Tutorial: Simulink® Model for 1D Phase Transformation Kinetics of Shape Memory Alloys with Thermal Effects

Vagner Candido de Sousa

lattes.cnpq.br/5679124885526239

researcherid.com/rid/K-7099-2012

scholar.google.com.br/citations?user=fLrmfTsAAAAJ

Department of Aeronautical Engineering

São Carlos School of Engineering

University of São Paulo

São Carlos – SP, Brazil, 2014

ABSTRACT

A model in MATLAB/Simulink for the simulation of 1D phase transformation kinetics of shape memory alloys (SMAs) is presented. The transformation kinetics related to pseudoelasticity (and to the shape memory effect above the martensite start temperature) is based on Brinson's model. A strategy for detecting the onset of phase transformations and redefining the initial conditions as soon as a new transformation is detected is included. No such a strategy is given in the original model and it helps in the representation of more complex thermomechanical loadings. A heat transfer model is included to represent thermal effects such as Joule heating, natural convection, and latent heat associated with phase transformation. The pertinent theory is stated along with the model implementation. The entire model and supporting source-code are provided. An example of use is given and followed by several analyses which include parametric studies, thermal boundary condition studies, and comparisons between simplified assumptions (such as isotherm and adiabatic processes) and more realistic approaches (which account for the latent heat phenomenon). The effect of latent heat on SMA temperature and therefore on the critical stresses for phase transformations is discussed. It is exemplified that depending on the thermomechanical conditions, phase transformations may not complete, what could yield in unexpected behavior or unsatisfactory performance of an SMA-based structure. Finally, the effect of different electrical resistances between the martensitic and austenitic phases is illustrated for a case of Joule heating.

1. INTRODUCTION

Shape memory alloys (SMAs), such as the quasi-equiatomic nickel-titanium alloys, can recover relatively large inelastic strains through heating. Two major effects of SMAs are the shape memory effect and the pseudoelastic effect. In the shape memory effect, the inelastic strain occurs at a relatively low temperature (below the temperature at which the austenitic phase is stable – the austenite finish temperature) and the original (undeformed) shape is recovered by increasing the SMA temperature above the austenite finish temperature. Since the shape recovery takes place at a desired time (when temperature is increased), the shape memory effect is commonly exploited in SMA-based actuators. In the pseudoelastic effect, the inelastic strain occurs at a relatively high temperature (above the austenite finish temperature) and the original shape is recovered while the deforming force is removed.

SMAs have drawn growing attention for the solution of problems in many areas. SMAs show up in recent literature besides piezoceramics, magnetorheological and electrorheological fluids for active control of civil structures (Song et al., 2006; Korkmaz, 2011). A comprehensive review on current research, applications and opportunities (with over 100 SMA-based state-of-the-art patents) is presented in Jani et al. (2014). One can enumerate biocompatibility (Machado and Savi, 2003; Elahinia et al., 2012), mechanical energy dissipating capability (Ibrahim, 2008; Sousa and De Marqui, 2014), scalability (Nespoli et al., 2010), and higher ratio of actuation power to actuator mass (Lagoudas, 2008) among the most remarkable characteristics of SMAs. The latter is among the numberless properties which make SMAs suitable for actuation problems. SMA-based actuators have been considered as an alternative to traditional mechanisms such as motors or hydraulic actuators. Several recent researches review SMA actuators for adaptive (morphing) aircraft applications (Bil et al., 2013; Barbarino et al., 2014; Ko et al., 2014). Miniature SMA-actuated robots with climbing, crawling and jumping abilities (e.g. aiming in-pipe inspection) have been reported (Kheirikhah et al., 2011). SMA-based robotic modules can be attached together in chain or lattice configurations to form custom constructions and have been considered for position

control with multiple degrees of freedom (Hadi et al., 2010). The development of novel, low-cost, five-fingered prosthetic hand with SMA actuators is reported in Andrianesis and Tzes (2014). Grasp experiments with common objects revealed the potential of the proposed prosthetic hand and improvements promoted by the SMA actuators regarding size, weight and noise when compared to the most advanced commercial devices.

However, SMAs usually exhibit slow response (related to heating and cooling times) in addition to strongly nonlinear behaviors such as hysteresis. Furthermore, thermal effects related to phase transformation can modify the hysteretic response of SMAs. SMAs release energy during forward transformation and absorb energy during reverse transformation (Lagoudas, 2008). This latent heat phenomenon induces temperature variations in SMAs, and since phase transformations depend on definite thermomechanical conditions to occur, such temperature variations should be considered in control strategies for satisfactory actuation performance (Elahinia and Ahmadian, 2005a, 2005b). In problems in which the phase transformation rate (related to actuation) is low, the heat exchange with the environment can be able to compensate the temperature variations by natural convection. On the other hand, when fast actuation is required, convection (or external heating) may be unable to lead the SMA temperature back to its initial (reference) value before the next actuation cycle. In such a circumstance, the SMA may undergo incomplete transformations and deteriorate the actuator performance. Incomplete transformations may yield in partial shape recovery and hence smaller strokes. Concerning the importance of temperature variations for the modeling of SMA applications, this paper considers a well-known phase transformation model (Brinson, 1993) and a model for heat transfer (Faulkner et al., 2007; Hadi et al., 2010) to provide a useful basis for the simulation of SMA phase transformation kinetics with thermal effects.

This paper describes the implementation of a model in MATLAB/Simulink® for the simulation of the phase transformation kinetics related to the pseudoelastic effect of SMAs (and shape memory effect above the martensite start temperature). The development of the model is presented in a “block-by-block” approach and thus the reader can obtain a library convenient for

reutilization. Brinson's model (Brinson, 1993) is followed to represent the SMA phase transformation behavior. For brevity, only the austenitic phase and stress-induced martensitic phase are considered (which are enough to represent the pseudoelastic effect). Nevertheless, the reader should be able to modify the presented model if the representation of self-accommodated (low temperature-induced) martensite is also desired. This paper also contributes with a strategy for both detecting the onset of phase transformations and redefining the initial conditions (i.e. the initial martensitic fraction) as soon as a new phase transformation is detected. No such a strategy is included in the original model and it can help in the representation of more complex thermomechanical loadings, which may involve, for example, changes in stress and temperature and arbitrarily resumed or reversed phase transformations (common in actuation problems). Additionally, a heat transfer model which accounts for Joule heating, natural convection and latent heat phenomenon is provided. Latent heat is important in SMA modeling because it affects the SMA temperature during phase transformations. Consequently, the temperature-dependent hysteretic behavior of the SMA can deviate from the expected behavior and yield in (e.g.) unsatisfactory performance of an SMA-based structure. The presented model considers the critical stresses as function of temperature. Joule heating is assumed for controlling temperature and the electrical resistance of the SMA is expressed as function of the martensitic fraction. The model can represent several aspects of the phase transformation kinetics of SMAs and allows parametric studies, thermal boundary conditions analyses (isothermal, adiabatic, with or without latent heat effect), investigation of electrical activation, and so on. After the description of all blocks, examples are given. The examples address the general behavior of SMA phase transformation kinetics. Thermal effects are exploited to show differences between simplified and more realistic approaches for heat transfer phenomena.

2. MODEL DEVELOPMENT

This paper follows Brinson (1993)'s model, which is a well-established phenomenological model for the representation of the thermomechanical behavior of SMAs under uniaxial loading. Brinson's model is able to represent the shape memory effect (either below or above the martensite start temperature) and the pseudoelastic effect. Due to its simplicity and yet powerful representation of 1D SMA behavior, it is among the most used and extended models in literature.

First, low-level blocks (or subsystems) are presented. Such blocks compute the critical stresses for the onset and completion of phase transformations (forward and reverse), the martensitic fractions and provide information on whether a transformation is occurring. Then, high-level blocks are presented. Such blocks receive information from low-level blocks and determine the appropriate output. Lastly, a heat transfer block is presented. Such a block can be used for the simulation of non-isothermal conditions. The blocks are "masked" to provide a convenient user interface for entering material parameters. Physical quantities such as mechanical stress, temperature and voltage come from input ports and thus one can specify custom input values (e.g. time-varying signals given by ramp or oscillatory functions).

2.1. Shape Memory Alloy Phase Transformation Kinetics

2.1.1. Critical Stresses and Temperatures

A remarkable point of SMAs is their (microscopic and macroscopic) reversible changes associated with phase transformation. Phase transformations can be induced by low-temperature (resulting in self-accommodated martensite), by high-temperature (resulting in austenite) or by mechanical stress (resulting in reoriented martensite). In practice, phase transformations are related to combinations of temperature and stress, as shown in Figure 1.

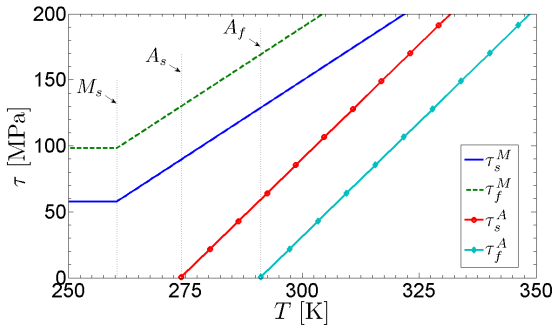


Figure 1: Stress-temperature relationship.

