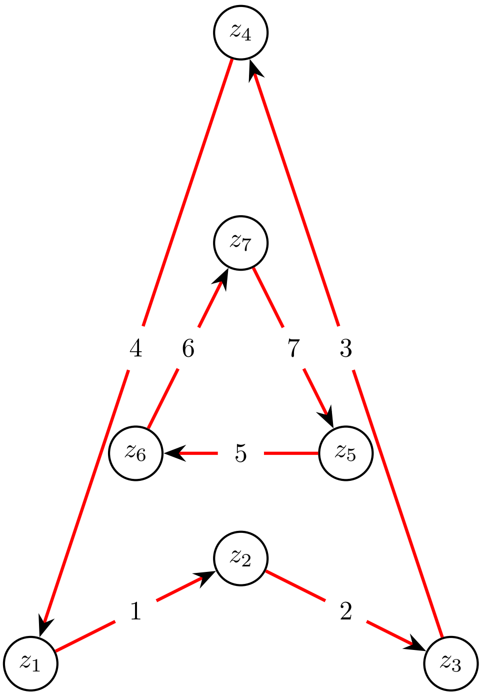
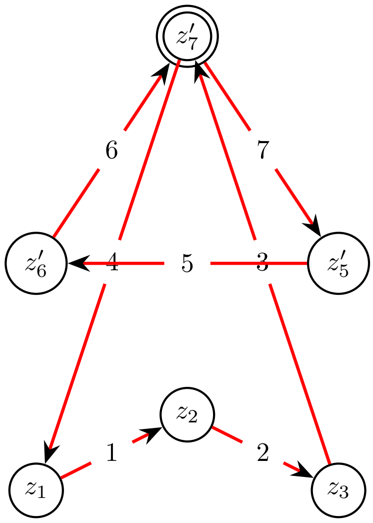
Scripta-Ingenia

A study of the letter A

This study of the letter A serves to illustrate the concept of abstract complex link. A complex link structure is a mathematical structure consisting of three things: a set of indexes, say A, an endomap of A, say $\varphi\colon A \to A$, and a geometrical realization map into the complex plane, say $g\colon A \to \mathbb{C}$. It can be seen as an abstraction to the concept of parametrized planar curve, that is, a continuous function from the unit interval into the complex plane.





Let us consider the complex points z_1 to z_7 (as illustrated on the left) with $z_1 = 1+0i$, $z_2 = 3+1i$, $z_3 = 5+0i$, $z_4 = 3+6i$, $z_5 = 4+2i$, $z_6 = 2+2i$ and $z_7 = 3+4i$. In this case, the set A is a set of indexes $\{1, \ldots, 7\}$, the endomap $\varphi\colon A \to A$ encodes the cyclic permutations $(1, 2, 3, 4)$ and $(5, 6, 7)$ whereas the geometric realization map $g\colon A \to \mathbb{C}$ associates each complex number to the respective index, that is, $g(k) = z_k$, with $k = 1, \ldots, 7$. There are several advantages in interpreting the abstract structure (A, φ, g) as a generalization of a discretization of a parametrized planar curve. In the first place there is a clear separation between the geometric information and the topological one (the geometry has to do with angles and distances whereas topology is concerned with the way things are connected). For example, if we move the points z_6, z_7 and z_5 to new locations $z'_6 = 1+3i$, $z'_7 = 3+6i$ and $z'_5 = 5+3i$ we are only changing the image of the map g, as illustrated in the second image on the left. The fact that the points z'_7 and z_4 overlap is a mere geometrical coincidence, there is no real topological connection between them. However, we observe that an intersection was created between the edge 5 and edges 3 and 4. Once again this intersections are merely geometrical. If one wishes to make them topological then it is required to introduce new vertices and split the arrows 3 and 4 into new edges, conveniently labeled respectively as 3.0, 3.5, 4.0, 4.5 whereas edge 5 needs to be split into three new edges, conveniently labeled as 5.0, 5.25, 5.75. This is convenient since the arrows 3 and 4 are intersected at $t = 0.5$ of its length while edge 5 is intersected at $t = 0.25$ and $t = 0.75$ of its length. Moreover, taking advantage of the structure of the complex plane we may define a first differential of the structure as $d = g\varphi - g$ and a second differential as $d^2 = \frac{d(\varphi)}{d}$. Then g is recovered from d^2 up to two constants c_1 and c_2 as $g = c_1 + \mathrm{cumsum}(c_2 * \mathrm{cumprod}(d^2))$.

A Scripta-Ingenia assume-se como uma revista de divulgação científica tratando temas da ciência e da tecnologia, cobrindo todas as áreas do saber no domínio das ciências exactas ou aplicadas. Interessa-se ainda por artigos de opinião, sobre tópicos científicos ou não, desde que escritos por autores na área das ciências e da engenharia, e que reflitam as suas opiniões enquanto membros dessa comunidade. Este é o seu número 11 e corresponde ao Solstício de Inverno de 2022.

Director and Chief Editor — Nelson Martins-Ferreira
CDRSP-ESTG, IPLeiria

The Eiffel Tower under construction in July 1888. Roger Viollet/Getty Images

Ficha Técnica
Director: Nelson Martins Ferreira
Proprietário: Instituto Politécnico de Leiria, anotado na ERC
Morada: Rua General Norton de Matos; Apartado 4133; 2411-901 Leiria, Portugal
Director Adjunto: Nuno Alves; Sub Director: Artur Mateus
Editor: Nelson Martins Ferreira; Sede de Redação: Rua General Norton de Matos; Apartado 4133; 2411-901 Leiria, Portugal; Contacto: scripta.ingenia@ipleiria.pt
Colaboradores e estatuto editorial: `http://cdrsp.ipleiria.pt/scriptaingenia/`

Modeling and control of a suspended load movement testbed apparatus

by Carlos Neves[a,b], Diogo Baptista[a,b,c], Vicente Caneiro[a]

[a]ESTG, Polytechnic of Leiria,
Campus 2, Morro do Lena - Alto do Vieiro
2411-901 Leiria - Portugal

[b]INESC Coimbra, Pólo II, R. Silvio Lima,
3030-290 Coimbra - Portugal

[c]CIMA-UE, University of Évora
Colégio Luís Verney,
Rua Romão Ramalho, 59
7000-671 Évora - Portugal

Abstract The control of lifting devices is very important for improving task performance and safety. In this paper, the construction of a laboratory test rig is presented and consists of a system driving a wagon rail, which can move along 18 m, with a 1 m long free-wing pendulum attached. Actuation in the presented system model has the velocity of the cart as input, rather than the more traditional force input models. A comparison between simulated and real operation is presented, in closed loop as well as in open loop.

Keywords: Laboratory apparatus, Crane control, Suspended load, System modeling, linearization.

1 Introduction

Hoisting and load transportation devices are commonly used in different industrial and logistic operations ranging from manufacturing to construction, from quarries and mines to port operation. There are different construction and operation modes, as well as different number of degrees of freedom of the movements and configurations. These devices can be classed into Boom, Rotary, and Bridge/Gantry cranes, the last of which is the more frequent in industrial manufacturing facilities [1], [10].

Common to (horizontal) rotary and gantry cranes, there is a cart moving along a horizontal rail, to which the load suspension cable is attached. Generally, there is a considerable length to this cable, inducing significant oscillatory movements of the load, both due to the (operator) imposed movement as well as due to external disturbances like the wind. This circumstance can impose restrictions on operational performance and can be a hazard as well, and there may be a considerably lengthy learning curve for operators to deal efficiently and safely with their task.

This clearly motivates the application of automatic control systems and strategies to overcome this problem, which in turn proves to be a good control problem to test and contrast different control strategies, as well as controller implementations in a laboratory environment.

As a result, a test rig was built to serve as a testbed for the study of different approaches to this control problem at a laboratory scale. The system is built on a 18 m long rail with a sliding cart from which a 1 m long rod is freely hanging. The cart is driven by a stepper motor to which the controller imposes speed, while both the cart position and the rod's oscillation are being measured.

This paper describes the device built, the mathematical models derived and some encouraging preliminary control results used to validate the rig as a testbed and to ensure the significant improvement of introducing automatic control on the operation of such a system.

In the remainder of this work, a review of recent results on the subject is presented in section two, followed, in section 3, by the derivation of the relevant mathematical models used for simulation and controller tuning. Then, section 4 shows the description of the device built and the main aspects of the construction. Next, in section 5, simulated and real results of the application of double PID control are presented and compared. Finally, some conclusions are drawn and future direction for this work are pointed out.

2 Related work

The challenge provided by the control of crane-type devices has been taken by researchers for quite some time and there is a well-established base of related work in the literature. Two review publications have been the starting point for the present work, the first [1] gathers research results up to 2003, and the second reviews the published work up to 2016 [10].

In what the control approaches are concerned, most initial works were based on simplified and linear modeling. However, in [1] there is a report of some limitations of these models to cope with the inherent nonlinearities stemming from external disturbances and non-modeled behaviors. According to [10], other authors have studied the influence of such nonlinearities in the robustness of the controlled systems.

To improve the oscillation of the load, mainly happening in the beginning and in the end of the trajectories, there is the reference to open loop, closed-loop and hybrid approaches. By using open-loop control, it is possible to shape the reference command so that the acceleration imposed by the movement is gradually applied, taming load oscillation [6]. This input-shaping technique can be improved if more initial information, such as cable length or load mass is also taking into consideration [2].

Closed-loop techniques take the benefits of real-time measuring of the controlled variables (mainly load oscillation and cart position) to improve performance, even in the presence of disturbances as well as unmodeled or poorly modeled features [8], [7]. This closed-loop operation showed good results even if the oscillation angle is not directly measured, but estimated using the mathematical model [12]. However, [10] refers that closed-loop systems in large cranes can be negatively influenced by sensor noise, which may result in stability problems and, consequently, in safety issues.

A hybrid approach, combining input shaping for reducing expected oscillations and feedback control to deal with unexpected disturbances has been also tested with good results [5]–[9].

Other works were relevant for this development, namely those concerned with modeling and testing pendulum-based systems (natural or inverted), such as [3]–[11].

3 System Model derivation:

3.1 Mechanics

For the derivation of the system's model, a simplified cart-pendulum (Fig. 1) approach was used, where a rod with no mass was considered and the masses of load and cart were placed in their respective centers of mass.

In the exposed simplified model, the following parameters stand out:

- u is the external force applied to the car, in Newton (N);
- l is the length of the cable connecting the car to the suspended mass, in meters (m);
- θ is the angle between the vertical position and the instantaneous position of the pendulum, in degrees (°);
- m is the suspended mass, in kilogram (kg);
- M is the mass of the cart, in kilogram (kg);
- $M_f^{pendulum}$ is the moment of friction of the pendulum, in Newton × meter (N.m).
- F_f^{cart} is the frictional force of the cart, in Newton (N);

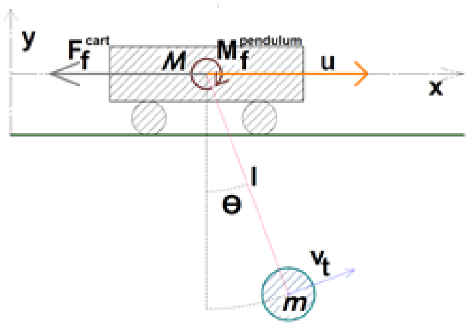


Figure 1: Simplified model of the cart-pendulum system.

