pulls from the Charge list. If it implies discharging ("230-180"), it pulls from the Discharge list.

PLC Conversion: Converts raw PLC integers into HH:MM:SS:MS format.

Cycle Step (Row)	Excel Column Source (List Index)	Description
1. 0-230 bar	LOX_CHG_TIME[0]	First Initial Charge
2. 230-180 bar	LOX_DISCHG_TIME[0]	First Discharge
3. 180-230 bar	LOX_CHG_TIME[1]	First Recharge
4. 230-180 bar	LOX_DISCHG_TIME[1]	Second Discharge
... (Repeating)	...	Alternating Charge/Discharge
Last Recharge	LOX_CHG_TIME[-1]	Final Top-up before launch

TABLE 3: CRYO ARM DATA AFTER RETRACTION

Logic Overview: This table reports specific "snapshots" of events occurring during the arm retraction phase.

Pressures/Angles: Divided by 100 if required (e.g., 3700 becomes 37.00).

Timers: Converted from PLC counts to Time format.

```
def convert_plc_time(val):
    # 1. Parse raw integer
    ct = int(float(str(val)))
    # 2. Extract components
    milliseconds = ct % 100        # Last 2 digits
    total_sec    = ct // 100       # Remove MS
    hours   = total_sec // 3600
    minutes = (total_sec % 3600) // 60
    seconds = total_sec % 60
    # 3. Format string (HH:MM:SS:MS)
    return
f"{hours:02d}:{minutes:02d}:{seconds:02d}:{milliseconds:02d}"
```

Event Name	Excel Column (LOX / LH2)	Processing Logic
Header 1 Pressure	LOX_H1P1_UCU / LH2_H1P1_UCU	Raw Value
Pivot Pressure	LOX_PVT1_UCU / LH2_PVT1_UCU	Raw Value
UCU Separation	LOX_UCU_DETECTED / LH2_...	Convert PLC Integer to Time
Source Cutoff	LOX_SRCCUTOFF / LH2_...	Convert PLC Integer toTime
Boom at 37 Deg	LOX_BOOM37 / LH2_BOOM37	Divide by 100 (3700 $\to$ 37.00)
Boom at 50 Deg	LOX_BOOM50 / LH2_BOOM50	Divide by 100

PRESSURE GRAPHS (LOX & LH2)

- **Logic Overview:** This function visualizes the pressure trends of multiple sensors (Headers & Pivots) over a specific time window around the launch.
- **Time Parsing:** Merges the separate "Date" and "Time" columns from Excel into a single comparable datetime object.
- **Data Filtering:** Slices the dataframe to only include data from **T-40 mins to T+10 mins** (relative to the user-provided Launch Time).
- **Multi-Sensor Plotting:** Iterates through a predefined list of sensor columns (e.g., LOX_HDR1, LOX_PVT1) and plots each as a separate line on the same chart.
- **Smart Axis Formatting:**
  - **X-Axis:** Formats timestamps to show only HH:MM:SS.
  - **Y-Axis:** Dynamically calculates min/max values and snaps the limits to the nearest **20 bar** step.

BOOM ANGLE GRAPH

- **Logic Overview:** This graph compares the physical movement of the LOX and LH2 arms on a single timeline.
- **Dual-Sheet Reading:** Reads two separate sheets ("LOX" and "LH2") from the same Excel file.
- **Column Detection:** Dynamically finds columns containing keywords "TIME" (X-axis) and "ANGLE" (Y-axis).
- **Overlay Plotting:**
  - **LOX:** Plotted as a **Solid Red Line**.
  - **LH2:** Plotted as a **Dashed Blue Line** (for visual distinction).
- **Unit Handling:** The X-axis is kept in **Milliseconds** (raw PLC time) as per the source data.

GSLV REPORT

LVM3 OUTPUT

TECHNOLOGIES USED

Pandas: The backbone of data processing.
It reads the raw Excel files, cleans the data, and filters the specific time windows (e.g., T-20 to T+10 min) needed for the report.

NumPy / Math: Handles numerical operations and mathematical calculations.
It ensures accurate computation of pressure values and conversion of sensor data types.

Matplotlib: The plotting engine.
It generates the high-resolution **Pressure vs. Time** and **Boom Angle vs. Time** graphs, customizing axis labels, legends, and gridlines.

Tkinter: Provides the Graphical User Interface (GUI).
It creates the application window, buttons, and file upload dialogs, making the tool user-friendly for operators.

ReportLab: The PDF generation engine.
It programmatically builds the final report, arranging tables, headers, and embedding the generated graphs into a professional A4 layout.

OpenPyXL: The Excel driver.
It acts as the bridge allowing Python to read and extract data from .xlsx files, preserving cell formats and data integrity.

Datetime Module: Manages temporal logic.
It handles all time-based calculations, such as determining "T-minus" count-downs and synchronizing data across different files.

Regex (re): Ensures data consistency.
It uses pattern matching to "normalize" column names (e.g., removing spaces or special characters), ensuring the code finds the right data regardless of small formatting differences in the input files.

IMPACT & BENEFITS

1) **Instant Reporting:** Generates an industry-standard **PDF report** automatically using **ReportLab**.

2) **Efficiency Boost:** Reduces analysis turnaround time from **hours to seconds**.

3) **Error Elimination:** Removes the risk of human transcription errors in data logging.

4) **Digital Transformation:** Represents a significant leap in the digitalization of **Ground Support Equipment (GSE)** analysis.

THANK YOU

The Cryo Control System serves as a critical component in the launch infrastructure, managing the supply and control of Liquid Oxygen (LOX) and Liquid Hydrogen (LH2) to the launch vehicle. The analysis of the Intelligent Control System (ICS) performance, particularly the Accumulator Charging Systems and the Cryo Arm Retraction dynamics, is essential for validating mission success and safety.

This project introduces a Python-based Automation Suite designed to streamline the post-mission analysis of these systems. By leveraging libraries such as Pandas for data manipulation and Matplotlib for visualization, the tool processes high-frequency telemetry data from Excel logs. It implements complex logic to map PLC timestamps to real-world events, analyzes accumulator charging cycles, and visualizes boom retraction trajectories.

The final output is an auto-generated, industry-standard PDF report created via ReportLab, reducing the analysis turnaround time from hours to mere seconds while eliminating human transcription errors. This tool represents a significant step forward in the digitalization and automation of Ground Support Equipment (GSE) data analysis.