THE UNIVERSITY OF WESTERN ONTARIO
DEPARTMENT OF CIVIL AND
ENVIRONMENTAL ENGINEERING

Water Resources Research Report

Automated Control Flaw Generation Procedure: Cheakamus Dam Case Study

By:
Bogdan Pavlovic
and
Slobodan P. Simonovic

Report No: 093
Date: March 2016

ISSN: (print) 1913-3200; (online) 1913-3219
ISBN: (print) 978-0-7714-3113-5; (online) 978-0-7714-3114-2

Automated Control Flaw Generation Procedure: Cheakamus Dam Case Study

By

Bogdan Pavlovic

and

Slobodan P. Simonovic

Department of Civil and Environmental Engineering

Western University, Canada

February 2016

Summary

This report deals with the problem of aging hydropower infrastructure systems and system components, a problem that is very common across Canada. Flaws of common risk assessment methods are noted, and the need for new risk assessment approaches is identified. System dynamics simulation method is introduced as an implementation mechanism for the System Theoretic Process Analysis (STPA). STPA and its adaptation to complex hydropower systems are explained thoroughly. The main objectives of the report include (a) the implementation of STPA for investigation and identification of potential hazardous actions and hazardous system states, and (b) development of an automated generic approach for the identification of potential hazardous actions and hazardous system states. The developed methodology is illustrated using a case study based on the BC Hydro's Cheakamus Dam, British Columbia, Canada.

Table of Contents

List of figures

List of tables

1. Introduction

Aging hydropower systems across Canada pose a serious threat to the Canadian economy. Many components of these systems are near the end of serviceable life and will require significant investments in order to be replaced or upgraded (if possible). Many components have been poorly maintained, and require remedial attention, even if maintenance has been adequate, simply because of their age. Technological advances over the past few decades have resulted in increasing complexity of integrated civil infrastructure systems, making management and operations of these systems more of a challenge (Leveson 2011). Constant upgrades and replacement of the components also attributed to the complexity of the infrastructure. Interdependencies of the system components are poorly understood in spite of the fact that system performance and reliability are results of interactions between engineered, natural and human system components (Regan 2010; Leveson 2011; Thomas 2012; Baecher et al. 2013;). Traditional methods of engineering analysis tend to decompose the system into smaller, more manageable components, which essentially ignore interactions between them (Regan 2010; Leveson 2011; Thomas 2012). Limited emphasis is placed on events that could occur within the design envelope (Regan 2010). Dominant risks, to be managed, derive not from extreme events but adverse combinations of less severe events and/or an unusual combination of usual events (Baecher et al. 2012). It is established in the literature that traditional risk assessment methods lack in means of identifying the hazards and initiating events, or if they do have means of identifying the hazards, they focus on major hazards and do not provide a way to include all instigating events. Resulting scenarios that are analyzed do not cover unsafe situations when there were no component failures but lack of safety results from control actions. Similarly, failure components or unsafe control actions might not result in a hazard. This report presents a method for automated generation of control flaw requirements using a new hazard analysis method, the System Theoretic Process Analysis, and system dynamics simulation for investigation of control flaws and states of the hydropower system.

Traditional methods and systems approach are covered in the literature review. STPA, system dynamics simulation, objectives and approach are covered in the problem formulation and methodology sections of the report. Automated generation is presented in the methodology section. Data used and results are presented in the Cheakamus Dam study section.

2. Literature Review

2.1. Definitions and terminology

Hazard is a dangerous phenomenon, substance, human activity or condition that may cause loss of life, injury or other health impacts, property damage, loss of livelihoods and services, social and economic disruption, or environmental damage (UNISDR, 2009). Hazard analysis is the first step in the risk assessment process. Risk is the combination of the probability of an event and its negative consequences. Risk assessment is a methodology to determine the nature and extent of risk by analysing potential hazards and evaluating existing conditions of vulnerability that together could potentially harm exposed people, property, services, livelihoods and the environment on which they depend (UNISDR, 2009). Safety is defined as freedom from accidents (losses) (Leveson, 1995)**.** Safety is defined absolutely as a quality that may not be entirely achievable, but that can still be defined in absolute terms as a desirable quality that can be improved. Reliability is the probability that a piece of equipment or component will perform its intended function satisfactorily for a prescribed time and under stipulated environmental conditions (Leveson, 1995). Unreliability is the probability of failure. Failure is the non-performance or inability of the system or system component to perform its intended function for a specified time under specified environmental conditions (Leveson, 1995).

2.2. Traditional risk assessment methods

Traditional risk assessment methods include Fault Tree Analysis (Watson, 1962), Event Tree Analysis (Rasmussen, 1975), Failure Modes and Effects Analysis and Failure Modes Effects and Criticality Analysis (U.S. Military, 1949). Each method has its limitations, disadvantages, and advantages. FTA (Fault Tree Analysis) begins with an undesirable event but does not provide means to identify undesirable events. The analysis must also be based on existing model of the system. The FTA does not include any standard system model. Expert judgement has been used as a way to identify and quantify operator errors in a fault tree (Thomas, 2012) which is subjective. Event trees also begin with an initiating event but do not provide a way to identify systematically the initiating events or how to include all relevant events. Human behaviour is reduced to a binary decision that is connected to a context in which it occurs. Event tree introduces set of barriers or protective functions intended to prevent an event leading to an accident. Barriers in the event tree are often assumed to operate independently, while in practice that is often not the case, especially if human behaviour is involved. FMEA (Failure Modes and Effects Analysis) and FMECA (Failure Modes Effects and Criticality Analysis) were developed to evaluate the effect of component failures on system performance systematically. FMEA follows the bottom-up approach. Various components in the system are identified and then failure modes - mechanisms by which a component may fail to achieve its designed function, are investigated. FMECA follows the same process but assigns a criticality to each failure mode based on severity and probability of each identified effect. Resulting scenarios that are analyzed include both, hazardous and non - hazardous scenarios triggered by a failure. Unfortunately a set of scenarios triggered by failure does not include all unsafe scenarios. FMECA does not capture non - linear and feedback relationships and omits scenarios that result from a combination of several failures.

Event based techniques are not suited to handle complex software intensive systems, complex human - machine interactions, and systems - of - systems with distributed decision - making that cut across both physical and organizational boundaries (Dulac, 2007).

2.3. Systems approach to risk assessment

Many researchers, including Rasmussen (1997), Hollnagel (2002), Woods (2002) and Leveson (2004), have been advocating an alternative, systems, approach to safety. The primary differences between traditional techniques and a systems approach are: (1) traditional approach relies on top - down systems thinking rather than bottom – up; (2) traditional view has reliability engineering focus (Dulac, 2007).

Systems analysis is defined as "the use of rigorous methods to help determine preferred plans, design and operations strategies for complex, often large-scale, systems" (Simonović 2009). Techniques that can be used in system analysis include simulation and optimization (with single and multiple objective functions). Simulation models describe how the system operates, and are used to assess what changes in system behaviour will result from a specific course of action. Simulation models describe the state of the system in response to change in system structure and various inputs but give no direct measure of what decisions should be taken to improve the performance of the system (Simonović 2009).

System-Theoretic Accident Model and Processes (STAMP) is an accident model created by Nancy Leveson that is based on systems theory. STAMP treats safety as a control problem, rather than as a failure problem. Unsafe control includes inadequate handling of failures, software design errors and erroneous human decision making. Accidents are viewed as the result of inadequate enforcement of constraints on system behaviour. The reason behind the inadequate enforcement may involve classic component failures, but it can also result from unsafe interactions among components operating as designed or from erroneous control actions by software or humans (Thompson, 2012). STAMP is based on the observation that there are four types of hazardous control actions that need to be eliminated or controlled to prevent accidents:

1. A control action required for safety is not provided or is not followed.
2. An unsafe control action is provided that leads to a hazard.
3. A potentially safe control action is provided too late, too early, or out of sequence.
4. A safe control action is stopped too soon or applied for too long.

The process model contains the controller's understanding of (a) the current state of the controlled process, (b) the desired state of the controlled process, and (c) the ways the process can change the state. This model is used by the controller to determine what control actions are needed (Thompson, 2012).

System Theoretic Process Analysis (STPA) is a hazard analysis technique built on STAMP. It can be applied in order to derivate causal factors related to human controllers within the context of the system and its design. The objective of STPA is to identify scenarios of inadequate control that could potentially lead to an accident.

STPA is performed on generic control system structure outlined by Leveson (2011). Stabilizing control loop includes a controller, actuators, a controlled process (the infrastructure) and sensors which relay information back to the controller. According to Leveson (2011) this high-level system structure represents a hierarchical system of systems, with each box representing its own system.

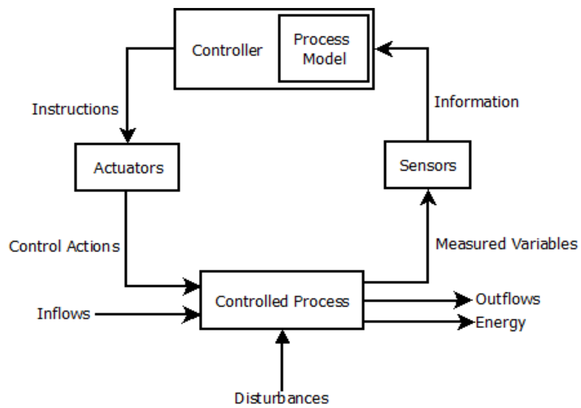


Figure 1. Generic control system (after Leveson 2011)

Detailed control loop, as it relates to hydropower system, is presented in Figure 2. States of all inflows, disturbances and system components are automatically generated. Sensors relay system state information to controller, part (a) of the process model. A fuzzy inference system (FIS) introduced below, is used to model controller's decision making in the particular state of the system. Controller issues instructions that are performed by actuators (if possible). Controlled process is operation of the spillway gate. System dynamics simulation is used to simulate water level change in the reservoir over time. Sensors monitor water level and relay information to the controller, closing the control feedback loop.

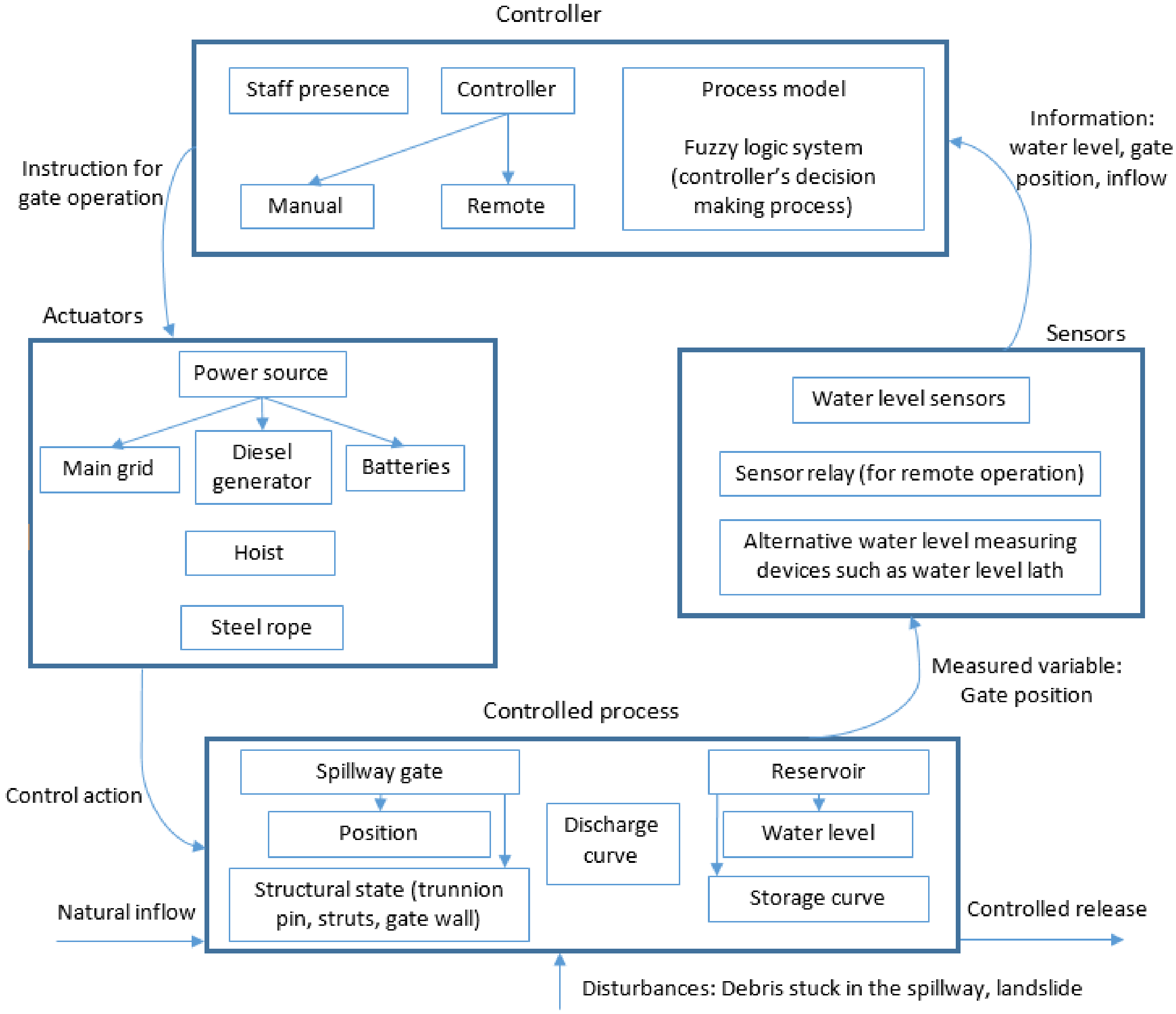


Figure 2. Detailed control feedback loop for a hydropower system

2.4. Fuzzy inference systems

In certain cases, experienced operators achieve better results while operating complex systems than automated control systems. Operator's management strategies can be expressed as a set of heuristic rules that are difficult to express using traditional algorithms. These difficulties are caused by the fact that people mainly use qualitative expressions for description of certain situations. Theory of fuzzy sets and fuzzy logic offer an approach to computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. Fuzzy logic systems were created from the desire to incorporate human experience, intuition and behaviour in the process of making decisions (Zimmermann, 1991). The idea of developing a model of decision making based on imprecise, qualitative data and descriptive linguistic rules that are combined using fuzzy logic comes from work of Lotfi Zadeh (Zadeh, 1973).

2.4.1. Fuzzy set theory

In classical set theory, membership of objects are assessed in binary terms. An object either belongs or does not belong to a set which is expressed with a 1 or a 0. Classical set membership function $\mu_{\tilde{A}}$ for an element $x \in X$ can be expressed in mathematical form as:

$$\mu_{\tilde{A}}(x) = \begin{cases} 1, if\ x \in A \\ 0, if\ x \notin A \end{cases} \tag{2.1}$$

where $\mu_{\tilde{A}}(x)$ is the function denoting the membership of x in set A.

Fuzzy set theory permits intermediate membership classes to sets. Characteristic function takes values between 1 and 0, i.e. values in the real unit interval [0, 1]. If *X* is a universal set whose elements are {x}, then a fuzzy set is defined by its membership function:

$$\mu_{\tilde{A}}: X \rightarrow [0, 1], \tag{2.2}$$

which assigns a degree in the interval [0, 1] of membership to every element x.

Fuzzy set can be represented by a set of ordered pairs of elements, which present the element together with its membership value to the fuzzy set:

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | x \in X\} \tag{2.3}$$

Membership functions can be generated using several methods: intuition, inference, rank ordering, neural networks, genetic algorithms and inductive reasoning (Ross, 2010).

Fuzzy set is normal fuzzy set if at least one of its elements has a membership value of 1.

2.4.2. Set-theoretic operations for fuzzy sets

Most common membership function shapes are presented in Figure 3. The basic operations of fuzzy sets

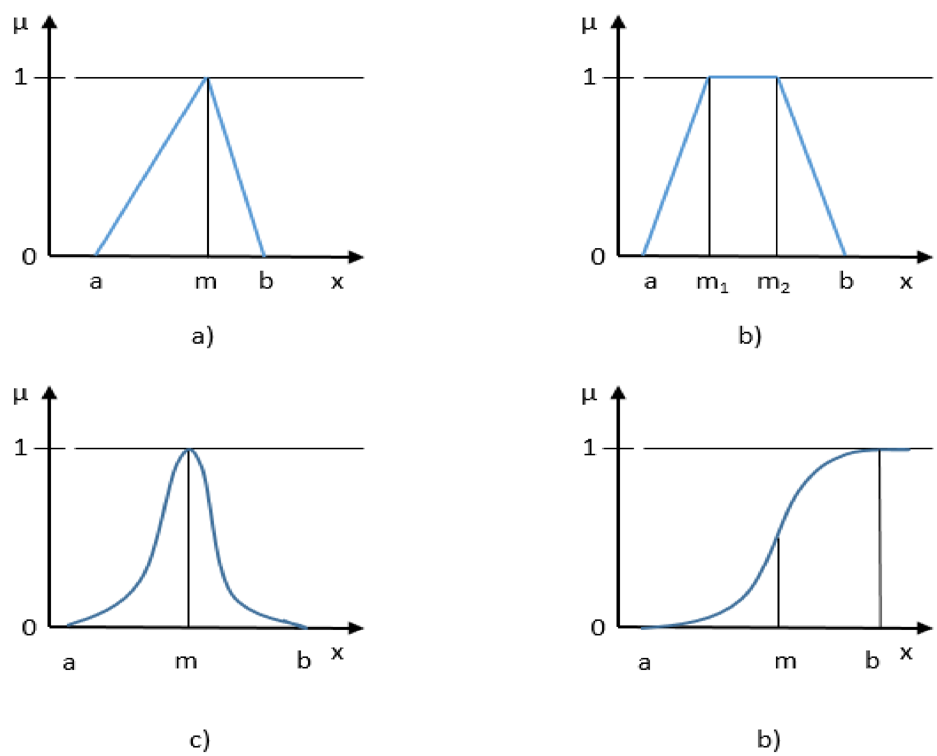


Figure 3. (a) triangular, (b) trapezoid, (c) Gaussian and (d) sigmoid membership functions

include intersection and union.

Intersection of fuzzy set Ã with B̃, C̃ = Ã ∩ B̃ is defined by:

$$\mu_{\tilde{C}}(x) = min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in \mathrm{X} \qquad (2.4)$$

where:

$\mu_{\tilde{C}}$ (*x*) is the membership of the fuzzy intersection of Ã and B̃;

min ( ) is the ordinary minimum operator;

$\mu_{\tilde{A}}$ (*x*) is the membership of fuzzy set Ã; and

$\mu_{\tilde{B}}$ (*x*) is the membership of fuzzy set B̃.

Union of fuzzy set Ã with B̃, C̃ = Ã ∪ B̃ is defined by:

$$\mu_{\tilde{C}}(x) = max\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\}, x \in X \qquad (2.5)$$

where:

$\mu_{\tilde{C}}$ (*x*) = the membership of the fuzzy union of Ã and B̃;

max ( ) = the ordinary maximum operator;

$\mu_{\tilde{A}}$ (*x*) = the membership of fuzzy set Ã; and

$\mu_{\tilde{B}}$ (*x*) = the membership of fuzzy set B̃.

Graphical presentation of intersection and union is shown in Figure 4.

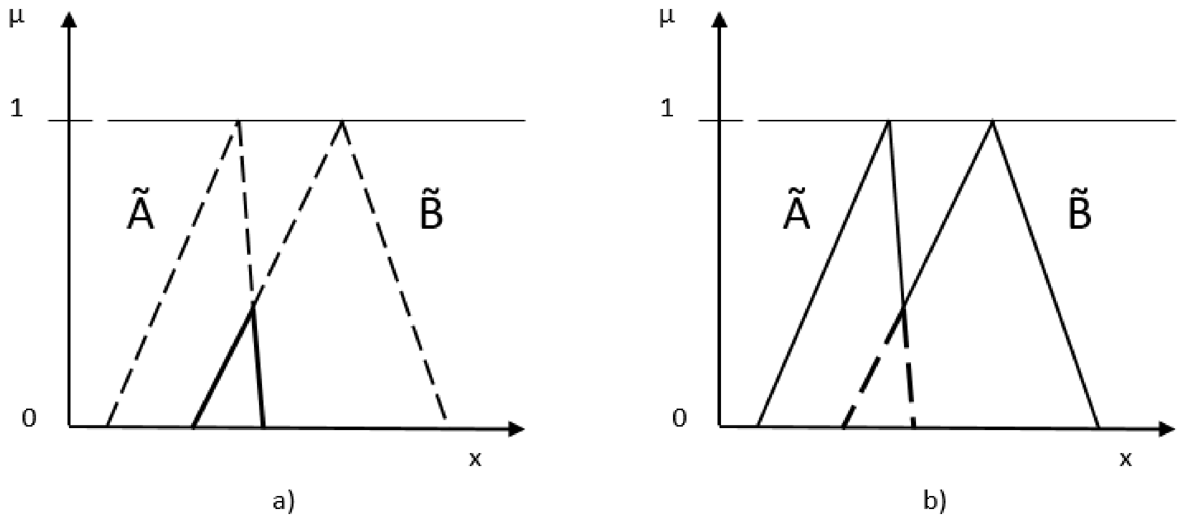


Figure 4. Union (a) and intersection (b) of fuzzy sets Ã and B̃

A fuzzy number is a special case of fuzzy set that has the following properties:

- it is defined in the set of real numbers;
- it is a normal fuzzy set; and
- it is convex.

Fuzzy number can be defined as follows:

$$\tilde{X} = \{(x, \mu_{\tilde{X}}(x)) \colon x \in R;\ \mu_{\tilde{X}}(x) \in [0,1]\} \qquad (2.6)$$

where

$\tilde{X}$ is the fuzzy number;

$\mu_{\tilde{X}}(x)$ is the membership value of element x to the fuzzy number; and

R is the set of real numbers.