A mechanical model can be calculated from the Lagrangean approach by differentiating the Lagrange function L

$$L\left(r_1, r_2, ..., r_n; v_1, v_2, ..., v_n; t\right) \equiv T - U, \tag{3.1}$$

where T and U are the kinetic and potential energies, respectively, using

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i^{ext}, \tag{3.2}$$

In this equation, q_i are the generalized coordinates and Q_i^{ext} the external generalized forces applied. Since the system has two degrees of freedom, (cart displacement and load oscillation) the chosen generalized coordinates were those displacements and the generalized forces involved are the applied force to the cart and the friction force and moment applied at the cart's wheels and to the oscillating rod axis, respectively.

The potential energy of the system, U_s, is only that of the pendulum, U_p since the cart has a horizontal displacement, so,

$$U_s = U_p = mgl - mgl\cos(\theta) = mgl\left(1 - \cos\left(\theta\right)\right) \tag{3.3}$$

As for the kinetic energy, both bodies contribute, so, considering that the velocity vector for the pendulum can be decomposed as

$$v_t^x = \dot{\theta} l \cos\left(\theta\right) \tag{3.4}$$

$$v_t^y = \dot{\theta} l \sin\left(\theta\right) \tag{3.5}$$

the kinetic energy of the pendulum becomes

$$T_P = \frac{1}{2} m \left(\left(\dot{x} + v_t^x \right)^2 + \left(v_t^y \right)^2 \right). \tag{3.6}$$

The kinetic energy of the car is given by

$$T_C = \frac{1}{2} M \dot{x}^2 \tag{3.7}$$

and therefore the kinetic energy of the system, T_s it becomes

$$\begin{aligned} T_s =& T_C + T_P = \\ &= \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m \left(\dot{x}^2 + 2l\dot{x}\dot{\theta}\cos(\theta) + \dot{\theta}^2 l^2 \right), \end{aligned} \tag{3.8}$$

The Lagrangean function results in

$$\begin{aligned} L &= T_S - U_S = \\ &= \frac{1}{2} \left[(m+M) \dot{x}^2 + 2ml\dot{x}\dot{\theta}\cos(\theta) + m\dot{\theta}^2 l^2 \right] \\ &- mgl \left(1 - \cos(\theta) \right) \end{aligned} \tag{3.9}$$

As mention previously, the generalized coordinates and generalized external forces are,

$$q = \begin{bmatrix} x \\ \theta \end{bmatrix} \text{ and } Q^{ext} = \begin{bmatrix} u - F_f^{cart} \\ -M_f^{pendulum} \end{bmatrix} \text{ or}$$

$$Q^{ext} = \begin{bmatrix} u - D_x \cdot \dot{x} \\ -D_\theta \cdot \dot{\theta} \end{bmatrix} \tag{3.10}$$

where D_x and D_θ are the damping coefficients for the cart wheels and the rod pin. From equation (3.2) applied to each of the generalized coordinates one gets,

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial L}{\partial x} = Q_x^{ext} \Leftrightarrow$$

$$\ddot{x}(m+M) + ml\ddot{\theta}\cos(\theta) - ml\dot{\theta}^2 \sin(\theta) = u - D_x \cdot \dot{x}, \tag{3.11}$$

and

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q_\theta^{ext} \Leftrightarrow \tag{3.12}$$

$$ml\ddot{x}\cos(\theta) + mgl\sin(\theta) + ml^2\ddot{\theta} = -D_\theta \cdot \dot{\theta}$$

which are the nonlinear equations of movement for the system.

3.2 Nonlinear state-space model

Using as the state variables the linear and angular displacements of the cart and pendulum and their respective velocities, a nonlinear state-space model can be calculated as

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, x_3, x_4) \\ f_2(x_1, x_2, x_3, x_4) \\ f_3(x_1, x_2, x_3, x_4) \\ f_4(x_1, x_2, x_3, x_4) \end{bmatrix} \tag{3.13}$$

where

$$f_1(x_1, x_2, x_3, x_4) = x_2 \tag{3.14}$$

$$\begin{aligned} & f_2(x_1, x_2, x_3, x_4) = \\ & - \frac{l^2 . \sin(x_3) . m . x_4^2}{l . \left[\cos^2(x_3) . m - m - M \right]} - \frac{\cos(x_3) . l . \sin(x_3) . g . m}{l . \left[\cos^2(x_3) . m - m - M \right]} \\ & + \frac{\cos(x_3) . D_\theta . x_4}{l . \left[\cos^2(x_3) . m - m - M \right]} - \frac{-l . D_x . x_2 + l . u}{l . \left[\cos^2(x_3) . m - m - M \right]} \end{aligned} \tag{3.15}$$

$$f_3(x_1, x_2, x_3, x_4) = x_4 \tag{3.16}$$

$$\begin{aligned} & f_4(x_1, x_2, x_3, x_4) = \frac{\cos(x_3) . l^2 . \sin(x_3) . m^2 . x_4}{l^2 . m . \left[\cos^2(x_3) . m - m - M \right]} + \\ & \frac{- \cos(x_3) . l . D_x . m . x_2 + l . M . \sin(x_3) . g . m}{l^2 . m . \left[\cos^2(x_3) . m - m - M \right]} + \\ & + \frac{l . \sin(x_3) . g . m^2 + \cos(x_3) . l . m . u}{l^2 . m . \left[\cos^2(x_3) . m - m - M \right]} + \\ & \frac{M . D_\theta . x_4 + m . D_\theta . x_4}{l^2 . m . \left[\cos^2(x_3) . m - m - M \right]} . \end{aligned} \tag{3.17}$$

3.3 Linear state-space model

Because the system will be operating around an equilibrium point, a linear approach to the model can be proposed by stating a small oscillations hypothesis and by expanding the differential equations as a first-order Taylor series.

Let $\bar{x}$ be the equilibrium point of the system, that corresponds to the rest position, i.e.

$$\bar{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.18}$$

and

$$u = 0, \tag{3.19}$$

we obtain the linearized state equation of the nonlinear state equation 3.13 as

$$\dot{x} = Ax + Bu \tag{3.20}$$

where

$$\dot{x} = \left.\frac{\partial f}{\partial x}\right|_{(x,u)=(\bar{x},0)} .x + \left.\frac{\partial f}{\partial u}\right|_{(x,u)=(\bar{x},0)} .u \text{ , i.e.}$$

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{-D_x}{M} & \frac{g.m}{M} & \frac{D_\theta}{l.M} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{D_x}{l.M} & \frac{-g.(m+M)}{M.l} & \frac{-D_\theta.(m+M)}{M.m.l^2} \end{bmatrix} .x$$

$$+ \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{M.l} \end{bmatrix} .u \tag{3.21}$$

Since the system measures both displacements, the output equation is trivial [4] and becomes

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} .x \tag{3.22}$$

3.4 Velocity driven linear state-space model

Since the mathematical model given in (3.21) takes into account that the system input consists of a force, u, imposed on the car, and since in this work a stepper motor with a simple speed (frequence) imposing driver is used as the cart actuator, there is a need to rewrite the mathematical model with the linear velocity of the car as the input variable. So, from the state equations

$$\begin{cases} \dot{x} = v \\ \ddot{x} = \dot{v} = -\frac{D_x}{M}v + \frac{g.m}{M}\theta + \frac{D_\theta}{l.M}\dot{\theta} + \frac{1}{M}u \\ \dot{\theta} = \dot{\theta} \\ \ddot{\theta} = \frac{D_x}{l.M}v - \frac{g.(m+M)}{M.l}\theta - \frac{D_\theta.(m+M)}{M.m.l^2}\dot{\theta} - \frac{1}{M.l}u \end{cases} \tag{3.23}$$

the second equation can be solved for the input u and introduced in the fourth equation resulting, in

$$\ddot{\theta} + \left(\frac{D_\theta}{m.l^2}\right)\dot{\theta} + \frac{g}{l}\theta = -\frac{1}{l}\dot{v} \tag{3.24}$$

Following [13], by integrating the equation twice and setting the first state variable as $X_1 = \theta$ one gets,

$$\dot{X}_1 = -\left(\frac{D_\theta}{m.l^2}\right)\theta - \frac{g}{l}\int\theta - \frac{1}{l}v. \tag{3.25}$$

If the second state variable is defined as $X_2 = -\frac{g}{l}\int\theta$, the equation 3.25 can be written as

$$\dot{X}_1 = -\left(\frac{D_\theta}{m.l^2}\right)\theta + X_2 - \frac{1}{l}v. \tag{3.26}$$

Adding, as the third state variable, the displacement of the cart, a simpler linear third-order model can be obtained:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \end{bmatrix} = \begin{bmatrix} -\left(\frac{D_\theta}{m.l^2}\right) & 1 & 0 \\ -\frac{g}{l} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} -\frac{1}{l} \\ 0 \\ 1 \end{bmatrix} v$$

$$\begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} \tag{3.27}$$

3.5 Control strategy

To validate the built system as testbed as well as the derived models, a decoupled double PID controller was implemented, as suggested by several authors [7], [12], [3]–[11],

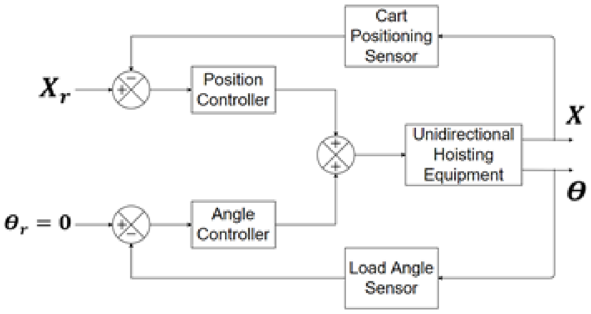


Figure 2: Schematic of control system using 2 closed loop PID controllers.

As can be seen in Fig. 2, the control strategy is implemented using two similar controllers, one respecting to linear displacement and the other to angular displacement. Both controllers use standard PID formulations and the control input to the system is calculated by combining the output of both controllers with equal weights. Given that the objective is to reduce oscillations during the whole excursion of the movement the angular displacement reference is naturally set to zero, effectively turning the angle displacement loop into a regulator.

4 The built rig

The experimental setup built is an 18 m long rail with a $1,5$ kg cart to which a 1 m long aluminum rod is attached through a bearing and allowed to pivot freely. The mass is a set of cylindrical weights that can be used to vary the mass of the pendulum up to 2 kg at this stage. The motion is imposed by a stepper motor attached to the structure and connected to the cart by a timing belt Fig. 3.

The system is sensorized using a rotary incremental encoder attached to the motor shaft, and a potentiometer synchronized with the rod's pivoting axis.

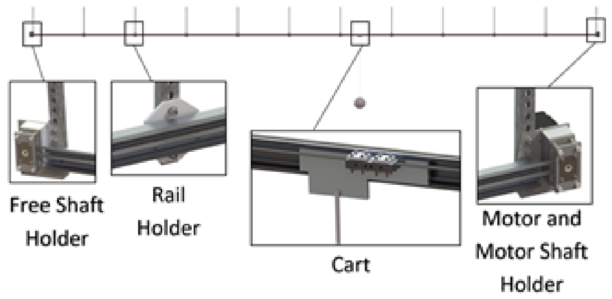


Figure 3: Equipment modeling rendering.

The control hardware, Fig. 4, consists of a microcontroller board (Arduino Mega) and an external board with the connections for the sensors as well as a set of operating and emergency switches.

