MODELS OF LONGITUDINAL WEB BEHAVIOR IN THE HANDLING PROCESS

By

MASAKAZU AKATSUKA

Bachelor of Engineering

Kyoto University

Kyoto, Japan

1982

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 1990

MODELS OF LONGITUDINAL WEB BEHAVIOR

IN THE HANDLING PROCESS

Thesis Approved

Gary E. Young

Thesis Advisor

John J. Shelton

[illegible]

Norman N. Durham

Dean of the Graduate College

ACKNOWLEDGEMENTS

There are really many people who have supported, and encouraged me to study here. No exaggeration lies in my appreciation to all of them, though all names may not appear.

Out of these people, I wish to extend my sincere gratitude to my academic adviser, Dr. Gary E. Young for his guidance in my work. I learned many invaluable things from his advice including his lecture. Without this background, I could not come to this mark.

Having excellent faculty members, I could enrich my life and the study in OSU. I would like to take this opportunity to thank my thesis committee members, Dr. Peter M. Moretti and Dr. John J. Shelton for their advisement in this work. Dean Dr. Karl N. Reid gave me the precious advice about the research and the activities in this college despite his busy schedule.

I could not even be in this University, or in this country, without the great help of Mr. Richard J. Adams and Mr. Haruyoshi Fujiwara. In the beginning, they spent a lot of their precious time to make my study possible.

Reaching to my graduation, I would like to recognize Mitsubishi Heavy Industries, ltd. for their employee cultivating program and perfect support for my convenience. I am deeply indebted to their people in the Headquarters, Mihara Machinery Works, and Hiroshima R&D Center for allowing me to have this once in a lifetime experience.

One of them is Dr. Yoshirou Takahashi, former General manager of MHI Hiroshima Technical Institute, who recommended I to this program so that his man could thrive.

My gratitude is also to the upper echelon in Mihara office, Mr. Shinichi Emi, Mr. Hideaki Nomoto, Mr. Toshiya Oyamoto, Mr. Toyoo Nimoda, Mr. Kenji Suzuki, and Mr. Akishiro Yoshihara who have encouraged me to study abroad in this busiest time.

In any event, I owe much to my family and friends in Japan, and thank all of them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A	cross section area of web	(m^2)
a1, a2,..., ai	speed ratio of i-th roller to roller 0	
e0, e1,...,ei	strain before the roller i	
$\overline{e0}$, $\overline{e1}$	steady state value of e0 and e1	
$\widetilde{e0}$, $\widetilde{e1}$	changes from the steady state value	
E1, E2,..., Ei	viscous coefficient of i-th Voigt part	(Ns/m^2)
E1m	E1 in Maxwell model	(Ns/m^2)
eg	strain of elastic part	
Es	viscous coefficient of plastic part	(Ns/m^2)
E[x]	expectation of variable x	
G1, G2,..., Gi	spring coefficient of i-th Voigt part	(N/m^2)
G1m	G1 in Maxwell model	(N/m^2)
Gg	spring coefficient for elastic part	(N/m^2)
Ggm	Gg in Maxwell model	(N/m^2)
h(t)	impulse response function	
H(w)	frequency response function	
$H^*(w)$	complex conjugate of H(w)	
i	$\sqrt{-1}$	
L1, L2,..., Li	length of i-th span	(m)
N	total number of elements of Voigt part	
r0, r1,...,ri	$\rho * A$ in i-th span	(kg/m)

$R_x(\tau)$	auto-correlation function of variable x	
S0	spectral density of $\widetilde{\varepsilon 0}$	(s)
S0(w), S1(w)	spectral density function of $\widetilde{\varepsilon 0}$ and $\widetilde{\varepsilon 1}$	
s1, s2,..., si	stress in the i-th span	(N/m^2)
S	Laplace operator	
$S_x(w)$	spectral density function of variable x	
t	time	(s)
T	$\frac{l1}{a1*v0}$	(s)
v	speed of web	(m/s)
v0	speed of web at roller 0	(m/s)
w	angle velocity	(rad/s)
x	length from roller 0	(m)
x(t)	input function	
x1, x2	starting and ending point of the dryer	(m)
y(t)	response function	
Δ	changing ratio of Gg in the dryer	(1/m)
ε	total strain of the model	
$\varepsilon 1, \varepsilon 2, \ldots, \varepsilon i$	strain of i-th Voigt part	
$\varepsilon 0$	initial total strain of the model	
$\varepsilon 10, \varepsilon 20, \ldots, \varepsilon i0$	initial strain of i-th Voigt part	
ε	Laplace transform of ε	
εg	strain of elastic part in Voigt model	
$\hat{\varepsilon}$	$\varepsilon - \varepsilon 0$	
εs	strain of plastic part in Voigt model	
$\varepsilon s0$	initial strain of plastic part in Voigt model	
ρ	density of web	(kg/m^3)

μ	friction coefficient	
μ_1	friction coefficient (high)	
μ_2	friction coefficient (low)	
σ	loading stress of the model	(N/m^2)
$\sigma 0$	initial loading stress of the model	(N/m^2)
$\sigma 1$	new loading stress after step change at time 0	(N/m^2)
σ	Laplace transform of σ	
σg	stress of elastic part in Voigt model	(N/m^2)
$\hat{\sigma}$	$\sigma 1 - \sigma 0$	(N/m^2)
σs	stress of plastic part in Voigt model	(N/m^2)
τ	time, lag time	(s)
θ	lapping angle around the roller	(rad)
θ	$t - \tau$	(s)

CHAPTER I

INTRODUCTION

Every thing has its surface and volume. They are inseparable but some things are useful for their surface while others for their volume. A spherical body is ideal to maximize its volume for a given surface area. The opposite of this may be a form of membrane or thin stuff. Paper, for example, has a much larger surface area than wood of same weight has, though their components are almost the same. Information is recorded basically on the surface. Thus paper can record more information than wood can for any given weight. So does magnetic tape and wrapping film, the surface areas of which are important for recording or covering things.

From the industrial point view, requirements for this type of thin stuff are that it be thinner, stronger, and more uniform, and that it have better quality, especially on its surface. If the surface area is the concern, a thinner product saves raw materials and reduces transportation and storage costs. Uniformity is also important because irregularity in property or thickness, thus in strength, causes problems, especially when the product becomes smaller or narrower, where any defect is more critical. Similarly there is a requirement for manufacturing and handling systems. Thin stuff is mass-production oriented because of its uniform nature and its relatively low unit price. This fact yields the need for faster and larger scale machines with high enough precision to meet the requirements for the products.

The web is a convenient form for manufacturing thin stuff to meet this requirement. After manufacturing, moreover, handling is easy since it runs inside machines by itself even at a high speed. Transportation and storage are also convenient since it comes in the form of a roll. Then we can use it as a web again or can make cut-sheets from it whenever necessary. Once cut, no stuff with same thickness can be treated the same.

Studies regarding the web can be classified into two groups, which are web handling and web manufacturing or processing. Even after many studies, there remain theoretical and practical problems in these fields. New problems, moreover, have been raised with these new requirements, and they are becoming more complicated because of the nature of large scale manufacturing. In web handling, problems are classified into longitudinal (machine direction), lateral (cross machine direction), and out of plane problems. These correspond to each dimension of the web movement. Among them, the longitudinal problem has been studied for a long time, partly because web tension is the key factor which causes problems such as breaks, wrinkles, and lateral motions.

This paper discusses longitudinal problems in web handling systems. Since this field has varieties in its scope, this paper is designed as follow. First the basic equation of the propagation is presented in Chapter II. This is the principle relation which governs all the phenomena shown in this paper. The next two chapters are dedicated to discussions of web models. Researchers used to use an elastic model because of its simple expression, but extended models are necessary for explaining things which a simple model cannot handle. Chapter III discusses the elastic model and its behavior in the handling system. Chapter IV expands this discussion to visco-elastic models and their behavior. The next two chapters are designed to show that even phenomena under the fundamental equations may cause strange behavior. Chapter V discusses the stick-slip phenomenon which generates undesirable periodic changes in web strain. Chapter VI shows the problem of slackness through results in the previous chapters. Apart from the deterministic treatment, Chapter VII discusses a statistical approach in order to deal with uncertainty in the system and web. The final summary and recommended further study is contained in Chapter VIII to conclude this paper. Since every chapter has a different aspect, each has a brief introduction and a chapter summary.

CHAPTER II

PROPAGATION OF STRAIN

Introduction

Every web handling machine has several devices, commonly rollers, with which it transports the web, and they separate the entire web into one or more spans. Consequently, it is useful to calculate the strain or the stress of the web inside each span and determine how it propagates from one span to another. This chapter discusses the basic equation on the strain propagation. This is preparation for the various applications which appear later in this paper.

Researchers [1][2][3] in web handling have already investigated the basic idea, which can be classified into two groups. The first gives the relation of strain, or it uses strain as the boundary. This is valid if the span has rollers which do not slip with the web. The second one gives that of stress, and this is useful when there is slippage on the roller in the span, so that the actual speed of the web is not available. It is possible to calculate both strain and stress in either way. However the relation of the strain gives easier understanding in many cases, especially if the handling system contains both slip and non slip condition. The latter case can be treated as two problems then. The first one is to determine the web speed through the relation of stress, and the second one to calculate the strain using the equations which appear in this chapter.

This chapter, therefore, puts the emphasis on the relation of the strain and its propagation from one span to another.

Case 1: Spans Without Slippage Between Rollers and Web

Figure 2.1 shows the typical picture of a web path in the handling machine. The web moves from point A to B through roller 0, 1, and 2. The first span (span 1) is defined with roller 0 and 1, and the web whose length, strain just before roller 1, and stress is L1, e1, and s1 respectively. Similarly, the second span (span 2) is given with roller 1 and 2, and the web whose length, strain just before roller 2, and stress is L2, e2, and s2 respectively. In addition to these, let the strain just before the roller 0 be e0. Notice that these values can be either variables or constant at this stage, and that the speed of the web just before each roller is given from that of the roller.

By observing the mass of the web inside span 1 in the time interval between 0 and t, the following relations are constructed.

$$\text{Incoming mass} = \int_0^t \frac{r0*v0}{1+e0} d\tau \tag{2.1}$$

$$\text{Outgoing mass} = \int_0^t \frac{r1*a1*v0}{1+e1} d\tau \tag{2.2}$$

$$\text{Change of mass} = \left.\frac{r1*L1}{1+e1}\right|_{\text{Time}=t} - \left.\frac{r1*L1}{1+e1}\right|_{\text{Time}=0} \tag{2.3}$$

These three equations yield;

$$\left.\frac{r1*L1}{1+e1}\right|_{\text{Time}=t} = \left.\frac{r1*L1}{1+e1}\right|_{\text{Time}=0} + \int_0^t \frac{r0*v0}{1+e0} d\tau - \int_0^t \frac{r1*a1*v0}{1+e1} d\tau \tag{2.4}$$

By differentiating (2.4) with respect to t;

$$\frac{d}{dt}\left[\frac{r1*L1}{1+e1}\right] = \left.\frac{r0*v0}{1+e0}\right|_{\text{Time}=t} - \left.\frac{r1*a1*v0}{1+e1}\right|_{\text{Time}=t} \tag{2.5}$$

Same is true to span 2. This is given as;

$$\frac{d}{dt}\left[\frac{r2*L2}{1+e2}\right] = \left.\frac{r1*a1*v0}{1+e1}\right|_{\text{Time}=t} - \left.\frac{r2*a2*v0}{1+e2}\right|_{\text{Time}=t} \tag{2.6}$$

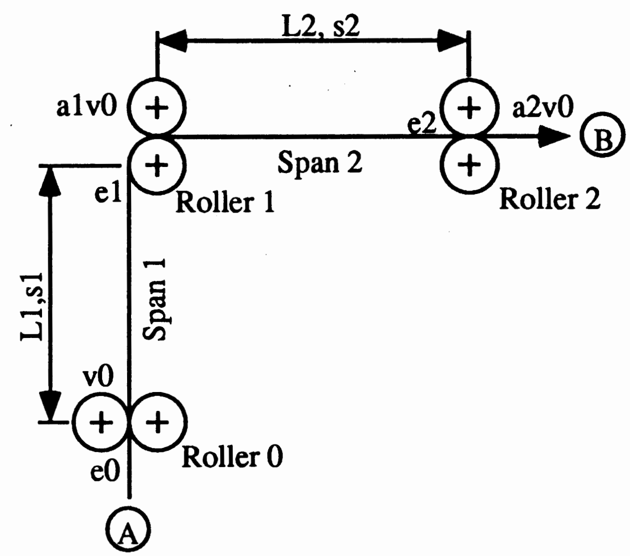

Figure 2.1. Web Path Without Slippage

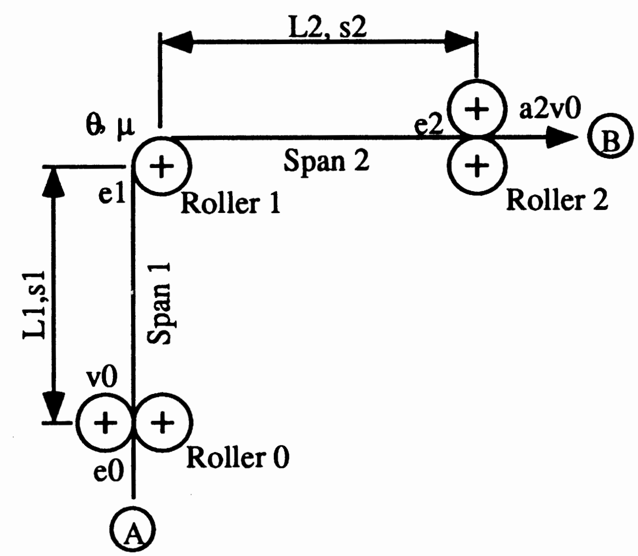

Figure 2.2. Web Path With Slippage

Case 2: Spans With Slippage Between Rollers and Web

Figure 2.2 shows the picture of a web path in the handling machine. The only difference between this and the one in Figure 2.1 is the fact that the web slips on roller 2 around which the web laps with angle θ.

In this case, equation (2.5) and (2.6) are still valid, but the value of a1 is not given directly. Consequently another relation is necessary to calculate the strain in the spans, and it is common to use the relationship of stress s1 and s2 through the roller 2.

Figure 2.3 indicates the value of the friction coefficient with respect to the slipping speed, or the speed difference between roller and web.[1] In the actual design, however, the range of this slippage is limited so that excess slippage may not degrade the surface of the web. Then it is agreeable to assume that the friction coefficient μ is constant when web is slipping. This yields the relationship between stress s1 and s2 to be

$$s1 = s2 * e^{\mu\theta} \qquad (2.7)$$

This relation is depicted in Figure 2.4.

Equation (2.7) gives the relation with respect to stress, not to strain, but it is possible to transform it to the relation of strain by using the appropriate models of the web. For example, strain is easily calculated only with the elastic modulus if the elastic web model is used.

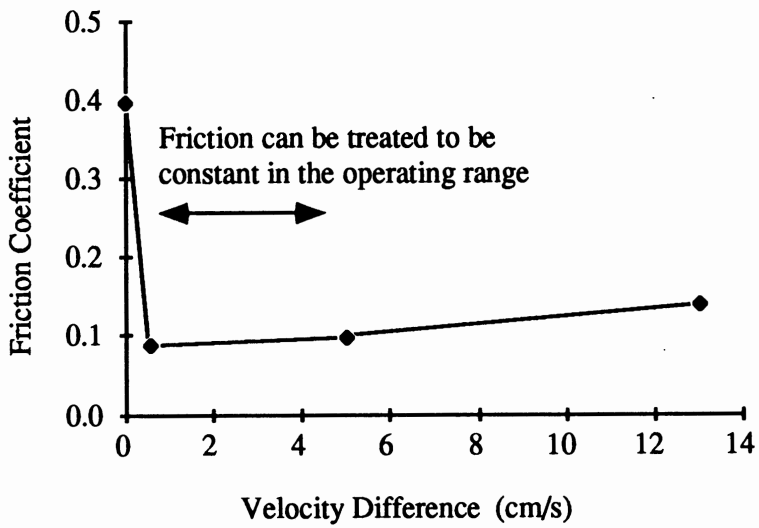

Figure 2.3. Friction Coefficient

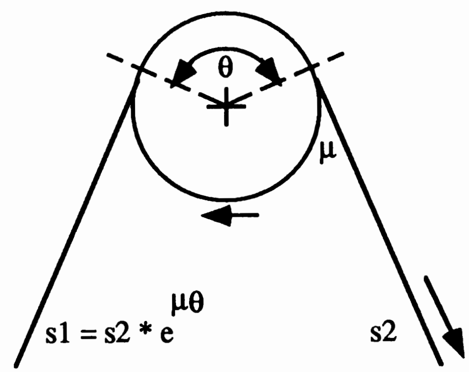

Figure 2.4. Relationship of Strain

Simplification of the Equation

In many cases, it is practically desirable and possible to simplify the equation (2.5) and (2.6) with some agreeable assumption. Followings are those commonly used:

(1) Density of the web ρ is constant in a span though it may change span by span.

(2) Length of the span, L1 and L2 (m), is constant. This is valid in many actual spans unless they have moving devices, such as dancer rollers.

(3) Strain, e0, e1, e2, are small. This is valid broadly; because the order of 0.1 or 0.3 % is the value for the normal paper handling.

(4) Cross sectional area A is constant in a span. This is valid with assumption (3).

Using assumption (1), (2), (4), and constant density through out spans, equation (2.5) and (2.6) yield (2.8) and (2.9) respectively.

$$L1 * \frac{d}{dt}\left[\frac{1}{1+e1}\right] = \left.\frac{v0}{1+e0}\right|_{Time=t} - \left.\frac{a1*v0}{1+e1}\right|_{Time=t} \tag{2.8}$$

$$L2 * \frac{d}{dt}\left[\frac{1}{1+e2}\right] = \left.\frac{a1*v0}{1+e1}\right|_{Time=t} - \left.\frac{a2*v0}{1+e2}\right|_{Time=t} \tag{2.9}$$

Using (3) with the Taylor series expansion $\frac{1}{1+e} = 1\text{-}e + \frac{1}{2}e^2 \text{-} \cdots \cong 1\text{-}e$, (e<<1);

$$-L1 * \frac{de1}{dt} = v0*(1-e0)\,|_{Time=t} - a1*v0*(1-e1)\,|_{Time=t} \tag{2.10}$$

$$-L2 * \frac{de2}{dt} = a1*v0*(1-e1)\,|_{Time=t} - a2*v0*(1-e2)\,|_{Time=t} \tag{2.11}$$

These equations are convenient for the practical use because of their linearity.

Chapter Summary

Equations which express propagation of the spans are shown. If there is no slippage on the rollers, equation (2.5) and (2.6) give the strain just before the rollers. Equation (2.7) should be incorporated with them if there is slippage. Simplified equations (2.8) - (2.11) are also given with the following assumptions.

(1) Density of the web ρ is constant in a span though it may change span by span.

(2) Length of the span, L1 and L2 (m), is constant.

(3) Strain, e0, e1, e2, are small.

(4) Cross sectional area A is constant in a span.

It should be noted again that these equations give the strain just before each roller. Elastic web models give the uniform strain in the span while visco-elastic models give distributed strain as is mentioned in a later chapter.

CHAPTER III

THE ELASTIC WEB MODEL AND TRANSPORTATION

Introduction

The equations shown in Chapter II give the relation of the strain in the span and the propagation from one span to another. It is useful itself, but it is necessary to use the proper web model to calculate the stress and tension in the span.

Researchers have commonly used the elastic web model for their study.[1][2] The main reasons are as follow.

(1) The elastic model consists of only one element (spring) for the expression. In this case, the elastic modulus of this web is the spring coefficient in the model. Conversion from strain to stress is easy because only the spring coefficient is needed. It is possible to assume uniform strain inside the span, and this is convenient for calculations.

(2) The elastic model does not contain any delaying element, damper for example. This makes dynamic analysis simple.

(3) In the actual web handling, the typical material of the web is paper or plastic film which is observed to be elastic under normal operation. The elastic model, therefore, is practically valid.

This chapter briefly discusses the behavior of the strain both in a steady state and in a dynamic condition. In addition, strain distribution of the web, which suffers changes in its elastic modulus, is discussed.

Web Model

Figure 3.1 shows the schematic drawing of the single span which has two couples of rollers at both limits and the web stretching in between them. The elastic model of the web is simply given as the spring of which the coefficient is Gg. Stress is given with this coefficient Gg and the strain εg. It should be noted that the spring represents a small segment of the entire span of the web and each segment moves to the right in this picture as time proceeds.

If Gg is constant inside the span, it is possible to assume that the strain and stress are uniformly distributed inside the span at each time. In this case, equations (2.8) and (2.9) can be calculated with respect to time only.

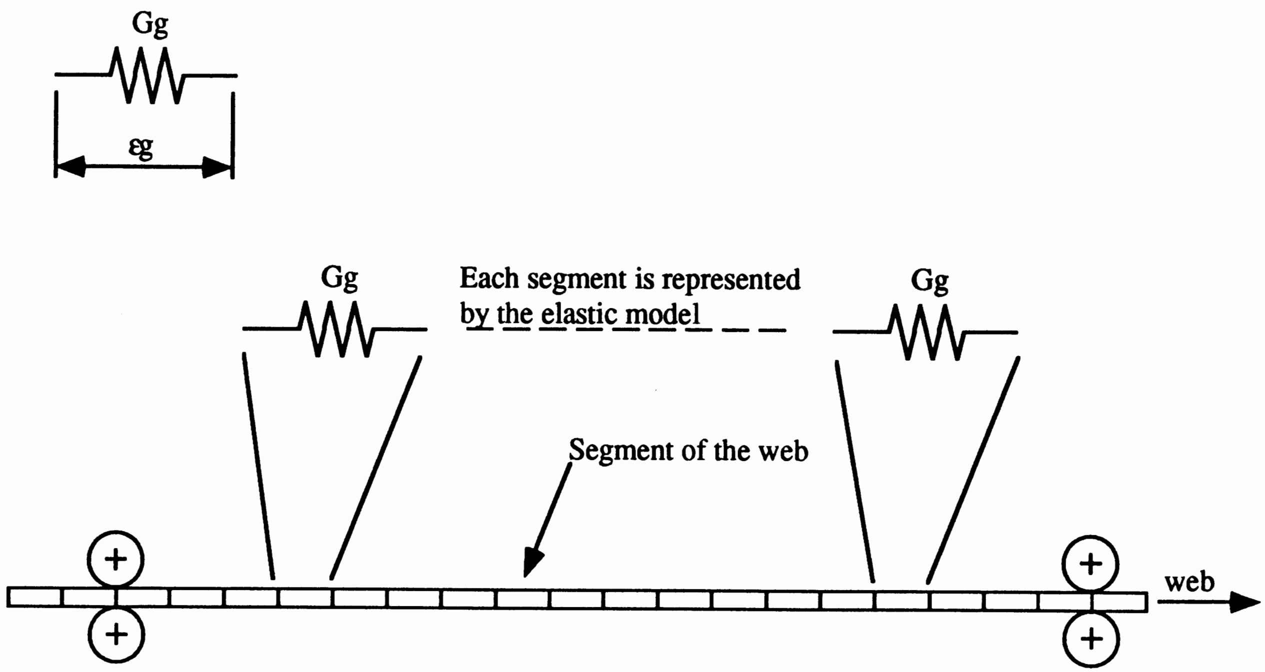

Figure 3.1. Elastic Model of the Web

Simulation of the Behavior of the Strain

Figure 3.2 and 3.3 show the results of a simulation with an elastic model. Figure 3.2 shows the strain in span 1 and 2 at a steady state. Uniformity in the strain distribution inside the span comes from the assumption made here.