A Fuzzy set is convex if and only if it satisfies following property:

$$\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)) \quad (2.7)$$

where λ is the interval [0, 1] and $x_1 < x_2$. Visually it is the same as a convex polygon.

At any α-level, the fuzzy number Ã can be represented in the interval form as follows:

$$\tilde{A}(\alpha) = [a_1(\alpha), a_2(\alpha)] \quad (2.8)$$

where

$\tilde{A}(\alpha)$ is the fuzzy number at α-level;

$a_1(\alpha)$ is the lower bound of the α-level interval; and

$a_2(\alpha)$ is the upper bound of the α-level interval.

From here, the arithmetic operations of real numbers can be extended to the four main arithmetic operations of fuzzy numbers, i.e. addition, subtraction, multiplication and division. The fuzzy operators of two fuzzy numbers Ã and B̃ are defined at any α-level cut as follows:

$$\tilde{A}(\alpha)\ (+)\ \tilde{B}(\alpha) = [a_1(\alpha)+b_1(\alpha), a_2(\alpha)+b_2(\alpha)] \quad (2.9)$$

$$\tilde{A}(\alpha)\ (-)\ \tilde{B}(\alpha) = [a_1(\alpha)+b_2(\alpha), a_2(\alpha)-b_1(\alpha)] \quad (2.10)$$

$$\tilde{A}(\alpha)\ (*)\ \tilde{B}(\alpha) = [a_1(\alpha)*b_1(\alpha), a_2(\alpha)*b_2(\alpha)] \quad (2.11)$$

$$\tilde{A}(\alpha)\ (/)\ \tilde{B}(\alpha) = [a_1(\alpha)/b_2(\alpha), a_2(\alpha)/b_1(\alpha)] \quad (2.12)$$

Note that for multiplication and division:

$$(\tilde{A}(/)\tilde{B})(*)\ \tilde{B} \neq \tilde{A}\ . \quad (2.13)$$

Also true for addition and substraction:

$$(\tilde{A}(-)\tilde{B})(+)\tilde{B} \neq \tilde{A}\,. \qquad (2.14)$$

2.4.3. Mamdani inference system

Approximate or fuzzy reasoning involves combinations of imprecise logic rules into a single management strategy. Fuzzy logic allows processing of fuzzy data and making decisions based on inaccurate statements and inaccurate data (Ross, 2010). Because of these properties, fuzzy inference approach is used in this work to model the operator's decision making or control actions in a hydropower system.

Following up from Zadeh's approximate reasoning or fuzzy reasoning, team from Queen Mary College, London, UK, led by Mamdani (1974) worked on many applications of approximate reasoning for various industrial systems. Most famous is the fuzzy controller of a steam engine and boiler. The fuzzy controller was based on a set of linguistic control rules obtained from experienced operators. Linguistic rules are representations of human knowledge in IF-THEN rule - based form. Using rule-based simulation, the inference of a conclusion (consequent) given an initially known fact (premise, hypothesis, antecedent) can be made (Ross, 2010). Typical form of IF-THEN rule (also referred to as deductive form) is:

IF *premise (antecedent)*, THEN *conclusion (consequent)* (2.15)

Mamdani inference method is a graphical technique that follows five main steps:

1. Development of fuzzy sets and linguistic rules.
2. Fuzzification of inputs.
3. Application of fuzzy operators.
4. Aggregation of all outputs.
5. Defuzzification of aggregated output.

Step 1. Development of fuzzy sets and linguistic rules

Fuzzy rules represent knowledge and experience of an experienced operator that controls certain system, process, or performs a certain task. Rules are created through interview or observation of the operator at work.

Mamdani form rules may be described by the collection of n linguistic IF-THEN expressions. Following expression shows a rule for the fuzzy inference system with two non - interactive inputs (antecedents) x_1 and x_2 and a single output (consequent) y:

$$\text{IF } x_1 \text{ is } \mathbf{A_1} \text{ AND (OR) } x_2 \text{ is } \mathbf{A_2} \text{ THEN } y \text{ is } \mathbf{B} \tag{2.16}$$

where $\mathbf{A_1}$, $\mathbf{A_2}$ and **B** are the fuzzy sets representing the antecedent pair and consequent. These fuzzy sets may represent fuzzy linguistic concepts such as “large” or “small”, “hot” or “cold” and so forth.

Step 2. Fuzzification of inputs

Inputs to the system, x_1 and x_2 are scalar values. In order to proceed with the inference method, the corresponding degree to which the inputs belong to appropriate fuzzy sets via membership functions needs to be found. Fuzzification of the input thus requires the membership function of the fuzzy linguistic set to be known and through function evaluation the corresponding degree of membership for the scalar input belonging to the universe of discourse is then found. In graphical form:

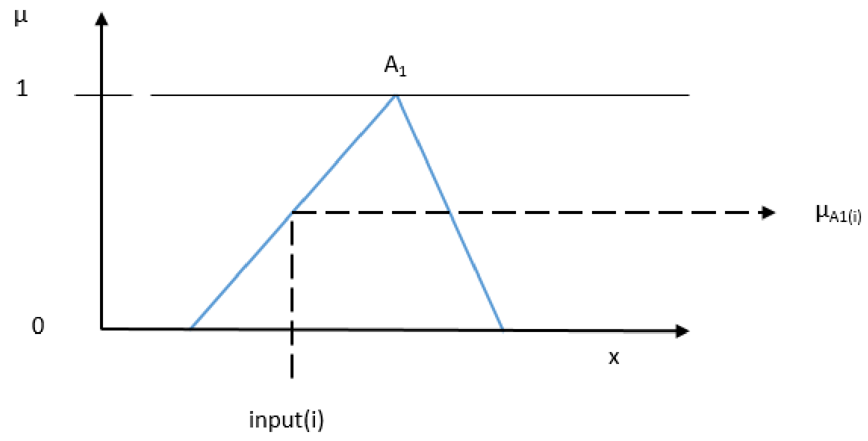


Figure 5. Fuzzification of scalar input from known membership function

Step 3. Application of fuzzy operators

Since there is usually more than one input for a rule, fuzzy operators are used to obtain one number that will represent premise for that rule. That number is applied to output function producing a single truth value for the rule. Usually used logical operators are AND and OR for conjunctive and disjunctive premises. For conjunctive premises we assume new fuzzy subset A_s as intersection:

$$A_S^k = A_1^k \cap A_2^k \qquad \text{for k=1,2, …, r} \tag{2.17}$$

expressed using membership function:

$$\mu_{A_S^k}(x) = \min\left[\mu_{A_1^k}, \mu_{A_2^k}\right] \qquad \text{for k=1,2, …, r.} \tag{2.18}$$

For disjunctive premises we assume a new fuzzy subset A_s as union:

$$A_S^k = A_1^k \cup A_2^k \qquad \text{for k=1,2, …, r} \tag{2.19}$$

expressed using membership function:

$$\mu_{A_S^k}(x) = \max\left[\mu_{A_1^k}, \mu_{A_2^k}\right] \qquad \text{for k=1,2, …, r.} \tag{2.20}$$

Given the above, rule may be rewritten as:

$$\text{IF } A_S^k \text{ THEN } B_s^k \quad \text{for k=1, 2, …, r} \tag{2.21}$$

where r is the number of rules. Graphical representation of operators' application is in Figure 5:

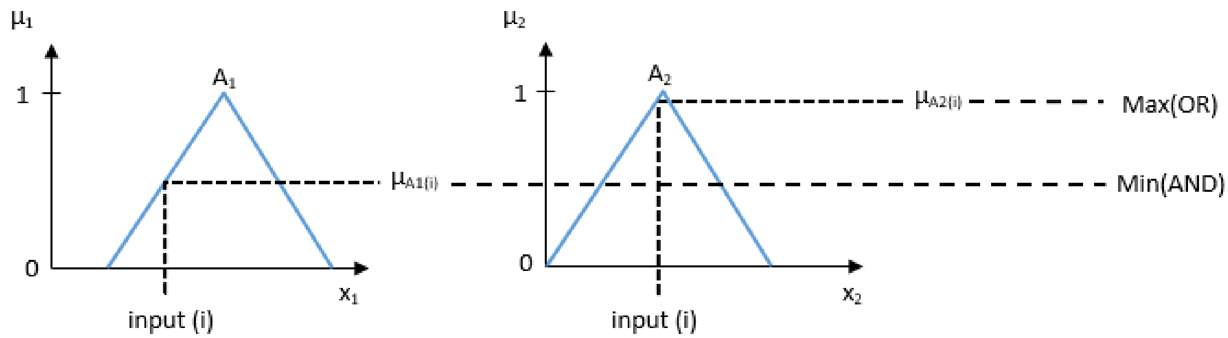


Figure 6. Graphical representation of operators' application

Step 4. Aggregation of outputs

Since it is common for fuzzy inference systems to have more than one rule aggregation of individual consequents contributed by each rule is required, so that that all outputs are combined into a single fuzzy set that may be defuzzified in the final step to obtain a single scalar value.

There are two most often used ways of aggregating outputs, min-max truncation and max-product scaling, and former will be presented. Min-max truncation is the process of propagation of minimum or maximum membership function values from the premises (depending on the operator in each rule) through to the consequent and in doing so truncating the membership function for the consequent of each rule. Then, the truncated membership functions of each rule are combined. That is achieved through the use of disjunctive or conjunctive rules using the same fuzzy operators from the previous step. Disjunctive rules will be applied because of the nature of the inference system. Rules cannot be combined conjunctively. For example, there is a no way to have two states of hydrological data and consequences of those. We can have either one situation or another (see Figure 7). Therefore disjunctive rules are applied in this work.

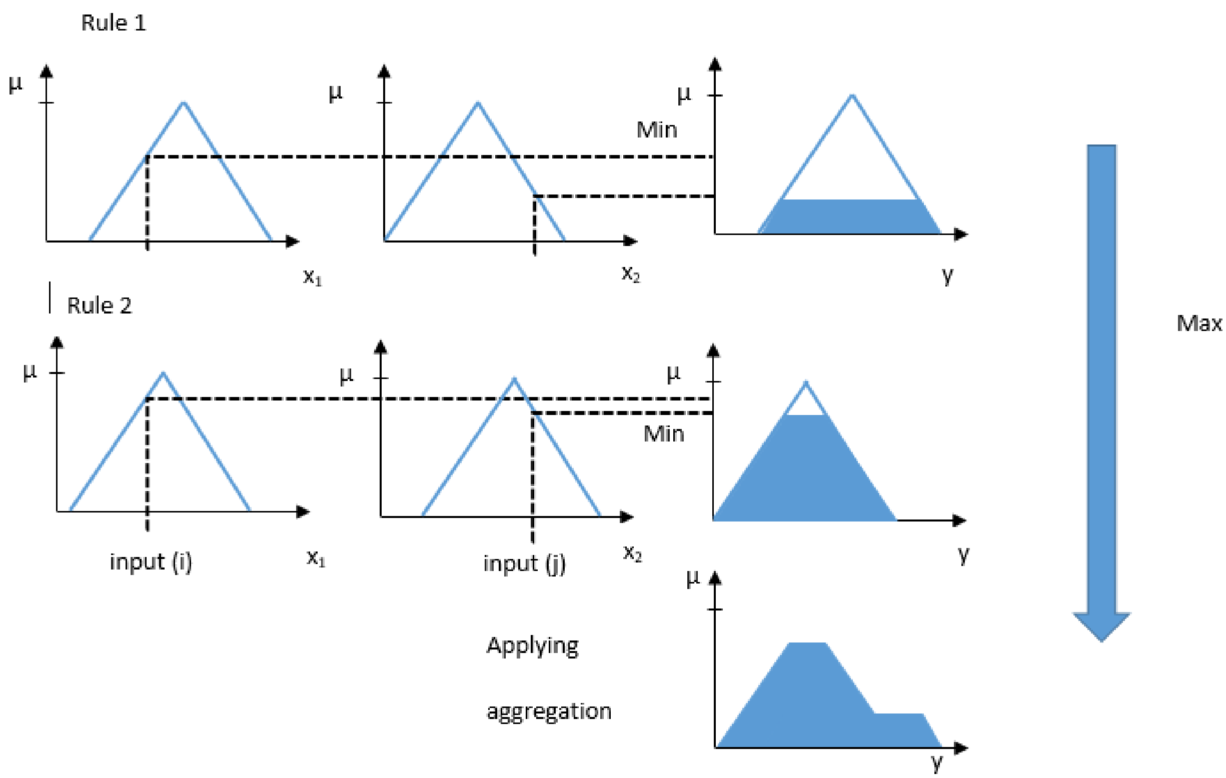


Figure 7. Aggregation of rule outputs into a single fuzzy membership function

Step 5. Defuzzification of aggregated result

The objective of the rule-based system is typically to reach a single value obtained from the defuzzification of the aggregated fuzzy set of all outputs. Defuzzification is the process, or method, of extracting a single value from the aggregated fuzzy set. There are many defuzzification methods: max membership principle, centroid method, weighted average method and many others (Simonović, 2009, Ross 2010, Teodorović, 2012). There is not one most suitable method, depending on the shape of the premise, membership functions and problem under consideration, an appropriate method should be selected. The centroid method is used in this project. It is also referred to as the center of gravity, or center of an area. Its expression is given as:

$$y *= \frac{\int_{xmin}^{xmax} \mu(x) * x \, dx}{\int_{xmin}^{xmax} \mu(x) dx} \tag{2.22}$$

Graphical representation of the centroid method is shown in Figure 5.

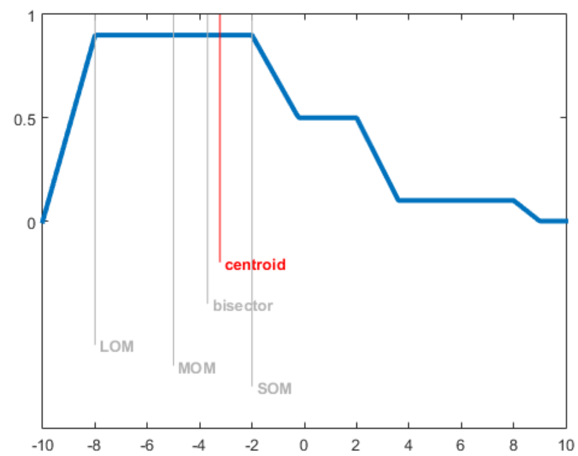


Figure 8. Defuzzification methods - centroid method result in red

An overview of deffuzification methods is available online http://www.mathworks.com/help/fuzzy/examples/defuzzification-methods.html (viewed on 22/1/2016).

3. Methodology

It is documented in the literature that there is a need for automated generation and investigation of scenarios that will describe hazardous states of the system originating from the failure of components, control actions and the combination of the two. Hydropower systems are complex systems that are sensitive to component failures and unsafe control actions that can result in major disasters. Component failures, human behavior and control actions should be evenly investigated in hazard analysis. Consider the two following examples:

- In the 2010 Deepwater Horizon oil spill, a critical factor was that workers reported a successful negative pressure test when in reality oil had already begun seeping into the well. The workers did not know that earlier tests had clogged a pipe that rendered a key instrument reading invalid. Note that in this case the behavior was compliant—the workers followed the required procedures but the procedures were unsafe. The behavior was not a “failure event” because nothing failed – the flaw existed from the beginning in the form of inadequate procedures and feedback for the crew. A failure-based method could help focus engineering efforts on preventing the pipe from getting clogged or perhaps preventing workers from deviating from procedures, but would not help address the flawed requirements and inadequate feedback that existed (Thomas 2012).

- In the 2005 Texas City explosion a critical factor was that operators did not follow standard operating procedures to release hydrocarbons via the 3-pound venting system. Instead, they bypassed the venting system and released hydrocarbons through a blowdown stack into open air. In the absence of any knowledge about the system it might appear that these operators "flipped a coin" to decide whether to follow the procedure, but this is far from true. The decision was a direct result of influence from supervisory personnel who advocated the bypass because it significantly shortened the start-up time and had been used successfully many times in the past (Baker et al. 2007).

System dynamics simulation method is introduced in this work as an implementation tool for STPA. Control actions, which are assigned by the fuzzy inference system, and investigation of the system states are achieved through system dynamics simulation. Scenarios that result in hazard are recorded and written in another Excel file and will be used as input for system safety simulation that will provide system operating conditions and assess them using resilience metric. Resilience is a dynamic quantitative measure of system performance that covers the time from the beginning of an undesirable event to full system recovery from it (Simonović and Peck, 2013).

3.1. Introduction to STPA

In this section of the report, the STPA steps are explained using a generic dam system, formal specification for hazardous control actions (according to Thomas, 2012) and use of system dynamics simulation and fuzzy inference system to automate the generation of hazardous control actions, i.e., the scenarios for causing a hazard are introduced. Assume a dam system that consists of an arch dam with one spillway radial gate. Sensors read water level and relay information to operator's office that is in a house close to the dam. Operator manually controls the gate position. Hoist is used for lifting and lowering the gate. Hoist is powered by electric power from the existing power grid. Populated area is located downstream of the dam and reservoir is used for flood control. Reservoir water level is controlled by planned releases achieved by the operation of the spillway gate.

Before beginning STPA hazard analysis, potential hazards need to be identified. Take for example the previously described dam system Hazards in that simple system include:

- H-1: Dam overtopping and destruction
- H-2: Uncontrolled spill and downstream flooding

STPA Step One: The first step is to identify potentially unsafe control actions for the specific system being considered that can lead to one or more defined system hazards. STPA is performed on a functional control diagram. In this simple system, the control actions could be: open gate, stop opening the gate, close gate, stop closing gate. Control actions can be documented using a table like Table 1.

Table 1: Potentially hazardous control actions for a simple gate controller

Control Action	1. Not given	2. Given incorrectly	3. Wrong timing of order	4. Stopped too soon or applied for too long
Gate open command	Gate not open when the water level is high and inflow is high (H1)	Gate open spilling more than inflow (H2)	Gate open and there is no risk of flooding (H2)	Stopped too soon can lead to (H1)
	Gate not open to release minimum flow requirements (H3)			Applied too long can drain the reservoir and cause (H2)
Gate close command	Gate not closed after flood event is over leading to (H2)	Gate not fully closed leading to unnecessary spilling(may not be hazardous)	Gate closed during regular release. May not be hazardous or hazardous for downstream river ecosystem (H3)	Gate closed too soon when water level is high and peak inflow still has not passed (H1)
				Gate closed too soon but peak inflow passed(may not be hazardous)

STPA Step Two: The second step examines each control loop in the safety control structure to identify potential causal factor for each hazardous control action, i.e., a scenario causing hazard. Figure 9 shows a generic control loop that can be used to guide this step. Step one focused on provided control actions while step two expands the analysis to consider causal factors along the rest of the control loop (Thomas, 2012).

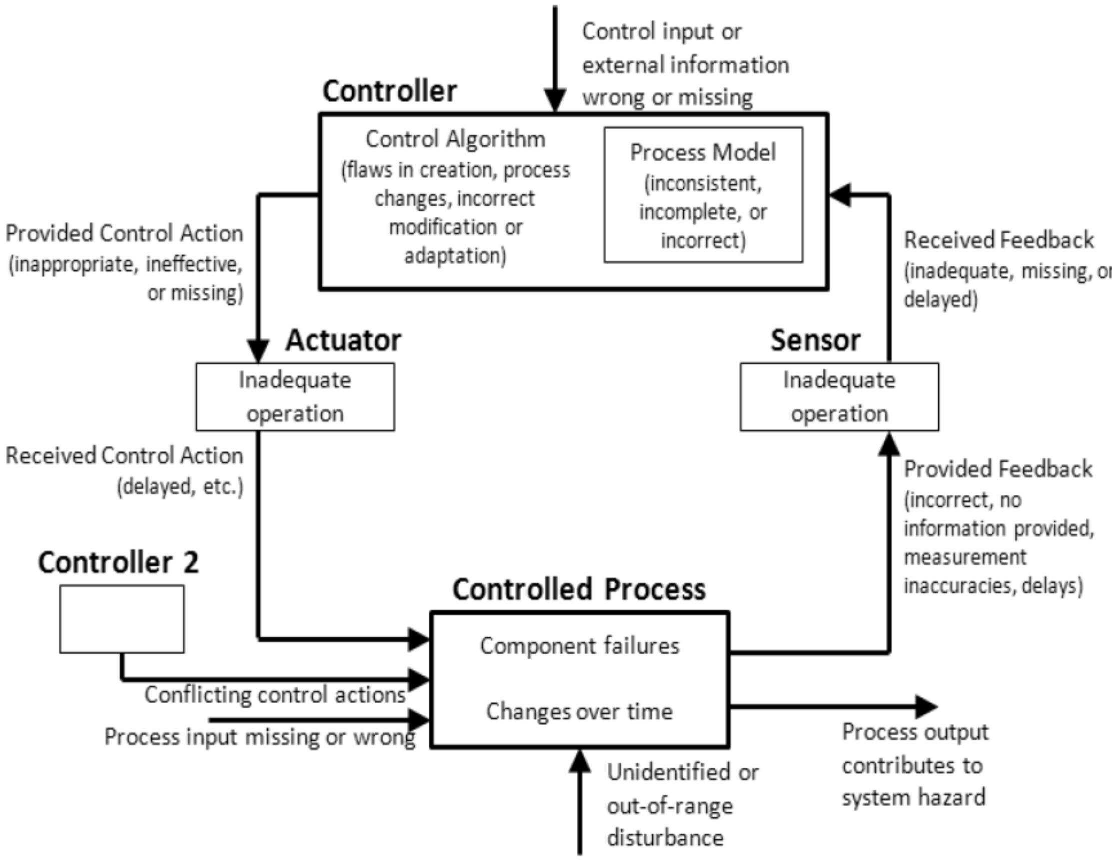


Figure 9. General control loop with causal factors (after Thomas, 2012)

For example, if the gate is closed too soon, one of the causes may be the faulty feedback that controller received from the sensors. Once the second step is over, and potential causes are determined for each hazardous control, they should be eliminated or controlled in the design.

