Benchmarking of Flatness-based Control of the Heat Equation

Stephan Scholz[1*], Lothar Berger[1], Dirk Lebiedz[2]

[1]Control and Process Engineering, University of Applied Sciences Ravensburg-Weingarten, Weingarten, Germany
[*]*stephan.scholz@rwu.de*

[2]Institute of Numerical Mathematics, Ulm University, Ulm, Germany

Abstract. Flatness-based control design is a well established method to generate open-loop control signals. Several articles discuss the application of flatness-based control design for (reaction-) diffusion problems in various scenarios. Beside the pure analytical derivation also the numerical computation of the input signal is crucial to yield a reliable trajectory planning. Therefore, we derive the input signal step-by-step and describe the influence of system and controller parameters on the computation of the input signal. In particular, we benchmark the control design of the one-dimensional heat equation with Neumann-type boundary actuation for pure aluminum and steel 38Si7, and discuss the applicability of the found input signals for realistic scenarios.

1 Introduction

The flatness-based control method is an open-loop technique to steer the system output along a reference trajectory [1]. In case of finite-dimensional linear and nonlinear systems the input signal $u(t)$ is found by a finite number of derivatives of a (differentially flat) output which equals the reference signal. This approach is extended to infinite-dimensional and distributed parameter systems where theoretically an infinite number of derivatives of output signal $y(t)$ is necessary to compute the input signal $u(t)$, see [2, 3, 4]. However, for practical reasons we can only consider a finite number of derivatives of the output signal. Thus, we need to show that the computation of input signal $u(t)$ converges for a certain number of derivatives of $y(t)$. In general, this estimation of convergence is not trivial because the computation of $u(t)$ depends on system and control parameters. A related approach about the controllability of the heat equation with a finite number of derivatives of y is discussed in [4].

In this contribution, we assume a one-dimensional linear heat equation with Neumann boundary actuation

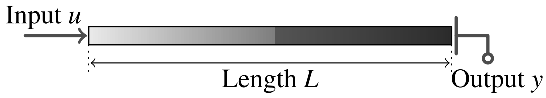


Figure 1: One-dimensional rod with heat input (left) and temperature measurement (right).

as depicted in Fig. 1 to discuss the impact of system and control parameters on the computation of input signal $u(t)$. For this purpose, we compare pure aluminum and steel 38Si7 to exemplify our findings. They differ in their material properties: thermal conductivity λ, specific heat capacity c and density ρ. Regarding the control parameters, we design the reference trajectory as a smooth step which is configured by the transition time and the steepness [5]. In each step of the analysis, we evaluate numerically the significance of the system and control parameters on the final control signal. Hence, we show the transition from a pure analytical towards a simulation-based control design, which enables us to distinguish whether or not a control signal is indeed applicable for a system.

In section 2 we introduce the flatness-based modeling for the one-dimensional heat equation and derive input signal $u(t)$. The influence of the system parameters are analyzed in section 3. The trajectory planning problem and the subsequent discussion of the control parameters are described in section 4 and 5, respectively. Finally, in section 6 we present the simulation results of the open-loop system and review the applicability for realistic scenarios.

2 Flatness-based Control

We assume a one-dimensional heat conduction model as portrayed in Fig. 1 which is described by the linear

ARGESIM Report 47 (ISBN 978-3-903347-65-6), p 103-111, DOI: 10.11128/arep.47.a4703

equation

$$\dot{\vartheta}(t,x) = \alpha \frac{\partial^2}{\partial x^2} \vartheta(t,x) \tag{1}$$

for $(t,x) \in (0,T) \times (0,L)$ and the Neumann boundary conditions as actuation on the left side

$$u(t) = \lambda \frac{\partial}{\partial x} \vartheta(t,x) \cdot \vec{n}_0 \Big|_{x=0}, \tag{2}$$

and thermal insulation on the right side

$$0 = \lambda \frac{\partial}{\partial x} \vartheta(t,x) \cdot \vec{n}_L \Big|_{x=L} \tag{3}$$

where the outer normal vectors are known as $\vec{n}_0 = -1$ and $\vec{n}_L = 1$. Here, we denote the temperature as ϑ, the thermal conductivity as $\lambda > 0$ and the diffusivity as $\alpha = \frac{\lambda}{c\,\rho}$ with specific heat capacity $c > 0$ and density $\rho > 0$. This heat conduction model is strongly simplified because in real world scenarios, often we have to consider two- or three-dimensional heat conduction with temperature-dependent material properties and probably thermal emissions consisting of linear heat transfer and nonlinear heat radiation towards the environment, see also [6, 7]. However, such realistic heat conduction scenarios lead to a much more complex mathematical discussion which is out of scope of this contribution, and the presented control method and its numerical analysis might not be applicable anymore. The initial temperature distribution is arbitrarily defined by

$$\vartheta(0,x) = \vartheta_0(x)$$

for $x \in [0,L]$ and the temperature is measured on the right boundary as

$$y(t) = \vartheta(t,L). \tag{4}$$

As known from the literature [2, 3, 4] the heat equation can be represented by a power series approach. So, we define power series

$$w(t,x) := \sum_{i=0}^{\infty} w_i(t) \frac{(L-x)^i}{i!}$$

and find its derivatives with respect to position x as

$$\frac{\partial}{\partial x} w(t,x) = -\sum_{i=0}^{\infty} w_{i+1}(t) \frac{(L-x)^i}{i!} \quad \text{and} \tag{5}$$
$$\frac{\partial^2}{\partial x^2} w(t,x) = \sum_{i=0}^{\infty} w_{i+2}(t) \frac{(L-x)^i}{i!}.$$

