

Soft Computing Applications on SR-30 Turbojet Engine

Airo Watanabe, Semih M. Ölçmen, Robert Leland, and Kevin W. Whitaker
The University of Alabama, Tuscaloosa, AL 35487-0280

Luis C. Trevino
NASA George C. Marshall Space Flight Center, Huntsville, AL 35812

The use of soft-computing algorithms in hardware-in-the-loop applications has been investigated. A Proportional-Integral-Derivative (PID) classical controller and a Fuzzy Logic Controller (FLC) were designed and successfully tested on the Turbine Technologies SR-30 turbojet engine for the main-stage operation of the engine. Transfer function model of the plant was obtained using frequency-response method. Closed-loop controllers both with the PID and FLC algorithms were tested in a simulated environment before their application. Additionally, application of Bayesian Belief Networks (BBN) for the start-up sequence of the engine was investigated and a BBN design has been made.

NOMENCLATURE

PR	=	pressure ratio
D_{exit}	=	nozzle exit diameter
A/D	=	analog to digital
I/O	=	input and output
\dot{m}	=	mass flow rate
$P(s)$	=	SR-30 turbojet engine transfer function
$G_c(s)$	=	PID controller transfer function
K_p	=	proportional gain
T_i	=	integral time constant
T_d	=	derivative time constant

With this in mind, applications of the soft computing technologies to improve rocket engine control have been investigated. PID (Proportional-Derivative-Integral and FLC (Fuzzy Logic) controllers have been designed and implemented on a small-scale turbojet engine to demonstrate these technologies in a real hardware-in-the-loop environment. A design of a BBN (Bayesian Belief Networks) controller has been made and the concerns in application of such a controller have been identified.

In the following sections of the paper, first a brief description of a small-scale turbojet engine manufactured by the Turbine Technologies is presented. Next, the details of frequency-response design method and the plant transfer function identification are discussed. The use of the plant transfer function in developing a closed-loop model with a PID controller using MATLAB and the demonstration of the closed-loop model on the engine using NI LabVIEW software are presented next. These discussions are next followed by the presentation of the fuzzy-logic controller design, its verification on a simulated engine using the same plant-transfer function mentioned earlier and its application on the engine. Both the simulation and the application of the fuzzy-logic controller were made using NI LabVIEW software. The rest of the paper describes the applicability and design approaches for the BBN algorithm.

I. INTRODUCTION

The main principle of the joint effort undertaken by NASA-MSFC and the University of Alabama was to provide alternative methods in improving overall engine control. Real life challenges such as, harsh engine environments of cryogenics, vibrations, real-time control demands, and different engine configurations from test to test continually encourage researchers to determine alternative solutions and improvements to engine control, anomaly detection and mitigation. Current control technologies depend on proven, sometimes archaic, hardware and logical programming techniques, which are costly to implement and maintain, and do not account for unforeseen conditions such as premature engine shutdowns, cavitating pumps, propellant leaks, and numerous cases of anomalous sensors and data.

II. SR-30 TURBOJET ENGINE AND FACILITY

Turbine Technologies' SR-30 Turbojet Engine

The SR-30 engine is a turbojet engine with a single-stage radial-flow compressor with a maximum pressure ratio of, $PR=3.4$, a single-stage axial-flow turbine, and a reverse-flow annular combustion chamber and it operates obeying the Brayton thermodynamic cycle in the same fashion as large turbojet engines (Figure 1). The engine, as produced by Turbine Technologies, includes five pressure transducers, five thermocouples, a load-cell for thrust measurements, a custom motor winding for reading the engine RPM, and a fuel-flow-rate measurement system to monitor/measure the operating parameters of the engine. The engine generates 20 lbs of thrust at 90,000 RPM while ingesting $\dot{m} = 1.1$ lb/sec of air. The engine has a length of 10.75 in., and the exit exhaust diameter of $D_{exit}=2.25$ inches. Further details can be found in the paper by Watanabe *et. al* [1].



Figure 1 SR-30 Turbojet Engine

The engine system is equipped with a data acquisition board, addition of connection panels for the sensors, a stepper motor as fuel-flow rate valve positioner, and several relays to control several switches for engine operation.

Data Acquisition System

The current system at the University of Alabama includes a National Instruments NI PCI-6031E A/D board, which has a 16-bit resolution, 100 kSamples/sec sampling rate, 64 analog input, 2 analog output, and 8 digital I/O ports. Signal connections to the A/D board are made using two enclosures, which are attached to

the A/D board using cables. While one of these units (NI-SC-2345) houses the thermocouple input modules (NI-SCC-TC02), the millivolt range input modules (NI-SCC-AI06), a connector block for digital output signals, and the analog output modules, the other unit (NI-CA-1000) houses a connector block (NI-CB-68LPR) to connect other analog inputs. Each input module includes self-contained signal-conditioning units such as low pass filters and instrumentation amplifiers.