Figure 3.3 shows the dynamic behavior of the strain in span 1 and 2 with respect to time. This indicates how strain e1 and e2 approach their steady state values.

The conditions of these simulations are as follow.

(1) Speed at roller 0; v0 = 20.0

(2) Strain before roller 0; e0 = 0.001

(3) Speed at roller 1; v1 = 1.0001 * v0 (time < 0.05)
= 1.0010 * v0 (afterward to steady state)

(4) Speed at roller 2; v2 = v0

(5) Length of span 1;L1 = 1.2

(6) Length of span 2;L2 = 2.0

Program 1 and 2 in Appendix B were used to simulate this behavior. Program 1 was originally developed to simulate the steady state strain distribution of visco-elastic model.(Chapter IV) Program 2 is for the dynamic analysis of the stick-slip phenomenon with an elastic model.(Chapter V) This program solves equations (2.8) and (2.9) simultaneously with the 4th order Runge-Kutta method either in slip or non slip operation.

These simulations are possible because the visco-elastic model includes the elastic model in it, and non slip operation is a special case of the stick-slip phenomenon. Operation with slippage is covered in Chapter V as a part of this unsteady operation.

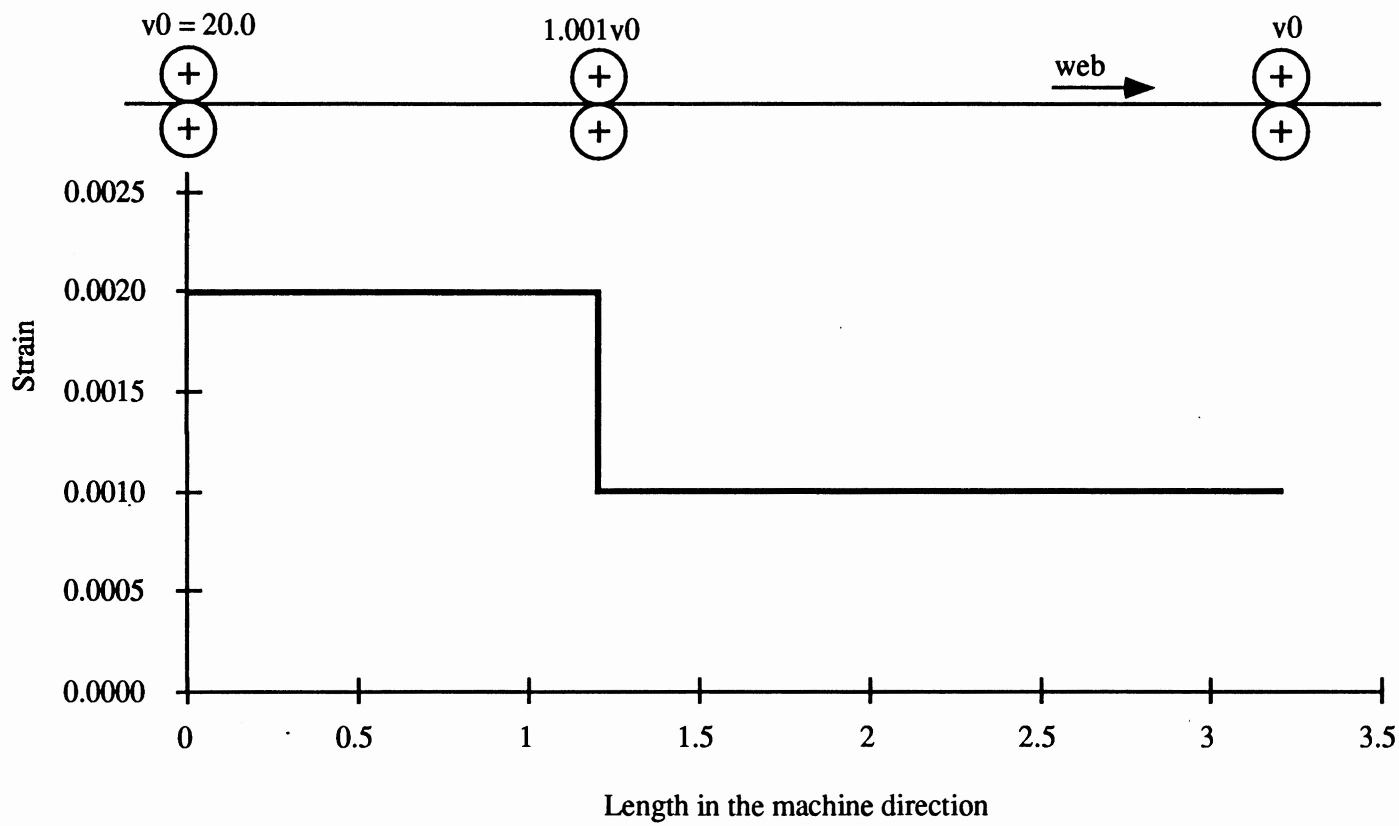

Figure 3.2. Strain in Span 1 and 2 at Steady State

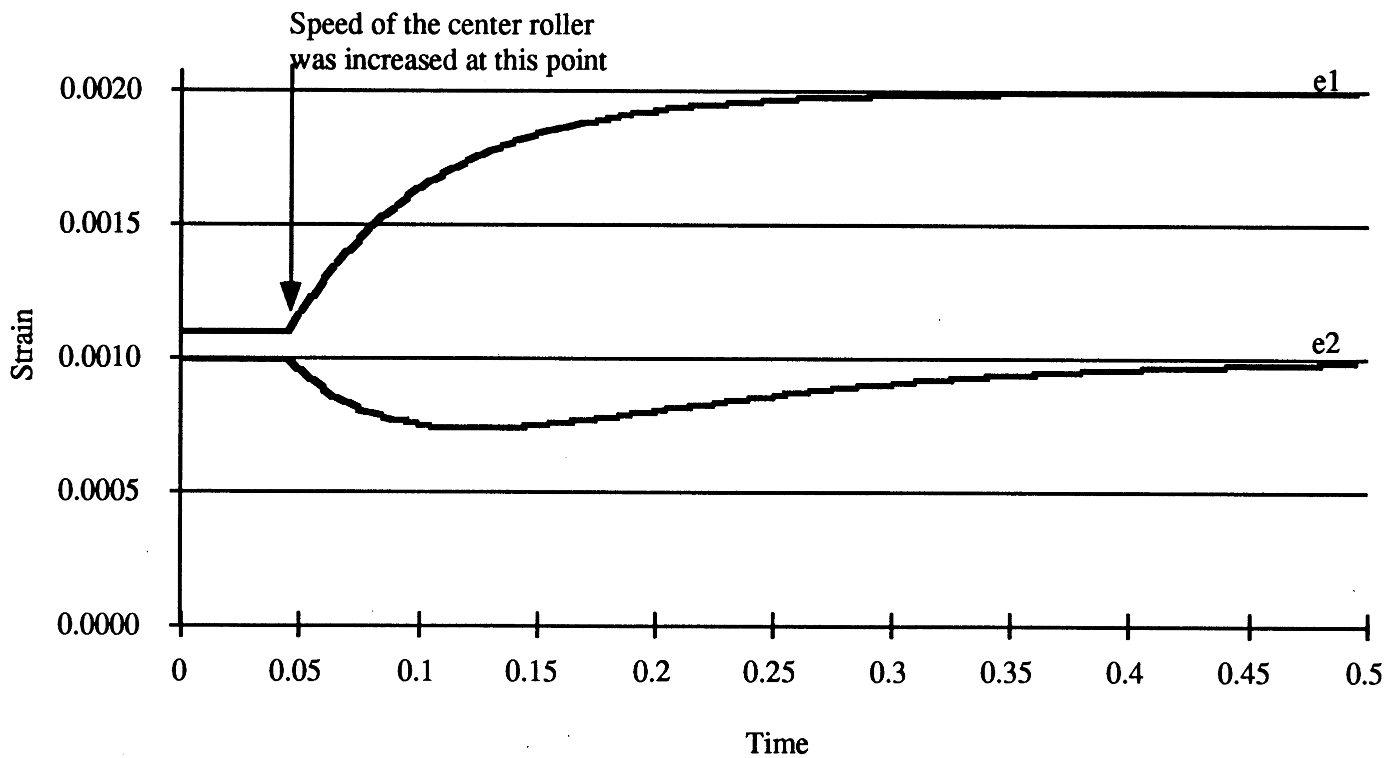

Figure 3.3. Strain in Span 1 and 2

Changes in Property

The elastic model does not require that the elastic modulus be constant, and there are two practical extensions from the discussion shown earlier. The first one is to allow strain before roller 0; e0 to be time variant which represents the disturbances outside the span. This is inescapable in the actual handling because there is some irregularity in the elastic modulus and/or the web tension before roller 0.[4] The second one is to allow changes inside the span. This simulates the span with a dryer or moisturizer which affects the elastic modulus of the web.

Disturbance Given Outside the Span

Equation (2.8) can handle this case with a given disturbance. Program 3 in Appendix B simulates the case where strain e0 suffers irregular disturbance. Figure 3.4 shows the result of this simulation. The condition of the calculation is the same as was used previously with exceptions that this has only one span (span 1) and that e0 is irregular.

Equation (2.10), which is a simplified version of (2.8), indicates that this propagation itself is a first order low pass filter for the strain with a cutoff frequency of $\frac{a1 * v0}{L1}$ (1/s). Actually, the result shows its filtering effect. This is preferable for the machines because disturbances of high frequency may not propagate from one span to another.[2]

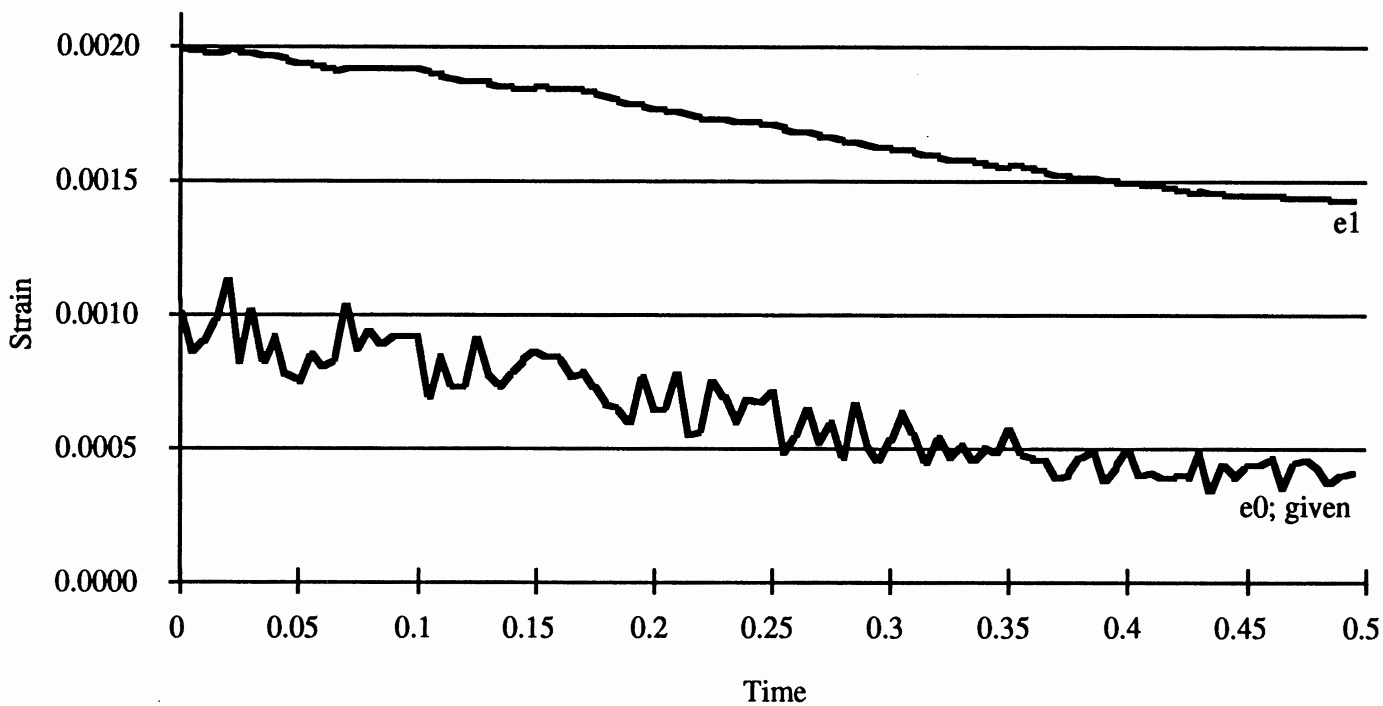

Figure 3.4. Strain in Span 1 With Disturbance in Given e0

Disturbance Given Inside the Span

Equation (2.8) is valid for the strain just before the roller and it does not guarantee a uniform distribution of strain in the span if the property of the web is not uniform. It is, however, impossible to determine the distribution only from equation (2.8) and the web model. The most agreeable assumption to supplement this is that the tension should be uniform in the span. This is based on the idea that the elastic element (spring) acts instantaneously. This is valid in almost all cases where the handling speed of the web is much slower than the speed of sound in the web.

Figure 3.5 shows a case where the spring coefficient changes in the span because of the drying process. Program 4 is used for this simulation. The purpose of this simulation is to demonstrate the idea, and the numbers are fictional. They are as follow.

(1) Assumption: Uniform tension in the span, in addition to those given in Chapter II. This gives the uniform stress s1 in the span.

(2) Equation of distribution: Let strain e1 and spring coefficient Gg be the function of the location in the span. The origin is at roller 0 (x=0) and the end point is at roller 1 (x=L1).

The strain at point A is given from (2.8) at its steady state.

$$e1(L1) = \frac{s1}{Gg(L1)} = a1 * (1 + e0) - 1 \qquad (3.1)$$

The strain at point B is given using Gg(x) as

$$e1(x) = \frac{s1}{Gg(x)} \qquad (3.2)$$

(3.1) and (3.2) yield

$$e1(x) = \frac{Gg(L1)}{Gg(x)} * (a1 * (1 + e0) - 1) \tag{3.3}$$

In actual use, the value of $\frac{Gg(L1)}{Gg(x)}$ should be gained either from experiments or theoretical analysis. In this case, however, it is given as follow.

$$\frac{Gg(L1)}{Gg(x)} = \begin{cases} 1 - \Delta * (x2 - x1) & 0 \le x \le x1 \\ 1 - \Delta * (x2 - x) \quad \text{for} & x1 \le x \le x2 \\ 1 & x2 \le x \end{cases} \tag{3.4}$$

where Δ is a changing ratio of Gg for a unit length of the dryer.

Equations (3.3) and (3.4) give the distribution of strain in the span.

(3) Condition of calculation: The same condition is used as was used in the previous one.

Values of x1 and x2 are set as 0.5 and 0.8 respectively. Δ is given as 2.0.

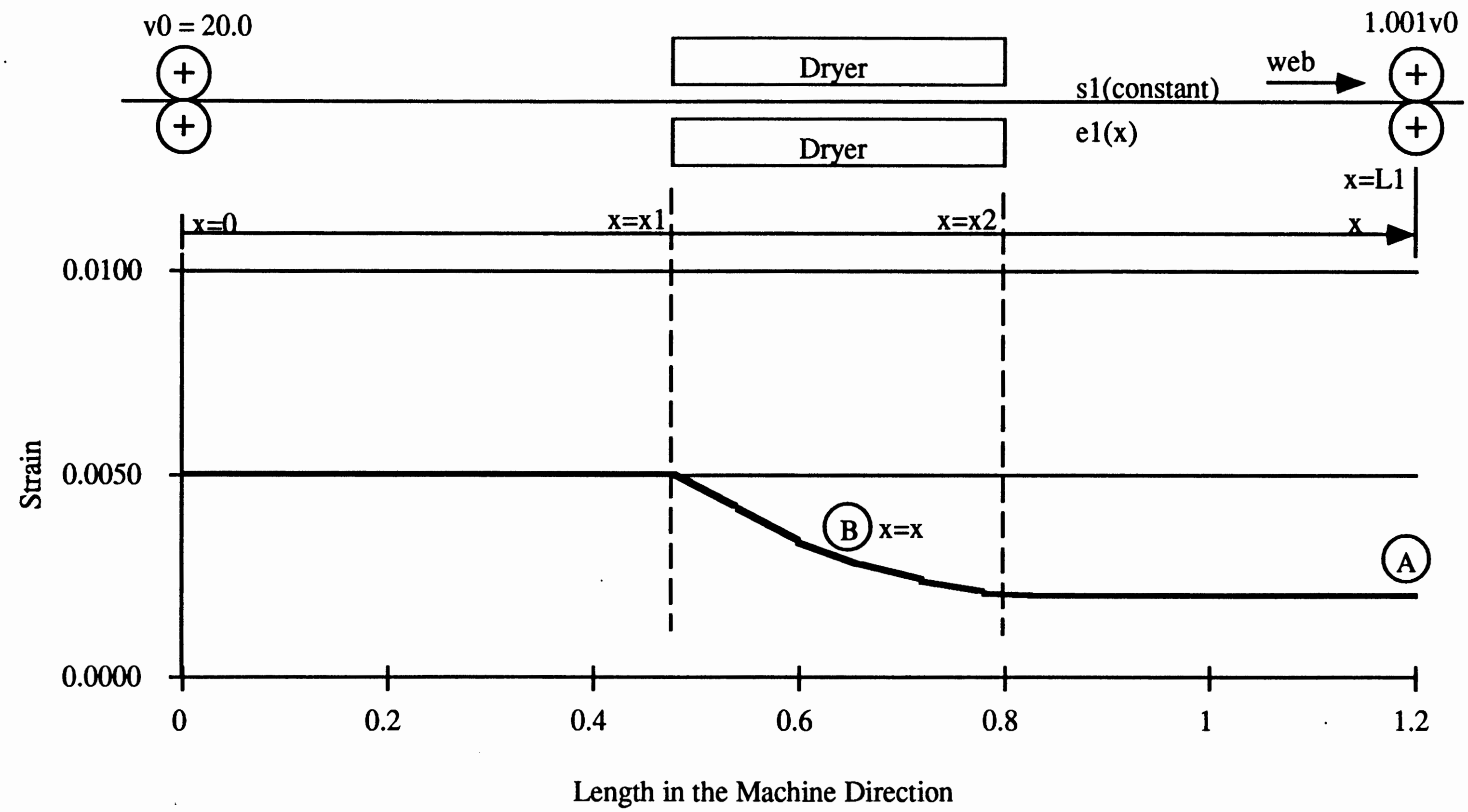

Figure 3.5. Strain Distribution in the Span With Changes in Property at Steady State

Chapter Summary

The elastic web model is convenient and acceptable for many practical applications. The main topics in this chapter are as follow.

(1) The basic behavior of the strain was discussed in both dynamic and steady state conditions. This propagation is the first order low pass filtering.

(2) As an extension of this discussion, two cases were examined. Both of them allow changes in the elastic modulus of the web. The first case is to simulate the disturbance outside the span which is inescapable in actual operation. The result supports this filtering nature.

(3) The second one is to simulate the steady state drying process where the elastic modulus changes inside the span. This shows the distributed strain in the span under uniform stress.

CHAPTER IV

THE VISCO-ELASTIC WEB MODEL AND TRANSPORTATION

Introduction

The elastic web model which was discussed in Chapter III is handy and practically useful. It is, however, necessary to expand the model so that it can handle the phenomena which elastic models cannot. Among these cases is a handling of wet paper or heated plastic film where the web is operated under relatively high strain, the order of 1%, causing plastic deformation.

Another reason to study this visco-elastic web is to investigate the behavior of strain itself in the span. Researchers have used an elastic model because it is close to the visco-elastic model. No discussion has been made on the behavior of the visco-elastic model in the web handling system. It is also important to confirm whether elastic model is a good approximation.

There are two approaches to the problem of visco-elasticity. The first one is a micro-scopic approach which tries to explain the phenomena from the actual elements of the material. In the case of paper, for example, fibers and their bonding determine how it deforms under a given external force. This may give more accurate understanding of the material and its behavior, but the equation is very complicated and it is hard to incorporate with the web handling.[8]

The second approach, a macro-scopic approach, observes the behavior of the material and gives the model that approximates this behavior. Models from this approach do not necessarily have physical meanings. For example, the spring element in the model may not correspond to any structures in actual material. It just represents how this material looks in an action. Still this is useful to explain the behavior of the material mainly because of its simple mathematical expression.

The visco-elastic model belongs to the second approach, and models shown in this chapter were originally presented in the 19th Century by Maxwell and Voigt. Improvement and addition have been made on these models, but they are still the basic models which express visco-elastic behavior.[7]

In this chapter, various models are shown first, followed by the discussion of the strain distribution and their simulation.[5][6][7]

Web Model

Figure 4.1 shows a schematic drawing of the single span which has two pairs of rollers at both ends and the web stretching between them. In this case, each small segment of web is represented by the visco-elastic model which includes springs and dampers. Details of this model are given in a later portion of this chapter. It should be noted that each element may act independently based on its history. Even in this case, super-positioning of strain is available, and each segment can be treated independently. From here, some pages are dedicated to the discussion of typical models which are shown in figures 4.2 through 4.8.

