NPAS Physics Pump Models for Autonomous Monitoring

PREPARED BY PETTE JOSEF RAMOS
ELGIN COMMUNITY COLLEGE
FERNANDO FIGUEROA, MENTOR
AUTONOMOUS SYSTEMS LABORATORY
Overview

- Introduction
- Background
- Project Objectives
- Solutions
- Importance of ISHM
- Lessons Learned
- Skills Acquired
- Future Plans
- Acknowledgements
Introduction

- **Major:** Biomedical Engineering
- **College:** Elgin Community College in Elgin, Illinois
- **Prior to Internship:**
  - NASA Community College Aerospace Scholar
  - Biomedical Engineering Volunteer
  - Taekwondo Instructor
Background

- NPAS is a platform for autonomous operations developed at SSC’s Autonomous Systems Laboratory (ASL)

- The Intelligent Stennis Gas House Technology (INSIGHT) is a project that uses NASA Platform for Autonomous Systems (NPAS) to implement autonomous operations at the Gas House

- Initially INSIGHT is implementing autonomous operations of the Nitrogen system
Background From Previous Interns

- Andrew Yeo, a former NASA intern, provided the theory of how the centrifugal pump should work and left behind equations and parameters to begin coding in NPAS.

- Former Interns Kyle Bentain and Wilson Barce laid the groundwork for reasoning using the physics pump model.
Project Objectives

- Learn how to program using the G2 Software environment

- Create procedures and methods to detect cavitation in a centrifugal pump and other features such as efficiency

- Verify that the code and theory work by implementing test cases

- Begin implementation of the generic rotational pump model package for the High Pressure Gas Facility
Solutions

- Created a workspace to test the code of the previous interns and modified the code as needed
- Acquired data at the High Pressure Gas Facility to verify calculations
- Create rules to make relevant calculations automatically in 5 second intervals
PM-Functions workspace

This workspace was used to simulate a running pump and motor.
Instances/Objects

- Each instance is a representation of an object that is used at the Gas House with the exception of the “RULE” instance (more on that later).

- RP1 represents a centrifugal pump that is connected to MOTOR-1.

- The connectivity automatically enables use of the linked motor and pump parameters in the modeled equations.

- N2 represents the commodity that is being passed through the pump (RP1), which is Liquid Nitrogen.

- RULE is used to disable and enable the use of the model when needed.
Each instance contains a set of attributes

The attributes in this case hold raw data and calculations

These calculations are made through procedures and methods

Raw data was collected by reading pressure, temperature, and flow rate sensors at the HPGF
The following procedures have different functionalities:

- **Begin-calculations** calculates all attributes of a class through the use of a button.
- **Enable-and-disable-rule** turns a rule on/off when in administrator mode.

Buttons trigger these procedures to begin.

Rules were used to have the procedures run automatically.
Procedures, Rules and Buttons (cont.)
Programmatically Disabling and Enabling a Rule

DISABLE AND ENABLE RULE

- **UUID**: 761d99a0de0611e8813ad4258bf7213b
- **Notes**: OK
- **Item configuration**: none
- **Names**: RULE
- **X**: 1

**Rules**
- Unconditionally start begin-calculations (RP1)

**Buttons**
- Enable/Disable Rule

**Procedure**

```plaintext
RRULE, a pm-rules

<table>
<thead>
<tr>
<th>UUID</th>
<th>761d99a0de0611e8813ad4258bf7213b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notes</td>
<td>OK</td>
</tr>
<tr>
<td>Item config</td>
<td>none</td>
</tr>
<tr>
<td>Names</td>
<td>RULE</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
</tr>
</tbody>
</table>

**DISABLE AND ENABLE RULE**

- **UUID**: de9c03a36ed0711e8813ad4258bf7213b
- **Notes**: OK
- **Tracing and breakpoints**: default
- **Class of procedure invocation**: none
- **Default procedure priority**: E:

**Procedure**

```plaintext
FUNCTION disable-enable-rule ( RULE.OrdinalIgnoreCase )
BEGIN
  // This procedure attempts to enable and disable a rule when a button is pressed.
  // Note: To allow procedure to run, you need to be in administrator mode.
  