We model heat equation (1) in terms of

$$\dot{w}(t,x) = \alpha \frac{\partial^2}{\partial x^2} w(t,x),$$

identify both sides by its power series expressions as

$$\sum_{i=0}^{\infty} \dot{w}_i(t) \frac{(L-x)^i}{i!} = \alpha \sum_{i=0}^{\infty} w_{i+2}(t) \frac{(L-x)^i}{i!}$$

and yield identity

$$\dot{w}_i(t) = \alpha\, w_{i+2}(t). \tag{6}$$

Next, we apply the information of both boundary sides on identity (6) to derive the input signal. Firstly, we consider the output signal (4) as

$$y(t) = w(t,L) = \sum_{i=0}^{\infty} w_i(t) \frac{0^i}{i!} = w_0(t)$$

which implies $\frac{d^i}{dt^i} y(t) = \frac{d^i}{dt^i} w_0(t) = \alpha^i w_{2i}$ with identity (6). Secondly, the boundary condition on the right side (3) is formulated as

$$\lambda \frac{\partial}{\partial x} w(t,L) = -\lambda \sum_{i=0}^{\infty} w_{i+1}(t) \frac{0^i}{i!} = -\lambda w_1(t) = 0$$

and we find $\frac{d^i}{dt^i} w_1(t) = \alpha^i w_{2i+1} \equiv 0$. Thus, identity (6) is described by the sequences

$$w_{2i}(t) = \alpha^{-i} y^{(i)}(t) \quad \text{and} \quad w_{2i+1}(t) = 0 \tag{7}$$

for all $n \in \{0, 1, \ldots, \infty\}$. In the definition of boundary actuation (2) we insert Equation (5) to derive the input signal $u(t)$ as

$$u(t) = -\lambda \frac{\partial}{\partial x} w(t,0) = \lambda \sum_{i=0}^{\infty} w_{i+1}(t) \frac{L^i}{i!}$$

and further with $i \to 2i+1$ and Equation (7) as

$$u(t) = \lambda \sum_{i=0}^{\infty} \frac{L^{2i+1}}{\alpha^{i+1}} \frac{1}{(2i+1)!} y^{(i+1)}(t). \tag{8}$$

3 Influence of System Parameters

We are interested in the sequence values of series (8) because for implementation reasons we need to how much memory has to be reserved for the computation of u and at which iteration i the summation can be stopped. The power series to compute input signal $u(t)$ can be separated in sequence

$$\eta_i = \frac{L^{2i+1}}{\alpha^{i+1}} \frac{1}{(2i+1)!}. \tag{9}$$

and the derivatives of the (desired) output signal $y^{(i+1)}(t)$. In this section we discuss the influence of the physical properties length L and diffusivity α on sequence η_i, and in section 5 we analyze the parameters of (target) output $y(t)$ and its derivatives.

Sequence η_i is positive for all $i \in \{0,1,\dots,\infty\}$ as we assume $L > 0$, $\alpha > 0$, and has a crucial influence on the computation of the input function u because it scales the derivatives $y^{(i+1)}$. Thus, we need to know the approximate values of η_i. We use a rescaled version of sequence (9) as

$$\tilde{\eta}_i := \left(\frac{L^2}{\alpha}\right)^{i+1} \frac{1}{(2i+1)!} = \frac{\gamma^{i+1}}{(2i+1)!} = L\,\eta_i$$

where $\gamma := \frac{L^2}{\alpha}$ to show that η_i and $\tilde{\eta}_i$ increase up to some index i and decrease afterwards to zero. Increasing iterator i by one we yield

$$\begin{aligned}\tilde{\eta}_{i+1} &= \frac{\gamma^{[i+1]+1}}{(2[i+1]+1)!} \\ &= \frac{\gamma^{i+1}}{(2i+1)!} \frac{\gamma}{(2i+2)(2i+3)} = \tilde{\eta}_i\,\beta_i\end{aligned}$$

where $\beta_i = \frac{\gamma}{(2i+2)(2i+3)}$ and we notice

$$\frac{\tilde{\eta}_{i+1}}{\tilde{\eta}_i} > 1 \quad \Leftrightarrow \quad \beta_i > 1$$

and

$$\frac{\tilde{\eta}_{i+1}}{\tilde{\eta}_i} < 1 \quad \Leftrightarrow \quad \beta_i < 1.$$

Due to the definition of $\tilde{\eta}$ this concept holds also for the original sequence (9) as $\eta_{i+1} = \beta_i\,\eta_i$. So, the maximum value of $\tilde{\eta}_i$ and η_i and its corresponding iterations i_{max} depend only on γ. For example, if we assume $\gamma = 100$ then $\gamma < (2i+2)(2i+3)$ holds for $i \in \{1,2,3\}$ and we find the maximum value $\tilde{\eta}_4 = \frac{100^5}{9!} \approx 27557$.