In Figure 1, the curve σ_s^M is the critical value of stress for the onset of stress-induced phase transformation (or forward transformation), σ_f^M is the critical stress for the completion of stress-induced transformation, σ_s^A is the stress value for the onset of the recovery of the austenitic phase (or reverse transformation), σ_f^A is the stress value at which the austenitic phase is completely recovered. Such critical stresses are given by,

$$\sigma_s^M = \sigma_s^{\min} + C_M \left(T - M_s\right) \tag{1}$$

$$\sigma_f^M = \sigma_f^{\min} + C_M \left(T - M_s\right) \tag{2}$$

$$\sigma_s^A = C_A \left(T - A_s\right) \tag{3}$$

$$\sigma_f^A = C_A \left(T - A_f\right) \tag{4}$$

where $\sigma_s^{\min}$ and $\sigma_f^{\min}$ are minimum stresses at which stress-induced transformation begins and completes, respectively; C_M and C_A represent the influence of temperature on critical stresses for

forward and reverse transformations, respectively (the slopes of the curves in Figure 1); M_s is the martensite start temperature, A_s is the austenite start temperature and A_f is the austenite finish temperature; T is the SMA temperature.

Figure 2(a) to (d) shows the subsystems which represent the critical stresses.

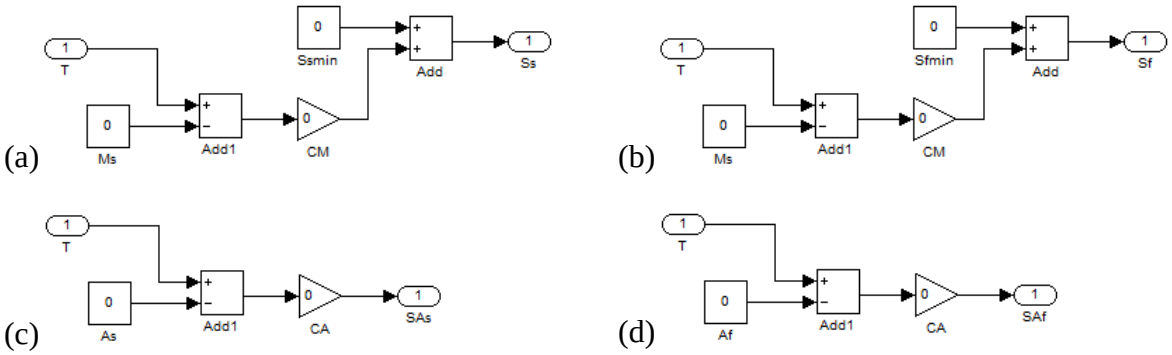


Figure 2: Subsystems related to critical stresses for phase transformations: (a) *StressStartAM* for σ_s^M, (b) *StressFinishAM* for σ_f^M, (c) *StressStartMA* for σ_s^A and (d) *StressFinishMA* for σ_f^A.

Material parameters (which usually do not vary during a simulation) are entered through masks. A mask is a set of dialog parameters which are associated to variables. Table 1 to 4 shows the respective mask of each block shown in Figure 2(a) to (d).

Prompt	**Variable**
Martensitic stress-temperature slope	varCM
Minimum martensite stress start	varSsmin
Martensite start temperature	varMs

Table 1: Mask parameters for *StressStartAM* block of Figure 2(a), which represents σ_s^M.

Prompt	**Variable**
Martensitic stress-temperature slope	varCM

Minimum martensite stress finish	varSfmin
Martensite start temperature	varMs

Table 2: Mask parameters for *StressFinishAM* block of Figure 2(b), which represents σ_f^M .

Prompt	**Variable**
Austenitic stress-temperature slope	varCA
Austenite start temperature	varAs

Table 3: Mask parameters for *StressStartMA* block of Figure 2(c), which represents σ_s^A.

Prompt	**Variable**
Austenitic stress-temperature slope	varCA
Austenite finish temperature	varAf

Table 4: Mask parameters for *StressFinishMA* block of Figure 2(d), which represents σ_f^A .

Initialization commands (in Mask Editor) can be used when values are changed through a mask. Block for σ_s^M (*StressStartAM* of Figure 2(a)) runs the following instruction when Ok (or Apply) button is clicked,

```
StressStartAM_InitMask(gcb);
```

where such a user function is given in Appendix. Similarly, blocks for σ_f^M , σ_s^A and σ_f^A (in Figure 2(b) to (d)) run, respectively,

```
StressFinishAM_InitMask(gcb);
```

```
StressStartMA_InitMask(gcb);
```

```
StressFinishMA_InitMask(gcb);
```

which are given in Appendix as well.

The temperature at which the austenitic phase begins to stabilize (the austenite start temperature) in the presence of applied stress, accordingly to Figure 1, is,

$$A_s^{\sigma} = A_s + \frac{\sigma}{C_A} \tag{5}$$

and the corresponding block is shown in Figure 3.

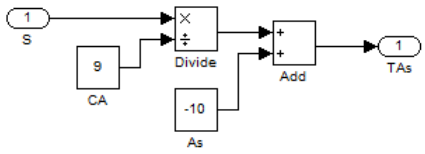


Figure 3: *TempStartMA* block, which computes the critical temperature of start of shape recovery in the presence of applied stress, A_s^{σ}.

The mask of *TempStartMA* block is given in Table 5.

Prompt	**Variable**
Austenitic stress-temperature slope	varCA
Austenite start temperature	varAs

Table 5: Mask parameters for *TempStartMA* block of Figure 3, which represents A_s^{σ}.

The initialization command of *TempStartMA* block is,

```
TempStartMA_InitMask(gcb);
```

which is given in Appendix.

2.1.2. Martensite Volumetric Fraction

The martensitic fraction represents the amount of transformation, varying from zero (fully austenite) up to unity (fully martensite). The martensitic fraction increases during forward transformation, decreases during reverse transformation and remain unchanged when no transformation is occurring.

The martensitic fraction in a forward transformation (from Brinson's model) is given by

$$\xi^{A\to M} = \frac{1-\xi_0}{2}\cos\left[\frac{\pi}{\sigma_s^{\min}-\sigma_f^{\min}}\left(\sigma-\sigma_f^{M}\right)\right]+\frac{1+\xi_0}{2} \qquad (6)$$

where ξ_0 is the initial martensitic fraction.

MartFracAM block represents Equation (6) and is shown in Figure 4.

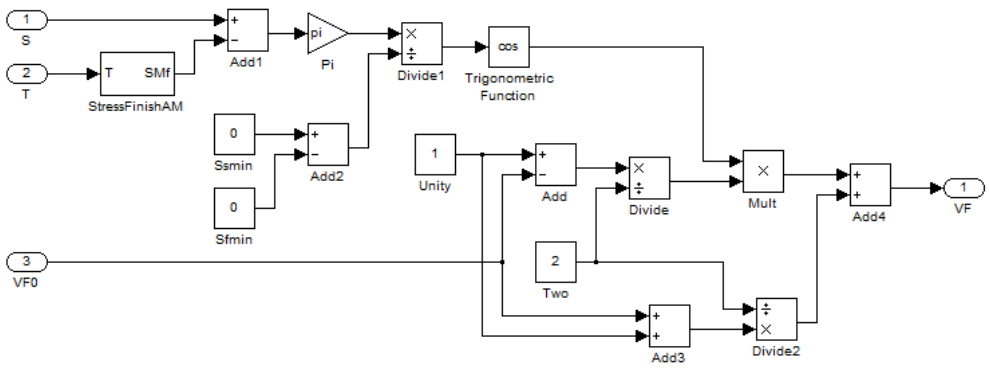


Figure 4: *MartFracAM* block. VF stands for (martensite) volumetric fraction.

MartFracAM block's mask is given in Table 6.

Prompt	Variable
Minimum martensite stress start	varSsmin
Minimum martensite stress finish	varSfmin
Martensitic stress-temperature slope	varCM
Martensite start temperature	varMs

Table 6: Mask of *MartFracAM* block.

Initialization command of *MartFracAM* block is

```
MartFracAM_InitMask(gcb);
```

which is given in Appendix.

The martensite fraction in a reverse transformation is given by,

$$\xi^{\mathrm{M}\to\mathrm{A}} = \frac{\xi_0}{2}\left\{\cos\left[\frac{\pi}{A_f - A_s}\left(T - A_s^{\sigma}\right)\right] + 1\right\} \tag{7}$$

MartFracMA block represents Equation (7) and is shown in Figure 5.

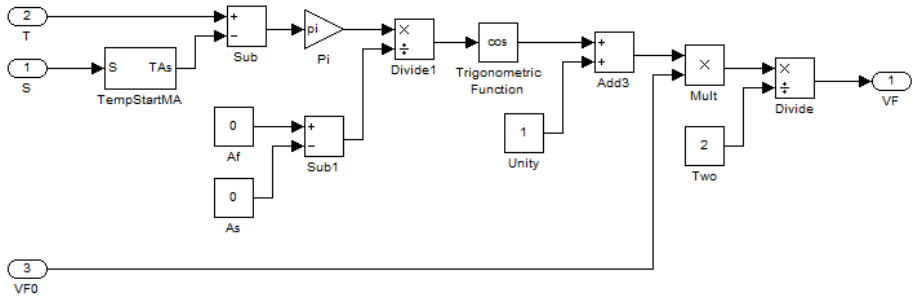


Figure 5: *MartFracMA* block.