3.2. Formal specification of the hazardous control actions

Thomas provided a formal specification of hazardous control actions that is used during step one of STPA. This specification is used to develop automated algorithm that assists in identifying the actions and generating requirements that enforce safe behaviour (Thomas, 2012). Hazardous control action in the STPA accident model can be expressed formally as a four-tuple (S, T, CA, C) where:

- S is a controller in the system that can issue control actions. The controller may be automated or human.
- T is the type of control action. There are two possible types: *Provided* describes a control action that is issued by the controller while *Not provided* describes a control action that is not issued.
- CA is the control action or command that is output by the controller, like an *Open gate*.
- C is the context in which the control action is or is not provided. Context C is further decomposed into:
    - V- a variable or attribute in the system or environment that may take on two or more values. For example, *water level* and *gate position* are two potential variables for a dam system.
    - VL- a value that can be assumed by a variable. For example, *closed* is a value that can be assumed by the variable *gate position*.
    - CO - a condition expressed as a single variable/value pair. For example, the *gate is closed* is a condition.
- The context C is the combination of one or more conditions and defines a unique state of the system or environment in which a control action may be given.
- To qualify as a hazardous control action, the event (S, T, CA, C) must cause a hazard H ∈ $\boldsymbol{H}$, where $\boldsymbol{H}$ is the set of system level hazards.

Each element of hazardous control action is a member of a larger set, i.e. the following properties must hold:

- S ∈ $\boldsymbol{\mathcal{S}}$, where $\boldsymbol{\mathcal{S}}$ is the set of controllers in the system
- T ∈ $\boldsymbol{\mathcal{T}}$, where $\boldsymbol{\mathcal{T}}$ = {Provided, Not Provided}
- CA ∈ $\boldsymbol{\mathcal{CA}}$ (S), where $\boldsymbol{\mathcal{CA}}$ (S) is the set of control actions that can be provided by controller S
- C ∈ $\boldsymbol{\mathcal{C}}$ (S), where $\boldsymbol{\mathcal{C}}$ (S) is the set of potential contexts for controller S
    - V ∈ $\boldsymbol{\mathcal{V}}$ (S), where $\boldsymbol{\mathcal{V}}$ (S) is the set of variables referenced in the system hazards $\boldsymbol{\mathcal{H}}$
    - VL ∈ $\boldsymbol{\mathcal{VL}}$ (V), where $\boldsymbol{\mathcal{VL}}$ (V) is the set of values that can be assumed by variable V

- $CO = (V, VL) \in \mathcal{CO}$ (S), where $\mathcal{CO}$ (S) is the set of conditions for controller S
- $C = (CO_1, CO_2, ...)$, where each CO_i is independent. That is, no two CO_i refer to the same variable V.

Finally, each hazardous control must be linked to a system-level hazard:

- Event(S, T, CA, C) must cause a hazard $H \in \mathcal{H}$, where $\mathcal{H}$ is the set of system hazards.

Using this formal specification is important for identifying hazardous control actions since the idea is that some actions are only hazardous in certain contexts. For example opening the spillway gate is not hazardous by itself but in a certain context it may be. Therefore, Thomas proposed a procedure that involves identification of potential control actions (presented by S, T, CA), potential hazardous states (presented by context C) and then analyzes that yield a hazardous control action. Using formal specification, a following example (of the previously described system) of the procedure is shown where action is expressed by following four-tuple:

- S = Human
- T = Not provided
- CA = open gate
- C:
  - V = Gate position, Water level, Inflows
  - VL = Closed, Partially open, Fully open, Normal operating range, Above spillway crest, Low, Normal, High
  - CO = Gate is Closed, Gate is Partially open, Gate is Fully open, so forth (each variable gets assigned a value, according to formal specification).

Results can be documented in tabular form. Table 2 shows context for the lack of an open gate control action.

Table 2: Contexts for the lack of an open gate control action

Control Action	Gate position	Water level	Inflows	Hazardous if not provided in this context?
Gate open command not provided	Closed	Above spillway crest	High	Yes
Gate open command not provided	Closed	Above spillway crest	Normal	Yes*
Gate open command not provided	Closed	Above spillway crest	Low	No
Gate open command not provided	Closed	Normal operating range	(does not matter)	No
Gate open command not provided	Partially open	Above spillway crest	High	Yes*
Gate open command not provided	Partially open	Above spillway crest	Normal	No
Gate open command not provided	Partially open	Above spillway crest	Low	No
Gate open command not provided	Partially open	Normal operating range	(does not matter)	No
Gate open command not provided	Fully open	(doesn’t matter)	(does not matter)	No

Values of the variables in the previous example are intentionally provided in verbal form to assist in easy identification are some actions hazardous or not, depending on the context. Control actions that may or may not be hazardous depending on the values behind verbal phrases are marked with an asterisk.

3.3. Implementation approach and programming

While the tabular presentation of actions and contexts is clear, another problem appears. Hydropower dams are complex systems, and high level of detail is needed to achieve proper analysis of the system, hazardous actions and contexts (or scenarios). Therefore, contexts, C, will be automatically generated. To explain further, each system and its components in the control loop, hydrologic data and disturbances are represented by several variables. Each variable, V, can have several values, VL, from two (binary 0 and 1) to multiple values (inflow has several values ranging from 1 to PMF). Sets V_1, V_2, …, V_n (where n is the number of variables) containing their own values are multiplied using Cartesian product to create the all the possible combinations of variables and their respective values, therefore creating all the possible contexts:

$$V_1 = [\, VL_{11}, VL_{12}, \dots, VL_{1m1}] \tag{3.1}$$

$$V_2 = [VL_{21}, VL_{22}, \dots, VL_{2m2}] \tag{3.2}$$

.

.

.

$$V_n = [\, VL_{n1}, VL_{n2}, \dots, VL_{nml}] \tag{3.3}$$

where m_1 is the number of values variable V_1 can assume; m_2 is the number of values variable V_2 can assume; and m_n is the number of values variable V_n can assume.

Following simple combinatorics:

$$|S_1| \cdot |S_2| \cdot \dots \cdot |S_n| = |S_1 \times S_2 \times \dots \times S_n| \tag{3.4}$$

the context is then expressed as:

$$C = |V_1 \times V_2 \times \dots \times V_n| \tag{3.5}$$

To provide the necessary control actions for the procedure, Mamdani fuzzy inference system (FIS) is created which describes operator decisions on how much to open (or close) the gate, depending on the inflow and reservoir water level with guidance not to spill more than inflow. Fuzzy inference systems like this are best created after series of interviews with experienced operators. For the purpose of testing the methodology, FIS has been created on the basis of hydraulic capabilities of the spillway and guidelines for

BC Hydro's operators not to spill more than inflow until peak inflow has passed. FIS consists of rules in the following format:

$$\text{IF } \textit{water level} \text{ is 371 m AND } \textit{inflow} \text{ is 1000 cms, THEN } \textit{gate position} \text{ is 2} \quad (3.6)$$

where membership functions for both inputs (water level and inflow) are triangular functions as shown in Figure 10.

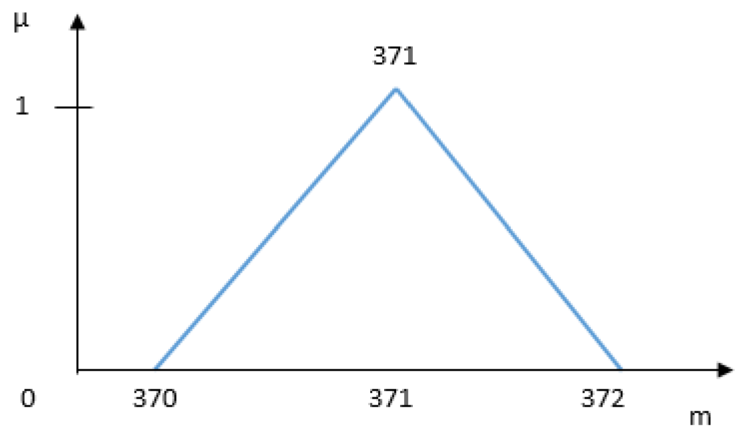


Figure 10. Membership function "371" for water level input of FIS

Membership functions usually have descriptive names, like "low", "medium", "high" but in this context the number just represents closeness to that value. As it can be seen from Figure 10, inputs are not crisp values.

Since system state cannot be assessed from a single moment in time or single context and control action, hydropower system is simulated over several hours (simulation time horizon can be changed), where starting conditions of the simulated system are each of the contexts created. Therefore, depending on the other variables, negative state of one variable will not necessarily mean that system is in a hazardous state. For example, if for some reason gate cannot be opened, or due to faulty sensors operator decides not to open the gate, depending on the water level and inflow no harm may happen to the dam in the following hours. That time might or might not be enough to eliminate the fault, or repair the critical system component. Variables used in the model are shown in Figures 11 and 12.

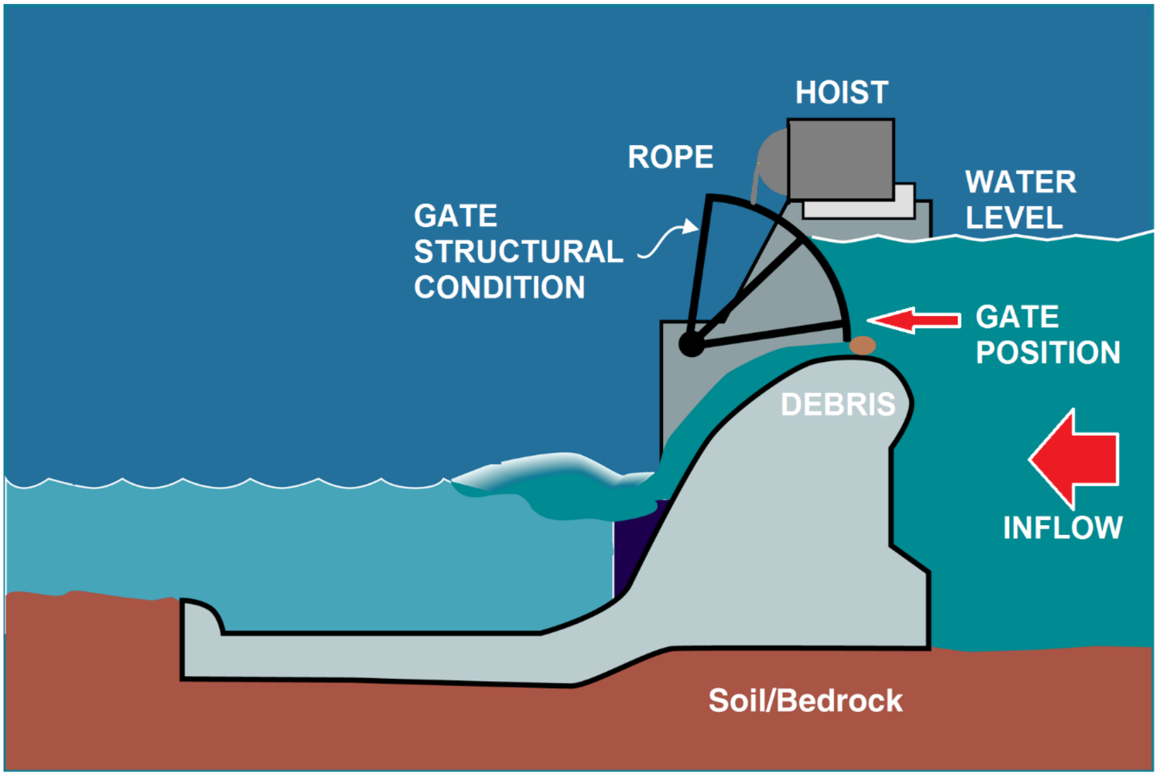


Figure 11. Cross section of a spillway section of a dam with simulation variables (U.S. Army Corps of Engineers, 29/11/2015.)

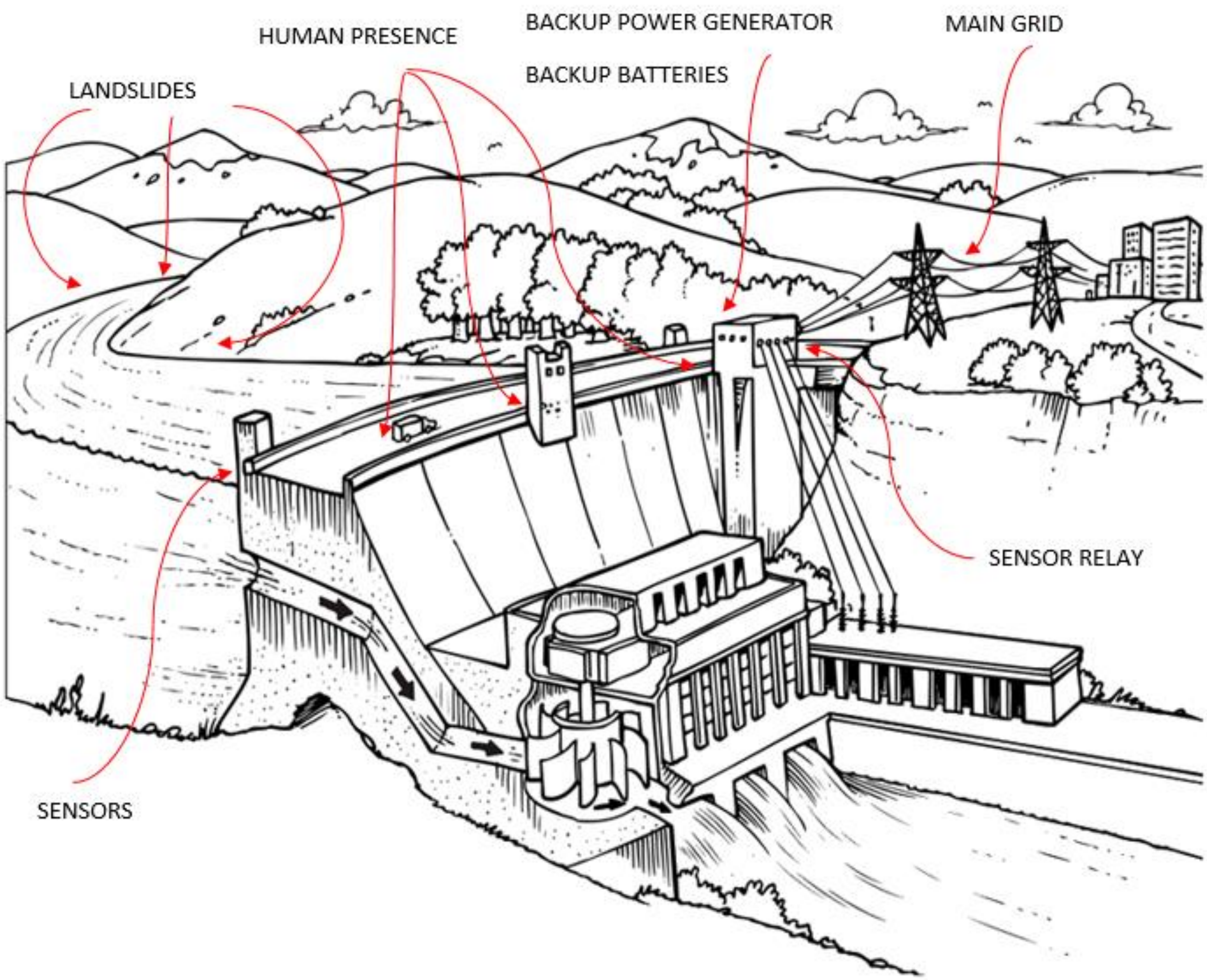


Figure 12. Dam and reservoir diagram with simulation variables (Summit Hydropower, Inc. 2015)

To sum up, the variables used in the presented work are: water level, inflow, gate position, hoist condition, steel cable condition, gate condition, main grid availability, backup power generator availability, backup batteries availability, sensors, sensor relay, human presence, debris and landslide.

3.4. Data

Because of the nature of the system dynamics simulation, a lot of data is needed.

- Hydrologic inflow data. Range of inflows from minimum to maximum inflow (probable maximum flood).

- Hydraulic data. Reservoir storage curve, spillway gate discharge curve, free crest and overtopping spill curves.
- Technical data: Information on actuator systems, power systems, sensors and gates.
- Structural data: Locations of all the system components and their structure
- Geologic data: Landslides existences and their probable mass.

3.5. Simulation

A continuous simulation approach is used for the determination of reservoir storage. Inputs for the simulation are (a) all the variables from the context; (b) storage curve, gate discharge curve and free crest weirs discharge curve; (c) FIS; and (d) simulation time horizon. The simulation time step is 1 hour. Data preparation and simulation flowchart is shown in Figure 14. At the beginning of each simulation step landslide impact is assessed. If the volume of the landslide mass is significant compared to the reservoir size, the dam may be overtopped regardless of the dam state. Therefore simulation may end after landslide impact assessment. If that is not the case, the sensors are then inspected. Availability of the sensor components is inspected because it influences the operator's decisions. Fuzzy inference model based on information from the sensors provides the operator's decision. (Figure 13).

```
if sensors are available then
	if CWL ≥ spillway sill elevation AND IN ≤ maximum gate discharge then
		set gate_position to output of FIS for inputs CWL and IN
	else if CWL ≥ spillway sill elevation AND IN ≥ maximum gate discharge then
		set gate_position to maximum gate elevation
	end
end
```

Figure 13. Pseudocode for sensors inspection and operator's decision (CWL - current water level; IN - inflow)

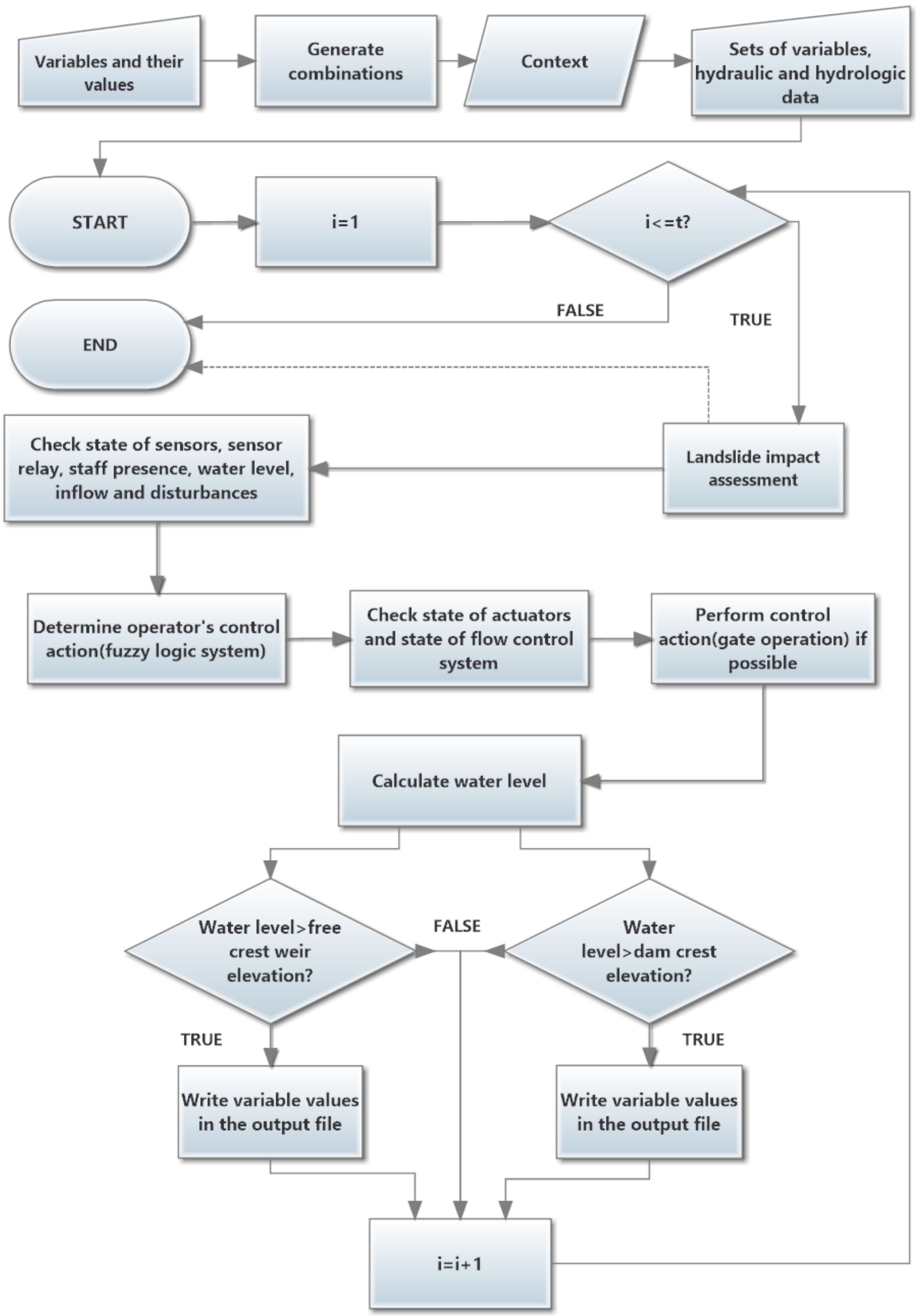


Figure 14. Preparation and simulation flowchart

After the operator's decision is determined, the state of the actuators and flow control system (gate) is investigated. The actuator system is divided into (a) power source: main power grid, backup generators (gasoline or diesel) and backup batteries; (b) mechanical component, the actual hoist mechanism and steel cable that lifts or lowers the gates; and (c) structural component, gate and its training wall and trunnions. State of the actuators and gates is determined and if possible, the issued control action is performed. Using discharge curves (gate discharge, and free crest discharge) spillway and free crest discharge are calculated. Simulation revolves around single equation, the calculation of water level using the continuity equation:

$$V_{i+1} = V_i + 3600 \times (IN_i - GD(wl, go)_{1,i} - FCS(wl)_i) \tag{3.7}$$

where V_{i+1} is the volume of the water in the reservoir in the next time step; V_i is the current volume; IN_i is the inflow; GD is the gate discharge; and FCS is the free crests spill. It is assumed that there are no losses due to infiltration, leakage and evaporation. In the current version of the model, only the gated spill is considered. At the end of each step, if the water level is higher than certain free crest weirs and/or dam crest context is recorded in the output file and simulation ends. The simulation runs until it reaches time horizon or until water level overtops free crest weirs and/or dam and is repeated for every combination of the starting conditions of the system.

