ORNL/TM-2025/43522

Digital Twin User Guide for Tacoma Public Utilities

Sunil Subedi
Hong Wang
Zhun Yin
Scott Warnick

Approved for public release.
Distribution is unlimited.

December 2025

ORNL IS MANAGED BY UT-BATTELLE LLC FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-2025/43522

Electrification and Energy Infrastructures Division
Buildings and Transportation Science Division

DIGITAL TWIN USER GUIDE FOR TACOMA PUBLIC UTILITIES

Sunil Subedi
Hong Wang
Zhun Yin
Scott Warnick

December 2025

Prepared by
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831
managed by
UT-BATTELLE LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

$\Delta\omega$	Speed deviation
$h(t)$	Water pressure (time-domain signal)
J	Moment of inertia
P_{gv}	Governor power
P_{mech}	Mechanical power
u	Guided vane opening (time-variant)
v	Excitation control input for generator
$x(t)$	Turbine shaft speed (time-domain signal)
AC	Alternating current
AVR	Automatic voltage regulator
CV	Control variable
DC	Direct current
DT	Digital twin
EMF	Electromotive force
GV	Governor valve position
PID	Proportional–Integral–Derivative
PIDD	Proportional–Integral–Double–Derivative
PLC	Programmable logic controller

ABSTRACT

This user manual offers a comprehensive guide for developing a Digital twin (DT) model of a Francis Turbine at the Tacoma Power plant using neural networks. The growing integration of variable renewable energy has necessitated that hydropower systems operate with greater adaptability. This shift in operational demand emphasizes the nonlinear behavior of the Francis Turbine, making traditional modeling difficult.

To overcome this, the manual details a data-driven modeling and learning algorithm centered on structured neural networks. This approach is designed to accurately forecast critical operational parameters, including turbine shaft speed, penstock pressure, and generator power output, by leveraging real-time inputs like generator power control setpoint, field current, and field voltage. The neural network's effectiveness is validated using authentic operational data from a Francis Turbine at the Tacoma Power (TPU) plant. The outcome is a robust modeling approach that captures the system's complex dynamics, offering insights that enhance the operational efficiency and decision-making for the Tacoma Power plant.

1. INTRODUCTION

With an average machine age exceeding 64 years, the U.S. hydropower fleet faces a pressing need for smart modernization. This modernization is essential for reducing costs and enhancing the reliability and value of the nation's longest-serving renewable energy technology. As hydropower increasingly plays a vital role in maintaining grid reliability and resiliency, operations are becoming more complex and demanding, particularly with the introduction of nonlinear dynamics. The continual expansion of variable renewable energy production, such as solar and wind, further intensifies pressure on the grid. To remain competitive, hydropower technology must integrate the latest advancements in sensors, data management, control systems, analytics, simulation, optimization, and computing capabilities [1].

In this context, digital twins (DTs) emerge as powerful tools that provide a virtual environment for engineers and operators to simulate and optimize performance. A DT is a sophisticated virtual replica or simulator of a physical object, system, or process, mirroring the real-world characteristics, behaviors, and performance of its physical counterpart. Serving as an effective platform for research and development (R&D) and operational management, a well-designed DT facilitates the exploration of best practices for process optimization and real-time monitoring. Creating an effective DT requires meticulous modeling of various components within the hydropower system, including the penstock, turbine, generator, and their interconnections with the broader power grid [2].

As power demand increases, hydropower systems must operate over a broader range, which can induce nonlinear characteristics in generation units. For instance, a Francis hydroturbine exhibits two primary nonlinearities: mechanical torque and water flow [1]. Both are influenced by factors such as water head, hydroturbine shaft speed, and guide vane opening. Fig. 1 shows a functional diagram of a hydropower generation system, in which the water flows from the reservoir flows a penstock to the hydroturbine, which then drives a synchronous generator to generator electricity. To understand the dynamics of the hydroturbine system, it is essential to estimate these nonlinear functions using real-time operational data obtained from the programmable logic controller in the distributed control system [3]. In this context, neural network modeling may be particularly well-suited for adaptive learning, enabling the effective capture of complex nonlinear dynamics [5]. These advanced algorithms enhance the accuracy and predictive capabilities of DT models by leveraging historical and real-time data. For instance, neural networks analyze data on water inflow, turbine performance, and energy production to identify patterns that forecast future behavior of the hydropower system. By providing valuable insights, this modeling strategy enables operators to anticipate changes, optimize energy production, and make informed decisions, ultimately enhancing operational efficiency and minimizing downtime [6].

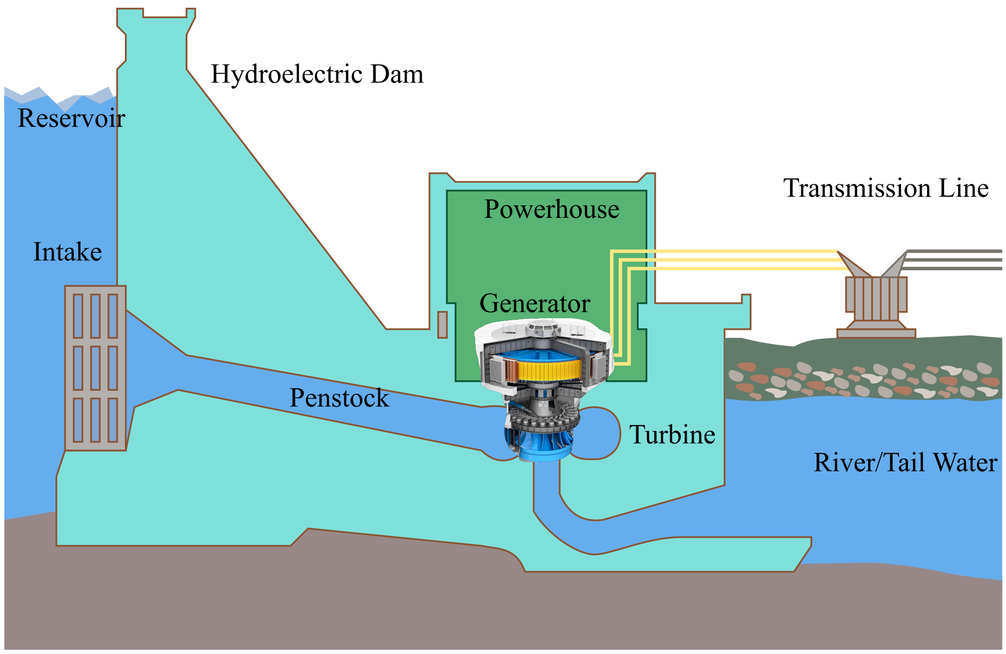

Figure 1. Functional diagram of a hydroelectric system with a turbine. The hydropower system, where the generator is driven by a turbine through a shaft whose speed is adjusted by the opening of a guide vane that controls the water flow from the reservoir through a penstock to the turbine [4].

2. COMPONENTS OF HYDROPOWER SYSTEMS

The operational system for hydropower generation units is comprised of several modules, which include penstock dynamics, a shaft speed sensor/controller, and a hydraulic servo. Fig. 2 shows a generic structure of a hydropower generation unit, in which the water from the reservoir flows through a penstock (pipe) to the hydroturbine, which then drives a synchronous generator to generate the electricity. The control of shaft speed and electrical power generation for grid integration is achieved by adjusting the guide vane, an inlet water valve that regulates the flow of water into the hydroturbine chamber. Thus, the guide vane opening serves as the control input, while shaft speed and generated power are the outputs transmitted to the power grid.

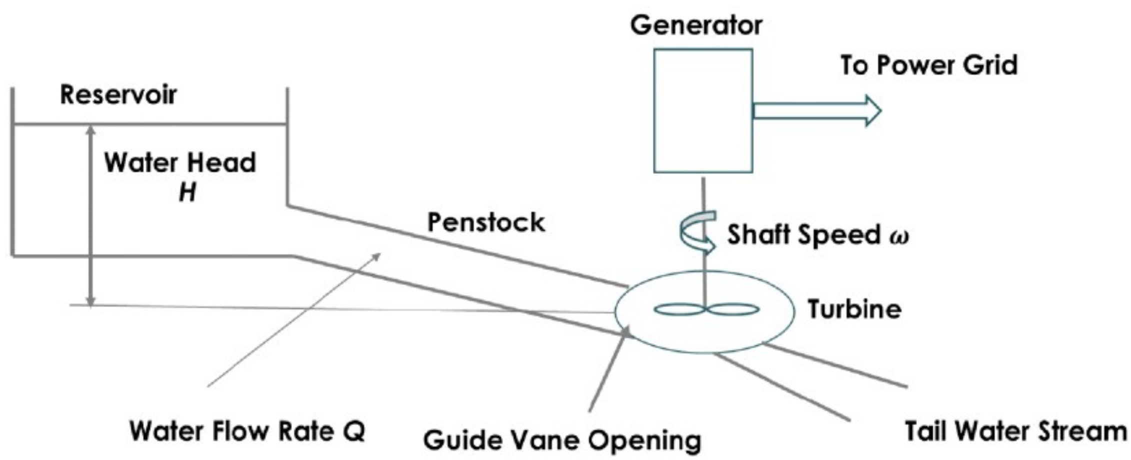

Figure 2. The hydropower system, where the generator is driven by a turbine through a shaft whose speed is adjusted by the opening of a guide vane that controls the water flow from the reservoir through a penstock to the turbine [1].

2.1 RESERVOIR

Acts as a water storage system, controlling flow and providing a supply of water for power generation. The reservoir can be formed by constructing a dam across a river or by utilizing existing natural lakes. The size and capacity of the reservoir are determined based on hydrological studies, ensuring it can store enough water to meet demand fluctuations. The water level in the reservoir affects the potential energy available for generation.

2.2 HYDROELECTRIC DAM

A barrier that impounds water to create the reservoir, controlling its release to optimize electricity production while managing flood risks.

2.3 PENSTOCK

The penstock is a critical component of a hydropower system, serving as a conduit for high-pressure water flow from the reservoir to the turbine. Understanding the dynamics of the penstock is essential for optimizing hydropower performance and ensuring system reliability. When evaluating the penstock dynamics in hydropower systems, depending on the length of pipe and water ahead, considering water elasticity and flow characteristics, is essential.

2.4 TURBINE

Converts the kinetic and potential energy of water into mechanical energy. Common types include Francis, Kaplan, and Pelton turbines, selected based on head conditions.

The Francis turbine operates with a combination of both kinetic and potential energy, using the water's pressure and flow velocity to drive the runner. Francis turbines are optimized for medium to high hydraulic heads, typically ranging from 30 to 600 meters. The higher the water head, the greater the potential energy available, which translates into higher water pressure.

2.5 SYSTEM DYNAMICS FOR NONELASTICITY WATER FLOW

For the Francis hydroturbine, under low water head and short penstock conditions, the water flow is considered nonelastic. Neglecting shaft rotational friction, the turbine system dynamics can be expressed as follows:

$$\begin{cases} J\dfrac{dx}{dt} = M(x, h, u) - \dfrac{P}{x} \\ h = \dfrac{dg(x, h, u)}{dt}, \end{cases}$$

where Moment of inertia (J), Turbine shaft speed (time-domain signal) ($x(t)$), Water pressure (time-domain signal) ($h(t)$), $u(t)$ is the guide vane opening, and $P(t)$ is the generated power. Here, P_x indicates load torque, and $M(x, h, u)$ is the mechanical torque from hydraulic flow. The term $g(x, h, u) = -T_W q(t)$ relates to the water flow rate through the turbine, with T_W representing the water starting constant influenced by penstock characteristics.

By dividing the initial equation by J and setting $\alpha = \frac{1}{J}$, we reformulate it as:

$$\begin{cases} \dfrac{dx}{dt} = f(x, h, u) - \alpha\dfrac{P}{x} \\ h = \dfrac{dg(x, h, u)}{dt}, \end{cases} \tag{2}$$

where $f(x, h, u) = M(x, h, u)J$. The dynamics of water head $h(t)$ can then be expressed using the chain rule:

$$h = \frac{dg(x, h, u)}{dt} = \frac{\partial g}{\partial x}\frac{dx}{dt} + \frac{\partial g}{\partial h}\frac{dh}{dt} + \frac{\partial g}{\partial u}\frac{du}{dt}. \tag{3}$$

Rearranging Eq. (3) for $\frac{dh}{dt}$ leads to:

$$\begin{cases} \dfrac{dx}{dt} = f(x, h, u) - \alpha\dfrac{P}{x} \\ \dfrac{dh}{dt} = \left(\dfrac{\partial g}{\partial h}\right)^{-1} \left\{ h - \dfrac{\partial g}{\partial x}\left[f(x, h, u) - \alpha\dfrac{P}{x}\right] - \dfrac{\partial g}{\partial u}\dfrac{du}{dt} \right\}. \end{cases} \tag{4}$$

The goal is to use measured values of shaft speed, water pressure, and guide vane opening to estimate $f(x, h, u)$ and $g(x, h, u)$. Given that $f(x, h, u)$ and $g(x, h, u)$ are continuous and bounded, they can be estimated using neural networks.