MartFracMA block's mask is given in Table 7.

Prompt	**Variable**
Austenitic stress-temperature slope	varCA
Austenite start temperature	varAs
Austenite finish temperature	varAf

Table 7: Mask of *MartFracMA* block.

Initialization command of *MartFracMA* block is

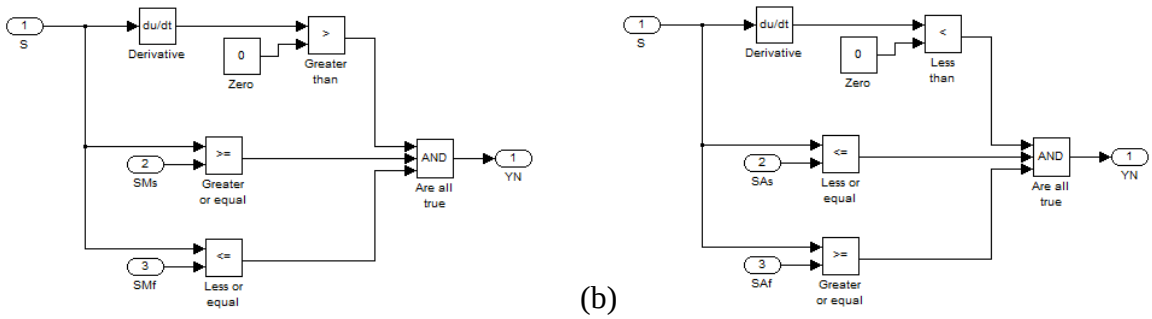
```
MartFracMA_InitMask(gcb);
```

which is given in Appendix.

2.1.3. Identification of Ongoing Phase Transformations

A pertinent point in simulating SMAs is identifying whether a phase transformation is occurring and, if so, what transformation (forward or reverse) it is. Forward transformations are related to mechanical loading and occur when the applied stress increases from σ_s^M up to σ_f^M. Reverse transformations are related to mechanical unloading and occur when the applied stress

decreases from σ_s^A to σ_f^A. Figures 6(a) and (b) show the blocks which identify forward and reverse transformations, respectively. Such blocks are straightforward and outputs a logical *true* (i.e. unity) when the stress input is in the respective critical range and the stress derivative is positive (for forward transformation) or the stress derivative is negative (for reverse transformation). Otherwise these blocks output a logical *false* (i.e. zero).



(a) (b)

Figure 6: Blocks for detecting ongoing (a) forward and (b) reverse transformations. YN stands for *yes* (logical *true*) or *no* (logical *false*).

2.1.4. Identification of the Onset of Phase Transformations and Redefinition of the Initial Conditions

Another important point in simulating SMAs is defining the initial conditions adequately (which include the current amount of martensite). The initial conditions are set at the beginning of each transformation. In common instances it is enough to set only the initial martensitic fraction with the value of the martensitic fraction at the previous time step of simulation ($\xi_0 \leftarrow \xi(t-\Delta t)$). Figure 7 shows *SMA_SelectVF* block, which performs this task.

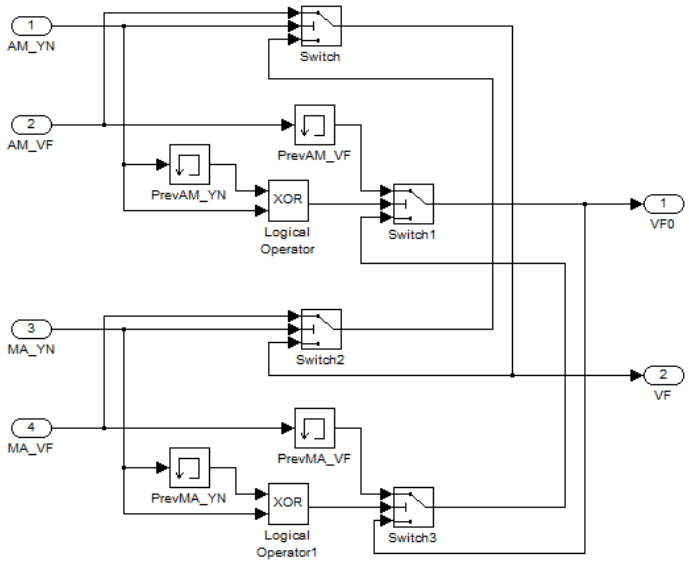


Figure 7: *SMA_SelectVF* block.

The logical operator XOR (Exclusive OR) outputs *true* when its operands are different from each other* [footnote: The XOR logic is used instead of NAND logic here because XOR detects both the onset and end of phase transformations, which may be interesting to the user]. Thus the onset of a forward transformation is identified if AM_YN signal at the current step is *true* and it was *false* at the previous time step (XOR(AM_YN(1), PrevAM_YN(0)) = 1). A *true* signal which outputs from AM_XOR block leads *Switch1* to output the value at its first input port (the martensitic fraction obtained from *MartFracAM* block at the previous time step, PrevAM_VF) to outside *SMA_SelectVF* block. The output of *Switch1* is also sent to the third input of *Switch3* block (*Switch3* is related to the counterpart logic for identifying the onset of a reverse transformation). Additionally, if AM_YN signal is *true*, then *Switch* outputs the value at its first input port (the martensitic fraction obtained from *MartFracAM* at the current time step). *Switch's* output goes outside *SMA_SelectVF* block and to the third input port of *Switch2* block (*Switch2* is also related to reverse transformation). At the next time step (assuming usual behavior), AM_XOR will be *false* (that is no longer the onset of the transformation) and *Switch1* will output the initial martensitic

fraction whereas *Switch* will output new values for the martensitic fraction until the transformation ends.

The same algorithm applies to reverse transformations. This assures both blocks related to computation of martensitic fraction (*MartFracAM* and *MartFracMA*) to know the correct value of the initial martensitic fraction and only the martensitic fraction related to the transformation which is indeed occurring is routed outside *SMA_SelectVF* block. However, due to the algebraic loops of *SMA_SelectVF* block, Simulink's solver may be unable to converge in some instances. In such a situation the manual definition of the maximum integration step size and the choice of a different solution method (in particular the more appropriates for higher-stiffness systems, such as *ode23tb*) may be required.

In terms of Boolean algebra, Table 8 represents the working (truth table) of the subsystem of Figure 7. For convenience, signals *AM_YN*, *PrevAM_YN*, *MA_YN* and *PrevMA_YN* are denoted by A, B, C and D, respectively. As well, switch blocks *Switch*, *Switch1*, *Switch2* and *Switch3* are denoted by *Sw*, *Sw1*, *Sw2* and *Sw3*. For demonstration purposes, it is assumed a process in which a forward phase transformation begins, proceeds for a few iterations and stops; then a reverse transformation takes place, proceeds for a few iterations and stops as well.

	Inputs				**Decisions**		**Outputs**			
#	**A**	**B**	**C**	**D**	**A ⊕ B**	**C ⊕ D**	**Sw**	**Sw1**	**Sw2**	**Sw3**
1	1	0	0	0	1	0	AM_VF	PrevAM_VF	Sw	Sw1
2	1	1	0	0	0	0	AM_VF	Sw3	Sw	Sw1
⋮	1	1	0	0	0	0	AM_VF	Sw3	Sw	Sw1
i	0	1	0	0	1	0	Sw2	PrevAM_VF	Sw	Sw1
⋮	0	0	0	0	0	0	Sw2	Sw3	Sw	Sw1
j	0	0	1	0	0	1	Sw2	Sw3	MA_VF	PrevMA_VF
⋮	0	0	1	1	0	0	Sw2	Sw3	MA_VF	Sw1
k	0	0	0	1	0	1	Sw2	Sw3	Sw	PrevMA_VF
⋮	0	0	0	0	0	0	Sw2	Sw3	Sw	Sw1

Table 8: Truth table and the corresponding output signals for subsystem of Figure 7.

2.1.5. Outputting the Martensitic Fraction

SMA_TransformingState block (in Figure 8) uses *IsForward*, *IsReverse* and *SMA_SelectVF* blocks for routing the appropriate input signal to the output. This block has no mask.

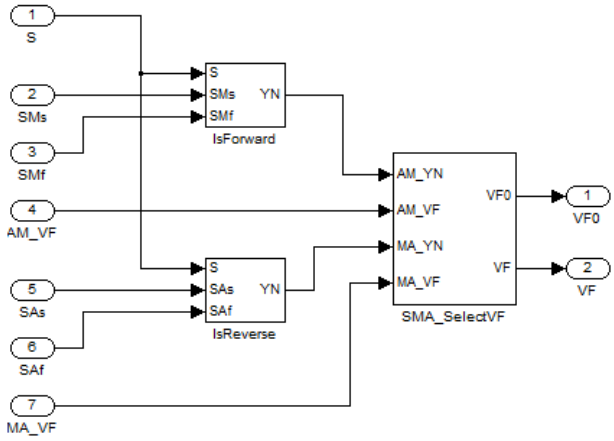


Figure 8: *SMA_TransformingState* block.

