

Gas Compressor System Simulation using Functional Mockup Interface for Human Machine Interface and Control

Ryan Magargle¹, Hemanth Kolera-Gokula¹

¹ANSYS, Inc. USA, {ryan.magargle, hemanth.kolera-gokula}@ansys.com

Abstract

This paper describes a gas compressor (GCS) system simulation for the purpose of verifying a controller's operation and interfacing simulation with measured data. The GCS is used to collect and compress low pressure gas streams for transmission into larger higher-pressure lines. This system is simulated with ANSYS TwinBuilder. It includes models of the compressor, motor, bypass valve, and piping along with pressure sources for the low and high-pressure lines, which are controlled with measured or test data. To cycle and pressurize the GCS, a controller is implemented using an FMU from ANSYS SCADE. To interact with the system and monitor current operating conditions, a human machine interface (HMI) is also implemented using ANSYS SCADE as an FMU.

Keywords: digital twin, vapor recovery unit, FMI, ANSYS, TwinBuilder, SCADE, control, HMI, Rapid Prototyper

1 Introduction

The GCS system shown in Figure 1 was created using ANSYS TwinBuilder (ANSYS, 2017). The GCS system compresses the low-pressure gas in the storage unit on the left through a low pressure reservoir and sends it to the outlet line on the bottom right. While it is pressurizing and sending the gas to the outline line, the bypass valve is closed so no gas is recycled back into the scrubber tank. If the input pressure fed from the storage goes too low, the compressor can enter a low

power mode and the bypass valve will open allowing gas to recycle back through the scrubber tank. The scrubber tank is responsible for removing contaminants from the gas flow, removing them as a second liquid phase, but is not critical for the flow analysis considered in this model, and is treated as a simple tank.

The usage cases for this type of model are 1) verification of controller operation, and 2) connection with real-time data to provide analytical support in health monitoring applications. The discussion considered within this paper primarily addresses the first usage case. Extension of the existing model to be a connected simulation based digital twin will be left to future work and has been discussed in previous papers.

In verifying the controller operation, measured source pressure data from the field can be replayed as an input to the system schematic and run under the supervision of the controller to ensure all states are properly entered and exited. Alternatively, test signals can be input at the source to cycle through all possible input pressures. The models implemented in the system described here are lower-fidelity, high speed quasi-steady state models able to simulate large time periods at the expense of local dynamic behavior.

2 Model Overview

The various models implemented for the system components are described in this section. The main schematic consists of numerous top-level macro-

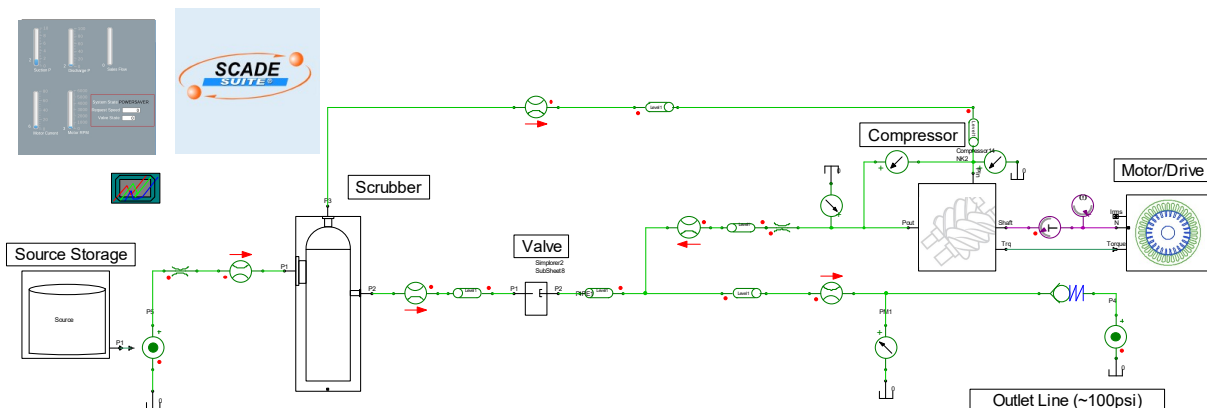


Figure 1. System Schematic in ANSYS TwinBuilder

models, or packaged models containing various sub-models and elements. This keeps the top-level schematic from getting too cluttered from components that may have multiple sub-components in the system. Macro-models schematics are preserved and maybe be entered and modified at any time, and any of their internal states may be made available to the main schematic.