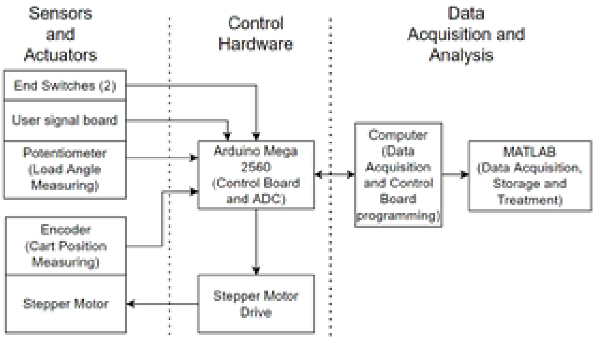

Sensors and Actuators
Control Hardware
Data Acquisition and Analysis
End Switches (2)
User signal board
Potentiometer (Load Angle Measuring)
Encoder (Cart Position Measuring)
Stepper Motor
Arduino Mega 2560 (Control Board and ADC)
Stepper Motor Drive
Computer (Data Acquisition and Control Board programming)
MATLAB (Data Acquisition, Storage and Treatment)

Figure 4: Architecture of the equipment control system.

The control software was built using a standard PID library for Arduino[1] and using interrupt driven routines to accurately generate the pulse train for the stepper motor drive operation to capture the encoder readings, as well as to respond to events generated by both the limit switches and the user-operated switches.

5 Tests and results

With implemented rig and the models derived, a set of tests were performed to establish preliminary results on both the proximity of the open-loop model predictions in simulation with real performance as well as to ascertain the improvement of the performance of the controlled system.

In fact, the modeling options were shown to be appropriate, since the simulated performance in open loop is very close to the measured test results as can be seen in Fig. 5, where a comparison is done with the free oscillation of the pendulum dropped with an initial condition of, approximately, 25 degrees.

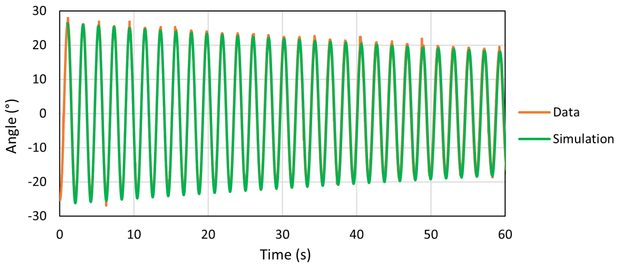


Figure 5: Comparison of real data acquired and simulation data on the evolution of the oscillation angle of a load (0.5 kg) when it is dropped from a height corresponding to approximately 25 Degrees

For the closed-loop system, we will consider, kp_{lin}, ki_{lin} and kd_{lin} as the proportional, integrative and derivative constants of the car position controller, and kp_{ang}, ki_{ang} and kd_{ang} as the counterpart constants of the angle controller. The closed-loop operation also validates the models as can be seen in Fig. 6, depicting a test where a step input in position with an amplitude of $2,25$ m was applied. Notice that, in these tests, the angle loop operates as a regulator, since the objective is to maintain a zero-degree (vertical) angle on the rod.

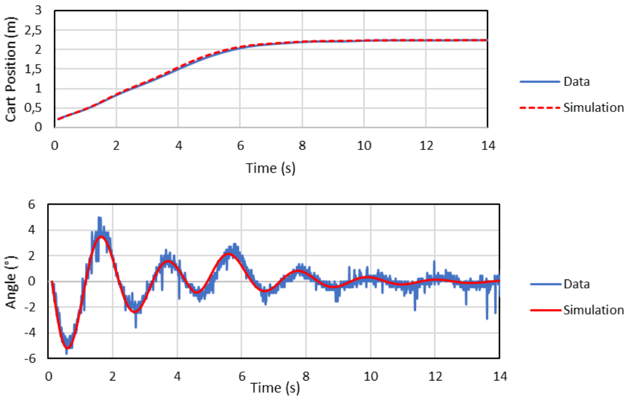
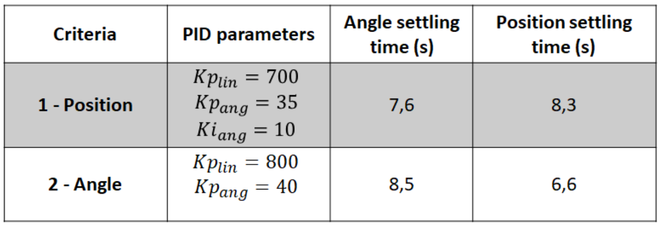


Figure 6: Position of the car and Angle of oscillation of the load, as a function of time, with $kplin = 500$ and $kpang = 10$.

[1] https://playground.arduino.cc/Code/PIDLibrary/

Taking into account the transient response predicted through the simulations, the table in Fig. 7 shows not only the best set of parameters (manually tuned) for each criteria, but also the respective settling times for each of the measured output variables.



Criteria	PID parameters	Angle settling time (s)	Position settling time (s)
1 - Position	$Kp_{lin} = 700$ $Kp_{ang} = 35$ $Ki_{ang} = 10$	7,6	8,3
2 - Angle	$Kp_{lin} = 800$ $Kp_{ang} = 40$	8,5	6,6

Figure 7: Accommodation times for the most appropriate sets of parameter values, according to each one of the criteria

In what concerns the potential for improvement of the application of automatic control to crane operation, it is quite clear from the results on Fig. 8, where is shown that, for the open-loop operation of the same step input, the system takes more than 300 s to stop oscillations and to attain steady-state as opposed to less than 10 seconds when the control was in operation. Even if a full PID controller was implemented, in the results shown the derivative part was not applied, since its use rendered the control too sensitive to noisy rod angle measurements. As can be noticed in Fig. 8, the controlled system had a slightly slower settling time in the cart position (red and orange curves), when compared with the open-loop (maximum velocity, blue line). However, this is more than superseded by the long time the pendulum oscillation takes to decay, once the cart motion is stopped. It can be noticed, however, that except for the initial and final short periods, the pendulum keeps near vertical for the intermediate part of the journey.

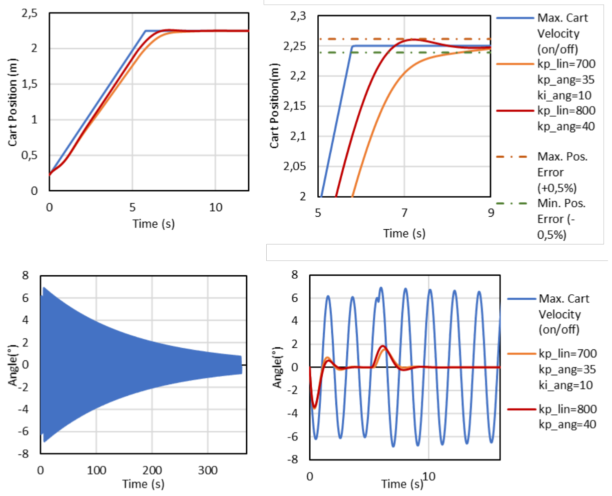


Figure 8: Simulation of the cart position (top) and the load sway angle (bottom) as a function of time, comparing the best results for each criteria.

6 Conclusions and future work

This work shows that the built test rig can be used as a testbed for controller studies applied to suspended load transportation systems, providing insight to the comparative results of different controlling algorithms, strategies and tuning situations.

Also, the models derived, even being simple linear models, are usable approaches within the range chosen for the angle variable, (mainly the maximum angular deviation of the rod from vertical) which, at 25 degrees, is a significantly large in real crane operation.

Finally, the potential for performance improvement of the application of closed-loop control on this type of systems is quite evident from these preliminary results.

The results obtained so far open the way for further comparative studies with other classes of controllers and tunning procedures as well as for more complex mechanical arrangements to approximate the rig to real-life situations. Near future addition to this study will be the replacement of the rigid rod with a cable, allowing the cable length to vary and to refine the models and control strategies to cope with these situations. These conclusions about the comparison of the system with and without control can be seen in the video available at `https://youtu.be/3w7Z_BNloHs`.

Funding

This work was partially supported by projects $UIBD/00308/2020$ and $UIDB/04-674/2020$, by FCT-Portuguese Foundation for Science and Technology

Referências

[1] Eihab M. Abdel-Rahman, Ali H. Nayfeh, Ziyad N. Masoud, Dynamics and control of cranes: A review, *Journal of Vibration and Control*, **9**(7):863–908, 2003, 10.1177/1077546303009007007.

[2] Khaled A. Alhazza, Asmahan H. Al-Shehaima, Ziyad N. Masoud, A continuous modulated waveform command shaping for damped overhead cranes, year, Volume Volume 4: 8th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A and B of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 08 2011, 10.1115/DETC2011-48336.

[3] H. C. Da Costa, Aplicação de técnicas de modelagem e controle em sistemas tipo ponte rolante, Master's thesis, Instituto Militar de Engenharia, Rio de Janeiro, 2010.

[4] Thomas Duriez, Steven L. Brunton, Bernd R. Noack, *Machine Learning Control - Taming Nonlinear Dynamics and Turbulence*, Springer Publishing Company, Incorporated, 1st edition, 2016.

[5] Jie Huang, Ehsan Maleki, William Singhose, Dynamics and swing control of mobile boom cranes subject to wind disturbances, *IET Control Theory & Applications*, **7**(9):1187–1195, 2013, 10.1049/iet-cta.2012.0957.

[6] Jeffrey T. Hubbell, Bruce Koch, Dennis McCormick, Modern crane control enhancements, in *Proc. of the Ports '92 Conference*, 1992.

[7] Ho-Hoon Lee, Modeling and Control of a Three-Dimensional Overhead Crane, *Journal of Dynamic Systems, Measurement, and Control*, **120**(4):471–476, 12 1998, 10.1115/1.2801488.

[8] Ho-Hoon Lee, Sung-Kun Cho, Jae-Sung Cho, A new anti-swing control of overhead cranes, *IFAC Proceedings Volumes*, **30**(13):115–120, 1997, ISSN 1474-6670, 10.1016/S1474-6670(17)44380-1. IFAC Workshop on Automation in the Steel Industry: Current Practice and Future Developments (ASI'97), Kyongju, Korea, 16-18 July 1997

[9] M. J. Maghsoudi, Z. Mohammed, A. F. Pratiwi, N. Ahmad, A. R. Husain, An experiment for position and sway control of a 3d gantry crane, in *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, year, Volume 2, 2012, 10.1109/ICIAS.2012.6306066.

[10] Liyana Ramli, Z. Mohamed, Auwalu M. Abdullahi, H.I. Jaafar, Izzuddin M. Lazim, Control strategies for crane systems: A comprehensive review, *Mechanical Systems and Signal Processing*, **95**:1–23, 2017, ISSN 0888-3270, 10.1016/j.ymssp.2017.03.015.

[11] Richard Sbresny, Andrew Getler, Nick Felker, Chris Frederickson, Implementation of an inverted pendulum pid control system using a stepper motor, 12 2016, 10.13140/RG.2.2.34564.12164.

[12] M. Iwan Solihin, Wahyudi, Sensorless anti-swing control strategy for automatic gantry crane system: Soft sensor approach, year, in *2007 International Conference on Intelligent and Advanced Systems*, 2007, 10.1109/ICIAS.2007.4658534.

[13] R.J. Vaccaro, *Digital Control: A State-space Approach*, New York: McGrawHill, 1st edition, 1995.

A multi-link finite-element modellation of heat conduction

by N. Alves, M. Belbut, P. Faria, J. Fatelo, P. Ferreira, T. Santos and N. Martins-ferreira

Centre for Rapid and Sustainable Product Development
Polytechnic Institute of Leiria
Authors email: nuno.alves@ipleiria; miguel.gaspar@ipleiria.pt; paula.faria@ipleiria.pt; jorge.fatelo@ipleiria.pt; pedro.castelo.ferreira@gmail.com; tiago.a.santos@ipleiria.pt; martins.ferreira@ipleiria.pt (corresponding author)

Abstract This is a preliminary work that explores future applications of mathematical structures such as multi-links to the heat conduction problem on discretized geometries with generic coordinates.
Keywords: Injection Molding, Warpage, Conformal Channels, Multi-Link, Thermal, Simulation.