2.1.6. Representation of the Phase Transformation Kinetics

SMA_PhaseTransf_Sin block (in Figure 9) receives stress and temperature inputs and outputs the martensitic fraction. In short, this block represents the 1D transformation kinetics of SMAs (the physical counterpart of this block can be the internal state of an SMA wire specimen).

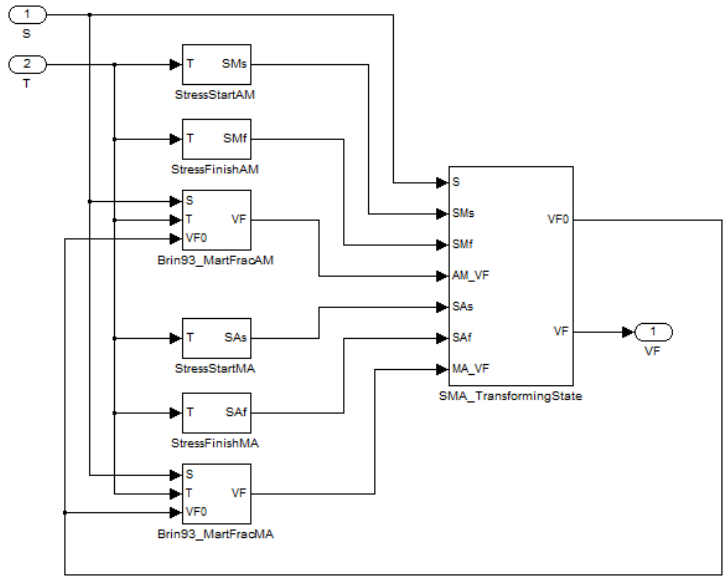


Figure 9: *SMA_PhaseTransf_Sin* block.

SMA_PhaseTransf_Sin block is masked for entering material parameters. Table 9 shows the mask of this block.

Prompt	**Variable**
Martensitic stress-temperature slope	varCM
Austenitic stress-temperature slope	varCA
Minimum martensite stress start	varSsmin
Minimum martensite stress finish	varSfmin
Martensite start temperature	varMs
Austenite start temperature	varAs
Austenite finish temperature	varAf

Table 9: Mask of *SMA_PhaseTransf_Sin* block.

The initialization function of *SMA_PhaseTransf_Sin* block is

```
SMA_PhaseTransf_Sin_InitMask(gcb);
```

which is given in Appendix.

2.2. Thermal Model

Temperature is a major quantity for consideration in designing an SMA application. Especially for pseudoelasticity, isothermal assumptions are often non-realistic due to higher strain rates. The SMA response (its hysteretic stress-strain behavior) strongly depends on temperature and temperature is very sensitive to perturbation. In addition to modeling convective heat transfer and forced heating (e.g. by Joule heating through electrical current), the inclusion of the latent heat phenomenon related to phase transformation contributes to more realistic representation of SMAs. Forward transformations are exothermic and reverse transformations are endothermic, yielding in temperature changes during loading and unloading processes (Lagoudas, 2008).

From the principle of conservation of energy, the heat transfer model considered here is given by (Faulkner et al., 2000; Hadi et al., 2010),

$$mC_p\dot{T} = \frac{V^2}{R(\xi)} + mL\dot{\xi} - h_c A_c \left(T - T_\infty\right) \tag{8}$$

where m is mass, C_p is specific heat, V is voltage, R is electrical resistance, L is latent heat, h_c is convection coefficient, A_c is the circumferential area of convection, T_∞ is ambient temperature and overdot denotes time derivative.

Equation (8) is represented by *SMA_HeatTransfer* block shown in Figure 10.

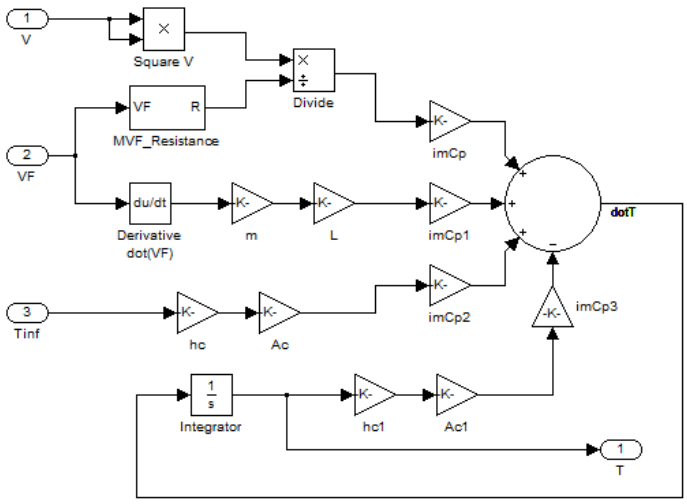


Figure 10: *SMA_HeatTransfer* block.

SMA_HeatTransfer block's mask is given in Table 10.

Prompt	**Variable**
Mass	varm
Latent heat	varL
Specific heat	varCp
Fully-austenitic electrical resistance	varRA
Fully-martensitic electrical resistance	varRM
Convective heat transfer coefficient	varhc
Circumferential area	varAc
Initial temperature	varT0

Table 10: Mask of *SMA_HeatTransfer* block.

The initialization command of *SMA_HeatTransfer* block is,

```
SMA_HeatTransfer_InitMask(gcb);
```

which is given in Appendix.

Last, the martensite-dependent electrical resistance (in Equation (8)) is given by,

$$R(\xi) = R_A + \xi(R_M - R_A) \tag{9}$$

where R_A and R_M are the electrical resistances of the fully-austenitic and fully-martensitic phases, respectively. The block corresponding to Equation (9) is shown in Figure 11.

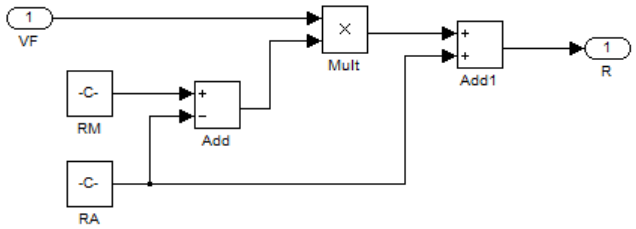


Figure 11: *MVF_Resistance* block.

MVF_Resistance block's mask is given in Table 11.

Prompt	**Variable**
Fully-austenitic electrical resistance	varRA
Fully-martensitic electrical resistance	varRM

Table 11: Mask of *MVF_Resistance* block.

The initialization command of *MVF_Resistance* block is,

```
MVF_Resistance_InitMask(gcb);
```

which is given in Appendix.

3. EXAMPLES

This section presents examples. A simple model is shown in Figure 12. SMA material parameters are given in Table 12 (Brinson, 1993). Thermal parameters are given in Table 13 (Hadi et al., 2010). Solution method *ode23tb* with maximum step size of 10^{-4} s is used. Ambient temperature is set at $23\,°\mathrm{C}$. The mechanical loading is represented by a stress input given by a sine function and peak value of $600\,\mathrm{MPa}$. Considering *a priori* only pseudoelasticity, $T \geq A_f$ and therefore $\sigma_s^M \geq 350\,\mathrm{MPa}$, $\sigma_f^M \geq 415\,\mathrm{MPa}$, $\sigma_s^A \geq 200\,\mathrm{MPa}$ and $\sigma_f^A \geq 0\,\mathrm{MPa}$ (from Equation (1) to (4) and Table 12). However, there exist instances in which the SMA temperature undergoes expressive drops (such that $T < A_f$) and yield in especial forms of the shape memory effect since non-zero martensitic fractions are verified after complete mechanical unloading.

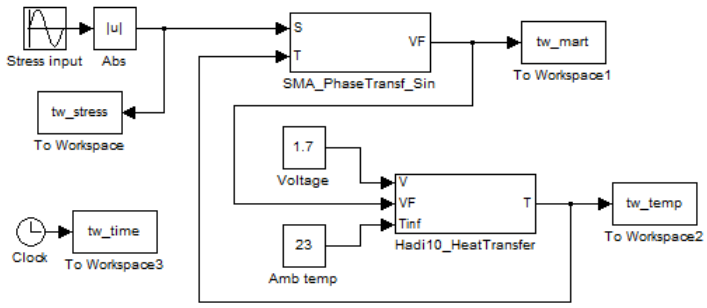


Figure 12: A model for investigating SMA phase transformation kinetics with thermal effects.

Parameter	Symbol	Value
Martensite start temperature	M_s	$18.4\,°\mathrm{C}$
Austenite start temperature	A_s	$34.5\,°\mathrm{C}$
Austenite finish temperature	A_f	$49\,°\mathrm{C}$
Martensitic stress-temperature slope	C_M	$8\,\mathrm{MPA}\cdot°\mathrm{C}^{-1}$
Austenitic stress-temperature slope	C_A	$13.8\,\mathrm{MPA}\cdot°\mathrm{C}^{-1}$
Minimum martensite stress start	$\sigma_s^{\min}$	100 MPA
Minimum martensite stress finish	$\sigma_f^{\min}$	170 MPA

Table 12: SMA parameters used in the phase transformation model (Brinson, 1993).

