A Duration Calculus Approach to Specifying a Steam-boiler Control System

Wang JuAn
Software Engineering Programme
University of Macau, P.O.Box 3001, Macau
E-mail: fstjaw@sftw.umac.mo

Li XiaoShan *
International Institute for Software Technology
United Nations University (UNU/IIST)
P.O.Box 3058, Macau
E-mail: Xiaoshan.Li@newcastle.ac.uk

Abstract

This paper presents a general method to cope with hybrid control system specification and verification by making full use of both the simplicity of finite automata and the excessive expressing power of Mean Value Duration Calculus. The method consists of four steps: First, specify the requirements of the whole system through Mean Value Duration Calculus formulas. Second, use two real-time communicating automata to characterize the design decisions, which describe the plant and control subsystems respectively. Third, translate the automata into Mean Value Duration Calculus formulas. Finally prove the design decisions conforming to the requirements specification. Taking the steam-boiler problem raised by J.-R. Abrial[1] as a running example, we demonstrate the above method in three versions of abstractions. Each version represents an abstract level for capturing the original problem requirements and provides a complete procedure of applying the above method. In addition to specifying safety property, emphasis is put on capturing non-functional requirements such as performance, reliability, and optimization. Some formal design techniques about finding an optimal implementation conforming to the system specification are also discussed.

Keywords: Specification, Verification, Mean Value Duration Calculus, Steam-boiler control, Hybrid systems.

1 Introduction

The importance of hybrid control systems has been recognized by scientists both in computer science and control theory communities recently. Hybrid control systems contain both continuous time-evolving subsystems and discrete event-driven subsystems. The challenge is how to design a hybrid control system formally so that we can prove the correctness of the design in the sense that the design conforms to the functional and non-functional requirements specification.

In this paper, we adopt the abstraction that a hybrid control system is divided into two layers. The first layer, which belongs to control engineering, consists of digital control loops with several plants and digital controllers. This layer corresponds to the continuous time evolving subsystem of a hybrid system,

*Present address is: Department of Electrical and Electronic Engineering, University of Newcastle upon Tyne, Newcastle upon Tyne NE1 7RU, UK.

which can be designed with desired stability, robustness and time or frequency domain performance based on control theories. The second layer, which belongs to computing engineering, makes decisions to coordinate and organize those digital control loops to meet the system performance specifications according to the events feeding from the digital control loops. This abstraction is depicted by Fig. 1.

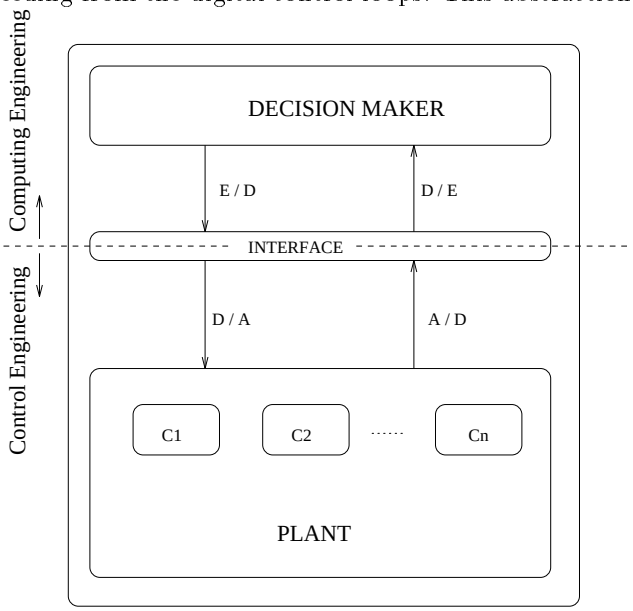

Fig. 1 Abstraction of a hybrid system

A plant consists of some physical devices controlled by controllers. The behaviors of a plant are specified by some natural laws, such as Newton laws or dynamic laws represented by differential equations.

A control loop (or mode) consists of one controller and a subpart of physical devices of the plant. In a complex system, there might exist many control loops. A decision maker is used to coordinate different control loops. At each time period during system operation, the decision maker is responsible to choose one of the control loops according to the received discrete event messages coming from the control loops.

Based on the above abstraction, a *formal design approach* for a hybrid control system consists of the following three stages:

Stage 1 Formalize the specification which captures the system requirements.

Stage 2 Design the event-driven decision maker and digital control loops.

Stage 3 Verify the design decisions in Stage 2 conforming to the requirements specification of Stage 1.

In control engineering, there exist well-developed theories which can be applied to design digital control loops. The other design activities (all except designing control loops) belong to computing engineering accordingly. The main purpose of this paper is to elaborate our approach, called *DC approach*, to specifying and verifying such hybrid control systems. Simply speaking, our approach consists of four steps:

Step 1 Requirements capturing;

Step 2 Expressing design decisions by means of communicating automata;

Step 3 Formalizing the design decisions in Duration Calculus;

Step 4 Proving the correctness of the design decisions.

After requirements capturing, the whole system is decomposed into time domain subsystem and event domain subsystem so that design decisions can be characterized by two finite automata. Based upon the first three steps, we must formally prove in the last step that the decision maker together with the digital control loops (represented by a plant automaton here) implement correctly the system requirements specification. This requires that some formal tools be available for description and validation. There are many formal languages and tools which can be chosen currently, including B, VDM, Z, CSP, Evolving Algebras, CCS, UNITY, Raise, Lacos, Eves, etc. In general, the above-mentioned four step method is independent of specification languages. Different languages provide only different semantics for the real-time communicating automata. Here in this paper we choose Mean Value Duration Calculus [4] (also called Mean Value Calculus, belonging to the family of Duration Calculi [5]) as such a formal tool because it is expressive enough for the design and verification of hybrid systems and flexible enough as well.

On the other hand, the rationales of choosing finite automata to represent design decisions are based on the following consideration: First, formal theories should make good use of structuring tools, especially graphical tools, in order to combine their power of abstraction with the intuition and concreteness of structuring tools. Second, finite automata are simple enough and well accepted by both computing engineering and control engineering. Using a steam-boiler control system as a running example, this paper demonstrates the procedure of designing a hybrid control system and then using Mean Value Duration Calculus to prove the correctness of the design in the sense that the design decisions conform to the system performance specification, assuming that the digital control loops behaves properly under some constraints. In order to make this paper self-contained, we describe the steam-boiler control problem briefly in the following.

The steam-boiler control problem was raised in [1] as a testing case for the existing formal methods. It supposes that in certain application a program is used to control the level of water within some safe range in a steam-boiler. The system comprises the following main units:

- The steam-boiler: A water tank which generates steam.
- The water sensor: A device measures the quantity q (in liters) of water in the steam-boiler.
- The pumps: Four pumps provide the steam-boiler with water. Each pump has its capacity P (in liters/sec). After having been switched on each pump needs five seconds to start pouring water into the boiler. It is stopped, however, with instantaneous effect.
- The pump sensors: Each pump has a sensor which provides the information that whether water flows from the pump to the steam-boiler or not. In other words, the pump sensor can tell at any time whether $p = 0$ or $p = P$ for its supervised pump, here p is the throughput of the pump.
- The steam sensor: A device measures the rate v (in liters/sec) of steam which comes out of the steam-boiler. There is a maximum rate W of steam (in liters/sec) at the exit of the steam-boiler. The maximal gradients of steam increase and decrease are U_1 and U_2 (in litre/sec^2) respectively.

The total capacity of the steam-boiler is C liters. There are two dangerous bounds for the water quantity in the boiler: the minimal limit quantity M_1 of water and maximal limit quantity M_2 (in liters). Below M_1 the steam-boiler would be in danger after five seconds, if the steam continued to come out at its maximum quantity without supply of water from the pumps. Above M_2 the steam-boiler would be in danger after five seconds, if the pumps continued to supply the steam-boiler with water without possibility to evacuate the steam.

There are also two normal water quantity bounds N_1 and N_2 (in liters). The preferable water quantity $q\,(N_1 < q < N_2)$ should be maintained during regular operation.

The program (decision maker) communicates with the physical units (plant and controllers) through messages transmitted via communication channels. The plant provides the program with information from various physical units and sensors while the program controls the plant based on the message received from the plant. The time for message transmission is supposed to be zero and we assume there

is no communication transmission failure during system operation. The main difficulties in designing this system are due to the device failures. When sensors are out of order, the control program should maintain the system operation according to physical laws and the sensors' estimated values. For more details please refer to [1].

This steam-boiler control problem represents a typical real-time and safety-critical hybrid system which is a challenge for specification and verification. The remaining part of this paper is arranged as follows: We first provide a brief introduction to Mean Value Duration Calculus in the next section. Mathematical model for the system is established in section 3 followed by the formalization of estimation. Then in section 5, safety specification of the steam-boiler control system is formulated. Based on this, a design is provided and it is proved to be a correct implementation of the safety specification. Section 6 refines the first version specification into safety, stability and performance. Then the corresponding design and correctness proof are provided. Section 7 gives a further refinement of the specification: safety, stability and performance with device failures, followed by designs and formal verification of the correctness of the design with respect to the specification. In a word, we refine the specification step by step, following the same methodology. Section 8 analyze rescue behavior of the system and design optimization. Some conclusion remarks and further research topics are provided in last section.

2 Mean Value Duration Calculus

The Mean Value Duration Calculus [4] is an extension of the Duration Calculus [3] , by adding a different structure to interval temporal variables, which are mean values of Boolean functions instead of integrals. Boolean functions are piecewise continuous, denoted *F, G,* etc., and 0 and 1 are constant Boolean functions. With the usual Boolean operator $\neg, \wedge, \vee, \cdots$ Boolean functions can be combined to form composite Boolean functions. When a Boolean function is a step function, it represents a *state*, denoted *P, Q, S,* etc., and it represents an *event*, denoted E_1, E_2, etc., when it is a δ-function.

For any Boolean function $F : \mathit{Time} \to \{0,1\}$, the mean value of F, denoted $\bar{F} : \mathit{Interval} \to \mathcal{R}$, is a *term* in the Mean Value Calculus, and is defined, for an arbitrary interval $[a,b]$,

$$\bar{F} \;\widehat{=}\; \begin{cases} \frac{1}{b-a}\int_a^b F & \text{if } b > a \\ F(b) & \text{if } b = a \end{cases}$$

Another *term* in the Mean Value Calculus is the special symbol ℓ, denoting the length of an interval. For an arbitrary interval $[a,b]$, the length of the interval is defined by

$$\ell \;\widehat{=}\; (b-a) \qquad a \leq b$$

The integral of Boolean function F then can be defined as:

$$\int F \;\widehat{=}\; \bar{F} * \ell$$

Terms $t : \mathit{Interval} \to \mathcal{R}$ in mean value calculus have the form:

$$t \;::=\; \bar{F} \mid \ell \mid r \mid t_1 \oplus t_2$$

where r is a real number, and $\oplus \in \{+, -, *\}$.

Formulae $A : \mathit{Interval} \to \{tt, ff\}$ are constructed from such terms with operators in real arithmetics and logical connectives, and have the form:

$$A \;::=\; t_1 = t_2 \mid t_1 \leq t_2 \mid A_1 \wedge A_2 \mid \neg A \mid A_1 ; A_2$$

where ';' is called *chop*. For an arbitrary interval $[a, b]$, $[a, b]$ satisfies $(A_1; A_2)$ iff there exists c such that $a \leq c \leq b$, $[a, c]$ satisfies A_1, and $[c, b]$ satisfies A_2.

The conventional modalities can be defined by *chop* operator as follows:

$$\begin{array}{lcll} \Diamond A & \widehat{=} & true; A; true & \text{(for some subinterval } A \text{ holds)} \\ \Box A & \widehat{=} & \neg\Diamond\neg A & \text{(for all subinterval } A \text{ holds)} \end{array}$$

Before listing axioms in the Mean Value Calculus, we list some definitions and some properties of *chop* operator first:

$$\begin{array}{lcl} \lceil\ \rceil & \widehat{=} & \ell = 0 \\ & & (true \text{ for any point interval}) \\ \lceil F \rceil^0 & \widehat{=} & \lceil\ \rceil \wedge (\overline{F} = 1) \\ & & (F \text{ has value 1 at a point}) \\ & & (\text{We use } F \text{ as an abbreviation of } \lceil F \rceil^0, \text{ when there is no confusion}) \\ \lceil F \rceil & \widehat{=} & (\ell > 0) \wedge \neg((\ell > 0); (\lceil\ \rceil \wedge \overline{F} = 0); (\ell > 0)) \\ & & (F \text{ has value 1 everywhere inside a non-point interval}) \\ \lceil F \rceil^* & \widehat{=} & \lceil F \rceil \vee \lceil F \rceil^0 \end{array}$$

The *chop* operator is associative, monotonic and distributive through disjunction and has the point interval $\lceil\ \rceil$ as unit, $false$ as zero, and length of any interval is nonnegative, and forms a continuum.