2.1 Source Storage

The storage source is treated as a pressure source. This behaves as the boundary to the simulation where measured data can be input or test signals can be applied.

As seen in Figure 2, the macro-model feeds a pressure source. The macro-model has two paths defined. The signal flow path on the top simply applies a triangular waveform to sweep the pressure. The signal flow path on the bottom imports a text file containing the pressure data obtained from the field and applies this to the output of the macro-model to feed the pressure source. One signal flow path can be active at a time, the other signal flow path is simply deactivated.

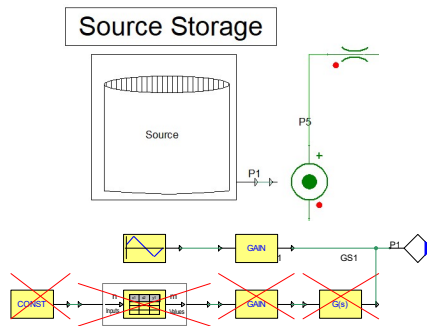


Figure 2. Macro-model and sub-circuit of source storage model.

2.2 Scrubber Tank

The scrubber tank is treated as a basic tank in this model. As seen in Figure 3, The macro-model has three connections, the source storage (left), the compressor suction (top), and the recycle from the bypass valve (right). The sub-circuit model is a standard Modelica model of a close volume vessel from the Open Modelica library (Modelica-Fluid-Vessels-ClosedVolume).

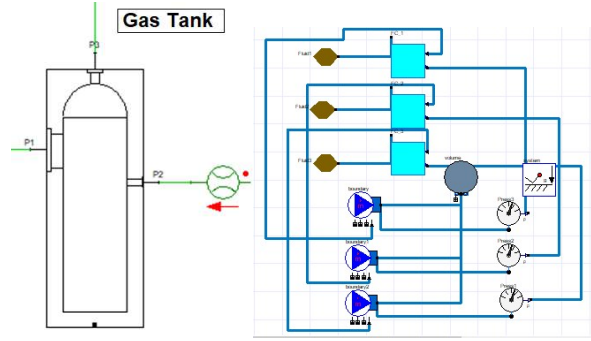


Figure 3. Macro-model and sub-circuit of scrubber tank.

2.3 Compressor

The compressor model uses an analytical pressure-flow relation,

$$Q = \eta \frac{Vol \omega}{2\pi} \Delta P \quad (1)$$

Where Q is the flow rate, η is the volumetric efficiency, Vol is the displacement volume of the compressor, ω is the compressor speed, ΔP is the compressor pressure. η is fit to an analytical function based on a regression on the measured datasheet values. And can be therefore customized to any manufactured pump.

Based on the datasheet information, shown in Figure 4, the flow pressure relations are derived. In general, a multi-dimensional function vs pressure and speed can be generated, but based on the low variation vs speed, the average curve was used to calculate the flow vs pressure. The resulting equation used for η is shown in Figure 4.

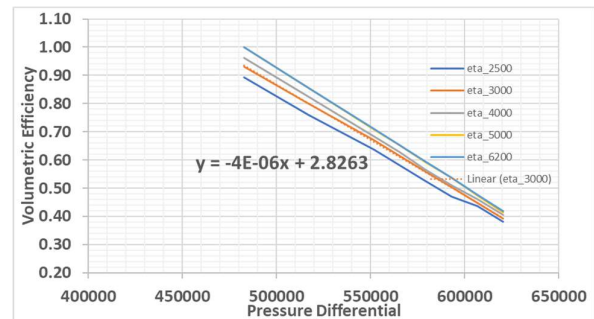


Figure 4. Datasheet values for compressor, shown as volume efficiency vs differential pressure. The different curves are for different compressor speeds.