Parameter	Symbol	Value
Mass	m	1.18×10^{-3} kg
Specific heat	C_p	$350\ \mathrm{J\cdot kg^{-1}\cdot{}^{\circ}C^{-1}}$
Latent heat	L	$6025\ \mathrm{J\cdot kg^{-1}\cdot{}^{\circ}C^{-1}}$
Austenitic resistance	R_A	$0.7246\ \Omega$
Martensitic resistance	R_M	$0.8197\ \Omega$
Convection coefficient	h_c	$150\ \mathrm{J\cdot m^{-2}\cdot{}^{\circ}C^{-1}\cdot s^{-1}}$
Circumferential area	A_c	9.77×10^{-4} m^2

Table 13: SMA parameters used in the thermal model (Hadi et al., 2010).

3.1. Parametric Studies

A few possible parametric studies performed with the presented model are illustrated below. Figure 13 illustrates thermal activation through Joule heating with increasing voltage. Figure 14(a) illustrates temperature changes (due to the latent heat phenomenon) with cyclic loading-unloading processes for increasing amount of phase transformation at a fixed loading frequency. Figure 14(b) illustrates temperature changes with cyclic loading-unloading processes for increasing loading frequency at a fixed amount of transformation.

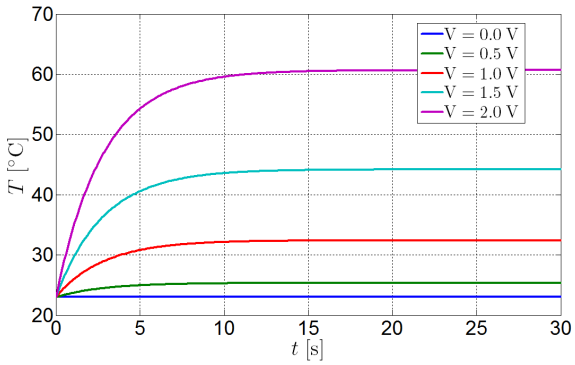
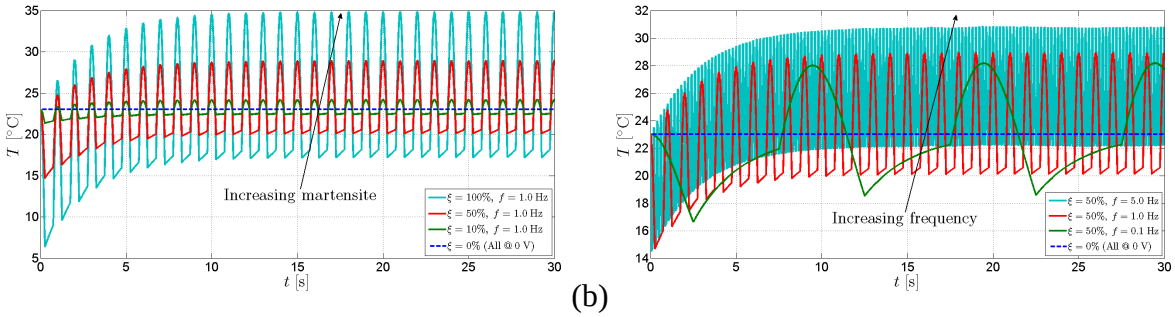


Figure 13: Example of parametric studies: thermal activation through Joule heating with increasing voltage.



(a) (b)

Figure 14: Example of parametric studies: effect of (a) amount of phase transformation and (b) loading frequency.

3.2. Thermal Boundary Condition Analyses

Different thermal boundary conditions are considered for $T = 50\,°\text{C}$. Isothermal and adiabatic processes are illustrated along with a parametric study concerning the influence of increasing convective heat transfer coefficient (Figure 15) and latent heat (Figure 16). Since higher convection coefficients are more favorable to heat exchange with the surroundings, the energy released on forward transformation is faster dissipated to the environment and when is later required by the reverse transformation, the SMA temperature drops below its initial value (Tabesh et al., 2012). This is evident in Figure 15 when $\sigma = \sigma_f^A$, where $\xi > 0$ because $T < A_f$. Additionally, the maximum achieved temperature decreases with increasing convection coefficient. On the other hand, higher latent heat yields in higher SMA temperature due to forward transformation. As well, the critical stresses increase more. If the same mechanical loading is assumed (as in Figures 15 and 16) the stress input may be unable to follow the critical stresses. In such a circumstance, the forward transformation may be incomplete. This is illustrated in Figure 16 when $\xi < 1$ despite the stress input is at its maximum. Nevertheless, incomplete reverse transformations are also predicted due to SMA temperature drops.

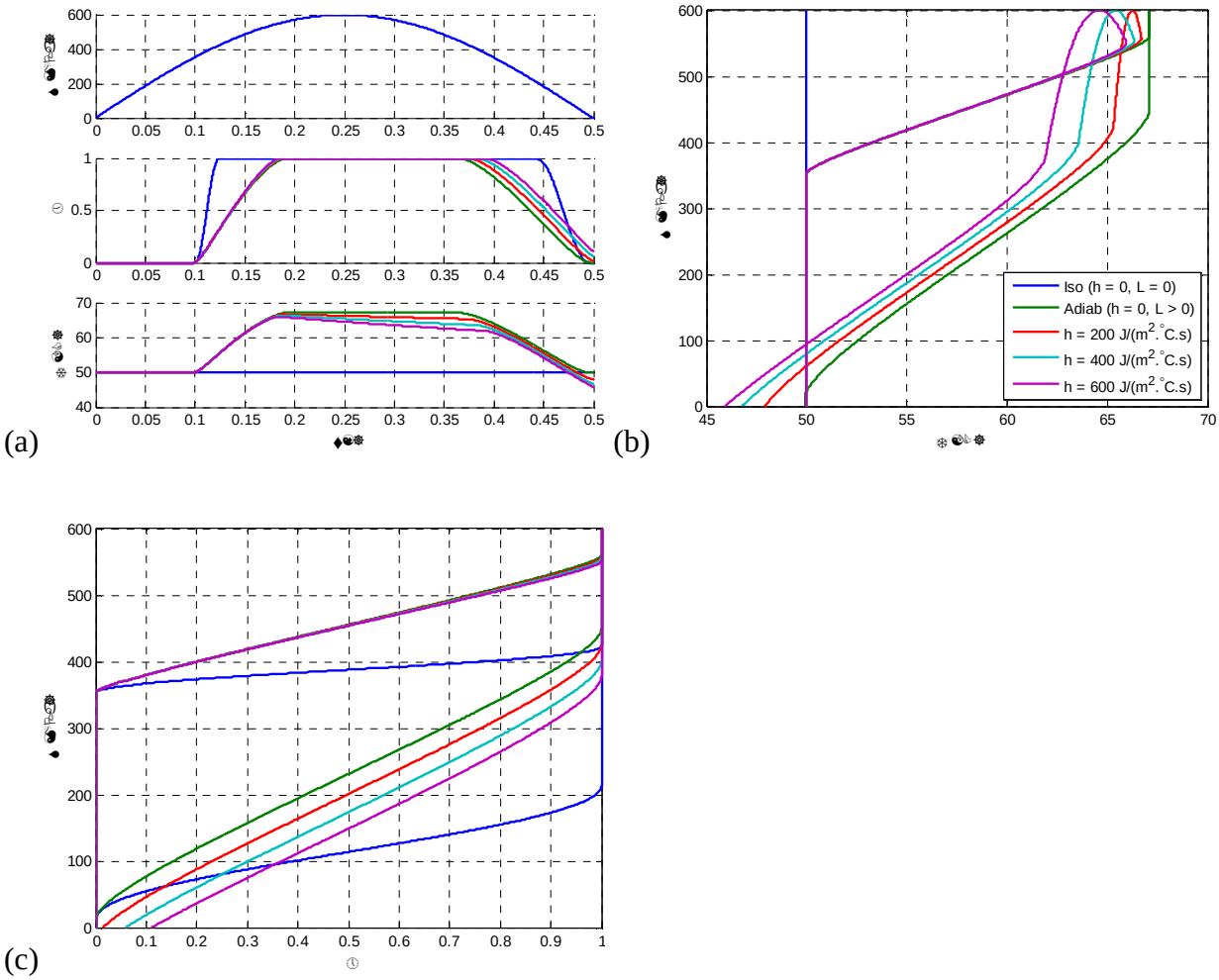


Figure 15: Isothermal and adiabatic processes in addition to non-isothermal conditions with increasing convective heat transfer coefficient. Reverse transformation is more affected by increasing convection coefficient.