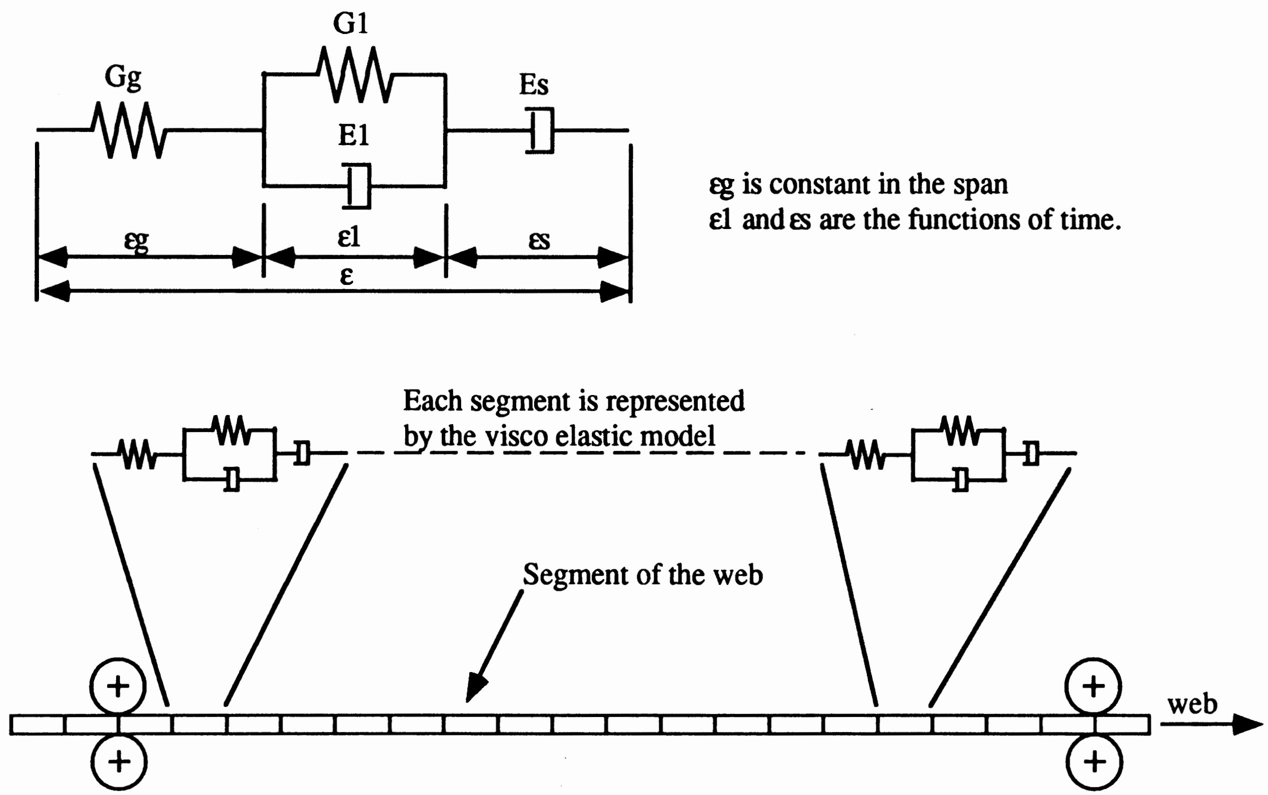

Figure 4.1. Visco Elastic Model of the Web

One Element Model

Figure 4.2 shows the one element model. This model is either a spring or a damper. The spring element alone is classified as an elastic model which was discussed before, but this is redrawn here preparation for explaining the other.
In the case of a spring, the relation between strain and stress is given as:

$$\varepsilon g = \frac{\sigma g}{Gg} \tag{4.1}$$

In case of a damper, this is given as:

$$\sigma s = Es \frac{d(\varepsilon s)}{dt} \tag{4.2}$$

The strain is given with the initial strain $\varepsilon s0$ as:

$$\varepsilon s = \varepsilon s0 + \frac{\sigma s}{Es} * t \tag{4.3}$$

Two Element Model

The two element model contains both a spring and a damper. The model in Figure 4.3 is called a Maxwell model that has one pair, a spring and a damper in series. That in Figure 4.4 is called a Voigt model with this pair in parallel. In the Maxwell model, total strain is a sum of the strain from the spring and the damper. This relation yields

$$\frac{d(\varepsilon)}{dt} = \frac{1}{Gg}\frac{d(\sigma)}{dt} + \frac{\sigma}{Es} \tag{4.4}$$

Also (4.1) and (4.2) give

$$\varepsilon = \varepsilon g + \varepsilon s = \sigma * \left(\frac{1}{Gg} + \frac{t}{Es}\right) + \varepsilon s0 \tag{4.5}$$

Equation (4.5) gives the relation between strain and stress if stress is given as step input.

In the Voigt model, total stress is a sum of that from the spring and that from the damper. This relation yields

$$\sigma = G1 * \varepsilon + E1 \frac{d(\varepsilon)}{dt} \tag{4.6}$$

It is necessary to give some conditions to calculate the strain ε from (4.6).

Let the conditions be as follow. (Step change of the stress)

$$\sigma = \sigma 0 = G1 * \varepsilon 0, \text{ and } \frac{d(\varepsilon)}{dt} = 0 \text{ for } t \leq 0$$

$$\sigma = \sigma 1 = G1 * \varepsilon + E1 \frac{d(\varepsilon)}{dt} \qquad \text{for } t > 0 \tag{4.7}$$

Now let $\hat{\sigma} = \sigma 1 - \sigma 0$, $\hat{\varepsilon} = \varepsilon - \varepsilon 0$, then equation (4.6) yields

$$\hat{\sigma} = G1 * \hat{\varepsilon} + E1 \frac{d(\hat{\varepsilon})}{dt} \tag{4.8}$$

Taking Laplace transform on both sides of (4.8) yields

$$\hat{\sigma} = (G1 + S\ E1) * \hat{\varepsilon} - E1 * \hat{\varepsilon}\big|_{t=0} = (G1 + S\ E1) * \hat{\varepsilon} \tag{4.9}$$

and step input of $\hat{\sigma}$ gives

$$\hat{\sigma} = \frac{\sigma 1 - \sigma 0}{S} \tag{4.10}$$

Equation (4.9) and (4.10) yields

$$\varepsilon = \varepsilon 0 + \frac{1}{G1} (\sigma 1 - \sigma 0)(1 - \exp(-\frac{G1}{E1} t)) \tag{4.11}$$

Using the relation of $\varepsilon 0 = \frac{\sigma 0}{G1}$, (4.11) can be written as

$$e = \frac{\sigma 1}{G1} (1 - \exp(-\frac{G1}{E1} t) + \frac{\sigma 0}{G1} \exp(-\frac{G1}{E1} t) \tag{4.12}$$

Equation (4.12) gives the relation between strain and stress if stress is given as step input.

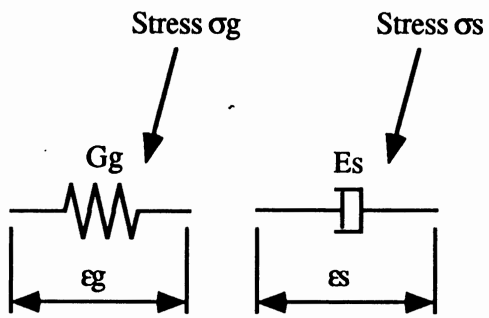

Figure 4.2. Spring and Damper

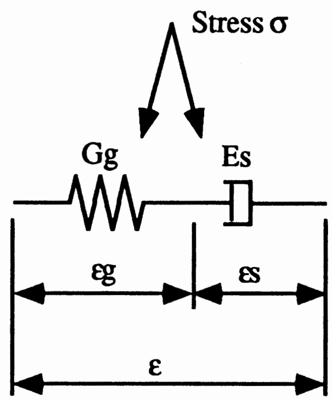

Figure 4.3. Maxwell Model

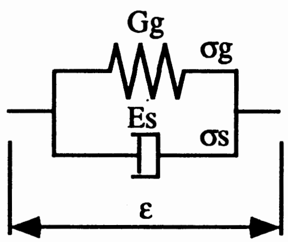

Figure 4.4. Voigt Model

Three Element Model

The three element model contains two springs and a damper either in series or in parallel. figures 4.5 and 4.6 show the models called a three elements Voigt model and a three elements Maxwell model respectively. These two models are interchangeable with each other, and it is up to the application which model is to be used.
Conversion is shown here for reference.[5]

$$Ggm = \frac{G1 * Gg}{G1 + Gg}, \quad G1m = \frac{Gg^2}{G1 + Gg}, \quad E1m = E1 * \left[\frac{G1}{G1 + Gg}\right]^2 \tag{4.13}$$

Basically, the Maxwell model is suitable for discussing stress under a given strain, and the Voigt model is for strain under a given stress. As was mentioned before, the assumption of uniform tension in the span is more agreeable than that of uniform strain. In addition to this, the basic equation of conservation of mass, equation (2.5), deals with strain. Then, it is more convenient to use the Voigt type model in this study. Thus, only the Voigt model will appear in this paper from this point.

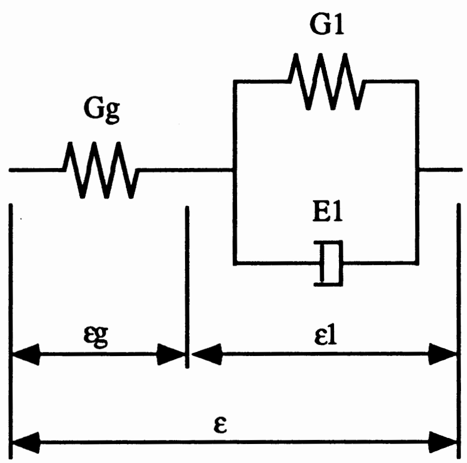

Figure 4.5. Three Elements Voigt Model

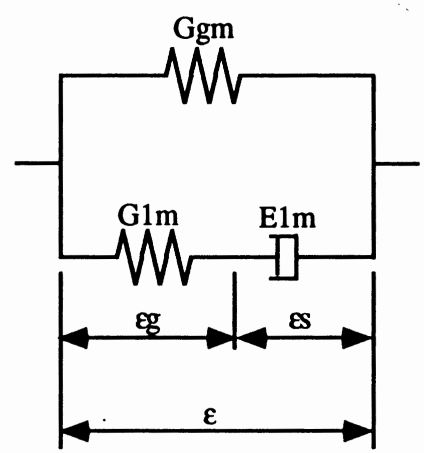

Figure 4.6. Three Elements Maxwell Model

Generalized Voigt Model

The Voigt model can be expanded to a generalized one which contains N-pairs of springs and dampers, each of which has a different coefficient, a spring element with coefficient Gg, and a damper element with coefficient Es. Figure 4.7 shows this picture. This can be interpreted as the combination of a Maxwell model and N-Voigt models. Thus, a similar relation of the strain is given through arguments shown before. The conditions are as follow.

$$\sigma = \sigma 0 = Gi * \varepsilon i0, \text{ and } \frac{d(\varepsilon i)}{dt} = 0 \ (i=1 \ldots N) \text{ for } t \leq 0$$

$$\sigma = \sigma 1 = Gi * \varepsilon + Ei \frac{d(\varepsilon i)}{dt} \qquad (i=1 \ldots N) \text{ for } t > 0 \tag{4.14}$$

Using the super-positioning of strain and equation (4.5) and (4.12);

$$\varepsilon = \varepsilon g + \varepsilon s + \sum_{i=1}^{N} \varepsilon i$$

$$= \sigma 1 * \left(\frac{1}{Gg} + \frac{t}{Es}\right) + \varepsilon s0 + \sum_{i=1}^{N} \left(\frac{\sigma 1}{Gi}\left(1 - \exp\left(-\frac{Gi}{Ei} t\right)\right) + \frac{\sigma 0}{Gi} \exp\left(-\frac{Gi}{Ei} t\right)\right)$$

$$= \sigma 1 * J(t) + I(t) \tag{4.15}$$

where
$$J(t) = \frac{1}{Gg} + \frac{t}{Es} + \sum_{i=1}^{N} \frac{1}{Gi}\left(1 - \exp\left(-\frac{Gi}{Ei} t\right)\right) \tag{4.16}$$

$$I(t) = \varepsilon s0 + \sum_{i=1}^{N} \frac{\sigma 0}{Gi} \exp\left(-\frac{Gi}{Ei} t\right) \tag{4.17}$$

Equations (4.15), (4.16), and (4.17) express the behavior of strain for the step change of stress (from $\sigma 0$ to $\sigma 1$).

The simplest model has four elements and this is shown in Figure 4.8. This is given by letting N = 1 in the generalized model and the equations are also available in this manner. The name "elastic part," "Voigt part," and "plastic part" are used for my convenience in this paper. The simulation which appears later in this chapter uses this four element model.

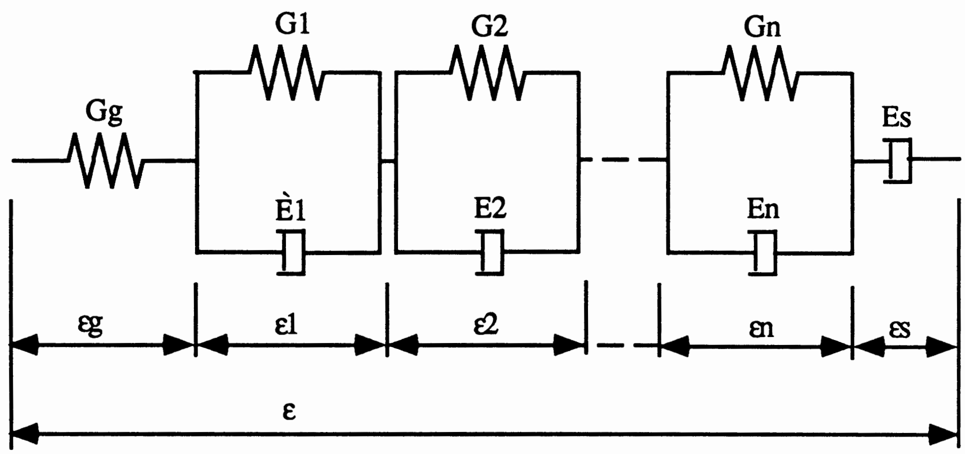

Figure 4.7. Generalized Voigt Model

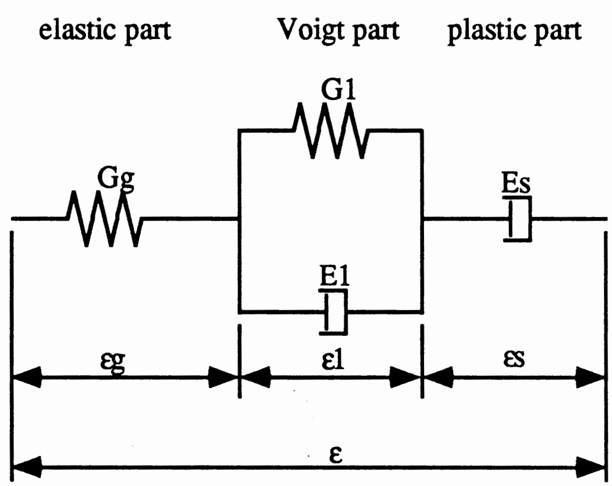

Figure 4.8. Four Element Voigt Model

Theoretical Analysis

As was mentioned in Chapter III, strain distribution inside the span is the main concern here. It is necessary, however, to add some assumptions to get the results because strain is not uniform in the span but it is a function of time. Two cases are discussed here. The first one is for a non slip condition, and the second one deals with slippage. Both of them are steady state analyses.

Case 1: Spans Without Slippage Between Rollers and Web

First, some assumptions are necessary. They are:

(1) Assumptions made in Chapter II.

(2) Uniform distribution of the stress inside the span

(3) Uniform mass flow rate which is given as $\frac{v}{1+\varepsilon}$

This one is important because changes of strain must cause changes in the web speed and there is no relation for this. This assumption is agreeable if there is no migration inside the material of the web.

With these assumptions and the equations developed before, it is possible to determine the strain distribution inside the span in a steady state. The procedure is as follows.

(1) Calculate the total strain at the end point of the span using equation (2.8) at its steady state. ie. e1 = a1 * (1+e0) - 1.

(2) Assume the traveling time between roller 0 and roller 1. Let this be T1.

(3) Calculate the function J(t) and I(t) which are defined in equations (4.16) and (4.17). Function J(t) gives the compliance of the web, and function I(t) gives the free response of initial conditions of strain. Both are time variants.

(4) Using J(t) and I(t) at roller 1 yields the stress of the web which is constant through out the span. This is $\sigma 1$.

(5) Now it is possible to calculate the stress at each time using equation (4.15), and this gives the distribution of the strain and the web speed through the relation of mass flow rate $\frac{v}{1+\varepsilon}$ = constant.

(6) Calculate the time which the web needs to travel from roller 0 to 1 using these data and correct the value T1 which was assumed at step (2).

(7) Repeat until both ends meet.

The initial conditions of strain propagate one span to the next. The elastic part of the model has no initial condition.

Case 2: Spans With Slippage Between Rollers and Web

The same assumptions are necessary as are used in Case 1. In this case, the actual speed of the web is not known due to the slippage on the roller. This is the same approach which was given in (2.7). A convenient way to deal with this problem is to use two spans. The first span begins with a non slip roller and ends with a slipping roller and the second one ends with a non slip roller. Then the problem is reduced to two parts: first to determine the actual speed of the web on the slipping roller and second, to calculate the strain using the method shown in Case 1.

It should be noted that the stress is given only from the elastic part of the model even though it has other viscous parts in it. N-Voigt parts and the damper Es in Figure 4.7 do not contribute to the stress though they support the stress internally between the damper and the spring.

Simulation

Two programs were developed to calculate the distribution of the strain in the span. Both use the four element Voigt model which is depicted in Figure 4.8. Values of Gg, G1, E1, and Es and initial values of dampers are fictional except in the last part of this section. Actual values measured for the news print are used in that part to see how the results differ from the approximated results with the elastic model.

In every picture, the web moves from roller 0 to 2 through roller 1.
Every result shows the strain with respect to the location of the span which corresponds to the picture on the top. Each mark shows the share of strain in the total value, except the solid bold line which indicates the result from the elastic model that only has two springs, Gg and G1. This line is the basis of conservation of mass; thus plots of this and the total strain of the visco-elastic model have the same values at each end.

Case 1: Spans Without Slippage Between Rollers and Web

Figure 4.9 shows the results of the simulation where there is no slippage on the rollers. It is observed that the elastic part has a sudden change of strain before and after roller 1 while the Voigt part and plastic part cannot act quickly due to their damper element. These delaying parts generate the distribution of strain in the span.

Figure 4.10 shows the same system except that the web is 10 times as fast as that in Figure 4.9. As is shown in equations (4.16) and (4.17), the time constant of this model is given by the value of Gi and Ei in the Voigt part, both of which are specified for a given material. Then the different travelling time causes different aspects.

It is interesting that the strain of the elastic part is also affected by the difference of the viscous part. This means that the stress or web tension varies according to the operational speed in the web handling. Actually, elastic strain changes from 0.0047 to 0.0065 in span 1, and this occurred without any disturbance in the system nor changes in the properties of the web. Elastic model indicated by the bold line in the figure stays the same and this cannot explain this change. Program 1 in Appendix B was used for this simulation.

Case 2: Spans With Slippage Between Rollers and Web

Figure 4.11 shows the results where roller 1 slips with the web. Program 5 in Appendix B calculated the slippage ratio so that this ratio satisfies the conservation of mass, visco-elastic changes, and the relation of stress simultaneously. After this, program 1 calculated the distribution with this slippage ratio. At steady state, there are no significant changes in concept between Case 1 and 2.

Case 3: Spans Without Slippage Between Rollers and Web. (Actual Web)

Figure 4.12 shows the results from using the actual data of printing paper.[1] This is data for a Maxwell model; thus the Voigt part does not contribute to the strain. Although the initial strain of the plastic part is fictional, this may not be apart from the point and the total strain gives the correct data on strain distribution. Authors in the report [1] concluded that the viscous part did not affect the total stain from the stage of modeling, and this is valid directly from this result. Likewise it is possible to examine another case by comparing results directly.

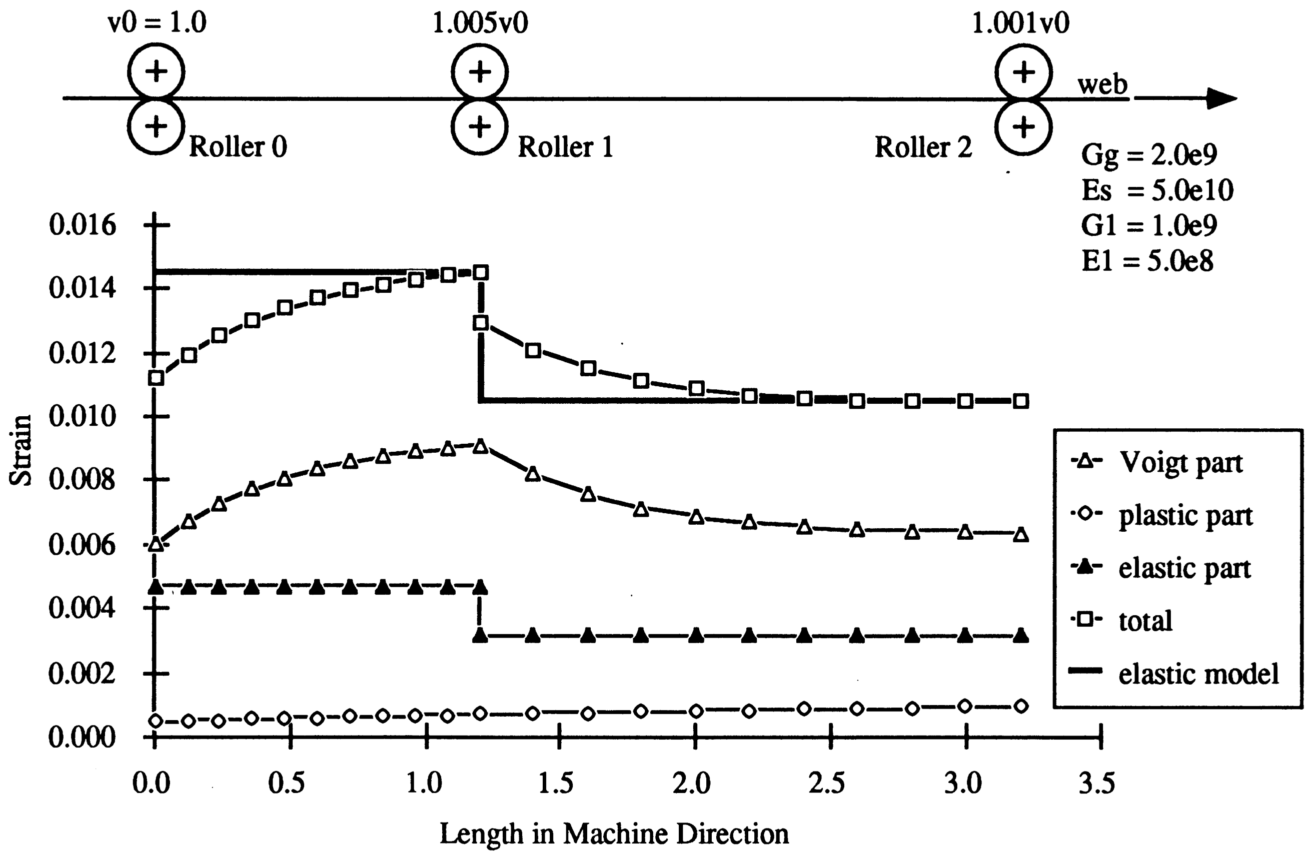

Figure 4.9. Strain in Span 1 and 2 at Steady State (v0=1, Non Slip)