The compressor sub-circuit then consists of an equation block that calculates the flow as a function of pressure and additionally calculates the compressor's flow torque according to:

$$\tau = \frac{Vol}{2\pi} \Delta P \quad (2)$$

The resulting flow is applied to a flow source in series with the macro-model flow input and output ports, as seen in Figure 5. The torque is passed as a quantity out of the compressor macro-model for use with the Motor/Drive to calculate the electrical loading, described in the following section.

The compressor model also contains pressure relief valves and volumetric capacity to numerically stabilize the solution and account for some compressibility in the excess volumes and connectors.

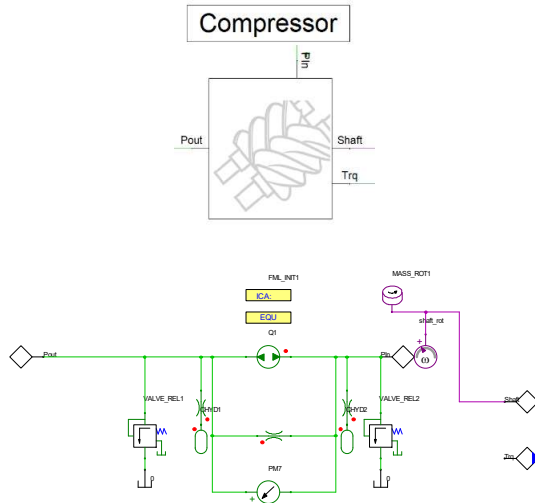


Figure 5. Macro-model and sub-circuit of compressor.

2.4 Motor/Drive

The motor and drive model in this system are a steady state representation of an induction machine with basic constant V/f drive. A standard TwinBuilder induction machine model was combined with a V/f drive, which was then parametrically analyzed over all possible input drive request speeds and required torques. Then the actual speed with slip and the resulting RMS current were saved to create a steady state table, as seen in Figure 6.

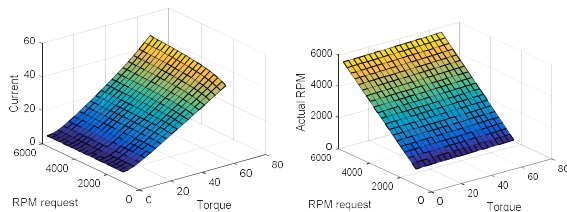


Figure 6. Steady state response surfaces for induction machine paired with V/f drive, given torque and RPM request inputs and current and actual RPM outputs.

The response surfaces are imported as CSV files into the TwinBuilder ND table element. Inside the motor/drive

macro-model, the inputs of Torque and Speed are wired to input ports and outputs of RMS current and Speed are wired to output ports, as seen in Figure 7.

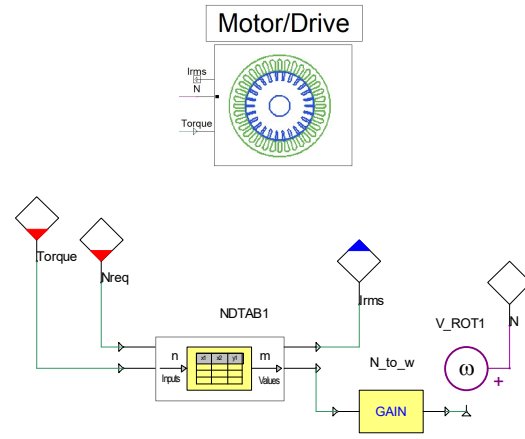


Figure 7. Macro-model and sub-circuit of electric motor and drive.

2.5 Bypass Valve

The bypass valve takes an input command from the controller and opens when in standby or low-pressure conditions and closes when pressurizing. As seen in Figure 8, the valve consists of a variable orifice component that fully opens and closes with a mechanical actuation of 0 to 10cm.