2.6 SYSTEMS WITH ELASTICITY WATER FLOW

When the water head is high and the penstock length is long, the water flow will become elastic. This elasticity means that the water is compressible. In this case, the fluid dynamics are much more complicated; in general, a full order Navier–Stokes equation is required to represent the water flow system—leading to partial differential equations that represent the time and space dynamics.

$$\begin{bmatrix} \frac{dh}{dt} \\ \frac{d^2h}{dt^2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & c\frac{\partial g}{\partial h} \end{bmatrix} \begin{bmatrix} h \\ \frac{dh}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ c\frac{\partial g}{\partial u} \end{bmatrix} \frac{du}{dt} + \begin{bmatrix} 0 \\ c\frac{\partial g}{\partial x} \end{bmatrix} \frac{dx}{dt}, \tag{7}$$

where $\beta = 1480\frac{m}{s}$ is the velocity of the sound traveling in the water, and T_e is the wave travel time determined by $\frac{L}{\beta}$, with L being the length of the penstock.

2.7 GENERATOR

Transforms mechanical energy from the turbine into electrical energy through electromagnetic induction, essential for power output. Most hydropower plants use synchronous generators, which operate at a constant speed depending on the frequency of the electrical grid. The rotor spins within a magnetic field produced by stator windings, inducing current in the coils.

The dynamic model of a synchronous generator, described by five differential equations in Equation (1) in the d-q reference frame, captures its electromechanical and electromagnetic behavior for transient stability analysis [**?**].

$$\begin{cases} \frac{d\delta}{dt} = \omega_0 \Delta\omega \\ \frac{d\omega}{dt} = \frac{1}{2M}(T_m - T_e - K_D \Delta\omega) \\ T'_{d0}\frac{dE'_q}{dt} = X_{ad}I_f - E'_q - I_d(X_d - X'_d) \\ T''_{d0}\frac{dE''_q}{dt} = E'_q - E''_q - I_d(X'_d - X'_{d'}) \\ T''_{q0}\frac{dE''_d}{dt} = -E''_d + I_q(X_q - X''_q) \end{cases} \tag{1}$$

In (1), the first term adjusts rotor angle (δ) via speed deviation ($\Delta\omega = \omega - \omega_0$) with ω_0 as synchronous frequency. The second term sets rotor speed (ω) using mechanical torque (T_m), electromagnetic torque (T_e), inertia (M), and damping (K_D). The third term updates q-axis transient voltage (E'_q) with field current (I_f), d-axis current (I_d), reactances (X_{ad}, X_d, X'_d), and time constant (T'_{d0}). The fourth term manages q-axis subtransient voltage (E''_q) with reactance (X''_d) and time constant (T''_{d0}). The fifth term controls d-axis subtransient voltage (E''_d) with q-axis current (I_q), reactances (X_q, X''_q), and time constant (T''_{q0}).

In the given equation, T_m is the mechanical torque from water flow to the turbine, $T_e = P/\omega$ is the load torque, K_D is the generator's damping ratio, and $\Delta\omega = \omega - \omega_0$ is the shaft speed deviation from its nominal value ω_0. Since the unit is grid-connected, $\Delta\omega$ is typically small and hence we can omit it and hence Equation 1 is simplified as:

$$\begin{cases} T'_{d0}\frac{dE'_q}{dt} = X_{ad}I_f - E'_q - I_d(X_d - X'_d) \\ T''_{d0}\frac{dE''_q}{dt} = E'_q - E''_q - I_d(X'_d - X''_d) \\ T''_{q0}\frac{dE''_d}{dt} = -E''_d + I_q(X_q - X''_q) \end{cases} \tag{2}$$

2.8 CONTROL SYSTEM

Manages plant operations, including flow monitoring and turbine adjustments, and encompasses sensors, Programmable logic controller (PLC), and analysis software. Two of the main control systems in the hydropower generation i.e., turbine and synchronous generator systems are valve opening for Guided vane opening (time-variant) (u) and excitation control input for Excitation control input for generator (v). u controls frequency and active power generation while v controls voltage and reactive power.

2.8.1 Excitation Systems

Excitation in a synchronous generator fundamentally involves supplying a precisely controlled Direct current (DC) to the rotor's field windings, thereby establishing a fundamental-frequency rotating magnetic field. This field, rotating synchronously with the prime mover, electromagnetically couples with the stationary stator windings, inducing an Electromotive force (EMF) whose magnitude is directly proportional to the field current and rotor speed. An Automatic voltage regulator (AVR) constitutes the core of the control mechanism, continuously sensing the generator's terminal voltage and comparing it to a reference. Any detected voltage deviation generates an error signal, which the AVR processes via sophisticated control algorithms (e.g., PI, lead-lag compensation) to adjust the exciter's output. The exciter, whether a static rectifier, brushless Alternating current (AC) exciter with rotating rectifiers, or a DC exciter, then modulates the DC current to the main generator's field, closing the feedback loop to maintain target terminal voltage and contribute to reactive power control and overall system stability.

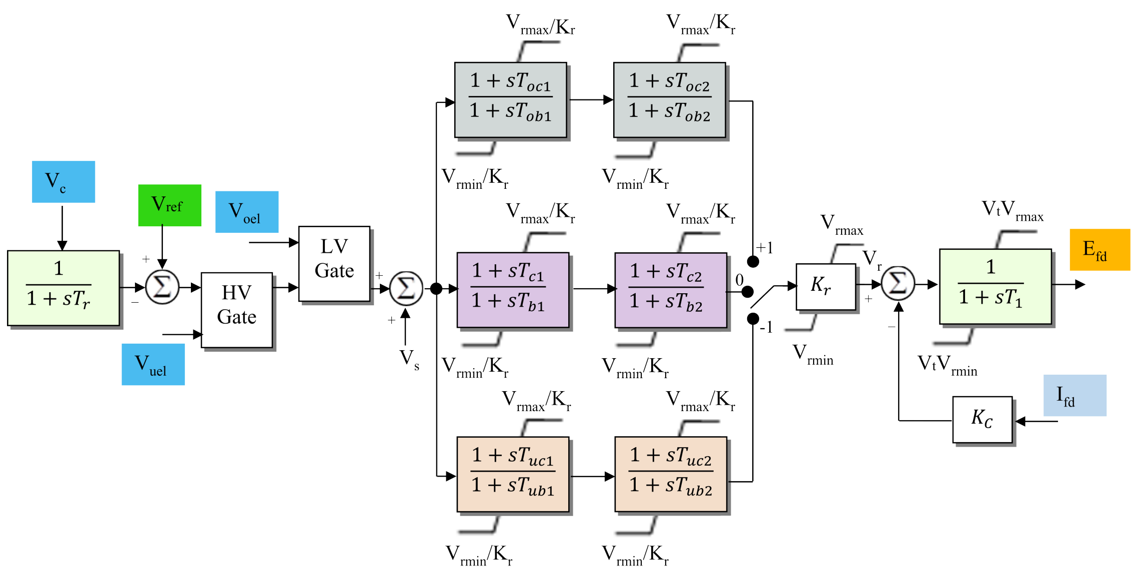

Figure 3. Block diagram of PSLF excitation system model ESST5B [7].

2.8.2 PIDD Controller

Fig. 4 illustrates a sophisticated control system, centered on the Proportional–Integral–Double–Derivative (PIDD) controller, an advanced variant of the standard Proportional–Integral–Derivative (PID). This controller is used by many hydropower systems due to its enhanced transient response and oscillation damping.

PIDD controller processes the Speed deviation ($\Delta\omega$) to produce a precise Control variable (CV). Its output sums proportional , integral, primary derivative, and a critical second derivative term.

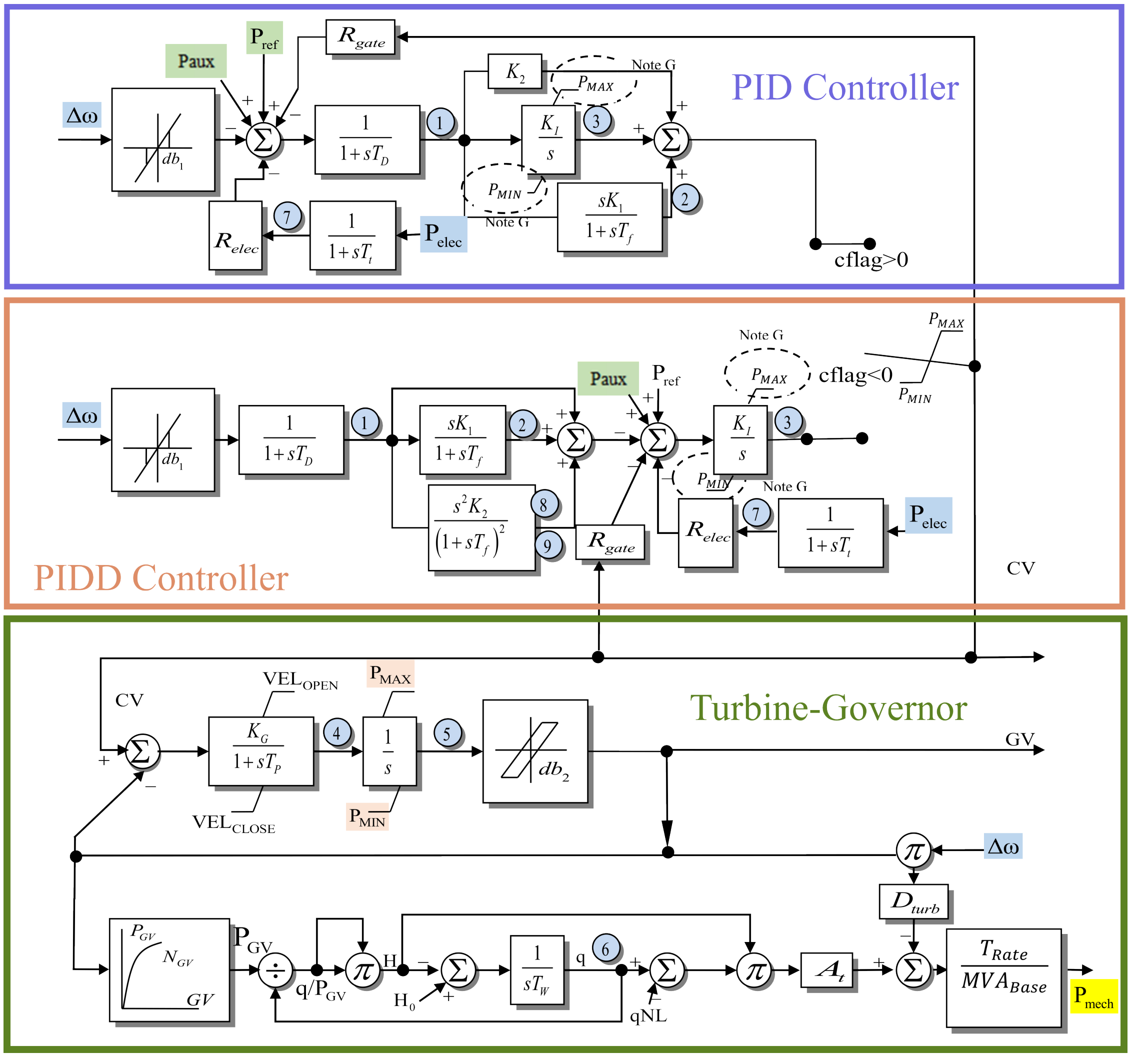

Figure 4. Block diagram of PIDD controller [7].

Following the PIDD controller, the output (CV) is fed into the combined governor-turbine block which translate the control variable into a physical Governor valve position (GV) and commanded Governor power (P_{gv}). This output then fed into turbine model converting the fluid energy into the actual Mechanical power (P_{mech}) delivered to the generator shaft.