Example: Comparison Aluminum and Steel

For our numerical evaluations we consider a rod of length $L = 0.2$ for two case scenarios: a rod made of pure aluminum [8] and a rod made of steel 38Si7 [9]. The physical properties of both materials are listed in Table 1. For aluminum we have $\gamma_{al} \approx 410$ and for steel 38Si7 we have $\gamma_{st} \approx 3588$. The sequences $\eta_{al,i}$ and $\eta_{st,i}$ and their ratios $\frac{\eta_{al,i+1}}{\eta_{al,i}}$ and $\frac{\eta_{st,i+1}}{\eta_{st,i}}$ which describe evolution of the sequences by iteration are portrayed in Fig. 2 in semi-logarithmic scaling. We find that inequality $\frac{\eta_{i+1}}{\eta_i} > 1$ or equally $\log_{10}\left(\frac{\eta_{i+1}}{\eta_i}\right) > 0$ holds in case of aluminum for $i \in \{1,\dots,8\}$ and in case of steel $i \in \{1,\dots,28\}$. Thus the maximum values of η_i for aluminum and steel are calculated by

$$\eta_{al,9} = \frac{L^{19}}{\alpha_{al}^{10}\ 19!} \approx 5.53 \cdot 10^9$$

and

$$\eta_{st,29} = \frac{L^{59}}{\alpha_{st}^{30}\ 59!} \approx 1.59 \cdot 10^{27}.$$

As both sequences $\eta_{al,i}$ and $\eta_{st,i}$ reach such enormous maximum values, computational issues related to big numbers and data types have to be considered in the implementation process.

Moreover, sequence $\log_{10}(\eta_{al,i})$ drops below zero for $i > 27$: $\eta_{al,28} \approx 0.73$, $\log_{10}(\eta_{al,28}) \approx -0.13$; and $\log_{10}(\eta_{st,i})$ drops below zero for $i > 82$: $\eta_{st,83} \approx 0.13$, $\log_{10}(\eta_{st,83}) \approx -0.87$ (not displayed in Fig. 2).

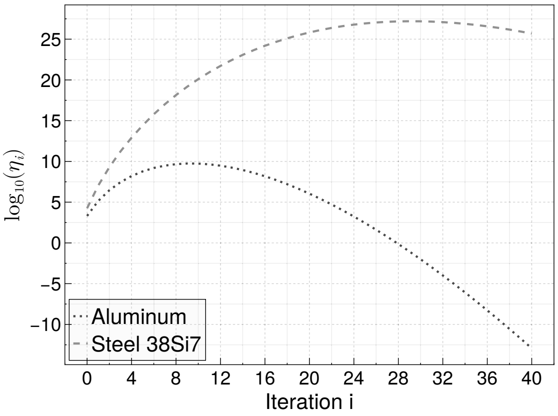
4 Trajectory Planning

According to [3, 5] we consider a transition from one fixed operating point to the next one as

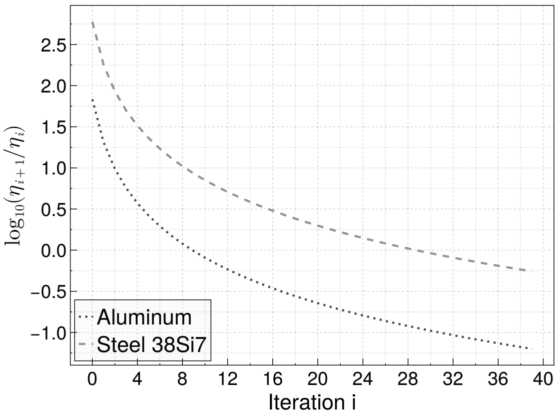
$$y(t) = y_0 + \Delta y\ \Phi_{\omega,T}(t) \tag{10}$$

Table 1: PHYSICAL PROPERTIES

	λ	ρ	c	$\alpha = \frac{\lambda}{\rho c}$
Aluminum	237	2700	900	$9.75 \cdot 10^{-5}$
Steel 38Si7	40	7800	460	$1.11 \cdot 10^{-5}$



(a) Sequence η_i



(b) Ratio η_{i+1}/η_i

Figure 2: Sequence η_i (top) and ratio $\frac{\eta_{i+1}}{\eta_i}$ (bottom) for aluminum and steel 38Si7.