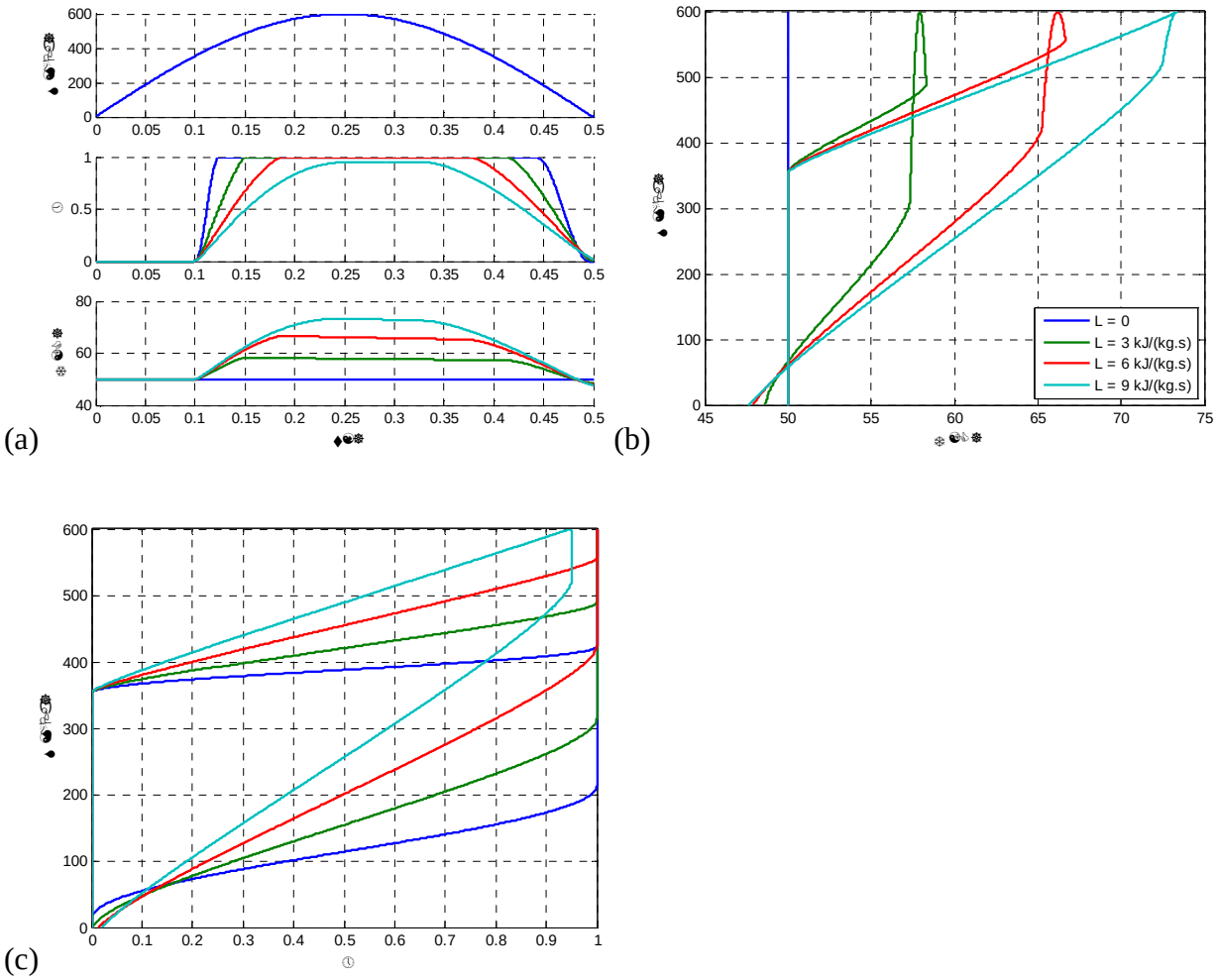


Figure 16: Isothermal process in addition to non-isothermal conditions with increasing latent heat (convection coefficient is fixed at $200\ \mathrm{J \cdot m^{-2} \cdot °C^{-1} \cdot s^{-1}}$).

Figure 17 illustrates two configurations discussed in Figure 16, namely $L = 6\ \mathrm{kJ \cdot kg^{-1} \cdot °C^{-1}}$ and $9\ \mathrm{kJ \cdot kg^{-1} \cdot °C^{-1}}$ (both at $50\ °\mathrm{C}$). In Figure 17(a) the forward transformation is complete whereas in Figure 17(b) is incomplete, because the martensite finish stress is never reached for the same loading conditions.

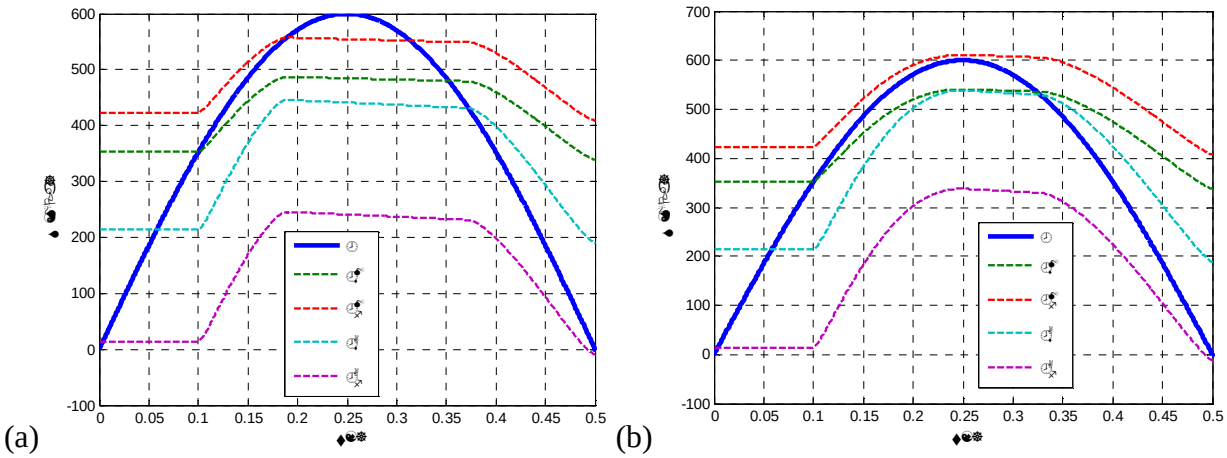


Figure 17: Effect of temperature variations (due to latent heat) on critical stresses – (a) complete and (b) incomplete forward phase transformations are illustrated for $L = 6\ \text{kJ} \cdot \text{kg}^{-1} \cdot {}^{\circ}\text{C}^{-1}$ and $9\ \text{kJ} \cdot \text{kg}^{-1} \cdot {}^{\circ}\text{C}^{-1}$, respectively. Initial SMA temperature of $50\ {}^{\circ}\text{C}$.

3.3. Effect of Temperature on SMA Phase Transformation Behavior

Figure 18 shows the SMA behavior for two different material temperatures, $T = 50\ {}^{\circ}\text{C}$ and $60\ {}^{\circ}\text{C}$ (due to Joule heating at $V = 1.77\ \text{V}$ and $2.05\ \text{V}$). At $50\ {}^{\circ}\text{C}$ and after the first forward transformation, the SMA temperature drops below the austenite finish temperature. The temperature drop is due to some amount of natural convection (after the forward transformation has stopped) and energy absorption related to the latent heat of reverse transformation. Since T is slightly above A_f, the oscillations in temperature below A_f impede complete reverse transformations. However, at $60\ {}^{\circ}\text{C}$ the SMA temperature oscillates above A_f and therefore the austenitic phase is completely recovered. One may conclude that the lower temperature could deteriorate the performance of an SMA-based actuator while the higher temperature would require higher stresses for achieving phase transformations.