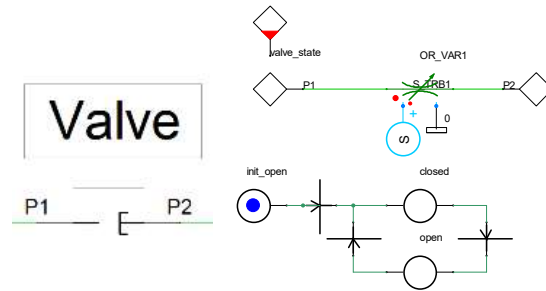


Figure 8. Macro-model and sub-circuit for bypass valve.

The actuation is described by a function, S , that transitions from on to off and off to on continuously (Funahashi, 1989),

$$S = \frac{cmd_{on}}{1 + e^{(-\alpha(time - t_o) - t_d)}} + \left(\frac{1 - cmd_{on}}{1} - \frac{1 - cmd_{on}}{1 + e^{(-\alpha(time - t_o) - t_d)}} \right) \quad (3)$$

Where cmd_{on} is the controller command, α is the switch rate, t_o is the activation time of cmd_{on} , and t_d is the delay from activation time until switching begins. In this case, $\alpha = 2000$, and $t_d = 5 \times 10^{-3}$ s, as seen in Figure 9, with $t_o = 5 \times 10^{-3}$ s.

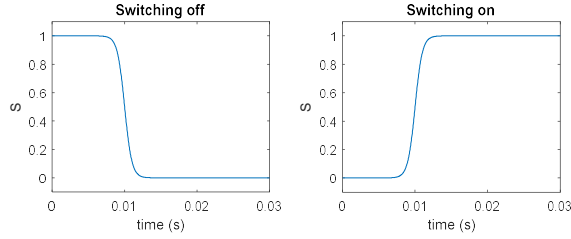


Figure 9. Switching function for opening and closing valve continuously when closing (left) and opening (right).

The state machine in Figure 8 sets t_o equal to the current time whenever it transitions from open to closed and vice versa. The transition event will occur whenever the controller sends the open or close command. This will create a 5ms delayed continuous transition from open to closed or closed to open. The continuity of the transition function and its derivatives helps with the stability and speed of the simulation.

2.6 Piping

The pipe models used to connect all components use standard TwinBuilder pipe elements. As seen in Figure 10, the pipe elements are characterized by the geometry and material properties of the flowing media. These are single phase media models only.

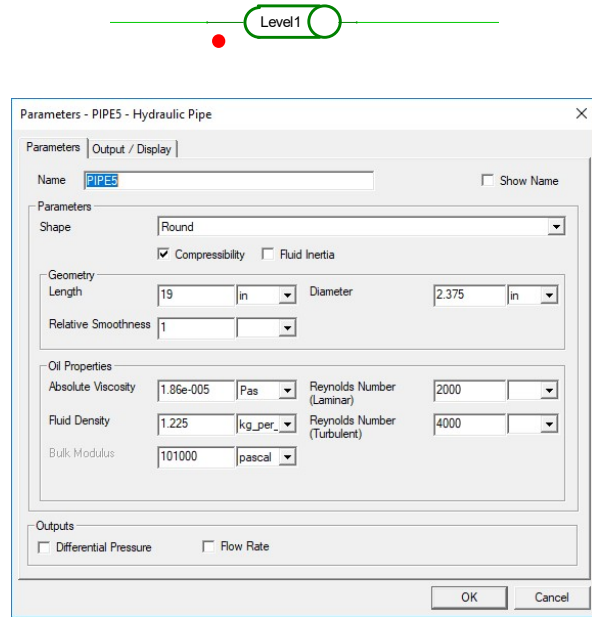


Figure 10. The pipe element and component dialog box for parameter input.

They use a standard flow, pressure relation based on the pipe friction function (Haaland, 1983):

$$P = f \frac{l \rho V^2}{2D} \quad (4)$$

Where f is the friction function using Hagen-Poiseuille and Blasius laws in laminar and turbulent flow respectively, l is the pipe length, ρ is the density, V is the flow velocity, and D is the diameter.