- *Associativity:* $(A; B); C \Leftrightarrow A; (B; C)$
- *Monotonicity: If* $A \Rightarrow B$ *and* $C \Rightarrow D$, *then* $(A; C) \Rightarrow (B; D)$
- *Distributivity:* $(A; (B \vee C)) \Leftrightarrow ((A; B) \vee (A; C))$ *and*
  $((A \vee B); C) \Leftrightarrow ((A; C) \vee (B; C))$
- *Unit:* $(A; \lceil\ \rceil) \Leftrightarrow (\lceil\ \rceil; A) \Leftrightarrow A$
- *Zero:* $(A; false) \Leftrightarrow (false; A) \Leftrightarrow false$
- *Nonnegative:* $\ell \geq 0$
- *Continuum:* $(\ell = r + s) \Leftrightarrow (\ell = r); (\ell = s) \qquad r, s \geq 0$

The axioms of the Mean Value Calculus are:

(A1) $\overline{0} = 0$

(A2) $\overline{F} \geq 0$

(A3) $\overline{F \vee G} = \overline{F} + \overline{G} - \overline{F \wedge G}$

(A4) $(\overline{F} * \ell = r + s) \Leftrightarrow (\overline{F} * \ell = r); (\overline{F} * \ell = s) \qquad r, s \geq 0$

(A5) $\lceil\ \rceil \Rightarrow (\lceil F \rceil^0 \vee \lceil \neg F \rceil^0)$

(A6) $\lceil F \rceil^* \Rightarrow (\bar{F} = 1)$

For every given state P there are two corresponding primitive events: the rising edge and the falling edge of P:

- $\uparrow P$ holds at the point where there is a changing from $\neg P$ to P
- $\downarrow P$ holds at the point where there is a changing from P to $\neg P$

$\uparrow P$ and $\downarrow P$ are called *germ-ships*. A transition of a system from state P to state Q can be considered as a composite germ-ship:

$$\downarrow P \wedge \uparrow Q$$

which means, leaving state P and at the same time entering state Q.

States are assumed *finitely variable*. That is, a state can have only finite alternations of its presence and absence in finite interval. DC establishes two induction rules which axiomatize finite variability. Let X denote a formula letter occurring in the formula $R(X)$ and let P be a Boolean state.

Forward Induction Rule:
If $R(\lceil\rceil)$ *holds, and* $R(X \vee (X; \lceil P\rceil)) \wedge R(X \vee (X; \lceil\neg P\rceil))$ *is provable from* $R(X)$*, then* $R(true)$ *holds.*

Backward Induction Rule:
If $R(\lceil\rceil)$ *holds, and* $R(X \vee (\lceil P\rceil; X)) \wedge R(X \vee (\lceil\neg P\rceil; X))$ *is provable from* $R(X)$*, then* $R(true)$ *holds.*

Here we extend MVDC by adding part of Extended Duration Calculus (EDC) for dealing with the functions of real value, where point properties of real-valued functions are taken to be states, and lifted to interval properties by the ceiling operator $\lceil\cdot\rceil$, such as

$$\lceil Continuous(f)\rceil, \qquad \lceil f_1 \leq f_2\rceil, \qquad \lceil \dot{f} = w\rceil, \quad etc.$$

where $\lceil Continuous(f)\rceil$ means that real-valued function f is continuous everywhere *within* a non-point interval, $\lceil f_1 \leq f_2\rceil$ means that function f_1 is less than or equal to function f_2 everywhere inside a non-point interval, etc. In general, given a point property H of function f, $\lceil H(f)\rceil$ holds for interval $[c,d]$ iff $[c,d]$ is not a point interval, and $H(f(t))$ is true for all $t, c < t < d$. That is, $H(f)$ holds everywhere *within* the interval.

EDC can also describe point values of functions by introducing new interval variables $\mathbf{b}.f$ and $\mathbf{e}.f$, where f is an arbitrary given function. For a non-point interval $[c,d]$, the values of $\mathbf{b}.f$ and $\mathbf{e}.f$ in the interval $[c,d]$ are defined as the *right limit* of f at point b and the *left limit* of f at point e respectively. That is,

$$\mathbf{b}.f = f(c^+) = \lim_{t \to c+0} f(t), \qquad \mathbf{e}.f = f(d^-) = \lim_{t \to d-0} f(t).$$

When f is *right continuous* at c, $\mathbf{b}.f$ is equal to the value of f at c. Similarly $\mathbf{e}.f = f(d)$, if f is *left continuous* at d.

Reasoning in the extended DC cannot avoid conventional mathematical theories of functions, such as theory of differential equations. A rule, called MT, is therefore established in the extended DC.

MT :
Let $R(\lceil H(f)\rceil, \mathbf{b}.f, \mathbf{e}.f, l)$ be a formula of the extended DC, which contains no occurrence of the chop (;) operator. If

$$R(\forall t \in (c,d).H(f(t)), f(c^+), f(d^-), d-c)$$

is true in a given mathematical theory of f, then

$$\lceil\rceil \vee R(\lceil H(f)\rceil, \mathbf{b}.f, \mathbf{e}.f, l)$$

is a theorem.

This rule allows us to apply a mathematical theory in reasoning about functions defined in a single interval. Concerning functions defined in consecutive intervals, the calculus establishes rules, such as

Cont :

$$\lceil Continuous(f) \rceil \wedge ((l > 0 \wedge \mathbf{e}.f = r_1); (\mathbf{b}.f = r_2 \wedge l > 0)) \Rightarrow (r_1 = r_2)$$

3 Mathematical Model for System Behavior

The behavior of the steam-boiler system can be characterized by the following functions and state variables:

$$q, v, p_i, WS, SS, PS_i, PO_i, R$$

where q and v are continuous functions; p_i is a piecewise continuous function. Let $Time$ represent the observation time domain, which is a non-negative real number set corresponding to the interval $[0, \infty)$. Boolean value functions WS, SS, PS_i, PO_i and R are used to express the working states of the system.

q : $Time \rightarrow \mathcal{R}$,
$q(t)$ is the water quantity in the steam-boiler at time t.

v : $Time \rightarrow \mathcal{R}$,
$v(t)$ is the rate of steam coming out of the steam-boiler at time t.

p_i : $Time \rightarrow \{0, P\}$,
$p_i(t)$ is the throughput of the i-th $(1 \leq i \leq 4)$ pump at time t.

WS : $Time \rightarrow \{0, 1\}$,
$$WS(t) = \begin{cases} 1, & \text{The water sensor is working properly.} \\ 0, & \text{Otherwise.} \end{cases}$$

SS : $Time \rightarrow \{0, 1\}$,
$$SS(t) = \begin{cases} 1, & \text{The steam sensor is working properly.} \\ 0, & \text{Otherwise.} \end{cases}$$

PS_i : $Time \rightarrow \{0, 1\}$,
$$PS_i(t) = \begin{cases} 1, & \text{The i-th pump sensor is working properly,} \quad (1 \leq i \leq 4). \\ 0, & \text{Otherwise.} \end{cases}$$

PO_i : $Time \rightarrow \{0, 1\}$,
$$PO_i(t) = \begin{cases} 1, & \text{The i-th pump is open at time t} \quad (1 \leq i \leq 4). \\ 0, & \text{Otherwise.} \end{cases}$$

R : $Time \rightarrow \{0, 1\}$
$$R(t) = \begin{cases} 1, & \text{The system is running.} \\ 0, & \text{Otherwise.} \end{cases}$$

The mathematical model of the steam-boiler problem can be established through the following physical laws:

Phy-Law

$\Box\lceil 0 \leq q < C \rceil$

$\Box\lceil 0 \leq p_i \leq P \rceil \qquad (1 \leq i \leq 4)$

$\Box\lceil 0 \leq v \leq W < 4P \rceil$

$\Box\lceil -U_2 \leq \dot{v} \leq U_1 \rceil$

$\Box\lceil \dot{q} = \sum_{i=1}^{4} p_i - v \rceil$

$M_1 < N_1 < N_2 < M_2$

$\Box\lceil (p_i = P) \Rightarrow PO_i \rceil \qquad i \in \{1, 2, 3, 4\}$

$\Box((\uparrow PO_i; \lceil PO_i \rceil \wedge l \geq 5) \Rightarrow ((\lceil p_i = 0 \rceil \wedge l = 5); \lceil p_i = P \rceil; true)) \qquad i \in \{1, 2, 3, 4\}$

$\neg\Diamond(\lceil PO_i \wedge p_i = P \rceil; \lceil PO_i \wedge p_i \neq P \rceil) \qquad i \in \{1, 2, 3, 4\}$

The meaning of these laws are obvious. The first law says that the water quantity q can never exceed the total capacity of the steam-boiler. Similarly the second law prescribes that the maximal throughput of one pump is P. The third law is actually an assumption which says that the maximal throughput of four pumps together is always greater than that of steam. The fourth law says nothing but U_1, U_2 are maximal gradient of increase and decrease, respectively, of steam coming out of the steam-boiler. The fifth law specifies that the changing rate of water quantity is decided by p_i and v. The sixth law assumes that the normal bounds N_1, N_2 and dangerous bound M_1, M_2 are consistent. The last three laws specify the behaviour of pumps. There is water flowing into the steam-boiler only if some pump has been opened. Five seconds delay is necessary between pumps being on and water flowing into the boiler. Any pump will keep on pumping water provided it is opened more than five seconds. It is not the case that a pump is supplying water to the steam-boiler followed by no water flow while the pump is still opened. In other words, pumps themselves will not make physical mistakes.

We assume that q, v, p_i are continuous and all other functions are right continuous:

$$\Box(\lceil \rceil \vee \lceil Continuous(f) \rceil \wedge \lceil Right - Continuous(g) \rceil)$$

$$f \in \{q, v, p_i\},\ g \in \{WS, SS, PS_i, PO_i, R\} \qquad (\mathbf{Assum_1})$$

Since the program follows a cycle which takes place each five seconds to sample data and make decisions, we need a mathematical model for sampling in order to capture the requirements and establish our verification. In each sampling cycle, the program receives messages from physical units, makes decisions based on current status of physical units, and sends out control command to physical units. We use the following functions to characterize sampling behavior:

$\bar{q}$: $Time \rightarrow \mathcal{R}$,
$\bar{q}(t)$ is the sampling value of water quantity in the steam-boiler at time t.

$\bar{v}$: $Time \rightarrow \mathcal{R}$,
$\bar{v}(t)$ is the sampling rate of steam coming out of the steam-boiler at time t.

$\bar{p}_i$: $Time \rightarrow \{0, P\}$,
$\bar{p}_i(t)$ is the sampling throughput of the i-th pump at time t, $i = 1, 2, 3, 4$.

All of the above functions are assumed to be right continuous:

$$\Box(\lceil \rceil \vee \lceil Right - Continuous(g) \rceil \qquad g \in \{\bar{q}, \bar{v}, \bar{p}_i\},\ i = 1, 2, 3, 4. \qquad (\mathbf{Assum_2})$$

We need introducing the following events before discussing the formal model for sampling:

Event	**Meaning**
sp	data is sampled
ppo_i	the i-th pump opens, $i = 1, 2, 3, 4$.
ppf_i	the i-th pump closes, $i = 1, 2, 3, 4$.
$\uparrow R$	the steam-boiler starts
$\downarrow R$	the steam-boiler stops, $\downarrow R \equiv stop$
$\downarrow WS$	the water sensor breaks
$\uparrow WS$	the water sensor is repaired
$\downarrow SS$	the steam sensor breaks
$\uparrow SS$	the steam sensor is repaired
$\downarrow PS_i$	the i-th pump sensor breaks
$\uparrow PS_i$	the i-th pump sensor is repaired

Then the sampling behavior can be formalized by the following sampling laws.

Samp-Law

$$\Box(\lceil R\rceil \wedge (l=5) \Rightarrow \Diamond sp)$$

$$\Box(sp; \lceil \neg sp\rceil; sp \Rightarrow l=5)$$

$$\Box(sp; \lceil WS\rceil \Rightarrow \mathbf{b}.\bar{q} = \mathbf{b}.q)$$

$$\Box(sp; \lceil SS\rceil \Rightarrow \mathbf{b}.\bar{v} = \mathbf{b}.v)$$

$$\Box(sp; \lceil PS_i\rceil \Rightarrow \mathbf{b}.\bar{p_i} = \mathbf{b}.p_i) \qquad i \in \{1,2,3,4\}$$

$$\bigwedge_i (ppo_i \vee ppf_i \vee stop) \Leftrightarrow sp \qquad i \in \{1,2,3,4\}$$

The first two sampling laws specify that there will be a sampling event occurring in five seconds during the system operation and there are exact five seconds between two adjacent samplings. Thus the sampling events occur in every five seconds exactly. The following three laws say that when the physical sensors work properly, they provide correct data at sampling points. The last sampling law prescribes that decision making actions occur at the exact sampling time.

When physical sensors are broken, the control decisions are made based on estimation values of q, v, and p_i, which will be discussed in next section.

4 Formalization of Estimation

It is fairly easy to specify and verify the steam-boiler control system if all physical devices always work perfectly. When physical device failures are taken into consideration, however, correct control decisions must be made at each sampling cycle in order to guarantee the whole system works properly. Obviously correct control decisions depend on appropriate estimation under this situation. It is necessary to formalize the estimation activity firstly in this section.

Similar to notations proposed in [1], we define *calculated* values as the following functions:

$qc_1, qc_2 : Time \rightarrow \mathcal{R}$,

$qc_1(t), qc_2(t)$ are minimal and maximal *calculated* quantity of water.