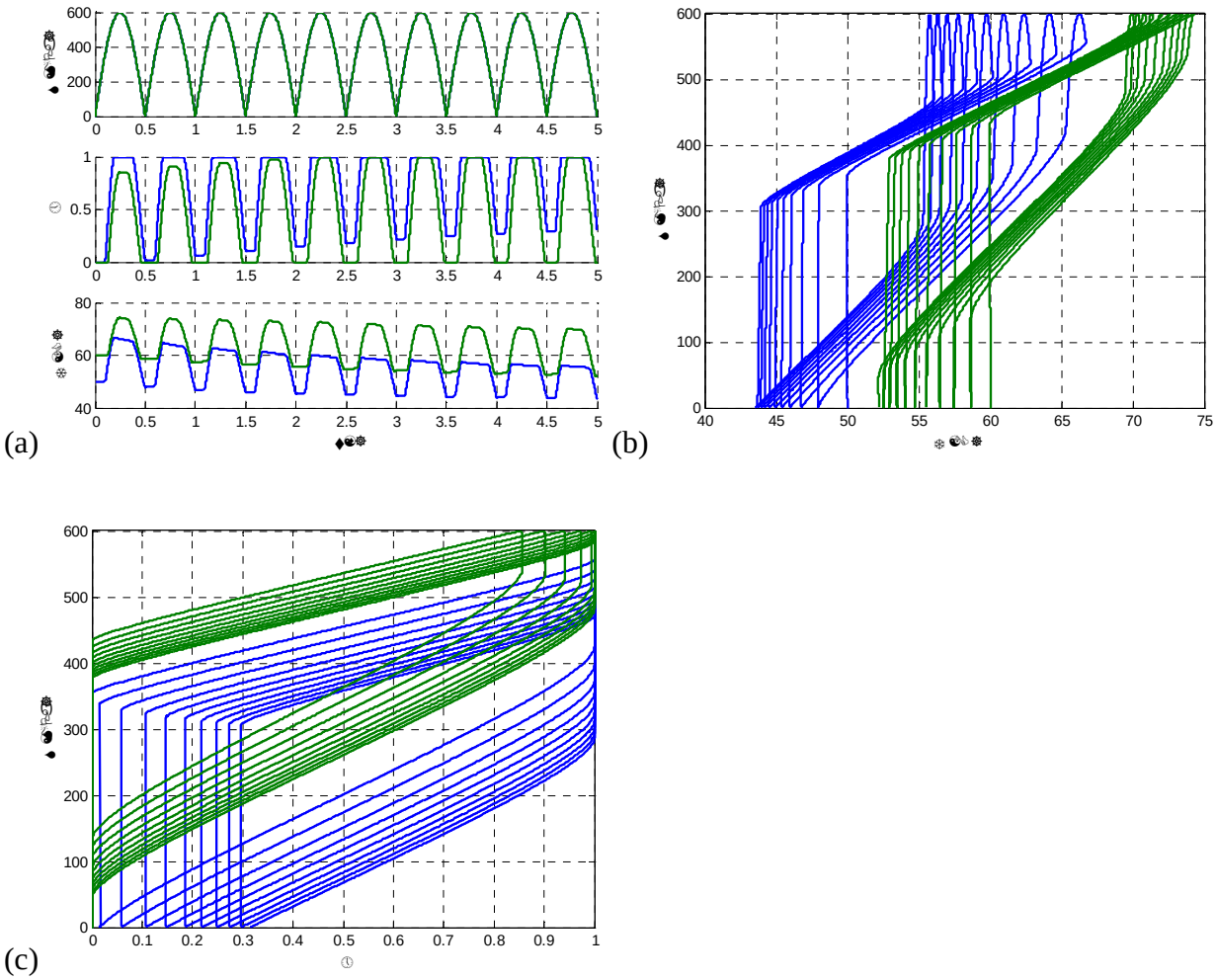


Figure 18: SMA phase transformation and thermal behaviors for increasing material temperature ($T = 50\,°\text{C}$ and $60\,°\text{C}$ due to Joule heating at $V = 1.77\ \text{V}$ and $2.05\ \text{V}$, respectively).

3.4. Effect of SMA Electrical Resistance Variations

Figure 19 illustrates two assumptions for the SMA electrical resistance. First, constant resistance is assumed throughout the forward and reverse transformations. Second, resistance as function of martensitic fraction is assumed. The SMA temperature (due to Joule heating) is shown for both instances for $V = 1.77\ \text{V}$ (which corresponds to an expected temperature of $50\,°\text{C}$). Faulkner et al. (2000) report 14.7% of variation in electrical resistivity between the austenitic and martensitic phases. Here, the average electrical resistance (between R_A and R_M from Table 13) is assumed to vary $\pm 25\%$.

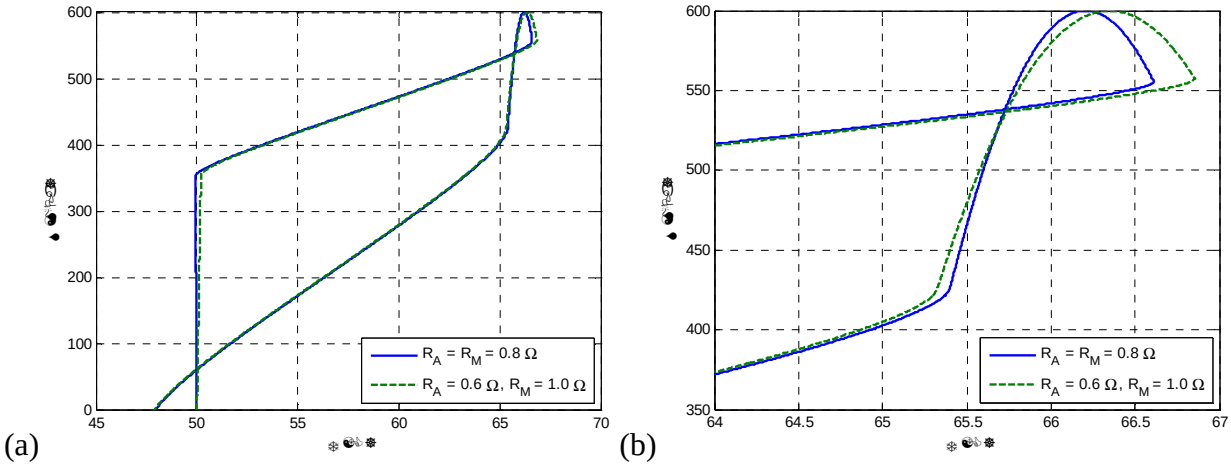


Figure 19: Effect of SMA electrical resistance on temperature for $V = 1.77\ \mathrm{V}$ – (a) complete process of forward and reverse phase transformations and (b) detail of the ending stage of the forward transformation and begin of the reverse transformation.

4. CONCLUSIONS

The implementation of a model in MATLAB/Simulink® for the simulation of 1D phase transformation kinetics of shape memory alloys (SMAs) has been addressed. The transformation kinetics is based on Brinson (1993)'s model, a well-established phenomenological model. The transformation kinetics related to the pseudoelastic effect and shape memory effect above the martensite start temperature is of concern. A heat transfer model based on Faulkner et al. (2000) and Hadi et al. (2010) is included to represent the thermal effects. Joule heating activation, natural convection and the latent heat associated with phase transformation are modeled. The pertinent theory is stated along with the blocks (subsystems) representing the formulae. When convenient, the blocks are "masked" to provide a user interface for entering SMA parameters. The source-code required for the proper working of the masks is also provided.

After the model description, an example of use is provided. The example is followed by several analyses which are important when modeling SMA-based applications. The analyses include parametric studies, thermal boundary conditions studies, and compare simplified assumptions (such as isotherm and adiabatic processes) with more realistic approaches (which account for the latent heat phenomenon). The latter study illustrates how the latent heat affects the SMA temperature and therefore the critical stresses for phase transformation. It is shown that forward transformations increase the SMA temperature (due to energy being released) and bias the martensite start and finish stresses to higher levels. Depending on the loading conditions, forward transformations may not complete. It is also shown that reverse transformations decrease the SMA temperature (due to energy being absorbed). As well, depending on the thermomechanical conditions, reverse transformations may not complete. It is exemplified that if the SMA temperature is held little above the austenite finish temperature, the temperature drop related to reverse transformation may lead the SMA temperature below the austenite finish temperature and would impede the full shape recovery (e.g. yielding in smaller actuation stroke than expected). Finally, the effect of different electrical resistance for the martensitic (usually higher) and austenitic phases is illustrated. When Joule heating is of concern, it may be useful to consider this effect because the SMA temperature may increase due to lower electrical resistance than in its previous phase. As well, if the difference in electrical resistance is expressive, lower electrical current through the SMA element due to the higher electrical resistance may impede the proper heating and therefore actuation performance.

APPENDIX

Function *GetMaskParamPos* returns the index of a given mask parameter. This function is used by the initialization functions below to get values entered by the user through the masks.

```
function val = GetMaskParamPos(block, parname)
    vars = get_param(block, 'MaskVariables');
```

```
    vars = strrep(vars, '@', '');
    spos = strfind(vars, parname); % 1st letter
    epos = spos + length(parname) + 1; % equality char
    scpos = strfind(vars(epos:end), ';'); % semi-colon position
    scpos = scpos(1);
    % the index of the parameter is between '=' and ';'
    val = str2num(vars(epos:(epos+scpos-1))); %#ok<ST2NM>
end
```

Function *StressStartAM_InitMask*:

```
function StressStartAM_InitMask(block)
    vals = get_param(block, 'MaskValues');
    CM = str2double(vals{GetMaskParamPos(block, 'varCM')});
    Ssmin = str2double(vals{GetMaskParamPos(block, 'varSsmin')});
    Ms = str2double(vals{GetMaskParamPos(block, 'varMs')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'CM'], 'Gain', num2str(CM));
    set_param([prefix 'Ssmin'], 'Value', num2str(Ssmin));
    set_param([prefix 'Ms'], 'Value', num2str(Ms));
end
```

Function *StressFinishAM_InitMask*:

```
function StressFinishAM_InitMask(block)
    vals = get_param(block, 'MaskValues');
    CM = str2double(vals{GetMaskParamPos(block, 'varCM')});
    Sfmin = str2double(vals{GetMaskParamPos(block, 'varSfmin')});
    Ms = str2double(vals{GetMaskParamPos(block, 'varMs')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'CM'], 'Gain', num2str(CM));
    set_param([prefix 'Sfmin'], 'Value', num2str(Sfmin));
    set_param([prefix 'Ms'], 'Value', num2str(Ms));
end
```

Function *StressStartMA_InitMask*:

```
function StressStartMA_InitMask(block)
    vals = get_param(block, 'MaskValues');
    CA = str2double(vals{GetMaskParamPos(block, 'varCA')});
    As = str2double(vals{GetMaskParamPos(block, 'varAs')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'CA'], 'Gain', num2str(CA));
    set_param([prefix 'As'], 'Value', num2str(As));
end
```

Function *StressFinishMA_InitMask*:

```
function StressFinishMA_InitMask(block)
```

```
    vals = get_param(block, 'MaskValues');
    CA = str2double(vals{GetMaskParamPos(block, 'varCA')});
    Af = str2double(vals{GetMaskParamPos(block, 'varAf')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'CA'], 'Gain', num2str(CA));
    set_param([prefix 'Af'], 'Value', num2str(Af));
end
```