2.9 TAILRACE

A channel or pipe for discharging water downstream after it passes through the turbine, helping to maintain the ecological flow of the river.

2.10 ELECTRICAL INFRASTRUCTURE

Comprises transformers, switchgear, and transmission lines, facilitating the integration of generated power into the electrical grid. The electrical infrastructure must comply with standards to ensure safety and reliability, addressing factors such as load flow analysis and short circuit calculations.

2.11 FISH LADDERS/BYPASS SYSTEMS

Facilitate the migration of aquatic life around or past the dam, mitigating ecological impacts.

3. UNDERSTANDING THE DIGITAL TWINS

DT concept was initially introduced in 2003 and the researches rapid growth in recent years which cover a lot of areas and industry fields. DT is defined as a virtual replica of a physical object, system or process that serves as a digital counterpart. Fig. 5 illustrates the essential components of a DT for hydropower systems—an open platform framework. This begins with real data from a reference hydropower unit/plant, which is processed through a central data processing module. This data feeds into a system of dynamic models, including penstock dynamics, hydro turbine dynamics, generator dynamics, and tail water dynamics, which collectively form the core of the DT operation and control. The processed data and model outputs are then can be interfaced with the user's visualization tools, enabling interaction and analysis. Simultaneously, the DT can also interacts with a power rid module, incorporating different load modes and forecasting, to optimize operation and control The DT captures the plant's operations in a digital format,

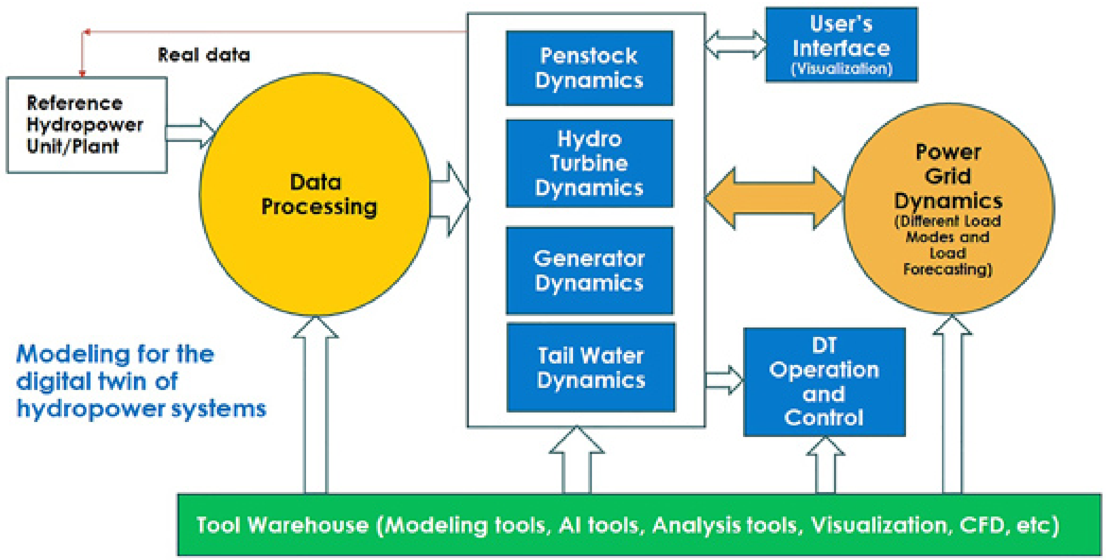

Figure 5. The flow diagram shows the overview of proposed structure for the modeling scope of a DT for hydropower systems. This requires the modeling of the penstock system, the hydro-turbine, the generator and the tail water system and thus motivates the novel hybrid modeling of hydro-turbine dynamics using a neural network [1].

supporting operational optimization, condition monitoring, and workforce training. In order to develop an effective DT, dynamic modeling of the entire system is crucial, encompassing the penstock, turbine, generator, and power grid connections. To achieve accurate replication of the hydropower system's dynamics, adaptive learning techniques should be employed for online training of the dynamic models, enabling them to reliably mirror real-time behavior.

3.1 TURBINE SYSTEM MODEL

The water system for electricity generation must account for the reservoir, penstock, turbine chamber, and discharge (tail water stream), forming a complex nonlinear hydropower dynamic system. This system requires data-driven modeling along with initial manufacturing data from turbine manufacturers. Utilizing hydropower turbine-generator experimental data allows us to create a systematic model for simulating the hydropower generator system in digital twin development.

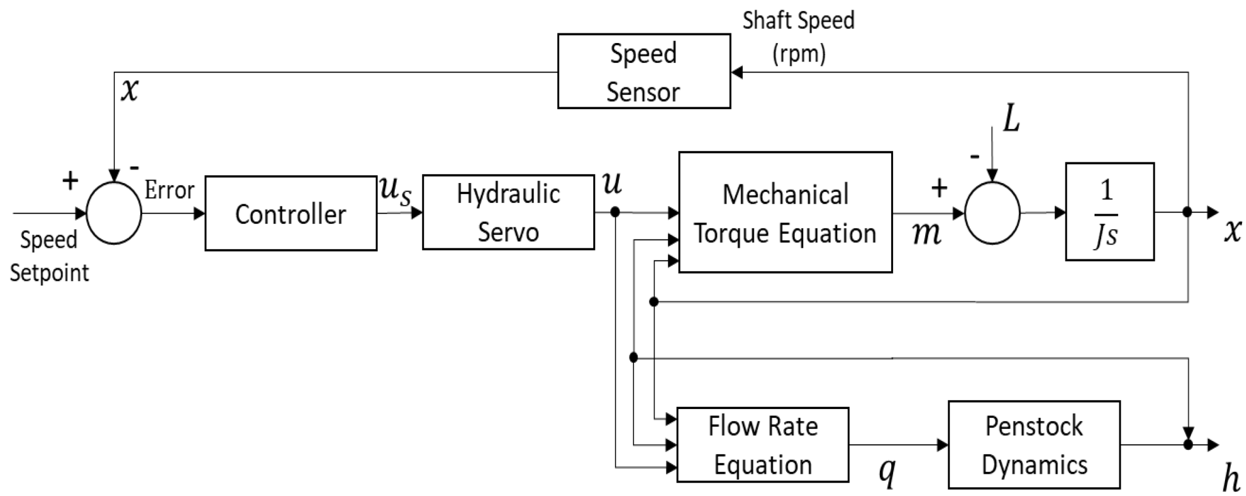

Figure 6. Control flow of the hydropower turbine control system.

3.1.1 Turbine Speed Control System

Fig. 6 depicts the control structure for turbine speed, which comprises a closed-loop system involving the dynamics of the turbine, hydraulic servo, and controller. The guide vane opening serves as the input, controlling the water flow into the turbine, while the outputs are the shaft speed (frequency) and water head. The objective of this closed-loop control is to ensure the shaft speed remains at the desired level amidst load changes (denoted by L) when connected to the power grid, while also minimizing fluctuations in water head.

3.1.2 Torque and Water Flow Module

Figure 2 illustrates the torque and water flow rate modules for the hydropower turbine during operation. The dynamics of the system indicate that the inputs are the guide vane opening, shaft speed, and water head, while the outputs are the turbine torque M and water flow rate Q. The following nonlinear functions are typically defined:

$$Q = g(H, \omega, u)$$

$$M = f(H, \omega, u)$$

where ω represents the turbine speed (rad/s), related to frequency by $\omega = 2\pi f_0$, with f_0 maintained around 60 Hz for the US system when the hydropower unit is connected to the grid. In Equations (1) and (2), H is the water head, and u is the guide vane opening. The functions g and f are unknown nonlinear functions that must be learned using real-time operational data.

DT is created based on the assumption that the turbine operates at a fixed point $O = \{\omega_0, H_0, Q_0, u_0\}$. The normalized incremental values for the turbine shaft speed, water head, water flow rate, guide vane opening, and torque are defined as follows:

$$x = \frac{\omega - \omega_0}{\omega_0}; \quad h = \frac{H - H_0}{H_0}; \quad q = \frac{Q - Q_0}{Q_0}; \quad \Delta u = \frac{u - u_0}{u_0}; \quad m = \frac{M - M_0}{M_0}, \tag{3}$$

where M_0 is the operating torque maintaining the fixed operating point O. Since x, h, q, m, and Δu are normalized incremental variables, they are unitless.

For the Francis hydroturbine, when the water head is low and the length of the penstock is short, the water flow is nonelastic. In this case, under the assumption that the shaft rotational friction can be ignored, the turbine system's dynamics can be expressed as follows:

$$\begin{cases} \dfrac{dx}{dt} = f(x,h,u) - \alpha_1 \dfrac{P}{x} \\ h = \dfrac{dg(x,h,u)}{dt}, \end{cases} \tag{3}$$

From above Eqs., according to the chain rule, the dynamics of the water head $h(t)$ can be expressed as:

$$h = \frac{dg(x,h,u)}{dt} = \frac{\partial g}{\partial x}\frac{dx}{dt} + \frac{\partial g}{\partial h}\frac{dh}{dt} + \frac{\partial g}{\partial u}\frac{du}{dt}, \tag{3}$$

$$\frac{dh}{dt} = \left(\frac{\partial g}{\partial h}\right)^{-1} \left\{ h - \frac{\partial g}{\partial x}\left[f(x,h,u) - \alpha_1 \frac{P}{x}\right] - \frac{\partial g}{\partial u}\frac{du}{dt} \right\}, \tag{4}$$

where we use the partial derivative for g since it is a function of more than one variables. More specifically, it is a function of shaft speed x , water pressure h , and guide vane opening u ; and we use ordinary derivative for $x(t), h(t)$ and $u(t)$ as they are set to be just the functions of time t . Then, by rearranging:

$$\begin{cases} \dfrac{dx}{dt} = f(x,h,u) - \alpha_1 \dfrac{P}{x} \\ \dfrac{dh}{dt} = \left(\dfrac{\partial g}{\partial h}\right)^{-1} \left\{ h - \dfrac{\partial g}{\partial x}\left[f(x,h,u) - \alpha_1 \dfrac{P}{x}\right] - \dfrac{\partial g}{\partial u}\dfrac{du}{dt} \right\}. \end{cases} \tag{5}$$

where:

- $x(t)$: turbine speed (state variable),
- $h(t)$: water pressure (state variable),
- $u(t)$: gate opening (input),
- $P(t)$: power generated (disturbance),
- J : inertia
- $\alpha_1 = \frac{1}{J}$

When the water head is high and the penstock length is long, the water flow will become elastic. This elasticity means that the water is compressible. In this case, the fluid dynamics are much more complicated. Due to the elasticity case, the system dynamic order will increased to the third order.

$$\begin{bmatrix} \dfrac{dh}{dt} \\ \dfrac{d^2h}{dt^2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -c & c\frac{\partial g}{\partial h} \end{bmatrix} \begin{bmatrix} h \\ \dfrac{dh}{dt} \end{bmatrix} + \begin{bmatrix} 0 \\ c\dfrac{\partial g}{\partial u} \end{bmatrix} \frac{du}{dt} + \begin{bmatrix} 0 \\ c\dfrac{\partial g}{\partial x} \end{bmatrix} \frac{dx}{dt} \tag{6}$$

where $\beta \ = \ 1480\, ms^{-1}$ is the velocity of the sound traveling in the water, and T_e is the wave travel time determined by $\frac{Ł}{\beta}$, with L being the length of the penstock and $c = \frac{1}{\beta T_e^2}$.

As a result, in any cases the objective of learning is to use the measured shaft speed, the water pressure, and the guide vane opening to estimate the mechanical torque dynamics $f(x, h, u)$ and the water flow dynamics $g(x, h, u)$.