$vc_1, vc_2 : Time \rightarrow \mathcal{R}$,

$vc_1(t), vc_2(t)$ are minimal and maximal *calculated* rate of exiting steam.

$p_ic_1, p_ic_2 : Time \rightarrow \{0, P\},\ i \in \{1,2,3,4\}$,

$p_ic_1(t), p_ic_2(t)$ are minimal and maximal *calculated* throughput of the i-th pump.

At each sampling cycle, values of $qc_1, qc_2, vc_1, vc_2, p_ic_1, p_ic_2$ are calculated by the decision maker. If the corresponding sensors are working properly, then

$$fc_1 = fc_2 = f,\ \ f \in \{q, v, p_i\}$$

Otherwise, these calculated quantities are determined from the following laws:

Est-Law

$$\Box(sp; \lceil WS\rceil \Rightarrow \lceil \mathbf{b}.qc_1 = \mathbf{b}.qc_2 = \mathbf{b}.q\rceil)$$

$$\Box(sp; \lceil SS\rceil \Rightarrow \lceil \mathbf{b}.vc_1 = \mathbf{b}.vc_2 = \mathbf{b}.v\rceil)$$

$$\Box(sp; \lceil PS_i\rceil \Rightarrow \lceil \mathbf{b}.p_ic_1 = \mathbf{b}.p_ic_2 = \mathbf{b}.p_i\rceil) \qquad i \in \{1,2,3,4\}$$

$$\Box((sp \wedge (\mathbf{b}.qc_1 = q_0); l=5; sp; \lceil \neg WS\rceil) \Rightarrow (l=5; \lceil \mathbf{b}.qc_1 = q_0 - 5(\mathbf{b}.vc_2) - \tfrac{25}{2}U_1 + 5(\mathbf{b}.pc_1)\rceil))$$

$$\Box((sp \wedge (\mathbf{b}.qc_2 = q_0); l = 5; sp; \lceil \neg WS \rceil) \Rightarrow (l = 5; \lceil \mathbf{b}.qc_2 = q_0 - 5(\mathbf{b}.vc_1) - \tfrac{25}{2}U_2 + 5(\mathbf{b}.pc_2) \rceil))$$

$$\Box((sp \wedge (\mathbf{b}.vc_1 = v_0); l = 5; sp; \lceil \neg SS \rceil) \Rightarrow (l = 5; \lceil \mathbf{b}.vc_1 = v_0 - 5U_2 \rceil))$$

$$\Box((sp \wedge (\mathbf{b}.vc_2 = v_0); l = 5; sp; \lceil \neg SS \rceil) \Rightarrow (l = 5; \lceil \mathbf{b}.vc_2 = v_0 + 5U_1 \rceil))$$

$$\Box \lceil pc_1 = \textstyle\sum_{i=1}^{4} p_i c_1 \rceil$$

$$\Box \lceil pc_2 = \textstyle\sum_{i=1}^{4} p_i c_2 \rceil$$

where p_ic_1 and p_ic_2 are the minimal and maximal throughput of each individual pump determined by the following formulas:

$$p_ic_1 = \begin{cases} 0, & \text{if the i-th pump sensor is broken;} \\ p_i, & \text{Otherwise.} \end{cases}$$

$$p_ic_2 = \begin{cases} P, & \text{if the i-th pump sensor is broken;} \\ p_i, & \text{Otherwise.} \end{cases}$$

These calculated values are estimated quantities for decision making when the corresponding physical devices are broken. Here we only consider minimal and maximal estimated values. Actually more delicate estimations are possible. We will mention some of the estimating strategies in Section 8.

The following two assumptions are intended to make our proofs simpler:

$$\uparrow R \Rightarrow \top((M_1{<}q{<}M_2) \wedge WS \wedge SS \wedge PS) \vee \uparrow((M_1{<}q{<}M_2) \wedge WS \wedge SS \wedge PS) \quad \textbf{(Assum}_\mathbf{3}\textbf{)}$$
$$\text{where } PS \equiv PS_1 \wedge PS_2 \wedge PS_3 \wedge PS_4.$$

$$\Box(\lceil\rceil \vee \lceil R \rceil \vee (\lceil R \rceil; sp; \lceil \neg R \rceil) \vee \lceil \neg R \rceil \vee \uparrow R; \lceil R \rceil) \quad \textbf{(Assum}_\mathbf{4}\textbf{)}$$

5 Specification and Verification: The First Version

5.1 Requirements specification Req_1

The mathematical model discussed in previous section provides a foundation for specification and verification in later sections. To avoid symbol abuse, we try to keep the same notations and their meanings as that in [1]. In order to concentrate on the methodology itself we consider a simplified version of the problem at first.

1. There is only one pump in the system.
2. All physical units are always working properly.

In addition to these, we only capture the safety requirement in the following:

Safety The time of the water quantity q exceeding the safe bounds can never last more than 5 seconds.

$$\Box(((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5))$$

5.2 Design decisions

We use two real-time communicating automata to describe the design decisions first, the MVDC (Mean Value Duration Calculus) semantics of these automata is given afterwards.

The first automaton is a decision maker automaton **DM** representing the program controlling the physical units by messages transmission via communication channels. **DM** consists of only two phases[1]: S_0 and S_1. **DM** starts in phase S_0, waiting for the sampling value $\bar{q}$ of water quantity q in the boiler. Based on $\bar{q}$, **DM** sends out controlling commands *ppo* or *ppf* representing *open* or *close* the pump respectively. We suppose the calculation and communication take no time so events *sp* and *ppo*(*ppf*) take place simultaneously. **DM** remains in phase S_0 after transitions $sp \wedge ppo$ or $sp \wedge ppf$. If **DM** realizes that the water quantity risks to reach one of the two limit bounds, however, it will send out *stop* command to shut down the whole system. This decision-making procedure is quite simple. Our idea is, however, to use a simplest case to demonstrate our approach first and to refine it later step by step. The phase transition diagrams of **DM** is illustrated in Fig. 2(a).

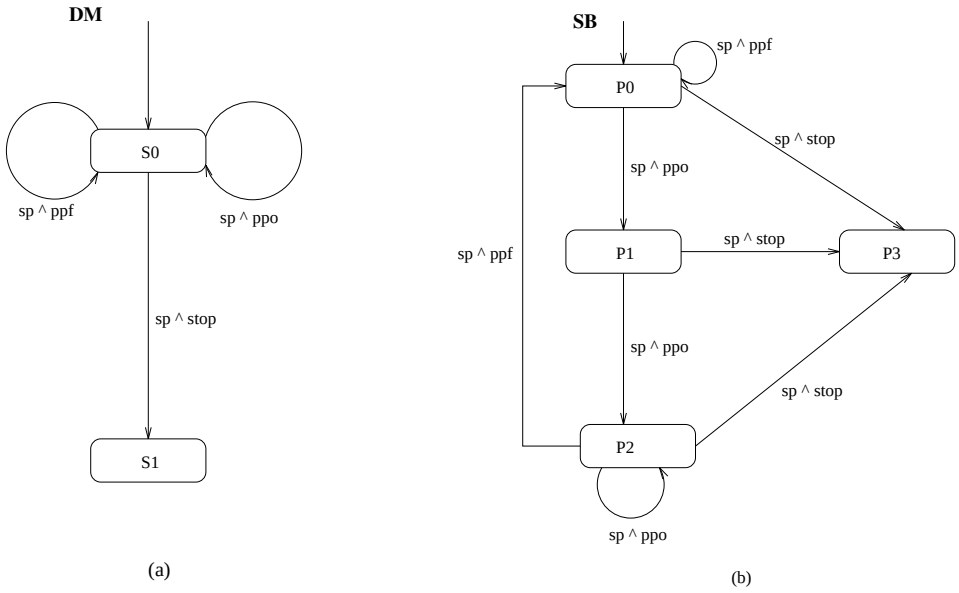

Fig. 2 Communicating automaton models for the first version

Fig. 2(b) illustrates the second automaton **SB** which is an abstraction of the plant part of the whole steam-boiler control system. **SB** has four phases but at any time it is in one and only one of these four phases. Phase transitions are driven by three events:

$sp \wedge ppo$**: SB** provides with sampling value of water quantity while accepting the open-pump command from **DM**.

$sp \wedge ppf$**: SB** provides with sampling value of water quantity while accepting the close-pump command from **DM**.

$sp \wedge stop$**: SB** provides with sampling value of water quantity while accepting the shut-down command from **DM**.

The behavior of **SB** is explained informally as below:

P_0**:** The pump is closed during this phase. If it receives a *ppf* command while supplying sampling data, it will remain in phase P_0. If it receives a *ppo* command, however, it will enter into phase P_1.

P_1**:** In this phase the pump has been opened but there is no water supplying into the boiler yet. After having spent five seconds in this phase, the **SB** goes into phase P_2.

[1]The word *state* is used to denote a time function in MVDC. To avoid ambiguity we use the word *phase* to denote a state of an automaton.

P_2**:** Water is continuously flowing into the boiler at this phase. If it receives a ppf command while supplying sampling data, it will enter into the initial phase P_0.

At each phase of P_0, P_1, and P_2, if a *stop* command is received while supplying sampling data, **SB** will go to phase P_3 immediately.

P_3**:** The whole system is shut-down and no actions at all in this phase.

The two automata **SB** and **DM** cooperate with each other through synchronized communication.

5.3 MVDC Semantics of the Automata

5.3.1 Semantics of DM

1. **Phase completeness and exclusiveness**

   $S_0 \vee S_1 = 1$

   $S_0 \wedge S_1 = 0$

   At any time the automaton **DM** is in one and at most one of its phases. Recall that the definition of state variable R we have: $(S_0 \equiv R) \wedge (S_1 \equiv \neg R)$.

2. **Phase transitions** The transition from S_0 to S_1 is started by *stop* while transitions from S_0 to itself are driven by either ppo or ppf.

   $\uparrow R; l > 0 \Rightarrow \lceil S_0 \rceil; true$

   $\downarrow S_0 \wedge \uparrow S_1 \Leftrightarrow stop \Leftrightarrow \downarrow R$

   $sp; \lceil S_0 \rceil \Leftrightarrow (ppo \vee ppf); \lceil S_0 \rceil$

   $sp; \lceil S_1 \rceil \Leftrightarrow stop; \lceil S_1 \rceil$

3. **Event causality**

   $sp; \lceil \bar{q} \geq M_2 \rceil^* \Rightarrow stop; true$

   $sp; \lceil \bar{q} \leq M_1 \rceil^* \Rightarrow stop; true$

   $sp; \lceil N_2 < \bar{q} < M_2 \rceil^* \Rightarrow ppf; true$

   $sp; \lceil M_1 < \bar{q} < N_1 \rceil^* \Rightarrow ppo; true$

5.3.2 Semantics of SB

1. **Phase completeness and exclusiveness**

   At any time the automaton is in one and at most one of its phases.

   $P_0 \vee P_1 \vee P_2 \vee P_3 = 1$

   $P_i \wedge P_j = 0 \ (i \neq j)$

   According to the definitions of state variables R and PO we can derive the following properties:

   $P_0 \vee P_1 \vee P_2 \equiv R$

   $P_3 \equiv \neg R$

   $P_0 \Rightarrow \neg PO$

   $P_1 \Rightarrow PO$

   $P_2 \Rightarrow PO$

   $\uparrow P_3; l > 0 \equiv \lceil P_3 \rceil$

2. **Phase transitions**

   $(\downarrow P_0 \vee \downarrow P_1 \vee \downarrow P_2) \wedge \uparrow P_3 \Leftrightarrow stop$

   $sp \wedge ppo \Rightarrow (\downarrow P_0 \wedge \uparrow P_1) \vee \top P_2$

   $sp \wedge ppf \Rightarrow (\downarrow P_2 \wedge \uparrow P_0) \vee \top P_0$

3. **Time constraints on transitions**

   $\uparrow P_1; \lceil P_1 \rceil; \downarrow P_1 \Rightarrow l = 5$

5.3.3 Communication of DM and SB

The two automata **DM** and **SB** are combined together by events sp, ppo, ppf, and $stop$.

$sp \Rightarrow (\mathbf{b}.\bar{q} = \mathbf{b}.q) \wedge (\mathbf{b}.\bar{v} = \mathbf{b}.v) \wedge (\mathbf{b}.\bar{p} = \mathbf{b}.p)$

5.4 Verification

Let $Assum$ and Law be defined as below:

$$Assum \hat{=} Assum_1 \wedge Assum_2 \wedge Assum_3 \wedge Assum_4$$

$$Law \hat{=} Phy - Law \wedge Samp - Law \wedge Est - Law$$

The following theorem concludes that the design decisions correctly implement the corresponding specification:

Theorem 1

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Req_1$$

where $Req_1 = \Box(((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5))$.