1 Introduction

In the injection molding industry economic viability requires the injection process to be as fast and energetically efficient as possible with the objective of producing the largest number of parts in the minimum possible time. Hence the process optimization must ensure the process to be as fast as possible maintaining the desired part quality. In particular to address process optimization by mold design was suggested in [1, 2] that conformal channels would allow much lower heating and cooling times reducing the overall injection time significantly and simultaneously allowing for higher energy efficiency. This alternative channel system consists of a channel network distributed at a fixed distance from the mold cavity (plastic part). It was also clearly identified in the original studies that the distance of the channels to the mold cavity was one of the most relevant parameters affecting the cooling time. However it also affects temperature gradients across the part, hence increasing part warpage and shrinkage as the cooling time decreases. Hence, a compromise situation must be considered in the process optimization. Other relevant conclusions include the fact that square channels reduce temperature gradients however also increase mechanical deformation of the mold due to injection and packing pressures [3]. More recent studies show that a moderate decrease of both the injection time process $(5.8-6.8\%)$ and average part warpage is easily achievable [4, 5].

The present study proposes a simple algorithm for the generation of conformal channels that simultaneously minimizes the part warpage in the injection phase of cooling. The objective is to minimize temperature gradients as suggested for instance in [1, 2, 6, 8, 9, 10] instead of directly minimizing mechanical deformations as this is an equivalent solution being computationally less expensive.

It considers a cubic discretization similarly to [8] and explicitly consider square section channels as this particular shape uniformizes the heat distribution of the part cavity [1, 2, 3].

For a given closed system, the rate of heat transferred per unit volume at each point within the system is:

$$Q = -\nabla\cdot\mathbf{q}\ (W\,m^{-3})\ , \tag{1.1}$$

where $\mathbf{q}$ quantities denote 3-vectors and $\cdot$ the standard internal product. The heat transfer flux is given by Fourier's law of heat conduction

$$\mathbf{q} = -k\,\nabla T (W\,m^{-2})\ . \tag{1.2}$$

$T\,(K)$ is the temperature and $k\,(W\,m\,K^{-1})$ is the thermal conductivity. Hence we obtain

$$Q = \nabla\cdot(k\,\nabla T)\ . \tag{1.3}$$

From the overall energy balance of the system we obtain the heat equation

$$\rho\,c_p\dot{T} = \nabla\cdot(k\,\nabla T) + \dot{\varepsilon}\ , \tag{1.4}$$

where $\rho\,(kg\,m^{-3})$ is the mass density, $c_p\,(J\,kg^{-1}\,K^{-1})$ is the specific heat capacity at constant pressure and $\dot{\varepsilon}\,(W\,m^{-3})$ represents the rate of heat transfer per unit volume with the exterior of the system (a heat sink or heat source) [17].

For a generic Riemann manifold M the heat equation can be derived by minimizing the Dirichlet Energy using Simon's Asymptotic Theorem [12]. In covariant formulation for a 4-manifold with metric $g_{\mu\nu}$ and the standard connections $\Gamma^{\mu}{}_{\lambda\rho}$ [13] we obtain

$$\rho c_p\,\gamma\,\dot{T} = g^{ij}D_i\cdot(k\,D_jT) + \gamma\,\dot{\varepsilon}\ ,\ \gamma = \frac{d\tau}{dt} = \sqrt{g_{\mu\nu}\dot{x}^{\mu}\dot{x}^{\nu}}\ , \tag{1.5}$$

where we are employing the standard conventions, implicit summation over repeated indices, greek indices μ stand for full space-time coordinates $\mu = 0, 1, 2, 3$, arabic indices stand for space coordinates $i = 1, 2, 3$ and indices are lowered and raised by contraction with the metric tensor. As usual the derivative operator for 4-vectors is replaced by the covariant derivative $D_\mu X_\nu = \partial_\mu X_\nu + \Gamma_{\mu\nu\alpha}X^{\alpha}$. Explicitly writing D_i and simplifying the equation, the covariant formulation of the heat equation is obtained

$$\rho c_p\dot{T} = \frac{1}{\gamma}\,k\,\left[\partial_i k\partial_j T + \partial_i\partial_j T + kg^{\mu\nu}\Gamma_{ji\mu}\partial_\nu T\right] + \dot{\varepsilon}\ . \tag{1.6}$$

Aiming at a finite element discretization we further note that for a manifold with boundary $\partial M \neq 0$ this equation can be integrated over volume

$$\int_M \sqrt{-g}\,\gamma\,\rho c_p\,\dot{T} = \int_M \sqrt{-g}\,k\,g^{ij} D_i(kD_jT) + \int_M \sqrt{-g}\,\gamma\,\dot{\varepsilon}\,. \tag{1.7}$$

such that applying the divergence theorem we obtain

$$\int_M \sqrt{-g}\,\gamma\,\rho c_p\,\dot{T} = \int_{\partial M} \sqrt{-g}\,k\,g^{ij}\,\partial_i T\,n_j + \int_M \sqrt{-g}\,\gamma\,\dot{\varepsilon}\,. \tag{1.8}$$

where h stands for the pull-back metric on the manifold boundary ∂M.

2 The Simulation of the Heat Equation using Multi-Link Structure

Assuming a given finite element discretization for flat Euclidean coordinates let us define the following quantities to be constant inside each finite element x: temperature $T(x)$, density $\rho(x)$, capacity $c_p(x)$, conductivity $k(x)$, position $g(x)$ and volume $V(x)$. Hence integrating equation (1.4) over the finite element x of volume $V(x)$ with boundary $\partial V(x)$ and applying the divergence theorem we obtain

$$\int_{V(x)} \rho c_p \dot{T} = \int_{\partial V(x)} k\nabla T \cdot \mathbf{n} + \int_{V(x)} \dot{\varepsilon}\,. \tag{2.1}$$

Further defining the quantities $A(x,y)$ and $n(x,y)$ respectively as the area and average normal of the contact surface between adjacent finite elements x and y, for a given spatial discretization we obtain $V(x)\rho(x)c_p(x)\dot{T}(x)$ equals to

$$\sum_{\{y_x\}} k(x,y_x)(\nabla T(x)\cdot n(x,y_x))\,A(x,y_x) + V(x)\dot{\varepsilon}. \tag{2.2}$$

For the definition of the gradient at each finite element face let us consider the first order forward derivative $\nabla T(x)\big|^{(\text{discrete})}_{x\to y_x} = j(x,y_x)$ with

$$j(x,y_x) = \frac{T(y_x)-T(x)}{||\mathbf{g}(\mathbf{y_x})-\mathbf{g}(\mathbf{x})||}\,\frac{g(y_x)-g(x)}{||\mathbf{g}(\mathbf{y_x})-\mathbf{g}(\mathbf{x})||}\,. \tag{2.3}$$

Further considering a first-order discretization over time of time step Δt and introducing explicit temporal dependence [17, 18] on

$$\dot{T}(x)\big|^{(\text{discrete})} = \frac{T(x,t+\Delta t)-T(x,t)}{\Delta t}\,, \tag{2.4}$$

replacing the discretized expression in equation (2.2) and dividing the equation by the factor $V(x)\rho(x)c_p(x)$ we obtain the recursive discretized equation for the temperature T

$$T(x,t+\Delta t) = T(x,t) + \frac{\Delta t}{\rho(x)\,c_p(x)}\left(\frac{1}{V(x)}S + \dot{\varepsilon}_x\right) \tag{2.5}$$

with $S = \sum_{\{y_x\}} k(x,y_x)\,(j(x,y_x)\cdot n(x,y_x))\,A(x,y_x)$. Also note that generally $\rho(x)$, $c_p(x)$ and $k(x)$ varies with temperature. When this variation is implemented in the finite element model these quantities should be evaluated at each time step for the respective element temperature $T(x)$. For simplicity, we will use a first-order approximation for the values of each function ρ, c_p and k. Thus we will have, in the form of the iterative process (as we will see further on, the mapping $k(x,y_x)$ will be decomposed into a function of only one variable, so that knowing the value of $k(x,t)$ for every point x and instant t is enough to determine $k(x,y_x)$):

$$\rho(x,t+\Delta t) = \delta_\rho(x) + \lambda_\rho(T(x,t)-T_\rho) \tag{2.6}$$

$$c_p(x,t+\Delta t) = \delta_c(x) + \lambda_c(T(x,t)-T_c) \tag{2.7}$$

$$k(x,t+\Delta t) = \delta_k(x) + \lambda_k(T(x,t)-T_k) \tag{2.8}$$

where δ_ρ, λ_ρ, T_ρ, δ_c, λ_c, T_c, δ_k, λ_k and T_k are constants.

It is missing to evaluate $k(x,y_x)$ between adjacent finite elements x and y_x. Its value will depend whether the elements are from the same material or distinct materials. The following possibilities can be considered [14]:

1. same material or solid-solid:

$$k(x,y) = \frac{k(x)+k(y)}{2} \tag{2.9}$$

2. convection for solid-liquid:

$$k(x,y) = h||g(y)-g(x)|| \tag{2.10}$$

where the average for the same material is considered to account for the variation of $k(x)$ with temperature (generally $T(x)\neq T(y) \Leftrightarrow k(x)\neq k(y)$) and is also applicable to solid-solid interfaces. h is the convection coefficient at each solid-fluid interface and depends on the Reynold's number. The factor $||g(y)-g(x)||$ is considered to offset the discretized gradient as convection is proportional to the temperature difference $T(y)-T(y)$.

For the purpose of this work, and to have a simpler way for computing the result, we choose to make the value of k dependent on a single variable. By taking the inverse of k we obtain the resistivity [17] and it is reasonable to assume that its value on the interface of two different types of materials is given by its average value, that is,

$$k(x,y) = 2\left(\frac{1}{k(x)}+\frac{1}{k(y)}\right)^{-1} = \frac{2k(x)k(y)}{k(x)+k(y)}.$$

Let us consider a discretization in cubic elements such that for each cube there is exactly 6 adjacent cubes. Let us point out that the cubes can be degenerated or be constrained to different metrics. The elementary cubes will depend on the geometry, for example, the typical cases are cartesian, cylindrical and spherical. The density, the specific heat, the thermal conductivity, volume and rate of heat transfer are also assumed constant for each finite element. Nevertheless, in a general situation they may vary for distinct finite elements so that these quantities have a spatial index $\rho(x)$, $k(x)$, $c_p(x)$, $V(x)$, $\dot{\varepsilon}(x)$. The Multi-link structure was introduced in the context of 3D-printing [15] and to apply this mathematical structure to the heat conduction problem we need the following information:

1. A set of linear indexes X. Each linear index refers to one cube in the discrete volume. So, for example, $V(X)$ is the volume of the cube indexed by $x \in X$, whereas $T(x,t)$ is the temperature on the cube $x \in X$ at instant, or iteration, in time t.

2. Transition maps defining a cubic link $\alpha_i \colon X \to X, \quad i = 1, 2, 3$. This creates a cubic structure as explained in [15]. These maps are the ones that determine the topology of the volume. In other words, they control how each individual cube $x \in X$ is assembled into a volume by specifying its neighbours along each one of the three dimensions. Other variations are possible, as explained in [15]. Here we consider the simplest one but only for expository purposes.