2.7 Functional Mockup Interface (FMI) Controller and Human Machine Interface (HMI)

The controller and HMI use ANSYS SCADE (ANSYS, 2017) to write and compile the code into an FMU for use in TwinBuilder.

The HMI uses the standard rapid-prototyping editor to create the display shown in Figure 11. This display is used to monitor the motor current and speed, in addition to the pump input and output pressures together with the outlet line flow (sales flow). This HMI can also be used to manipulate the simulation, such as manually opening or closing valves, turning pumps on or off, or modifying control gains on the fly. The current HMI, shown in Figure 11, is used for monitoring only.

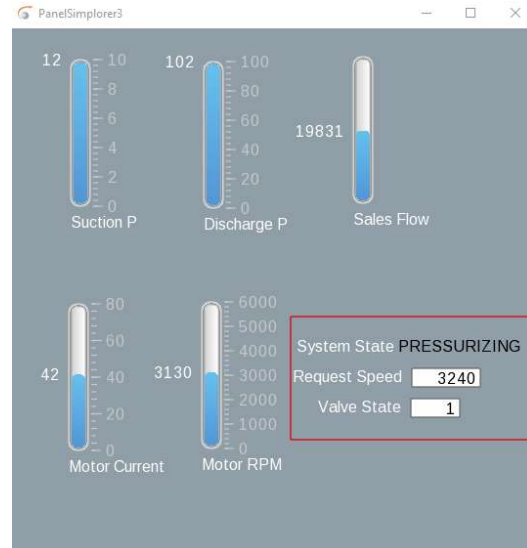


Figure 11. HMI interface for monitoring the simulation states.

Similarly, the controller is also created in SCADE and imported into the TwinBuilder schematic as an FMU. The controller monitors the input and output pressures and regulates the bypass valve together with the compressor speed. The current state of the controller is reflected in the HMI.

3 Results

The results of a sweep on the source pressure are shown in Figure 12. The input pressure is swept from 1 psi to 17 psi. The pump remains off until the pressure passes a given threshold of 6 psi, at which point the outlet pressure rises to match the outlet flow line, then steps up the motor speed as the inlet pressure continues to rise. The outlet flow rises accordingly.

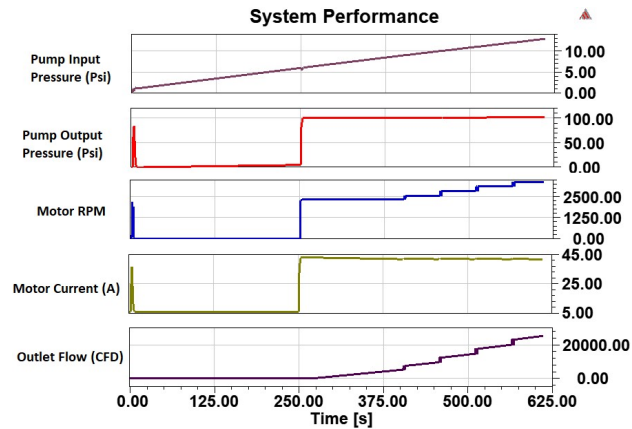


Figure 12. System results for swept input pressure.

4 Summary and Future Work

The current model can be used to validate controller operation for various input conditions. Interaction and display can be achieved using a human machine interface. Future models can be expanded to include real time connection with pressure readings from the field. This can enable additional operational insight using the simulation based digital twin virtual sensors.

References

- ANSYS, Inc, Canonsburg, PA. (2017). *TwinBuilder*. <http://www.ansys.com>.
- ANSYS, Inc, Canonsburg, PA. (2017). *SCADE*. <http://www.ansys.com>.
- S.E. Haaland. Simple and Explicit Formulas for the Friction Factor in Turbulent Pipe Flow. *J. Fluids Eng*, 105(1): 89-90, 1983. doi: 10.1115/1.3240948
- K-I Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3): 183-192, 1989. doi: 10.1016/0893-6080(89)90003-8