Proof:

	$\lceil q < M_1 \rceil \wedge \lceil R \rceil \wedge l > 5$	
$\Rightarrow$	$\lceil q \leq M_1 \rceil \wedge \lceil R \rceil \wedge (true; sp; (l > 0))$	Samp-Law
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); (sp \wedge \lceil q \leq M_1 \rceil^0 \wedge \lceil R \rceil^0); (\lceil q \leq M_1 \rceil^* \wedge \lceil R \rceil^*)$	Distributivity
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); (sp \wedge \lceil \bar{q} \leq M_1 \rceil^0 \wedge \lceil R \rceil^0); (\lceil q \leq M_1 \rceil^* \wedge \lceil R \rceil^*)$	Samp-Law
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); sp; (\lceil \bar{q} \leq M_1 \rceil^* \wedge \lceil R \rceil^*); true$	$\bar{q}, R$ right continuous
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); (stop \wedge \lceil R \rceil^0); true$	DM
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); (\downarrow R \wedge \lceil R \rceil^0); true$	DM: $stop \Leftrightarrow \downarrow R$
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); (\lceil \neg R \rceil^0 \wedge \lceil R \rceil^0); true$	R right continuous.
$\Rightarrow$	$(\lceil R \rceil^* \wedge \lceil q \leq M_1 \rceil^*); false; true$	FOL
$\Rightarrow$	$false$	Zero Law

Similarly we can get

$$\lceil q > M_2 \rceil \wedge \lceil R \rceil \wedge l > 5 \Rightarrow false.$$

That is

$$((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5).$$

Thus according to the generalization rule in MVDC we have:

$$\Box(((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5)).$$

□

6 Specification and Verification: The Second Version

6.1 Notations and assumptions

Although the **Safety** specification in the previous section guarantees that there will be no disaster for the whole system, it does not capture exactly the original requirements. Some implicit requirements must also be captured by the specification appropriately. For instance, there is a preferable interval, say $[N_1, N_2]$, for water quantity in the steam-boiler, which should be maintained in normal operation most of the time, in order to generate better working power of steam. In general, a good formal method should have the ability to specify and verify the system performance, reliability, efficiency, fault-tolerance and other non-functional requirements.

In the second version of formalization, we try to refine Req_1 of last section into a more realistic specification Req_2, which captures stability and performance requirements in addition to safety property. To make things not too complicated, we assume there are two pumps, and water sensor may have failures, but the steam sensor and pump sensors work well all the time of system operation. In order to make the paper consistent and readable, we use similar notations as that in previous sections.

For convenience in the following discussion we define synonyms for two events below:

$$q_f = \downarrow WS, \qquad q_r = \uparrow WS.$$

Here the subscript f means *failure* while r means *repaired*.

Because the sampling cycle is 5 seconds, and after having been switched on, the pump needs 5 seconds to start pouring water into the boiler, some delicate consistency relationships among parameters must be taken into consideration in order to guarantee the water quantity q will never below M_1 or above M_2 in more than 5 seconds.

Since we assume in this version there are two pumps supplying water into the steam-boiler, we should change the third law in Phy-Law into the following:

$$(2P > W \geq v \geq 0) \wedge (P < W).$$

That is, the total throughput of two pumps $2P$ should be greater than the maximal steam rate W coming out of the steam-boiler so that desirable water quantity can be maintained through controlling two pumps. However, the throughput P of each pump is less than the maximal steam rate W. Thus careful decisions must be made on how many pumps should be opened at every sampling cycle based on the current water quantity in the boiler.

The following assumption is reasonable and necessary in our proof:

$$(N_2 - N_1 \geq 20W + 5P) \wedge (N_1 - M_1) \geq 10W \qquad (\mathbf{Assum_5})$$

That is, there should have adequate space between N_1 and N_2, and also between N_1 and M_1 so that correct decisions can be made during control procedure and water quantity can be increased in time from below N_1 without risking to fall below dangerous bound M_1 more than 5 seconds.

6.2 Requirements specification Req_2

The specification of this version should be a refinement of Req_1. In other words, in addition to safety, Req_2 should capture more requirements. For instance, the stability and performance of the system should be included.

A stable system is defined in control theory as a system with a bounded system response to bounded input or disturbance. If we take the pump control sequence as system input, steam coming out of the boiler as disturbance, and the water quantity in the boiler as system output, then stability can be thought of as a desirable property of the system. We capture this requirement as *stability* specification.

On the other hand, if the water sensor goes wrong and cannot be repaired any more, the whole system should not be stopped without trying to rescue for at least some time. We capture this requirement as

performance specification. This specification excludes the trivial and unreasonable design, for instance, stop the whole system as soon as the water sensor goes wrong.

Safety The time of the water level q exceeding the safe bounds can't last more than 5 seconds.

$$\Box(((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5))$$

Stability If the water sensor keeps on working perfectly, the whole system should not be stopped and the water quantity should be kept within normal bounds continuously and stably.

$$\exists \delta_1 . \exists \delta_2 . ((\delta_1 < \mathbf{b}.q < \delta_2 \wedge (\uparrow R \vee \top R); \lceil WS \rceil) \Rightarrow \lceil R \wedge (N_1 \leq q \leq N_2) \rceil)$$

Performance Performance includes two requirements:

1. Even when water sensor goes wrong and cannot be repaired, the system can still work for certain period of time before it is shut down.

$$\Box((\mathbf{b}.q = q_0) \wedge (\downarrow WS; \lceil \neg WS \rceil; \uparrow Stop)$$
$$\Rightarrow l \geq min\{\lfloor \tfrac{M_2 - q_0}{2P} \rfloor + \lfloor \tfrac{M_2 - M_1}{W} \rfloor,\ \lfloor \tfrac{q_0 - M_1}{W} \rfloor + \lfloor \tfrac{M_2 - M_1}{2P} \rfloor\})$$

2. The system is shut down only when the water sensor is broken.

$$\downarrow R; l > 0 \Rightarrow \downarrow WS; l > 0$$

Thus the total specification Req_2 is formulated as below:

$$\mathbf{Req_2} = \mathbf{Safety} \wedge \mathbf{Stability} \wedge \mathbf{Performance}.$$

6.3 Design Decisions

We use two real-time communicating automata to describe the design decisions first, MVDC semantics of these automata will be given in the next section.

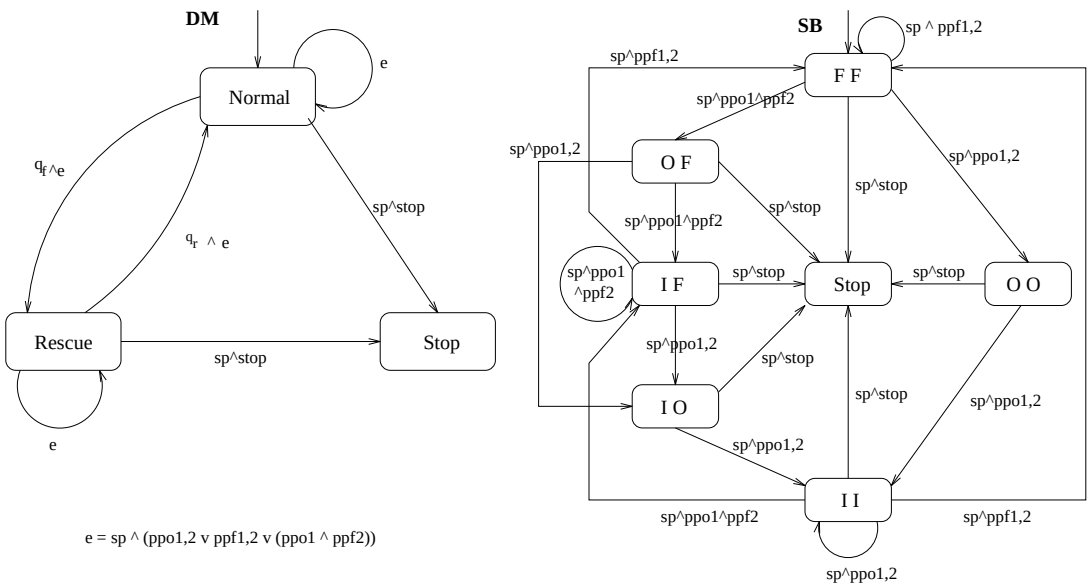

Fig. 3 Finite automaton models for the second version

These two automata communicate synchronously through message passing every five seconds via channels between them. The message is represented as

$$sp = (\bar{q}, \bar{v}, \bar{p}_1, \bar{p}_2).$$

Through these values **DM** can decide whether $WS = 0$ or $WS = 1$ by comparing $\bar{q}$ with calculated values qc_1, qc_2 and make appropriate decisions.

It is not necessary to explain the automaton **DM** as it is self-evident. The behavior of **SB** is explained informally as below:

FF: Both two pumps are closed at this phase. It will remain in this phase after receiving $ppf_{1,2}$. On receiving $ppo_1 \wedge ppf_2$ or $ppo_{1,2}$, it will enter into phase OF or OO respectively.

OF: The first pump is open while the second is closed in this phase. Five second later, it will go to phase IF.

IF: Water is flowing into the boiler through the first pump while the second pump is closed in this phase. It remains in this phase on receiving $ppo_1 \wedge ppf_2$ but goes into IO after event $sp \wedge ppo_{1,2}$.

IO: In this phase, both pumps are open but there is no water flowing through the second pump yet. Five seconds later it will enter into phase II.

II: The **DM** must have realized that the water level was too low during the last sampling cycle so two pumps now are supplying the steam-boiler with water. In this phase the request $ppo_{1,2}$ is ignored because two pumps have already been opened. If event $sp \wedge ppf_{1,2}$ takes place, due to the water level is too high, two pumps will be switched off and **SB** goes into FF phase.

OO: This is the phase when both pumps are open but they have not started pouring water into the boiler yet.

Stop: There is no actions in this phase since **SB** has been shut down by event $sp \wedge stop$.

6.4 MVDC Semantics of the Automata

6.4.1 Semantics of DM

1. **Phase completeness and exclusiveness**

   $Normal \vee Rescue \vee Stop = 1$

   $(Normal \wedge Rescue) \vee (Normal \wedge Stop) \vee (Rescue \wedge Stop) = 0$

   At any time the automaton **DM** is in one and at most one of its phases. Recall that the definition of state variable R and WS we have:

   $((Normal \vee Rescue) \equiv R) \wedge (Stop \equiv \neg R)$

   $(\lceil Normal \rceil \Rightarrow \lceil WS \rceil) \wedge (\lceil Rescue \rceil \Rightarrow \lceil \neg WS \rceil$.

2. **Phase transitions** The transition from $Normal$ to $Rescue$ is started by $q_f \wedge e$ while transitions from $Normal$ to itself is driven by e, where $e = sp \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2))$. The transition from $Normal$ (or $Rescue$) to $Stop$ is evoked by $sp \wedge stop$.

   $\downarrow Normal \wedge \uparrow Stop \Rightarrow sp \wedge stop \Leftrightarrow \downarrow R$

   $\downarrow Normal \wedge \uparrow Rescue \Rightarrow sp \wedge q_f \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2))$

   $\downarrow Rescue \wedge \uparrow Stop \Rightarrow sp \wedge stop \Leftrightarrow \downarrow R$

$\downarrow Rescue \wedge \uparrow Normal \Rightarrow sp \wedge q_r \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2))$

$sp; \lceil Normal \rceil \Leftrightarrow (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2)); \lceil Normal \rceil$

$sp; \lceil Rescue \rceil \Leftrightarrow (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2)); \lceil Rescue \rceil$

3. **Event causality**

$sp \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2)) \Rightarrow (\downarrow Normal \wedge \uparrow Normal) \vee (\downarrow Rescue \wedge \uparrow Rescue) \vee (\downarrow Normal \wedge \uparrow Rescue) \vee (\downarrow Rescue \wedge \uparrow Normal)$

$sp \wedge stop \Rightarrow (\downarrow Normal \wedge \uparrow Stop) \vee (\downarrow Rescue \wedge \uparrow Stop)$

$q_f \wedge sp \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2)) \Rightarrow (\downarrow Normal \wedge \uparrow Rescue)$

$q_r \wedge sp \wedge (ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2)) \Rightarrow (\downarrow Rescue \wedge \uparrow Normal)$

$sp; \lceil WS \wedge (q \leq M_1) \rceil^* \Rightarrow stop; true$

$sp; \lceil WS \wedge (q > M_2) \rceil^* \Rightarrow stop; true$

$sp; \lceil WS \wedge (M_1 < q \leq N_1 + 10W) \rceil^* \Rightarrow ppo_{1,2}; true$

$sp; \lceil WS \wedge (N_1 + 10W < q \leq N_2 - 5P) \rceil^* \Rightarrow (ppo_1 \wedge ppf_2); true$

$sp; \lceil WS \wedge (N_2 - 5W < q \leq M_2) \rceil^* \Rightarrow ppf_{1,2}; true$

$sp; \lceil \neg WS \wedge (qc_1 \leq M_1) \rceil^* \Rightarrow stop; true$

$sp; \lceil \neg WS \wedge (qc_2 \geq M_2) \rceil^* \Rightarrow stop; true$

$sp; \lceil \neg WS \wedge (M_1 < qc_1) \wedge (qc_2 < N_1) \rceil^* \Rightarrow ppo_{1,2}; true$

$sp; \lceil \neg WS \wedge (M_1 < qc_1 \leq N_1) \wedge (N_1 < qc_2 < N_2) \rceil^* \Rightarrow ppo_{1,2}; true$