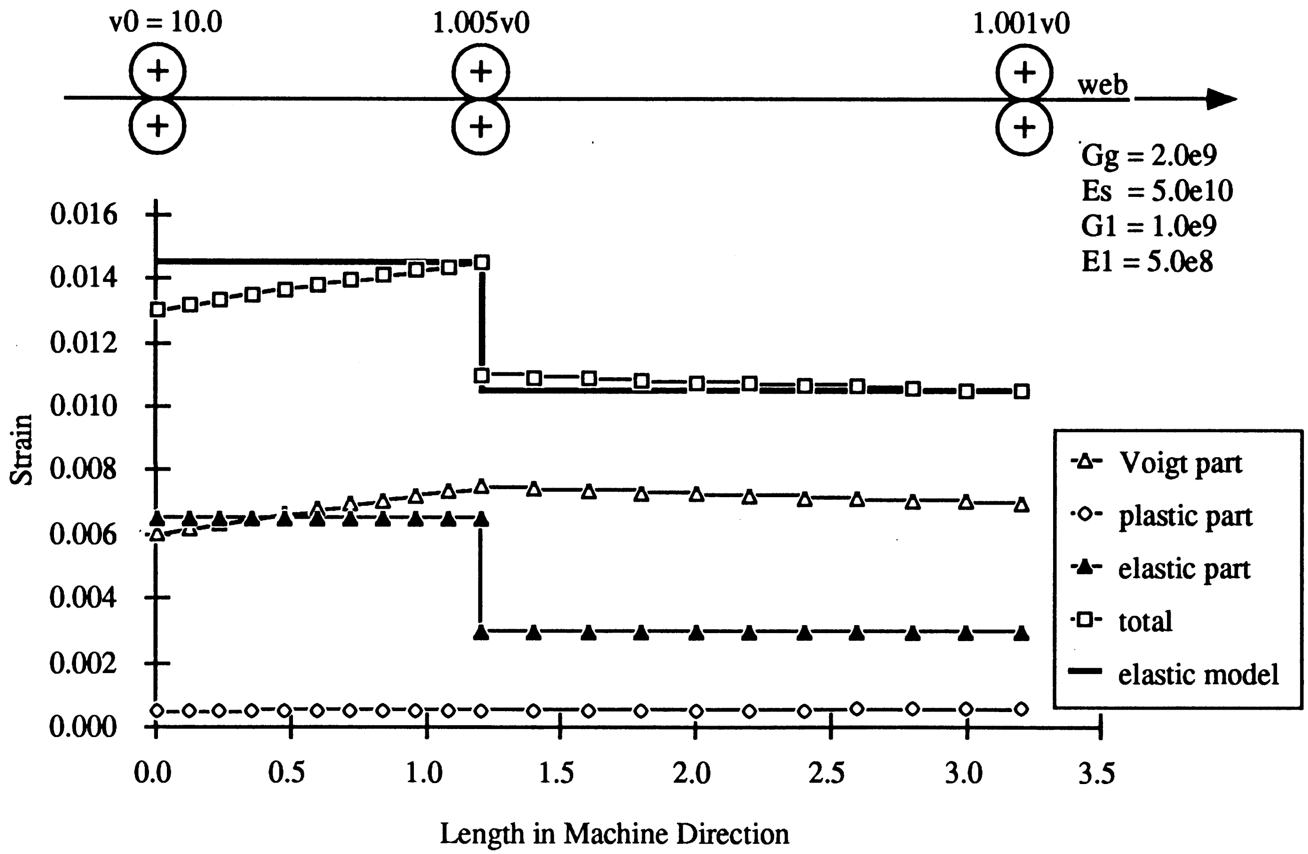

Figure 4.10. Strain in Span 1 and 2 at Steady State (v0=10, Non Slip)

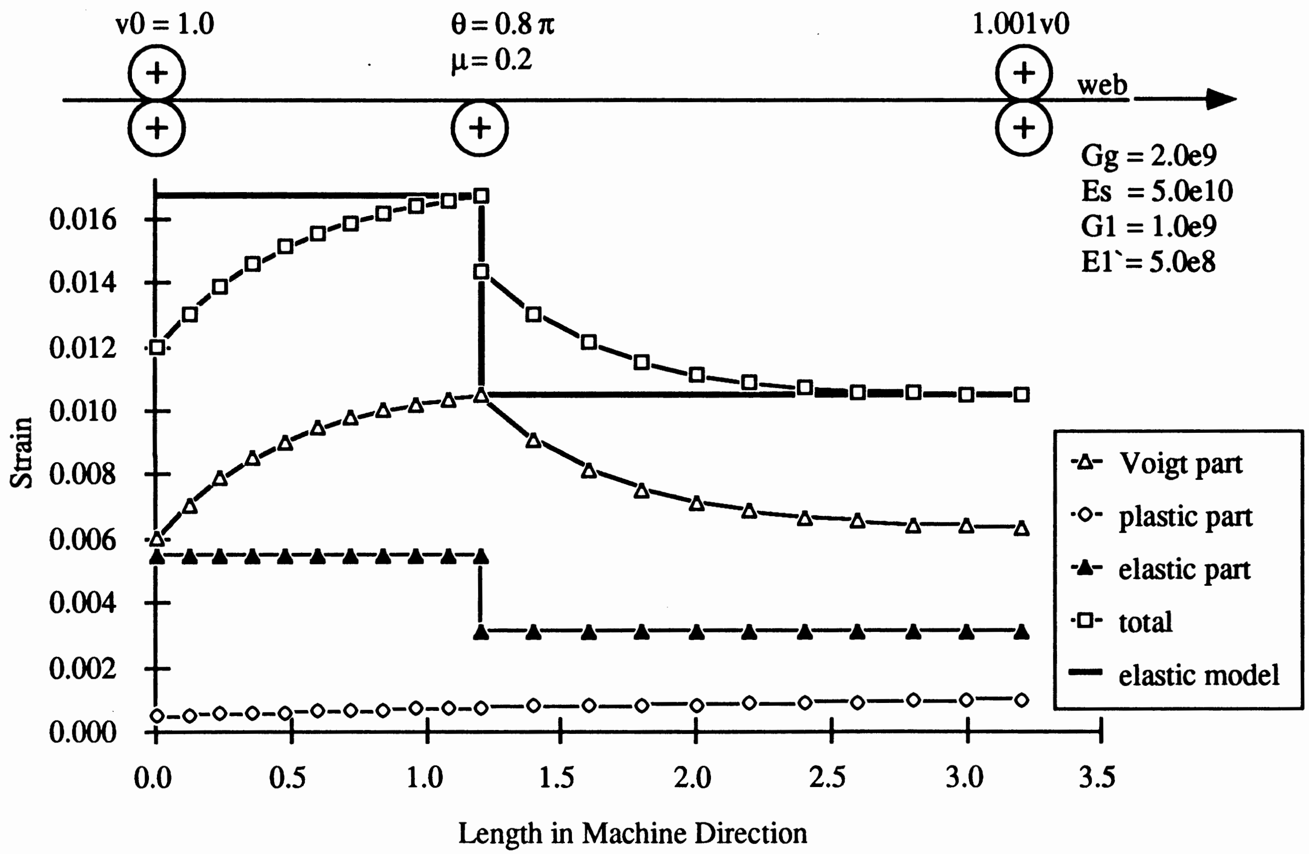

Figure 4.11. Strain in Span 1 and 2 at Steady State (v0=1, Slip)

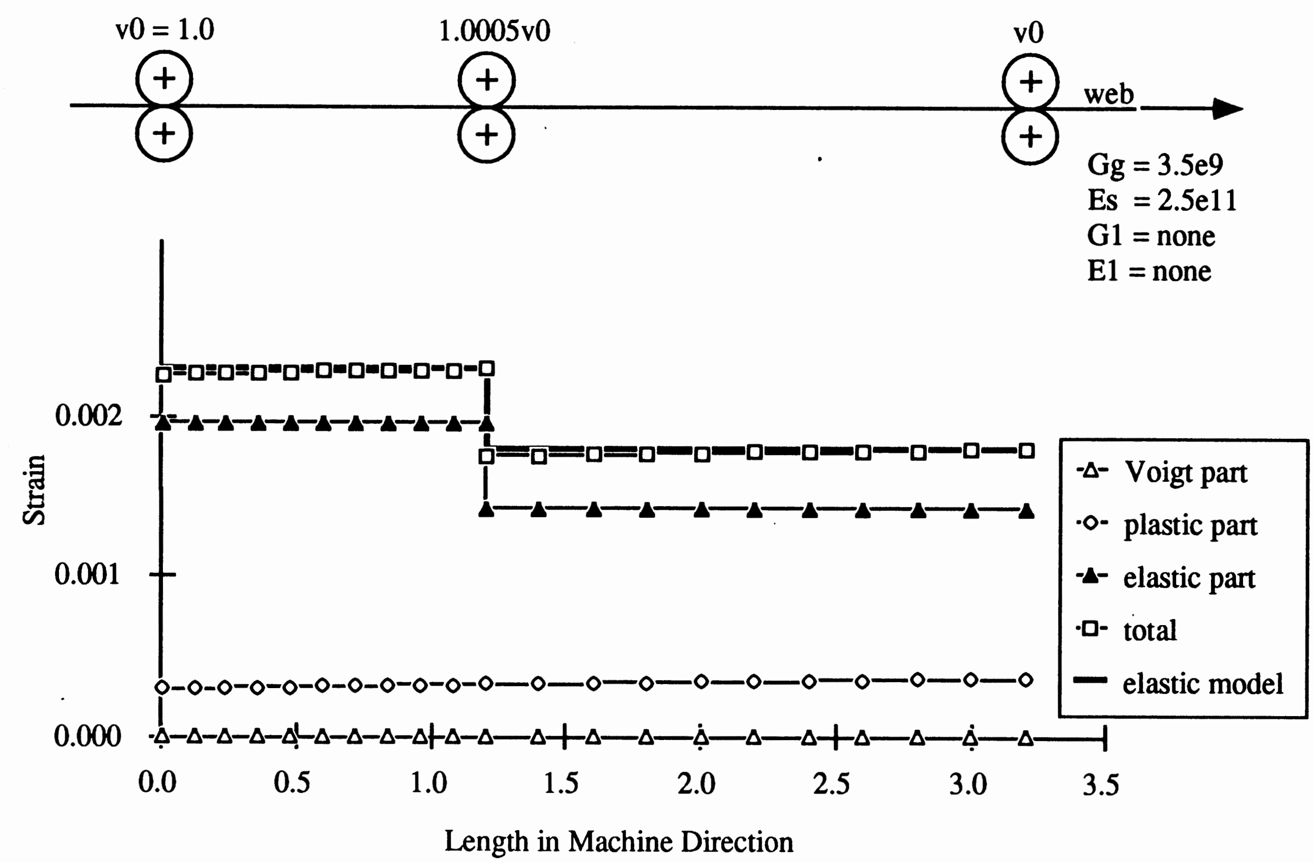

Figure 4.12. Strain in Span 1 and 2 at Steady State (v0=1, Non Slip, Paper)

Chapter Summary

Visco-elastic model and strain distribution were discussed in order to extend the theory of web handling. The main topics are as follow.

(1) Various visco-elastic models are discussed. Among them is the Maxwell model which is suitable for the discussion of stress under a given strain, and the Voigt model which is proper for the discussion of strain under a given stress. The Voigt model is convenient to discuss problems in web handling.

(2) Strain responses for a given stress input (step) were examined. This was done for models from the simplest one to the complicated one which has spring, damper, and N-Voigt elements in it.

(3) Theoretical analysis was made to incorporate the models to the handling equation which was shown in the previous chapter. It is necessary to use an additional assumption of a uniform mass flow rate. The analysis gives the strain distribution inside the span for both slipping and non slipping cases.

(4) Simulations and their results are shown to demonstrate the result of analysis. It is found that visco-elastic web changes its stress in the span according to the operating speed. This is what an elastic model cannot predict.

(5) Data from an actual web was used for verifying the analysis done with the elastic model. The result indicates that no significant difference exists; thus the elastic model is good for this application. Now it is possible to examine another case by comparing results directly.

CHAPTER V

STICK-SLIP PHENOMENON IN THE TRANSPORTATION

Introduction

The previous chapters deal with a stable operation. This means that there are two cases in the operation, either with or without slippage on the roller, but these two can be separated from each other. Under this condition, it is possible to calculate the strain or stress in the span with the proper web model and equations (2.7), (2.8), and (2.9).

There is, however, another case which allows the existence of both in the same operation. In this case, each one should be treated with corresponding relations, but there is a chance this interchanging will cause an unstable operation if they interact each other.

This chapter discusses this instability in the operation with a simple friction model which has non linearity in its value. The elastic web model without property changes is used to simplify the discussion.

Friction Model

Figure 5.1 shows the friction model with the actual data. (Also in Figure 2.3) This is a simplified non-linear model which has two values of friction coefficients according to its condition, whether the web slips or not on the roller. This is based on the study for the friction of paper.[9] Practically speaking, acceleration of the web may not affect the phenomena because the mass of web on the roller is negligible; thus the negative damping part can be substituted for vertical lines which have a threshold level.

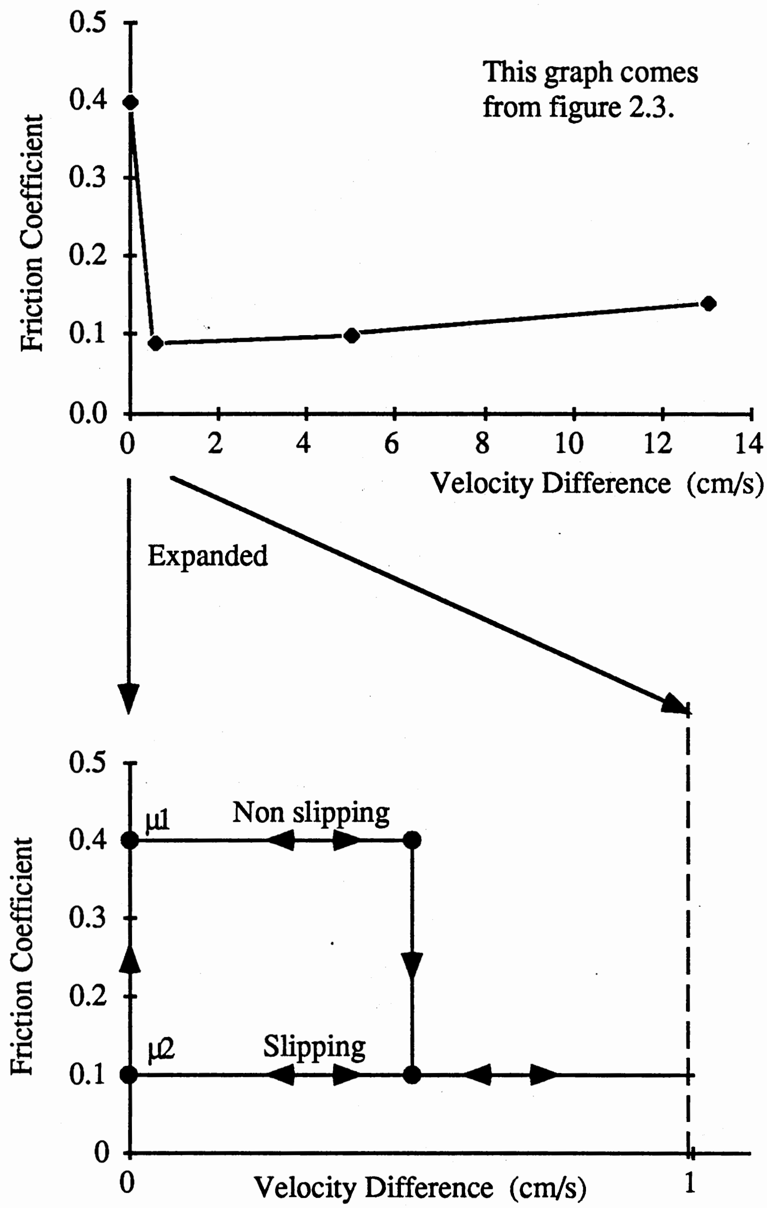

Figure 5.1. Non linearity in Friction Coefficient

Behavior of Stress With This Friction Model

Figure 5.2 shows the diagram of a pair of stresses s1 and s2 in many cases.

(1) Non slip condition (Point B <-> C). Stress s1 is given by equation (2.8) which expresses the conservation of mass. The solid line from point B to C shows this relationship. Notice that the value of s1 is independent from that of s2.

$$L1 * \frac{d}{dt}\left[\frac{1}{1+e1}\right] = \left.\frac{v0}{1+e0}\right|_{Time = t} - \left.\frac{a1*v0}{1+e1}\right|_{Time = t} \tag{2.8}$$

(2) Slip condition (Point O <-> A <-> B). Stress s1 is given from the equation (2.7). The solid line from point O to B shows this relation. Stress s2 and s1 have a linear relationship in this case.

$$s1 = s2 * e^{\mu_2 \theta} \tag{2.7}$$

(3) Transition from slip to non slip (Point A <-> B -> C). If s2 becomes big enough to tighten the roller, slippage stops and it turns to the non slip condition. Notice that there is no way back from Point C to A through B because the friction model has a threshold level and direction for this transition.

(4) Non slip condition (Point D <-> B). This is maintained non slip with a higher friction coefficient. Equation is (2.8).

(5) Transition from non slip to slip (Point D -> A). According to the friction model, this transition causes the sudden change in the friction coefficient from $\mu 1$ to $\mu 2$. Thus, point D moves suddenly to point A. Notice that this change is irreversible.

This partly one way diagram generates the stick-slip phenomenon. The reason is that the stress of the web, traction of the roller, and the condition (slipping or not) are related to each other. If one of them changes, it affects the rest of them. Then this causes another change in the first one accordingly. Suppose there is no slip condition initially; the typical scenario to stick slip is as follows.

(1) The speed of roller increases due to some disturbance.

(2) This increases the stress s1 and decreases s2, as was shown in Figure 3.3

(3) Decreased s2 reduces the traction of the roller. This corresponds to the movement from point C toward D in Figure 5.2.

(4) Once it hit point D, which is also changing due to the increased speed, the web slips suddenly. This is movement from point D to A.

(5) This change carries a certain amount of mass of the web from span 2 to span 1, which decreases stress s1 and increases s2.

(6) The roller attempts to withdraw the mass from span 1 again.

(7) Return to (2).

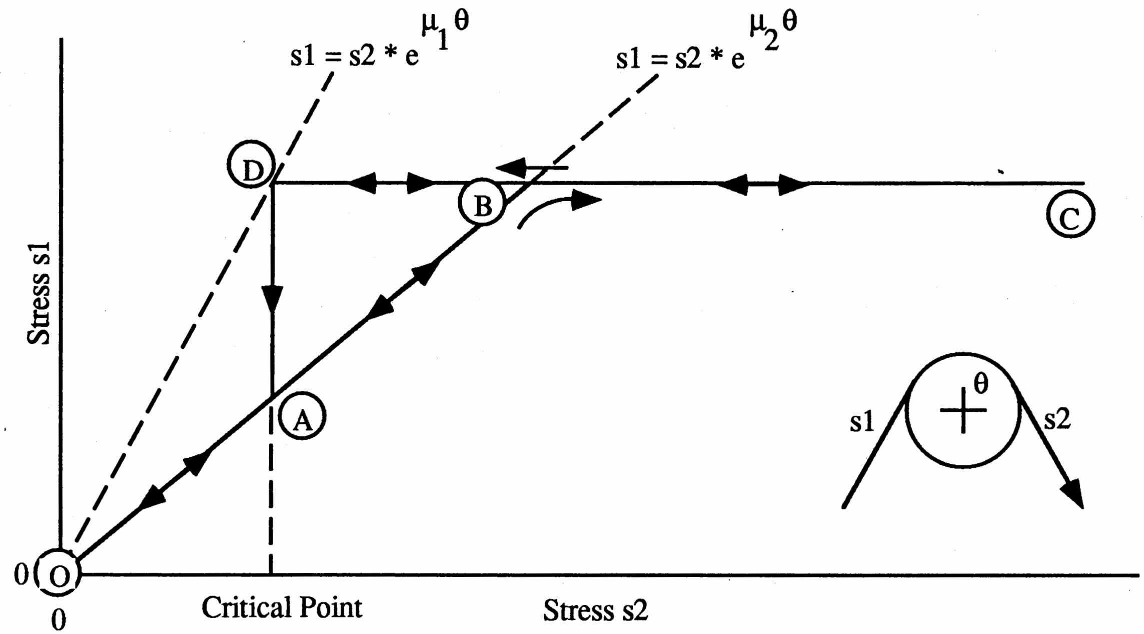

Figure 5.2. Criteria of Slip and Non Slip

Simulation

It is necessary to develop equations which treat the relation with a slipping roller. By using the system shown in Figure 2.2, equations (2.10) and(2.11) yield the equation for span 1 and 2 combined. This is given as

$$\frac{d}{dt}[\,L2*(1\text{-}e2)+L1*(1\text{-}e1)\,]=v0\left[\frac{1}{1+e0}-\frac{a2}{1+e2}\right] \tag{5.1}$$

The relation of stress is given from (2.7) as

$$s1 = s2 * e^{\mu\theta} \tag{2.7}$$

With the elastic model, this is valid also for strain. This gives the relation as

$$e1 = e2 * \ e^{\mu\theta} \tag{5.2}$$

(5.1) and (5.2) yield the final equation

$$\frac{de2}{dt}=\frac{v0\left[\frac{-1}{1+e0}+\frac{a2}{1+e2}\right]}{L2+L1\ {*}e^{\mu\theta}} \tag{5.3}$$

Program 6 in Appendix B was used to simulate the behavior. In this program, two sets of equations are used to calculate the strain in non slip and slip conditions, and algorithm judges which is occurring using the relationship shown in Figure 5.2. Once the strain comes to the critical point, shown as the transition point A and D, sudden change occurs in the friction coefficient. Then the mass in span 2 backs to span 1 so that it returns to the equilibrium point under the slip condition.

In the simulation, a system with two spans is used like in the previous chapter. The picture is shown in figures 5.3 and 5.4. In this system, both ends are in a non slip condition while the roller in the middle may or may not slip. Other conditions of this simulation are as follow.

(1)	Speed at roller 0; v0	= 10.0
(2)	Strain before roller 0; e0	= 0.001
(3)	Speed at roller 1; v1	= 1.002 * v0 (time < 0.05)
		= 1.003 * v0 (afterward to steady state)
(4)	Speed at roller 2; v2	= 1.001 * v0
(5)	Length of span 1; L1	= 1.2
(6)	Length of span 2; L2	= 2.0
(7)	Friction coefficient $\mu 1$	= 0.4
(8)	Friction coefficient $\mu 2$	= 0.15
(9)	Lapping angle θ	$= 2\pi / 3$ (for 5.3), $\pi / 2$ (for 5.4)

Step change of roller speed were added to the system as a disturbance. Two similar cases in figures 5.3 and 5.4 shows entirely different pictures of this phenomenon.

Figure 5.4 shows a similar result to that shown in Figure 3.3. The increased speed of roller 1 changed e1 permanently and e2 temporarily. Strain e2 does not hit the critical strain ecr in this case. Thus there is no stick-slip phenomenon.

Figure 5.4 shows a different behavior though only the lapping angle is different. In this case, there are sudden changes in strain when e2 hits ecr and this instability continues unless there is another disturbance or energy loss. If a1, the speed ratio of roller 1, has an initial value close to this one, it will end up with this stable limit cycle after it suffers a disturbance.