where $\Delta y = y_f - y_0$, and with transition function

$$\Phi_{\omega,T}(t) = \begin{cases} 0 & t \leq 0, \\ 1 & t \geq T, \\ \frac{\int_0^t \Omega_{\omega,T}(\tau)\,\mathrm{d}\tau}{\int_0^T \Omega_{\omega,T}(\tau)\,\mathrm{d}\tau} & t \in (0,T) \end{cases}$$

which uses the integral of the bump function

$$\Omega_{\omega,T}(t) = \begin{cases} 0 & t \notin [0,T], \\ \exp\left(-1/\left(\left[1-\frac{t}{T}\right]\frac{t}{T}\right)^{\omega}\right) & t \in (0,T). \end{cases}$$

Parameter ω steers the steepness of transition $\Phi_{\omega,T}$ and is chosen such that the Gevrey order $go = 1 + \frac{1}{\omega} < 2$ or equally $\omega > 1$. A small value of ω, e.g. $\omega = 1.1$ means a rather flat transition, whereas a large value, e.g. $\omega = 3.0$ means a quite steep transition, as depicted in Fig. 3. To compute the input signal $u(t)$ in Equation (8) we only need to find the derivatives

$$\frac{d^i}{dt^i} y(t) = \Delta y \, \Phi^{(i)}_{\omega,T}(t) \tag{11}$$

where the derivatives of transition $\Phi_{\omega,T}$ are calculated as

$$\Phi^{(i)}_{\omega,T}(t) = \frac{\Omega^{(i-1)}_{\omega,T}(t)}{\hat{\Omega}_{\omega,T}} \quad \text{for} \quad t \in (0,T) \tag{12}$$

and $\Phi^{(i)}_{\omega,T}(t) = 0$ for $t \notin (0,1)$, using integral

$$\hat{\Omega}_{\omega,T} := \int_0^T \Omega_{\omega,T}(\tau)\,\mathrm{d}\tau. \tag{13}$$

In Fig. 3 trajectory $\Phi_{\omega,T}(t)$ and its first derivative are portrayed for varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$. The derivatives $\Phi^{(i)}_{\omega,T}(t)$ can be computed symbolically using for example computer-algebra systems (see for example the MATLAB implementation [10]), numerically (which we do not recommend here). In this contribution, we compute the derivatives $\Omega^{(i)}_{\omega,T}$ with the JULIA library *BellBruno.jl* [11]. We note without a proof that an increasing order of differentiation of $\Omega^{(i)}_{\omega,T}$ leads to stronger oscillations because bump function $\Omega_{\omega,T}$ is a function composition and smooth as $\Omega_{\omega,T} \in \mathscr{C}^{\inf}((0,T))$, see also [2].

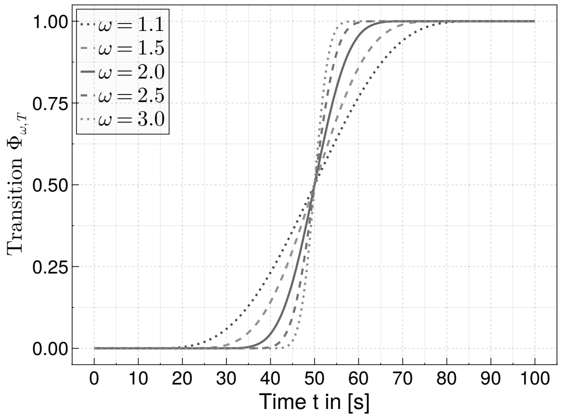
5 Influence of Control Parameters

The configuration of transition $\Phi_{\omega,T}$ and its derivatives are mainly driven by two parameters: final time T and exponent ω. In this section, we apply the L^2 norm

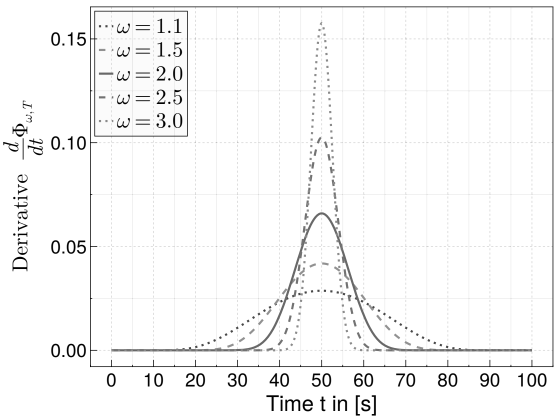
$$\|f\|_{L^2} = \sqrt{\int_0^T |f(t)|^2 dt}$$

on $\frac{d^i}{dt^i}\Omega_{\omega,T}(t)$ to unveil the influence of final time T and exponent ω on the computation of input signal (8). Noting the input signal with sequence η_n as

$$u(t) = \lambda \sum_{i=0}^{\infty} \eta_i \, y^{(i+1)}(t) \; = \; \frac{\lambda \, \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i \, \Omega^{(i)}_{\omega,T}(t)$$