While the thermocouple input modules are used to read the temperatures, the millivolt input range modules are used to read the load-cell output voltage and one of the pressure signals. Digital I/O lines are used to generate TTL signals to turn on and off the relays. Relays (BASCO Company, ELK 924) are used to replace manual switches, which are used to start and to stop the ignition, the fuel flow and to turn on and off the valve for high-pressure air.

Fuel Flow Valve Controller

The control output signal for the Single-input-single-output (SISO) system under study is the signal to control the fuel-flow rate. The fuel-flow rate in the engine is adjusted by controlling valve opening to obtain a desired thrust value. The operation of the control-valve is automated with the use of a Compumotor-ES22BS stepper motor. The ES22BS stepper motor is mounted on a simple stand to hold the motor steadily. The stepper motor has a maximum torque of 9.4 lb-in, a maximum resolution of 128,000 Step/rev with the use of the GT-L5 drive, and the shaft inertia is 1.1 oz-in².

III. PID CONTROLLER DESIGN USING THE FREQUENCY-RESPONSE METHOD

A PID type classical controller was designed and implemented using the frequency-response method to identify and learn the engine characteristics during closed-loop operations. This section outlines the step-by-step design of the PID controller and the results of its implementation.

A series of data sets for the frequency-response design was obtained using a LabVIEW program. The chosen input and output for the plant, the SR-30 turbojet engine, are the fuel flow valve position in degrees and the thrust force in pounds, respectively. To give the sinusoidal input excitation, the valve position was excited in a sine-wave manner with 5 degrees of a mean position, corresponding to a steady-state thrust value of about 5.3 pounds, and 4 degrees of peak-to-

peak amplitude. Different frequencies of the excitation were attempted to cover a frequency range of interest.

In order to construct a pair of Bode Plots for the frequency-response design method, gains and phases need to be calculated in the region of frequencies. However, raw measurements of signals from transducers usually contain noise. The load-cell system of the SR-30 engine is not an exception and the thrust output signal contains such noise. Therefore, the thrust output signal was conditioned using the “filtfilt” command of MATLAB is used to condition the thrust output signals. After the signal conditioning, the peaks of the thrust output were found by seeking sign changes of signal derivatives. A Gain at each frequency was calculated finding an average of the peak-to-peak values. An example of the raw measurement and the filtered signals of the thrust output with the input signal frequency of 6.0 rad/sec are shown in Figure 2.

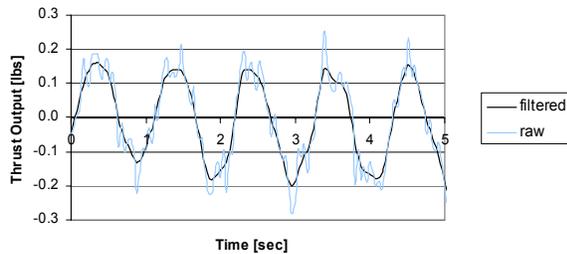
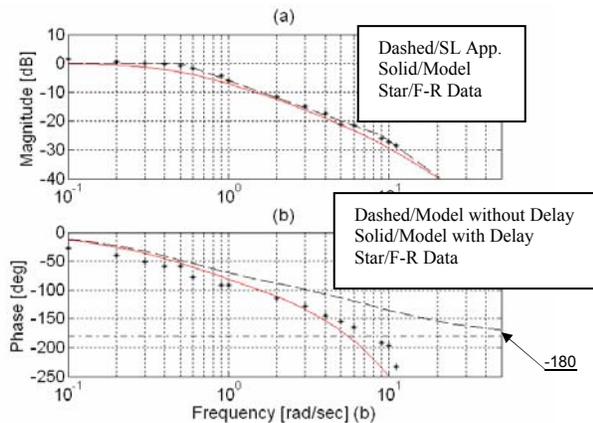


Figure 2 Raw and Filtered Thrust

III-A. PLANT IDENTIFICATION

The frequency region of our interest ranges from 0.1 rad/sec to 11.0 rad/sec. The data obtained from the sine-wave data collection is plotted on the pair of Bode



Figures 3a & 3b Open-Loop Bode Plots

plots in Figures 3a and 3b. The plant is modeled with MATLAB. Assuming a second order system, corner frequencies of 0.5 and 9.0 rad/sec were identified. The assumed open-loop transfer function of the plant is:

$$P(s) = \frac{4.5}{(s + 0.5)(s + 9.0)} \quad (1)$$

Although, the first-estimated model, the red solid line, matches in a satisfactory manner in the magnitude Bode plot, there is an obvious discrepancy between the first-estimated model (black-dashed) and the frequency-response data in the phase Bode plot. Therefore, the presence of a time-delay in the model was suspected. After several attempts of time-delay modeling, adding 0.2 seconds of time-delay model with the MATLAB “pade” command to the plant (red-solid) showed a consistent matching with the frequency-response data. The estimated model with the time delay is plotted in the Bode plots with dashed-blue lines. The assumed open-loop transfer function of the plant with the time-delay is:

$$P(s) = \frac{4.5}{(s + 0.5)(s + 9.0)} e^{-0.2s} \quad (2)$$

The assumed open-loop transfer function of the plant with the 4th order “pade” approximation time delay is:

$$P(s) = \frac{4.5(s^2 - 57.9s + 914)(s^2 - 42.1s + 1149)}{(s + 0.5)(s + 9.0)(s^2 + 57.9s + 914)(s^2 + 42.1s + 1149)} \dots(3)$$