The turbine model uses past values of power and control input to predict the current output power. The neural network learns the nonlinear function $\hat{P}(k) = g_\phi(\cdot)$ as follows:

$$\hat{P}(k) = g_\phi\left(P(k-1), P(k-2), u(k-1), u(k-2)\right) \tag{7}$$

3.2 GENERATOR SYSTEM MODEL

The following generator model is modeled using NN.

$$\begin{cases} T'_{d0}\frac{dE'_q}{dt} = X_{ad}I_f - E'_q - I_d(X_d - X'_d) \\ T''_{d0}\frac{dE''_q}{dt} = E'_q - E''_q - I_d(X'_d - X''_d) \\ T''_{q0}\frac{dE''_d}{dt} = -E''_d + I_q(X_q - X''_q) \end{cases} \tag{8}$$

The generator model is more complex and estimates $\hat{Q}(k)$ and $\hat{V}_t(k)$ using a combination of recent and delayed values, expressed as a sum of two neural network outputs:

$$\begin{aligned} \hat{Q}(k), \hat{V}_t(k) = f_{\theta_1}(&P(k-1), V_f(k), V_f(k-1), V_f(k-2), \\ &V_t(k-1), V_t(k-2), V_t(k-3), Q(k-1), Q(k-2), Q(k-3), I_f(k)) \\ &+ f_{\theta_2}(P(k-1), V_f(k), V_f(k-1), V_f(k-2), I_f(k)) \end{aligned} \tag{9}$$

$$\begin{aligned} \Delta\hat{Q}(k) = f_{\theta_3}(&P(k-1), V_f(k), V_f(k-1), V_f(k-2), \\ &V_t(k-1), V_t(k-2), V_t(k-3), Q(k-1), Q(k-2), Q(k-3), I_f(k)) \end{aligned} \tag{10}$$

$$\tilde{Q}(k) = Q(k) - \Delta\hat{Q}(k) \tag{11}$$

$$\text{Loss} = \text{MSE}((\hat{Q}(k), \hat{V}_t(k)),\ (\tilde{Q}(k), V_t(k))) \tag{12}$$

The model then incorporates an iterative update using the mean square error and the difference $\Delta\hat{Q}(k)$ as $\Delta\hat{Q}(k)$ is computed based on the mean square error to adjust the output iteratively through the feedback loop and added to $\hat{Q}(k)$ to refine the estimate (i.e., $\hat{Q}(k) = \hat{Q}(k) + \Delta\hat{Q}(k)$).

where:

- $V_f(k)$: Field voltage,
- $V_t(k)$: Terminal voltage,
- $Q(k)$: Reactive power,
- $I_f(k)$: Field current,
- $\theta_1, \theta_2, \theta_3$: Parameters of the respective neural networks.

4. DATA REQUIREMENTS

Instead of constructing physical modeling for simulating the turbine and generator system, this study simplifies the hydropower system and solved through the neural network modeling. Thus, it is important to obtain accurate real-world hydropower data as much as possible for the neural network model's training and validation to predict and analyze system behavior. In specific, this project conducts neural network modeling with the data collected from the Unit 12 Hydropower generator owned and operated by Tacoma Public Utilities.

4.1 DATA SOURCES AND SIGNALS

The dataset comprises measurements recorded by the plant's supervisory control and data acquisition (SCADA) system and other process instrumentation.

Table 1. Signals and descriptions (Unit U12)

Signal	Descriptions
ECC_MW_CTRL_REQ	Generator Power Control Setpoint
GEN_VOLT_L3L1	Generator Voltage L3L1 Direction
GEN_MW	Generator Power Output
GOV_SPD_RPM_FB	Turbine Shaft Speed
EXC_FLD_VOLT	Generator Field Voltage
GEN_VOLT_L1L2	Generator Voltage L1L2 Direction
GEN_VOLT_L2L3	Generator Voltage L2L3 Direction
TRB_PEN_PRES	Penstock Pressure
EXC_FLD_AMP	Generator Field Current
GOV_GATE_POS	Turbine Guided Vane Opening Position

These signals capture the system's dynamic response to control actions and external disturbances. Fig. 7, 8, and 9 shows the data format that is required to train the NN model.

4.2 DATA PROCESSING

To prepare the data for modeling, several preprocessing steps were applied:

- **Downsampling:** Reducing the data frequency by selecting every *n*th sample to balance temporal resolution and computational efficiency.
- **Truncation:** Limiting the dataset to a maximum number of samples to ensure manageable dataset size.
- **Normalization and scaling:** Centering variables around nominal operating points and scaling by suitable factors to improve training stability and convergence.

4.3 DATASET PREPARATION

The dataset was divided into training and testing subsets, typically using a test ratio (e.g., 20%) to reserve a portion of the data for model validation. Time-series lag features were created to capture temporal dependencies, using historical values of input and output variables (e.g., values at $t-1$, $t-2$, etc.). This approach helps the model learn dynamic relationships and improve prediction accuracy.

	Time	u12-bkr-close-stat	u12-bkr-status	u12-bkr-trip	u12-ecc-mw-ctrl-req
0	2023-02-13 00:00:00	1	0	0	9.002
1	2023-02-13 00:00:01	1	0	0	9.002
2	2023-02-13 00:00:02	1	0	0	9.002
3	2023-02-13 00:00:03	1	0	0	9.002
4	2023-02-13 00:00:04	1	0	0	9.002

	Time	u12-exc-fld-amp	u12-exc-fld-volt	u12-exc-gen-voltc	u12-exc-volt-sp-s.
0	2023-02-13 00:00:00	247.2	125.5	13.6897	13.74
1	2023-02-13 00:00:01	246	118.7	13.6898	13.74
2	2023-02-13 00:00:02	244.4	100.9	13.69	13.74
3	2023-02-13 00:00:03	250.4	115.8	13.69	13.74
4	2023-02-13 00:00:04	246.2	119.3	13.69	13.74

	Time	u12-gen-kv	alder-u12-gen-mvar	u12-gen-mw	u12-gen-ph3-amp
0	2023-02-13 00:00:00	13.7123	1.279	9.121	384.942
1	2023-02-13 00:00:01	13.7237	1.3205	9.046	384.905
2	2023-02-13 00:00:02	13.7223	1.362	9.066	384.868
3	2023-02-13 00:00:03	13.7223	1.43	8.989	384.831
4	2023-02-13 00:00:04	13.7297	1.42567	9.018	384.795

	Time	u12-gen-volt-l1l2	u12-gen-volt-l3l1	u12-gov-gate-pos	u12-gov-mw
0	2023-02-13 00:00:00	13.68	13.729	37.9386	8.96302
1	2023-02-13 00:00:01	13.693	13.739	37.9383	8.96438
2	2023-02-13 00:00:02	13.693	13.737	37.9385	8.96575
3	2023-02-13 00:00:03	13.693	13.737	37.9378	8.96711
4	2023-02-13 00:00:04	13.691	13.737	37.9389	8.96848

	Time	u12-gov-spd-rpm	u12-gov-started	u12-gov-status	u12-head
0	2023-02-13 00:00:00	225.085	1	1	246.147
1	2023-02-13 00:00:01	225.086	1	1	246.147
2	2023-02-13 00:00:02	225.087	1	1	246.147
3	2023-02-13 00:00:03	225.088	1	1	246.147
4	2023-02-13 00:00:04	225.089	1	1	246.147

	Time	u12-mw-ctrl-reg-limit	u12-mw-ctrl-sp	u12-trb-pen-pres	u12-var-ctrl-pide
0	2023-02-13 00:00:00	22.1528	9.002	97.1688	1
1	2023-02-13 00:00:01	22.1529	9.002	97.1688	1
2	2023-02-13 00:00:02	22.1529	9.002	97.1688	1
3	2023-02-13 00:00:03	22.1529	9.002	97.1688	1
4	2023-02-13 00:00:04	22.1529	9.002	97.1688	1

Figure 7. Input data sample collected from unit 12 of Tacoma public utility hydropower.

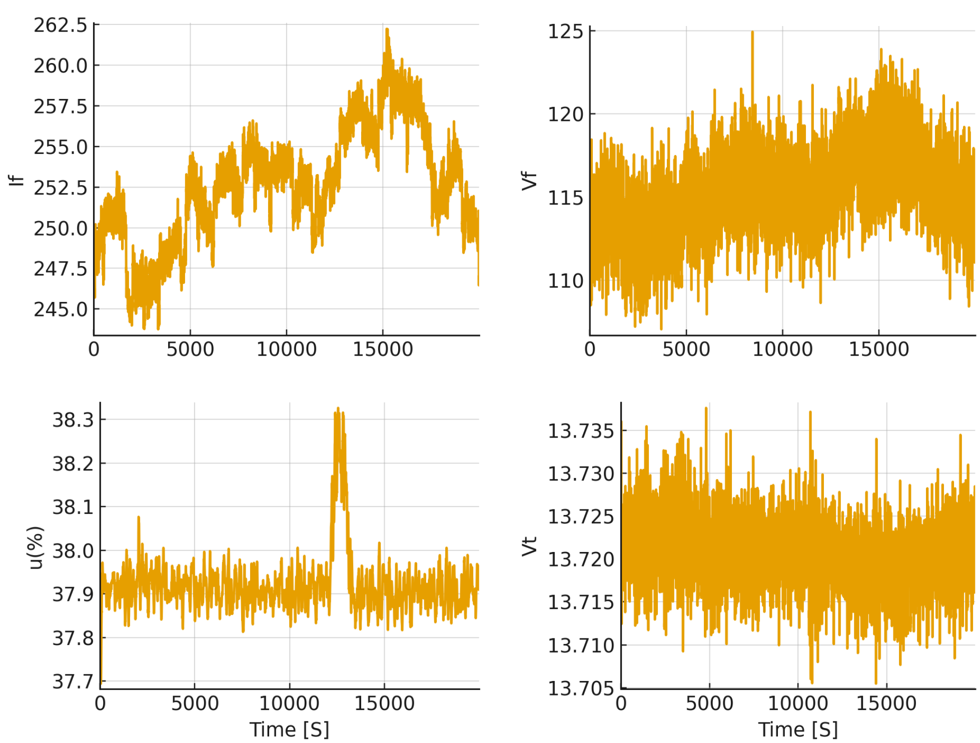

Figure 8. Input data sample: Field current, voltage, gate opening, and terminal voltage.

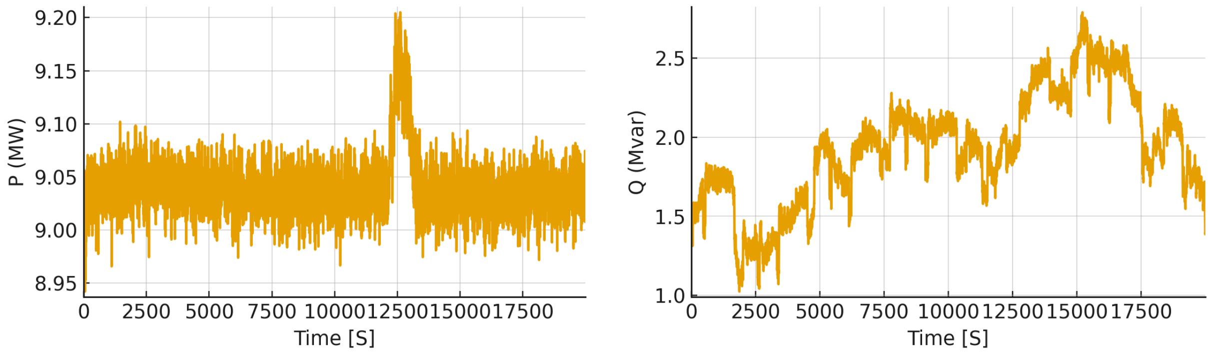

Figure 9. Input data sample: Active and reactive power.

4.4 DATASET SUMMARY

The processed datasets were combined to create comprehensive training and testing datasets. A suitable batch size was chosen for model training to balance convergence speed with available computational resources. This structured data preparation ensures that the model can capture the underlying dynamics and produce reliable predictions during deployment.

5. MODELING TECHNIQUES

As power demand increases, hydropower systems must operate over a broader range, which can induce nonlinear characteristics in generation units and hence, it is essential to estimate these nonlinear functions using real-time operational data obtained from the programmable logic controller in the distributed control system. In this context, neural network modeling may be particularly well-suited for adaptive learning, enabling the effective capture of complex nonlinear dynamics. These advanced algorithms enhance the accuracy and predictive capabilities of DT models by leveraging historical and real-time data.

5.1 TYPES OF MODELING

Various modeling techniques can be applied to turbine-generator dynamics, each suited to different scenarios. This section explores these types, with a focus on neural network models and the rationale for selecting the Multi-Layer Perceptron (MLP) in the current implementation.