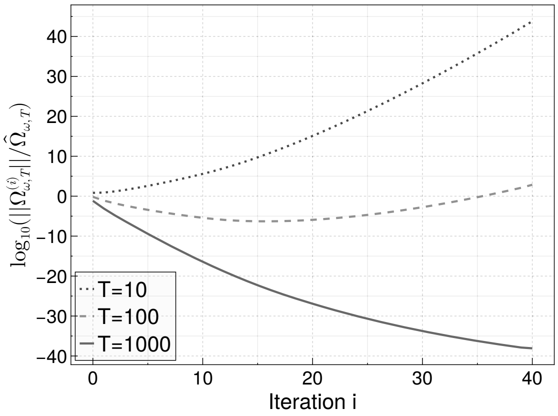


(a) Trajectory

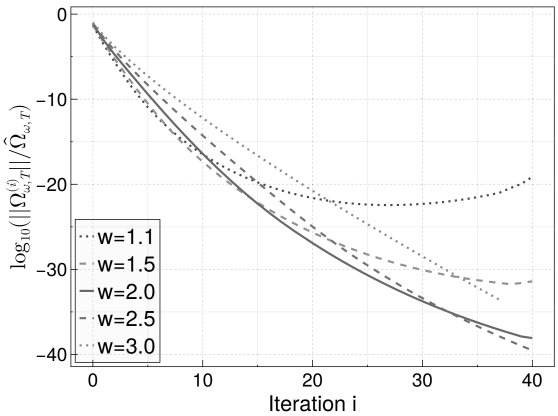


(b) First derivative

Figure 3: Trajectory $\Phi_{\omega,T}$ (top) and its first derivative (bottom) with $T = 100$ seconds and varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$.



(a) Fixed $\omega = 2.0$, varying T



(b) Fixed $T = 1000$, varying ω

Figure 4: Norm of $\Omega^{(i)}_{\omega,T}$ with fixed $\omega = 2.0$ (top), and fixed $T = 1000$ (bottom).

using identities (11,12,13), we find the L^2 norm of $u(t)$ as

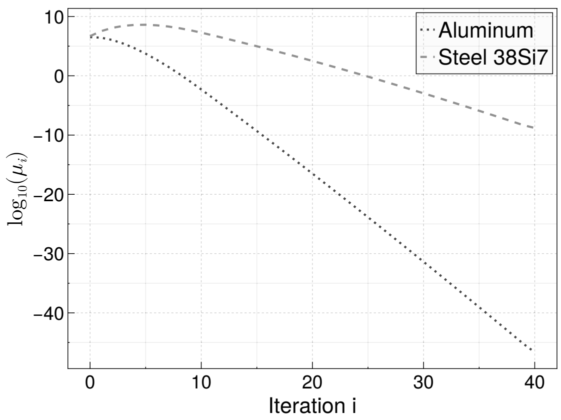
$$\begin{aligned} \|u(t)\|_{L^2} &= \left\| \frac{\lambda\ \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i\ \Omega^{(i)}_{\omega,T}(t) \right\| \\ &\leq |\Delta y|\ \frac{\lambda}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{\infty} \eta_i\ \left\| \Omega^{(i)}_{\omega,T}(t) \right\| \end{aligned}$$

where we assume $\lambda, \hat{\Omega}_{\omega,T}, \eta_i > 0$. We see that the power series is mainly driven by η_i (as discussed before) and derivatives $\Omega^{(i)}_{\omega,T}(t)$. Therefore, we are able to describe the quantitative behavior of the input signal by evaluating sequence

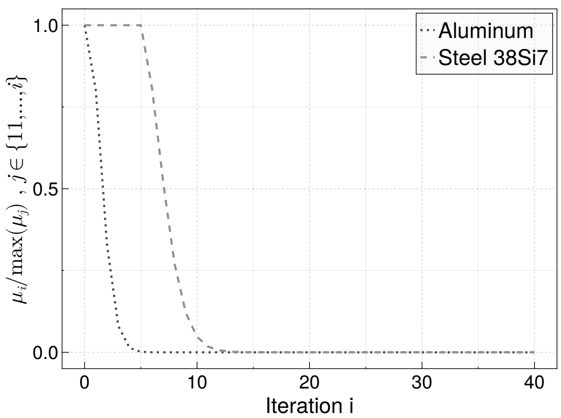
$$\mu_i := \frac{\lambda\,|\Delta y|}{\hat{\Omega}_{\omega,T}}\ \eta_i\ \left\| \Omega^{(i)}_{\omega,T}(t) \right\|_{L^2}. \tag{14}$$

Scaled norm $\|\frac{d^i}{dt^i}\Omega_{\omega,T}(t)\|_{L^2}/\hat{\Omega}_{\omega,T}$ is portrayed in Figure 4 in logarithmic scaling for two scenarios: fixed $\omega = 2$ and varying $T \in \{10, 100, 1000\}$; and fixed $T = 1000$ and varying $\omega \in \{1.1, 1.5, 2.0, 2.5, 3.0\}$. One notes that an increasing value only of final time T leads to a reduction of $\|\Omega^{(i)}_{\omega,T}(t)\|/\hat{\Omega}_{\omega,T}$, the influence of steepness ω may not be so clear here.