III-B. PID CONTROLLER DESIGN AND TEST

Proportional-Integral-Derivative (PID) control approach was chosen for the engine control since it is one of the popular closed-loop control approaches that can be applied to a wide range of engineering problems. One customary representation of such controller can be expressed as:

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (4)$$

After attempting several combinations of numbers for the PID gains using MATLAB SISO tool program, a combination of $K_p = 3.68$, $T_i = 2.16$ sec, and $T_d = 0.061$ sec was found to be a reasonable controller for the application.

Figure 4, shows that the proposed controller gives a positive gain margin of 12.6 db and a positive phase margin of 65.8 deg, which make the system stable. Since the system has a positive gain margin, there is a tolerance of raising the proportional gain of the PID controller.

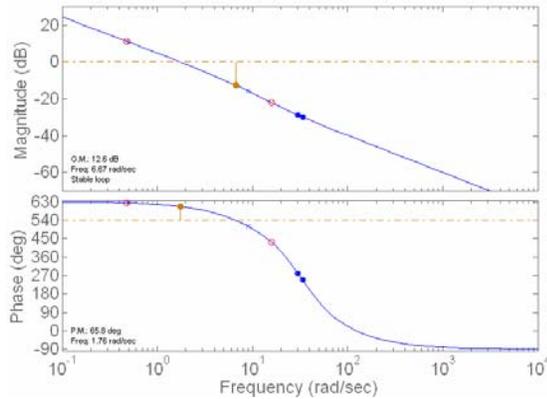
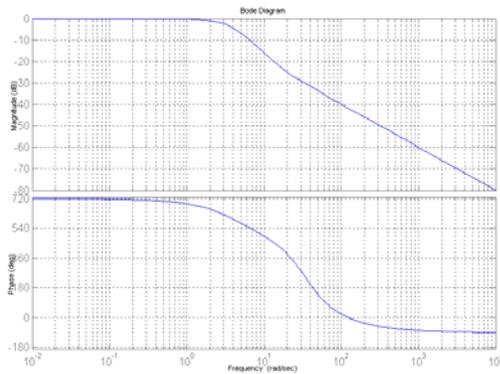


Figure 4 Open-Loop Bode Plots

Figure 5 is the closed-loop Bode plots for the system. From the gain plot, the proposed closed-loop model has a cutoff frequency of about 3.01 rad/sec.



Figures 5 Closed-Loop Bode Plots

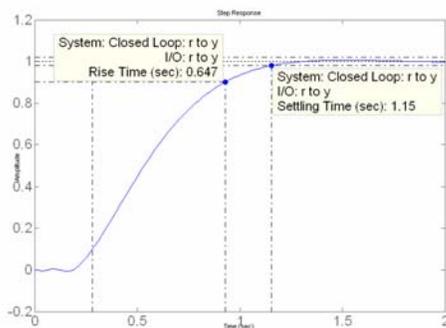


Figure 6 Simulated Unit Step Response

Cutoff frequency is the maximum frequency at which the output of a system will track an input sinusoid in a satisfactory manner.

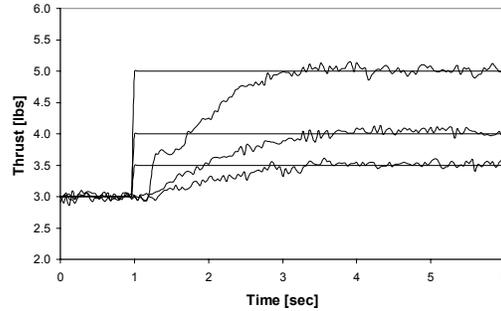


Figure 7 Step Responses from SR-30 Engine

Step input tests were conducted to test the designed PID controller. Figure 6 is the result from estimated model using MATLAB. It shows a rise time of 0.65 sec and a settling time of 1.15 sec. Figure 7 shows the experimental results of step input tests using the SR-30 engine. In the figure, three cases were plotted. For each case, a step command was issued at $t = 1.0$ sec. The thrust values for the three cases were 3.5, 4.0, and 5.0 lbs. For all cases the results show about 2 seconds of settling time with near zero steady-state errors.

The large initial jump observed in the 3 to 5 lb increase thrust run is undesirable and it is due to unforeseen nonlinear effects. This jump does not appear when using the fuzzy controller.

IV. FUZZY LOGIC CONTROLLER DESIGN AND APPLICATION [2]

The employment of Bayesian and fuzzy techniques derive from the “soft computing” family of algorithms. Here “soft computing” refers to computational mechanisms that can determine suitable relationships (in a system data set) to assess and determine a quantitative opinion(s) based on future conditions. Since the scope of this paper is not on the depth of understanding Bayesian and fuzzy techniques, application of them is presented. In short, within MSFC, such computational mechanisms are viewed as a collection of algorithms that can achieve optimal or near-optimal results in the presence of imprecise data, uncertainty [3], unknown physics, and probabilistic outcomes. The central goal in soft computing is to attain more robust response.