$sp; \lceil \neg WS \wedge (M_1 < qc_1 < N_1) \wedge (N_2 < qc_2 < M_2) \rceil^* \Rightarrow (ppo_1 \wedge ppf_2); true$

$sp; \lceil \neg WS \wedge (N_1 < qc_1 < N_2) \wedge (N_1 < qc_2 < N_2) \rceil^* \Rightarrow (ppo_1 \wedge ppf_2); true$

$sp; \lceil \neg WS \wedge (N_1 < qc_1 < N_2) \wedge (N_2 < qc_2 < M_2) \rceil^* \Rightarrow ppf_{1,2}; true$

$sp; \lceil \neg WS \wedge (N_2 < qc_1 < M_2) \wedge (N_2 < qc_2 < M_2) \rceil^* \Rightarrow ppf_{1,2}; true$

6.4.2 Semantics of SB

1. **Phase completeness and exclusiveness**

$FF \vee OF \vee IF \vee IO \vee II \vee OO \vee Stop = 1$

$\bigwedge_{Phase \neq Phase'} (Phase \Rightarrow \neg Phase'), \quad Phase, Phase' \in \{FF, OF, IF, IO, II, OO, Stop\}$

2. **Phase transitions**

$\downarrow FF \wedge \uparrow OF \Rightarrow (sp \wedge ppo_1 \wedge ppf_2)$

$\downarrow FF \wedge \uparrow OO \Rightarrow (sp \wedge ppo_{1,2})$

$\downarrow FF \wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$sp; \lceil FF \rceil \Leftrightarrow ppf_{1,2}; \lceil FF \rceil$

$\downarrow OF \wedge \uparrow IF \Rightarrow (sp \wedge ppo_1 \wedge ppf_2)$

$\downarrow OF \wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$\downarrow OF \wedge \uparrow IO \Rightarrow (sp \wedge ppo_{1,2})$

$\downarrow IF \wedge \uparrow FF \Rightarrow (sp \wedge ppf_{1,2})$

$\downarrow IF \wedge \uparrow IO \Rightarrow (sp \wedge ppo_{1,2})$

$\downarrow IF \wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$sp; \lceil IF \rceil \Leftrightarrow (ppo_1 \wedge ppf_2); \lceil IF \rceil$

$\downarrow IO\wedge \uparrow II \Rightarrow (sp \wedge ppo_{1,2})$

$\downarrow IO\wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$\downarrow II\wedge \uparrow FF \Rightarrow (sp \wedge ppf_{1,2})$

$\downarrow II\wedge \uparrow IF \Rightarrow (sp \wedge ppo_1 \wedge ppf_2)$

$\downarrow II\wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$sp; \lceil II \rceil \Leftrightarrow ppo_{1,2}; \lceil II \rceil$

$\downarrow OO\wedge \uparrow II \Rightarrow (sp \wedge ppo_{1,2})$

$\downarrow OO\wedge \uparrow Stop \Rightarrow (sp \wedge stop)$

$\uparrow Stop \Leftrightarrow sp \wedge stop \Leftrightarrow \downarrow R$

3. **Event causality**

   $(sp \wedge ppo_{1,2}) \Rightarrow (\downarrow FF\wedge \uparrow OO) \vee (\downarrow OO\wedge \uparrow II) \vee (\downarrow II\wedge \uparrow II) \vee (\downarrow IO\wedge \uparrow II) \vee (\downarrow IF\wedge \uparrow IO) \vee (\downarrow OF\wedge \uparrow IO)$

   $(sp \wedge ppf_{1,2}) \Rightarrow (\downarrow FF\wedge \uparrow FF) \vee (\downarrow II\wedge \uparrow FF) \vee (\downarrow IF\wedge \uparrow FF)$

   $sp \wedge stop \Rightarrow (\downarrow FF\wedge \uparrow Stop) \vee (\downarrow OF\wedge \uparrow Stop) \vee (\downarrow IF\wedge \uparrow Stop) \vee (\downarrow IO\wedge \uparrow Stop) \vee (\downarrow II\wedge \uparrow Stop) \vee (\downarrow OO\wedge \uparrow Stop)$

   $(sp \wedge ppo_1 \wedge ppf_2) \Rightarrow (\downarrow FF\wedge \uparrow OF) \vee (\downarrow OF\wedge \uparrow IF) \vee (\downarrow IF\wedge \uparrow IF)$

6.4.3 Communication of DM and SB

The two automata **DM** and **SB** are combined together by events sp, $ppo_{1,2}$, $ppf_{1,2}$, $ppo_1 \wedge ppf_2$, and $stop$.

$$sp \Leftrightarrow ppo_{1,2} \vee ppf_{1,2} \vee (ppo_1 \wedge ppf_2) \vee stop$$

6.5 Verification

Let $Assum$ and Law be defined as below:

$$Assum \mathrel{\widehat{=}} \bigwedge_{i=1}^{5} Assum_i$$

$$Law \mathrel{\widehat{=}} Phy-Law \wedge Samp-Law \wedge Est-Law$$

where the third law in Phy-Law is replaced properly. Thus we have following theorems concluding that the design decisions correctly implement the corresponding specification.

Theorem 2

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Safety$$

Proof:
Since $\lceil R \rceil \Leftrightarrow \lceil Normal \rceil \vee \lceil Rescue \rceil$ according to the semantics of DM, the proof of this theorem can be transformed into the following two lemmas:

Lemma 1 $\lceil Normal \rceil \wedge \lceil q < M_1 \rceil \wedge (l > 5) \Rightarrow false$.
Proof of Lemma 1:

$\lceil Normal \rceil \wedge \lceil q \leq M_1 \rceil \wedge (l > 5)$
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil \wedge (l > 5)$ Theorem of DC

$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil \wedge (true; sp; true)$ Samp-Law
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; (sp \wedge \lceil Normal \wedge q \leq M_1 \rceil^0; \lceil Normal \wedge q \leq M_1 \rceil$ Distributivity
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; (sp \wedge \lceil WS \rceil^0 \wedge \lceil q \leq M_1 \rceil^0; \lceil WS \wedge q \leq M_1 \rceil$ DM
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; (sp \wedge \lceil WS \rceil^0 \wedge \lceil q \leq M_1 \rceil^0; \lceil WS \wedge q \leq M_1 \rceil; \lceil WS \wedge q \leq M_1 \rceil$ DC
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; sp; \lceil WS \wedge q \leq M_1 \rceil; \lceil WS \wedge q \leq M_1 \rceil$ WS, q right continuous
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; stop; \lceil Normal \wedge q \leq M_1 \rceil$ DM
$\Rightarrow \lceil Normal \wedge q \leq M_1 \rceil; \lceil \neg R \rceil^0; \lceil R \rceil$ DM
$\Rightarrow false$ $Assum_4$, DC

Lemma 2 $\lceil Rescue \rceil \wedge \lceil q < M_1 \rceil \wedge (l > 5) \Rightarrow false.$
Proof of Lemma 2: Similar to the proof of Lemma 1.
Combining these two lemmas, Theorem 2 is proved.

□

Theorem 3

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Stability.$$

Proof: First prove two lemmas:
Lemma 1 $l \leq 5 \Rightarrow Stability.$
Lemma 2 $l > 5 \Rightarrow Stability.$
Then apply the *Forward Induction Rule*. Please refer to Appendix A for details.

□

Theorem 4

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Performance.$$

Proof: Please refer to Appendix B for details.

□

From Theorem 2 to Theorem 4 we conclude the following

Theorem 5 **(Correctness of design)**

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Req_2.$$

□

7 Specification and Verification: The Third Version

7.1 Requirements specification Req_3

The specification of this version is a refinement of Req_2 in the sense that we take into consideration of steam sensor failure and pump sensor failure as well as the water sensor failure.

Safety The time of the water level q exceeding the safe bounds can't last more than 5 seconds.

$$\Box(((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5))$$

Stability If all physical devices keep on working perfectly, the whole system should not be stopped and the water quantity should be kept within normal bounds continuously and stably.

$$\exists \delta_1 . \exists \delta_2 . (((\delta_1 < \mathbf{b}.q < \delta_2) \wedge (\uparrow R \vee \top R); \lceil WS \wedge SS \wedge PS_1 \wedge PS_2 \rceil)$$
$$\Rightarrow \lceil R \wedge (N_1 \leq q \leq N_2) \rceil)$$

Performance Performance contains two requirements:

1. Even when water sensor goes wrong and cannot be repaired, the system should still work for certain period of time provided that the steam sensor and pump sensor keep on working well.

$$\Box((\mathbf{b}.q = q_0) \wedge \downarrow WS; \lceil \neg WS \wedge SS \wedge PS_1 \wedge PS_2 \rceil; \uparrow Stop)$$
$$\Rightarrow l \geq min\{\lfloor \tfrac{M_2 - q_0}{2P} \rfloor + \lfloor \tfrac{M_2 - M_1}{W} \rfloor,\ \lfloor \tfrac{q_0 - M_1}{W} \rfloor + \lfloor \tfrac{M_2 - M_1}{2P} \rfloor\})$$

2. The system is shut down only when some physical units were broken and can not be repaired.

$$\downarrow R; l > 0 \Rightarrow (\downarrow WS \vee \downarrow SS \vee \downarrow PS_1 \vee \downarrow PS_2); l > 0$$

Thus the total specification Req_3 is formulated as below:

$$\mathbf{Req_3} = \mathbf{Safety} \wedge \mathbf{Stability} \wedge \mathbf{Performance}.$$

7.2 Design Decisions

The design decisions can be illustrated by two automata **SB** and **DM** similar to that in Fig. 3. As two pumps are supposed in the system, the automaton **SB** is exactly the same as that in Fig. 3. However, since we take all physical device failures into consideration, the automaton **DM** is slightly different from that in Fig. 3. A *Degraded* phase is added and more transitions are drawn accordingly.

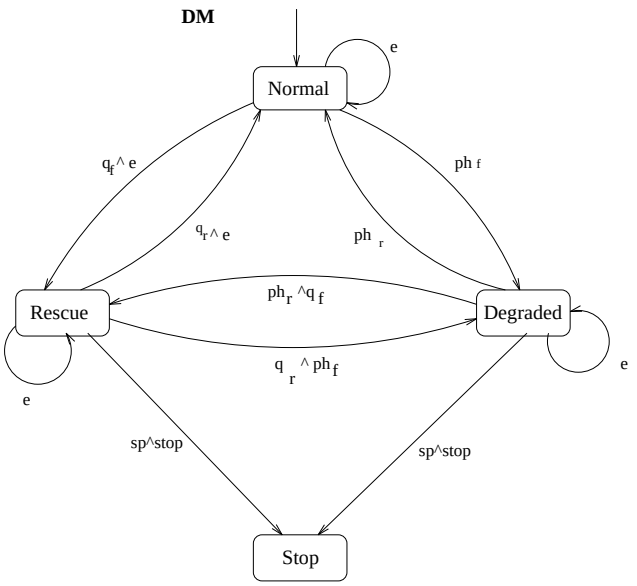

Fig. 4 Finite automaton models for Req_3

In Fig. 4, q_f and q_r represent water sensor failure and repaired respectively; ph_f and ph_r represent steam sensor or pump sensor failure and repaired respectively.

$$ph_f = \downarrow SS \vee \downarrow PS_1 \vee \downarrow PS_2$$
$$ph_r = (\uparrow SS \wedge \top PS_1 \wedge \top PS_2) \vee (\top SS \wedge \top PS_2 \wedge \uparrow PS_1) \vee (\top SS \wedge \top PS_1 \wedge \uparrow PS_2)$$

Here we omitted the semantics of **SB** and **DM** which can be defined in the same way as in previous two versions.

7.3 Verification

Applying the similar methodology used in the previous two versions we can prove the following theorem:

Theorem 6 **(Correctness of design)**

$$DM \wedge SB \wedge Assum \wedge Law \Rightarrow Req_3.$$

□

8 Design Optimization

For the steam-boiler control problem, a design is a set of sequences of pump control events characterized by the decision maker automaton. There might exist many designs which satisfy the basic requirements of the system. How to compare two different designs? Which one is better? Naturally, we hope to find an optimal design for a particular set of requirements.

For the steam-boiler control, the system is expected to be robust and fault-tolerant. That is to say, the system should be able to rescue as long as possible while the water sensor is broken. Thus an optimal design should be the one having maximal capability to prolong the rescue time. We formalize these idea in the following definitions.

Definition 8.1 *Rescue time* l_D for a design D is defined as the time period from the starting of rescue to the termination of the system due to the water sensor cannot be repaired.

Definition 8.2 Two designs can be *compared* through the following binary relation $\mathcal{B}$, which means "*better than*":

$$\mathcal{B} \mathrel{\widehat{=}} \{(D_1, D_2) \mid (\downarrow WS; \lceil \neg WS \wedge R \rceil; \downarrow R \Rightarrow l = l_{D_1}) \wedge$$
$$(\downarrow WS; \lceil \neg WS \wedge R \rceil; \downarrow R \Rightarrow l = l_{D_2}) \wedge (l_{D_1} \geq l_{D_2})\}$$

Obviously, $\mathcal{B}$ is a partial order relation.