5.1.1 Overview of Modeling Types

- **Physical Modeling**:
  - Based on differential equations from physical laws, e.g., the swing equation: $M\frac{d^2\delta}{dt^2} = P_m - P_e - D\frac{d\delta}{dt}$, where δ is the rotor angle, P_m is mechanical power, P_e is electrical power, M is inertia, and D is damping.
  - Advantages: High accuracy for known systems, interpretable.
  - Challenges: Requires detailed system data, computationally intensive.
- **State-Space Modeling**:
  - Uses a set of first-order differential equations: $\dot{x} = Ax + Bu$, $y = Cx + Du$, where x is the state vector, u is the input, and y is the output.
  - Advantages: Suitable for control design, handles MIMO systems.
  - Challenges: Assumes linearity, needs accurate state estimation.
- **Reinforcement Learning (RL)**:
  - Employs agents to learn optimal policies via trial and error, optimizing a reward (e.g., maximizing power while stabilizing voltage).
  - Advantages: Adapts to changes, no explicit model needed.
  - Challenges: Requires extensive data, computationally expensive.
- **Neural Network Modeling**:
  - Utilizes data-driven approaches to learn complex, non-linear relationships.
  - Discussed in detail below due to its use in the current implementation.

5.1.2 Neural Network Models

Within the neural network paradigm, several architectures can model turbine-generator dynamics:

- **Recurrent Neural Network (RNN)**:
  - Includes memory through recurrent connections, suitable for time-series data like turbine-generator outputs.
  - Strengths: Captures temporal dependencies directly.
  - Limitations: Prone to vanishing gradients, less effective for long sequences.
- **Long Short-Term Memory (LSTM)**:
  - An advanced RNN variant with memory cells, ideal for long-term dependencies in power system data.
  - Strengths: Better gradient flow, handles delayed inputs well.
  - Limitations: Higher computational cost, complex training.
- **Convolutional Neural Network (CNN)**:
  - Uses convolutional layers to extract spatial features, adaptable for time-series with 1D convolutions.
  - Strengths: Efficient feature extraction, parallelizable.
  - Limitations: Less suited for non-spatial turbine-generator data.
- **Multi-Layer Perceptron (MLP)**:
  - A feedforward network with fully connected layers, used here with architectures like turbine NN, generator NN, and Q refining NN model.
  - Strengths: Effective for static mappings, handles non-linearities with historical data.
  - Limitations: Limited in capturing temporal dynamics without additional preprocessing.

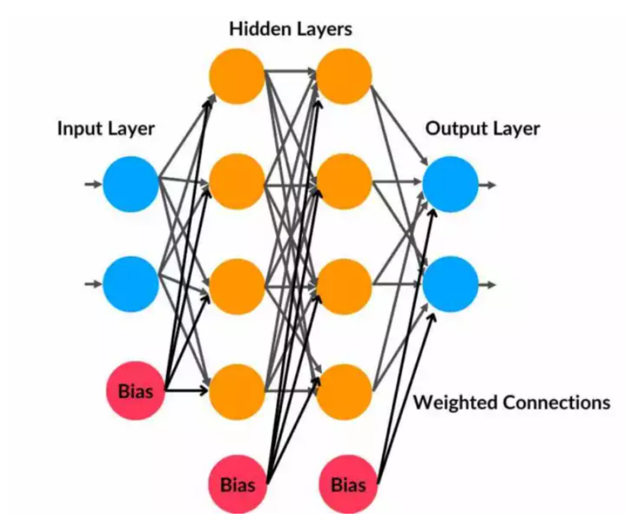

Figure 10. Architecture of MLP.

This research utilizes a Multilayer Perceptron (MLP), a foundational feedforward neural network characterized by its layered architecture (input, hidden, output) of fully connected neurons. As shown in Fig. 10, an MLP's structure is defined by its Layers: an input layer receives data, one or more hidden layers perform complex transformations, and an output layer delivers the final prediction. Within these layers, neurons utilize Activation Functions (like ReLU, ELU, or Sigmoid) to introduce non-linearity, enabling the network to learn intricate relationships. The network undergoes Learning by iteratively adjusting its internal parameters (weights and biases). This process often involves minimizing a Mean Squared Error (MSE) loss function, where the network's predictions are compared to actual targets, and the difference is minimized through optimization algorithms [**?**]. During operation, MLPs make predictions via forward propagation, where input data flows through the layers. Learning occurs through backpropagation, an iterative process that computes error gradients from predicted versus actual outputs, which an optimizer then uses to adjust the network's internal weights and biases, minimizing prediction errors.

5.2 FACTORS INFLUENCING NEURAL NETWORK DESIGN

The design should consider the following system-specific and data-specific attributes:

- **System Type:** Static (algebraic) vs. dynamic (time-dependent).
- **Order of Dynamics:** Number of significant past time steps influencing current behavior.
- **System Nonlinearity:** Degree of deviation from linear models.
- **Data Availability:** Volume, resolution, and quality of historical input/output data.
- **Computational Constraints:** Real-time vs. offline applications.

5.3 SELECTING NEURAL NETWORK TYPE

- **Feedforward Neural Network (FNN) / MLP:**
  - Suitable for static systems or when using pre-structured past input/output as features.
  - Requires manual inclusion of lag terms.
- **Recurrent Neural Network (RNN):**
  - Captures dynamic, time-sequential behaviors.
  - Better suited for systems where temporal memory is important (e.g., generator voltage and current response).
- **Long Short-Term Memory (LSTM):**
  - A type of RNN ideal for systems with long-term dependencies.
  - Recommended when system response depends on long sequences of past events.
- **NARX Networks (Nonlinear AutoRegressive with eXogenous inputs):**
  - Designed for dynamic systems using delayed inputs and outputs explicitly.
  - Well-suited when delays are known and bounded.

5.4 MODEL ORDER SELECTION

Model order defines how many past time steps (lags) are included as inputs:

- Use **system identification** or autocorrelation/partial correlation plots to determine relevant lags.
- For turbines and generators, orders between 2–5 are typical, depending on response time and sampling rate.
- Rule of thumb: include sufficient lags to cover at least 2–3 dominant time constants of the system.

5.5 NETWORK DEPTH AND WIDTH

- **Number of Layers (Depth):**
  - 1–2 hidden layers are sufficient for most physical system modeling.
  - Use 3+ layers only for highly nonlinear or composite systems.
- **Number of Neurons per Layer (Width):**
  - Start with 2–4 times the number of input features in the first layer.
  - Reduce neurons in deeper layers (pyramid structure) to avoid overfitting.
- Use grid search or Bayesian optimization to fine-tune architecture.

5.6 ACTIVATION FUNCTION SELECTIONS

- **ReLU (Rectified Linear Unit):** Preferred in deep networks to mitigate vanishing gradients.
- **tanh:** Useful for modeling both positive and negative values (e.g., voltage deviations).
- **sigmoid:** Rarely used due to saturation and slow convergence.
- **Linear:** Use in output layer for regression tasks.

5.7 REGULARIZATION AND OPTIMIZATION

- **Dropout:** Randomly disable neurons during training to prevent overfitting.
- **L2 regularization (weight decay):** Penalizes large weights.
- **Batch normalization:** Stabilizes learning.
- **Optimizer:** Use Adam or RMSProp for efficient convergence.
- **Learning rate schedule:** Reduce learning rate gradually during training.

5.8 EVALUATION AND VALIDATION

- Use **Mean Square Error (MSE)** or **Mean Absolute Error (MAE)** for regression.
- Perform **time-series cross-validation** or rolling window validation.
- Visualize predicted vs. actual outputs to verify accuracy over time.

Aspect	Recommendation
NN Type	FNN/MLP for static; RNN/LSTM/NARX for dynamic systems
Order	2–5 lags, based on system time constants
Hidden Layers	1–3 (more for high nonlinearity)
Neurons/Layer	2–4× number of inputs, then reduce in deeper layers
Activations	ReLU/tanh (hidden), Linear (output)
Optimizer	Adam or RMSProp
Regularization	Dropout, L2
Evaluation	MSE, MAE, time-series validation

5.9 CHOOSING THE RIGHT NEURAL NETWORK MODEL

The selection of the MLP for this turbine-generator modeling task was guided by several considerations:

- **Data Characteristics**: Availability of preprocessed historical data (e.g., P_{last}, $V_{t,\text{last}}$, Q_{last}) with up to three time delays allowed the MLP to learn static mappings effectively after proper input normalization.
- **Computational Efficiency**: MLPs require less computational resources compared to RNNs or LSTMs, making them suitable for offline simulation and pre-trained model evaluation under real-time constraints.
- **Model Simplicity**: The dual-network structure in MLPNet$_{Gene}$ and single-network Net$_{Q_{diff}}$ provided enough flexibility to handle 11 inputs without the complexity of recurrent architectures, since delays were included manually.
- **Training Data Availability**: The dataset and pre-trained checkpoints were optimized for MLP training, supporting this architecture choice.
- **How to Choose**: The process involved:
  - Framing the task as static regression with delayed inputs.
  - Validating performance of MLP versus alternatives (e.g., LSTM) on holdout data.
  - Accounting for deployment constraints such as inference speed and memory usage.
- **Rationale**: MLP was selected for its balance of accuracy, simplicity, and speed, leveraging delayed inputs to capture temporal effects. Net$_{Q_{diff}}$ added an iterative refinement mechanism akin to feedback control.

This approach ensures robust predictions while remaining computationally practical for the current implementation.

6. DESIGN AND JUSTIFICATION OF NN ARCHITECTURE FOR TURBINE–GENERATOR MODELING

6.1 OVERVIEW

Guided by the closed-loop structure in Fig. 11, the digital twin is decomposed into three compact neural blocks that mirror plant physics:

1. **NN2# Turbine / Hydraulic**: learns water-side net head/pressure dynamics.
2. **NN1# Rotor Speed**: updates shaft speed via a residual rule with explicit electrical loading.
3. **NN3# Generator/Electrical**: maps the operating point to terminal electrical quantities.

All three are shallow multi-layer perceptrons (MLPs) with one hidden layer of 128 units (Linear → BatchNorm1d → ELU → Linear). This balances capacity, stability, and real-time inference.

6.2 DISCRETE-TIME MAPS

Let $u(k)$ be the gate command (%), $x(k)$ rotor speed, $h(k)$ net head/pressure, $n_h(k)$ a measured/estimated net head, $P(k)$ active power, $V_t(k)$ terminal voltage, (I_f, V_f) exciter field current/voltage. The learned maps used in the implementation are:

$$\hat{h}(k) = f_H\big(x(k-1),\ u(k),\ h(k-1),\ n_h(k)\big) \quad \text{(NN2\#: hydraulic/pressure)}, \tag{13}$$

$$\hat{x}(k) = x(k-1) + f_\omega\big(h(k-1),\ u(k),\ x(k-1)\big) - a\,P(k) \quad \text{(NN1\#: residual speed with load)}, \tag{14}$$

$$\begin{bmatrix} \hat{P}(k) \\ \hat{V}_t(k) \end{bmatrix} = f_G\big(z(k)\big), \qquad z(k) = \big[u(k),\ I_f(k),\ V_f(k),\ V_f(k-1),\ V_f(k-2),\ V_f(k-3)\big] \quad \text{(NN3\#: electrical)}. \tag{15}$$

Note. Feeding $h(k)$ instead of $h(k-1)$ into $f_\omega(\cdot)$ is a supported variant; we use $h(k-1)$ by default to match the current runtime.

6.3 GENERAL NEURAL NETWORK DESIGN CONSIDERATIONS

Table 2. Guiding principles for NN selection in turbine–generator identification

Aspect	Design implication
Coupled physics	Keep explicit couplings: gate→head, head/gate→speed, P →load damping
System order/memory	Use one-step memory in h, x and short V_f lags within $z(k)$
Function complexity	One hidden layer (128) with ELU is sufficient; add width if needed
Data availability	Avoid overparameterization; BN provides light regularization
Real-time rollouts	Shallow MLPs run in real time; use `float64` where drift matters

6.4 HYDRAULIC/PRESSURE NEURAL NETWORK (`MLPNET_PRESSURE_NETHEAD`, NN2#)

Architecture (PyTorch)

```
nn.Sequential(
    nn.Linear(4, 128),
    nn.BatchNorm1d(128),
    nn.ELU(),
    nn.Linear(128, 1)
)
```

Forward rule

$$\hat{h}(k) \;=\; f_H(x(k{-}1),\; u(k),\; h(k{-}1),\; n_h(k)).$$

Design justification

- **Inputs (4):** $\{x(k{-}1), u(k), h(k{-}1), n_h(k)\}$ match (13); one-step memory captures dominant water inertia at the sampling rate.
- **Hidden width (128):** Sufficient for mild nonlinearity typical of turbine head dynamics.
- **ELU + BN:** Smooth gradients and stable training across operating ranges; no dropout required at this size.
- **Runtime flexibility:** Net-head term $n_h(k)$ can be disabled to revert to a 3-input variant without retraining the interface.