Similar to the PID design, for main-stage control, a fuzzy logic controller was designed and implemented to control the SR-30 engine and the response of the engine to a step input using the PID and Fuzzy controllers was compared. The use of fuzzy logic was seen suitable since it accommodates the uncertainties associated with controller architecture. Fuzzy logic is a branch of mathematics that deals with approximate reasoning. Fuzzy logic, developed by Lotfi A. Zadeh of the University of California at Berkeley, combines the topics of multi-valued logic, probability theory, and artificial intelligence for simulation of human thoughts by using computer software as a medium. The technology of fuzzy logic enables a computer to make decisions based on vagueness or imprecision intrinsic in most physical systems. Fuzzy logic also provides a convenient way to introduce useful nonlinearities into the control law to achieve specific effects, such as reducing large overshoots. In depth details of fuzzy logic are outside to scope of this paper. A more detailed description on fuzzy logic can be found in [5].

In the rest of this section the fuzzy logic controller design and related programming process are discussed. A brief description of the design of the fuzzy logic control algorithm is introduced. A simulation program written to test the fuzzy logic controller and the results of the simulation are discussed. Implementation of the fuzzy logic control algorithm on the SR-30 engine using LabVIEW is detailed and the hardware-in-the-loop test results are discussed.

IV-A. FUZZY LOGIC ALGORITHM DESCRIPTION

Fuzzy Logic algorithm differs from the PID control algorithm since it can be used for non-linear-time-variant systems. The algorithm intends to mimic human thoughts in the sense that the relation between an input and its output is assessed by a user, depending on the user's experience. In the fuzzy logic terminology, vague terms used such as hot, and cold are called linguistic terms and range of real numbers for the input and the output is referred to as universe of discourse. Range of the input values divided into sections by its linguistic terms is called antecedence and range of output values divided into sections by its linguistic terms is called consequence.

In a fuzzy logic control system, the fuzzification process translates real input values to linguistic input values, next the fuzzy inference system drives a

conclusion in a form of linguistic output values according to its rule base, which resembles IF-THEN structure used in many text-based programming languages. The final process called the defuzzification process translates back the linguistic output values to corresponding real output values for subsequent control actions.

IV-B. FUZZY LOGIC CONTROLLER ARCHITECTURE

The NI LabVIEW Fuzzy Logic Controller Toolkit was used to develop fuzzy logic controllers for the SR-30 engine control application. The toolkit allows the user to interface with LabVIEW programs. The Fuzzy Set Editor specifies the details of the fuzzification step, and the details of defuzzification step. The Rule Base Editor of the Fuzzy Logic Controller Toolkit enables the user to specify the details of the fuzzy inference system. Before testing a fuzzy logic algorithm with the SR-30 engine, a LabVIEW-fuzzy simulation program was developed using the plant model identified in the design process of the PID controller. The fuzzy simulation program had similar inputs to the ones of the LabVIEW-PID controller and the controller output again was the fuel-flow valve position. The fuzzy simulation program calculates appropriate valve position in every loop execution for adjusting the fuel-flow valve opening to obtain desired thrust. A picture of the Labview fuzzy simulation program developed for this purpose is shown in Figure 10. The input variables used in the design of the PD part of the controller were thrust error calculated at node (①) and thrust error rate calculated at node (②). As for the integral part of the program, the thrust error was the only input variable. The PD part and the integral part evaluate their outputs to the stepper motor separately from their inputs. The sum of the angle outputs calculated at node (③) from the two parts is the total angle output to the stepper motor to control the fuel-flow valve position.

The Figure 11-A shows the architectures of the fuzzy sets associated with the thrust error and the thrust error rate as the controller inputs and the Figure 11-B shows the architectures of the fuzzy sets associated with the valve position and the valve position error (See Table 1 for the corresponding linguistic values for each output). The use of the thrust error and thrust error rate each with five linguistic terms results in $5^2 = 25$ rules for the PD part. The use of the thrust error with five linguistic variables results in five more rules for the integral part bringing the number of the total rules to thirty. The choice of using thirty rules gives a reasonable number of rules and allows the user to