3. Reverse transition maps $\alpha_j \colon X \to X, \quad j = 4, 5, 6$ such that α_4 is a reverse to α_1, α_5 is a reverse to α_2, and α_6 is a reverse to α_3, as illustrated

$$\begin{tikzcd}
 & \alpha_3(x) & \alpha_4(x) \\
\alpha_5(x) & x \arrow[l] \arrow[r] \arrow[u] \arrow[ur] \arrow[dl] \arrow[d] & \alpha_2(x) \\
\alpha_1(x) & \alpha_6(x) &
\end{tikzcd}$$

further details on this construction may be found in [16]. Here we simply observe that the six indices stand for $i = 1$: front, $i = 2$: left, $i = 3$: top, $i = 4$: rear, $i = 5$: right, $i = 6$: bottom.

4. A projection map, $m \colon X \to M \subset \mathbb{N}$ associating to each linear index $x \in X$ its type of material $M = \{1, 2, 3, \ldots, n_M\}$, with n_M the total number of different materials involved in the system to be modelled.

5. A vector of physical parameters for each type of material $p \colon M \to \mathbb{R}^{3\times 3}$ with $p(j)$ as

$$(\delta_\rho(j), \lambda_\rho(j), T_\rho(j), \delta_c(j), \lambda_c(j), T_c(j), \delta_k(j), \lambda_k(j), T_k(j))$$

having the initial conditions defining the maps displayed in (2.6), (2.7) and (2.8).

6. To each linear index $x \in X$ of the grid we will have an assignment of its volume, denoted by $V(x)$, and its source or sink generator, denoted by $\dot{\varepsilon}(x)$. These values may vary with time. An alternative to the definition of $V(x)$ is to compute an approximation using the geometrical information (see below).

7. A geometrical realization of the points on the cubic-link on the 3D-space $g \colon X \to \mathbb{R}^3$. We observe that applying the map g to each one of the 6 transition maps α_i we obtain the six possible directions from which there is a face going to the exterior of the cube as illustrated above. In particular, each $g(x) \in \mathbb{R}^3$ is the center of such a cube.

If we let $T(x,t)$ be the temperature density at the index point $x \in X$ and time $t \in \mathbb{N}$, on the cubic grid, then the vector which defines the temperature gradient along each of the 6 directions $i = 1:6$ of the multi-link structure is obtained as

$$j_i(x,t) = \frac{T\alpha_i(x) - T(x)}{\|g\alpha_i(x) - g(x)\|^2}\,(g\alpha_i(x) - g(x)) \ . \qquad (2.11)$$

The iteration formula, assuming the initial conditions $u(x,0)$ are defined, is given by $T(x, t{+}dt)$ equals to $T(x,t)$ plus $\frac{dt}{\rho(x,t)c_p(x,t)}$ times

$$\left(\frac{1}{V(x)}\sum_{i=1}^{6} k_i(x) j_i(x,t)\cdot n(F_i(x)) + \dot{\varepsilon}(x)V(x)\right) \qquad (2.12)$$

where $k_i(x) = k(x, \alpha_i(x))$ is given as in (2.10), $F_i(x)$, $i = 1, 2, 3, 4, 5, 6$ denotes the i-th face of the cube indexed by x, accordingly to the assignment (F_1, front), (F_2, left), (F_3, top), (F_4, rear), (F_5, right), (F_6, bottom) and the expression $j_i(x,t)\cdot n(F_i(x))$ denotes the $3D$ inner product of the vector $j_i(x,t)$ and the weighted average normal to the face $F_i(x)$ (strictly n is a vector with the direction of the normal and magnitude of the cube face area, hence with units m^2). The weighted average normal has to be considered because in general there is no guarantee that the face determined by the points in $F_i(x)$ is planar. An explicit description of the points in each $F_i(x)$, $i = 1, \ldots, 6$ is presented below, for the moment we explain how to get the average normal to a polygon with four arbitrary points. Let us take the explicit example of the face orthogonal to the direction of α_1, called the first direction for simplicity. If we let $a_1 = \alpha_1(x)$, $a_2 = \alpha_2\alpha_1(x)$, $a_3 = \alpha_3\alpha_1(x)$, $a_5 = \alpha_5\alpha_1(x)$ and $a_6 = \alpha_6\alpha_1(x)$ (note that the element $a_4 = \alpha_4\alpha_1(x)$ is equal to x, since α_4 is the reverse direction to α_1). These points may be illustrated as shown

$$\begin{tikzcd}
 & a_3(x) & x \arrow[dl] \\
a_5(x) & a_1(x) \arrow[l] \arrow[r] \arrow[u] \arrow[d] & a_2(x) \\
 & a_6(x) &
\end{tikzcd}$$

and the average normal vector, $n_1(x)$, to this face is computed as

$$\frac{1}{4}\left(\begin{vmatrix}\cdots\cdots\\ a_2-a_1\\ a_3-a_1\end{vmatrix}+\begin{vmatrix}\cdots\cdots\\ a_3-a_1\\ a_5-a_1\end{vmatrix}+\begin{vmatrix}\cdots\cdots\\ a_5-a_1\\ a_6-a_1\end{vmatrix}+\begin{vmatrix}\cdots\cdots\\ a_6-a_1\\ a_2-a_1\end{vmatrix}\right)\tag{2.13}$$

with the cross product $(a_j-a_1)\times(a_k-a_1)$ represented in its matrix form

$$\begin{vmatrix}\cdots\cdots\\ a_j-a_1\\ a_k-a_1\end{vmatrix}.$$

An easy calculation shows that the average normal vector orthogonal to the first direction reduces to the formula

$$n_1=\frac{1}{4}\begin{vmatrix}\cdots\cdots\\ a_2-a_5\\ a_3-a_6\end{vmatrix}=\frac{1}{4}(a_2-a_5)\times(a_3-a_6)$$

More precisely, using g to get the spatial coordinates, and denoting $g\alpha_i\alpha_j$ by g_{ij}, we have

$$n_1(x)=\frac{1}{4}(g_{21}(x)-g_{51}(x))\times(g_{31}(x)-g_{61}(x)).\tag{2.14}$$

In a similar way we obtain the average normal vectors to the other faces:

$$n_2(x)=\frac{1}{4}(g_{42}(x)-g_{12}(x))\times(g_{32}(x)-g_{62}(x))\tag{2.15}$$

$$n_3(x)=\frac{1}{4}(g_{43}(x)-g_{13}(x))\times(g_{23}(x)-g_{53}(x))\tag{2.16}$$

$$n_4(x)=\frac{1}{4}(g_{34}(x)-g_{64}(x))\times(g_{24}(x)-g_{54}(x))\tag{2.17}$$

$$n_5(x)=\frac{1}{4}(g_{35}(x)-g_{65}(x))\times(g_{45}(x)-g_{15}(x))\tag{2.18}$$

$$n_6(x)=\frac{1}{4}(g_{26}(x)-g_{56}(x))\times(g_{16}(x)-g_{46}(x))\tag{2.19}$$

Using a similar technique, when the volume of the cubes is not imposed, we may get an approximation to its value, $V(x)$, at each index element x in the grid as

$$\frac{1}{8}(\|g\alpha_1(x)-g\alpha_4(x)\|\|g\alpha_2(x)-g\alpha_5(x)\|\|g\alpha_3(x)-g\alpha_6(x)\|)\tag{2.20}$$

Before continuing and give a precise description of the overall procedure, let us draw our attention on the source or sink heat generation function $\dot{\varepsilon}(x)$ at each point $x\in X$ of the grid. As a consequence of the structure of a link, a point $x\in X$ for which $\alpha_j(x)=x$ for some direction $j=1:6$, is a border element. This means that the interface between our system and the outside environment can be modelled using $\dot{\varepsilon}$. Another aspect which is important is the modelling of a conformal channel with refrigeration liquid. Instead of simulating the fluid dynamics and how it correlates with the heat transfer, we will assume that at each time there is a heat source that is being extracted on the elements from the grid which are of that type of material.

3 The general procedure

We are now in a position to outline the general procedure for the iterative simulation:

1. Intput Data:

   (a) the input data consists on the structure of a cubic directed link, as explained before and it is a system (X,g,α_j), with X a set of indexes, $\alpha_j\colon X\to X$ the transition maps along the six possible (oriented) directions, and $g\colon X\to\mathbb{R}^3$ is the geometric realization map that associates to each element in the grid its spatial coordinates.

   (b) Types of material. In order to encode the information, to each different type of material a number is associated, and refereed as an index. This mean that we will have a set, such as for example $M=\{1,2,3,4,5,6\}$ where different numbers may not necessarily correspond to different materials a priori. This is useful to change the type of material after the process has started without the need to reformulate the whole iterative process. We will use, as an example, the three materials Steel, PVC and Water, but we will allow six indexes for the materials in our system.

   (c) Assigning the materials to the elements on the grid. This is done with a projection map $m\colon X\to M$, which specifies the type of material at each position on the grid.

   (d) Material properties. This is an assignment of the needed parameters to specify each type of material. It is done with a mapping $p\colon M\to\mathbb{R}^{3\times3}$, specifying to each material index $j\in M$, $p(j)$ as explained on item 5 above as well as in equations (2.6), (2.7), (2.8).

   (e) The source and sink heat generator function. This functionality will be implemented in a future version of this work, for the moment we are considering the theoretical structure and its main properties.

   (f) The total number of iterations, say N_{Max} and the time interval dt and initial temperature $T(x,t_0)$.

2. The iterative process:

   (a) Suppose the first iterative step is determined, then we have

$$\begin{aligned}T(x,t+dt)\quad&=\quad T(x,t)+\frac{dt}{\rho(x,t)c_p(x,t)}\cdot\\&\qquad\cdot\left(\frac{1}{V(x)}\sum_{i=1}^{6}K+\dot{\varepsilon}(x,t)V(x)\right)\end{aligned}$$

with $K = k_i(x,t)j_i(x,t) \cdot n_i(x)$ where $\rho(x,t)$ and $c_p(x,t)$ are determined as in equations (2.6) and (2.7), $V(x)$ is determined as in (2.20), $k_i(x)$ is given by the formula

$$k_i(x) = \frac{2k(x,t)k(\alpha_i(x),t)}{k(x,t) + k(\alpha_i(x),t)}$$

with $k(x,t)$ determined by the equation (2.8). The gradient, $j_i(x,t)$ is obtained by the formula (2.11). The average normal vectors $n_i(x)$ are independent of time (at least in a first implementation where the geometry is not changing) and are given by the formulas (2.14)–(2.19). The value of the map $\varepsilon(x,t)$ should be defined accordingly to the physical interpretation of the problem.

3. The metric associated to the simulation. At the end of the iterative process we measure the matrix norm of the matrix $j(x,t)$ and give a measure of the performance of the simulation.

4 Conclusion

Conceptually, this work presents a new approach with direct impact and specific practical application regarding the heat transfer applied to the mold industry. The proposed method has the great advantage of being numerically less demanding both in terms of required processing power and in terms of processing time, being of great value for an evaluation and selection of the most promising molds for a specific application. This approach can also be applied in thermal mapping of 3D printing structures. Furthermore, the notion of a multi-link permits to work with a subset of indexes $I \subset X$ thus allowing to use only those elements which suffer some changes form one iteration to the following, thus, since the subset I can be dynamically adapted from one iteration to the following, it gives an optimized way for the whole process from a computational point of view.