Definition 8.3 An *optimal design* is the greatest element in design sets with respect to the binary relation $\mathcal{B}$.

Now the problem is that whether we can always find such an optimal design. In the following discussion we will see that it is possible if the rescue strategy is determined.

8.1 Analysis of Rescue Phase

The rescue phase is a time period in which the decision maker tries to maintain a satisfactory water level in spite of the failure of the water sensor. The *rescue strategy* is: The water quantity is estimated by a calculation which is done taking into account the maximum dynamics of the quantity of steam coming out of the steam-boiler. To simplify the discussion, let us consider the case that there is only one pump in the system.

Suppose we have a sequence of sampled values:

$$(\bar{q}_0, \bar{v}_0, \bar{p}_0),\ (\bar{q}_1, \bar{v}_1, \bar{p}_1),\ \cdots, (\bar{q}_i, \bar{v}_i, \bar{p}_i),\ (\bar{q}_{i+1}, \bar{v}_{i+1}, \bar{p}_{i+1}), \cdots$$

If at the i-th sampling time all physical units work perfectly but the water sensor is found broken at the $(i+1)$-th sampling time, we need to estimate the water quantity q_{i+1} by using q_i, v_i, v_{i+1} and p_i so that the control decision p_{i+1} can be made. However, estimating errors might exist at each calculation since we do not know exactly the intermediate values of steam emitted between two adjacent samplings.

The key point of rescue is to analyze these possible estimating errors and try to find maximal rescue time for a given sequence of sampling values.

According to Phy-Law, q_{i+1} depends on p_i and $v(t), t_i \leq t \leq t_{i+1}, (t_{i+1} = t_i + 5)$. We depict the possible change of steam as the following diagram:

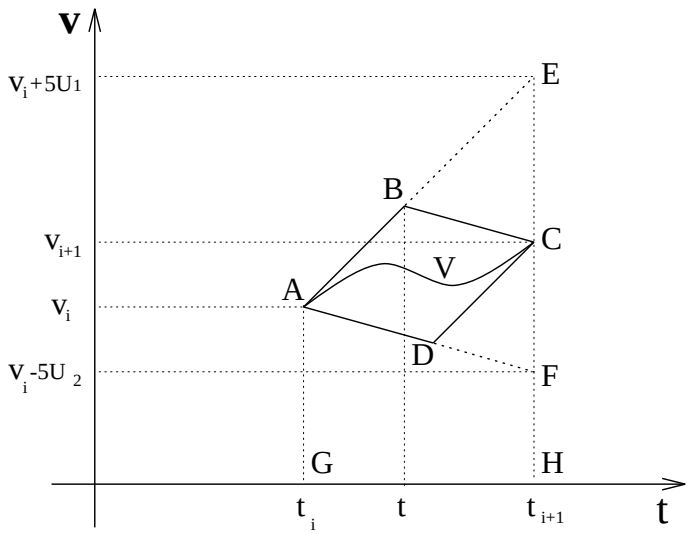

Fig. 5 Possible changes of steam

Obviously from Phy-Law and Fig. 5 we have the following inequation:

$$(v_i - 5U_2) \leq v_{i+1} \leq (v_i + 5U_1).$$

We can not decide exactly where lines $v = 0$ and $v = W$ should be in Fig 5 explicitly. However, all possible changes of steam rate between two successive sampling points t_i and t_{i+1} will certainly fall into the area of triangle AEF. There are two extreme cases in which steam quantity continuously increases from v_i to $v_i + 5U_1$ or decreases from v_i to $v_i - 5U_2$ on the time interval $[t_i, t_{i+1}]$, i. e., $v_{i+1} = v_i + 5U_1$ or $v_{i+1} = v_i - 5U_2$. If $v_i - 5U_2 < v_{i+1} < v_i + 5U_1$ holds, we have two degraded extreme cases: (1). v_i first increases continuously at maximal velocity U_1 from time t_i to t, some intermediate time point within interval $[t_i, t_{i+1}]$, and then decreases continuously at maximal velocity U_2 from time t to t_{i+1}. (2). Oppositely v_i first decreases continuously at maximal velocity U_2 and then increases continuously at maximal velocity U_1. All those degraded extreme changes of steam quantity fall into the area of quadrilateral $ABCD$. A typical curve of steam dynamically change between two successive sampling points is depicted as the curve AVC in Fig. 5.

It is not difficult to prove in the degraded extreme cases (where $v_i - 5U_2 < v_{i+1} < v_i + 5U_1$) that the broken line ABC represents the case in which greatest quantity of steam has been emitted out of the steam-boiler between two successive samplings. Similarly, the broken line ADC represents the situation when there is least quantity of steam coming out of the steam-boiler between two successive samplings.

By Phy-Law, the following equation holds immediately:

$$q_{i+1} = q_i + 5p_i - \int_{t_i}^{t_{i+1}} v.$$

Because only two instant values of v are known as v_i and v_{i+1}, the total quantity of emitted steam from t_i to t_{i+1}, i.e. $\int_{t_i}^{t_{i+1}} v$, should satisfy:

$$S_{ADCHG} \leq \int_{t_i}^{t_{i+1}} v \leq S_{ABCHG}$$

where S_p stands for the area of polygon p. So S_{ADCHG} and S_{ABCHG} are the least and greatest quantities of steam, respectively, coming out of the steam-boiler from time t_i to t_{i+1}. Thus their difference $S_{ABCHG} - S_{ADCHG} = S_{ABCD}$ is the greatest estimating error possibly made at the i-th sampling. If we estimate n times continuously starting from the beginning of a rescue phase, we get an accumulated estimating error $S^1_{ABCD} + \cdots + S^n_{ABCD}$. With the number n increasing, there is a least upper bound m which makes such accumulated estimating error greater than $(M_2 - M_1)$. That is, After rescuing m times of sampling cycle, we can not guarantee the quantity of water to stay between M_1 and M_2. In other words, the water quantity q may exceed the safety bound $[M_1, M_2]$ for any pump control sequences. Accordingly, the longest rescue times can be calculated by the following formula:

$$m = min\{n \mid \sum_{i=1}^{n} S^i_{ABCD} \geq (M_2 - M_1)\}$$

Thus $5(m-1)$ is the optimal rescue time.

8.2 Optimal Specification and Design

In this section, we first present an optimal specification for the steam-boiler control system. The optimal design decisions are provided afterwards which can be proved to be correct implementation of the optimal specification. Besides, these design decisions also satisfy the *Safety, Stability,* and *Performance* specifications.

Optimization If the water sensor is broken, the whole system should not be stopped until it must do so. In other words, the rescue time should be the longest one among all the possible cases.

$$\Box(\downarrow WS; \lceil \neg WS \wedge R \rceil; \downarrow R \Rightarrow \ell = 5(m-1)) \qquad \textbf{(Opt)}$$

where $m = min\{n \mid (\sum_{i=1}^{n} S^i_{\Box} + S^{n+1}_{\triangle}) \geq (M_2 - M_1)\}$, and
$S^i_{\Box} = [25U_1U_2 + 5(U_1 - U_2)(v_i - v_{i-1}) - (v_i - v_{i-1})^2]/(U_1 + U_2)$,
$S^{n+1}_{\triangle} = 25(U_1 + U_2)/2$.

By analysis and calculation, we can deduce the following design decisions:

$$sp; \lceil R \wedge WS \rceil^* \wedge \mathbf{b}.q < N_1 + 5W \Rightarrow \mathbf{b}.p = P; true \qquad \textbf{(Des}_\mathbf{1}\textbf{)}$$

$$sp; \lceil R \wedge WS \rceil^* \wedge \mathbf{b}.q > N_2 - 5P \Rightarrow \mathbf{b}.p = 0; true \qquad \textbf{(Des}_\mathbf{2}\textbf{)}$$

$$C \wedge sp \wedge (\mathbf{b}.x > S_{\triangle} \wedge \mathbf{b}.x \geq y) \Rightarrow \mathbf{b}.p = S_{max}/5; true \qquad \textbf{(Des}_\mathbf{3}\textbf{)}$$

$$C \wedge sp \wedge (\mathbf{b}.y > S_{\triangle} \wedge \mathbf{b}.y \geq x) \Rightarrow \mathbf{b}.p = S_{min}/5; true \qquad \textbf{(Des}_\mathbf{4}\textbf{)}$$

$$C \wedge sp \wedge (\mathbf{b}.x < S_{\triangle} \wedge \mathbf{b}.y < S_{\triangle}) \Rightarrow (\mathbf{b}.p = (2S_{min} + S_{\triangle} + x - y)/10); true \qquad \textbf{(Des}_\mathbf{5}\textbf{)}$$

$$sp \wedge (\mathbf{b}.x > 0 \wedge \mathbf{b}.y > 0 \wedge \mathbf{b}.(x+y) \leq S_{\triangle}); \lceil \neg WS \rceil^* \Rightarrow \downarrow R; true \qquad \textbf{(Des}_\mathbf{6}\textbf{)}$$

where $x = M_2 - qc_2$, $y = qc_1 - M_1$, $C \equiv (\mathbf{b}.x > 0 \wedge \mathbf{b}.y > 0 \wedge \mathbf{b}.(x+y) > S_{\triangle}); \lceil R \wedge \neg WS \rceil^*$.

Using these design decisions, we can prove that the implementation satisfies *Safety, Stability, Performance* and *Optimization* specifications.

Theorem 7

$$Assum \wedge Law \wedge Des_1 \wedge Des_2 \Rightarrow Safety,$$

where *Safety* is defined as before:

$$\Box((((\lceil q < M_1 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5) \wedge ((\lceil q > M_2 \rceil \wedge \lceil R \rceil) \Rightarrow l \leq 5))$$

Theorem 8

$$Assum \wedge Law \wedge Des_1 \wedge Des_2 \Rightarrow Stability,$$

where *Stability* is defined as before:

$$\exists\delta_1.\exists\delta_2.(\delta_1 < \mathbf{b}.q < \delta_2 \wedge (\uparrow R \vee \top R)); \lceil WS \rceil) \Rightarrow \lceil R \wedge (N_1 \leq q \leq N_2) \rceil$$

Theorem 9

$$Assum \wedge Law \wedge Des_1 \wedge Des_2 \Rightarrow Performance,$$

where *Performance* is defined as before:

$$\Box((\mathbf{b}.q = q_0) \wedge \downarrow WS; \lceil \neg WS \rceil; \uparrow Stop)$$
$$\Rightarrow l \geq min\{\lfloor \tfrac{M_2 - q_0}{2P} \rfloor + \lfloor \tfrac{M_2 - M_1}{W} \rfloor,\ \lfloor \tfrac{q_0 - M_1}{W} \rfloor + \lfloor \tfrac{M_2 - M_1}{2P} \rfloor\})$$

Theorem 10

$$Assum \wedge Law \wedge Des_3 \wedge Des_4 \wedge Des_5 \wedge Des_6 \wedge Des_7 \Rightarrow Optimization$$

where $Optimization \mathrel{\widehat{=}} \Box(\downarrow WS; \lceil \neg WS \wedge R \rceil; \downarrow R \Rightarrow \ell = 5(m-1))$ and
$m = min\{n \mid (\sum_{i=1}^{n} S_{\Box}^{i} + S_{\triangle}^{n+1}) \geq (M_2 - M_1)\}$, and
$S_{\Box}^{i} = [25U_1U_2 + 5(U_1 - U_2)(v_i - v_{i-1}) - (v_i - v_{i-1})^2]/(U_1 + U_2)$,
$S_{\triangle}^{n+1} = 25(U_1 + U_2)/2$.

For the proof details, please refer to [9].

9 Conclusions and Discussions

Using the steam-boiler system [1] as a testing case, this paper has discussed a general approach to specify and verify hybrid control systems. We first established a set of mathematical laws to characterize sampling, estimation and dynamic behaviors of the physical devices in the steam-boiler system. Based on this, we have specified *Safety, stability, Performance* and *Optimization* requirements. Functional design [11] in terms of communicating automata was given which captured the interaction between the physical plant and the decision-maker (controlling program). After providing semantics for these communicating automata, we have formally proved they are correct design in the sense that desired specifications can be derived from these design and mathematical laws. Thus the implementation based on these design will conform to the user requirements for the steam-boiler control system.

One feature of our approach is to combine operational style specification, say finite state machines, with descriptive style specification, i.e., Mean Value Duration Calculus. The benefits of such combination include taking advantages of explicitness and concreteness of operational style specification and abstraction, formality and deductive power from descriptive style specification. On the other hand, this paper provides a general method to cope with specification and verification of hybrid control systems.

Our approach corresponds to the first two layers, *requirements specification* and *functional design*, in a five-layer development approach proposed by Professor Anders P. Ravn [11]. At such development layers, [12] presents a modular description of the steam-boiler system using Duration Calculi [5]. One difference between [12] and our approach is that [12] uses *Extended Duration Calculus* while we choose *Mean Value Duration Calculus* as the specification language and underlying logic. From our experience we found Mean Value Duration Calculus is suitable for describing hybrid control systems containing critical events which occur at *instantaneous* time points instead of having *durations* on certain time intervals. Examples of such events include *open* or *close* pumps, *sample* data, *shut-down* the system etc. Schenke's work [13] illustrates transformations from Duration Calculus specifications to an architecture design with OCCAM-like processes. It is interesting to integrate our approach with methods proposed in [11, 12, 13] and complete the whole hierarchy of five-layer development approach proposed in [11].