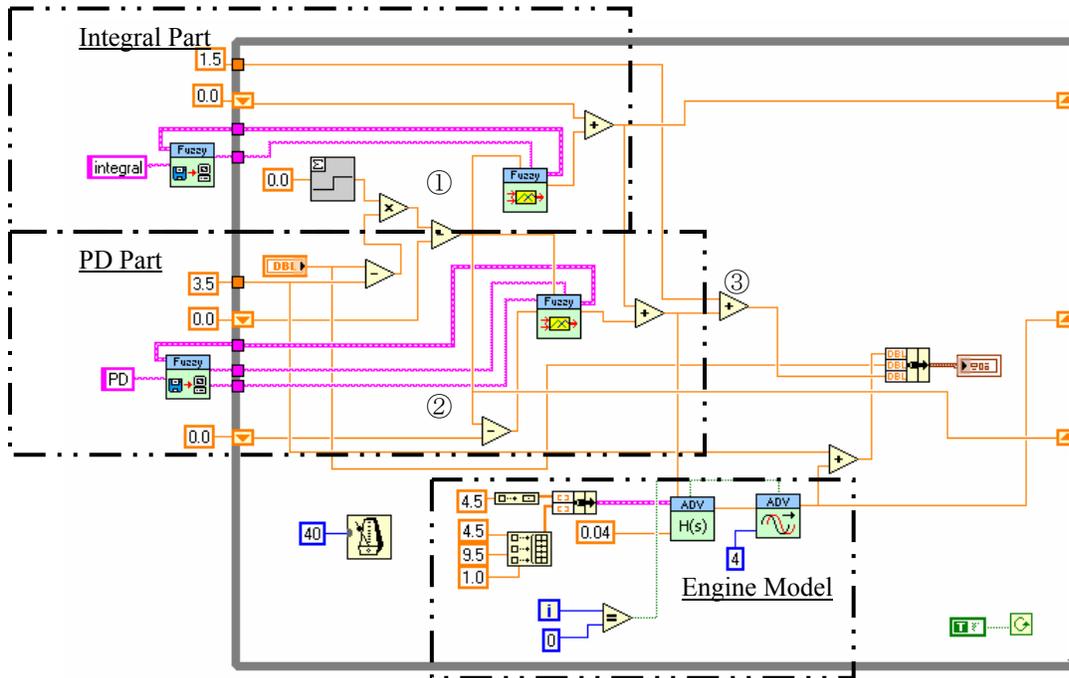


Figure 10: Fuzzy Logic Simulation VI (wiring diagram)

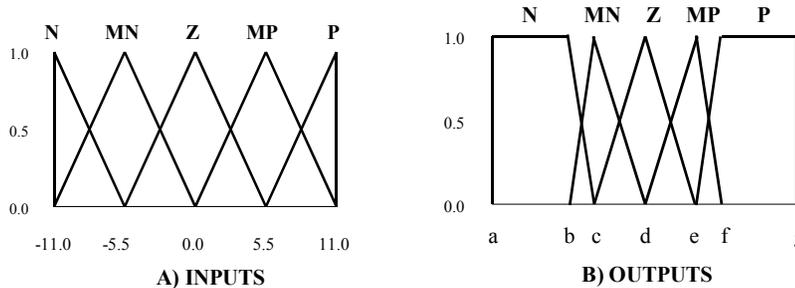


Figure 11: Fuzzy Sets Associated with Thrust Error and Thrust Error Rate Inputs and Fuzzy Sets associated with Valve Position and Valve Position Error Outputs

benefit from the capabilities of the fuzzy logic controller. Since the aim of the present study was to develop a test-bed and demonstrate the hardware-in-the-loop analysis, a relatively simple fuzzy logic design was used. Table 2 explains the meanings of the symbols to express each linguistic term. Table 3 shows a pair of the rule bases used in the design process.

Among the defuzzification methods such as the Center-of-Maximum, the Center-of-Gravity, and the Mean-of-Maximum that are available in the Fuzzy Logic Toolkit, the Center-of-Maximum method was used in the design. The Center-of-Maximum method results in a lesser computation time and it also results in

a linear variation of the output variable once fully overlapping linguistic terms were used.

IV-C. FUZZY LOGIC CONTROLLER TEST USING SIMULATED ENGINE

The LabVIEW-fuzzy logic controller design was first tested in a LabVIEW simulation model using the plant transfer function identified in the PID controller design process. It was assumed that the fuzzy logic control algorithm would be applicable to this engine control application because the architecture of the algorithm is capable of handling non-linearity. In the design of the fuzzy logic algorithm, several different universe of discourse ranges were tested for the

antecedence and the consequence ranges. The best ranges were identified which gave the least rise time without an overshoot.

Step Input Test: Figure 12 shows a unit step response from the simulation model, which includes the plant model and the fuzzy logic controller. The valve position very quickly rises to 3.6 degrees within 0.33 seconds. The simulated time delay is visible with about 0.2 seconds. The thrust output from the plant model increases without an overshoot to the desired thrust value within 3.0 seconds. The number below the horizontal axis in Figure 12 represents the number of loops executed and the rate of the loop execution was 25 time/sec.