Since the water level change is simulated, only the water level and gate position (if possible) variables change throughout the simulation. It is assumed that the other system components states (like sensors or hoist) do not change through the simulation. If it does change, it will be captured in another row of context, and, therefore, nothing is omitted from the final result.

4. Analysis and Results of Cheakamus Dam Case Study

The procedure was tested on a system based on the Cheakamus Dam in British Columbia. Cheakamus Dam is an earth dam with a concrete section where all the outlets are. It has several outlet structures including two 35 ft x 40 ft hoist operated spillway radial gates, lower level outlet gate, a hollow cone valve and three free crest weirs. System has been simplified for model testing by combining radial gates into a single rating curve. Low-level outlet gate and hollow cone are not incorporated in the model. All of the data has been provided by BC Hydro in the following documents: Local Operating Order 3G-CMS-06(Jarl, 2006) and Operations, Maintenance and Surveillance Manual for Dam Safety and Generation Operating Order CMS 4G-25 v2.2 (Oswell, 2009). Hazards that were investigated in this study are:

(H-1): Earth dam overtopping and destruction

(H-2): Uncontrolled spill over three separate free crest weirs (with same crest elevations).

Other hazards, like uncontrolled spill and downstream flooding are not yet incorporated in the model.

The physical layout of the concrete dam is shown in Figures 10 and 11.

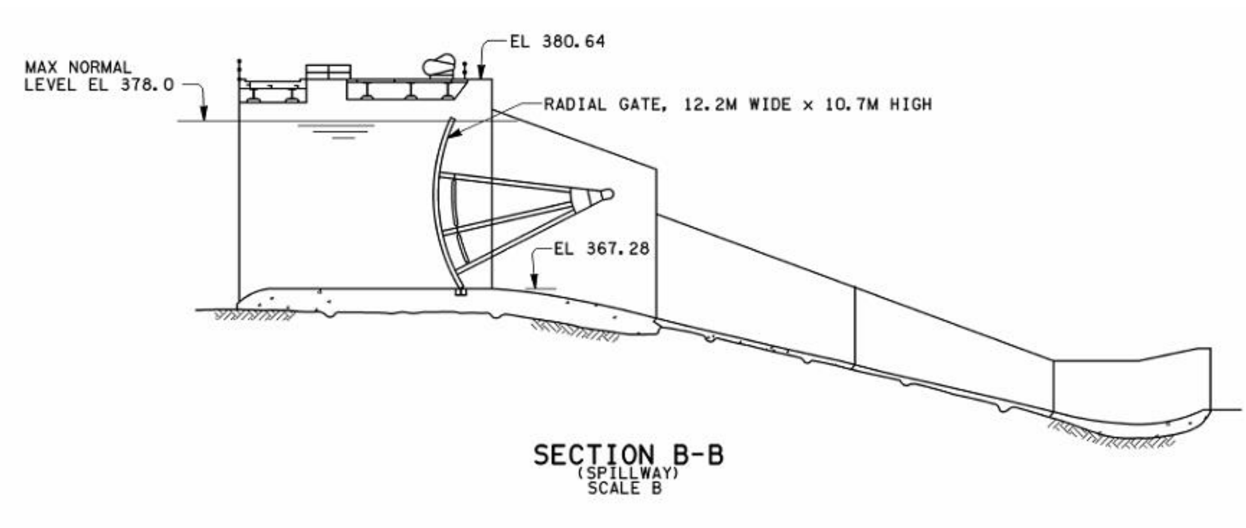


Figure 15. Spillway cross section – Cheakamus Dam (BC Hydro, 2009)

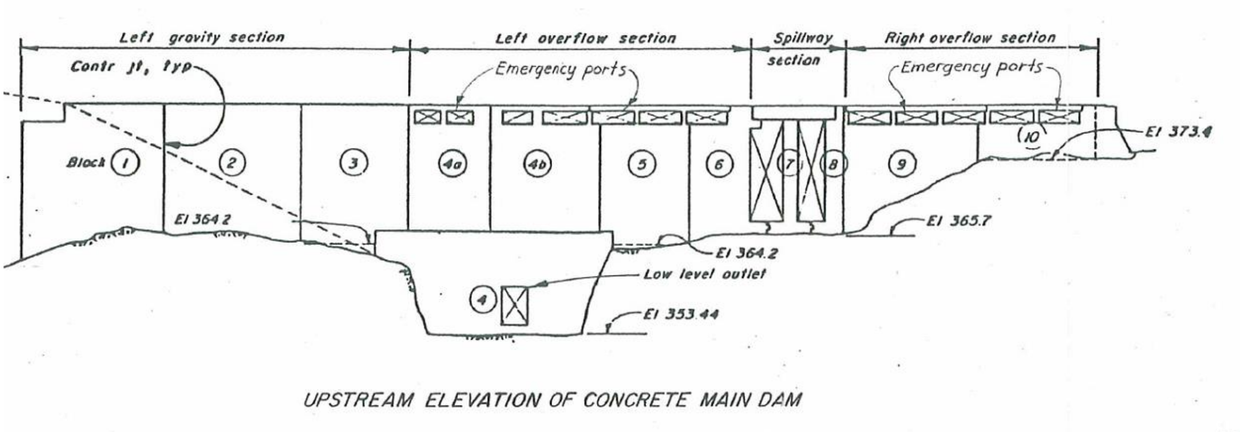


Figure 16. Upstream face of the concrete dam – Cheakamus Dam (BC Hydro, 2009)

Variables and values used in Cheakamus Dam case study are shown in Tables 3 and 4.

Table 3: Variables and values used in Cheakamus case study, part 1

Inflow(cms)	Water level(m)	Gate Position	Sensor state	Debris(m)	Main grid	Diesel generator	Batteries
1	365	0	0	0	0	0	0
100	367	1	1	1	1	1	1
300	369	3	2	2			
500	371	5					
700	373	7					
900	375	9					
1000	377	10					
1500	378	11					
2000	379	12					
2500	380						
3000	381						
3500	381.41						
4000							

Table 4: Variables and values used in Cheakamus case study, part 2

Hoist	Steel Cable	Landslide(m3)	Gate structural condition	Sensor Relay	Stuff Presence
0	0	0	0	0	0
1	1	300000	1	1	1
		15000000	2		
		30000000			

Figure 17. Programming flowchart

Variables and values are stored in a spreadsheet. From these variables and their values, using combination generator (described in the Python code – Appendix A), the full context for STPA is generated. Simulation is done using MATLAB software (Mathworks, 2015) and code is presented in Appendix A. Results are stored in an output spreadsheet. Since product of number of variable values is 19,408,896, that is the number of rows of the context. Part of the context is shown in Table 5.

Table 5: Part of the Cheakamus case study context

IN	WL	GP	Sensors	Debris	MG	DG	BAT	Hoist	Cable	Landslide	GSC	SR	Presence
1	365	0	0	0	0	0	0	0	0	0	0	0	0
1	365	0	0	0	0	0	0	0	0	0	0	0	1
1	365	0	0	0	0	0	0	0	0	0	0	1	0
1	365	0	0	0	0	0	0	0	0	0	0	1	1
1	365	0	0	0	0	0	0	0	0	0	1	0	0
1	365	0	0	0	0	0	0	0	0	0	1	0	1
1	365	0	0	0	0	0	0	0	0	0	1	1	0
1	365	0	0	0	0	0	0	0	0	0	1	1	1
1	365	0	0	0	0	0	0	0	0	0	2	0	0
1	365	0	0	0	0	0	0	0	0	0	2	0	1
1	365	0	0	0	0	0	0	0	0	0	2	1	0
1	365	0	0	0	0	0	0	0	0	0	2	1	1
1	365	0	0	0	0	0	0	0	0	300000	0	0	0
1	365	0	0	0	0	0	0	0	0	300000	0	0	1
1	365	0	0	0	0	0	0	0	0	300000	0	1	0
1	365	0	0	0	0	0	0	0	0	300000	0	1	1

4.1. Justification of the choice of variable values

For some variables, value increment might be significant. The increment value should be selected to provide detailed enough results. For now, one of the limitations is the physical computer memory, simply because of the size of the output data that has to be stored and accessed during simulation. Another important point is that there are simply too many iterations to go through depending on the length of simulation horizon (in hours). Shorter simulation time step may give more accurate results since 1 hour time step may be too big considering the size of the inflow.

Inflow range from minimum of 1 m^3/s (can be changed to zero) to probable maximum flood (PMF) that is according to BC Hydro data 4,129 m^3/s. That number has since the year of 2003 been updated to a range between 2,300 and 2,900 m^3/s. PMF of 4,000 m^3/s is kept as a maximum flow in this case study. Gate position is physically restricted to 12 meters, so the range is from 0 to 12 meters with increments of 1 to 2 meters. It is assumed that gate can be in any position at the start of simulation. Debris is assumed to create an impermeable block at the bottom of the spillway. If debris boom breaks, depending on the season, it is assumed that tree trunks and branches get stuck in the spillway and create an impermeable wall. According to BC Hydro data there have not been records of more than 1 meter of debris getting accumulated in the spillway, so 2 meters of maximum debris blockage is used to be on the safe side. Due to the lack of geologic and geomorphologic data, it is assumed that the landslide affects only the volume in the reservoir and that the whole land mass does not hit the surface of the water too fast (does not create big waves). Landslide volume values are: 0 m^3; 300,000 m^3 (that happened in the 20^{th} century); 15,000,000 m^3 (half of the historical maximum); and 30,000,000 m^3 (the historical maximum which did hit the Cheakamus River area in the 19^{th} century). Availability of power source and mechanical equipment is implemented in binary form, 0 or 1, not available or available. It is assumed that staff can arrive at the site in less than an hour or approximately one hour, if the need arises (for example if the sensor relay is not working). It is also assumed that in additional hour staff can determine rate of rise (of reservoir level) and start controlling the gates manually (from the control station on site). More information is needed on how sensors work and how exactly software for monitoring sensor output is working to improve the system dynamics simulation model. Detailed process flowchart diagram of the model is shown in Appendix B.

4.2. Results and Discussion

Results of the system dynamics simulation are automatically written in an Excel spreadsheet and ready to be used as input for the system safety simulation. The simplified system, presented in this report, can show how components interact and how lack of safe control action might not be always hazardous for the system. Sometimes external disturbance may be too large for the system as it is designed. For example, in the case of the Daisy Lake reservoir, the biggest landslide recorded was 30,000,000 m^3. However, this landslide landed at a location downstream of the today's dam but it is taken into account since all the possible contexts must be created for this system dynamics simulation model. Millions of combination rows or contexts also answer an important question: what happens if something changes during the simulation time (previously assumed that only water level and gate position are changing through simulation)? The answer to that lies in the robustness of this process. All of the physically possible values of the 14 variables are already in the context. If state of a system component changes during simulation, it will be captured in a simulation with different starting conditions. Every possible situation is covered by the range of values used for each of the variables. Tables 6 and 7 present some of the results of the analysis. Free crest spills occurred 12,909,444 times out of 19,408,896. Dam was overtopped 6,524,282 times out of 19,408,896. Further analysis of the results shows that landslide volume has a big impact due to the ration of historical maximum landslide volume to the current reservoir volume.

Table 6: Component states, inputs and disturbances that result in free crest spills

IN	WL	GP	Sensors	Debris	MG	DG	BAT	Hoist	Cable	Landslide	GSC	SR	Presence	Time
100	379	5	2	1	1	0	1	0	1	0	2	1	0	1
100	379	5	2	1	1	0	1	0	1	0	2	1	1	1
100	379	5	2	1	1	0	1	0	1	3.00E+05	0	0	0	1
100	379	5	2	1	1	0	1	0	1	3.00E+05	0	0	1	1
100	379	5	2	1	1	0	1	0	1	3.00E+05	0	1	0	1
100	379	5	2	1	1	0	1	0	1	3.00E+05	0	1	1	1
100	379	5	2	1	1	0	1	0	1	3.00E+05	0	1	1	1

Table 7: Component states, inputs and disturbances that result in dam overtopping

IN	WL	GP	Sensors	Debris	MG	DG	BAT	Hoist	Cable	Landslide	GSC	SR	Presence	Time
500	375	10	1	0	1	0	0	1	1	3.00E+07	0	0	1	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	0	1	0	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	0	1	1	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	1	0	0	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	1	0	1	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	1	1	0	1
500	375	10	1	0	1	0	0	1	1	3.00E+07	1	1	1	1

5. Conclusion and Future Work

This case study illustrates the need for a systems approach to reservoir infrastructure risk assessment. Many hazardous states are the product of an unusual combinations of usual events. The results clearly illustrate that extreme events are not the only source of the hazardous states. System structure, description using identified system components and their interdependencies, together with the identification of the control flaws is of primary importance for the assessment of system safety. Future work will include optimisation of the code, including the transfer of the computer code to a faster computing environment. Implementation of more variables will create even more interdependencies and therefore describe each system in the control loop more accurately. The research results illustrate clearly how complex the reservoir infrastructure systems are and what is the utility of the proposed assessment method. It is important to note that in spite our best efforts, there will always be an unforeseen external disturbance that cannot be easily incorporated in the model.

5. Acknowledgements

We would like to thank BC Hydro for providing the data used in this project. Special thanks to Professor Slobodan P. Simonović for supervising this project. Special thanks to Mr. Desmond Hartford and Mr. Derek Sakamoto for their much appreciated input, suggestions and help.

6. References

Baecher, G., Ascila, R., and Hartford, D. N. D. (2013). “Hydropower and dam safety.” *STAMP/STPA Workshop*.

Baker, J., Leveson, N., Bowman, F., and Priest, S. (2007). *The report of the BP US refineries independent safety review panel*. *Rapport technique*.

BC Hydro. (2005). *Cheakamus Project Water Use Plan Cheakamus Project Water Use Plan*. Vancouver, British Columbia.

Department Of Defense USA. (1980). “Procedures For Performing A Failure Mode, Effects And Critical Analysis.” *Military Standard*, 2072(August), 80.

Dulac, N. (2007). “A Framework for Dynamic Safety and Risk Management Modeling in Complex Engineering Systems.” *PhD Thesis*, Massachusetts Institute of Technology.

Hartford, D. N. D., and Baecher, G. B. (2004). *Risk and Uncertainty in Dam Safety*. Thomas Telford, London.

Komey, A., Deng, Q., Baecher, G. B., Zielinski, P. A., and Atkinson, T. (2015). “Systems Reliability of Flow Control in Dam Safety.” *12th International Conference on Application of Statistics and Probability in Civil Engineering, ICASP12*, 1–8.

Kong, G. (2013). *Generation Operating Order*. Vancouver, British Columbia.

Leveson, N. G. (2011). *Engineering a Safer World: Systems Thinking Applied to Safety*. *Vasa, The MIT PRESS*, Cambridge, Massachussets.

Mamdani, E. (1974). “Application of Fuzzy Algorithms for Control of Simple Dynamic Plant.” *IEEE 212*, 1585–1588.

Mathworks®, (2015). *Fuzzy Logic Toolbox™: User's Guide (R2015b).* Retrieved February 8, 2016 from http://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf

Mathworks®, (2015) MATLAB® R2015b. http://www.mathworks.com/downloads/

Mathworks®, (2015). *MATLAB® Primer (R2015b).* Retrieved February 8, 2016 from http://www.mathworks.com/help/pdf_doc/matlab/getstart.pdf

Oswell, M. T. (2009). *Cheakamus Dam: Operation, Maintenance and Surveillance Manual for Dam Safety*. Vancouver, British Columbia.

Putcha, C. S., and Patev, R. C. (2000). "Investigation of Risk Assessment Methodology for Dam Gates and Associated Operating Equipment." (November), 55.

Python.org. (2015) Python 3.4.4 Release https://www.python.org/downloads/release/python-344/

Python.org. (2015) History and Licence 3.4.4 Release https://docs.python.org/3.4/license.html

Rasmussen, N. C. (1975). *Reactor safety study. An assessment of accident risks in U. S. commercial nuclear power plants. Executive Summary. WASH-1400 (NUREG-75/014)*.

Regan, P. J. (2010). "Dams as systems - A holistic approach to dam safety." *USSD Annual Meeting and Conference*, Sacramento, California, 1307–1340.

Ross, T. J. (2010). *Fuzzy logic with engineering applications*. John Wilay & Sons, Ltd., Chichester, West Sussex, United Kingdom.

Simonovic, S. P. (2009). *Managing Water Resources: Methods and Tools for a Systems Approach*. Earthscan, London, UK.

Simonovic, S. P., and Peck, A. (2013). "Dynamic Resilience to Climate Change Caused Natural Disasters in Coastal Megacities Quantification Framework." *British Journal of Environment and Climate Change*, 3(3), 378–401.

Teodorović, D., and Šelmić, M. (2012). *Računarska inteligencija u saobraćaju*. University of Belgrade, Faculty of Transport and Traffic Engineering, Belgrade, Serbia.

Thomas, J. (2012). *Extending and Automating a Systems- Theoretic Hazard Analysis for Requirements Generation and Analysis*. Albuquerque, New Mexico.

Thornberry, C. (2014). "Extending the Human-Controller Methodology in Systems- Theoretic Process Analysis." Massachusetts Institute of Technology.

UNISDR. (2009). "UNISDR Terminoology on Disaster Risk Reduction." *International Stratergy for Disaster Reduction (ISDR)*, 1–30.

Vucetic, D., and Simonovic, S. P. (2011). *Water resources decision making under uncertainty (Report No: 073)*. London, Ontario, Canada.

Watson, H. A. (1961). *Launch Control Safety Study*. Bell Labs, Murray Hill, New Jersey.

Wood, S. (2009). *Cheakamus Local Operating Order*. Vancouver, British Columbia.

Zadeh, L. A. (1973). "Outline of a new approach to the analysis of complex systems and decision processes." *IEEE Trans. Systems, Man and Cybernetics*, 3: 28–44.

Zimmerman, H. J. (2001). *Fuzzy set theory and its applications - Fourth Edition*. Kluwer, Boston, 2nd ed., 1993.

Appendix A: Context Generation and Simulation Model Code

Python code for creating combinations of variables:

```
"""
Created on Tue Jul 14 12:33:26 2015
"""
from itertools import product
import csv


#Read in csv file
criteria = []
with open("Failure Modes.csv", 'rb') as f:
    for i,row in enumerate(csv.reader(f)):
        if i == 0:
            header = row
        else:
            criteria.append(row)
#Filter spaces and separate columns
criteria = [filter(lambda x: x != '', row) for row in zip(*criteria)]
#Unpack criteria and take cartesian product
combos = product(*criteria)
#Write out combos iterator and retain original column names
with open("Failure Combos.csv", 'wb') as f:
    writer = csv.writer(f)
    writer.writerow(header)
    for c in combos:
        writer.writerow(c)
```