Acknowledgments

Work supported by grant SFRH/BPD/34566/2007 up to January 2014 and by project CENTRO-01-0145-FEDER-000014 from August 2017 onwards. It was also funded by FCT/MCTES (PIDDAC) through the following Projects: Associate Laboratory ARISE LA/P/0112/2020; UIDP/04044/2020; UIDB/04044/2020; PAMI–ROTEIRO/0328/2013 (Nº 022158); MATIS (CENTRO-01-0145-FEDER-000014 - 3362); CENTRO-01-0247-FEDER-(069665, 039969); POCI-01-0247-FEDER-(069603, 039958, 039863, 024533); Generative thermodynamic; by CDRSP and ESTG from the Polytechnic of Leiria.

Referências

[1] E. M. Wylonis, *Production of injection molding tooling with conformal cooling channels using the three dimensional printing process*, Master of Science MIT 1995.

[2] E. Sachs, E. Wyloms, S. Allev, M. Cima and H. Guo, *Production of Injection Molding Tooling With Conformal Cooling Channels Using the Three Dimensional Printing Process*, Polymer Engineering and Science **40** (2000) 1232.

[3] X. Xu, E. Sachs and S. Allen, *The Design of Conformal Cooling Channels in Injection Molding Tooling*, Polymer Engineering and Science **41** (2001) 1265.

[4] Z. Shayfull, S. Sharif, A. Mohd Zain, R. Mohd Saad and M. A. Fairuz, *Milled Groove Square Shape Conformal Cooling Channels in Injection Molding Process*, Materials and Manufacturing Processes **28** (2013) 884–891.

[5] S. Marques, A. Fagali de Souza, J. Miranda and I. Yadroitsau, *Design of conformal cooling for plastic injection moulding by heat transfer simulation*, Polímeros **25** (2015) 564-574.

[6] A. Agazzi, V. Sobotka, R. Le Goff, D. Garcia and Y. Jarny, *A Methodology for the Design of Effective Cooling System in Injection Moulding*, Int. J. Mater. Form. **3** (2010) 13–16.

[7] J. M. Jauregui-Beckera, G. Tosellob, F. J. A. M. van Houtena and H. N. Hansenb, *Performance evaluation of a software engineering tool for automated design of cooling systems in injection moulding*, Procedia CIRP **7** (2013) 270–275.

[8] J. M. Jauregui-Becker, G. Tosello, F. J. A. M. van Houten and H. N. Hansen, *Performance evaluation of a software engineering tool for automated design of cooling systems in injection moulding*, Procedia CIRP **7** (2013) 270–275.

[9] A. Agazzi, V. Sobotka, R. Le Goff and Y. Jarny, *Optimal cooling design in injection moulding process - A new approach based on morphological surfaces*, Applied Thermal Engineering **52** (2013) 170-178.

[10] K. M. Au and K. M. Yu, *Variable Distance Adjustment for Conformal Cooling Channel Design in Rapid Tool*, Journal of Manufacturing Science and Engineering **136** (2014) 044501.

[11] J.-F. Agassan, P. Avenas, J.Ph. Sergent and P. J. Carreau, *Polymer Processing: Principles and Modeling*, Hanser Publishers ISBN 3-446-14584-2, ISBN 0-19-520864-1.

[12] P.B. Gilkey, *Invariance Theory: The Heat Equation and the Atiah-Singer index theorem*, CRC Press 1995, ISBN 0-8493-7874-4; B. Chow, P. Lu and L.Ni, *Hamilton's Ricci Flow*, Graduate Studies In Mathematics 77, American Mathematical Society, Science Press 2010, ISBN 0-821-88399-2.

[13] C. W. Misner, K. S. Thorne and J. A. Wheeler, *Gravitation*, W. H. Freeman and Company 1973, ISBN 0-7167-0334-3.

[14] G. P. Nikishkov, *Programming Finite Elements in Java*, Springer-Verlag London 2010, ISBN 978-1-84882-972-5.

[15] N. Martins-Ferreira, *The notion of multi-link, its applications and examples*, Scripta-Ingenia **7** (2016) 14.

[16] N. Martins-Ferreira and G. Mitchell, *Cubic-link structures via npcircles and gyroids*, CDRSP-IPLeiria Technical Report, GTLab(CubesApp-2) **81** (2017) 1-24.

[17] Incropera, F.P.; DeWitt, D.P.; Bergman, L.T.; Lavine., A.S. Fundamentals of Heat and Mass Transfer; 6th ed.; John Wiley Sons, 2007.

[18] Chapra, S.C. Applied Numerical Methods with MATLAB for Engineers and Scientists; Lange, M., Peter E. Massar, Lorraine Buczek, Eds.; 3rd ed.; Raghothaman Srinivasan: New York, USA, 2012; ISBN 978-0-07-339796-2.

A brief survey on MV-Algebras

by J. P. Fatelo and N. Martins-Ferreira

Centre for Rapid and Sustainable Product Development
Polytechnic Institute of Leiria
Author email: jorge.fatelo@ipleiria.pt; martins.ferreira@ipleiria.pt

Abstract We recall the notion of MV-algebra and present some of its properties, that are well known, with proofs that are not easily found in the literature. Our purpose is to have a common background in order to understand some aspects of the interplay between MV-algebras, De Morgan algebras, Boolean algebras and the new notion of mobi algebras which have recently been introduced by the authors. In particular, the usual derived operations on MV-algebras seem to be relevant to a better understanding of derived operations on mobi algebras.

1 Introduction

The concept of an MV-algebra was introduced by C. C. Chang [5], in 1958, as an axiomatization of the Lukasiewicz many-valued logic (MV stands for many-valued). In the same way as Boolean algebras stand to boolean logic, MV-algebras stand to Lukasiewicz infinite-valued logic. A variable in boolean propositional logic represents a $\{0,1\}$-valued observable, and identically transforms the output of this observable into a truth-value. In infinite-valued logic a variable transforms the output of a real-valued bounded observable into a truth-value lying in the unit real interval $[0,1]$. The completeness theorem for Lukasiewicz logic states that the rules of Modus Ponens and substitution are sufficient to obtain all tautologies (i.e., all equations of the form $\tau = \neg 0$ for τ an MV-term) in the infinite-valued calculus of Lukasiewicz starting from a few basic tautologies (originally due to Lukasiewicz) corresponding to the defining equations of MV-algebras. The need for infinitely many truth-values naturally arises, e.g., in the Rényi-Ulam game of Twenty Questions where some of the answers may be erroneous. Here answers do not obey classical two-valued logic. As a matter of fact, two equal answers to the same repeated question usually give more information than a single answer. Using Chang completeness theorem, it is possible to show that the underlying logic of Rényi-Ulam games is Lukasiewicz infinite-valued propositional logic (see e.g. [13]).

2 MV-algebras

Recall that an MV-algebra (see e.g. [1, 4]) is a structure of type $(2,1,0)$ which may be defined as follows.

Definition 1. *A MV-algebra is a system* $\mathbf{X}= (X,\cdot,',1)$ *such that:*

(MV1) $x\cdot(y\cdot z)=(x\cdot y)\cdot z$

(MV2) $x\cdot 1=x$

(MV3) $x''=x$

(MV4) $x\cdot 1'=1'$

(MV5) $(x'\cdot y)'\cdot y=(y'\cdot x)'\cdot x$

A list of properties is presented next. Note that, originally, the commutativity of the binary operation $\cdot$ was included as an axiom. The proof of Property **(P4)** below, showing that commutativity of $\cdot$ is a consequence of the other axioms, is due to Kolařík [12].

Proposition 1. *Let* $(X,\cdot,',1)$ *be a MV-algebra. It follows that:*

(P1) $1\cdot x=x$

(P2) $x'\cdot x=1'$

(P3) $(y\cdot x)'\cdot(x\cdot y)=1'$

(P4) $x\cdot y=y\cdot x$

Proof. We begin with the proof of **(P1)**:

$$\begin{aligned}
1\cdot x &= ((1\cdot x)')' && \textbf{(MV3)}\\
&= ((1\cdot x)'\cdot 1)'\cdot 1 && \textbf{(MV2)}\\
&= (1'\cdot(1\cdot x))'\cdot(1\cdot x) && \textbf{(MV5)}\\
&= ((1'\cdot 1)\cdot x)'\cdot(1\cdot x) && \textbf{(MV1)}\\
&= (1'\cdot x)'\cdot(1\cdot x) && \textbf{(MV2)}\\
&= ((1'\cdot x)'\cdot 1)\cdot x && \textbf{(MV1)}\\
&= (1'\cdot x)'\cdot x && \textbf{(MV2)}\\
&= (x'\cdot 1)'\cdot 1 && \textbf{(MV5)}\\
&= x'' && \textbf{(MV2)}\\
&= x. && \textbf{(MV3)}
\end{aligned}$$

Using **(P1)**, the complement property **(P2)** is easily proved:

$$\begin{aligned}
x'\cdot x &= (1\cdot x)'\cdot x && \textbf{(P1)}\\
&= (1''\cdot x)'\cdot x && \textbf{(MV3)}\\
&= (x'\cdot 1')'\cdot 1' && \textbf{(MV5)}\\
&= 1''\cdot 1' && \textbf{(MV4)}\\
&= 1\cdot 1' && \textbf{(MV3)}\\
&= 1'. && \textbf{(MV4)}
\end{aligned}$$

Now, we can use **(P2)** to prove **(P3)**:

$$\begin{aligned}
(y \cdot x)' \cdot (x \cdot y) &= ((y \cdot x)' \cdot x) \cdot y && \textbf{(MV1)}\\
&= ((y'' \cdot x)' \cdot x) \cdot y && \textbf{(MV3)}\\
&= ((x' \cdot y')' \cdot y') \cdot y && \textbf{(MV5)}\\
&= (x' \cdot y')' \cdot (y' \cdot y) && \textbf{(MV1)}\\
&= (x' \cdot y')' \cdot 1' && \textbf{(P2)}\\
&= 1'. && \textbf{(MV4)}
\end{aligned}$$

Properties **(P1)** and **(P3)** are used to prove **(P4)**:

$$\begin{aligned}
x \cdot y &= 1 \cdot (x \cdot y) && \textbf{(P1)}\\
&= (1')' \cdot (x \cdot y) && \textbf{(MV3)}\\
&= ((y \cdot x)' \cdot (x \cdot y))' \cdot (x \cdot y) && \textbf{(P3)}\\
&= ((x \cdot y)' \cdot (y \cdot x))' \cdot (y \cdot x) && \textbf{(MV5)}\\
&= 1'' \cdot (y \cdot x) && \textbf{(P3)}\\
&= y \cdot x. && \textbf{(MV3)}, \textbf{(P1)}
\end{aligned}$$

3 Derived Distributive Lattice

It is well known that a distributive lattice structure is obtained from an MV-algebra through a set of derived operations as explained in the following propositions.