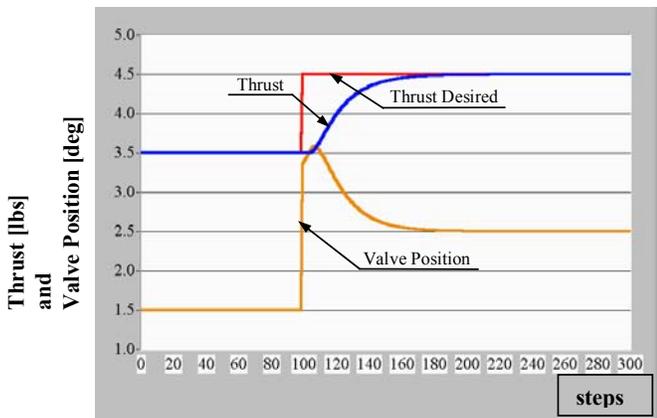


Figure 12: Step Response from Simulation Model. Twenty-five steps correspond to 1 second.

System Response with Limits on the Valve position: The fuzzy controller design was also tested to observe the effects on the system responses if valve position limits were set. The valve position limits were required because the controller tries to move the linkage out of the possible range of the motion. The reason of the

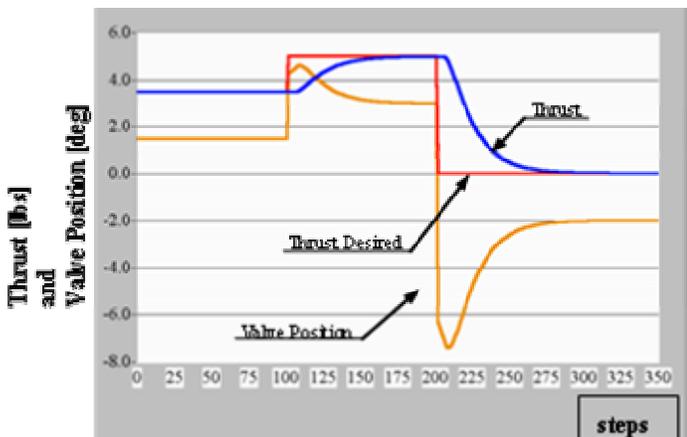


Figure 13: Negative Valve Position Angle Output from Fuzzy Inference System. Twenty-five steps correspond to 1 second.

necessity for the limits is shown in Figure 13. Once the desired thrust is obtained and subsequently decreased back to a lower value, the valve position needs to be reduced below zero degrees, which is not achievable due to the physical constraints of the fuel-flow valve system. Additionally, changing the valve position to below a certain angle would result in the engine to shut down. Although the effect of the minimum angle limit set results in a much slower response of the engine, it is required to avoid any damage to the engine. In order to eliminate such control scenarios occurring, limits on the integral controller was placed. The integrated control angle and the control angle sent to the engine were reset to zero once the controller required an angle below zero. The number below the horizontal axis in Figure 13 also represents the number of loops executed and the rate of the loop execution was 25 /sec.

IV-D. FUZZY LOGIC CONTROLLER TEST WITH SR-30 TURBOJET ENGINE

The Fuzzy Logic Controller LabVIEW-Virtual-Instrument was next tested to control the SR-30 engine. In order to compare the response of the simulated and the actual SR-30 engine responses, the same type of a step input test was attempted. The results are shown in Figure 14 (Case 1), Figure 15 (Case 2). Different ranges of the linguistic variable for the valve position output were used for each case while the same ranges of the linguistic variables for the inputs were used with the ones in the simulation. Refer to Figure 11 and Table 1 for the numbers used for each case.

A step command of 4.5 pounds was issued from the idling operation status for each case. Figure 14 shows the result from the first attempt and the same fuzzy inference system used in the test simulation was used for this case. There was an oscillation in the valve position trying to achieve the desired thrust number since the thrust was not getting a steady state. In Figure

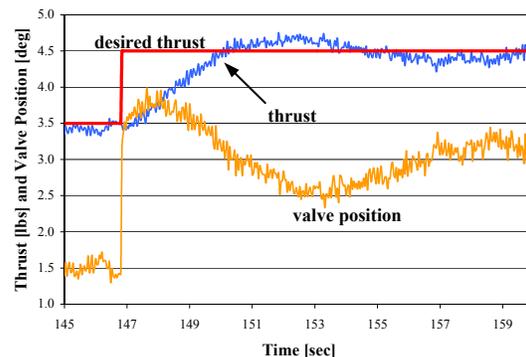


Figure 14: Step Response from SR-30 Engine Case 1

15, the result from the second attempt is shown. A steady state was achieved in about 5.0 seconds with a near no-steady error. Although the settling time was longer than the one observed in the simulation using the PID controller, the performance of the fuzzy controller for this case showed an improvement over the first case. Figure 15 shows a good agreement with the simulation result shown in Figure 12.

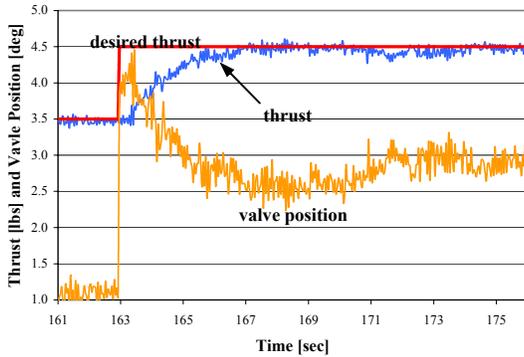


Figure 15: Step Response from SR-30 Engine Case 2

V. BAYESIAN BELIEF NETWORKS CONTROLLER DESIGN [2]

For the engine start phase, the primary soft computing technology to be utilized is Bayesian belief networks (BBN). The sole intent of the BBN is to qualify each of the states during engine start-up prior to reaching main-stage. This will further assure certainty in the health of the engine and proceeding into main-stage in addition to providing added assurance into preventing any premature engine shutdowns.