Furthermore, we take advantage of sequence μ_i to find a suitable maximum iteration number i_{max} to terminate the power series of $u(t)$ in Equation (8). Sequence μ_i consists of η_i as defined in Equation (9) and so we



(a) Sequence μ_i as in (14)



(b) $\mu_i / \max_{j \in \{1,...,i\}} \mu_j$

Figure 5: Sequence μ_i (top) and ratio $\frac{\mu_i}{\max_{j \in \{1,...,i\}} \mu_j}$ (bottom) for $\omega = 2.0$ and $T = 1000$.

distinguish aluminum and steel 38Si7 as noted in Table 1). The different values of η_i for aluminum and steel 38Si7 as in Fig. 2 lead to different values of μ_i: sequence μ_i approaching zero *faster* in case of aluminum than steel 38Si7 as depicted in Fig. 5 (a). Introducing the ratio $\frac{\mu_i}{\max_{j \in \{1,...,i\}} \mu_j}$ we find that the sequence elements μ_i vanish in case of aluminum for iterations approximately above $i = 5$ whereas in case of steel 38Si7 it takes at least $i = 12$ iterations - as portrayed in Fig. 5 (b).

The evaluation of μ_i and ratio $\frac{\mu_i}{\max_{j \in \{1,...,i\}} \mu_j}$ unveils two facts about the generation of input signal $u(t)$. Comparing the results for aluminum and steel 38Si7, we find in case of aluminum that only the very first derivatives of $\Phi_{\omega,T}$ are weighted by η_i and higher order derivatives have almost no influence on the computation of $u(t)$. Whereas in case of steel 38Si7 the weights of derivatives increase up to the fifth derivative so higher order derivatives (which tend to oscillatory behavior) influence the found input signal, too. We find an approximation of the signal input

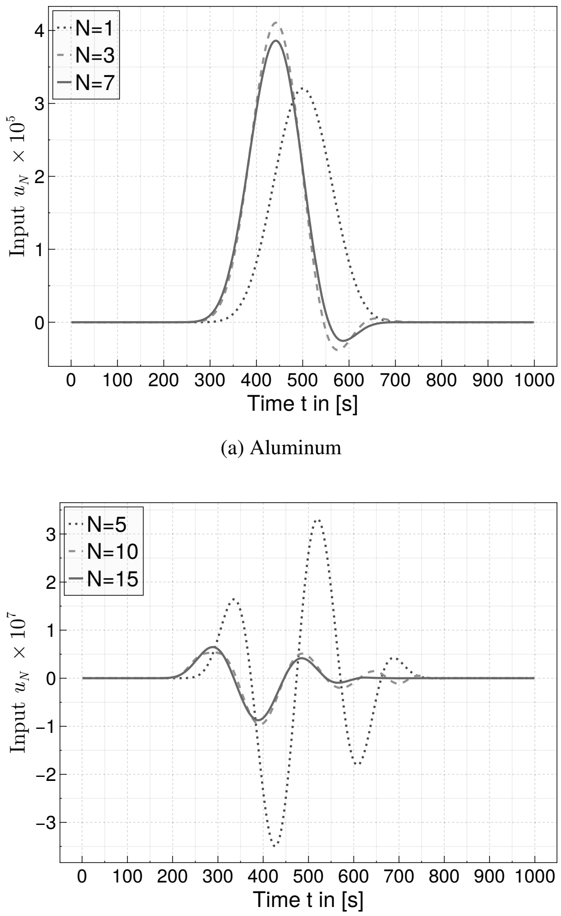
$$u(t) \approx \frac{\lambda\ \Delta y}{\hat{\Omega}_{\omega,T}} \sum_{i=0}^{N} \eta_i\ \Omega_{\omega,T}^{(i)}(t) =: u_N(t) \tag{15}$$

where $N \in \mathbb{N}_{\geq 0}$ denotes the upper limit of iterations. Following the previous ideas, in case of aluminum a small value of N, e.g. $N = 7$, suffices to generate a good approximation. However, for steel 38Si7 we need a higher number of iterations, e.g. $N = 15$. The progress of input signals for aluminum with $N \in \{1, 3, 7\}$ and steel 38Si7 with $N \in \{5, 10, 15\}$ are presented in Fig. 6. These results confirm our previous analysis that the input signal needs more series elements for steel 38Si7 than for aluminum, and this leads the stronger oscillations in Fig. 6 (b) because higher derivatives of trajectory $\Phi_{\omega,T}^{(i)}(t)$ are necessary. In a nutshell, we find four important parameters which influence the input signal: length of rod L and diffusivity α which define sequence η_n, and final time T and steepness ω which influences the derivatives of trajectory $\Phi_{\omega,T}$ - and thus also the necessary number of summation iterations.