MATLAB simulation code

```
function [count1, count2] = STPAcomb7( GO, WL, IN, gate_opening, discharge,
reservoir_elevation, storage, storagelvl,fcrl,fcd,t,sensors, debris, MG, DG,
BAT, Hoist, Rope, Landslide, FuzzyGP,gate_str, sr, presence)
F=scatteredInterpolant(gate_opening, reservoir_elevation, discharge);
filename='output10.csv';
maxit=19408896;
iter=1;
counter1 = 0;
counter2 = 0;
c1wl=(-1)*(ones(15000000,1));
c1in=(-1)*(ones(15000000,1));
c1go=(-1)*(ones(15000000,1));
c1sen=(-1)*(ones(15000000,1));
c1deb=(-1)*(ones(15000000,1));
c1mg=(-1)*(ones(15000000,1));
c1dg=(-1)*(ones(15000000,1));
c1bat=(-1)*(ones(15000000,1));
c1hoist=(-1)*(ones(15000000,1));
c1rope=(-1)*(ones(15000000,1));
c1land=(-1)*(ones(15000000,1));
c1gstr=(-1)*(ones(15000000,1));
c1sr=(-1)*(ones(15000000,1));
c1pres=(-1)*(ones(15000000,1));
c2wl=(-1)*(ones(15000000,1));
c2in=(-1)*(ones(15000000,1));
c2go=(-1)*(ones(15000000,1));
c2sen=(-1)*(ones(15000000,1));
c2deb=(-1)*(ones(15000000,1));
c2mg=(-1)*(ones(15000000,1));
c2dg=(-1)*(ones(15000000,1));
c2bat=(-1)*(ones(15000000,1));
c2hoist=(-1)*(ones(15000000,1));
c2rope=(-1)*(ones(15000000,1));
c2land=(-1)*(ones(15000000,1));
c2gstr=(-1)*(ones(15000000,1));
c2sr=(-1)*(ones(15000000,1));
c2pres=(-1)*(ones(15000000,1));
time1=(-1)*(ones(15000000,1));
time2=(-1)*(ones(15000000,1));
for i=1:maxit
    ly1=0;
    ly2=0;
    x=1;
    cwl=WL(iter);
    cwv=interp1(storagelvl, storage, cwl) + Landslide(iter);
    if cwv>=61214400
        ly2=1;
    end
    if cwv>=45463680
        ly1=1;
    end
```

```
cwv2=min(cwv, 61214400);
cwl=interp1(storage, storagelvl, cwv2);
y1=0;
y2=0;
for c=1:t
    cwv=interp1(storagelvl, storage, cwl);
    if (MG(iter)==0) && (DG(iter)==0) && (BAT(iter)==0)
        GP=GO(iter);
        GP2=max(GP-debris(iter), 0);
    else
    if sr(iter)==1
    if sensors(iter)==1
            if (cwl>367.28+debris(iter)) && (cwl<=382) && (IN(iter)<=1600)
                GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                if presence(iter)==0
                    GP2=max(GP-debris(iter), 0);
                else
                    GP2=min(12, GP+debris(iter));
                end
            elseif (cwl>367.28) && (IN(iter)>1600)
                GP2=12-debris(iter);
            end
    elseif (sensors(iter)==0) && (presence(iter)==1) && (c>=2)
           if (cwl>367.28+debris(iter)) && (cwl<=382) && (IN(iter)<=1600)
                GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                GP2=min(GP+debris(iter), 12);
           elseif (cwl>367.28) && (IN(iter)>1600)
                GP2=12-debris(iter);
           end
    elseif (sensors(iter)==0) && (presence(iter)==0) && (c<=2)
        GP=GO(iter);
        GP2=max(GP-debris(iter), 0);
    elseif (sensors(iter)==0) && (presence(iter)==0) && (c>=3)
        if (cwl>367.28+debris(iter)) && (cwl<=382) && (IN(iter)<=1600)
                GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                GP2=min(GP+debris(iter), 12);
           elseif (cwl>367.28) && (IN(iter)>1600)
                GP2=12-debris(iter);
        end
    elseif (sensors(iter)==0) && presence(iter)==1 && (c==1)
        GP=GO(iter);
        GP2=max(GP-debris(iter),0);
    elseif (sensors(iter)==2) && (c==1)
            if (cwl>367.28) && (cwl<=368.28)
                GP2=0;
            elseif (cwl>368.28) && (cwl<=382) && (IN(iter)<1600)
                k=cwl-1;
                GP=evalfis([(k-debris(iter)) IN(iter)], FuzzyGP);
                if presence(iter)==0
                    GP2=max(GP-debris(iter), 0);
                else
                    GP2=min(12, GP+debris(iter));
                end
            elseif (cwl>369.28) && (IN(iter)>1600)
                GP2=12-debris(iter);
            end
    elseif (sensors(iter)==2) && (c>=2)
```

```
            if (cwl>367.28+debris(iter)) && (cwl<=382) && (IN(iter)<=1600)
                 GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                 if presence(iter)==0
                     GP2=max(GP-debris(iter), 0);
                 else
                     GP2=min(12, GP+debris(iter));
                 end
             elseif (cwl>367.28) && (IN(iter)>1600)
                 GP2=12-debris(iter);
            end
     end
     elseif sr(iter)==0
         if presence(iter)==1
             if sensors(iter)==1
                 if    (cwl>367.28+debris(iter))    &&    (cwl<=382)    &&
(IN(iter)<=1600)
                     GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                     GP2=min(12, GP+debris(iter));
                 elseif (cwl>367.28) && (IN(iter)>1600)
                     GP2=12-debris(iter);
                 end
             elseif (sensors(iter)==0) && (c==1)
                 GP=GO(iter);
                 GP2=max(GP-debris(iter), 0);
             elseif (sensors(iter)==0) && (c>=2)
                 if    (cwl>367.28+debris(iter))    &&    (cwl<=382)    &&
(IN(iter)<=1600)
                     GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                     GP2=min(12, GP+debris(iter));
                 elseif (cwl>367.28) && (IN(iter)>1600)
                     GP2=12-debris(iter);
                 end
             elseif (sensors(iter)==2) && (c==1)
                  if (cwl>367.28) && (cwl<=368.28)
                     GP2=0;
                  elseif (cwl>368.28) && (cwl<=382) && (IN(iter)<1600)
                     k=cwl-1;
                     GP=evalfis([(k-debris(iter)) IN(iter)], FuzzyGP);
                     GP2=min(GP+debris(iter), 12);
             elseif (cwl>369.28) && (IN(iter)>1600)
                     GP2=12-debris(iter);
                  end
             elseif (sensors(iter)==2) && (c>=2)
                 if    (cwl>367.28+debris(iter))    &&    (cwl<=382)    &&
(IN(iter)<=1600)
                     GP=evalfis([(cwl-debris(iter)) IN(iter)], FuzzyGP);
                     GP2=min(GP+debris(iter), 12);
                 elseif (cwl>367.28) && (IN(iter)>1600)
                     GP2=12-debris(iter);
                 end
             end
             elseif presence(iter)==0
                 if c<=2
                     GP=GO(iter);
                     GP2=max(GP-debris(iter), 0);
                 end
                 if c>=3
```

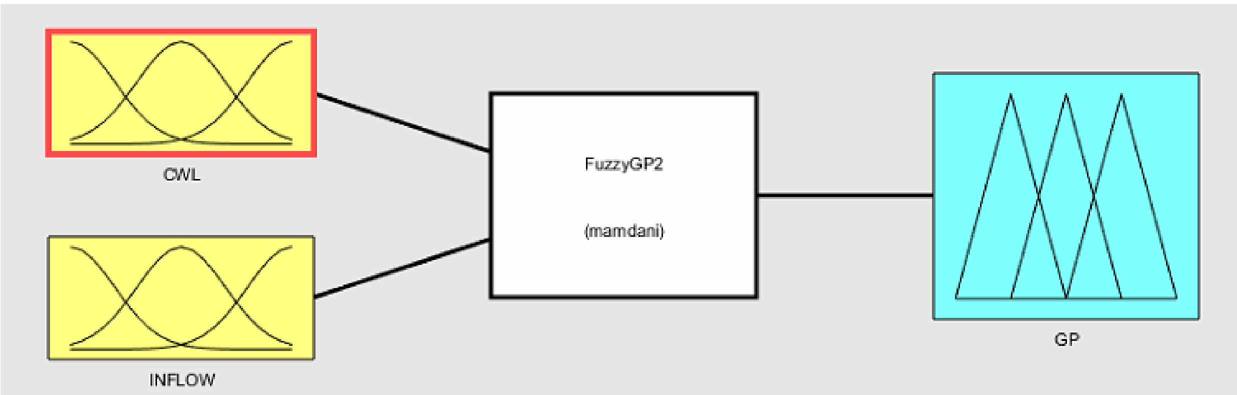
```
                         if    (cwl>367.28+debris(iter))    &&    (cwl<=382)    &&
(IN(iter)<=1600)
                             GP=evalfis([(cwl-debris(iter))          IN(iter)],
FuzzyGP);
                             GP2=min(GP+debris(iter), 12);
                         elseif (cwl>367.28) && (IN(iter)>1600)
                             GP2=12-debris(iter);
                         end
                     end
             end
         end
         end
                 if cwl<378.41
                     overflow=0;
                  else
                     overflow=interp1(fcrl, fcd, cwl);
                 end
                  if cwl<367.28
                      outflow=0;
                  else
                      if gate_str(iter)==1
                         if  ((MG(iter)==0)  &&  (DG(iter)==0)  &&  (BAT(iter)==0))
||  (Hoist(iter)==0)
                             outflow=F(GO(iter), cwl);
                         elseif (Rope(iter)==0)
                             outflow=0;
                         else
                             outflow=F(GP2, cwl);
                         end
                      elseif gate_str(iter)==0
                          outflow=0;
                      elseif gate_str(iter)==2
                          outflow=F(GO(iter), cwl);
                      end
                  end
                     newwv=min((cwv+3600*(IN(iter)-outflow-overflow)),
61214400);
                     cwl2=interp1(storage, storagelvl, newwv);
                     cwl=cwl2;
                 if (cwl>378.41 && y1==0) || (ly1==1)
                     counter1=counter1+1;
                     c1wl(iter,1)= WL(iter);
                     c1go(iter,1)= GO(iter);
                     c1in(iter,1)= IN(iter);
                     c1sen(iter,1)=sensors(iter);
                     c1deb(iter,1)= debris(iter);
                     c1mg(iter,1)=MG(iter);
                     c1dg(iter,1)= DG(iter);
                     c1bat(iter,1)=BAT(iter);
                     c1hoist(iter,1)= Hoist(iter);
                     c1rope(iter,1)= Rope(iter);
                     c1land(iter,1)= Landslide(iter);
                     c1gstr(iter,1)= gate_str(iter);
                     c1sr(iter,1)= sr(iter);
                     c1pres(iter,1)= presence(iter);
                     time1(iter,1)= x;
                     y1=1;
```

```
                    ly1=2;
                end
                if (cwl>381.42 && y2==0) || (ly2==1)
                    counter2=counter2+1;
                    c2wl(iter,1)=WL(iter);
                    c2go(iter,1)=GO(iter);
                    c2in(iter,1)= IN(iter);
                    c2sen(iter,1)= sensors(iter);
                    c2deb(iter,1)= debris(iter);
                    c2mg(iter,1)= MG(iter);
                    c2dg(iter,1)= DG(iter);
                    c2bat(iter,1)= BAT(iter);
                    c2hoist(iter,1)= Hoist(iter);
                    c2rope(iter,1)= Rope(iter);
                    c2land(iter,1)= Landslide(iter);
                    c2gstr(iter,1)= gate_str(iter);
                    c2sr(iter,1)= sr(iter);
                    c2pres(iter,1)= presence(iter);
                    time2(iter,1)= x;
                    y2=1;
                    ly2=2;
                end
                c=c+1;
                x=x+1;
    end
    iter=iter+1;
end
count1=counter1
count2=counter2
c1wl=c1wl(c1wl(:,1)>-1);
c1in=c1in(c1in(:,1)>-1);
c1go=c1go(c1go(:,1)>-1);
c1sen=c1sen(c1sen(:,1)>-1);
c1deb=c1deb(c1deb(:,1)>-1);
c1mg=c1mg(c1mg(:,1)>-1);
c1dg=c1dg(c1dg(:,1)>-1);
c1bat=c1bat(c1bat(:,1)>-1);
c1hoist=c1hoist(c1hoist(:,1)>-1);
c1rope=c1rope(c1rope(:,1)>-1);
c1land=c1land(c1land(:,1)>-1);
c1gstr=c1gstr(c1gstr(:,1)>-1);
c1sr=c1sr(c1sr(:,1)>-1);
c1pres=c1pres(c1pres(:,1)>-1);
time1=time1(time1(:,1)>-1);
c2wl=c2wl(c2wl(:,1)>-1);
c2in=c2in(c2in(:,1)>-1);
c2go=c2go(c2go(:,1)>-1);
c2sen=c2sen(c2sen(:,1)>-1);
c2deb=c2deb(c2deb(:,1)>-1);
c2mg=c2mg(c2mg(:,1)>-1);
c2dg=c2dg(c2dg(:,1)>-1);
c2bat=c2bat(c2bat(:,1)>-1);
c2hoist=c2hoist(c2hoist(:,1)>-1);
c2rope=c2rope(c2rope(:,1)>-1);
c2land=c2land(c2land(:,1)>-1);
c2gstr=c2gstr(c2gstr(:,1)>-1);
c2sr=c2sr(c2sr(:,1)>-1);
```

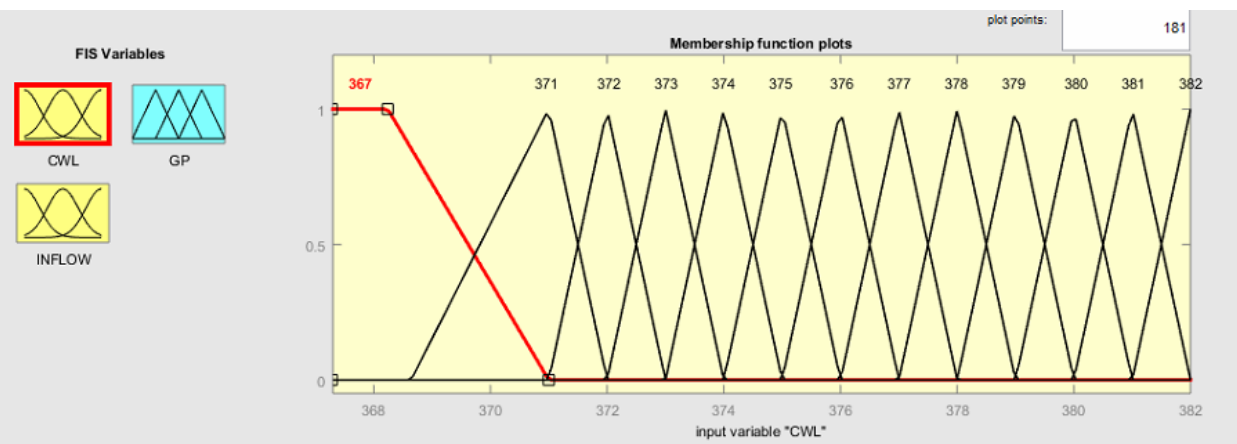
```
c2pres=c2pres(c2pres(:,1)>-1);
time2=time2(time2(:,1)>-1);
CO1=[c1in c1wl c1go c1sen c1deb c1mg c1dg c1bat c1hoist c1rope c1land c1gstr
c1sr c1pres time1];
CO2=[c2in c2wl c2go c2sen c2deb c2mg c2dg c2bat c2hoist c2rope c2land c2gstr
c2sr c2pres time2];
csvwrite(filename, CO1);
csvwrite('output11.csv', CO2);
end
```

Appendix B: Fuzzy Inference System

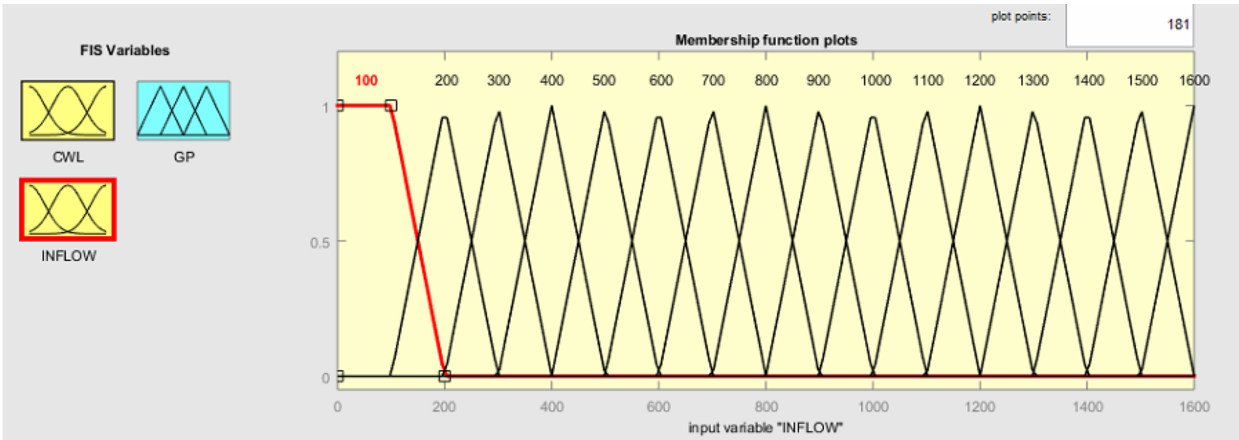
FIS creator application in MATLAB:



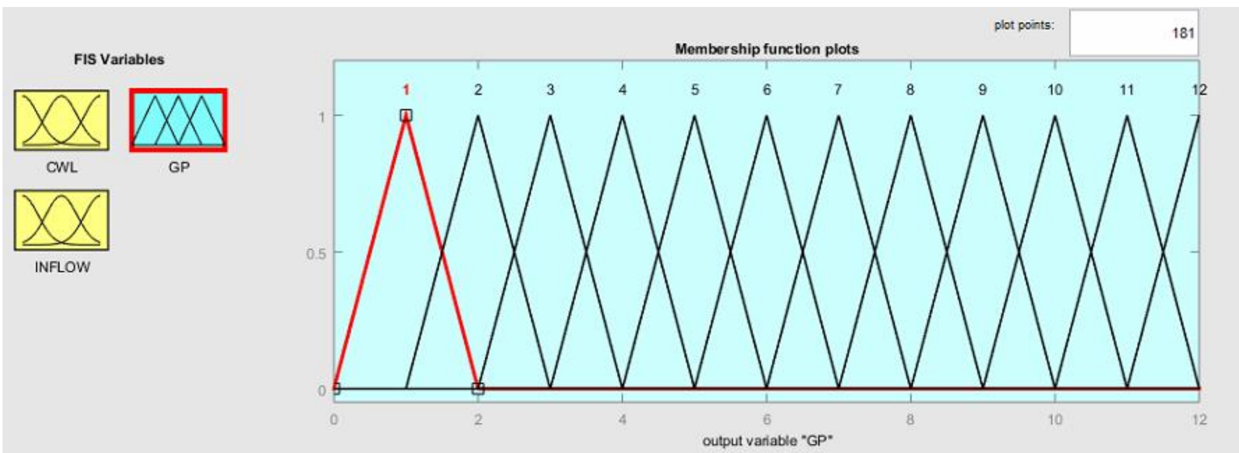
Current water level membership functions:



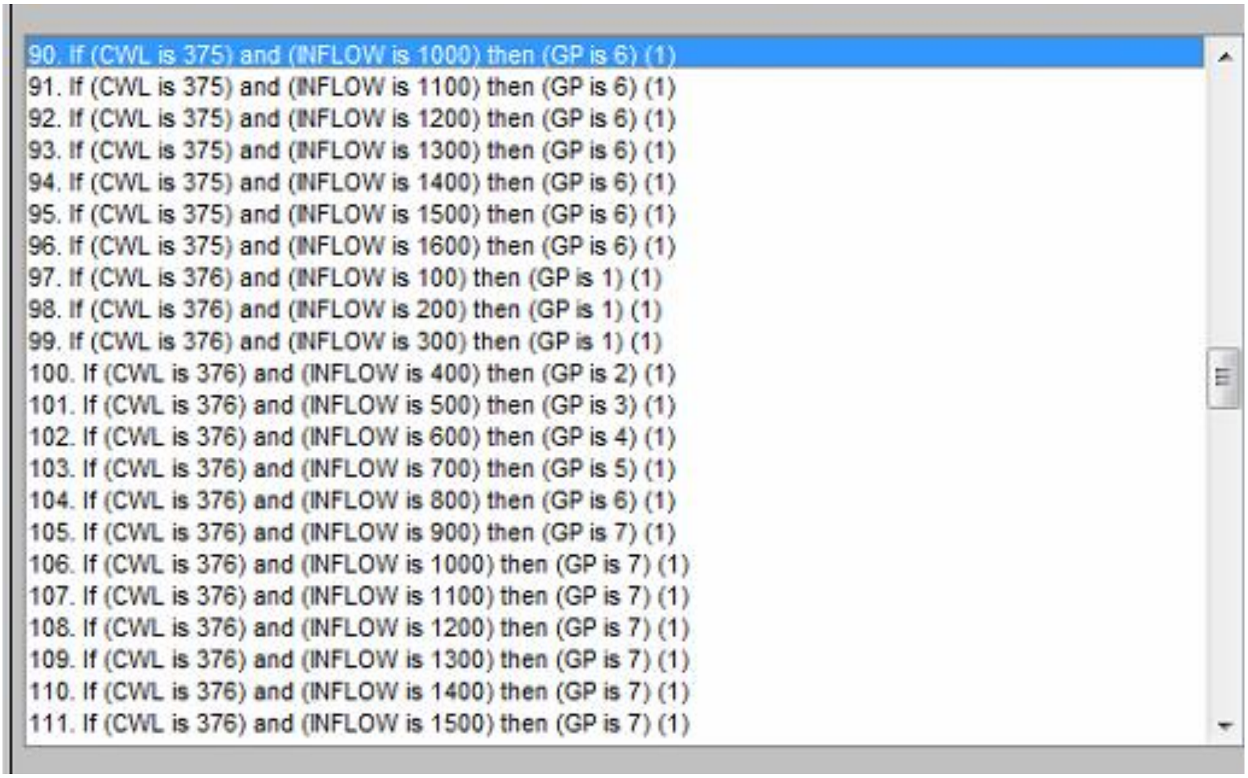
Inflow membership functions:



Gate position membership functions:



List of some of the rules:


90. If (CWL is 375) and (INFLOW is 1000) then (GP is 6) (1)
91. If (CWL is 375) and (INFLOW is 1100) then (GP is 6) (1)
92. If (CWL is 375) and (INFLOW is 1200) then (GP is 6) (1)
93. If (CWL is 375) and (INFLOW is 1300) then (GP is 6) (1)
94. If (CWL is 375) and (INFLOW is 1400) then (GP is 6) (1)
95. If (CWL is 375) and (INFLOW is 1500) then (GP is 6) (1)
96. If (CWL is 375) and (INFLOW is 1600) then (GP is 6) (1)
97. If (CWL is 376) and (INFLOW is 100) then (GP is 1) (1)
98. If (CWL is 376) and (INFLOW is 200) then (GP is 1) (1)
99. If (CWL is 376) and (INFLOW is 300) then (GP is 1) (1)
100. If (CWL is 376) and (INFLOW is 400) then (GP is 2) (1)
101. If (CWL is 376) and (INFLOW is 500) then (GP is 3) (1)
102. If (CWL is 376) and (INFLOW is 600) then (GP is 4) (1)
103. If (CWL is 376) and (INFLOW is 700) then (GP is 5) (1)
104. If (CWL is 376) and (INFLOW is 800) then (GP is 6) (1)
105. If (CWL is 376) and (INFLOW is 900) then (GP is 7) (1)
106. If (CWL is 376) and (INFLOW is 1000) then (GP is 7) (1)
107. If (CWL is 376) and (INFLOW is 1100) then (GP is 7) (1)
108. If (CWL is 376) and (INFLOW is 1200) then (GP is 7) (1)
109. If (CWL is 376) and (INFLOW is 1300) then (GP is 7) (1)
110. If (CWL is 376) and (INFLOW is 1400) then (GP is 7) (1)
111. If (CWL is 376) and (INFLOW is 1500) then (GP is 7) (1)

Appendix C: Flowchart Diagram

The following is presentation of detailed flow chart for the procedure developed in the presented work:

Notation:

CWV, CWL-current water volume and current water level

MG, DG, BAT-main grid availability, diesel generator availability and batteries availability

GP-gate position (calculated)

GO-gate opening, the starting position, given in the context

GS-structural condition of the gate

SR-sensor relay availability

Evalfis and FuzzyGP-evalfis – MATLAB functions that return output for the given input;