Proposition 2. *Let $(X, \cdot, ', 1)$ be a MV-algebra. For $x \triangle y = (x' \cdot y)' \cdot y$, the following properties hold:*

(P5) $x \triangle y = y \triangle x$

(P6) $x \triangle x = x$

(P7) $x \triangle (y \triangle z) = (x \triangle y) \triangle z$

(P8) $x \triangle 1 = x$

(P9) $x \triangle 1' = 1'$

Proof. Commutativity **(P5)** is just a rewriting of **(MV5)**. Idempotency **(P6)** is easily proved:

$$\begin{aligned}
x \triangle x &= (x' \cdot x)' \cdot x\\
&= (1')' \cdot x && \textbf{(P2)}\\
&= 1 \cdot x && \textbf{(MV3)}\\
&= x. && \textbf{(P1)}
\end{aligned}$$

Associativity **(P7)** can be proved as follows:

$$\begin{aligned}
x \triangle (y \triangle z) &= x \triangle (z \triangle y) && \textbf{(P5)}\\
&= (x' \cdot [(z' \cdot y)' \cdot y])' \cdot [(z' \cdot y)' \cdot y]\\
&= (x' \cdot [y \cdot (z' \cdot y)'])' \cdot [(z' \cdot y)' \cdot y] && \textbf{(P4)}\\
&= [(x' \cdot y) \cdot (z' \cdot y)']' \cdot (z' \cdot y)' \cdot y && \textbf{(MV1)}\\
&= [(x' \cdot y)'' \cdot (z' \cdot y)']' \cdot (z' \cdot y)' \cdot y && \textbf{(MV3)}\\
&= [(z' \cdot y)'' \cdot (x' \cdot y)']' \cdot (x' \cdot y)' \cdot y && \textbf{(MV5)}\\
&= [(z' \cdot y) \cdot (x' \cdot y)']' \cdot (x' \cdot y)' \cdot y && \textbf{(MV3)}\\
&= [z' \cdot y \cdot (x' \cdot y)']' \cdot (x' \cdot y)' \cdot y && \textbf{(MV1)}\\
&= [z' \cdot (x' \cdot y)' \cdot y]' \cdot (x' \cdot y)' \cdot y && \textbf{(P4)}\\
&= [z' \cdot (x \triangle y)]' \cdot (x \triangle y)\\
&= z \triangle (x \triangle y)\\
&= (x \triangle y) \triangle z. && \textbf{(P5)}
\end{aligned}$$

For unit relation **(P8)**, we have:

$$\begin{aligned}
x \triangle 1 &= (x' \cdot 1)' \cdot 1\\
&= x'' && \textbf{(MV2)}\\
&= x. && \textbf{(MV3)}
\end{aligned}$$

Finally, here is the proof of **(P9)**:

$$\begin{aligned}
x \triangle 1' &= (x' \cdot 1')' \cdot 1'\\
&= 1' && \textbf{(MV4)}
\end{aligned}$$

Now we need to introduce the dual operation.

Proposition 3. *Let $(X, \cdot, ', 1)$ be a MV-algebra. For $x \triangledown y = (y' \triangle x')'$, the following properties hold:*

(P10) $x \triangle y = (y' \triangledown x')'$

(P11) $x \triangledown y = y \triangledown x$

(P12) $x \triangledown x = x$

(P13) $x \triangledown (y \triangledown z) = (x \triangledown y) \triangledown z$

(P14) $1' \triangledown x = x$

(P15) $1 \triangledown x = 1$

Proof. Because $'$ is an involution, there is a duality between $\triangle$ and $\triangledown$:

$$\begin{aligned}
(y' \triangledown x')' &= (x'' \triangle y'')''\\
&= x \triangle y. && \textbf{(MV3)}
\end{aligned}$$

Then, of course, **(P11)**, **(P12)**, **(P13)**, **(P14)** and **(P15)** follow from Proposition 2.

Absorption rules connecting $\triangle$ and $\triangledown$ are easily proved.

Proposition 4. *Let $(X, \cdot, ', 1)$ be a MV-algebra. The following properties hold:*

(P16) $(x \triangle y) \cdot x' = 1'$

(P17) $x \bigtriangledown (x \bigtriangleup y) = x$

(P18) $(y \bigtriangledown x) \bigtriangleup x = x$

Proof. Property **(P16)** simplifies the proof of the absorption rules **(P17)** and **(P18)**:

$$\begin{aligned}(x \bigtriangleup y) \cdot x' &= ((x' \cdot y)' \cdot y) \cdot x' \\ &= (x' \cdot y)' \cdot (y \cdot x') && (\mathbf{MV1}) \\ &= 1'. && (\mathbf{P3})\end{aligned}$$

Then

$$\begin{aligned}x \bigtriangledown (x \bigtriangleup y) &= ((x \bigtriangleup y)' \bigtriangleup x')' \\ &= (((x \bigtriangleup y) \cdot x')' \cdot x')' \\ &= (1'' \cdot x')' && (\mathbf{P16}) \\ &= x, && (\mathbf{MV3}), (\mathbf{P1})\end{aligned}$$

and

$$\begin{aligned}(y \bigtriangledown x) \bigtriangleup x &= (x' \bigtriangleup y')' \bigtriangleup x \\ &= ((x' \bigtriangleup y') \cdot x)' \cdot x \\ &= 1'' \cdot x && (\mathbf{MV3}), (\mathbf{P16}) \\ &= x. && (\mathbf{MV3}), (\mathbf{P1})\end{aligned}$$

Distributivity of $\bigtriangleup$ and $\bigtriangledown$ over each other can also be proved.

Proposition 5. *Let $(X, \cdot, ', 1)$ be a MV-algebra. Then:*

(P19)
$$\begin{aligned}x \bigtriangledown (y \bigtriangleup z) &= (x \bigtriangledown y) \bigtriangleup (x \bigtriangledown z) \\ x \bigtriangleup (y \bigtriangledown z) &= (x \bigtriangleup y) \bigtriangledown (x \bigtriangleup z).\end{aligned}$$

According to [2], the next proposition is *folklore* of the theory of MV-algebras.

Proposition 6. *Let $\mathbf{X} = (X; \cdot, ', 1)$ be a MV-algebra. Define $x \leq y$ if and only if $x \cdot y' = 1'$. Then $(X; \leq)$ is a distributive lattice with the least element $1'$ and the greatest element 1 in which $x \wedge y = (x' \cdot y)' \cdot y$ and $x \vee y = (y' \wedge x')'$.*

Proof. Cited proof is [4].

The following results are related.

Proposition 7. *[4] Let $(X, \cdot, ', 1)$ be a MV-algebra. Then:*

(P20)
$$\begin{aligned}x \cdot (y \bigtriangledown z) &= (x \cdot y) \bigtriangledown (x \cdot z) \\ x \circ (y \bigtriangleup z) &= (x \circ y) \bigtriangleup (x \circ z).\end{aligned}$$

Where $x \circ y = (y' \cdot x')'$ is the dual operation of $\cdot$.

Some properties of the dual operation $\circ$ are presented.

Proposition 8. *Let $(X, \cdot, ', 1)$ be a MV-algebra. For $x \circ y = (y' \cdot x')'$, the following properties hold:*

(P21) $x \cdot y = (y' \circ x')'$

(P22) $x \circ y = y \circ x$

(P23) $x \circ (y \circ z) = (x \circ y) \circ z$

(P24) $1' \circ x = x$

(P25) $1 \circ x = 1$

4 Relation with Mobi Algebras

Consider a system $(A, p, 0, 1)$ where $p \colon A \times A \times A \to A$ is a ternary operation satisfying the following axioms:

(T1) $p(0, a, 1) = a$

(T2) $p(a, b, a) = a$

(T3) $p(a, p(b_1, b_2, b_3), c) = p(p(a, b_1, c), b_2, p(a, b_3, c))$

(T4) $p(a, 0, b) = a = p(b, 1, a)$.

This set of axioms was considered in [5, 6] for a characterization of Boolean algebras and it is related with the notion of mobi algebras introduced in [11] (see also [7, 8, 9, 10]).

It is interesting to observe the similarity between the derived operations from an MV-algebra, which give rise to a distributive lattice and hence can be specialized to De Morgan algebras or Boolean algebras, with the derived operations obtained form the ternary operation p as above, namely the ones $\bar{a} = p(1, a, 0)$, $a \cdot b = p(0, a, b)$ and $a \circ b = p(a, b, 1)$, as considered e.g. in [8]. Nevertheless, other possibilities of deriving binary operations from the ternary term p, such as $p(a, b, b)$ or $p(a, a, b)$ seem to be relevant as well. This will be explored in future work.

Acknowledgements This work is supported by the Fundação para a Ciência e a Tecnologia (FCT) and Centro2020 through the Project references: UID/-Multi/04044/2019; PAMI - ROTEIRO/0328/2013 (Nº 022158); Next.parts (17963), and also by CDRSP and ESTG from the Polytechnic Institute of Leiria.

Referências

[1] M. Bergmann, *An introduction to many-valued and fuzzy logic*, Cambridge University Press (2008).

[2] I. Chajda, R. Halš, J. Kühr, *Distributive lattices with sectionally antitone involutions*, Acta Scientiarum Mathematicarum **71**(1-2) (2005) 19-33.

[3] C. C. Chang, *Algebraic analysis of many-valued logic*, Trans. Amer. Math. Soc. **88** (1958) 467-490.

[4] R.L.O. Cignoli, I.M.L. D'Ottaviano, D. Mundici, *Algebraic foundations of many-valued reasoning*, Trends in Logic (2000).

[5] J. P. Fatelo, N. Martins-Ferreira, *A new look at ternary Boolean algebras*, ArXiv:2109.06259 (2021).

[6] J. P. Fatelo, N. Martins-Ferreira, *A refinement of ternary Boolean algebras*, ArXiv:2203.08012 (2022).

[7] J. P. Fatelo and N. Martins-Ferreira, *A connection between unitary rings and mobi algebras*, Scripta-Ingenia **10** (December) (2021) 3–4.

[8] J. P. Fatelo and N. Martins-Ferreira, *Boolean algebras and mobi algebras*, Scripta-Ingenia **10** (December) (2021) 5–12.

[9] J. P. Fatelo and N. Martins-Ferreira, *Mobi spaces and geodesics for the N-sphere*, Cah. Topol. Géom. Différ. Catég. **63** (1) (2022) 59–88.

[10] J. P. Fatelo and N. Martins-Ferreira, *Affine mobi spaces*, Boll Unione Mat Ital **15** () (2022) 589–604.

[11] J. P. Fatelo and N. Martins-Ferreira, *Mobi algebra as an abstraction to the unit interval and its comparison to rings*, Communications in Algebra **47** (3) (2019) 1197–1214.

[12] M. Kolařík, *Independence of the axiomatic systems for a MV-algebras*, Math. Slovaca **63** (2013) 1-4.

[13] D. Mundici, *MV-algebras, a short tutorial*, May 26, 2007
`https://www.matematica.uns.edu.ar/IXCongresoMonteiro/Comunicaciones/Mundici_tutorial.pdf`

On the winding number of a complex-link and its applications to morphological operations

by Nelson Martins-Ferreira

Centre for Rapid and Sustainable Product Development
Polytechnic Institute of Leiria
Author email: martins.ferreira@ipleiria.pt

keywords: Offsets, morphological operations, complex-link, winding number, symmetric difference.

Abstract The main goal of this paper is to provide an efficient and high level procedure to compute morphological operations for regions on the complex plane. The concept of a complex link is used to model the boundary of a region. The heart of the procedure lies on the partition of the complex plane, that is indexed by the integers and is obtained by extending the winding number of a curve (in this case the geometric realization of a link) from its interior points to the points in its boundary. Once this partition is obtained, it provides a straightforward method to compute intersections, unions, symmetric differences, dilations, contractions, and even more generally, it permits the computation of non-constant offsets defined independently at each point on the boundary of the region. This possibility is of particular importance in 3D printing with layer by layer fabrication, as well as in the generation of conformal cooling channels that are relevant to the mould industry.