6 Simulation Results

In this section we compare the computed input signals and the resulting heat conduction simulation for aluminum and steel 38Si7. As above we assume the physical properties as listed in Table 1 and the trajectory parameters $T = 1000$ seconds and $\omega = 2.0$. So, the integral of the bump function as in identity (13) is found as $\hat{\Omega}_{\omega,T} \approx 17.06 \cdot 10^{-6}$. Further, we assume an initial temperature $\vartheta_0(x) = 300$ Kelvin which shall be increased by $\Delta y = 100$ Kelvin. A maximum iteration number of $N = 40$ is considered for approximation (15) in case of both scenarios. As explained in Section 5 lower values than $N = 40$ are also sufficient but it may rather imitate a summation until $N = \infty$. Heat equation (1) is discretized in space using finite differences with 101 grid points and is simulated using a Runge-Kutta numerical integration method for stiff systems, see [12].

The input signals and the resulting temperatures are illustrated in Fig. 7 for aluminum (a,c) and steel 38Si7 (b,d). In both cases the output, meaning the temperature at $x = 0.2$ meter, follows the reference and reaches 400



(a) Aluminum

(b) Steel 38Si7

Figure 6: Progress of approximated input signals u_N for aluminum (top) and steel 38Si7 (bottom) with $T = 1000$ and $\omega = 2$.

Kelvin. So, from a pure *mathematical* point of view the input signals are computed correctly for both scenarios. However, from a *physical* point of view we need to discuss the input signals and the resulting temperatures rather critically. Beside the fact that it may not be possible to apply negative input signals, e.g. if the actuator offers only heating and not cooling, it is in fact not possible to reach temperatures below zero Kelvin as portrayed for steel reference in Fig. 7 (d). We highlight that the strong oscillations of the input signal for steel 38Si7 in Fig. 7 (b) lead to the unrealistic temperature evolution in Fig. 7 (d). Therefore, the control parameters final time T and steepness ω have to be readjusted to decrease the necessary number of series elements, to yield a lesser or no oscillating input signal and a realizable temperature evolution.

Conclusion & Discussion

In this article we presented the computation of input signals for trajectory planning of a one-dimensional heat equation using flatness-based control design. We found in our analysis of the influence of system and control parameters on the computation of the input signal that different material properties (aluminum, steel 38Si7) result in completely different input signals and open-loop dynamics - even if all other parameters (length of rod, final time, steepness of transition) are the same. We demonstrate that strictly following the flatness-based control design may not lead to an physically realizable input signal even if the series in Eq. (8) converges. Thus, we recommend to simulate the heat equation with input signal to gain trustworthy arguments for the applicability of the computed flatness-based input signal. We motivate further research on the proposed approach for realistic scenarios in two and three dimensions including thermal convection and radiation.

Source Code

The source code is developed in JULIA programming language [13] and is available on *GitHub* and *Zenodo*: [14]. We implemented the simulations with the JULIA libraries *OrdinaryDiffEq.jl* [15] for the numerical integration in time of the spatially approximated heat equation, and *Makie.jl* [16] to create the figures.

References

[1] Fliess M, Lévine J, Martin P, Rouchon P. Flatness and defect of non-linear systems: introductory theory and examples. International Journal of Control. 1995; 61(6) (1995): 1327–1361.

[2] Laroche B, Martin P, Rouchon P. Motion planning for the heat equation. International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal. 2000; 10(8): 629–643.

[3] Rudolph J, Winkler J, Woittennek F. Flatness Based Control of Distributed Parameter Systems: Examples and Computer Exercises from Various Technological Domains. Shaker. 2003.

[4] Martin P, Rosier L, Rouchon P. Null controllability of the 1D heat equation using flatness. IFAC Proceedings Volumes. 2013; 46(26): 7–12.

[5] Utz T, Graichen K, Kugi A. Trajectory planning and receding horizon tracking control of a quasilinear diffusion-convection-reaction system. IFAC Proceedings Volumes. 2010; 43(14): 587–592.

[6] Scholz S, Berger L. Modeling of a multiple source heating plate. arXiv preprint. arXiv:2011.14939 (2020).

[7] Scholz S, Berger L. Hestia.jl: A Julia library for heat conduction modeling with boundary actuation. Simulation Notes Europe SNE. 2023; 33(1): 27–30.

[8] Connor N. Aluminium - Periodic Table. [Online]. Available:https://www.periodic-table.org/Aluminium-periodic-table/. [Accessed: Mar. 28, 2024].

[9] Ovako AB. 38Si7. [Online]. Available: https://steelnavigator.ovako.com/steel-grades/38si7/. [Accessed: Mar. 28, 2024].

[10] Fischer F, Gabriel J, Kerschbaum S. coni-a Matlab toolbox facilitating the solution of control problems. Zenodo. 2022. Available: https://zenodo.org/record/6420876.