Some problems in specifying and verifying hybrid systems generally need to be pursued furthermore: First, requirements capturing should have systematic theory support so that a reasonable formal specification can be derived from an informal description for a hybrid control system. In the steam-boiler

control problem, for instance, *Safety* specification certainly does not capture all the requirements. Implicit requirements such as *Stability, Performance, Optimization* etc should be specified in addition to *Safety*. Unfortunately, as Professor Egon Börger et al point out in [14], we could not negotiate with the designers and real users to elicit requirements and validate them. Second, different criteria can be given in defining optimal design. Based on different *rescue strategies*, we should find different optimization specifications and design decisions. Third, we have only dealt with *local stability* property of the steam-boiler control. More generally, we should cope with *global stability* or *asymptotic stability* property of the system. For instance, by using Infinite Duration Calculus[9] formulas, the following global stability requirements should be specified and implemented:

$$\exists x.(l > x \Rightarrow true; \lceil N_1 < q < N_2 \rceil)^i.$$

It means that starting at any system input, the control system can guarantee an ideal output eventually. However, this specification can not be expressed by Mean Value Duration Calculus. Finally, transformations from Mean Value Duration Calculus specification into software architecture and furthermore into program implementation need further effort.

Acknowledgment

Many of these ideas have been improved by discussions with Prof. Zhou ChaoChen. The authors would like to thank him for valuable suggestions and guidance. We would also like to thank Prof. Dines Bjørner for his support, help and continuing leadership and nurture of an excellent research environment in UNU/IIST, which have made this research possible. Special thanks are due to Prof. Michael R. Hansen, Prof. Anders P. Ravn, Prof. Hans Rischel and Simon Mørk of DTU for their many useful criticisms and suggestions.

References

[1] J.-R. Abrial, Steam-boiler control specification problem, August 10, 1994.

[2] A. P. Ravn et al, Specifying and Verifying Requirements of Real-time Systems, *IEEE Trans. Software Eng.* ,vol. 19, no. 1, pp. 41 – 55, Jan. 1993.

[3] Zhou Chaochen, C.A.R.Hoare, and A.P.Ravn, A Calculus of Durations, *Information Processing Letters*, 40(5):269–276, 1992.

[4] Zhou Chaochen and Li Xiaoshan: A Mean Value Calculus of Durations, in *A Classical Mind.* Prentice-Hall, pages 431-451,1994.

[5] Zhou Chaochen, Duration Calculi: An Overview, In *the Proceedings of Formal Methods in Programming and Their Applications*, LNCS 735, pp. 256–266, Springer Verlag, July 1993.

[6] A. C. Shaw, Communicating Real-Time State Machines, *IEEE Trans. Software Eng.* ,vol. 18, no. 9, pp. 805 – 816, Sept. 1992.

[7] Zong Ji Chen, Ji Wang and Zhou Chaochen, An Abstraction of Hybrid Control Systems, *UNU/IIST Report Draft*, June 1994.

[8] Wang JuAn, Li XiaoShan, A Duration Calculus Approach to Specifying a Steam-boiler Control System, *UNU/IIST Technical Report 38*, February 1995.

[9] Li XiaoShan, Wang JuAn, Specifying Optimal Design of a Steam-boiler Control System, *UNU/IIST Technical Report 39*, February 1995.

[10] Zhou Chaochen, Dang Van Hung and Li Xiaoshan, A Duration Calculus with Infinite Intervals, *UNU/IIST Technical Report Draft*, February 1995.

[11] Anders P. Ravn, Varieties of Steam Boilers, Technical University of Denmark, *Dagstuhl seminar paper*, June 1995.

[12] Simon Mørk, Anders P. Ravn, Hans Rischel, A Modular Approach to the Steam Boiler Problem using Duration Calculus, Technical University of Denmark, *Dagstuhl seminar paper*, June 1995.

[13] Michael Schenke, The Steam Boiler: From Duration Calculus to Occam, Universität Oldenburg, *Dagstuhl seminar paper*, June 1995.

[14] Egon Börger et al, Evolving Algebra Solution for the Steam boiler Problem, *Dagstuhl seminar paper*, June 1995.

Appendix A: Proof of Theorem 3

Lemma 1 $l \leq 5 \Rightarrow \mathit{Stability}$.

Proof:

(1). $l \leq 5 \Rightarrow sp; \lceil\neg sp\rceil; sp \vee \lceil\neg sp\rceil; sp; \lceil\neg sp\rceil$ Samp-Law

(2). Let $\delta_1 = N_1 + 10W, \delta_2 = N_2 - 10P$, then we have:
$(sp; \lceil\neg sp\rceil; sp) \wedge ((\delta_1 < \mathbf{b}.q < \delta_2) \wedge \lceil R\rceil^0 \wedge \lceil R\rceil^0; \lceil WS\rceil)$
$\Rightarrow (sp; \lceil\neg sp\rceil; sp) \wedge \lceil \delta_1 < q < \delta_2 \wedge WS\rceil$ Phy-Law, $Assum_1$
$\Rightarrow sp; \lceil\neg sp \wedge \delta_1 < q < \delta_2 \wedge WS\rceil; sp$ Distributivity
$\Rightarrow (ppo_1 \wedge ppf_2); \lceil\neg sp \wedge WS\rceil; sp$ DM
$\Rightarrow ((\downarrow FF \wedge \uparrow OF) \vee (\downarrow OF \wedge \uparrow IF) \vee (\downarrow IF \wedge \uparrow IF); \lceil\neg sp \wedge WS\rceil; sp$ SB
$\Rightarrow (\lceil N_1 + 5W < q < N_2 - 10P\rceil \vee \lceil N_1 + 10W + 5P < q < N_2 - 5P\rceil$
$\vee \lceil N_1 + 10W + 10P < q < N_2\rceil) \wedge \lceil\neg sp \wedge WS\rceil; sp$ Phy-Law, $Assum_1$
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil \wedge \lceil R\rceil$ $Law, Assum, SB, DM$

(3). $(\lceil\neg sp\rceil; sp; \lceil\neg sp\rceil) \wedge (\delta_1 < \mathbf{b}.q < \delta_2) \wedge \lceil R\rceil^0; \lceil WS\rceil)$
$\Rightarrow (\lceil\neg sp \wedge WS\rceil \wedge \lceil N_1 + 5W < q < N_2 - 10P\rceil); (sp \wedge \lceil N_1 + 5W < q < N_2 - 10P\rceil^0$
$\wedge \lceil WS\rceil^0; \lceil\neg sp \wedge WS\rceil$ Phy-Law, $Assum_1$
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; sp \wedge (\lceil N_1 + 5W < q \leq N_1 + 10W\rceil^0$
$\vee \lceil N_1 + 10W < q < N_2 - 10P\rceil^0) \wedge \lceil WS\rceil^0; \lceil\neg sp \wedge WS\rceil$ $Assum_1$

(4). $\lceil N_1 \leq q \leq N_2\rceil; (sp \wedge \lceil N_1 + 5W < q \leq N_1 + 10W\rceil^0 \wedge \lceil WS\rceil^0); \lceil\neg sp \wedge WS\rceil$
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; ppo_{1,2}; \lceil\neg sp \wedge WS\rceil$ DM
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; ((\downarrow FF \wedge \uparrow OO) \vee (\downarrow OO \wedge \uparrow II) \vee (\downarrow II \wedge \uparrow II) \vee (\downarrow IO \wedge \uparrow II)$
$\vee (\downarrow IF \wedge \uparrow IO) \vee (\downarrow OF \wedge \uparrow IO)); \lceil\neg sp \wedge WS\rceil$ SB
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; (\lceil N_1 < q \leq N_1 + 10W\rceil \vee \lceil N_1 + 5W + 5P < q \leq N_1 + 10W + 5P\rceil$
$\vee \lceil N_1 + 5W + 10P < q \leq N_1 + 10W + 10P\rceil) \wedge \lceil\neg sp \wedge WS\rceil$ $Assum$, Phy-Law
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil \wedge \lceil R\rceil$ $Law, Assum, SB, DM$

(5). $\lceil N_1 \leq q \leq N_2\rceil; (sp \wedge \lceil N_1 + 10W < q < N_2 - 10P\rceil^0 \wedge \lceil WS\rceil^0); \lceil\neg sp \wedge WS\rceil$
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; ppo_1 \wedge ppf_2; \lceil\neg sp \wedge WS\rceil$ DM
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; ((\downarrow FF \wedge \uparrow OF) \vee (\downarrow OF \wedge \uparrow IF) \vee (\downarrow IF \wedge \uparrow IF)); \lceil\neg sp \wedge WS\rceil$ SB
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil; (\lceil N_1 + 5W < q < N_2 - 10P\rceil \vee \lceil N_1 + 10W + 5P < q < N_2 - 5P\rceil)$
$\wedge \lceil\neg sp \wedge WS\rceil$ $Assum$,Phy-Law
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil \wedge \lceil R\rceil$ $Law, Assum, SB, DM$

(6). $(\lceil\neg sp\rceil; sp; \lceil\neg sp\rceil) \wedge (\lceil N_1 + 10W < q < N_2 - 10P\rceil^0 \wedge \lceil R\rceil^0; \lceil WS\rceil)$
$\Rightarrow \lceil N_1 \leq q \leq N_2\rceil \wedge \lceil R\rceil$ (4), (5)

(7). $((sp;\lceil\neg sp\rceil;sp)\vee(\lceil\neg sp\rceil;sp;\lceil\neg sp\rceil))\wedge(\delta_1<\mathbf{b}.q<\delta_2)\wedge\lceil R\rceil^0;\lceil WS\rceil)$
$\Rightarrow\lceil N_1\leq q\leq N_2\rceil\wedge\lceil R\rceil$ (2), (6)

(8). $l\leq 5\Rightarrow Stability.$ (1), (7)

Lemma 2 $\quad l>5\Rightarrow Stability$

Proof:

Let δ_1 and δ_2 be defined as in Lemma 1. We prove Lemma 2 by induction on the number of sampling cycles. Suppose that there are k sampling cycles contained in the observation interval.

Basis: $k=2$

(1). $l>5\Rightarrow(sp;l=5;sp;\lceil\neg sp\rceil)\vee(\lceil\neg sp\rceil\wedge l<5;sp;l=5;sp;\lceil\neg sp\rceil^*)$ Samp-Law

(2). $l>5\wedge(\delta_1<\mathbf{b}.q<\delta_2)\wedge\lceil R\rceil^0;\lceil WS\rceil)$
$\Rightarrow((sp;l=5;sp;\lceil\neg sp\rceil)\vee(\lceil\neg sp\rceil\wedge l<5;sp;l=5;sp;\lceil\neg sp\rceil^*))$
$\quad\wedge(\delta_1<\mathbf{b}.q<\delta_2)\wedge\lceil R\rceil^0;\lceil WS\rceil)$ (1)

(3). $(sp;l=5;sp;\lceil\neg sp\rceil)\wedge(\lceil N_1+10W<q<N_2-10P\rceil^0\wedge\lceil R\rceil^0;\lceil WS\rceil)$
$\Rightarrow(sp\wedge\lceil N_1+10W<q<N_2-10P\rceil^0\wedge\lceil R\rceil^0\wedge\lceil WS\rceil^0;l=5;$
$\quad(sp\wedge\lceil WS\rceil^0);\lceil WS\wedge\neg sp\rceil$ $Assum_1$
$\Rightarrow(ppo_1\wedge ppf_2);l=5;(sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$ DM
$\Rightarrow((\downarrow FF\wedge\uparrow OF)\vee(\downarrow OF\wedge\uparrow IF)\vee(\downarrow IF\wedge\uparrow IF));l=5;(sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$ SB
$\Rightarrow true;l=5;((\lceil N_1+5W<q<N_2-10P\rceil^0\vee\lceil N_1+5W+5P<q<N_2-5P\rceil^0)$
$\quad\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$ Phy-Law
$\Rightarrow true;l=5;((\lceil N_1+5W<q<N_1+10W\rceil^0\vee\lceil N_1+10W\leq q<N_2-10P\rceil^0$
$\quad\vee\lceil N_1+5W+5P<q<N_1+10W\rceil\vee\lceil N_1+10W\leq q<N_2-5P\rceil^0)$
$\quad\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$ $Assum_1$

(4). $true;l=5;(\lceil N_1+5W<q<N_1+10W\rceil^0\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$
$\Rightarrow true;l=5;ppo_{1,2};\lceil\neg sp\wedge WS\rceil$ DM
$\Rightarrow true;l=5;((\downarrow FF\wedge\uparrow OO)\vee(\downarrow OO\wedge\uparrow II)\vee(\downarrow II\wedge\uparrow II)\vee(\downarrow IO\wedge\uparrow II)$
$\quad\vee(\downarrow IF\wedge\uparrow IO)\vee(\downarrow OF\wedge\uparrow IO));\lceil\neg sp\wedge WS\rceil$ SB
$\Rightarrow true;l=5;(\lceil N_1<q<N_1+10W\rceil\vee\lceil N_1+5W+5P<q<N_1+10W+5P\rceil$
$\quad\vee\lceil N_1+5W+10P<q<N_1+10W+10P\rceil)\wedge\lceil\neg sp\wedge WS\rceil$ Phy-Law
$\Rightarrow\lceil R\rceil\wedge\lceil N_1\leq q\leq N_2\rceil$ $Assum, Law, SB, DM, (3)$