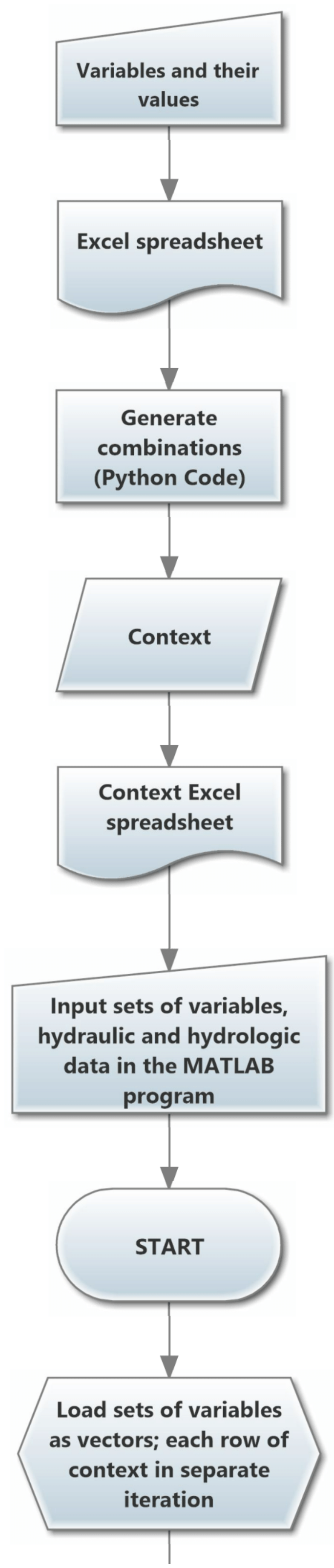
FuzzyGP - the fuzzy inference system (developed in MATLAB)

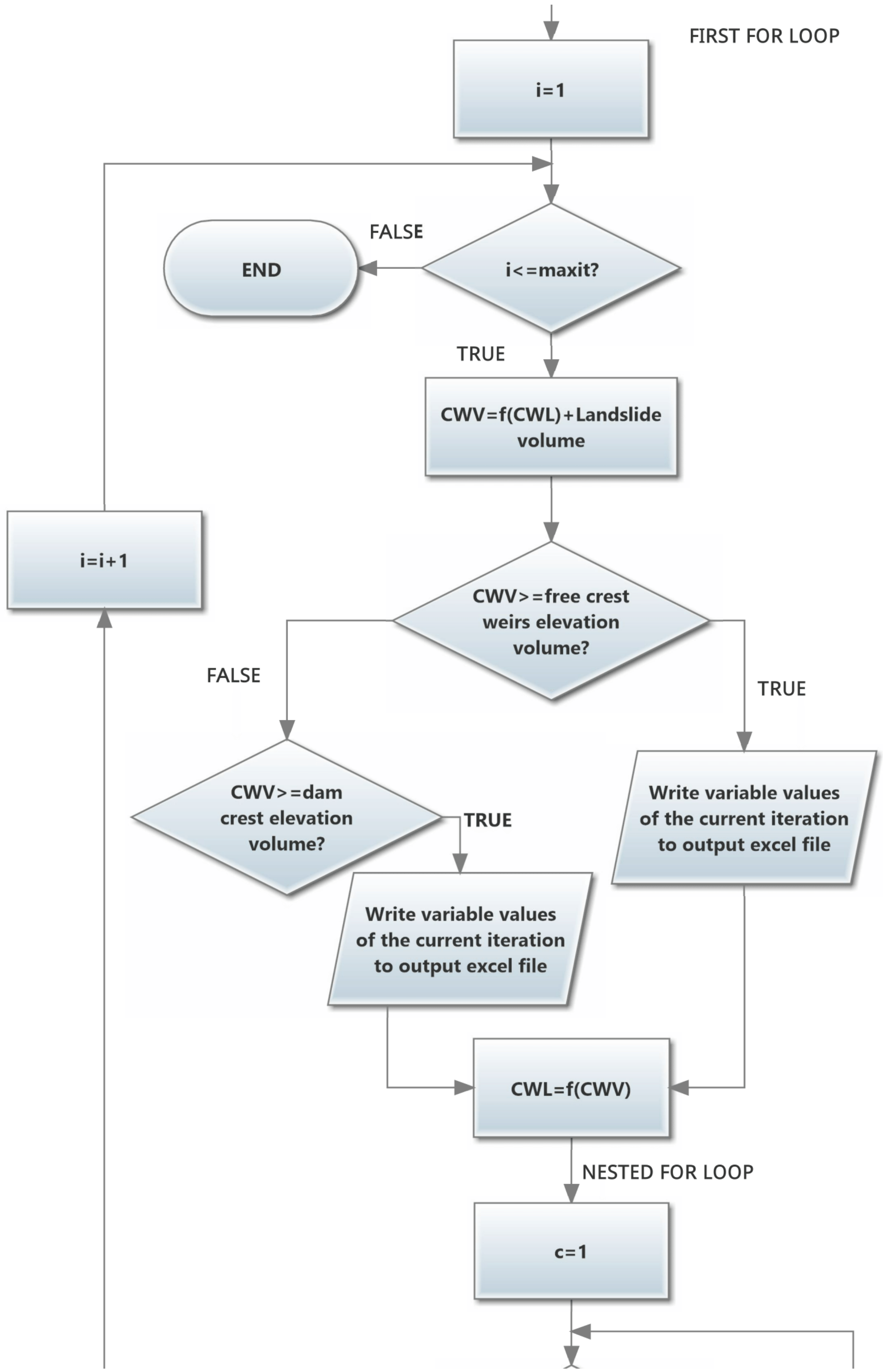
NEWWV-calculated reservoir water volume in the next time step

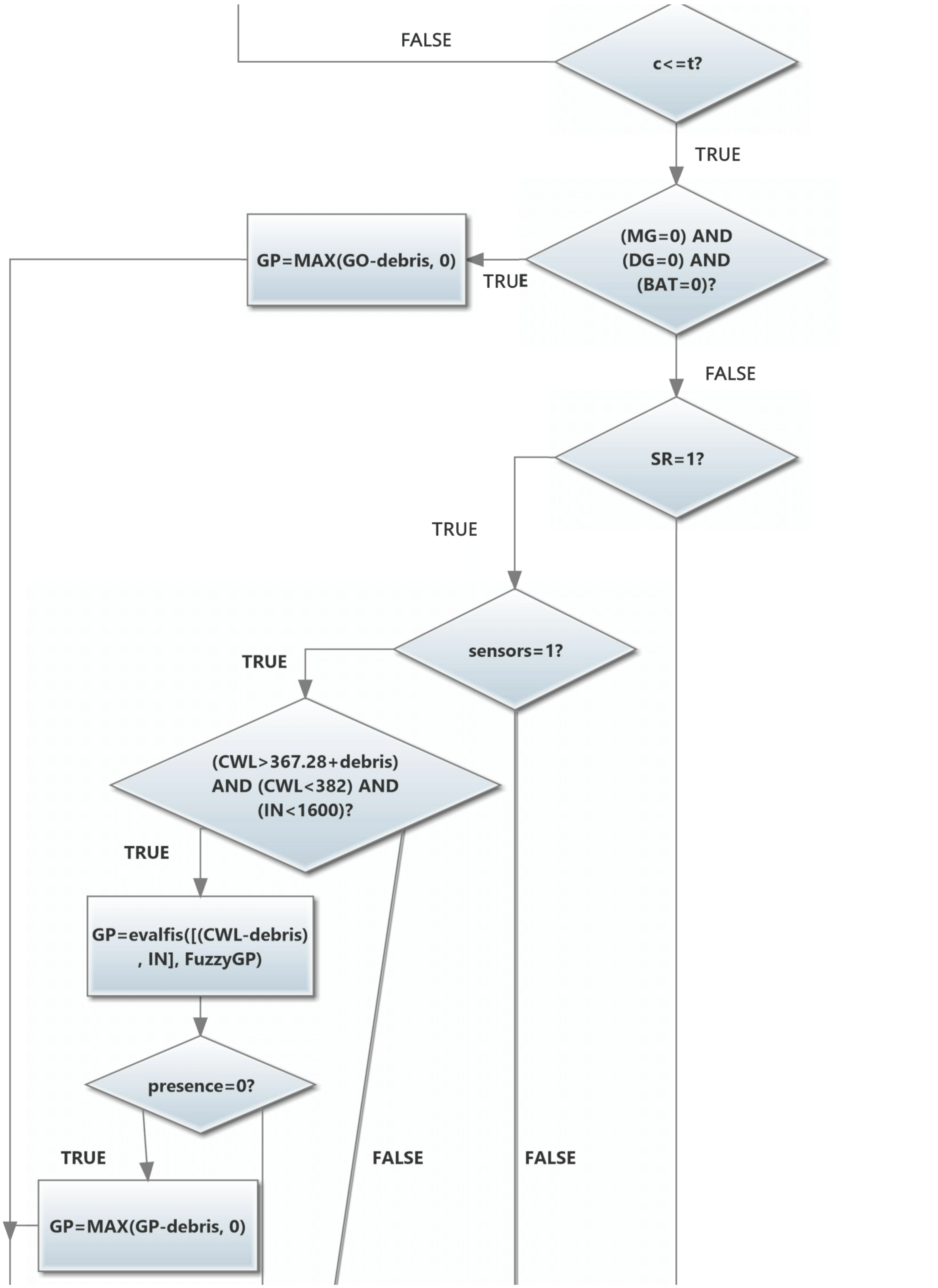
i, c - loop iteration counters

maxit – the number of context rows, and

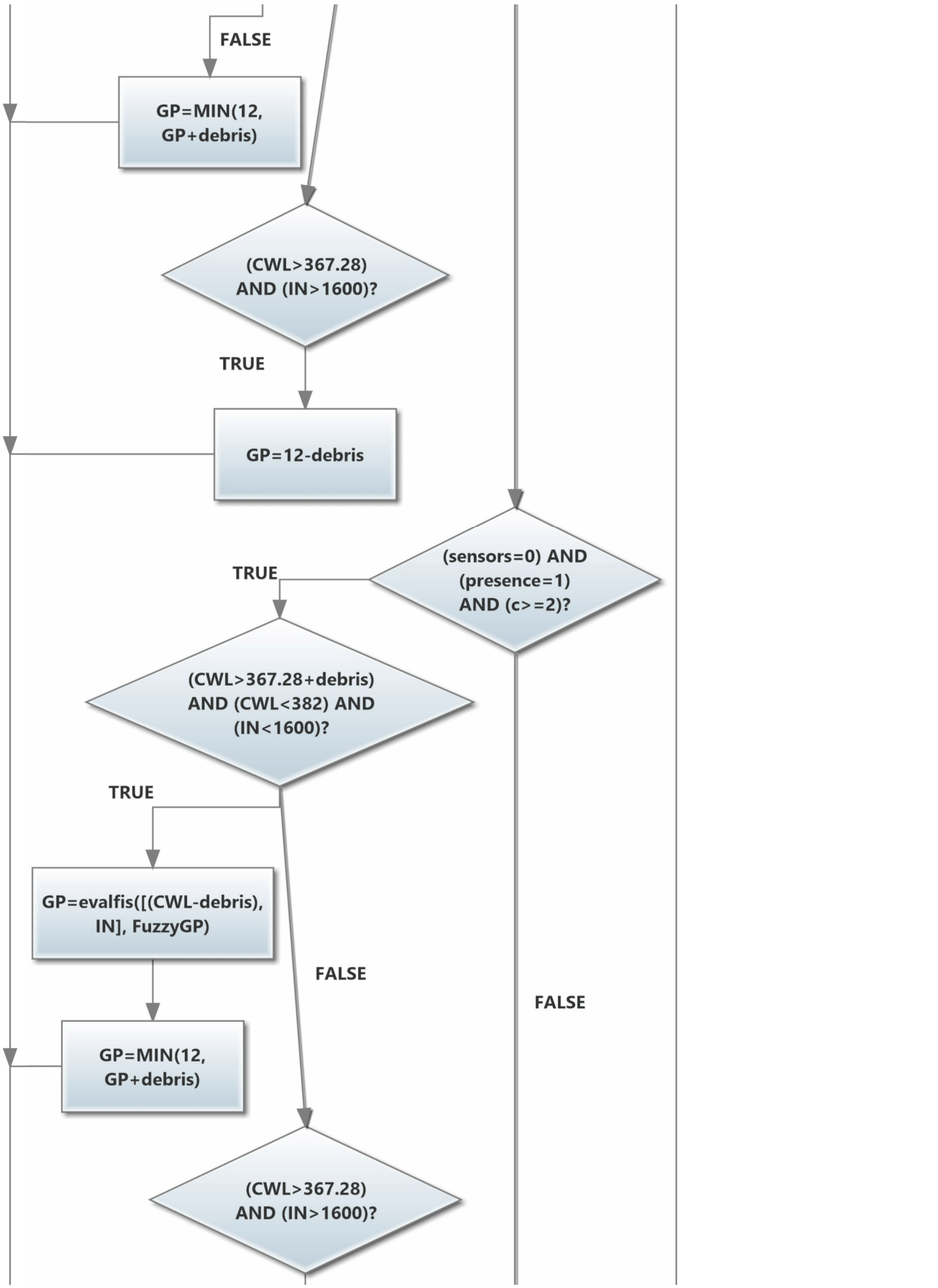
t - is time (maximum number of simulation time steps, given by the user).

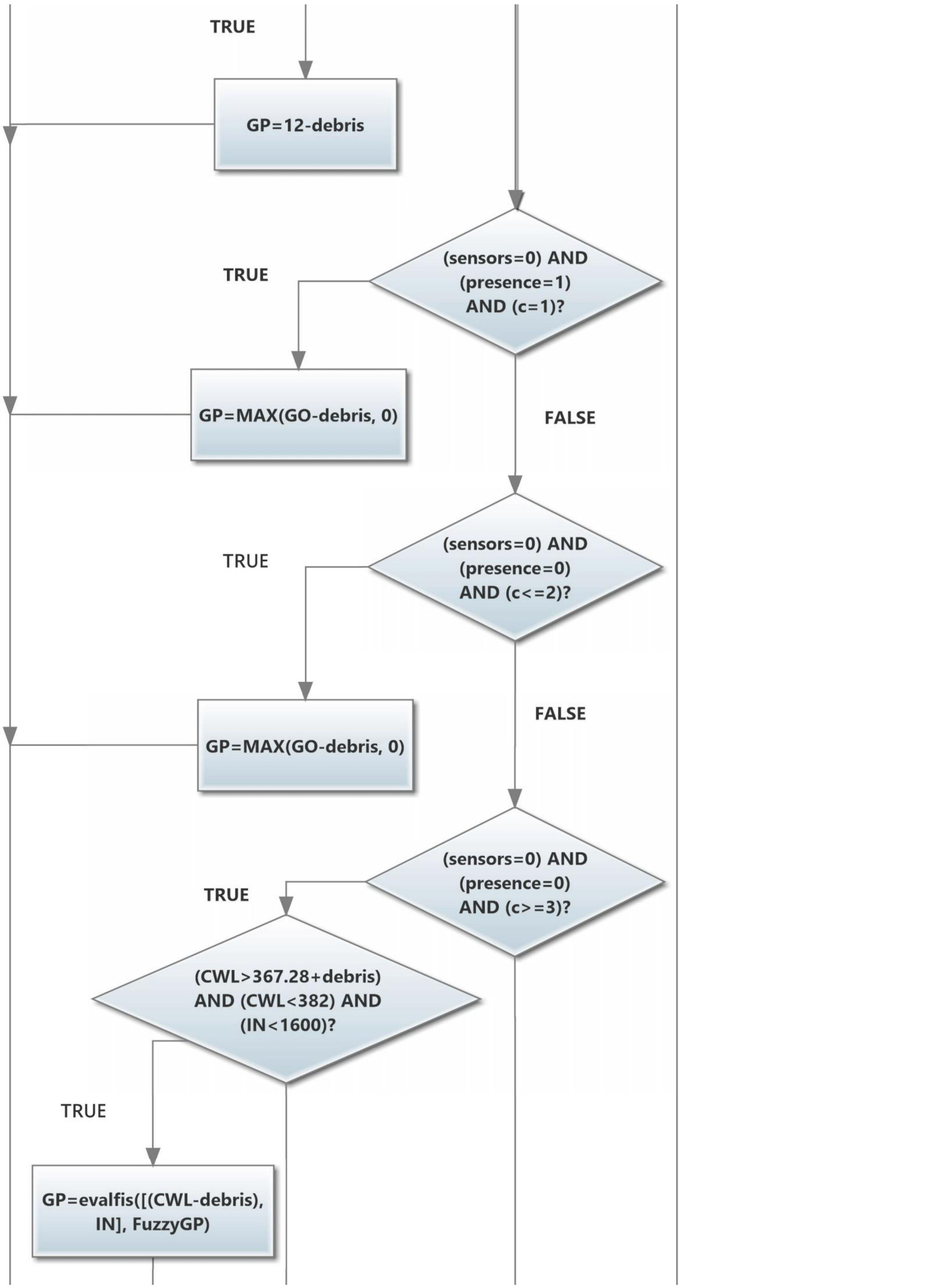

Variables and their values
Excel spreadsheet
Generate combinations (Python Code)
Context
Context Excel spreadsheet
Input sets of variables, hydraulic and hydrologic data in the MATLAB program
START
Load sets of variables as vectors; each row of context in separate iteration

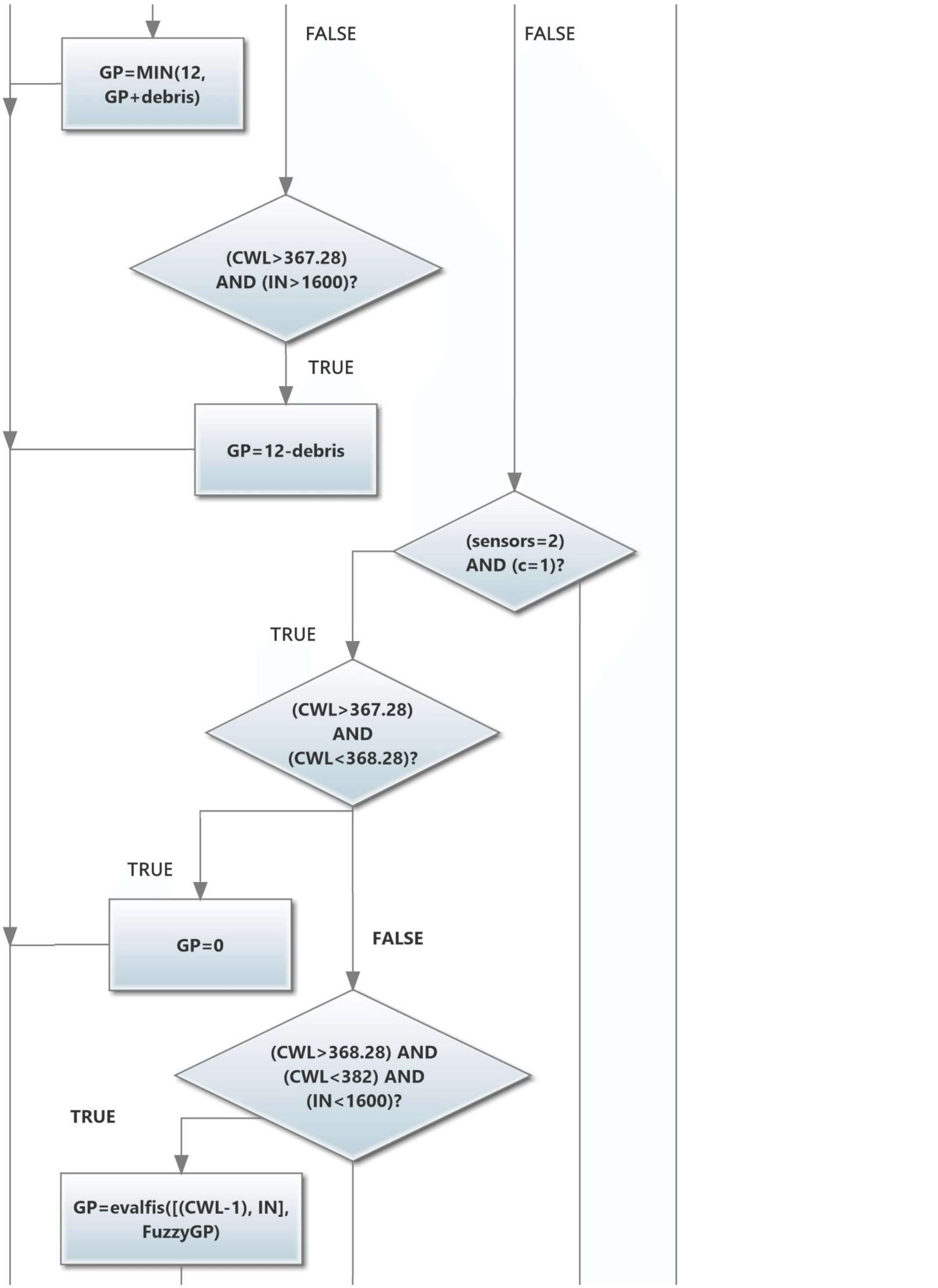

FIRST FOR LOOP
i=1
FALSE
END
i<=maxit?
TRUE
CWV=f(CWL)+Landslide volume
i=i+1
CWV>=free crest weirs elevation volume?
FALSE
TRUE
CWV>=dam crest elevation volume?
TRUE
Write variable values of the current iteration to output excel file
Write variable values of the current iteration to output excel file
CWL=f(CWV)
NESTED FOR LOOP
c=1