6.5 ROTOR SPEED NEURAL NETWORK (`MLPNET_SPEED`, NN1#)

Architecture (PyTorch)

```
nn.Sequential(
    nn.Linear(3, 128),
    nn.BatchNorm1d(128),
    nn.ELU(),
    nn.Linear(128, 1)
)
```

Forward rule (residual with load term)

$$\hat{x}(k) \;=\; x(k{-}1) + f_\omega\big(h(k{-}1), u(k), x(k{-}1)\big) - a\,P(k),$$

where $a \in \mathbb{R}$ is a *learned* scalar (PyTorch parameter) that captures the aggregate torque–speed sensitivity to active power (electrical loading).

Design justification

- **Inputs (3):** $\{h(k{-}1), u(k), x(k{-}1)\}$ implement the local physics; $P(k)$ enters as an explicit damping term outside the MLP.
- **Residual update:** Greatly reduces drift in long rollouts by learning *increments* around $x(k{-}1)$.
- **Learned a:** Lets the data set the strength/sign of P's effect instead of fixing it a priori.
- **Precision:** Evaluate in `float64` to prevent slow integration error.

6.6 GENERATOR/ELECTRICAL NEURAL NETWORK (`MLPNET_GENE`, NN3#)

Architecture (PyTorch)

```
nn.Sequential(
    nn.Linear(6, 128),
    nn.BatchNorm1d(128),
    nn.ELU(),
    nn.Linear(128, 2)   # outputs: [P, Vt] or [Vt, Q]
)
```

Forward rule

$$\begin{bmatrix} \hat{P}(k) \\ \hat{V}_t(k) \end{bmatrix} = f_G\big(z(k)\big), \quad z(k) = [u(k), I_f(k), V_f(k), V_f(k-1), V_f(k-2), V_f(k-3)].$$

(Variant: switch the two outputs to $[\hat{V}_t(k), \hat{Q}(k)]$ if reactive studies are primary.)

Design justification

- **Inputs (6):** Short exciter memory (V_f lags 0–3) and present operating point (u, I_f) give strong predictive power with a compact input.
- **Two outputs:** Jointly estimates terminal quantities with shared features; helps consistency between P and V_t (or V_t and Q).
- **Shallow & wide:** One hidden layer avoids vanishing-gradient issues; 128 units capture cross-terms adequately.
- **Precision:** Use `float64` for long closed-loop runs where V_t bias matters.

6.7 IMPLEMENTATION-SPECIFIC I/O AND POST-PROCESSING

NN3# Generator/Electrical (`MLPNet_Gene`)

6.7.0.1 Input order (size 6).

$$z(k) = \left[\, u(k),\ I_f(k),\ V_f(k),\ V_f(k-1),\ V_f(k-2),\ V_f(k-3)\, \right]^\top.$$

6.7.0.2 Raw outputs and scaling (affine, empirically tuned).

Let the raw network outputs be $(\tilde{P}, \tilde{V}_t)$. The final signals are

$$P(k) = (5\,\tilde{P}(k) + 12.5)\,P_{\text{NHComp}}, \tag{16}$$

$$V_t(k) = \frac{\tilde{V}_t(k)}{3} + V_{\text{exc,min}}. \tag{17}$$

NN2# Hydraulic/Pressure (`MLPNet_pressure_nethead`)

6.7.0.3 Warm start.

For the first `nn2_num_timestamps_for_initial_values` steps, seed with repeated measured $\{u, h, x\}$ to avoid cold-start artifacts.

6.7.0.4 Runtime call (net-head on/off).

$$\boxed{\hat{h}(k) = f_{H,\text{nh}}\big(x(k-1),\ u(k),\ h(k-1),\ n_h(k)\big)} \quad \text{or} \quad \hat{h}(k) = f_H\big(x(k-1),\ u(k),\ h(k-1)\big).$$

NN1# Rotor Speed (`MLPNet_speed`)

6.7.0.5 Residual + damping + first-order smoothing.

$$\hat{x}(k) = x(k-1) + f_\omega\big(h(k-1),\ u(k),\ x(k-1)\big) - a\,P(k),$$

then (implemented) post-processing:

$$x_{\text{raw}}(k) \leftarrow \tfrac{1}{2}\big(x_{\text{raw}}(k) - 225\big) + 225, \qquad x(k) \leftarrow \alpha\, x_{\text{raw}}(k) + (1-\alpha)\, x(k-1), \quad \alpha = \frac{1}{T_s + 1}.$$

6.8 ROLLOUT ORDER

Given PIDD controller output $u(k)$ and the latest measurements:

1. Build $z(k)$; evaluate $[\tilde{P}(k), \tilde{V}_t(k)] = f_G(z(k))$; scale to $P(k)$ and $V_t(k)$.
2. Evaluate $\hat{h}(k)$ using the selected pressure model (with/without net head); set $h(k) \leftarrow \hat{h}(k)$.
3. Compute $x(k)$ via the residual rule with $P(k)$, then apply damping and the low-pass filter.
4. Apply fixed plant feedbacks (RL, $K_e \frac{1+T_1 s}{1+T_2 s}$, sensors) as in Fig. 11.

6.9 QUICK REFERENCE (I/O AND ARCHITECTURE SUMMARY)

Table 3. Neural blocks, inputs, outputs, and salient implementation details

Block / Class	Inputs (order)	Outputs	Notes
`MLPNet_nethead` (NN2#)	$[x(k-1),\ u(k),\ h(k-1),\ n_h(k)]$	$\hat{h}(k)$	ELU+BN; net-head on/off
`MLPNet_speed` (NN1#)	$[h(k-1),\ u(k),\ x(k-1)]\ +\ P(k)$	$\hat{x}(k)$	Residual $+(-aP)$; damping & LPF
`MLPNet_Gene` (NN3#)	$[u, I_f, V_f, V_{f,-1}, V_{f,-2}, V_{f,-3}]$	$[\tilde{P}, \tilde{V}_t]$	Scale to P, V_t (alt. $[V_t, Q]$)

6.10 TRAINING NOTES

- **Losses:** MSE on h, residual x (predict increment), and the two generator outputs.
- **Optimizer:** Adam with BN and ELU; LR 1e-3 with decay; batch 256–1024.
- **Preprocessing:** Robust outlier removal on u, P, V_f, I_f; z-score or min–max normalization; persist scalers.
- **Validation:** Hold-out days; report MAE/RMSE and long-horizon drift in x and V_t.

7. CASE STUDIES: TACOMA PUBLIC UTILITIES

We model the turbine–generator as three coupled neural components that mirror the physics while remaining lightweight for fast rollouts. Each component is a compact MLP with the pattern

$$\text{Linear} \rightarrow \text{BatchNorm1d} \rightarrow \text{ELU} \rightarrow \text{Linear}$$

using 128 hidden units. The `speed` and `generator` networks are evaluated in double precision to improve long-horizon numerical stability.

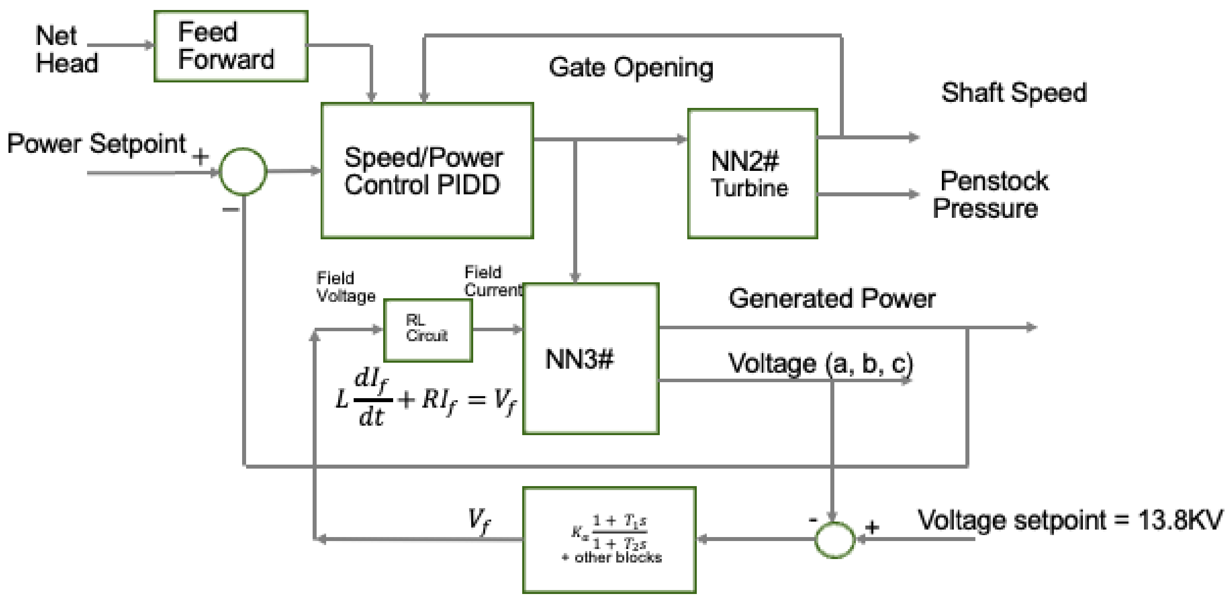

Figure 11. Neural network model structure for Tacoma power unit.

The hydropower unit control system, as depicted in Figure 11, is a closed-loop system designed for speed and power regulation. To develop its digital twin, the system was decomposed into key functional blocks:

- **Speed/Power Control PIDD:** This is the primary controller responsible for generating the gate position command (PID output) to regulate the unit's speed and power.
- **NN2# Turbine:** This block represents the turbine's hydraulic and mechanical dynamics, translating gate position and pressure changes into shaft speed variations.
- **NN3# Generator/Electrical System:** This block models the electrical characteristics of the generator, converting mechanical power from the turbine into electrical power and regulating terminal voltage.
- **Physical System Feedback:** This includes components like an RL circuit and a transfer function, $\frac{1+T_1s}{1+T_2s}$, which represent the inherent dynamic responses and feedback paths within the physical system.

Each dynamic component (NN2# and NN3#) is approximated by specialized MLP neural networks.

7.1 NEURAL NETWORK MODEL ARCHITECTURES

The core of the digital twin consists of several custom-designed MLP networks implemented in PyTorch. Each network is a subclass of `nn.Module` and is configured for double-precision floating-point operations (`self.double()`) to enhance numerical stability, which is critical for physical system simulations. Each MLP network is structured with an input layer, a hidden layer (128 neurons with `BatchNorm1d` and `ELU` activation), and an output layer.

- **`MLPNet_speed` (NN1#):**
  - **Input (3 features):** Current pressure (h), control input (u), and previous speed (x_{last}). (Note: Current power P is also used externally in the speed calculation).
  - **Output (1 feature):** Incremental speed change.
  - **Forward Pass:** $x = x_{last} + \mathrm{NN}(\mathrm{concat}(h, u, x_{last})) - a \cdot P$, where a is a learnable parameter. This implements a residual connection.
  - Speed NN: $\hat{x}(k) = f_\theta(h(k),\ u(k),\ x(k{-}1))$

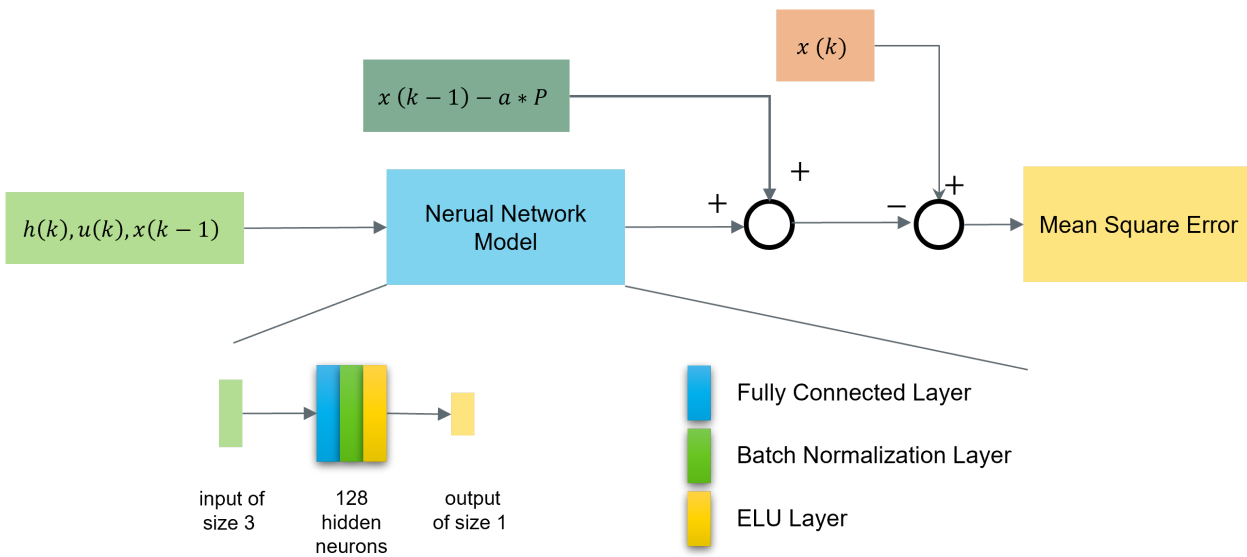

Figure 12. Neural network model structure for speed.