(5).$true;l=5;(\lceil N_1+10W\leq q<N_2-10P\rceil^0\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$
$\Rightarrow true;l=5;(ppo_1\wedge ppf_2);\lceil\neg sp\wedge WS\rceil$ DM
$\Rightarrow true;l=5;((\downarrow FF\wedge\uparrow OF)\vee(\downarrow OF\wedge\uparrow IF)\vee(\downarrow IF\wedge\uparrow IF));\lceil\neg sp\wedge WS\rceil$ SB
$\Rightarrow true;l=5;(\lceil N_1+5W\leq q<N_2-10P\rceil\vee\lceil N_1+5W+5P<q<N_2-5P\rceil)$
$\quad\wedge\lceil\neg sp\wedge WS\rceil$ Phy-Law
$\Rightarrow true;l=5;(\lceil N_1\leq q\leq N_2\rceil\wedge\lceil\neg sp\wedge WS\rceil)$ computation
$\Rightarrow\lceil R\rceil\wedge\lceil N_1\leq q\leq N_2\rceil$ $Assum, Law, SB, DM, (3)$

(6). $true;l=5;(\lceil N_1+5W+5P<q<N_1+10W\rceil^0\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$
$\Rightarrow true;l=5;ppo_{1,2};\lceil\neg sp\wedge WS\rceil$ DM
$\Rightarrow true;l=5;((\downarrow FF\wedge\uparrow OO)\vee(\downarrow OO\wedge\uparrow II)\vee(\downarrow II\wedge\uparrow II)\vee(\downarrow IO\wedge\uparrow II)$
$\quad\vee(\downarrow IF\wedge\uparrow IO)\vee(\downarrow OF\wedge\uparrow IO));\lceil\neg sp\wedge WS\rceil$ SB
$\Rightarrow true;l=5;(\lceil N_1+5P<q<N_1+10W\rceil\vee\lceil N_1+5W+10P<q<N_1+10W+5P\rceil$
$\quad\vee\lceil N_1+5W+15P<q<N_1+10W+10P\rceil)\wedge\lceil\neg sp\wedge WS\rceil$ Phy-Law
$\Rightarrow\lceil R\rceil\wedge\lceil N_1\leq q\leq N_2\rceil$ $Assum, Law, SB, DM, (3)$

(7). $true;l=5;(\lceil N_1+10W\leq q<N_2-5P\rceil^0\wedge sp\wedge\lceil WS\rceil^0);\lceil\neg sp\wedge WS\rceil$
$\Rightarrow true;l=5;(ppo_1\wedge ppf_2);\lceil\neg sp\wedge WS\rceil$ DM
$\Rightarrow true;l=5;((\downarrow FF\wedge\uparrow OF)\vee(\downarrow OF\wedge\uparrow IF)\vee(\downarrow IF\wedge\uparrow IF));\lceil\neg sp\wedge WS\rceil$ SB

$\Rightarrow true; l = 5; (\lceil N_1 + 5W \leq q < N_2 - 5P \rceil \vee \lceil N_1 + 10W + 5P \leq q < N_2 \rceil)$
$\wedge \lceil \neg sp \wedge WS \rceil$ Phy-Law
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ $Assum, Law, SB, DM, (3)$

(8). $(sp; l = 5; sp; \lceil \neg sp \rceil) \wedge (\lceil N_1 + 10W < q < N_2 - 10P \rceil^0 \wedge \lceil R \rceil^0; \lceil WS \rceil$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ (3), (4), (5), (6), (7)

(9). $(\lceil \neg sp \rceil \wedge l < 5; sp; l = 5; sp; \lceil \neg sp \rceil^*) \wedge (\lceil N_1 + 10W < q < N_2 - 10P \rceil^0 \wedge \lceil R \rceil^0; \lceil WS \rceil$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ similar to (8)

(10). $(l > 5 \wedge \lceil N_1 + 10W < q < N_2 - 10P \rceil^0 \wedge \lceil R \rceil^0; \lceil WS \rceil$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ (2), (8), (9)

Induction: Suppose for all $k < n$ Lemma 2 is true and now consider $k = n$.

(1). $(l > 5) \wedge (\lceil N_1 + 10W < q < N_2 - 10P \rceil^0 \wedge \lceil R \rceil^0; \lceil WS \rceil)$
$\Rightarrow true; sp_{n-1}; l = 5; sp_n; \lceil \neg sp \wedge WS \rceil$ Samp-Law
$\Rightarrow true; sp_{n-1}; \lceil N_1 \leq q \leq N_2 \rceil \wedge l = 5; sp_n; \lceil \neg sp \wedge WS \rceil$ induction hypothesis
$\Rightarrow true; sp_{n-1}; ((\lceil N_1 \leq q < N_1 + 5W \rceil \vee \lceil N_1 + 5W \leq q < N_1 + 10W \rceil$
$\vee \lceil N_1 + 10W \leq q < N_2 - 5W \rceil \vee \lceil N_2 - 5W \leq q \leq N_2 \rceil)$
$\wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$ $Assum_1$

(2). $true; sp_{n-1}; (\lceil N_1 \leq q < N_1 + 5W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
$\Rightarrow true; (sp_{n-1} \wedge ((\downarrow OO \wedge \uparrow II) \vee (\downarrow II \wedge \uparrow II) \vee (\downarrow IO \wedge \uparrow II) \vee (\downarrow IF \wedge \uparrow IO)$
$\vee (\downarrow OF \wedge \uparrow IO)); (\lceil N_1 \leq q < N_1 + 5W \rceil$
$\wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$ $Assum_1$, SB, DM
$\Rightarrow true; sp_{n-1}; l = 5; sp_n; \lceil N_1 < q < N_1 + 5W + 10P \rceil \wedge \lceil \neg sp \wedge WS \rceil$ $Assum_1$, Phy-Law
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ induction hypothesis, SB

(3). $true; sp_{n-1}; (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
$\Rightarrow true; ((ppo_{1,2} \wedge sp_{n-1}) \vee (ppo_1 \wedge ppf_2 \wedge sp_{n-1})); (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil$
$\wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$ DM
$\Rightarrow (true; \lceil N_1 \leq q < N_1 + 10W \rceil; (ppo_{1,2} \wedge sp_{n-1}); (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil$
$\wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$
$\vee (true; \lceil N_1 + 10W < q < N_2 - 10W \rceil; (ppo_1 \wedge ppf_2$
$\wedge sp_{n-1}); (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$
Distribution, Induction hypothesis, $Assum_1$

(4). $(true; \lceil N_1 \leq q < N_1 + 10W \rceil; (ppo_{1,2} \wedge sp_{n-1}); (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil$
$\wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$
$\Rightarrow true; \lceil N_1 \leq q < N_1 + 10W \rceil; (sp_{n-1} \wedge ((\downarrow OO \wedge \uparrow II) \vee (\downarrow II \wedge \uparrow II)$
$\vee (\downarrow OI \wedge \uparrow II) \vee (\downarrow IF \wedge \uparrow IO) \vee (\downarrow OF \wedge \uparrow IO)));$
$(\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$ $Assum_1$, SB, DM
$\Rightarrow true; sp_{n-1}; l = 5; sp_n; (\lceil N_1 < q < N_1 + 10W + 10P \rceil \wedge \lceil \neg sp \wedge WS \rceil)$ $Assum_1$, Phy-Law
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ induction hypothesis, SB

(5). $(true; \lceil N_1 + 10W \leq q < N_2 - 10W \rceil; (ppo_1 \wedge ppf_2$
$\wedge sp_{n-1}); (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$
$\Rightarrow true; \lceil N_1 + 10W < q < N_2 - 10W \rceil; (sp_{n-1} \wedge ((\downarrow FF \wedge \uparrow OF) \vee (\downarrow OF \wedge \uparrow IF) \vee (\downarrow IF \wedge \uparrow IF));$
$(\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil)$ $Assum_1$, SB, DM
$\Rightarrow true; sp_{n-1}; l = 5; sp_n; (\lceil N_1 + 5W + 5P \leq q < N_1 + 10W + 5P \rceil$
$\wedge \lceil \neg sp \wedge WS \rceil)$ $Assum_1$, Phy-Law
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ induction hypothesis, SB

(6). $true; sp_{n-1}; (\lceil N_1 + 5W \leq q < N_1 + 10W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ (3), (4), (5)

(7). $true; sp_{n-1}; (\lceil N_1 + 10W < q < N_2 - 5W \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ similar to (6)

(8). $true; sp_{n-1}; (\lceil N_2 - 5W \leq q \leq N_2 \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
$\Rightarrow true; (sp_{n-1} \wedge pp f_{1,2}); (\lceil N_2 - 5W \leq q \leq N_2 \rceil \wedge l = 5); sp_n; \lceil \neg sp \wedge WS \rceil$
induction hypothesis, $Assum_1$, DM
$\Rightarrow true; (sp_{n-1} \wedge pp f_{1,2}); (\lceil N_2 - 5W \leq q \leq N_2 \rceil \wedge l = 5); (sp_n \wedge pp f_{1,2}); \lceil \neg sp \wedge WS \rceil$ DM
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ induction hypothesis, SB, *Law*

(9). $(l > 5) \wedge ((\delta_1 < b.q < \delta_2) \wedge \lceil R \rceil^0; \lceil WS \rceil)$
$\Rightarrow \lceil R \rceil \wedge \lceil N_1 \leq q \leq N_2 \rceil$ (1), (2), (6), (7), (8)

Then Lemma 2 is proved.

Now we can apply the *Forward Induction Rule* to prove Theorem 3.

Let $R(X) \mathrel{\widehat{=}} (X \Rightarrow Stability)$.

Basis: $R(\lceil \rceil)$ is trivially true because the premise of Stability is false.

Induction: Suppose $R(X)$ is true, then

$$R(X \vee X; (l \leq 5) \vee (l > 5))$$

is true according to induction hypothesis, Lemma 1 and Lemma 2. Thus $R(true)$ holds. Theorem 3 is proved.

□

Appendix B: Proof of Theorem 4

Proof:

(1). $(\mathbf{b}.q = q_0) \wedge \downarrow WS; \lceil \neg WS \rceil; \uparrow Stop$
$\Rightarrow \lceil q = q_0 \rceil^0 \wedge \downarrow WS; \lceil \neg WS \rceil; \uparrow Stop$ $Assum_1$
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); \lceil \neg WS \rceil; \uparrow Stop$ $Axiom \downarrow$
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (sp \wedge stop)$ $Axiom \uparrow$, SB
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_1 \leq M_1 \vee qc_2 \geq M_2 \rceil^0 \wedge sp \wedge stop)$ DM

(2). $(\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_1 \leq M_1 \rceil^0 \wedge sp \wedge stop)$
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); sp; (l = 5 \wedge \lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_1 \leq M_1 \rceil^0$
$\wedge sp \wedge stop)$ Samp-Law
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (sp \wedge \lceil qc_1 > M_1 \rceil^0 \wedge \lceil \neg WS \rceil^0$
$\wedge \lceil R \rceil^0); (l = 5 \wedge \lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_1 \leq M_1 \rceil^0 \wedge sp \wedge stop)$ DM
$\Rightarrow l \geq \lfloor \frac{M_2 - q_0}{2P} \rfloor + \lfloor \frac{M_2 - M_1}{W} \rfloor$ $Assum, Law$,DM, SB

(3). $(\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_2 \geq M_2 \rceil^0 \wedge sp \wedge stop)$
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); sp; (l = 5 \wedge \lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_2 \geq M_2 \rceil^0$
$\wedge sp \wedge stop)$ Samp-Law
$\Rightarrow (\lceil q = q_0 \rceil^0 \wedge \lceil WS \rceil^0); (\lceil \neg WS \rceil \wedge \lceil R \rceil); (sp \wedge \lceil qc_2 < M_2 \rceil^0 \wedge \lceil \neg WS \rceil^0$
$\wedge \lceil R \rceil^0); (l = 5 \wedge \lceil \neg WS \rceil \wedge \lceil R \rceil); (\lceil qc_2 \geq M_2 \rceil^0 \wedge sp \wedge stop)$ DM
$\Rightarrow l \geq \lfloor \frac{q_0 - M_1}{W} \rfloor + \lfloor \frac{M_2 - M_1}{2P} \rfloor$ $Assum, Law$,DM, SB

(4). $(\mathbf{b}.q = q_0) \wedge \downarrow WS; \lceil \neg WS \rceil; \uparrow Stop$
$\Rightarrow l \geq min\{\lfloor \frac{M_2 - q_0}{2P} \rfloor + \lfloor \frac{M_2 - M_1}{W} \rfloor, \lfloor \frac{q_0 - M_1}{W} \rfloor + \lfloor \frac{M_2 - M_1}{2P} \rfloor\}$ (2), (3)

□