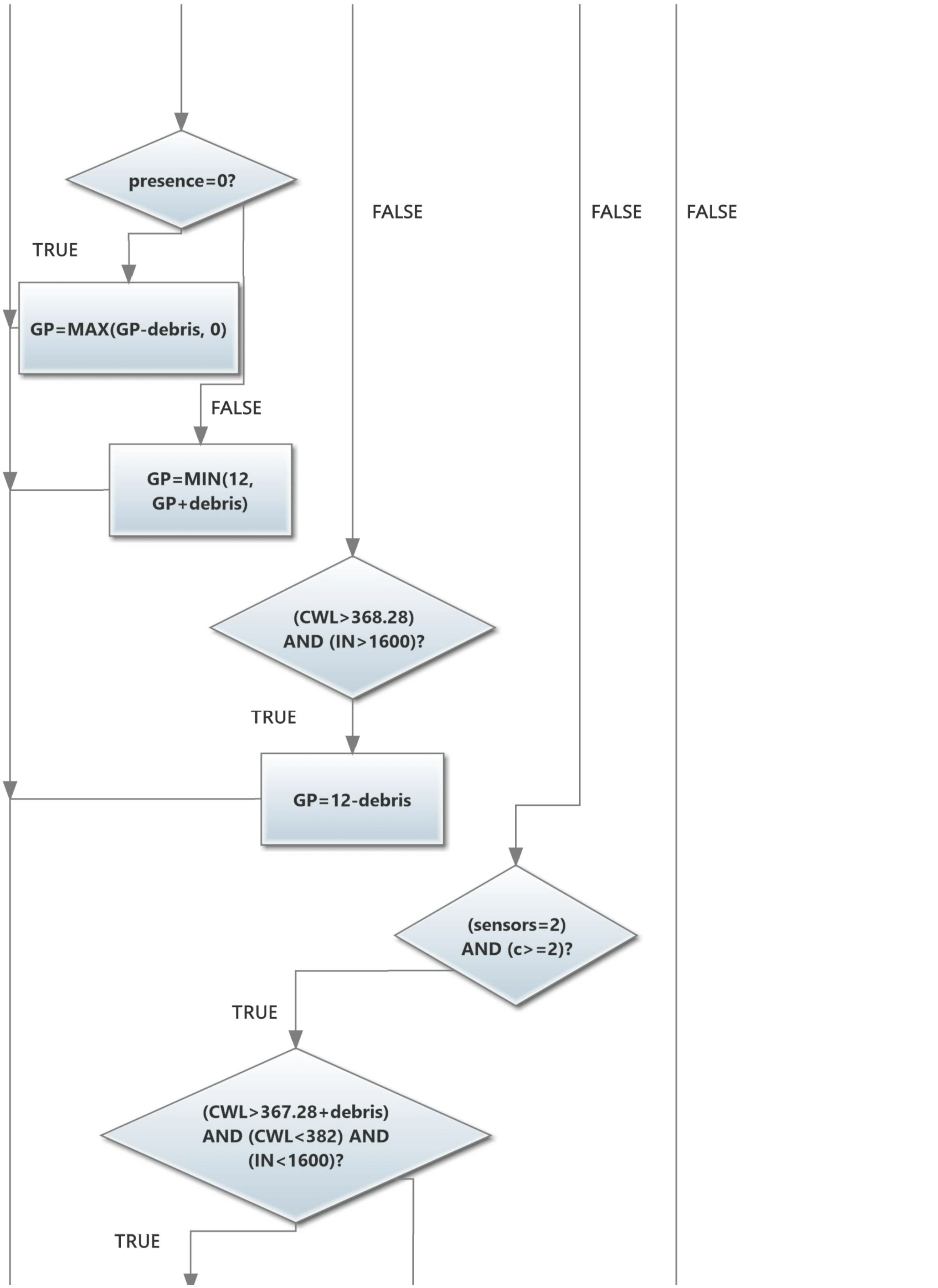


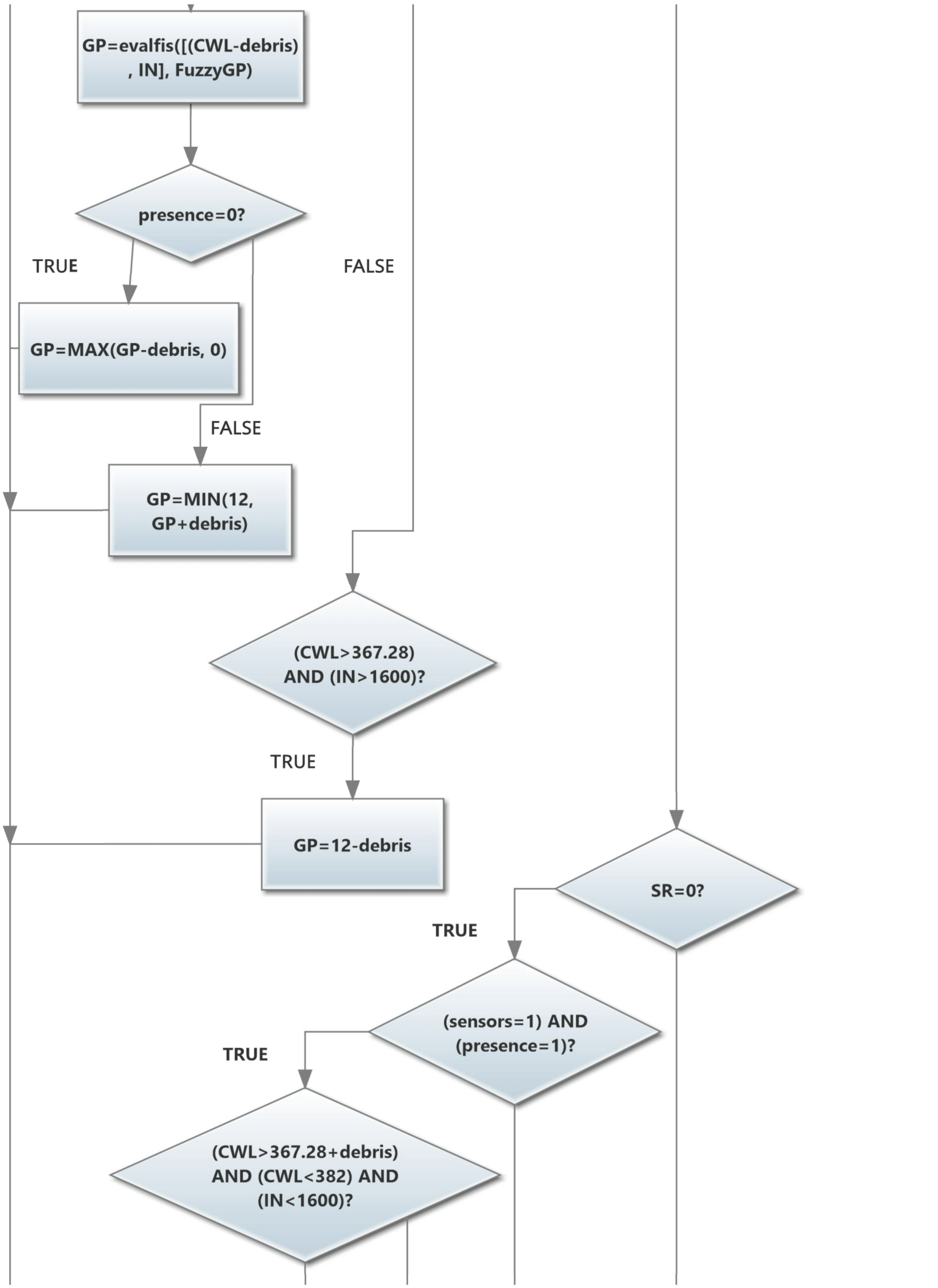
FALSE
c<=t?
TRUE
(MG=0) AND
(DG=0) AND
(BAT=0)?
GP=MAX(GO-debris, 0)
TRUE
FALSE
SR=1?
TRUE
sensors=1?
TRUE
(CWL>367.28+debris)
AND (CWL<382) AND
(IN<1600)?
TRUE
GP=evalfis([(CWL-debris)
, IN], FuzzyGP)
presence=0?
TRUE
FALSE
FALSE
GP=MAX(GP-debris, 0)

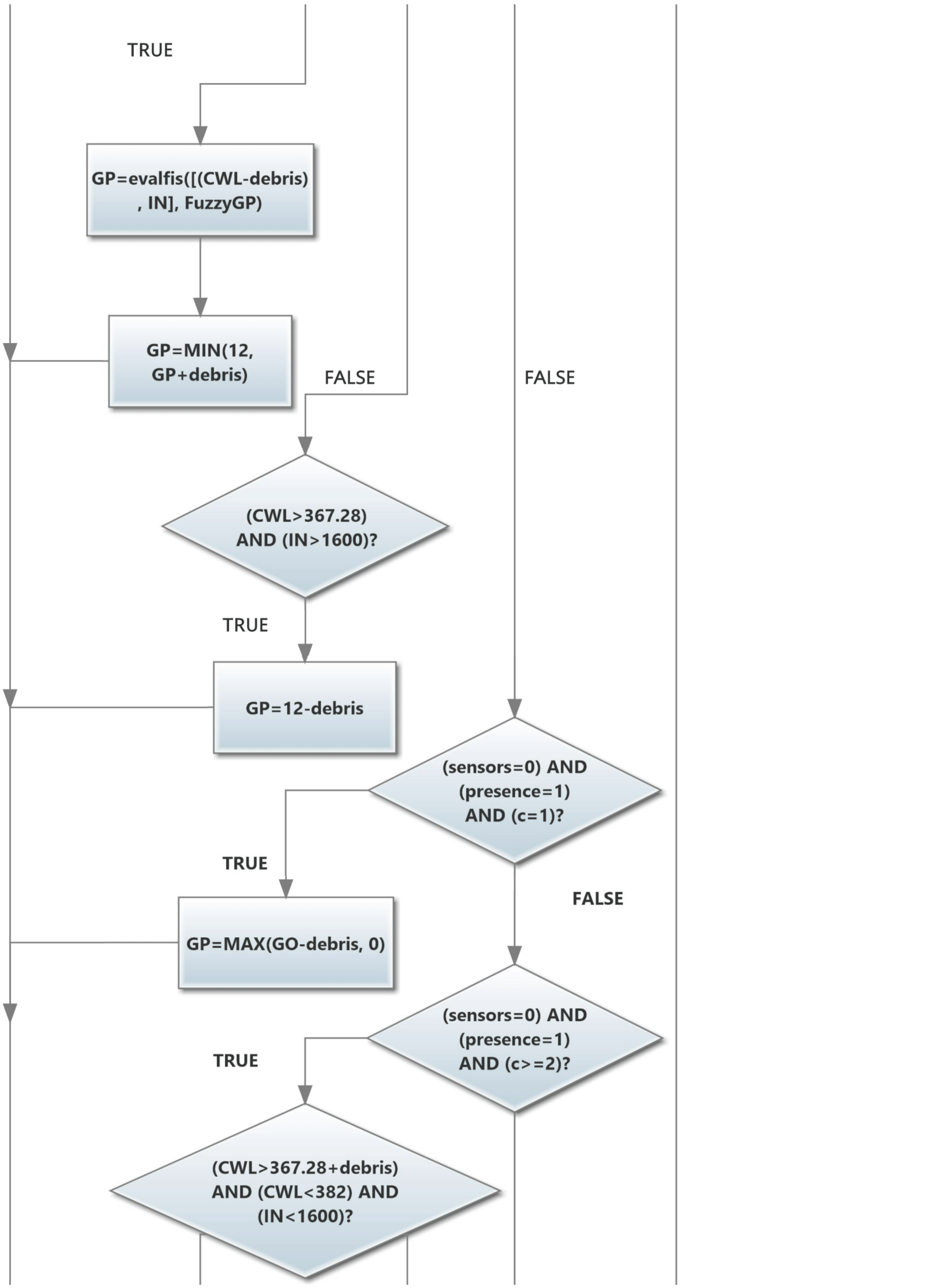

FALSE
GP=MIN(12, GP+debris)
(CWL>367.28) AND (IN>1600)?
TRUE
GP=12-debris
(sensors=0) AND (presence=1) AND (c>=2)?
TRUE
(CWL>367.28+debris) AND (CWL<382) AND (IN<1600)?
TRUE
GP=evalfis([(CWL-debris), IN], FuzzyGP)
FALSE
FALSE
GP=MIN(12, GP+debris)
(CWL>367.28) AND (IN>1600)?

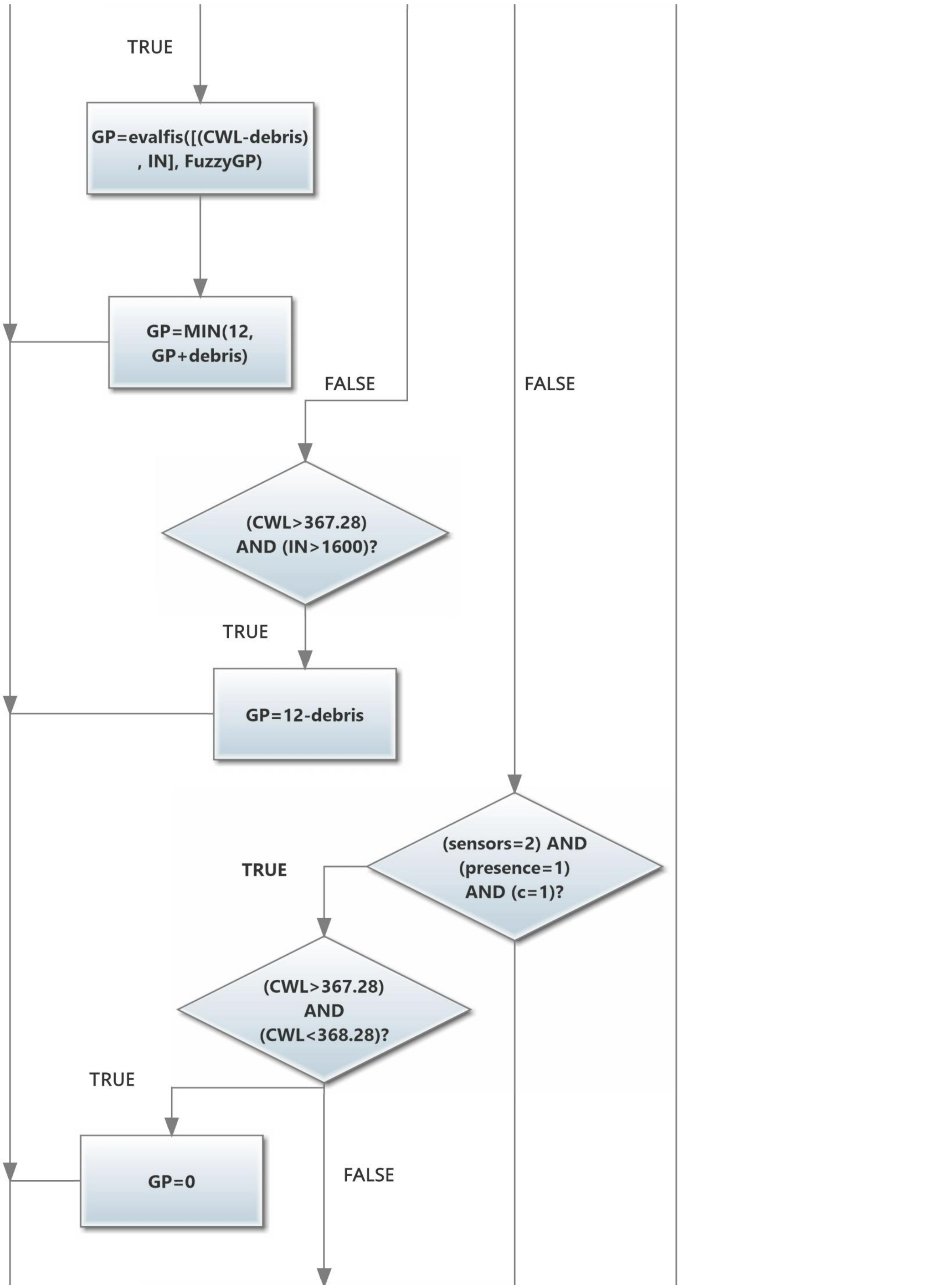

TRUE
GP=12-debris
(sensors=0) AND (presence=1) AND (c=1)?
TRUE
GP=MAX(GO-debris, 0)
FALSE
(sensors=0) AND (presence=0) AND (c<=2)?
TRUE
GP=MAX(GO-debris, 0)
FALSE
(sensors=0) AND (presence=0) AND (c>=3)?
TRUE
(CWL>367.28+debris) AND (CWL<382) AND (IN<1600)?
TRUE
GP=evalfis([(CWL-debris), IN], FuzzyGP)

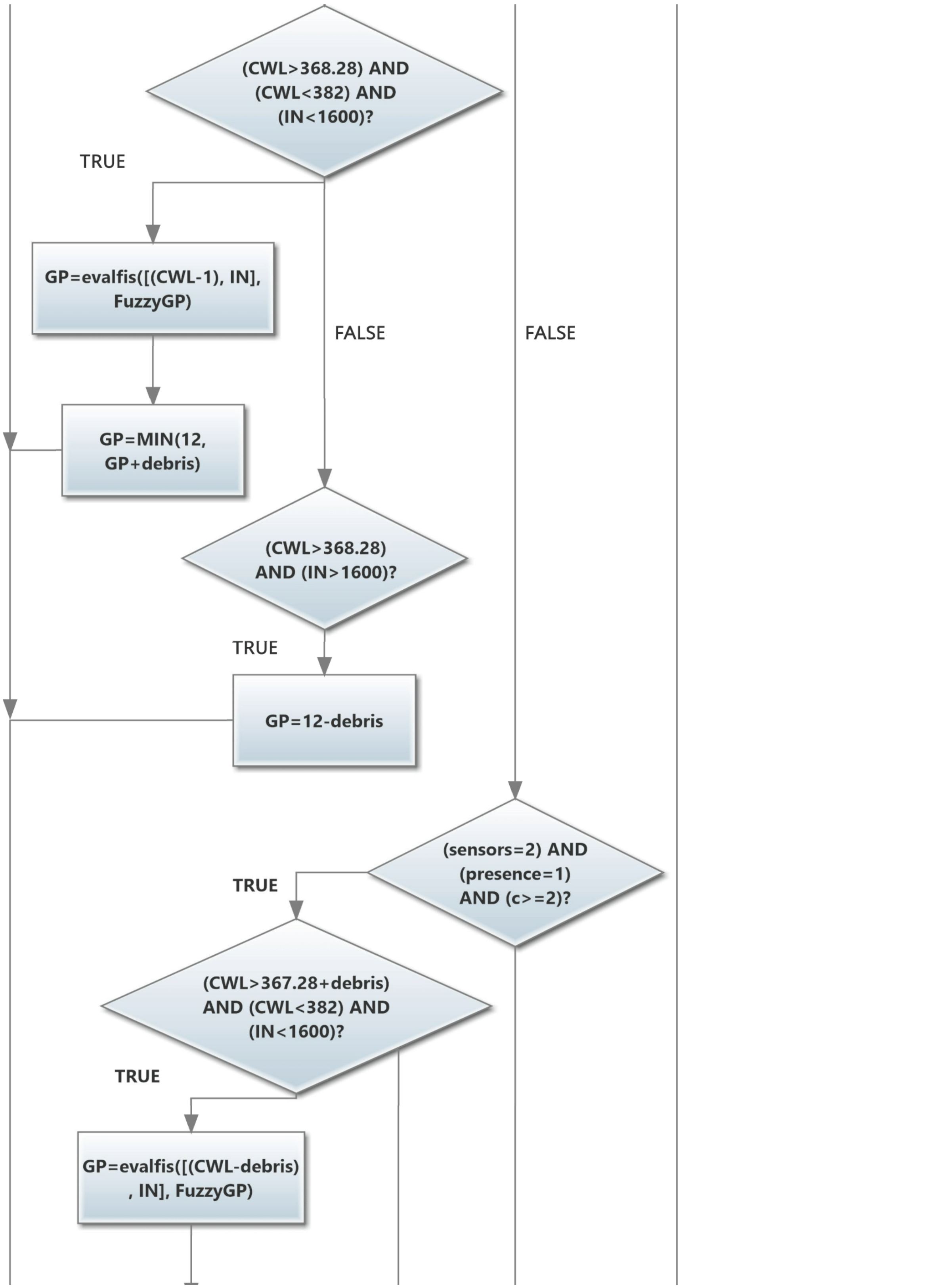

FALSE
FALSE
GP=MIN(12, GP+debris)
(CWL>367.28) AND (IN>1600)?
TRUE
GP=12-debris
(sensors=2) AND (c=1)?
TRUE
(CWL>367.28) AND (CWL<368.28)?
TRUE
GP=0
FALSE
(CWL>368.28) AND (CWL<382) AND (IN<1600)?
TRUE
GP=evalfis([(CWL-1), IN], FuzzyGP)