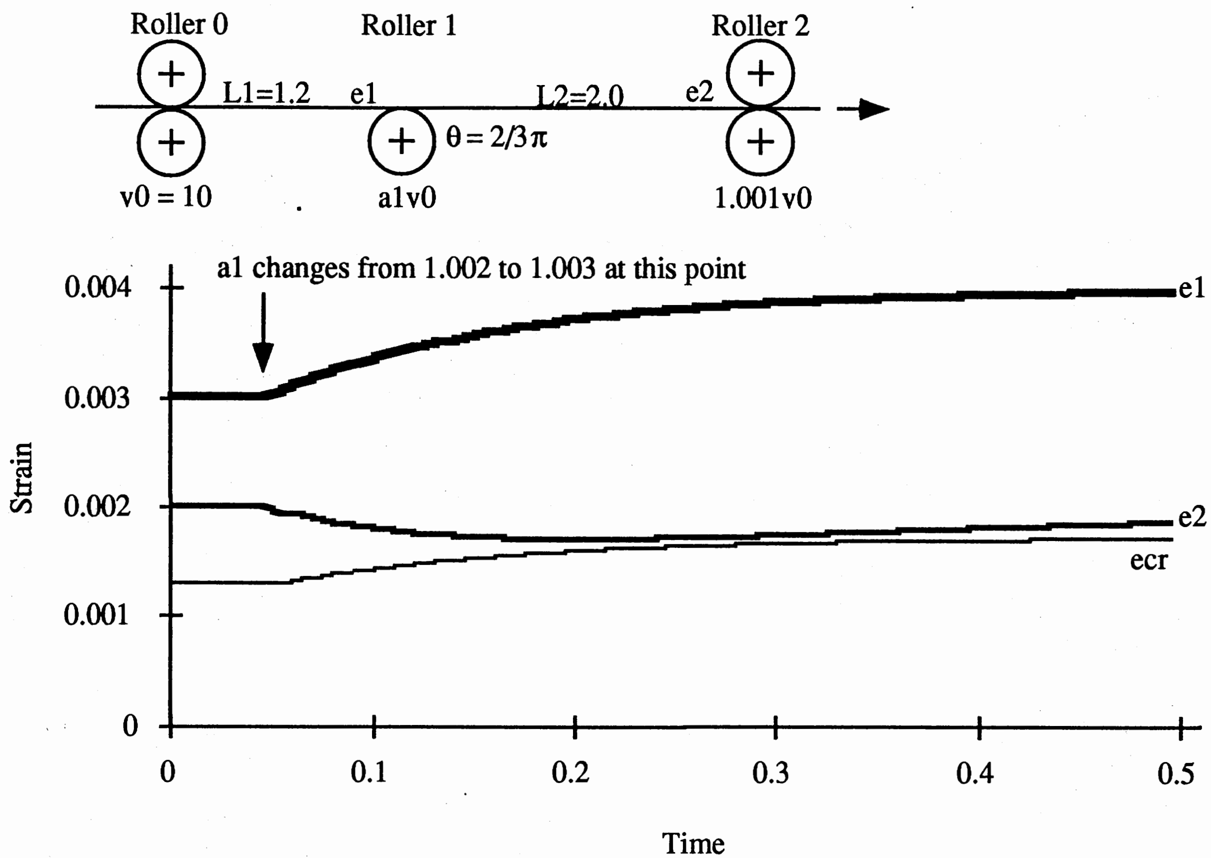

Figure 5.3. Strain in Span 1 and 2 without Stick Slip

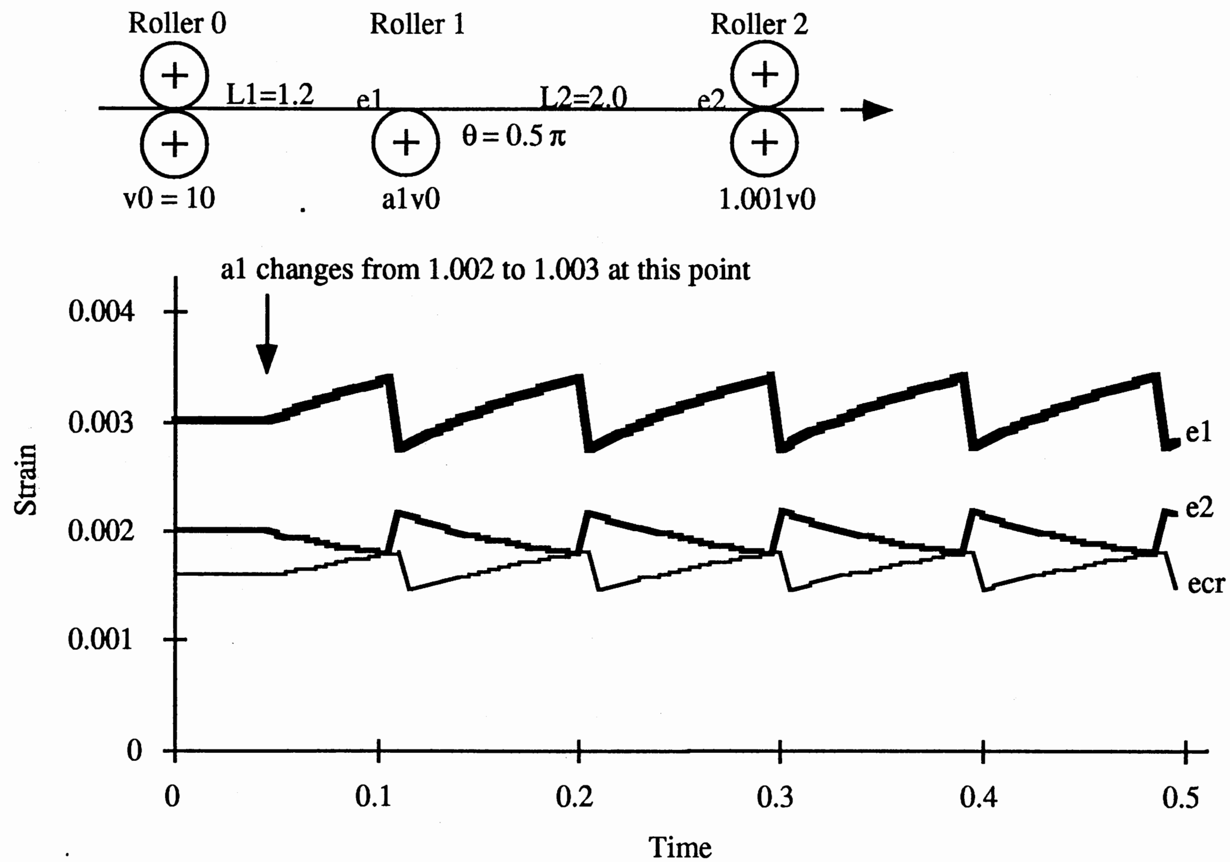

Figure 5.4. Strain in Span 1 and 2 with Stick Slip

Chapter Summary

The stick- slip phenomenon causes unstable strain and stress problems in the web handling. An analysis with simple non-linearity in the friction was made in this chapter. The main topics in this chapter are as follow.

(1) Non-linear friction models are presented which have two levels of friction coefficients according to the condition of slippage.

(2) The behavior of stain was examined in the handling system with two spans. Strain changes only one direction when it moves from one condition to another. This relation also gives the criteria to determine the condition of slippage in the simulation.

(3) Propagation of strain in the two span model was shown. This corresponds to equations shown in Chapter III.

(4) Simulation demonstrated the theory presented. Even similar configurations of the system with the same web may cause the significant differences if this stick-slip phenomenon may happen. Once it happens, it ends up with a stable limit cycle which does not disappear by itself.

CHAPTER VI

SLACKNESS IN THE TRANSPORTATION

Introduction

It is necessary to keep the web tension to a certain level through out the operation. Otherwise, it causes a lateral motion or breaks in the web in the worse case. To prevent this damage in the operation, slackness in the web and its causes should be studied.

As an application of the theoretical analysis presented in the previous chapters, this chapter shows some possible behaviors of the web model which cause slackness. They are for the elastic model (Chapter III), the visco-elastic model (Chapter IV), and the elastic model with stick-slip phenomenon on the roller (Chapter V). Programs developed in each chapter were used for the simulation. Conditions of the simulation are shown in the figures; otherwise they are unchanged from those in the original chapter.

Case 1: Elastic Model

Even with this simple model, slackness may happen.

Steady State

Equation (2.8) in its steady state yields as follow

$$e1 = a1 * (1+e0) - 1 \tag{6.1}$$

And $e1 \le 0$ if $a1 * (1+e0) \le 1$. It is possible to choose a1 that satisfies this relation. Although this is theoretically possible, this operation always causes slackness; thus it is no longer practical.

Dynamic Operation

As was shown in Figure 3.3, stain e2 decreases temporarily when increasing the speed of the center roller in a two span model. If this is too radical, strain e2 reaches a value low enough to cause the slackness. Figure 6.2 shows this case where e2 reaches nearly zero temporarily, but this is crucial in the actual operation because it would lead to a web break.

Program 6 in Appendix B was used for this calculation. This is an extracted one from Program 2 which was used for the simulation of the stick-slip phenomenon.

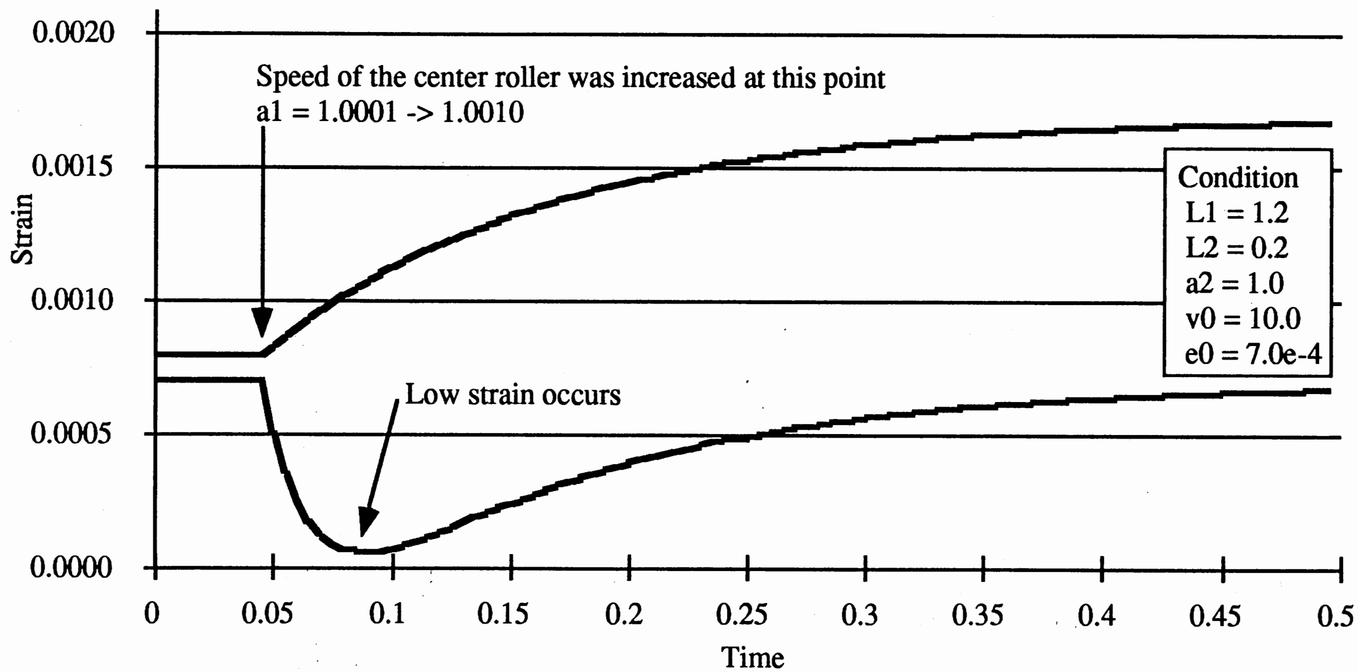

Figure 6.1. Strain in Span 1 and 2

Case 2: Visco-Elastic Model

The visco-elastic model has a more dangerous tendency because of the damper element which shares the strain but does not contribute to the stress. This means that even if the total strain determined from the conservation of mass is as high as that of elastic model, the visco-elastic model gives smaller stress which comes only from the elastic part.

Figure 6.2 indicates this problem. Total strain is equivalent to that of the elastic model, but this model gives nearly zero stress due to the strain in both Voigt and plastic parts.

Consequently, if the web has visco-elasticity, the speed of the roller should be increased to maintain the stress on the elastic part.

Case 3: Stick-Slip Phenomenon

As was discussed in Chapter V, stick-slip causes unstable behavior in the strain even if the web can be treated as a simple elastic model. Figure 6.3 shows an example of a case where the stress e2 is higher than e1; thus slippage happens in reverse. In this case the relation between s1 and s2 is given as $s1 = s2 * e^{-\mu\theta}$ (6.2)
Program 6 used in Chapter V deals with this case also.

It should be noted that the level of strain is still higher than that calculated from the system without stick-slip; this repeated low strain may cause a different problem.

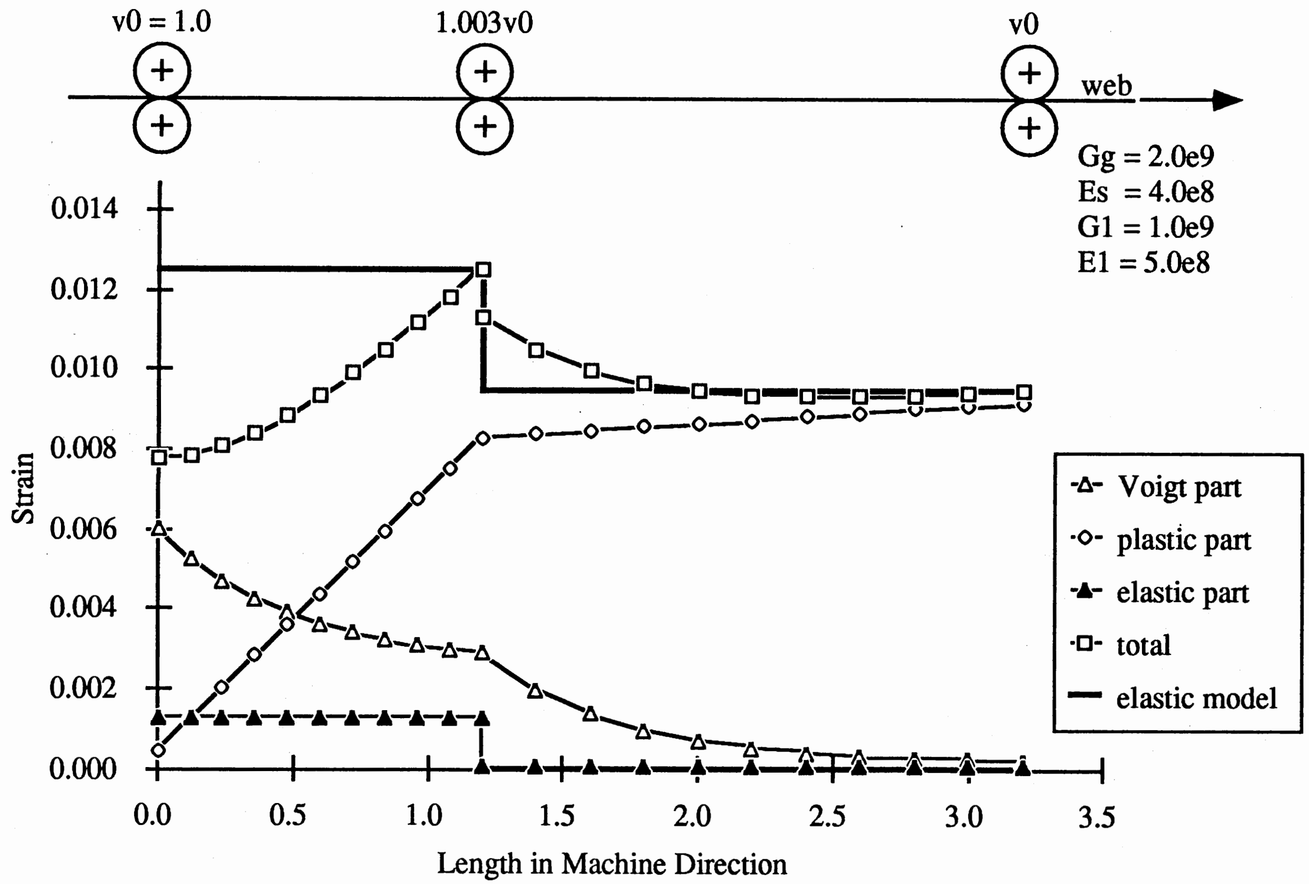

Figure 6.2. Slackness due to Visco-Elasticity

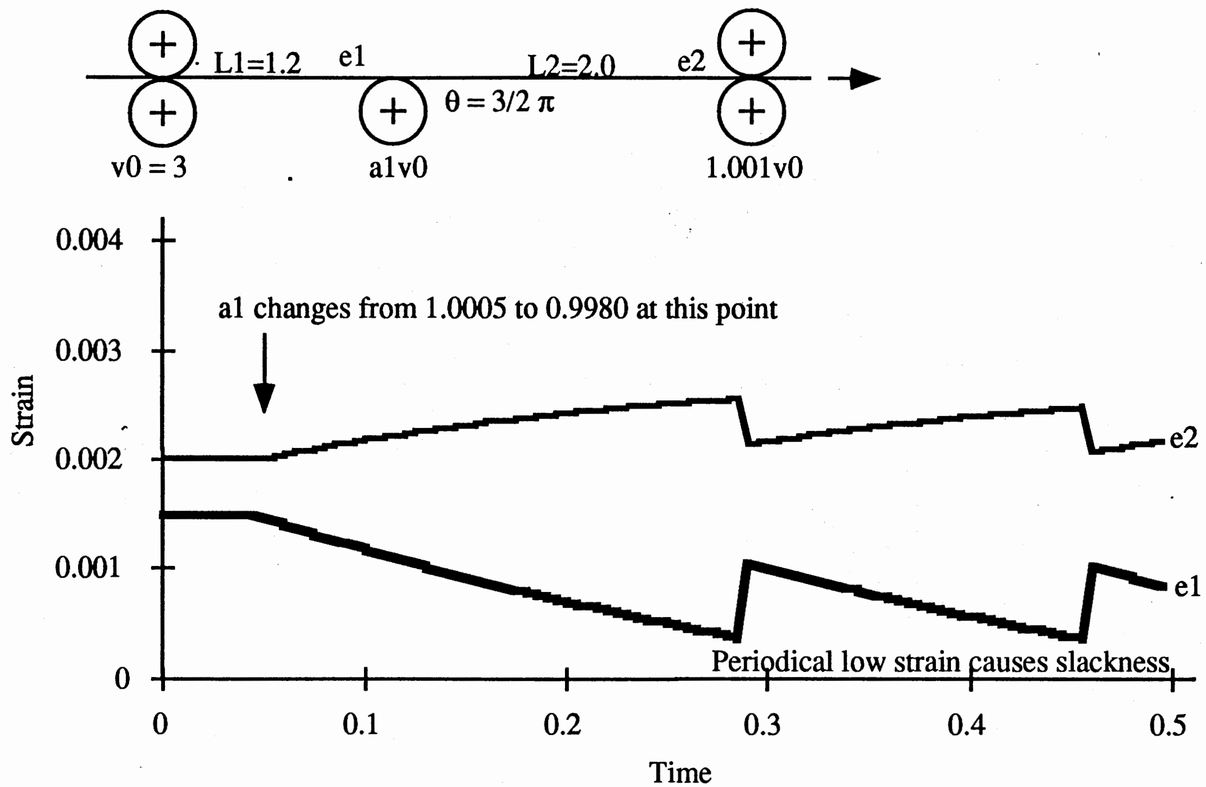

Figure 6.3. Strain in Span 1 and 2 with Stick Slip

Chapter Summary

Possible causes of web slackness were discussed.

The main topics in this chapter are as follow.

(1) The elastic model causes slackness if change in the span is radical. This is a temporary low strain, but it can be a crucial one in actual operation.

(2) The visco-elastic model may cause slackness more easily. This comes from the damper element which shares the strain but does not contribute to the stress. A handling system with this kind of web needs to have the speed increased along the process line.

(3) If stick-slip happens, it may cause slackness. Although this level of strain is higher than that that was calculated in Case 1, this is troublesome because of its repetitive nature while slackness in Case 1 is temporary.

CHAPTER VII

UNCERTAINTY AND STATISTIC DISCUSSION

Introduction

The previous chapters discussed assuming that parameters of models and/or disturbances are deterministic. In these cases, it is possible to calculate the propagation of strain using equations such as (2.8). In Chapter III, for example, strain in span 1 was simulated according to the disturbance of the strain in span 0.

In most cases, however, parameters and disturbances have some deviation or uncertainty in their values. An actual web is made from materials which are not perfectly uniform, and its properties may be affected through transportation and storage. For example, elastic modulus of paper is sensitive to its moisture content which varies from one place to another, from one season to another. Practically speaking, it is impossible to trace the output, as was done in the simulation in Chapter III, for all possible cases of input. Consequently, a statistical approach is necessary to handle this uncertainty.

If focused on the propagation of strain, this method has two major parts.
The first is to get the information about input. Some of them are mean and standard deviation of the properties of web and disturbance from the machine such as vibration of the rollers.
The second part is to discuss how the system propagates this uncertainty or irregularity from one span to another. This is important especially in machine design, because the tolerance depends on the ranges of possible output. In this paper, this second part is

discussed.

We can get the specific output without any tolerance by assuming a deterministic nature, while the ranges of possible occurrence are available if we allow uncertainty. These two approaches are both important in observing things and actually designing machines.

In this chapter, the general discussion of propagation is presented first, followed by its application to web handling and verification with a Monte Carlo simulation.

Propagation of Mean and Mean Square

Mean and mean square are important values for a most of statistical treatments. They give a basic understanding of the amplitude of the data series and determine the entire distribution if it is Gaussian. As a review of these statistics, propagation of mean and mean square are discussed here. Almost all equations in this general discussion are from Newland's book.[11] Though some have been defined or used differently in this paper, names of variables are unchanged as used in Newland's book in order to keep consistency. (θ, for example)

Propagation of Mean

Response y(t) can be described with input x(t) and the impulse response function of the system h(t) as

$$y(t) = \int_{-\infty}^{t} h(t-\tau)x(\tau)d\tau \tag{7.1}$$

Changing the variable converts (7.1) to

$$y(t) = \int_{0}^{\infty} h(\theta)x(t-\theta)d\theta \qquad \text{where} \qquad \theta = t - \tau$$

Since there is no response for $\theta < 0$ (or $t < \tau$), this can be expanded as

$$y(t) = \int_{-\infty}^{\infty} h(\theta)x(t-\theta)d\theta \tag{7.2}$$

By assuming a stationary process, expectation of both sides in (7.2) is given as

$$E[y] = E[x]\int_{-\infty}^{\infty} h(\theta)d\theta = E[x]\int_{-\infty}^{\infty} h(\theta)e^{-iw\theta}\Big|_{w=0}d\theta = E[x]H(0) \tag{7.3}$$

$$\text{where} \qquad H(w) = \int_{-\infty}^{\infty} h(\theta)e^{-iw\theta}d\theta \tag{7.4}$$

H(w) is the frequency response function which is the Fourier transformation of h(t). Equation (7.3) gives the propagation of mean value.

Propagation of Mean Square

The auto-correlation function of the response y(t) is given with (7.2) as

$$R_y(\tau) = E[y(t)y(t+\tau)] = E\left[\int_{-\infty}^{\infty} h(\theta_1)x(t-\theta_1)d\theta_1 \int_{-\infty}^{\infty} h(\theta_2)x(t+\tau-\theta_2)d\theta_2\right]$$

variables θ_1, θ_2 are used instead of θ to make relations clear.