[11] Scholz S. BellBruno.jl. Zenodo. 2023. Available: https://doi.org/10.5281/zenodo.7685927

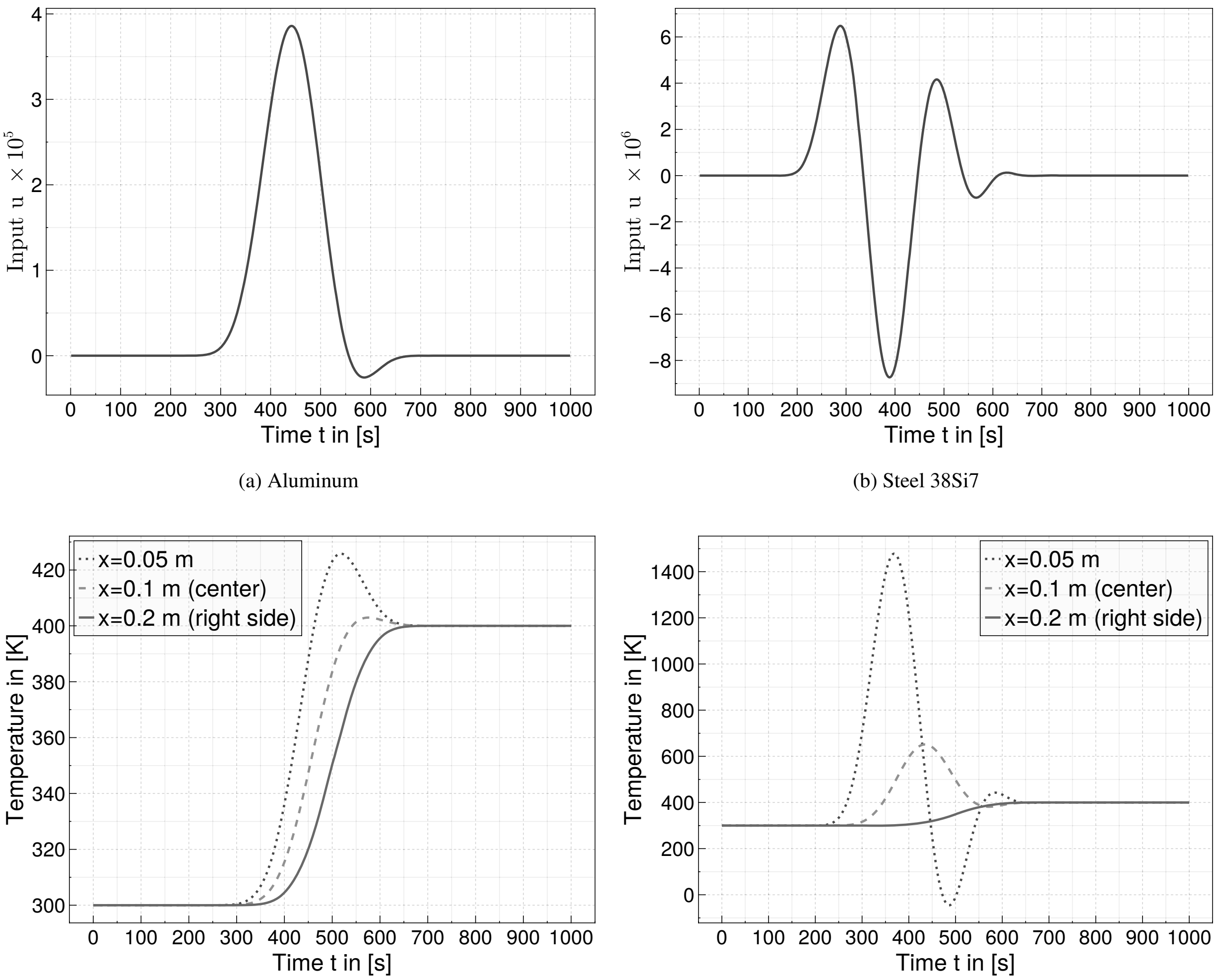
[12] Kennedy CA, Carpenter MH. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Applied numerical mathematics. 2003; 44(1-2): 139–181.

[13] Bezanson J, Edelman A, Karpinski S, Shah VB. Julia: A fresh approach to numerical computing. SIAM review. 2017; 59(1):65–98.

[14] Scholz S. BenchmarkFlatnessbasedControl.jl: BenchmarkFlatnessbasedControl.jl v0.1.0. 2024. Available https://doi.org/10.5281/zenodo.10890245

[15] Rackauckas C, contributors. SciML/DifferentialEquations.jl. Zenodo. 2022. Available: https://doi.org/10.5281/zenodo.7239171

[16] Danisch S, Krumbiegel J. Makie. jl: Flexible high-performance data visualization for Julia. Journal of Open Source Software. 2021; 6(65):3349.



(a) Aluminum

(b) Steel 38Si7

(c) Aluminum

(d) Steel 38Si7

Figure 7: Input signals and the resulting temperatures at position $x \in \{0.05, 0.1, 0.2\}$.