- **`MLPNet_pressure_nethead` (NN2#):**
  - **Input (4 features):** Current speed (x), control input (u), previous pressure (h_{last}), and net head.
  - **Output (1 feature):** Current pressure (h).
  - **Forward Pass:** $h = \mathrm{NN}(\mathrm{concat}(x, u, h_{last}, \mathrm{nethead}))$
  - Pressure NN: $\hat{h}(k) = f_\theta(x(k),\ u(k),\ h(k{-}1),\ \mathrm{nethead})$
- **`MLPNet_Gene` (NN3#):**
  - **Input (6 features):** Control input (u, gate position), exciter field current (I_f), and current/lagged filtered field voltages ($V_{f,filtered}$, $V_{f,filtered,last}$, etc.).
  - **Output (3 features):** Active/Reactive Power (P/Q) and terminal voltage (V_t).
  - **Forward Pass:** $[P, V_t] = \mathrm{NN}(\mathrm{concat}(u, I_f, V_{f,filtered}, \dots))$
  - Generator NN: $Q(k)$,$P(k)$,$V_t(k) = f_\theta(u(k), V_f(k), V_f(k-1), V_f(k-2), V_f(k-3), I_f(k))$

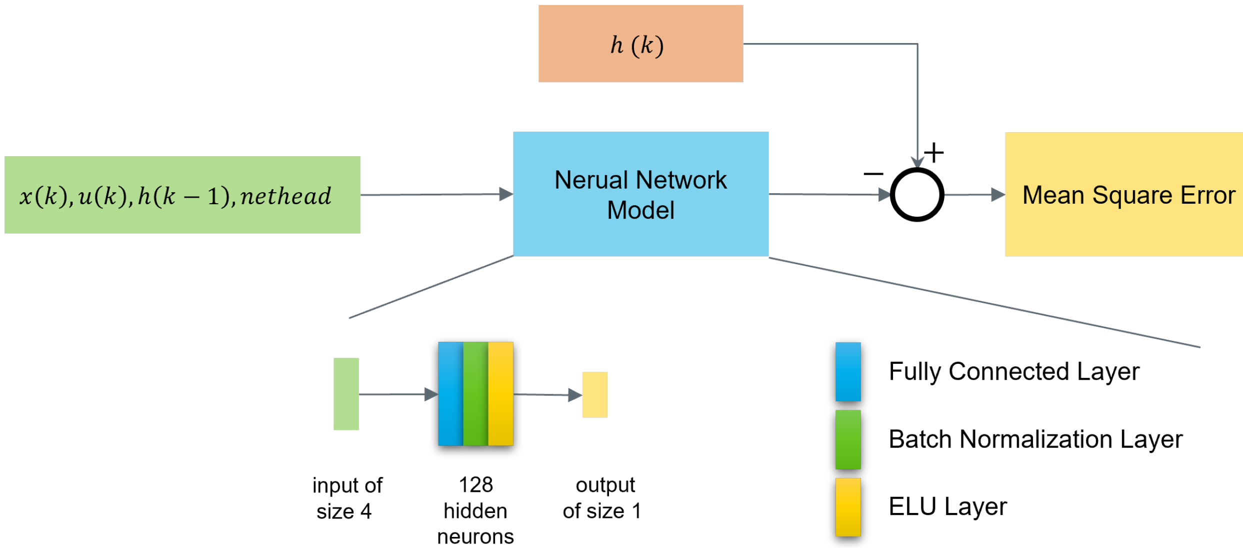

Figure 13. Neural network model structure for pressure model considering nethead.

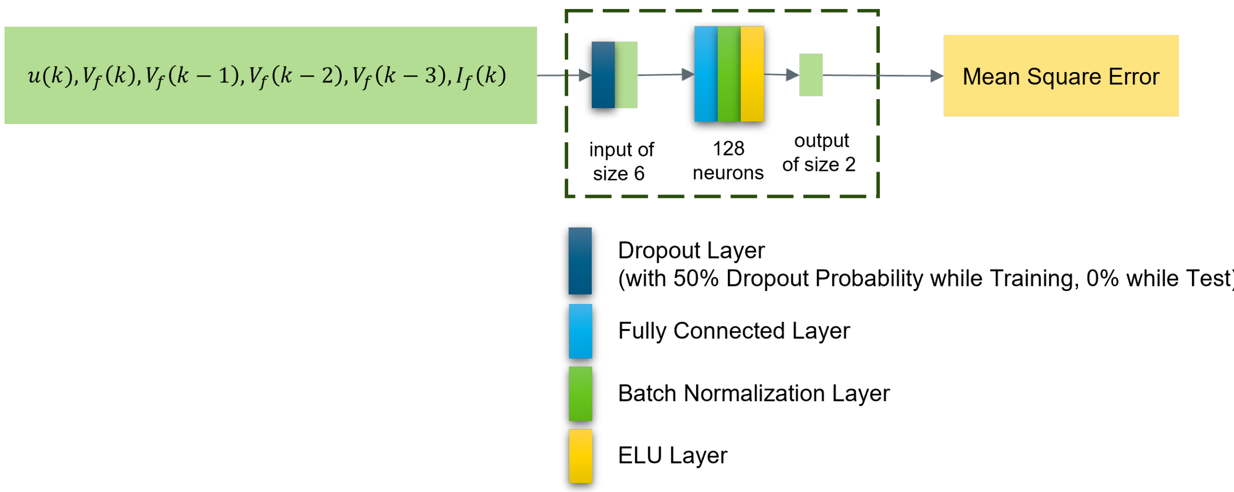

Figure 14. Neural network model structure for generator.

7.1.1 MLPNet_pressure

This model is designed to predict the pressure within the hydraulic system.

Inputs:

- `x`: Typically represents the current shaft speed.
- `u`: Represents the control input, the gate position (PID output).
- `last_h`: The pressure value from the previous time step.

Output:

- A single scalar representing the predicted current pressure.

Forward Pass: The inputs are concatenated: `x_h_u = torch.concatenate((x, u, last_h), dim=1)`. This combined tensor is then passed through the network.

7.1.2 MLPNet_pressure_nethead

An alternative pressure model that incorporates the net head as an additional input, allowing for more comprehensive modeling, especially in systems where net head variations significantly influence pressure.

Inputs:

- `x`: Current shaft speed.
- `u`: Control input (gate position).
- `last_h`: Pressure value from the previous time step.
- `nethead`: The measured or calculated net head of the hydropower unit.

Output:

- A single scalar representing the predicted current pressure.

Forward Pass: All four inputs are concatenated: `inputs = torch.concatenate((x, u, last_h, nethead), dim=1)`. The result is fed into the network.

7.1.3 MLPNet_speed

This model is responsible for predicting the shaft speed of the hydropower unit. It incorporates a residual connection and a learnable parameter for fine-tuning.

Inputs:

- `h`: Current pressure.
- `u`: Control input (gate position).
- `last_x`: Shaft speed from the previous time step.
- `P`: Current electrical power output.

Output:

- A single scalar representing the predicted current shaft speed.

Forward Pass: The inputs `h`, `u`, and `last_x` are concatenated. The output of the network's sequential layers is then added to `last_x` (residual connection) and an amount proportional to `P` is subtracted (`-self.a * P`). The parameter `self.a` is a learnable parameter that allows the model to adjust the influence of power on speed dynamics.

7.1.4 MLPNet_Gene

This network models the generator and electrical system, predicting the active power output and terminal voltage.

Inputs:

- A concatenated tensor of 6 inputs, which in the simulation loop are:
  - `PID_output` (current gate position).
  - `If_exciter` (exciter field current).
  - `Vf_filtered`, `Vf_filtered_last`, `Vf_filtered_lastlast`, `Vf_filtered_lastlastlast` (current and historical filtered field voltages).

Output:

- A tensor of two scalars representing the predicted active power (P) and terminal voltage (V_t).

Forward Pass: The 6 input features are directly passed through the network.

The development relies on collecting extensive operational data from a real hydropower unit. This data is used to train each MLP network using standard supervised learning techniques (e.g., Mean Squared Error loss, Adam optimizer). The goal is for the networks to learn the complex, non-linear mappings between their inputs and outputs. The core of the digital twin is a time-stepping simulation that integrates the trained neural networks and external control logic.

7.2 TRAINING AND VALIDATION (CONCEPTUAL)

The development of this digital twin implicitly involved:

- **Data Acquisition:** Collecting extensive operational data from a real hydropower unit under various load conditions, disturbances, and control actions. This data would include all inputs and outputs for each neural network.
- **Model Training:** Training each MLP network on this collected data using appropriate loss functions (e.g., Mean Squared Error) and optimization algorithms (e.g., Adam). This process would minimize the discrepancy between the network's predictions and the true system behavior.
- **Validation:** Rigorously validating the trained digital twin against unseen real-world data to ensure its accuracy, robustness, and ability to generalize to new operating conditions. Performance metrics such as RMSE or MAE would be used.

7.3 DIGITAL TWIN SIMULATION LOOP

The simulation loop integrates the trained neural networks to emulate the hydropower unit's dynamic behavior over time. The process is sequential, where the output of one model feeds into the next, mimicking the causality of the real system.

7.3.1 Initialization

For the initial timestamps, the simulation leverages pre-defined initial values for gate position, pressure, and shaft speed to warm up the system and provide necessary historical context to the recurrent inputs of the neural networks.

- **Sequential Prediction:** At each time step t, the simulation sequentially calculates outputs:
  1. **Turbine Dynamics (NN2#):**
     - Pressure is predicted using current speed, gate position, and previous pressure (and net head if applicable).
     - Raw speed is predicted using current pressure, gate position, previous speed, and current power.
     - A first-order filter is applied to the raw speed: speed = $\frac{1}{\text{Speed\_Ts+1}} \cdot$ speedraw + $\frac{\text{Speed\_Ts}}{\text{Speed\_Ts+1}} \cdot$ speedraw$_{last}$. This dampens the response, mimicking turbine inertia.
  2. **Generator Dynamics (NN3#):**
     - Electrical power (P) and terminal voltage (V_t) are predicted based on gate position, exciter field current, and historical filtered field voltages.

- Outputs are scaled (e.g., $P = (\text{raw\_P} \cdot 5 + 12.5) \cdot \text{P\_NHComp}$).

- **Feedback and State Update:**
  - Power and speed errors (relative to setpoints) are calculated. These errors would typically feed back into the external PIDD controller for the next time step.
  - Critical state variables (e.g., current pressure, speed, filtered voltages) are updated to become "previous" values for the next time step, preserving the system's dynamic memory. This includes updating `last`, `lastlast`, and `lastlastlast` values for inputs to the NNs.

This integrated approach allows the digital twin to accurately mimic the real-time, dynamic behavior of the hydropower unit, providing a valuable tool for analysis, control design, and optimization.