Using the auto-correlation function of x(t) yields

$$R_y(\tau) = \iint_{-\infty}^{\infty} h(\theta_1)h(\theta_2)R_x(\tau-\theta_2+\theta_1)d\theta_1 d\theta_2 \qquad (7.5)$$

Taking Fourier transformation for both sides, the left hand side of (7.5) yields

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} R_y(\tau)e^{-iw\tau}d\tau = S_y(w) \qquad (7.6)$$

where $S_y(w)$ is the spectral density function of y. Similarly, the right hand side yields

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-iw\tau}d\tau\left\{\int_{-\infty}^{\infty} d\theta_1 \int_{-\infty}^{\infty} d\theta_2 h(\theta_1)h(\theta_2)R_x(\tau-\theta_2+\theta_1)\right\}$$

$$= \frac{1}{2\pi}\int_{-\infty}^{\infty} h(\theta_1)d\theta_1 \int_{-\infty}^{\infty} h(\theta_2)d\theta_2\left\{e^{iw(\theta_1-\theta_2)}\int_{-\infty}^{\infty} R_x(\tau-\theta_2+\theta_1)e^{-iw(\tau+\theta_1-\theta_2)}d(\tau+\theta_1-\theta_2)\right\}$$

$$= \int_{-\infty}^{\infty} h(\theta_1)e^{iw\theta_1}d\theta_1 \int_{-\infty}^{\infty} h(\theta_2)e^{-iw\theta_2}d\theta_2 S_x(w)$$

$$= H^*(w)H(w)\ S_x(w) = |H(w)|^2 S_x(w) \qquad (7.7)$$

where $H^*(w)$ is the complex conjugate of H(w). Now (7.6) and (7.7) yield

$$S_y(w) = |H(w)|^2 S_x(w) \qquad (7.8)$$

Using the inverse Fourier transform, mean square response $E[y^2]$ is given as

$$E[y^2] = R_y(\tau=0) = \int_{-\infty}^{\infty} S_y(w)\, e^{iw*0}\, dw = \int_{-\infty}^{\infty} |H(w)|^2 S_x(w) dw \tag{7.9}$$

If we can get the spectral density function of the input $S_x(w)$ and the frequency response function of the system H(w), the mean square response $E[y^2]$ is given with (7.9)

Application to Web Handling System

In order to use (7.3) and (7.9), it is necessary to use the frequency response function of the system H(w). This is done by modifying equation (2.10).

$$-L1 * \frac{de1}{dt} = v0*(1-e0)\,|_{Time=t} - a1*v0*(1-e1)\,|_{Time=t} \tag{2.10}$$

Now let $e0 = \overline{e0}$, $e1 = \overline{e1}$ at the steady state, then $\frac{d\overline{e1}}{dt} = 0$ and from (2.10),

$$\overline{e1} = 1 - \frac{1}{a1}(1 - \overline{e0}) \tag{7.10}$$

Also let $\widetilde{e0} = e0 - \overline{e0}$, $\widetilde{e1} = e1 - \overline{e1}$, then (2.10) yields

$$-L1 * \frac{d(\widetilde{e1} + \overline{e1})}{dt} = v0[\{1-(\widetilde{e0} + \overline{e0})\} - a1 * \{1-(\widetilde{e1} + \overline{e1})\}]$$

Using (7.10) yields

$$\frac{d(\widetilde{e1})}{dt} = \frac{v0}{L1}[\widetilde{e0} - a1 * \widetilde{e1}] \tag{7.11}$$

Equation (7.11) gives the relation of changes from the steady state points.

Now let $\widetilde{e0} = e^{iwt}$, $\widetilde{e1} = H(w)e^{iwt}$ in order to get H(w); then (7.11) yields

$$H(w) = \frac{\frac{1}{a1}}{1 + iTw} \quad \text{where } T = \frac{L1}{a1*v0} \tag{7.12}$$

Consequently, the propagation of spectral density is given as

$$S1(w) = S0(w)\ \frac{\left(\frac{1}{a1}\right)^2}{1+(Tw)^2} \tag{7.13}$$

where S0(w) and S1(w) represent the spectral density function of $\widetilde{e0}$ and $\widetilde{e1}$ respectively. If $\widetilde{e0}$ is white noise, S0(w) is a constant value S0, and (7.14) becomes

$$S1(w) = \frac{S0\left(\frac{1}{a1}\right)^2}{1+(Tw)^2} \tag{7.14}$$

Figure 7.1 shows S0 and S1(w) for w>0, with a1=1.0005 and T = 0.05997.

Using (7.3), (7.9), (7.12), and (7.14), propagations of mean and mean square are calculated as follow.

Mean:

$$E\left[\widetilde{e1}\right] = H(0)E\left[\widetilde{e0}\right] = \frac{1}{a1}E\left[\widetilde{e0}\right] \tag{7.15}$$

Mean square:

$$E\left[\widetilde{e1}^2\right] = \int_{-\infty}^{\infty} |\,H(w)\,|^2 S0(w)dw = \int_{-\infty}^{\infty} \left(\frac{1}{a1*T}\right)^2 \frac{S0(w)}{\left(\frac{1}{T}\right)^2 + w^2} dw \tag{7.16}$$

in case the input is white noise whose spectral density is S0, (7.16) yields

$$E\left[\widetilde{e1}^2\right] = \int_{-\infty}^{\infty} \left(\frac{1}{a1*T}\right)^2 \frac{S0}{\left(\frac{1}{T}\right)^2 + w^2} dw = \frac{\pi S0}{T * a1^2} \tag{7.17}$$

$$\text{since} \int_0^{\infty} \frac{dw}{a^2 + w^2} = \frac{\pi}{2a}$$

Equations (7.15) - (7.17) give the relation of mean and mean square between span 0 and 1.

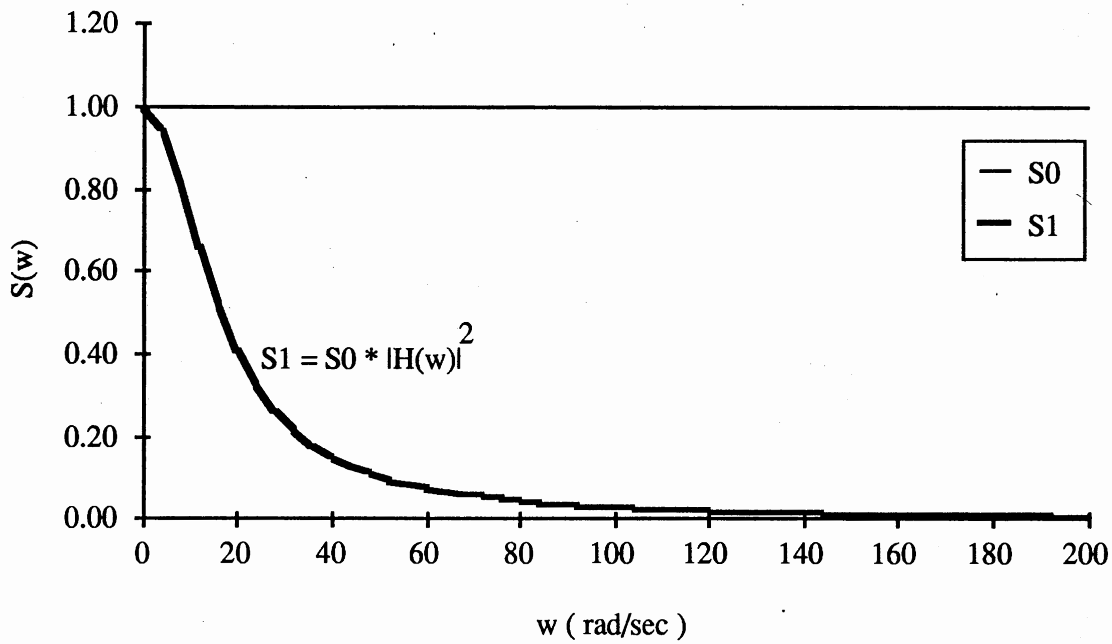

Figure 7.1. Propagation of Spectral Density

Simulation and Results

In order to verify the theoretical analysis given in (7.15) - (7.17), the Monte Carlo simulation was used. Before this simulation, a pseudo-random data generator was examined and was found that it generated the prescribed random data series. Requirements set here are uniform distribution in their amplitude with zero mean and unit variance, and whiteness in frequency. Details are shown in Appendix A.

Figures 7.2, 7.3, and 7.4 show the results of simulation with the elastic model in the single span. Infeed strain e0 and thus $\widetilde{e0}$ are under stochastic disturbance which is uniformly distributed in amplitude and has a white nature in frequency. Figure 7.2 shows a one time series which has 100 pieces of data in it. In the Monte Carlo simulation, a total of 5000 series are used. Actually, a random data generator creates 500,000 data sequentially, and 5000 series, each of which has 100 data, are generated from these series of random data.

The sample number 5000 was determined through trail calculation with various sampling numbers. Table 7.1 shows the propagation of mean and mean square of strain by comparing results from theoretical analysis and those from simulations. Conditions of these simulations are as follow.

(1)	Speed of roller 0; v0	= 1.0
(2)	Speed of roller 1; a1v0	= 1.0005 * v0
(3)	Length of span; L1	= 1.2
(4)	Infeed strain; $\widetilde{e0}$	mean = 0, mean square = $2\pi S0$ where S0 = 3.0E-10
(5)	Time step; dt	= 0.005
(6)	Sampling number	= 1000, 5000, 10000, 20000

Program 7 in Appendix B was used for this simulation. In this program, equation (7.11) was calculated using the 4th order Runge Kutta method.

As for the propagation of mean, results of the simulations do not agree with theoretically predicted ones. Increasing the sample number from 5000 to 20000 does not improve this much. Possible reasons are as follow.

(1) Error from discretizing white noise: Any digital computer cannot generate real white noise by nature. Ideally, the auto-correlation function of input vanishes other than the point of $\tau = 0$, but generated data must have the same value in the interval of [0,dt] where dt is the time step of the simulation. This is a source of discrepancy.

(2) Error from the pseudo-random process: As is mentioned in Appendix A, a computer cannot generate real random series of numbers; thus generated series has limitations in the properties. In this case, a shift from the desired mean, zero, may have affected the results.

(3) Error from the numerical calculation: The program uses huge numbers of iterations which causes accumulated error. In the Runge Kutta routine, the global error has the order of $O(h^4)$. Thus decreasing time step has a significant advantage in this routine. It is however difficult to do this because of its long calculation time.

As for the propagation of mean square, both results agree within a few percentage points of relative error. This validates the theoretical analysis. Different sample numbers gave slightly different relative errors in this calculation, but they are not out of the mark. The number 5000 was determined for convenience in calculation and this result.

Table 7.2 shows results similar to those shown in Table 7.1. Here two cases of speed v0 were compared and both results show the same tendency of those in Table 7.1. In this case,the error lies within 1%. This result confirms the validation more, because speed v0 affects the system itself through equation (7.12).

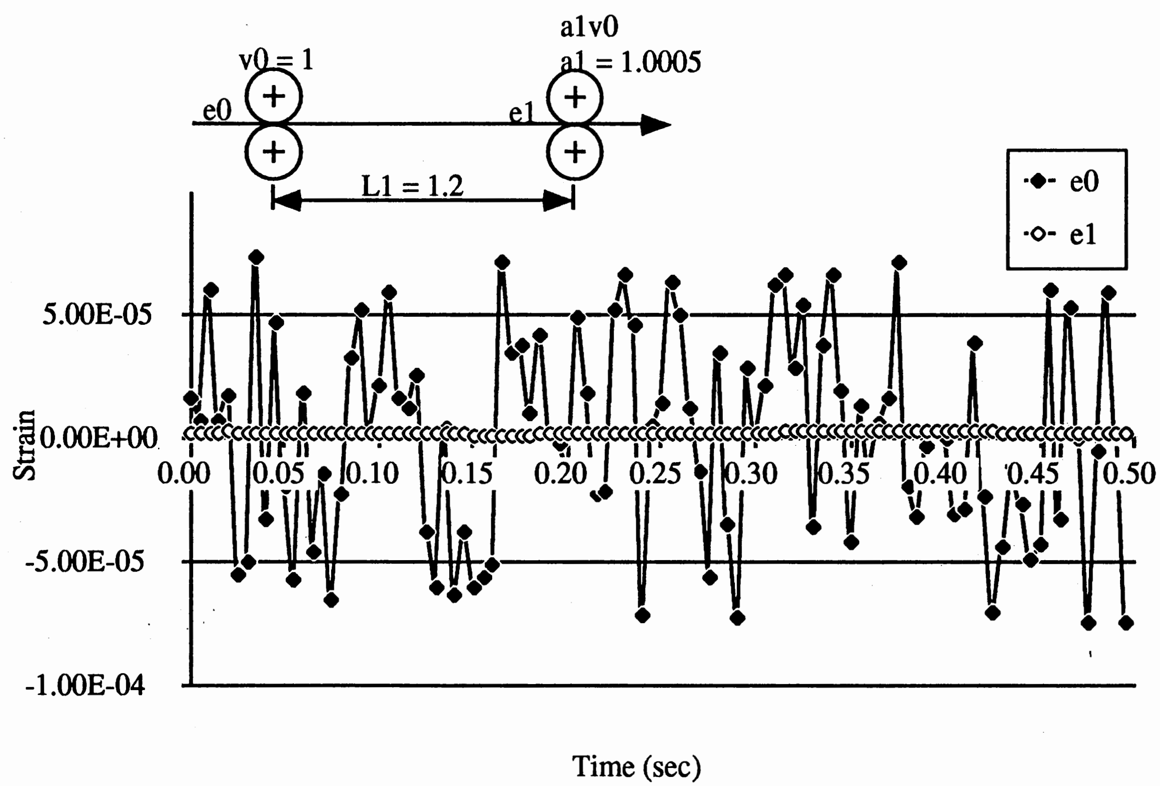

Figure 7.2. Time Series of Strain

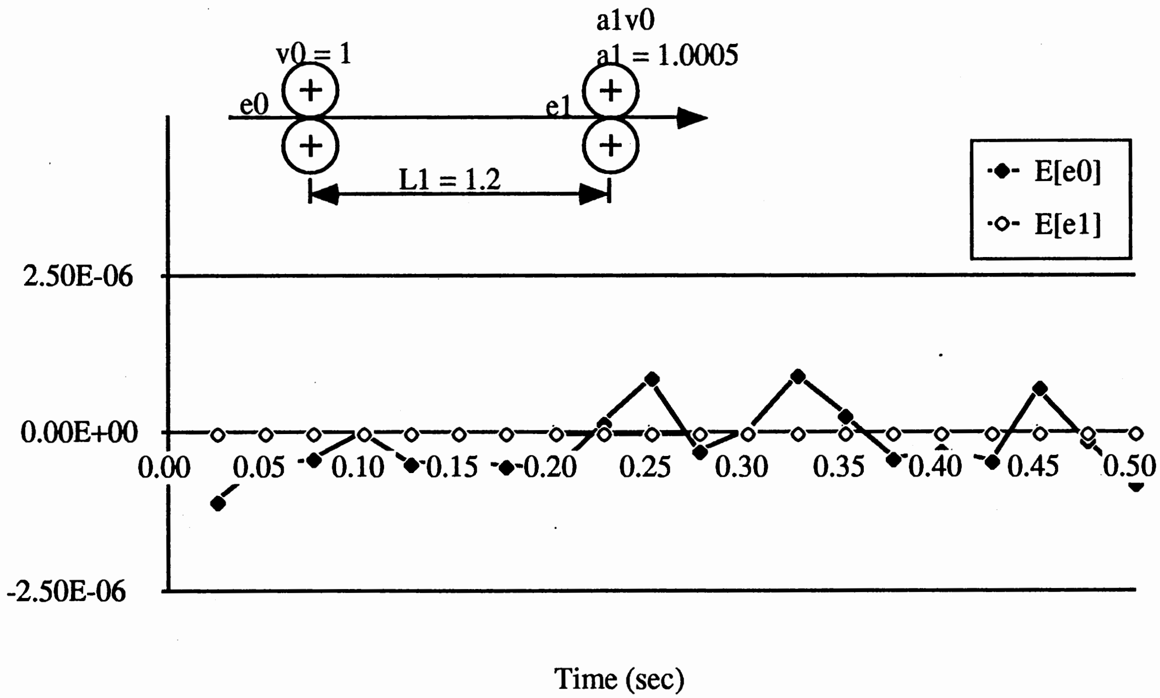

Figure 7.3. Propagation of Mean E[strain]

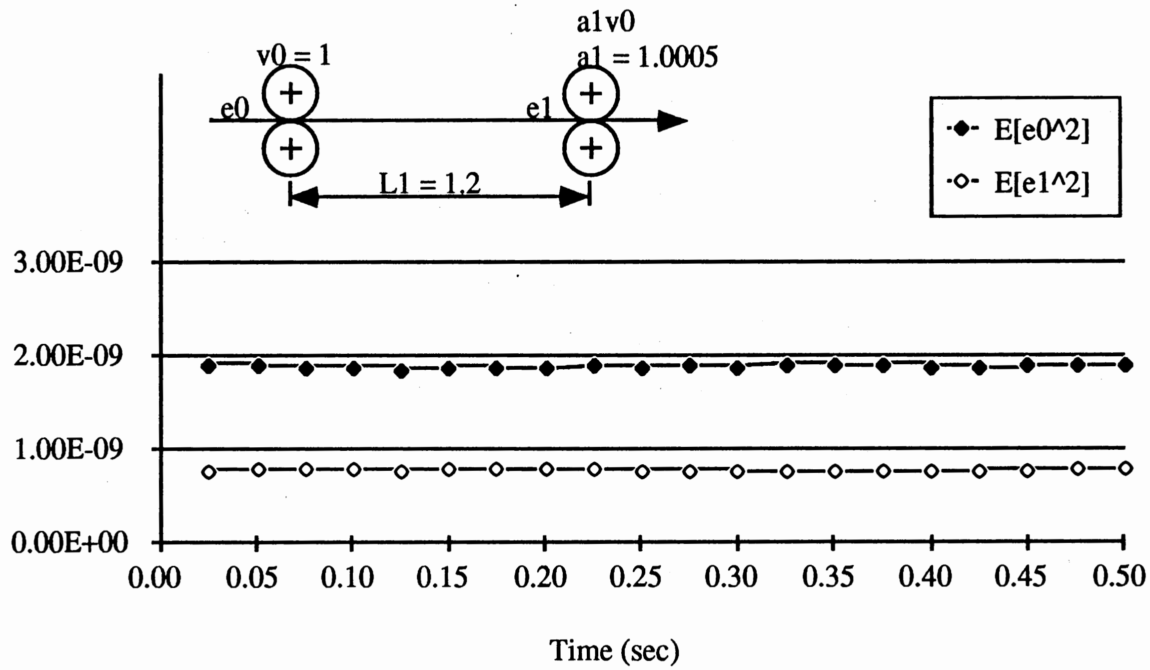

Figure 7.4. Propagation of Mean Square E[strain^2]

TABLE 7.1

COMPARISON OF RESULTS FOR DIFFERENT AMOUNT OF SAMPLING NUMBER

Sample		Simulation	Theory	Relative Error
1000	E[e0]	-1.982e-07	0	–
	E[e0^2]	1.892e-09	1.885e-09	0.38%
	E[e1]	-5.525e-08	0	–
	E[e1^2]	7.664e-10	7.850e-10	-2.37%
5000	E[e0]	-1.600e-07	0	–
	E[e0^2]	1.887e-09	1.885e-09	0.09%
	E[e1]	3.491e-09	0	–
	E[e1^2]	7.797e-10	7.850e-10	-0.67%
10000	E[e0]	9.795e-10	0	–
	E[e0^2]	1.883e-09	1.885e-09	-0.09%
	E[e1]	3.274e-09	0	–
	E[e1^2]	7.733e-10	7.850e-10	-1.50%
20000	E[e0]	1.860e-08	0	–
	E[e0^2]	1.885e-09	1.885e-09	-0.01%
	E[e1]	-8.379e-10	0	–
	E[e1^2]	7.658e-10	7.850e-10	-2.44%

$$* \text{ relative error} = \frac{\text{simulation - theory}}{\text{theory}}$$

TABLE 7.2

COMPARISON OF RESULTS

Web speed		Simulation	Theory	Relative Error
vo = 1 (m/s)	E[e0]	-1.600e-07	0	–
	E[e0^2]	1.887e-09	1.885e-09	0.09%
	E[e1]	3.491e-09	0	–
	E[e1^2]	7.797e-10	7.850e-10	-0.67%
vo = 10 (m/s)	E[e0]	-1.600e-07	0	–
	E[e0^2]	1.887e-09	1.885e-09	0.09%
	E[e1]	2.876e-09	0	–
	E[e1^2]	7.805e-09	7.850e-09	-0.57%

Chapter Summary

This chapter discussed the statistical approach to handling uncertainty of the web properties and/or disturbances. The main topics are as follow.

(1) The theoretical analysis on propagation of mean and mean square was reviewed. This is based on spectral analysis and propagation can be calculated using the spectral density function of the input and the frequency response function of the system. Final equations are as follow.

Mean: $$E[y] = E[x]H(0) \tag{7.3}$$

Mean square: $$E\left[y^2\right] = \int_{-\infty}^{\infty} |H(w)|^2 S_x(w)dw \tag{7.9}$$

where H(w) is frequency response function of the system.

(2) Application of these general results to the web handling system gives the relation

Mean: $$E\left[\tilde{e1}\right] = \frac{1}{a1} E\left[\tilde{e0}\right]$$

Mean square: $$E\left[\tilde{e1}^2\right] = \frac{\pi S0}{T * a1^2}$$ (for white noise input with density S0)

These are based on the equation (7.11); $\frac{d(\tilde{e1})}{dt} = \frac{v0}{L1}\left[\tilde{e0} - a1 * \tilde{e1}\right]$

(3) These relations are verified with the Monte Carlo simulation for mean and mean square. Results for mean square agree with those from the theoretical analysis within 1% of relative error, while there are some discrepancies in the calculation of mean. Possible reasons for this discrepancy are presented.

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

In order to expand the present theory of web handling, various topics have been discussed in this paper. This chapter summarizes all of them.

In Chapter II, fundamental equations of strain propagation are shown. They provide an understanding of how strain in one span relates to that in adjacent spans. If there is no slippage between roller and web, their principle is the conservation of transferring mass. If slippage occurs, and thus the speed of the web is unknown, the relation of force is necessary in addition. It should be noted that these equations give strain just before each roller and a specific web model is necessary to calculate its distribution in the span and/or to observe its behavior. Also a web model is required to calculate stress from the strain.

In chapters III and IV, two types of models are discussed. Both of them are macroscopic models which use a spring and/or a damper element(s) to express the behavior of the web. However, they are not necessarily related to physical meanings, actual bondage of pulp fiber for example. The elastic web model discussed in Chapter III has only one spring element in it. This model is convenient and acceptable for many practical applications, which makes it very popular among researchers. However, the visco-elastic model in Chapter IV is necessary if rheological behavior of the web is the concern. This model extends the expression using additional damper(s) and spring element(s) and can explain relaxations, creep, and so on.

The main topics in Chapter III are as follow.

(1) Basic behavior of the strain was discussed in dynamic and steady state condition. This propagation process shows the first order low pass filtering.

(2) As an extension of this discussion, two cases were examined. Both of them allow for changes in the elastic modulus of the web. The first case is to simulate the disturbance outside the span which is inescapable in actual operation. The result show the filtering nature.

(3) The second one is to simulate the drying process at its steady state where the elastic modulus changes inside the span. This indicates the distributed strain of the web under uniform stress.

The main topics in Chapter IV are as follow.

(1) Various visco-elastic models are discussed. The voigt model is convenient to discuss problems of strain under a given stress which is common in web handling system. Their strain responses for a given stress input (step) were examined.

(2) Theoretical analysis was made to apply them to a web handling system with additional assumption of uniform mass flow rate. Analysis gives strain distribution inside the span for both slipping and non slipping cases.

(3) Simulations demonstrate the result of analysis. It is found that a visco-elastic web changes stress according to the operating speed, which the elastic model cannot predict.

(4) The result with an actual web (newspaper) indicates the suitability of the elastic model for this application.

Now it is possible to examine another case by comparing results directly.

A web handling system can be operated without problems so long as the relationships in Chapter II govern it. This shows the first order filtering nature and strains propagate in this manner. There is, however, a chance of self-induced oscillation even when everything seems to be right. Some systems depend on web tension to press web to the roller, and this might cause "self reference" which creates the stick-slip phenomenon in the operation. Moreover, there are problems of slackness. This is partly from gaps between models and actual systems. A certain level of stress is necessary to keep stable handling in an actual system, while the model has no problem even if its stress is almost zero or negative.

Chapter V discusses the stick-slip phenomenon and Chapter VI shows various sources of slackness. The main topics in Chapter V are as follow.