BBN's have been proven to be good predictive and diagnostic mechanisms for reasoning about the state of events in environments where uncertainty is universal. Suppressing the details, the genealogy of this SCT is strongly rooted in classic statistical Bayesian inference theory where a subjectivist viewpoint is taken. In short Bayesian inference uses a different interpretation of probability where one's degree of belief in some event is part of the reasoning. BBN's are computational architectures that permit declarative (prior conditional probabilistic values) and subjective opinions (posterior probabilistic values) about world (factual) knowledge to be part of the reasoning and assessment through a visual network representation and a unique syntactic message-passing feature. Furthermore, the visual network makes it easier to view the top-down cause and effect (or condition to consequence) relationships. For a more detailed account of this SCT, the reader is referred

to Pearl's treatment of the subject [4]. The specific BBN approach to be employed in this effort is shown in Figure 16 and its basic procedure is beyond the scope of this paper. The basic steps are listed below.

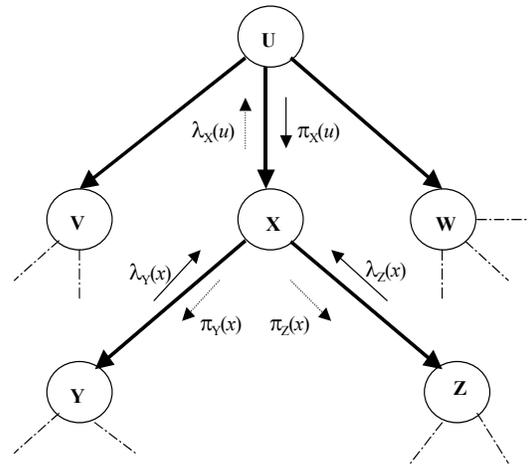


Figure 16 Bayesian Belief Network

1. **Belief Updating** – When node X is activated to update its parameters for belief updating, it first inspects all messages transmitted to it by its parent (π) and its children nodes (λ). Then using all input, it updates its belief.
2. **Bottom-up Propagation** – Using messages transmitted by Y&Z, compute message to transmit to parent node U.
3. **Top-down Propagation** – Node X then computes new messages to be sent to its children nodes Y&Z.

A first design of the Bayesian Belief Networks for the SR-30 Engine is presented in the Figure 17a and 17b. Table 4 gives the initial design for the BBN algorithm.

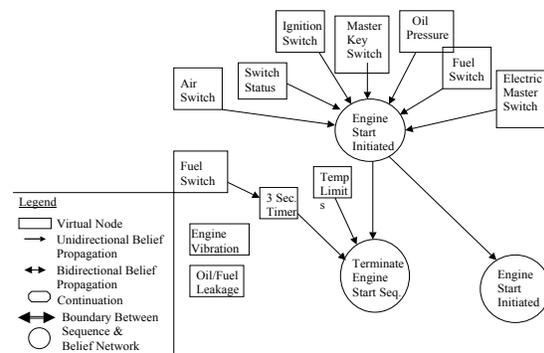


Figure 17a Bayesian Belief Networks for SR-30 Turbo Jet Engine Start-Up

The design required analysis of the engine-start data and careful choice of the parameters for the design. For

this purpose six sets of engine-start data were acquired and analyzed. This initial design will allow application of the BBN on the existing test bed.

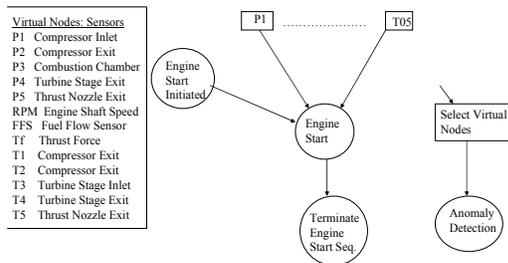


Figure 17b Bayesian Belief Networks for SR-30 Turbo Jet Engine Start-Up

VI. CONCLUSIONS

Development of a test-bed for soft-computing studies has been presented. Modifications on an existing SR-30 Turbine Technologies small-scale turbojet engine included the integration of a new data-acquisition board, replacement of the manual switches and manual lever with computer controlled switches and a stepper motor. The operation and control of the engine has been automated using a PC using National Instruments LabVIEW program.

Two different control algorithms have been designed, tested in a simulated environment and integrated into the test bed. Both the PID and the fuzzy-logic algorithms have been shown to successfully control the operation of the engine.

Additionally a design of the BBN algorithm is presented. The algorithm will be integrated into the system in a future study.