Function *TempStartMA_InitMask*:

```
function TempStartMA_InitMask(block)
    vals = get_param(block, 'MaskValues');
    CA = str2double(vals{GetMaskParamPos(block, 'varCA')});
    As = str2double(vals{GetMaskParamPos(block, 'varAs')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'CA'], 'Value', num2str(CA));
    set_param([prefix 'As'], 'Value', num2str(As));
end
```

Function *SMA_PhaseTransf_Sin_InitMask*:

```
function SMA_PhaseTransf_Sin_InitMask(block)
    vals = get_param(block, 'MaskValues');
    Ssmin = str2double(vals{GetMaskParamPos(block, 'varSsmin')});
    Sfmin = str2double(vals{GetMaskParamPos(block, 'varSfmin')});
    CM = str2double(vals{GetMaskParamPos(block, 'varCM')});
    CA = str2double(vals{GetMaskParamPos(block, 'varCA')});
    Ms = str2double(vals{GetMaskParamPos(block, 'varMs')});
    As = str2double(vals{GetMaskParamPos(block, 'varAs')});
    Af = str2double(vals{GetMaskParamPos(block, 'varAf')});

    prefix = sprintf('%s/', block);

    % 'StressStartAM' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'StressStartAM'], 'varCM', num2str(CM));
    set_param([prefix 'StressStartAM'], 'varSsmin', num2str(Ssmin));
    set_param([prefix 'StressStartAM'], 'varMs', num2str(Ms));

    % 'StressFinishAM' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'StressFinishAM'], 'varCM', num2str(CM));
    set_param([prefix 'StressFinishAM'], 'varSfmin', num2str(Sfmin));
    set_param([prefix 'StressFinishAM'], 'varMs', num2str(Ms));

    % 'Brin93_MartFracAM' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'Brin93_MartFracAM'], 'varCM', num2str(CM));
    set_param([prefix 'Brin93_MartFracAM'], 'varSsmin', num2str(Ssmin));
    set_param([prefix 'Brin93_MartFracAM'], 'varSfmin', num2str(Sfmin));
    set_param([prefix 'Brin93_MartFracAM'], 'varMs', num2str(Ms));

    % 'StressStartMA' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'StressStartMA'], 'varCA', num2str(CA));
    set_param([prefix 'StressStartMA'], 'varAs', num2str(As));

    % 'StressFinishMA' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'StressFinishMA'], 'varCA', num2str(CA));
    set_param([prefix 'StressFinishMA'], 'varAf', num2str(Af));
```

```
    % 'Brin93_MartFracMA' is inside 'SMA_PhaseTransf_Sin' block
    set_param([prefix 'Brin93_MartFracMA'], 'varCA', num2str(CA));
    set_param([prefix 'Brin93_MartFracMA'], 'varAs', num2str(As));
    set_param([prefix 'Brin93_MartFracMA'], 'varAf', num2str(Af));
end
```

Function *MVF_Resistance_InitMask*:

```
function MVF_Resistance_InitMask(block)
    vals = get_param(block, 'MaskValues');
    RA = str2double(vals{GetMaskParamPos(block, 'varRA')});
    RM = str2double(vals{GetMaskParamPos(block, 'varRM')});
    prefix = sprintf('%s/', block);
    set_param([prefix 'RA'], 'Value', num2str(RA));
    set_param([prefix 'RM'], 'Value', num2str(RM));
end
```

Function *SMA_HeatTransfer_InitMask*:

```
function SMA_HeatTransfer_InitMask (block)
    vals = get_param(block, 'MaskValues');

    m = str2double(vals{GetMaskParamPos(block, 'varm')});
    L = str2double(vals{GetMaskParamPos(block, 'varL')});
    Cp = str2double(vals{GetMaskParamPos(block, 'varCp')});
    RA = str2double(vals{GetMaskParamPos(block, 'varRA')});
    RM = str2double(vals{GetMaskParamPos(block, 'varRM')});
    hc = str2double(vals{GetMaskParamPos(block, 'varhc')});
    Ac = str2double(vals{GetMaskParamPos(block, 'varAc')});
    T0 = str2double(vals{GetMaskParamPos(block, 'varT0')});

    imCp = 1 / (m * Cp);

    prefix = sprintf('%s/', block);

    set_param([prefix 'm'], 'Gain', num2str(m));
    set_param([prefix 'L'], 'Gain', num2str(L));
    set_param([prefix 'hc'], 'Gain', num2str(hc));
    set_param([prefix 'Ac'], 'Gain', num2str(Ac));
    set_param([prefix 'hc1'], 'Gain', num2str(hc));
    set_param([prefix 'Ac1'], 'Gain', num2str(Ac));
    set_param([prefix 'imCp'], 'Gain', num2str(imCp));
    set_param([prefix 'imCp1'], 'Gain', num2str(imCp));
    set_param([prefix 'imCp2'], 'Gain', num2str(imCp));
    set_param([prefix 'imCp3'], 'Gain', num2str(imCp));
    set_param([prefix 'Integrator'], 'InitialCondition', num2str(T0));

    % 'Hadi10_Resistance' is inside ' SMA_HeatTransfer' block
    set_param([prefix 'MVF_Resistance'], 'varRA', num2str(RA));
    set_param([prefix 'MVF_Resistance'], 'varRM', num2str(RM));
end
```

REFERENCES

A. Hadi, A. Yousefi-Koma, M.M. Moghaddam, et al., Developing a novel SMA-actuated robotic module, J. Sens. Actuators A 162 (2010) 72-81.

A. Nespoli, S. Besseghini, S. Pittaccio, et al., The high potential of shape memory alloys in developing miniature mechanical devices: A review on shape memory alloy mini-actuators, J. Sens. Actuators A 158 (2010) 149-160.

C. Bil, K. Massey, E.J. Abdullah, Wing morphing control with shape memory alloy actuators, J. Intelligent Mater. Syst. Struct. 24 (2013) 879-898.

D.C. Lagoudas (Ed.), Shape memory alloys: Modeling and engineering applications, New York: Springer (2008).

G. Song, N. Ma, H.-N. Li, Applications of shape memory alloys in civil structures, Engineering Structures 28 (2006) 1266-1274.

J. Andrianesis, A. Tzes, Development and control of a multifunctional prosthetic hand with shape memory alloy actuators, J. Intell. Robot. Syst. (2014) (33pp).

J.M. Jani, M. Leary, A. Subic, et al., A review of shape memory alloy research, applications and opportunities, Materials and Design 56 (2014) 1078-1113.

L.C. Brinson, One-dimensional constitutive behavior of shape memory alloys: thermomechanical derivation with non-constant material functions and redefined martensite internal variable, J. Intelligent Mater. Syst. Struct. 4 (1993) 229-242.

L.G. Machado, M.A. Savi, Medical applications of shape memory alloys, Braz. J. Med. Biol. Res. 36 (2003) 683-691.

M. Tabesh, B. Lester, D. Hartl, et al., Influence of the latent heat of transformation and thermomechanical coupling on the performance of shape memory alloy actuators, In: Proc. ASME 2012 Conf. on Smart Materials, Adaptive Structures and Intelligent Systems, SMASIS2012, September 19-21, 2012, Stone Mountain, Georgia, USA.

M.G. Faulkner, J.J. Amalraj, A. Bhattacharyya, Experimental determination of thermal and electrical properties of Ni-Ti shape memory wires, Smart Mater. Struct. 9 (2000) 632-639.

M.H. Elahinia, M. Ahmadian, An enhanced SMA phenomenological model: I. The shortcomings of the existing models, Smart Mater. Struct. 14 (2005a) 1297-1308.

M.H. Elahinia, M. Ahmadian, An enhanced SMA phenomenological model: II. The experimental study, Smart Mater. Struct. 14 (2005b) 1309-1319.

M.H. Elahinia, M. Hashemi, M. Tabesh, et al., Manufacturing and processing of NiTi implants: A review, Progress in Materials Science 57 (2012) 911-946.

M.M. Kheirikhah, S. Rabiee, M.E. Edalat, A review of shape memory alloy actuators in robotics, In: J. Ruiz-del-Solar, E. Chown, P.G. Plöger (Eds.) RoboCup 2010: Robot Soccer World Cup XIV, LNAI 6556 (2011) 206-217.

R.A. Ibrahim, Recent advances in nonlinear passive vibration isolators, J. Sound Vibration 314 (2008) 371-452.

S. Barbarino, E.I. Saavedra Flores, R.M. Ajaj, et al., A review on shape memory alloys with applications to morphing aircraft, Smart Mater. Struct. 23 (2014) 063001 (19pp).

S. Korkmaz, A review of active structural control: challenges for engineering informatics, Computers and Structures 89 (2011) 2113-2132.

S.-H. Ko, J.-S. Bae, J.-H. Rho, Development of a morphing flap using shape memory alloy actuators: the aerodynamic characteristics of a morphing flap, Smart Mater. Struct. 23 (2014) 074015 (21pp).

V.C. Sousa, C. De Marqui Jr., Effect of pseudoelastic hysteresis of shape memory alloy springs on the aeroelastic behavior of a typical airfoil section, J. Intelligent Mater. Syst. Struct. In Press (2014).