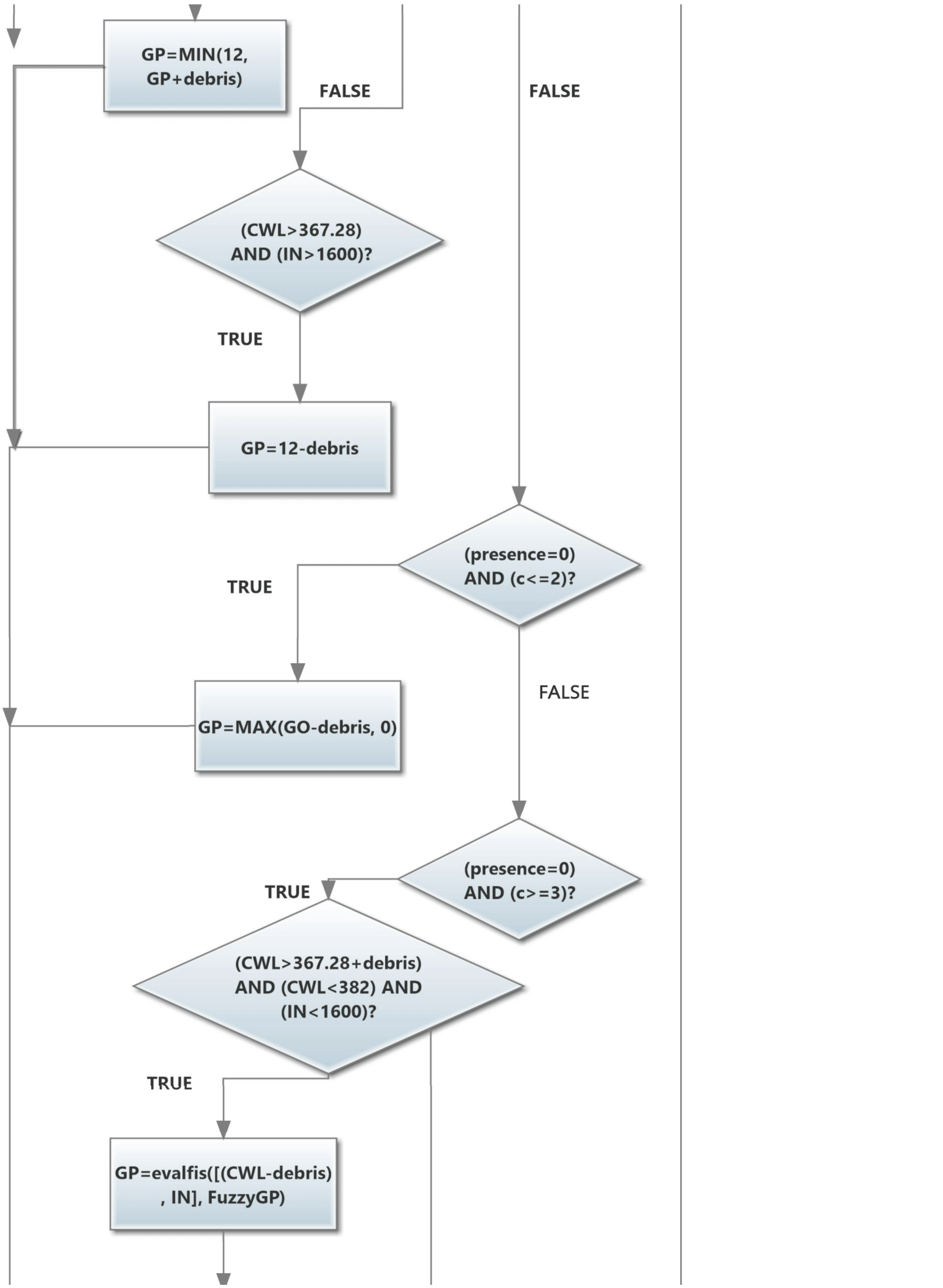

presence=0?
FALSE
FALSE
FALSE
TRUE
GP=MAX(GP-debris, 0)
FALSE
GP=MIN(12, GP+debris)
(CWL>368.28) AND (IN>1600)?
TRUE
GP=12-debris
(sensors=2) AND (c>=2)?
TRUE
(CWL>367.28+debris) AND (CWL<382) AND (IN<1600)?
TRUE

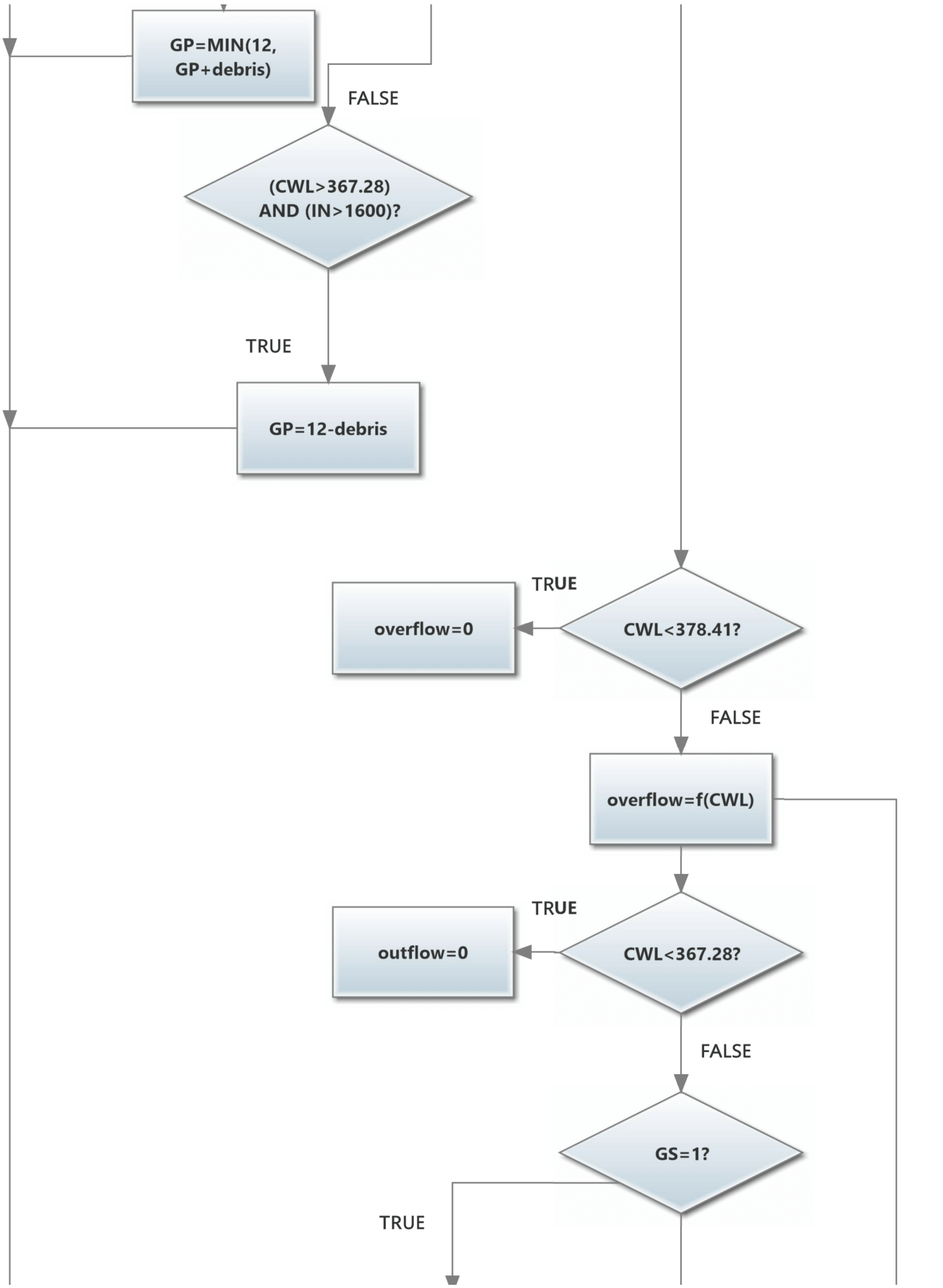

GP=evalfis([(CWL-debris), IN], FuzzyGP)
presence=0?
TRUE
FALSE
GP=MAX(GP-debris, 0)
FALSE
GP=MIN(12, GP+debris)
(CWL>367.28) AND (IN>1600)?
TRUE
GP=12-debris
SR=0?
TRUE
(sensors=1) AND (presence=1)?
TRUE
(CWL>367.28+debris) AND (CWL<382) AND (IN<1600)?

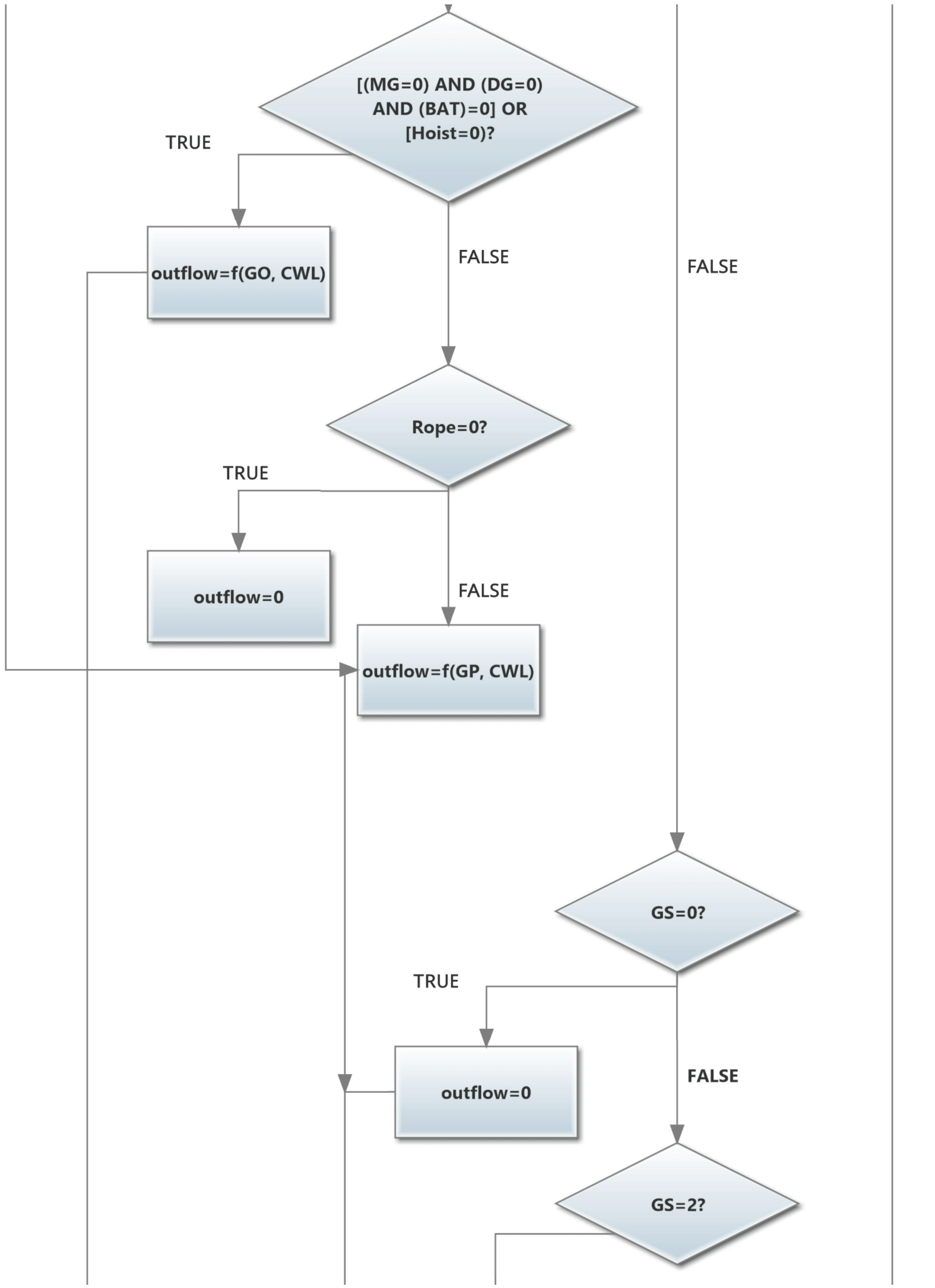

TRUE
GP=evalfis([(CWL-debris)
, IN], FuzzyGP)
GP=MIN(12,
GP+debris)
FALSE
FALSE
(CWL>367.28)
AND (IN>1600)?
TRUE
GP=12-debris
(sensors=0) AND
(presence=1)
AND (c=1)?
TRUE
GP=MAX(GO-debris, 0)
FALSE
(sensors=0) AND
(presence=1)
AND (c>=2)?
TRUE
(CWL>367.28+debris)
AND (CWL<382) AND
(IN<1600)?


TRUE
GP=evalfis([(CWL-debris)
, IN], FuzzyGP)
GP=MIN(12,
GP+debris)
FALSE
FALSE
(CWL>367.28)
AND (IN>1600)?
TRUE
GP=12-debris
(sensors=2) AND
(presence=1)
AND (c=1)?
TRUE
(CWL>367.28)
AND
(CWL<368.28)?
TRUE
GP=0
FALSE

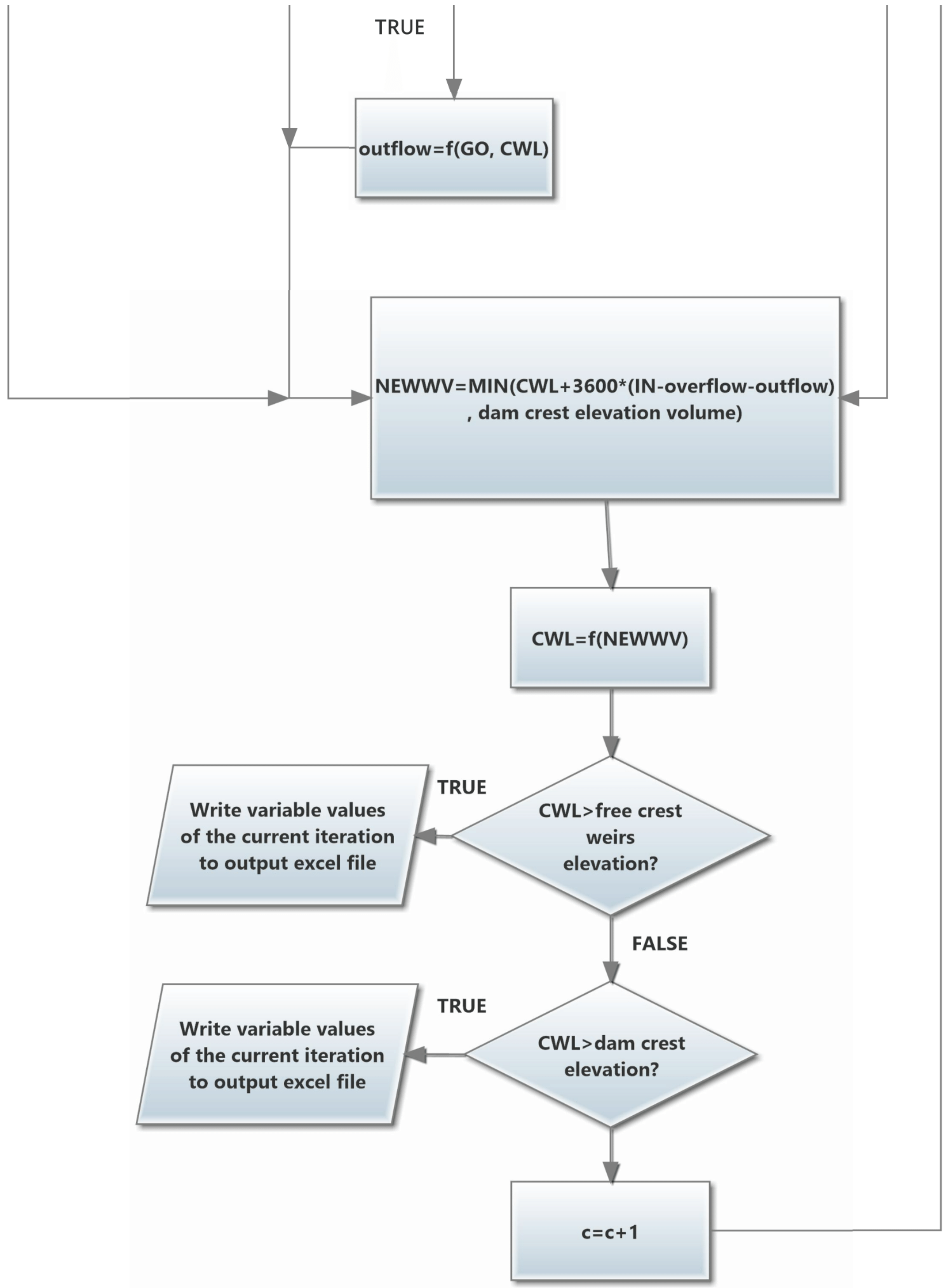

(CWL>368.28) AND
(CWL<382) AND
(IN<1600)?
TRUE
GP=evalfis([(CWL-1), IN],
FuzzyGP)
FALSE
FALSE
GP=MIN(12,
GP+debris)
(CWL>368.28)
AND (IN>1600)?
TRUE
GP=12-debris
(sensors=2) AND
(presence=1)
AND (c>=2)?
TRUE
(CWL>367.28+debris)
AND (CWL<382) AND
(IN<1600)?
TRUE
GP=evalfis([(CWL-debris)
, IN], FuzzyGP)


GP=MIN(12, GP+debris)
FALSE
FALSE
(CWL>367.28) AND (IN>1600)?
TRUE
GP=12-debris
(presence=0) AND (c<=2)?
TRUE
FALSE
GP=MAX(GO-debris, 0)
(presence=0) AND (c>=3)?
TRUE
(CWL>367.28+debris) AND (CWL<382) AND (IN<1600)?
TRUE
GP=evalfis([(CWL-debris) , IN], FuzzyGP)


GP=MIN(12,
GP+debris)
FALSE
(CWL>367.28)
AND (IN>1600)?
TRUE
GP=12-debris
TRUE
overflow=0
CWL<378.41?
FALSE
overflow=f(CWL)
TRUE
outflow=0
CWL<367.28?
FALSE
GS=1?
TRUE



[(MG=0) AND (DG=0)
AND (BAT)=0] OR
[Hoist=0)?
TRUE
outflow=f(GO, CWL)
FALSE
FALSE
Rope=0?
TRUE
outflow=0
FALSE
outflow=f(GP, CWL)
GS=0?
TRUE
outflow=0
FALSE
GS=2?



TRUE
outflow=f(GO, CWL)
NEWWV=MIN(CWL+3600*(IN-overflow-outflow)
, dam crest elevation volume)
CWL=f(NEWWV)
TRUE
Write variable values
of the current iteration
to output excel file
CWL>free crest
weirs
elevation?
FALSE
TRUE
Write variable values
of the current iteration
to output excel file
CWL>dam crest
elevation?
c=c+1

Appendix D: Previous Reports in the Series

ISSN: (Print) 1913-3200; (online) 1913-3219

In addition to 78 previous reports (No. 01 – No. 78) prior to 2012

Samiran Das and Slobodan P. Simonovic (2012). Assessment of Uncertainty in Flood Flows under Climate Change. Water Resources Research Report no. 079, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 67 pages. ISBN: (print) 978-0-7714-2960-6; (online) 978-0-7714-2961-3.

Rubaiya Sarwar, Sarah E. Irwin, Leanna King and Slobodan P. Simonovic (2012). Assessment of Climatic Vulnerability in the Upper Thames River basin: Downscaling with SDSM. Water Resources Research Report no. 080, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 65 pages. ISBN: (print) 978-0-7714-2962-0; (online) 978-0-7714-2963-7.

Sarah E. Irwin, Rubaiya Sarwar, Leanna King and Slobodan P. Simonovic (2012). Assessment of Climatic Vulnerability in the Upper Thames River basin: Downscaling with LARS-WG. Water Resources Research Report no. 081, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 80 pages. ISBN: (print) 978-0-7714-2964-4; (online) 978-0-7714-2965-1.

Samiran Das and Slobodan P. Simonovic (2012). Guidelines for Flood Frequency Estimation under Climate Change. Water Resources Research Report no. 082, Facility for Intelligent Decision

Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 44 pages. ISBN: (print) 978-0-7714-2973-6; (online) 978-0-7714-2974-3.

Angela Peck and Slobodan P. Simonovic (2013). Coastal Cities at Risk (CCaR): Generic System Dynamics Simulation Models for Use with City Resilience Simulator. Water Resources Research Report no. 083, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 55 pages. ISBN: (print) 978-0-7714-3024-4; (online) 978-0-7714-3025-1.

Roshan Srivastav and Slobodan P. Simonovic (2014). Generic Framework for Computation of Spatial Dynamic Resilience. Water Resources Research Report no. 085, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 81 pages. ISBN: (print) 978-0-7714-3067-1; (online) 978-0-7714-3068-8.

Angela Peck and Slobodan P. Simonovic (2014). Coupling System Dynamics with Geographic Information Systems: CCaR Project Report. Water Resources Research Report no. 086, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 60 pages. ISBN: (print) 978-0-7714-3069-5; (online) 978-0-7714-3070-1.

Sarah Irwin, Roshan Srivastav and Slobodan P. Simonovic (2014). Instruction for Watershed Delineation in an ArcGIS Environment for Regionalization Studies.Water Resources Research Report no. 087, Facility for Intelligent Decision Support, Department of Civil and Environmental

Engineering, London, Ontario, Canada, 45 pages. ISBN: (print) 978-0-7714-3071-8; (online) 978-0-7714-3072-5.

Andre Schardong, Roshan K. Srivastav and Slobodan P. Simonovic (2014). Computerized Tool for the Development of Intensity-Duration-Frequency Curves under a Changing Climate: Users Manual v.1 Water Resources Research Report no. 088, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 68 pages. ISBN: (print) 978-0-7714-3085-5; (online) 978-0-7714-3086-2.

Roshan K. Srivastav, Andre Schardong and Slobodan P. Simonovic (2014). Computerized Tool for the Development of Intensity-Duration-Frequency Curves under a Changing Climate: Technical Manual v.1 Water Resources Research Report no. 089, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 62 pages. ISBN: (print) 978-0-7714-3087-9; (online) 978-0-7714-3088-6.

Roshan K. Srivastav and Slobodan P. Simonovic (2014). Simulation of Dynamic Resilience: A Railway Case Study. Water Resources Research Report no. 090, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 91 pages. ISBN: (print) 978-0-7714-3089-3; (online) 978-0-7714-3090-9.

Nick Agam and Slobodan P. Simonovic (2015). Development of Inundation Maps for the Vancouver Coastline Incorporating the Effects of Sea Level Rise and Extreme Events. Water Resources Research Report no. 091, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 107 pages. ISBN: (print) 978-0-7714-3092-3; (online) 978-0-7714-3094-7.

Sarah Irwin, Roshan K. Srivastav and Slobodan P. Simonovic (2015). Instructions for Operating the Proposed Regionalization Tool "Cluster-FCM" Using Fuzzy C-Means Clustering and L-Moment Statistics. Water Resources Research Report no. 092, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 54 pages. ISBN: (print) 978-0-7714-3101-2; (online) 978-0-7714-3102-9.