(1) Non-linear friction models are presented which have two levels of friction coefficients according to the condition of slippage.

(2) Strain behavior with this model was theoretically examined, and the propagation of strain in a two span model was shown.

(3) According to the simulation, even a similar configuration may cause significant differences in their behavior. Once the stick-slip phenomenon happens, it ends up with a stable limit cycle which does not disappear by itself.

The main topics in Chapter VI are as follow.

(1) The elastic model causes slackness if the change of the span is radical. This is a temporary low strain, but it can be crucial in actual operation.

(2) The visco-elastic model may cause slackness more easily. This comes from the damper element which shares the strain but does not contribute to the stress. A handling system with this kind of web needs speed increases along the process line.

(3) If stick-slip happens, it may cause slackness. Although this level of strain is

higher than that which is calculated for the elastic model, this is troublesome because of its repetitive nature while slackness in the elastic model is a temporary one.

It should be noted that they are problems under the fundamental relationships shown in Chapter II.

Finally in Chapter VII, a statistical approach to uncertainty is shown. Actual webs and machines are not ideally made, and they are not free from uncertainty in their nature. Thus, it is important to estimate the ranges of possible output for a given uncertain input.

The main topics in Chapter VII are as follow.

(1) Propagation of mean and mean square was reviewed theoretically, and was applied to the web handling system. This is based on the spectral analysis, and propagation can be calculated with the spectral density function of the input and the frequency response function of the system.

(2) These relationships are verified using the Monte Carlo simulation. Results for mean square agree with those from the theoretical analysis within 1% of relative error, while there are some discrepancies in results of mean. Possible reasons for this discrepancy are presented.

Recommendations for Further Study

Table 8.1 shows the scope of this paper with respect to items treated. The mark (√√√) indicates newly covered items, while (√) shows a review of previous works and (*) shows items which were reported but not reviewed here.

TABLE 8.1

SCOPE OF STUDY

Classification of the study			properties of web					
			elastic			visco-elastic		
			deterministic		sto-chastic	deterministic		sto-chastic
	condition	stick slip	constant	varying		constant	varying	
theory	static	no	√	–	√√√	√√√	–	
	dynamic	yes	√√√	√√√				
		no	√	√	√√√			
experiment	static	no	*	–			–	
	dynamic	yes						
		no	*	*				

Notes. √ : covered, √√√ : newly covered in this paper, * : done, -- : not available

As for this scope, filling blanks extends these studies. It is, however, difficult to combine everything at once, and my recommendations for the next studies are as follow.

Study with the elastic model should emphasize practical application. This includes development of control strategies and analysis of complicated phenomena such as stick-slip on the machines. These studies will contribute to actual matters including designing machines and trouble shooting.

Study with the visco-elastic model should emphasize both theoretical and experimental work. At first, it is necessary to identify the parameters of the model through experiments. As was mentioned, springs and dampers in the model are fictional, and the number of elements and their parameters come from overall behavior through experiments. This is done partly for paper and plastic film, but systematic data will be useful. Experiments for verifying the theory shown here are also necessary before expanding this to a more complicated one. As for the theory concerned, extending it to a dynamic system is an important project, but distributed strain and its dependency on time will make this work difficult. These studies will contribute to a better understanding of the behavior of wet papers and hot plastic film.

Problems like slackness and stick-slip need experimental work also. Since they are relatively practical problems, some guide lines for preventing them will be helpful to people in this industry.

The stochastic approach needs input information, such as mean and standard deviation of properties of web. This includes how they are affected by certain environments. This approach can also be extended to the visco-elastic model. Analysis in steady state may be done first, because this is complicated enough to investigate. The visco-elastic model makes the frequency response function H(w) complex and its initial value makes calculation complicated. These studies will contribute to predicting possible occurrence of output, and thus the ability to design machines which have better reliability.

Finally, the author hopes that these studies contribute not only to the extension of theory itself but to making our lives better by improving the machines in real life.

A SELECTED BIBLIOGRAPHY

1. Taguchi, T., Morimoto, K., Terada, Y., and Sakamoto, Y. "Web Behavior on Newspaper Web Rotary Press" Technical Review Mitsubishi Heavy Industries, ltd. October (1985)

2. Watanabe, T. "Paper Tension and Misregistration on Web Rotary Press" JSPE Vol. 5, No. 10, 33-40 (1961)

3. Pfeiffer, J. D. "Web guidance concepts and applications" TAPPI Vol. 60, No. 12 (1977)

4. Sceuter, K. R., and Belau, L. "The elasticity modulus of a running web" 18th IARIGAI conference, June (1985)

5. Fujii, Y. "On the Rheological Properties of Paper (part I. Mechanical model)" Japan TAPPI Vol. 10, No. 69, 660-669 (1956)

6. Hauptmann, E. G., and Cutshall, K. A. "Dynamic mechanical properties of wet paper webs" TAPPI Vol. 60, No. 10, 106-108 (1977)

7. Oka, S., Kawai, H., etal. Rheology Nyumon (An Introduction to Rheology) Kogyo Chousa-kai, Tokyo, Japan, 1970

8. Perez, M. "A Model for the Stress - Strain Properties of Paper" TAPPI Vol. 53, No. 12, 2237-2242 (1970)

9. "Friction at Paper Surface and Transfer from Adhesion to Slip" Japan TAPPI Vol. 34, No. 6, 53-56 (1980) This article was translated from Wochenbl für Papierfahr 107, (23/24), 941-942, 944-946 (1979)

10. Soeda, T., Tokumaru, H., Nakamizo, T., and Iwai, Z. Shindou Kougakuno Kiso (Basic Vibration Theory) Nisshin Shuppan, Tokyo, Japan, 1978

11. Newland, D. E. An introduction to RANDOM VIBRATIONS and SPECTRAL ANALYSIS Longman Scientific & Technical, Essex, England, 1984

12. Press, W.H., Flannery, B. P., Teukolsky, S.A., and Vetterling, W. T. NUMERICAL RECIPES in C Cambridge university Press, Cambridge, England, 1988

APPENDICES

APPENDIX A

VERIFICATION OF RANDOM DATA GENERATOR

Introduction

The Monte Carlo simulation needs a certain amount of random data whose nature fits the prescribed distribution and whiteness. Practically, the Chi square test is widely used to verify the distribution, and the auto-correlation function is used to check the whiteness of the data series. In order to verify the nature of the generator used in Chapter VII, both are calculated here and results indicate that this generator satisfies the requirements.

Pseudo-Random Data Generator

It is impossible for any digital computer to generate a pure random data series because all programs are deterministic in their nature. There are, however, several ways to create pseudo-random data series which can be used for most of the calculations. [12] In Chapter VII, one type of these programs is used to create the random data series of unit deviates. (ran1(), a part of program 7 in Appendix B) This program uses the linear congruential and shuffling method. The data series with these two methods can avoid periodic sequences and sequential correlations effectively, both of which the system provided routine tends to suffer from.[12] After generating the data, program unit(), also a part of program 7, converts them so that the series have zero mean and a variance of 1.0.

Verification of the Nature of Uniform Distribution

From the calculation of 500,000 pieces of data, the mean and the variance are 1.329e-04 and 1.000 respectively. Both are close to values prescribed. (mean = 0.0, variance = 1.0)

Figure A.1 shows the results of the Chi square test for 5000 data series, each of which has 12 pieces of data in it. According to the calculation,

$X^2 = 5.19 < \chi^2_{9,\,0.80} = 5.38$ where X^2 is defined as

$$X^2 = \sum_{i=0}^{11} \frac{(F_i - f_i)^2}{F_i},$$

Fi : expected amount of data in each interval.
fi : actual amount of data in each interval.

The results indicate that this is uniform distribution at the 0.80 level of significance. [12] Program 8 and 10 in Appendix B were used for the calculations shown here.

Verification of the Nature of Whiteness

Figure A.2 shows the auto-correlation function for 500 data series each of which has 11 data in it. The values are normalized so that they become 1.0 when the time lag $\tau = 0$. Values other than that of $\tau = 0$ vanish equally, and this fact indicates the nature of whiteness.

Program 9 in Appendix B was used for the calculations shown here.

Conclusion

This generator with conversion gives a series of data which is distributed uniformly with a mean of zero and a variance of 1.0, and having the nature of whiteness. Thus this generator satisfies the requirements of the simulation discussed in Chapter VII.

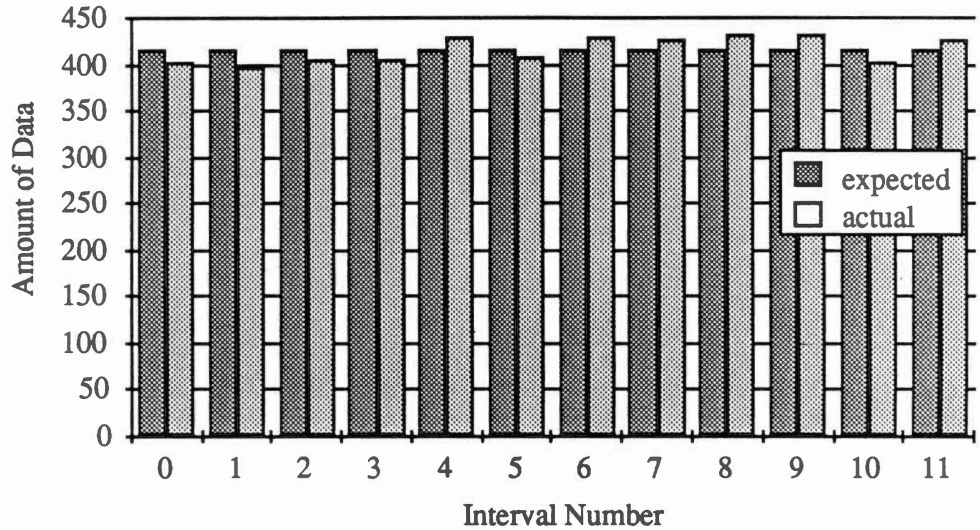

Figure A.1. Number of Data in Each Interval

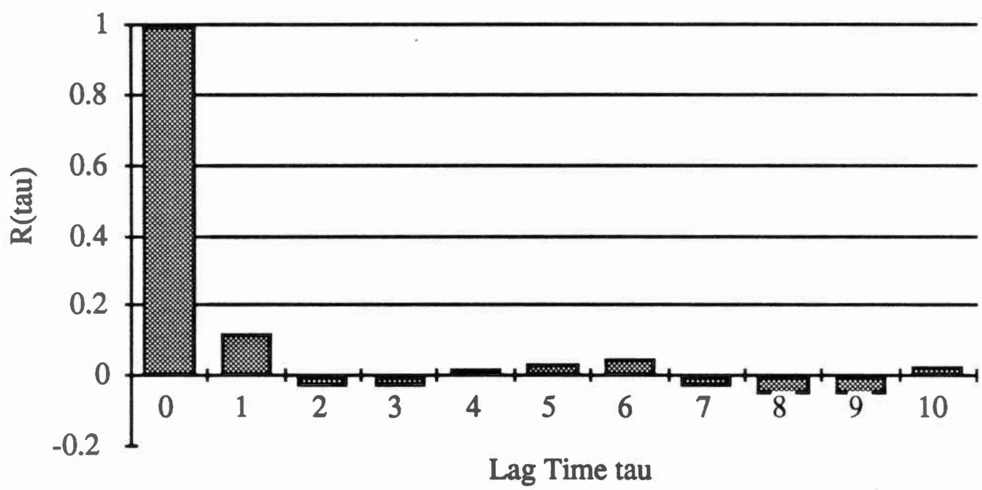

Figure A.2. Auto Correlation Function

APPENDIX B

COMPUTER PROGRAM LISTINGS

All program listed here are for the system shown below.
Notice that there is slight difference in the way of writing program from compiler to compiler.

Computer : Macintosh II (Apple Computer)

Compiler : Think C (v4.0) (SYMANTEC)

: All calculation were done with math coprocessor option ON.

Some functions necessary for the calculation may not appear in all programs to avoid the redundancy. They are attached to the programs appeared beforehand.

PROGRAM 1

```
/*****************************************************m*a*s*a*k*a*z*u*

        output      : strain distribution in the span
        web         : visco-elastic / generalized Voigt model
                    : deterministic
        operation   : steady state
                    : deterministic
        slip        : none

*****************************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <math.h>

#define   MAX_ITER    50          /* #max of iteration                    */
#define   CRITERIA    0.0005      /* that of correction                   */
#define   NSUBSPAN    10          /* #subspan in a span                   */
#define   MAX_ELEM    10          /* #max of element in the model         */
#define   MAX_SPAN    10          /* #max of span in the calculation      */

int       nspan, nelement;        /* # of span & element in model         */
double    gg, ep_0g, ep_1g;       /* additional elastic element           */
double    es, ep_0s, ep_1s;       /* additional plastic element           */
double    sg_1;                   /* stress in the span 1                 */
double    vg[MAX_ELEM];           /* ith spring  coeff in Voigt model     */
double    ve[MAX_ELEM];           /* ith damper coeff in Voigt model      */
double    lm[MAX_ELEM];           /* ith delay time: lm = ve / vg         */
double    ep_0e[MAX_ELEM];        /* ith strain @ end of span 0           */
double    ep_1e[MAX_ELEM];        /* ith strain @ end of span 1           */
```

```
main()
{
	int		i, j, k;				/* counter	*/
	int		flag, iter, ispan;			/* loop controller	*/

	double		al[MAX_SPAN];				/* alpha  of each span			*/
	double		ls[MAX_SPAN];				/* length of each span			*/
	double		al0, al1;				/* alpha  of span 0 & 1			*/
	double		ls1;					/* length of span 1			*/
	double		v0;					/* base speed				*/
	double		ep_INg;					/* elastic strain @ span IN		*/
	double		ep_INs;					/* plastic strain @ span IN		*/
	double		ep_INe[MAX_ELEM];			/* ith  strain @ end of span IN		*/
	double		ep_INet;				/* total strain @ end of span IN	*/
	double		ep_0et, ep_1et;				/* total strain @ end of span 0 & 1	*/
	double		jt, it;					/* J(t) & I(t)  @ end of span 1		*/
	double		t1, tm, dt;				/* total time, temp time, delta time	*/
	double		kx;					/* mass transfer (constant value)	*/
	double		lc;					/* calculated length of the span	*/
	double		er;					/* relative error in the iteration	*/
	double		ep[NSUBSPAN+1];				/* strain in the subspan		*/
	double		lx[NSUBSPAN+1];				/* position of subspan in the span	*/
	double		fne(), fni(), fnj();			/* functions for the model		*/
	double		bs;					/* for graphic use			*/
	double		q1;					/* dummy				*/

	nspan = 2;
	v0		= 1.0;
	al[0]		= 1.0;
	al[1]		= 1.003;	ls[1]		= 1.2;
	al[2]		= 1.0;		ls[2]		= 2.0;

	nelement	= 1;
	gg		= 2.0e9;	ep_INg		= 0.003;
	es		= 4.0e8;	ep_INs		= 0.0005;
	vg[1]		= 1.0e9;	ve[1]		= 5.0e8;

	ep_INet		= ep_INg + ep_INs;
	for ( i=1; i<=nelement; ++i ) {
		ep_INe[i]	= ep_INg * gg / vg[i];
		ep_INet		+= ep_INe[i];
		lm[i]		= ve[i] / vg[i];
	}

	ep_0g = ep_INg;
	ep_0s = ep_INs;
	ep_0et= ep_INet;
	for ( i=1; i<=nelement; ++i ) {
		ep_0e[i] = ep_INe[i];
	}
	kx	= v0  / ( 1.0+ep_INet );
	bs	= 0.0;

	for ( ispan=1; ispan<=nspan; ++ispan ) {
		al0	= al[ispan-1];
		al1	= al[ispan];
```

```
        ls1     = ls[ispan];
        ep_1et = al1 * (1.0+ep_lNet) - 1.0;
        t1      = 2.0 * ls1 / (al1+al0) / v0;
        er      = 999.0;
        iter    = 0;

        while ( (er > CRITERIA) && (++iter < MAX_ITER) ) {
                jt      = fnj( t1 );
                it      = fni( t1 );
                sg_1    = ( ep_1et - it ) / jt;
                ep_1g= sg_1 / gg;
                dt      = t1 / NSUBSPAN;
                q1      = 0.0;
                for ( j=0;j<=NSUBSPAN; ++j ) {
                        tm      = dt * (double) j;
                        ep[j]   = fnj( tm ) * sg_1 + fni( tm );
                        q1      += 2.0 * ( 1.0 + ep[j] );
                }
                lc      = ( q1 - ep[0] - ep[NSUBSPAN] -2.0 ) * kx * dt / 2.0;
                t1      = t1 * ls1 / lc;
                er      = (double) fabs((lc-ls1)/ls1);
        }

        lx[0] = 0.0;
        for ( j=1;j<=NSUBSPAN; ++j )
                lx[j] = lx[j-1] + kx * ( 1 + (ep[j-1]+ep[j])/2.0 ) * dt;
        printf( "span number %d\n\n", ispan );
        for ( j=0;j<=NSUBSPAN; ++j ) {
                ep_1s = sg_1 * ( dt * (double) j ) / es + ep_0s;
                printf( "%lf\t%lf\t", lx[j]+bs, ep[j] - ep_1g - ep_1s );
                printf( "%lf\t%lf\t%lf\t%lf\n", ep_1s, ep_1g, ep[j], ep_1et );
        }

        for ( i=0; i<=nelement; ++i ) {
                fne( t1 );
                ep_0e[i] = ep_1e[i];
        }
        ep_0s = ep_1s;
        bs      += lx[NSUBSPAN];
    }
}
```

```
double fnj( t )        /* function J(t) */
double t;
{
        double vj, f1();
        int    i;
        vj = 1.0 / gg + t / es;
        if ( nelement != 0 ) {
                for (i=1; i<=nelement; ++i )
                        vj += f1(t, i);
        }
        return( vj );
}

double fni( t )        /* function I(t) */
double t;
{
        double vi, f2();
        int    i;
        vi = ep_0s;
        if ( nelement != 0 ) {
                for (i=1; i<=nelement; ++i )
                        vi += f2(t, i);
        }
        return( vi );
}

double fne( t )        /* strain caused by dampers */
double t;
{
        double f1(), f2();
        int    i;
        if ( nelement != 0 ) {
                for (i=1; i<=nelement; ++i )
                        ep_1e[i] = sg_1 * f1(t, i) + f2(t, i);
        }
}

double f1( t, i )
double t;
int    i;
{
        return( ( 1.0 - (double) exp( -t/lm[i] ) ) / vg[i] );
}

double f2( t, i )
double t;
int    i;
{
        return( (double) exp( -t/lm[i] ) * ep_0e[i] );
}
```

PROGRAM 2

```
/*****************************************************m*a*s*a*k*a*z*u*

        output      : strain with respect to time
        web         : elastic
                    : deterministic
        operation   : dynamic
                    : deterministic
        slip        : stick-slip

*****************************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <math.h>

#define     MAX     100             /* #calculation                     */
#define     EVR     5               /* #skip for the display            */
#define     DT      0.005           /* delta time                       */
#define     THE     120             /* lapping angle at roller (deg)    */
#define     MYU1    0.4             /* friction coefficient (high)      */
#define     MYU2    0.15            /* friction coefficient (slip)      */

double      ep0     = 0.001;        /* initial strain                   */
double      l1      = 1.2;          /* length of span 1                 */
double      l2      = 2.0;          /* length of span 2                 */
double      v0      = 10.0;         /* standard speed of the web        */
double      t       = 0.0;          /* time                             */

double      ep1, ep2;               /* strain at each span              */
double      epc1, epc2, epc3, epc4; /* critical strain of ep2           */
double      kf1, kf2;               /* friction const = exp(myu*the)    */
int         cnd;                    /* 0:non slip,1:slip,-1,:reverse    */
```

```
main()
{
        int             i;                                  /* counter    */
        double          a1(), a2();                         /* alpha1,2   */
        double          slip1(), slip2(), nslip();          /* ep1,ep2    */

        kf1     = (double) exp( MYU1 * THE * 3.14159265359 / 180.0 );
        kf2     = (double) exp( MYU2 * THE * 3.14159265359 / 180.0 );
        ep1     = ( 1.0+ep0 ) * a1(t) - 1.0;
        ep2     = ( 1.0+ep0 ) * a2(t) - 1.0;
        cnd     = 0;

        for ( i=0; i<MAX; ++i ) {
                t       = DT * (double) i;
                epc1    = ep1 / kf1;  epc2    = ep1 / kf2;
                epc3    = ep1 * kf2;  epc4    = ep1 * kf1;

                if      ( ((cnd== 1) && (ep2<epc2)) || (ep2<=epc1) )
                        slip1();
                else if ( ((cnd==-1) && (epc3<ep2)) || (epc4<=ep2) )
                        slip2();
                else
                        nslip();

                printf( "%lf\t%lf\t%lf\t%lf\t%lf\n",t,ep1,ep2,epc1,epc4 );
        }
}


double a1()                                         /* alpha 1    */
{
        return( (t<10.0*DT)?1.002:1.003 );
}

double a2()                                         /* alpha 2    */
{
        return( 1.001 );
}
```

```
double nslip()                          /* calculate ep1, ep2 with RK4 */
{
        double h, k1, k2, k3, k4, q1;
        double f1(), f2();

        h       = DT;
        k1      = h * f1( t, ep1 );
        k2      = h * f1( t+h/2.0, ep1+k1/2.0 );
        k3      = h * f1( t+h/2.0, ep1+k2/2.0 );
        k4      = h * f1( t+h, ep1+k3 );
        q1      = ep1 + ( k1 + k2*2.0 + k3*2.0 + k4 ) / 6.0;

        k1      = h * f2( t, ep2 );
        k2      = h * f2( t+h/2.0, ep2+k1/2.0 );
        k3      = h * f2( t+h/2.0, ep2+k2/2.0 );
        k4      = h * f2( t+h, ep2+k3 );
        ep2     = ep2 + ( k1 + k2*2.0 + k3*2.0 + k4 ) / 6.0;
        ep1     = q1;

        cnd     = 0;
}

double slip1()                          /* calculate ep1, ep2 in normal slip */
{
        double h, k1, k2, k3, k4;
        double f3();

        if ( cnd==0 )
        ep2     = ( l1*ep1 + l2*ep2 ) / ( l1*kf2 + l2 );
        h       = DT;
        k1      = h * f3( t, ep2 );
        k2      = h * f3( t+h/2.0, ep2+k1/2.0 );
        k3      = h * f3( t+h/2.0, ep2+k2/2.0 );
        k4      = h * f3( t+h, ep2+k3 );
        ep2     = ep2 + ( k1 + k2*2.0 + k3*2.0 + k4 ) / 6.0;
        ep1     = ep2 * kf2;

        cnd     = 1;
}
```

```
double slip2()                          /* calculate ep1, ep2 in reverse slip */
{
        double h, k1, k2, k3, k4;
        double f4();

        if ( cnd==0 )
        ep2     = ( l1*ep1 + l2*ep2 ) / ( l1/kf2 + l2 );
        h       = DT;
        k1      = h * f4( t, ep2 );
        k2      = h * f4( t+h/2.0, ep2+k1/2.0 );
        k3      = h * f4( t+h/2.0, ep2+k2/2.0 );
        k4      = h * f4( t+h, ep2+k3 );
        ep2     = ep2 + ( k1 + k2*2.0 + k3*2.0 + k4 ) / 6.0;
        ep1     = ep2 / kf2;

        cnd     = -1;
}


double f1( t, x )
double t, x;
{
        double a1();
        return( ( a1(t)/(1.0+x) - 1.0/(1+ep0) ) * v0 / l1 );
}

double f2( t, x )
double t, x;
{
        double a1(), a2();
        return( ( a2(t)/(1.0+x) - a1(t)/(1+ep1) ) * v0 / l2 );
}

double f3( t, x )
double t, x;
{
        double a2();
        return( ( a2(t)/(1.0+x) - 1.0/(1+ep0) ) * v0 / (l2+l1*kf2) );
}

double f4( t, x )
double t, x;
{
        double a2();
        return( ( a2(t)/(1.0+x) - 1.0/(1+ep0) ) * v0 / (l2+l1/kf2) );
}
```