  // Initialize the window required for the procedure.
  INTEGER = the x of RULE;
  BEGIN
    // Conclude that the x of RULE = 0;
    IF ( x of RULE = 0 ) THEN
      // Conclude that item configuration is none;
      IF ( item configuration is none ) THEN
        // Conclude that names are RULE;
        IF ( names are RULE ) THEN
          // Conclude that x is 1;
          IF ( x = 1 ) THEN
            // Call 2-system-command (the symbol enable, ThisWindow, pump-rule, the symbol none);
            CALL 2-system-command ( enable, ThisWindow, pump-rule, none );
            // Post "Calculation Rule has been disabled";
            POST "Calculation Rule has been disabled";
          END;
        END;
      END;
    END;
  END;
END;
```

**Message Board**

- #840 4:03:41 p.m.
  The rotary pump RP1 does not fit the pump model.

- #841 4:03:41 p.m.
  The rotary pump RP1 does not fit the pump model.

- #842 4:03:46 p.m.
  The rotary pump RP1 does not fit the pump model.

- #843 4:03:46 p.m.
  The rotary pump RP1 does not fit the pump model.

- #844 4:03:48 p.m.
  Calculation Rule has been disabled
Logic

Button is pressed

Procedures and methods begin

Rule calculates automatically if enabled

Procedures calculate/update attributes

If a problem (in this case, cavitation) occurs, a message is posted to the operator
Common Problems

- One of the biggest problems when running a procedure lied within the units being used for each equation.
- As a way to distinguish the required units, documentation and comments were made for each procedure to keep all units uniform to the equations.
- Miscellaneous equation errors.

**UNITS:**

- Length/Distance: Inches (in)
- Weight: Pounds (lbs)
- Time: Minutes (min)
- Volume: Gallons (G) *[in^3 for calculations] [conversion factor: *231]
- Pressure: Pounds per square-inch (psi)
- Density: Pounds per cubic inch (lbs/in^3)
- Rotational Speed: Revs/Minute (rpm) [rev/s for calculations] [conversion factor: * 1/60]
- Linear Speed: Inches per second (in/s)
- Volumetric Flow Rate: Gallons per minute (GPM) [in^3/s for calculations] [conversion factor: * 3.85]
- Gravitational Constant: inches per second^2 (in/s^2)
- Power: in-lbs/s

**Note:** When doing calculations, convert all units to pounds, inches, and seconds.
Importance of ISHM

- The Integrated System Health Management (ISHM) is a component of NPAS that utilizes 4 functionalities:
  - Anomaly Detection
  - Diagnostics
  - Prognostics
  - Integrated Awareness
- Each of these functionalities work hand in hand to detect a problem in a system and to predict possible outcomes.
Moving Forward

- Further develop symcure functionalities for root-cause analysis
- Implementation to the Gas House
- Implementing a saturation table that include temperature, pressure, and density
Lessons Learned

- How to program in the G2 Software environment
- Familiarity and coding for NPAS
- Application of physics in real world situations
- The properties of cryogenics and liquid nitrogen and their affects in the operations at Stennis Space Center
- How the Gas House system functions
Skills Acquired

- Programming for the first time
- Coding using object-oriented programming
- Project management using scrum
- Root cause analysis
- Physics models for rotary pumps and motors
Future Plans

Spring 2019
Finish ECC with an Associates in Engineering Science

Fall 2019
Transfer to the University of ???

Future
Graduate with a Bachelors in Biomedical Engineering

Spring 2020
Co-Op?

Graduate School
Masters in Biomedical Engineering
Acknowledgements

- Fernando Figueroa, NASA Mentor
- Andrew Tidwell, D2K Contractor
- Mark Turowski, NASA- High Pressure Gas Facility
- Wilson Barce, Former NASA Intern
- Lauren Underwood, NASA
- D2K Technologies (Mark Walker, Jon Morris, Neal Gross, Joshua Broberg, Brian Rey)
Thank You NASA!