REFERENCES

- ¹ Watanabe, A., Davis, R. N., Ölçmen, S. M., Polites, M. E., and Trevino, L. C., "A Test bed for Evaluating Soft Computing Technologies for Rocket Engine Control." 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, January 05-08, 2004.
- ² Trevino, L. C., Ölçmen, S. M., and Polites, M. E., "Use of Soft Computing Technologies for Rocket Engine Control." 22nd Digital Avionics Systems Conference, Indianapolis, IN, October 12-16, 2003.
- ³ Steincamp, J., First Annual Report, Marshall Space Flight Center, Huntsville, AL, September 2001.
- ⁴ Pearl, J., "Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference," Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1988.
- ⁵ Zadeh, L. A., "Fuzzy Logic," Internal Report, University of California, Berkeley, CA, 1988.
- ⁶ Watanabe, A. "Development of a Test-Bed for Rocket Engine Control Algorithm Improvement." Masters Thesis, The University of Alabama, Tuscaloosa, AL, 2004.

Table 1: Linguistic Values in Figure 11

Case	Output	A	B	C	D	E	F	G
Simulation and Case 1 with SR-30	Valve position	-20.0	-12.5	-10.0	0.0	10.0	12.5	20.0
Case 2 with SR-30	Valve position	-30.0	-18.8	-15.0	0.0	15.0	18.8	30.0
Case 3 with SR-30	Valve position	-35.0	-21.9	-17.5	0.0	17.5	21.9	35
All	Valve position error	-1.6	-0.8	-0.2	0.0	0.2	0.8	1.6

Table 2: Linguistic Terms

N1I	thrust error is negative	N1O	valve position change will be negative
MN1I	thrust error is medium negative	MN1O	valve position change will be medium negative
Z1I	thrust error is near zero	Z1O	valve position change will be near zero
MP1I	thrust error is medium positive	MP1O	valve position change will be medium positive
SP1I	thrust error is positive	P1O	valve position change will be positive
N2I	thrust error rate is negative	N2O	adjustment for valve position error will be negative
MN2I	thrust error rate is medium negative	MN2O	adjustment for valve position error will be medium negative
Z2I	thrust error rate is near zero	Z2O	adjustment for valve position error will be near zero
MP2I	thrust error rate is medium positive	MP2O	adjustment for valve position error will be medium positive
P2I	thrust error rate is positive	P2O	adjustment for valve position error will be positive

Table 3: Rule Bases for Valve Position and Valve Position Error Outputs

	N1I	MN1I	Z1I	MP1I	P1I		N1I	MN2O
N2I	N1O	MN1O	MN1O	MN1O	Z1O		MN2O	
MN2I	N1O	MN1O	Z1O	Z1O	Z1O		MN2O	
Z2I	MN1O	MN1O	Z1O	MP1O	MP1O		Z2	
MP2I	Z1O	Z1O	Z1O	MP1O	P1O		MP2O	
P2I	Z1O	MP1O	MP1O	MP1O	P1O		MP2O	

Valve Position Output

Valve Position Error Output

Table 4: SR-30 Turbo Jet Engine Bayesian Design Parameters				
Link Matrices (M): All link matrices (conditional probabilities) are unity for convenience				
Master Key Switch	Prior: $\pi_{MKS} = [0.95 \quad 0.05]$	[Probability True	Probability False]	
Electric Master Switch	Prior: $\pi_{EMS} = [0.99 \quad 0.01]$			
Engine Start Initiated Start Node	Prior: $\pi_{ESI} = [0.99 \quad 0.01]$			
Engine Start Node	Prior: $\pi_{ES} = [0.99 \quad 0.01]$			
Temperature (T) Likelihood Probabilities T1 & T2 Disabled T3 Compressor Exit (Normalization Factor 1200) T4 Turbine Exit T5 Nozzle Exit e/s = engine start	$\lambda_{T3} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 5] = [0.15 \quad 0.85]$ $\lambda_{T4} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 5] = [0.15 \quad 0.85]$ $\lambda_{T5} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 5] = [0.10 \quad 0.9]$	Probability [Too High	Probability Too Low]	
Pressure Likelihood Probabilities P1 Compressor Inlet P2 Compressor Exit P3 Combustion Chamber P4 Turbine Stage Exit P5 Thrust Nozzle Exit	$\lambda_{p2} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.05 \quad 0.95]$ $\lambda_{p3} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.1 \quad 0.9]$ $\lambda_{p4} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.05 \quad 0.95]$ $\lambda_{p5} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.1 \quad 0.9]$	Probability [Too High	Probability Too Low]	(Note: > 0.65) (Note: > 0.625) (Note: > 1)
	$\lambda_{RPM} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.1 \quad 0.9]$ $\lambda_{Thrust} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.10 \quad 0.9]$ $\lambda_{Flow} [-\infty \rightarrow e/s] = [0 \quad 1]$ $[e/s \rightarrow 9] = [0.05 \quad 0.95]$			(Note: > 4800) (Note: > 3.5) (Note: > 1.7)