PROGRAM 3

```
/*****************************************************m*a*s*a*k*a*z*u*

        output      : strain with respect to time
        web         : elastic modulus changes
        operation   : dynamic, deterministic

*****************************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define     MAX     100          /* #calculation                     */
#define     EPIN    0.001        /* initial strain                   */
#define     DT      0.005        /* delta time                       */
#define     AMP     0.2          /* amplitude of the disturbance     */
#define     TAU     0.5          /* time constant  for the decaying  */
#define     OMEGA   50.0         /* angle velocity for the Sin input */

double      l1  =   1.2;         /* length of span 1                 */
double      a1  =   1.001;       /* speed ratio of span1 to span 0   */
double      v0  =   20.0;        /* standard speed of the web        */
double      t   =   0.0;         /* time                             */
double      ep1, ep0;            /* strain at each span              */

main()
{
        int     i;                      /* counter    */
        double EPS(), RK4();            /* ep0, ep1   */

        ep1   = a1 * ( 1.0 + EPIN ) - 1.0;
        for ( i=0; i<MAX; ++i ) {
                t       = DT * (double) i;
                ep0     = EPS( t );
                ep1     = RK4( t, ep1 );
                printf( "%lf\t%lf\t%lf\n", t, ep0, ep1 );
        }
}

double EPS( t )                         /* ep0 */
double t;
{
        double x = EPIN;
        int   rand();

        x += AMP*EPIN*( 2.0*(double) rand()/RAND_MAX-1.0 ) + exp(-t/TAU);
        return( x );
}
```

PROGRAM 4

```
/*****************************************************m*a*s*a*k*a*z*u*

        output      : strain distribution in the span
        web         : elastic modulus varies in the middle of the web
        operation   : steady state
                    : deterministic
        slip        : none

 *****************************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <math.h>
#define     NSUBSPAN        20              /* #subspan in a span                  */
#define     DLG             2.0             /* changing ratio of elastic modulus   */

main()
{
        int         j;                      /* counter                             */
        double      al1     = 1.001;        /* alpha1 of the span                  */
        double      ls1     = 1.2;          /* length of the span                  */
        double      x1      = 0.5;          /* start position of the dryer         */
        double      x2      = 0.8;          /* end   position of the dryer         */
        double      ep_INg  = 0.001;        /* elastic strain at the span IN       */
        double      ep, x, dx, ke;
        double      fng();                  /* elastic modulus at the point        */

        ke    = al1 * (1.0+ep_INg) - 1.0;   /* constant                            */
        dx    = ls1 / NSUBSPAN;             /* length of subspan                   */
        for ( j=0;j<=NSUBSPAN; ++j ) {
                x       = dx * (double) j;
                ep      = ke / fng( x, ls1, x1, x2 );
                printf( "%lf\t%20.16lf\n",x, ep );
        }
}

double fng( p, l, p1, p2 )  /* strain ratio at x (ep=1 at end)   */
double p, l, p1, p2;
{
        double g;                   /* value of Gg */

        if ( ( 0<=p) && (p<p1) )
                g = 1.0 - (p2-p1) * DLG;
        else if ( (p1<=p) && (p<p2) )
                g = 1.0 - (p2- p) * DLG;
        else
                g = 1.0;
        return( g );
}
```

PROGRAM 5

```
/*****************************************************m*a*s*a*k*a*z*u*

      output        : strain distribution in the span
      web           : visco-elastic / generalized Voigt model
                    : deterministic
      operation     : steady state
                    : deterministic
      slip          : at the central roller

 *****************************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <math.h>

#define   MAX_ITER    50              /* #max of iteration*/
#define   CRITERIA    0.0005          /* that of correction */
#define   NSUBSPAN    10              /* #subspan in a span                 */
#define   MAX_ELEM    10              /* #max of element in the model       */
#define   PI          3.1415926536    /* π                                  */

int       nelement;                   /* # of element in the model          */
double    gg,ep_0g,ep_1g,ep_2g;       /* additional elastic element         */
double    es,ep_0s,ep_1s,ep_2s;       /* additional plastic element         */
double    sg_1, sg_2;                 /* stress in the span 1 & 2           */
double    vg[MAX_ELEM];               /* ith spring  coeff in Voigt model   */
double    ve[MAX_ELEM];               /* ith dumping coeff in Voigt model   */
double    lm[MAX_ELEM];               /* ith delay time: lm = ve / vg       */
double    ep_0e[MAX_ELEM];            /* ith strain @ end of span 0         */
double    ep_1e[MAX_ELEM];            /* ith strain @ end of span 1         */
```

```
main()
{
	int		i, j, k,iter,iter_drag;

	double		al0, al1, al2;			/* alpha  of span 0, 1 & 2			*/
	double		ls1, ls2;			/* length of span 1 & 2				*/
	double		v0;				/* base speed					*/
	double		myu;				/* friction coefficient/ web roller		*/
	double		the;				/* wrapping angle (rad)				*/
	double		ep_INg;				/* elastic strain @ span IN			*/
	double		ep_INs;				/* plastic strain @ span IN			*/
	double		ep_INe[MAX_ELEM];		/* i-th  strain @ end of span IN		*/
	double		ep_INet;			/* total strain @ end of span IN		*/
	double		ep_0et;				/* total strain @ end of span 0			*/
	double		ep_1et, ep_2et;			/* those @ end of span 1 & 2			*/
	double		jt, it;				/* J(t) & I(t)  @ end of span 1			*/
	double		t1, t2, tm, dt;			/* total time, temp time, delta time		*/
	double		kx;				/* mass transfer (constant value)		*/
	double		lc;				/* calculated length of the span		*/
	double		er, er_drag;			/* relative error in the iteration		*/
	double		ep[NSUBSPAN+1];			/* strain in the subspan			*/
	double		lx[NSUBSPAN+1];			/* position of subspan in the span		*/
	double		fne(), fni(), fnj();		/* functions for the model			*/
	double		q1;				/* dummy					*/

	al0		= 1.0;		v0		= 1.0;
	myu		= 0.2;		the		= 0.8 * PI;
	al1		= 1.005;	ls1		= 1.2;
	al2		= 1.001;	ls2		= 2.0;

	nelement	= 1;
	gg		= 2.0e9;	ep_INg		= 0.003;
	es		= 5.0e10;	ep_INs		= 0.0005;
	vg[1]		= 1.0e9;	ve[1]		= 5.0e8;

	ep_INet		= ep_INg;
	for ( i=1; i<=nelement; ++i ) {
		ep_INe[i]	= ep_INg * gg / vg[i];
		ep_INet	+= ep_INe[i];
		lm[i]		= ve[i] / vg[i];
	}
	ep_0g = ep_INg;
	ep_0s = ep_INs;
	ep_0et= ep_INet;
	for ( i=1; i<=nelement; ++i ) {
		ep_0e[i] = ep_INe[i];
	}
	kx		= v0  / ( 1.0+ep_INet );
	er_drag		= 999.0;	iter_drag	= 0;
```

```
        while ( (er_drag > CRITERIA) && (++iter_drag < MAX_ITER) ) {
                ep_1et= al1 * (1.0+ep_lNet) - 1.0;
                t1      = 2.0 * ls1 / (al0+al1) / v0;
                er      = 999.0;        iter    = 0;
                while ( (er > CRITERIA) && (++iter < MAX_ITER) ) {
                        jt      = fnj( t1 );                    it      = fni( t1 );
                        sg_1    = ( ep_1et - it ) / jt; ep_1g= sg_1 / gg;
                        dt      = t1 / NSUBSPAN; ql     = 0.0;
                        for ( j=0;j<=NSUBSPAN; ++j ) {
                                tm      = dt * (double) j;
                                ep[j]   = fnj( tm ) * sg_1 + fni( tm );
                                ql      += 2.0 * ( 1.0 + ep[j] );
                        }
                        lc      = ( ql - ep[0] - ep[NSUBSPAN] -2.0 ) * kx * dt / 2.0;
                        t1      = t1 * ls1 / lc;
                        er      = (double) fabs((lc-ls1)/ls1);
                }
                for ( i=0; i<=nelement; ++i ) {
                        fne( t1 );
                        ep_0e[i] = ep_1e[i];
                }
                ep_0s = ep_1s;
                ep_2et= al2 * (1.0+ep_lNet) - 1.0;
                t2      = 2.0 * ls2 / (al1+al2) / v0;
                er      = 999.0;        iter    = 0;
                while ( (er > CRITERIA) && (++iter < MAX_ITER) ) {
                        jt      = fnj( t2 );
                        it      = fni( t2 );
                        sg_2    = ( ep_2et - it ) / jt;
                        ep_2g= sg_2 / gg;
                        dt      = t2 / NSUBSPAN;
                        ql      = 0.0;
                        for ( j=0;j<=NSUBSPAN; ++j ) {
                                tm      = dt * (double) j;
                                ep[j]   = fnj( tm ) * sg_2 + fni( tm );
                                ql      += 2.0 * ( 1.0 + ep[j] );
                        }
                        lc      = ( ql - ep[0] - ep[NSUBSPAN] -2.0 ) * kx * dt / 2.0;
                        t2      = t2 * ls2 / lc;
                        er      = (double) fabs((lc-ls2)/ls2);
                }
                ql              = (double) exp( ((sg_1<sg_2)?-myu:myu) * the ) * sg_2;
                er_drag         = ql / sg_1 - 1.0;
                al1             = al1 * ( 1.0 + er_drag / 500.0 );
                printf( "%2d\t%lf\t%lf\n", iter_drag, er_drag, al1 );
                er_drag         = (double) fabs( er_drag );
        }
        printf( "\n\nValue of al1 = %lf\n\n", al1 );
}
```

PROGRAM 6

```
/*****************************************************m*a*s*a*k*a*z*u*

        output          : strain with respect to time
        web             : elastic,  deterministic
        operation       : dynamic, deterministic
        slip            : none

*****************************************************a*k*a*t*s*u*k*a*/

#include <stdio.h>
#include <math.h>

#define     MAX     100                 /* #calculation                 */
#define     EVR     5                   /* #skip for thedisplay         */
#define     DT      0.005               /* delta time                   */

double      ep0     = 0.0007;           /* initial strain               */
double      l1      = 1.2;              /* length of span 1             */
double      l2      = 0.2;              /* length of span 2             */
double      v0      = 10.0;             /* standard speed of the web    */
double      t       = 0.0;              /* time                         */
double      ep1, ep2;                   /* strain at each span          */

main()
{
        int         i;                  /* counter   */
        double      a1(), a2();         /* alpha1,2  */
        double      nslip();            /* ep1,ep2   */

        ep1     = ( 1.0+ep0 ) * a1(t) - 1.0;
        ep2     = ( 1.0+ep0 ) * a2(t) - 1.0;
        for ( i=0; i<MAX; ++i ) {
                t       = DT * (double) i;
                nslip();
                printf( "%lf\t%lf\t%lf\n", t, ep1, ep2 );
        }
}

double a1()                         /* alpha 1   */
{
        return( (t<10.0*DT)?1.0001:1.001 );
}

double a2()                         /* alpha 2   */
{
        return( 1.0 );
}
```

PROGRAM 7

```
/*********************************************m*a*s*a*k*a*z*u*

        output      : mean & mean sq. of strain with respect to time
        web         : elastic
                    : elastic modulus changes (stochastic)
        operation   : dynamic
                    : deterministic
        slip        : NA

**********************************************a*k*a*t*s*u*k*a*/


#include <stdio.h>
#include <math.h>


#define     OUTPUT   fopen("s/11.o","w")   /* output file name */
#define     DATAO    fopen("sd/1.o","w")   /* crude data output file name          */
#define     PICK     3                     /* #series to be picked up for display  */
#define     PI       3.14159265358979      /* π                                    */
#define     DT       0.005                 /* delta time                           */
#define     EN       5000                  /* #series in the ensemble              */
#define     NC       20                    /* #cross-section to be checked         */
#define     EY       5                     /* #skip for output */
#define     S0       3.0e-10               /* magnitude of the noise               */

double      l1    =  1.2;                  /* length of span 1 */
double      a1    =  1.0005;               /* speed ratio of span1 to span 0       */
double      v0    =  1.0;                  /* standard speed of the web            */
double      e0, e1;                        /* strain at span 0 & 1                 */
```

```
main()
{
	int	i, j, k;				/* counter				*/
	int	idum = -13;				/* seed for unit()			*/
	double unit(), RK4();			/* e0, e1				*/
	double av0[NC], av1[NC];		/* E[x]	of e0 & 1		*/
	double sq0[NC], sq1[NC];		/* E[x^2]	of e0 & 1		*/
	double sig;				/* magnitude of the generated noise	*/
	FILE *fp, *fopen();

	sig = sqrt( S0 * 2.0 * PI );
	e1 = 0.0;
	for ( j=0; j<NC; j++)
		av0[j] = av1[j] = sq0[j] = sq1[j] = 0.0;

	fp = DATAOUT;
	for ( i=0; i<EN; i++ ) {
		for ( j=0; j<NC; j++) {
			for ( k=0; k<EY; k++) {
				e0 = sig * (double) unit( &idum );
				e1 = RK4( e1 );
				if ( i==PICK )
					fprintf( fp, "%f\t%le\t%le\n", DT*(float) (j*EY+k), e0, e1 );
			}
			av0[j] += e0; sq0[j] += e0 * e0;
			av1[j] += e1; sq1[j] += e1 * e1;
		}
	}
	for ( j=0; j<NC; j++) {
		av0[j] /= (double) EN;	sq0[j] /= (double) EN;
		av1[j] /= (double) EN;	sq1[j] /= (double) EN;
	}
	fclose( fp );

	fp = OUTPUT;
	fprintf( fp, "time\tE[e0]\tE[e0^2]\tE[e1]\tE[e1^2]\n" );
	for ( j=0; j<NC; j++ ) {
		fprintf( fp,"%6.3lf\t", (j+1)*DT*EY );
		fprintf( fp,"%10.3le\t%10.3le\t" , av0[j], sq0[j] );
		fprintf( fp,"%10.3le\t%10.3le\n", av1[j], sq1[j]/DT );
	}
	fclose( fp );
}
```

```
double RK4( x )                         /* 4th order Runge Kutta */
double x;
{
        double h, k1, k2, k3, k4;
        double f1();

        h  = DT;
        k1 = h * f1( x );
        k2 = h * f1( x + k1/2.0 );
        k3 = h * f1( x + k2/2.0 );
        k4 = h * f1( x + k3 );
        x  += ( k1 + k2*2.0 + k3*2.0 + k4 ) / 6.0;
        return( x );
}

double f1( x )
double x;
{
        return( ( e0 - a1 * x) * v0 / l1 );
}


/*      function unit() returns a uniformly distributed deviate with zero mean  and unit variance using ran1()
as the source of uniform deviates.                    */

float   unit( idum )
int     *idum;
{
        float x, ran1();

        x  = 2.0 * ran1( idum ) - 1.0;    /*      [-1,1]          */
        x *= sqrt( 3.0 );                 /*      set  variance -> 1
                                                  Suppose this distributes uniformely in the
                                                  interval of [-a,a], then
                                                  E[X^2] = a^2/3, E[X] = 0,
                                                  Var[X] = a^2/3 = 1, thus a = sqrt(3)          */
        return ( x );
}
```

/* function rand1(idun) (from NUMERICAL RECIPES in C [12]) returns a uniform random deviate between 0.0 and 1.0. Argument idum = (any negative number) to initialize or reinitialize the sequence. */

```
#define M1   259200
#define IA1  7141
#define IC1  54773
#define RM1 (1.0/M1)
#define M2   134456
#define IA2  8121
#define IC2  28411
#define RM2 (1.0/M2)
#define M3   243000
#define IA3  4561
#define IC3  51349

float ran1(idum)
int *idum;
{
	static long	ix1,ix2,ix3;
	static float	r[98];
	float		temp;
	static int	iff=0;
	int		j;

	if (*idum <0 || iff==0) {
		iff = 1;
		ix1 = (IC1-(*idum)) % M1;
		ix1 = (IA1*ix1+IC1) % M1;
		ix2 = ix1 % M2;
		ix1 = (IA1*ix1+IC1) % M1;
		ix3 = ix1 % M3;
		for(j=1; j<=97; j++) {
			ix1 = (IA1*ix1+IC1) % M1;
			ix2 = (IA2*ix2+IC2) % M2;
			r[j] = (ix1+ix2*RM2) * RM1;
		}
		*idum=1;
	}

	ix1 = (IA1*ix1+IC1) % M1;
	ix2 = (IA2*ix2+IC2) % M2;
	ix3 = (IA3*ix3+IC3) % M3;
	j=1 + ((97*ix3)/M3);
	if (j > 97 || j< 1) printf(" RAN1: This cannot happen.");
	temp = r[j];
	r[j] = (ix1+ix2*RM2) * RM1;

	return temp;
}
```

PROGRAM 8

```
/**************************************************m*a*s*a*k*a*z*u*

        output          : chi-square "GOODNESS of FIT" test
        notes           : original author of this program is unknown.

 ***************************************************a*k*a*t*s*u*k*a*/


#include "stdio.h"
#include "math.h"

#define POINTS    5000              /* maximum number of data points          */
#define N         12                /* number of intervals                    */
#define BIGNUM    1.0e6             /* BIGNUM should be > largest data        */
#define STDZ      0.4               /* width of intervals in fraction of z    */
#define CHI_TBL   5.38              /* chi^2 for 9 dof and alpha = 0.80       */
#define OUTPUT    stdout            /* Filename for the output                */


main()
{
        float   x[POINTS];          /* data points                            */
        float   x_upper[N];         /* upper bound of the data intervals      */
        float   p[N];               /* probability for the upper boundary     */
        float   F[N];               /* expected   freq. in each interval      */
        float   f[N];               /* actual     freq. in each interval      */
        float   chi, diff, dum;
        float   mean, vari;         /* mean, variance                         */
        float   convert(), ran1();  /* ran1(): unit deviates                  */

        int     i, j;
        int     idum = -13;
        FILE    *fp, *fopen();

        chi     = diff = mean = vari = 0.0;
        for( i=0; i<N; ++i )
                p[i] = F[i] = f[i] = 0.0;

        for( i=0; i<POINTS; ++i ) {
                dum     = ran1( &idum );
                x[i]    = dum;
                dum     = convert( dum );
                mean    += dum;
                vari    += dum * dum;
        }
        mean    /= POINTS;
        vari    /= POINTS;          vari -= mean * mean;
```

```
        statu( x_upper, p );                    /* Obtain variables for Chi^2 Table             */

        for( i=0; i<POINTS; ++i ) {
                for( j=0; j<N-1; ++j )
                        if( x_upper[j] < x[i] && x[i] <= x_upper[j+1] )  f[j+1] += 1.0;
                if( x[i] <= x_upper[0] )  f[0] += 1.0;
        }

        fp = OUTPUT;
        fprintf( fp, "mean = \t%f\n", mean );
        fprintf( fp, "variance = \t%f\n\n", vari );
        fprintf( fp, "No.\texpect\tactual\tdiff.\n" );
        for( j=0; j<N; ++j ) {
                F[j] = POINTS * p[j];
                diff = (F[j] - f[j]) * (F[j] - f[j]) / F[j];
                chi += diff;
                fprintf( fp, "%2d\t%6.3f\t%6.3f\t%6.4f\n",j, F[j], f[j],diff );
        }
        fprintf( fp, "\n\n" );

        fprintf( fp, "The calculated value of X^2    is %6.2f\n\n",chi );
        fprintf( fp, "The chi^2 value from the table is %6.2f\n\n",CHI_TBL );
        fprintf( fp, "The hypothesis of homogeneity  is " );
        if(chi > CHI_TBL) fprintf( fp, "not " );
        fprintf( fp, "accepted\n" );
        fclose( fp );
}


statu( x_upper, p )                             /* for unit deviates */
float  x_upper[], p[];
{
        int i;

        for( i=0; i<N; ++i ) {
                x_upper[i]      = 1.0 / ((float) N) * (float) (i+1);
                p[i]            = 1.0 / ((float) N);
        }
        return;
}


float convert( x )                              /* from unit deviates to zero mean unit variance */
float x;
{
        return ( (2.0 * x -1.0) * sqrt(3.0) );
}
```

PROGRAM 9

```
/**************************************************m*a*s*a*k*a*z*u*

	output		: auto-correlation function
				R[i] = Sigma[ d[j][0] * d[j][i] ]/ NO_ENSEMBLE
				for (i=0 to MAX_DATA).

**************************************************a*k*a*t*s*u*k*a*/


#include "stdio.h"
#include "math.h"

#define MAX_DATA		11		/* #data in each series			*/
#define NO_ENSEMBLE		500		/* #series for an ensemble		*/


main()
{
	int	i, j;
	int	idum  = -13;			/* seed for the generator ran1()	*/
	float	unit();
	float	d[NO_ENSEMBLE][MAX_DATA], r, r0;

	for ( j=0; j<NO_ENSEMBLE; ++j ) {
		for( i=0; i<MAX_DATA; ++i )
			d[j][i] = unit( &idum );
	}

	for ( i=0; i<MAX_DATA; ++i ) {
		r  = 0.0;
		for ( j=0; j<NO_ENSEMBLE; ++j )
			r += d[j][0] * d[j][i];
		r /= NO_ENSEMBLE;
		if (i==0) r0 = r;
		printf( "%d\t%f\n", i, r/r0 );
	}
}
```

PROGRAM 10

```
/****************************************************m*a*s*a*k*a*z*u*

        output          : mean and variance of the random data series

****************************************************a*k*a*t*s*u*k*a*/


#include "stdio.h"

#define PNT  500000
#define OUT  stdout


main()
{
        double  mean, vari, dum;
        float           unit();
        long int        i;
        int             idum = -13;
        FILE            *fp, *fopen();

        mean = vari = 0.0;
        for( i=0; i<PNT; ++i ) {
                dum     = (double) unit( &idum );
                mean  += dum;
                vari    += dum * dum;
        }
        mean  /= (double) PNT;
        vari    /= (double) PNT;     vari -= mean * mean;

        fp = OUT;
        fprintf( fp, "mean = \t%le\n", mean );
        fprintf( fp, "variance = \t%le\n\n", vari );
        fclose(fp);
}
```

2

VITA

Masakazu Akatsuka

Candidate for the Degree of

Master of Science

Thesis : MODELS OF LONGITUDINAL WEB BEHAVIOR IN THE HANDLING PROCESS

Major Field : Mechanical Engineering

Biographical :

Personal Data : Born in Takarazuka, Japan, February 18, 1958, the son of Tsuneo and Kikuko Akatsuka.

Education : Received Bachelor of Engineering from Kyoto University in March 1982; completed requirements for the Master of Science degree at Oklahoma State University in July, 1990.

Professional Experience : Research Engineer, Mitsubishi Heavy Industries, ltd., Tokyo, Japan, 1982 - present.