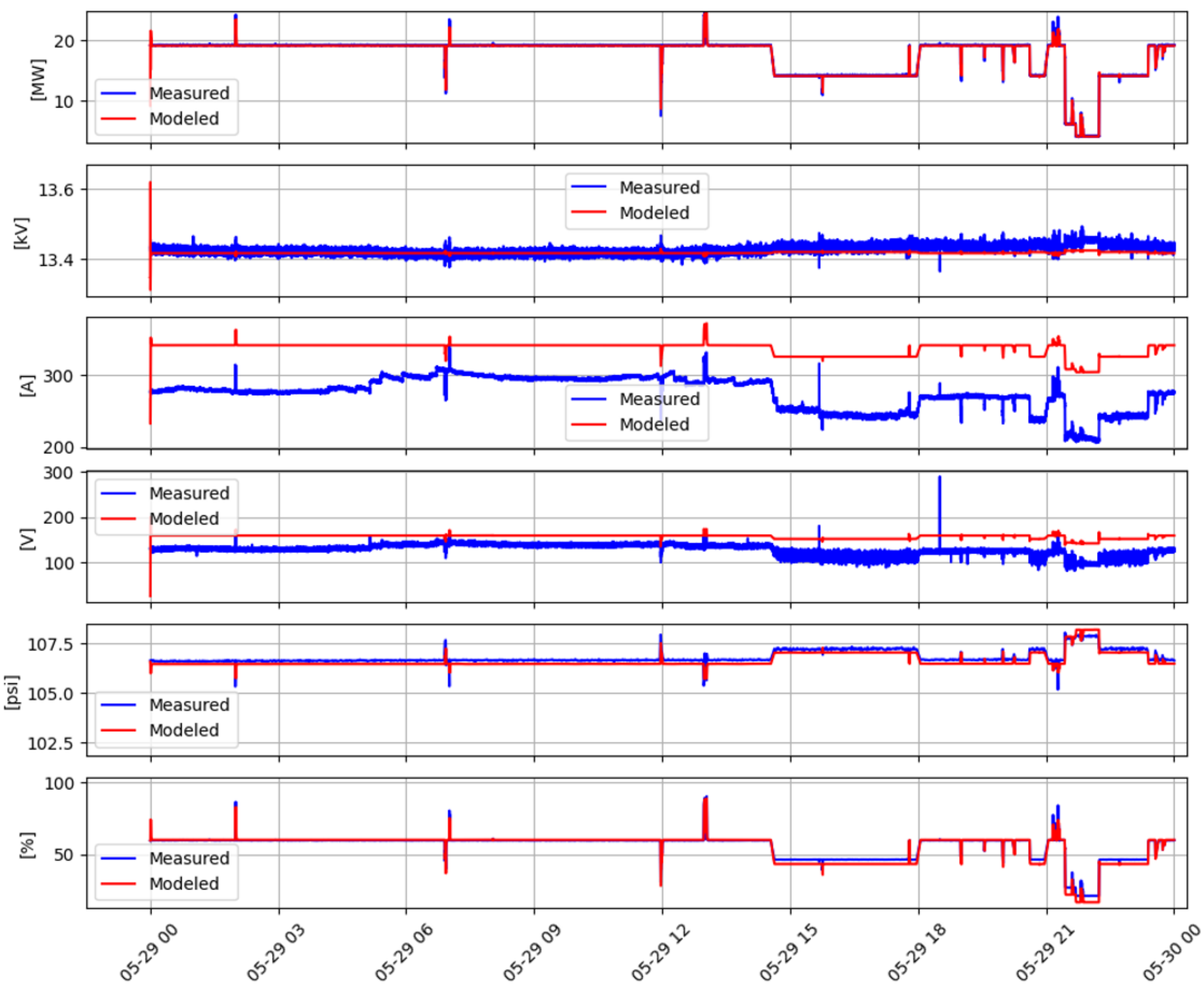

Figure 15. Closeloop simulation output response from NN models compared to the actual plant output.

7.3.2 Feedback and Data Update

After each simulation step, the following feedback and data updates occur:

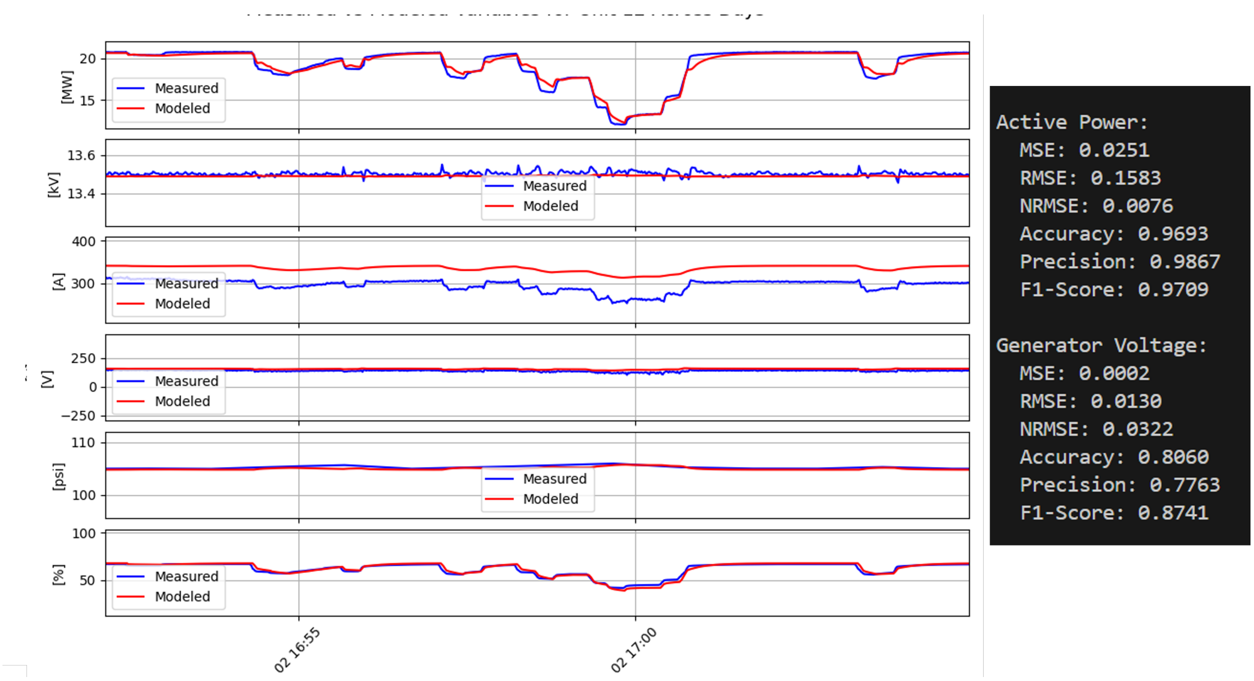

Figure 16. Closeloop response from NN models.

- **Error Calculation:** The predicted power and speed are compared against their respective setpoints (`P_setpoint`, 225 for speed) to calculate the power error (`power_err`) and speed error (`speed_err`). These errors would then feed into the external PIDD controller for the next time step.
- **State Variable Update:** Crucially, the historical values for pressure, speed, and filtered voltages are updated. For example, `pressure_last` becomes the current `pressure`, `pressure_lastlast` becomes `pressure_last`, and so on. This ensures that the neural networks have access to the necessary past states to predict future behavior, simulating the system's memory and dynamic dependencies.

This modular, data-driven approach, coupled with careful system decomposition and integration, allows for the creation of a high-fidelity digital twin capable of accurately mimicking the complex dynamics of a hydropower unit.

8. CONCLUSIONS

The development and implementation of digital twins for hydropower systems represent a significant advancement in the management and optimization of renewable energy resources. This user manual has provided a comprehensive, step-by-step framework for creating and utilizing digital twins, with a particular focus on leveraging data-driven modeling techniques such as neural networks. By accurately capturing the nonlinear dynamics of hydropower generation units, digital twins enable operators to monitor, predict, and optimize system performance in real time.

For the Francis unit, the digital twin combines a residual speed model with learned load coupling, a pressure model with optional net-head input, and a compact generator network for electrical outputs. Evaluated in double precision and paired with first-order smoothing where appropriate, the twin remains numerically stable over long horizons while accurately reproducing shaft speed, penstock/scroll-case pressure, and terminal quantities. The approach leverages historical signals and short input memories to capture waterway inertia, governor–turbine coupling, and exciter effects without overparameterizing the model.

Operationally, the Francis twin enables scenario testing for setpoint changes and disturbances, provides early insight into pressure pulsations and speed excursions, and supports controller retuning and outage planning. Future enhancements include expanding training data to cover extreme head/flow regimes, refining pressure dynamics with additional hydraulic context (e.g., surge/tunnel effects where relevant), and adding physics-informed regularization to honor limits and efficiency bands. Extending the framework to multi-unit coordination and automating periodic retraining will strengthen generalization and sustain accuracy as operating conditions evolve.

REFERENCES

[1] H. Wang, Z. Yin, and Z.-P. Jiang, “Real-time hybrid modeling of Francis hydroturbine dynamics via a neural controlled differential equation approach," *IEEE Access*, vol. 11, pp. 139133–139146, 2023.

[2] H. Wang, O. Ahmed, K. DeSomber, C. Sasthav, P.-T. Storli, O. Dahlhaug, H. I. Skjelbred, and I. Vilberg, “Adaptive Hybrid 1D Modeling for Digital Twin of Hydropower Systems," Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States), Tech. Rep., 2023.

[3] H. Wang, S. Subedi and W. Jia, “Dynamic Modeling of a Kaplan Hydroturbine Using Optimal Parametric Tuning and Real Plant Operational Data." *Dynamics*, 5(2), 2025.

[4] T. Tomia (Original by Tennessee Valley Authority), “Hydroelectric Dam Diagram,” Wikimedia Commons, 30 Dec. 2007. [Online]. Available: https://commons.wikimedia.org/wiki/File:Hydroelectric_dam.svg.

[5] H. Wang and S. Ou, “Structured Neural Network Modeling for Developing Digital Twins Models of Hydropower Generation Units," in *2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA)*, 2024, pp. 1–6.

[6] Z. Yin, H. Wang, and Z.-P. Jiang, “Parameter Estimation of Synchronous Generator Using Neural Controlled Differential Equations," in *2024 IEEE 18th International Conference on Control & Automation (ICCA)*, 2024, pp. 332–339.

[7] PowerWorld Corporation, “ESST5B Excitation System Model Description,” Technical Documentation, 2025.

APPENDIX A. FIRST APPENDIX

APPENDIX A. FIRST APPENDIX

Table A.1. ESST5B Exciter Model Parameters

Parameter	Description
Tr	Filter Time Constant
Kr	Regulator Gain
T1	Firing Circuit Time Constant
Kc	Rectifier Regulation Factor
Vrmax	Maximum Regulator Output
Vrmin	Minimum Regulator Output
Tc1	Regulator Lead Time Constant
Tb1	Regulator Lag Time Constant
Tc2	Regulator Lead Time Constant
Tb2	Regulator Lag Time Constant
Toc1	OEL Lead Time Constant
Tob1	OEL Lag Time Constant
Toc2	OEL Lead Time Constant
Tob2	OEL Lag Time Constant
Tuc1	UEL Lead Time Constant
Tub1	UEL Lag Time Constant
Tuc2	UEL Lead Time Constant
Tub2	UEL Lag Time Constant

APPENDIX B. SECOND APPENDIX

APPENDIX B. SECOND APPENDIX

Table B.1. HYG3 Governor Model Parameters

Parameter	Description
MWCap	Turbine MW Rating
Pmax	Maximum Gate Opening
Pmin	Minimum Gate Opening
Cflag	Governor Control Flag
Rgate	Steady State Droop for Governor Output Feedback
Relec	Steady State Droop for Electrical Power Feedback
Td	Input Filter Time Constant
Tf	Washout Time Constant
Tp	Gate Servo Time Constant
Velop	Maximum Gate Opening Velocity
Velcl	Maximum Gate Closing Velocity
K1	Derivative Gain
K2	Double Derivative Gain
Ki	Integral Gain
Kg	Gate Servo Gain
Tt	Power Feedback Time Constant
Db1	Intentional Dead-Band Width
Eps	Intentional Dead-Band Hysteresis
Db2	Unintentional Dead-Band
Tw	Water Inertia Time Constant
At	Turbine Gain
Dturb	Turbine Damping Factor
qnl	No-Load Turbine Flow at Nominal Head
H0	Turbine Nominal Head
Gv1	Nonlinear Gain Point 1
Pgv1	Nonlinear Gain Point 1
Gv2	Nonlinear Gain Point 2
Pgv2	Nonlinear Gain Point 2
Gv3	Nonlinear Gain Point 3
Pgv3	Nonlinear Gain Point 3
Gv4	Nonlinear Gain Point 4
Pgv4	Nonlinear Gain Point 4
Gv5	Nonlinear Gain Point 5
Pgv5	Nonlinear Gain Point 5
Gv6	Nonlinear Gain Point 6
Pgv6	Nonlinear Gain Point 6