1 Introduction

Every planar region gives rise to a closed and oriented simple curve, namely its boundary. This process can be reversed and every simple planar curve determines a well defined planar region. However, when the planar curve is not simple (i.e. has self intersections) there is an ambiguity in defining the planar region. This problem is solved by using a link structure [1] to model a closed and oriented planar curve. The structure of a link has some desirable properties which are suitable for computational implementations. In particular it gives rise to algorithms which provide mathematical consistency. In this paper we describe a procedure that can be used for resolving ambiguous regions on the plane. It assigns a partition of the plane defined by non-simple planar curves to every link structure. This procedure has particular cases which can be used in computing offsets and other morphological operations, such as union, intersection or symmetric difference.

Regions on the complex plane which are defined by a closed and oriented planer curve are identified with their boundary curve. When the curve is simple the region is obtained without any ambiguity, however, when the curve is not simple there is possibly an ambiguity in determining the region. This is a real problem since, for example, the offset of a simple curve is not necessarily a simple curve. And yet, the offset of a region should again be a region. This creates the problem of resolving ambiguities in regions which are defined as having the boundary of a not necessarily simple curve. The notion of a link [1] is used as a model for a continuous closed and oriented planar curve. We call it a complex-link and identify the euclidean plane with the complex numbers. The notion of a complex-link, as a mathematical structure (consisting of an indexing set, an endomap of indexes and a realization map into the complex numbers, see [1] for more details) has certain desirable properties. For example, it is an efficient way of encoding a planar curve in a clear and concise way (see the editorial article *A study of the letter A* at the opening of this newsletter). It is suitable for practical computational calculations as well as to produce mathematical consistency. On the top of the mathematical structure of a complex-link, we derive a general procedure that assigns to every such structure a classifying map. This map creates a partitioning of the complex plane into a family of regions indexed by the integers. The region labeled by 0 is unlimited, a region labeled by $n+1$ is contained into a region labeled by n. In particular, the partition can be transferred to the indexing sets and the ambiguity is resolved by choosing the appropriate indexing family. For example, in the case of offsets, the resulting region is the one whose boundary is the indexing family indexed by 0, if the region is limited, or -1 is the complement of the region is limited. Applications to other morphological operations are also derivable from this general procedure. For instance, the procedure outlined in [2] is covered by this approach.

Let us see some preliminary details on how to extract a link structure from a planar curve and how the notion of winding number is transferred to that structure.

Let $\gamma\colon [0,1] \to \mathbb{C}$ be a closed curve on the complex plane and suppose that we are given a sequence of ordered real numbers on the unit interval, say $0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq 1$. We may consider a discretisation for γ as:

1. $A = \{1, 2, \ldots, n\}$
2. $\varphi(k) = k+1$ if $k < n$, and $\varphi(n) = 1$

3. $g(k) = \gamma(t_k)$

If $z \in \mathbf{C}$ is not of the form $z = g(k)$ for some $k \in A$, we define the *winding number* $w(z)$ as:

$$w(z) = \frac{1}{2\pi} \sum_{k \in A} \angle \left(\frac{g(\varphi(k)) - z}{g(k) - z} \right) \tag{1.1}$$

with $-\pi < \angle(z) \leq \pi$ the usual phase angle of a complex number $z \in \mathbb{C}$ measured in radians. It will be important to understand the behaviour of the formula above when z is of the form $g(k) + t(g(\varphi(k)) - g(k))$ for some $k \in A$ and $t \in]0, 1[$. For that reason we present here three simple examples which are at the same time easy to compute by hand and clearly illustrate the formula above for the winding number of a point which is not coincident with a vertex used in the discretisation of a curve γ.

Let us take the simple example of the unit circle centred at zero. So, we take the curve $\gamma\colon [0, 1] \to \mathbb{C}$, defined as $\gamma(t) = e^{it2\pi}$, with i the imaginary unit and $t \in [0, 1]$. Let us now consider three different discretizations and analyse the winding number $w(0)$ for each one. In each case we consider a discretization with only three elements with the set of indexes $A = \{1, 2, 3\}$. As above, we also consider the re-indexing map $\varphi\colon A \to A$ defined as $\varphi(1) = 2$, $\varphi(2) = 3$ and $\varphi(3) = 1$. Then, for any $0 \leq t_1 < t_2 < t_3 \leq 1$ we have $g(k) = e^{it_k 2\pi}$ and we may compute the winding number at $z = 0$. Recall that the formula is

$$w(0) = \frac{1}{2\pi} \sum_{k \in A} \angle \left(\frac{g(\varphi(k))}{g(k)} \right)$$

and it will depend on the discretization we choose:

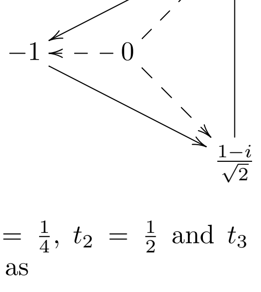
1. if we take $t_1 = \frac{1}{8}$, $t_2 = \frac{1}{2}$ and $t_3 = \frac{7}{8}$, then we compute $w(0)$ as

$$\frac{1}{2\pi} \left(\angle \left(e^{i\left(\frac{1}{2} - \frac{1}{8}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{7}{8} - \frac{1}{2}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{1}{8} - \frac{7}{8}\right)2\pi} \right) \right)$$

which simplifies to

$$\frac{1}{2\pi} \left(\frac{3}{4}\pi + \frac{3}{4}\pi - \frac{3}{2}\pi \right)$$

and we get $w(0) = 1$. The following diagram illustrates the situation.



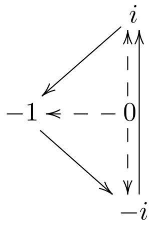
2. if we take $t_1 = \frac{1}{4}$, $t_2 = \frac{1}{2}$ and $t_3 = \frac{3}{4}$, then we compute $w(0)$ as

$$\frac{1}{2\pi} \left(\angle \left(e^{i\left(\frac{1}{2} - \frac{1}{4}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{3}{4} - \frac{1}{2}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{1}{4} - \frac{3}{4}\right)2\pi} \right) \right)$$

which simplifies to

$$\frac{1}{2\pi} \left(\frac{\pi}{2} + \frac{\pi}{2} + \angle \left(e^{-i\pi} \right) \right)$$

and since $\angle \left(e^{-i\pi} \right) = \pi$ we get $w(0) = 1$. The following diagram illustrates the situation.



We observe that if the definition of $\angle$ would be such that $-\pi \leq \angle(z) < \pi$ then we would have $w(0) = 0$.

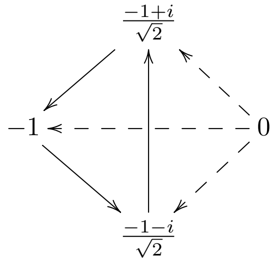
3. if we take $t_1 = \frac{3}{8}$, $t_2 = \frac{1}{2}$ and $t_3 = \frac{5}{8}$, then we compute $w(0)$ as

$$\frac{1}{2\pi} \left(\angle \left(e^{i\left(\frac{1}{2} - \frac{3}{8}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{5}{8} - \frac{1}{2}\right)2\pi} \right) + \angle \left(e^{i\left(\frac{3}{8} - \frac{5}{8}\right)2\pi} \right) \right)$$

which simplifies to

$$\frac{1}{2\pi} \left(\frac{\pi}{4} + \frac{\pi}{4} - \frac{\pi}{2} \right)$$

and we get $w(0) = 0$. The following diagram illustrates the situation.



2 Some useful operations on the structure of a link

The notion of a link was introduced in [1] and it is a generalization of a directed graph. A link is a triple (A, g, φ) where A is a finite set, called the set of indexes (and so the elements in A are called the indexes of the link), a map $g\colon A \to B$ whose domain is A, and admitting any set B as codomain (usually it will be a structured set, for example we will say real link when B is the set of the reals and comlex-link when $B = \mathbb{C}$, the complex numbers, which will also be identified with the plane, so that we may also say planar-link), and $\varphi\colon A \to A$ is a bijection on the set of indexes (in some cases we do not require it to be a bijection, however, that will not be the case in this paper). Every link gives rise to a directed graph of the form

$$A \overset{d}{\underset{c}{\rightrightarrows}} B \qquad (2.1)$$

where $d = g$ and $c = g\varphi$. See [1] for more details.

While comparing (1.1) with the standard formula in complex analysis for the winding number [4] of a curve γ around a base point z as being

$$w(z) = \frac{1}{2\pi i}\int_0^1 \frac{\gamma'(t)}{\gamma(t) - z}dt \qquad (2.2)$$

it becomes clearer how the structure of a complex-link (A, g, φ) is suitable as a model for an oriented curve in the plane.

The purpose of the following subsections is to develop further how the abstract mathematical structure of a complex-link is a model to a parameterized planar curve while detailing a couple of useful operations.

2.1 The concrete structure of a complex link

Let us recall that a complex or planar-link is a triple (A, g, φ) where A is a set (of indexes), $\varphi\colon A \to A$ is a re-indexing map (usually rquired to be a bijection) and $g\colon A \to \mathbb{C}$ is a map assigning to each index from the set A its geometrical realization as a complex number.

Every planar closed curve, that is, a continuous map

$$u\colon [0, 1] \to \mathbb{R}^2; \quad u(t) = (u_1(t), u_2(t))$$

with $u(0) = u(1)$, together with an ordered sequence of real numbers in the unit interval, say $0 \leq t_1 \leq t_2 \leq \ldots \leq t_n \leq 1$, gives rise to a link structure as follows:

1. $A = \{1, 2, \ldots, n\}$
2. $\varphi(i) = i + 1$ if $i < n$, and $\varphi(n) = 1$
3. $g(i) = u_1(t_i) + \sqrt{-1}u_2(t_i)$

For a simple concrete example let us consider the grid off complex numbers pictured as points in the plane as displayed in Figure 1 which can be represented by a link structure as displayed in Table 1.

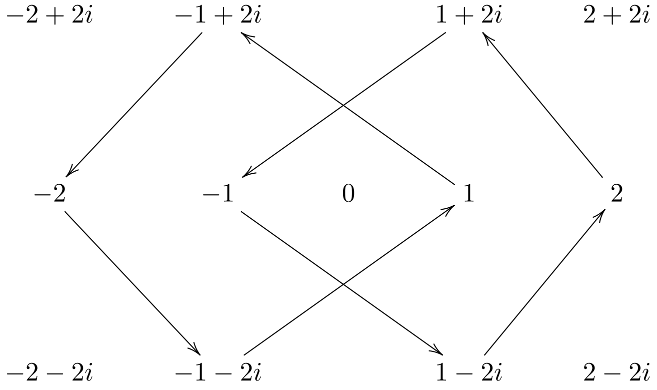


Figure 1: Picturing the link structure displayed in Table 1 as a directed graph in the complex plane

$x \in A$	$g(x)$	$\varphi(x)$
1	1	2
2	$-1 + 2i$	3
3	-2	4
4	$-1 - 2i$	1
5	2	6
6	$1 + 2i$	7
7	-1	8
8	$1 - 2i$	5

Table 1: Example of a link structure whose geometric realization is illustrated in Figure 1

The following subsection illustrates the process of resolving self intersections on a complex link.

2.2 Self intersections Algorithm

Given a link, say (A, g, φ), the self-intersections procedure returns a new link with the following property: if any two indexes have geometric realization direction vectors that intersect each other, then their intersection is the image of some index. This procedure requires the creation of new indexes every time an intersection occurs.

For example, the link depicted in Table 1 and illustrated in Figure 1 will return a new link with new indexes, as shown in Table 2 and illustrated in Figure 2.

The link-intersection procedure finds intersections, adds new vertices, and returns a new link structure $(\overline{A}, \overline{g}, \overline{\varphi})$ with the property that if two edges intersect then their intersection occurs on the starting points or the end points of the given edges.

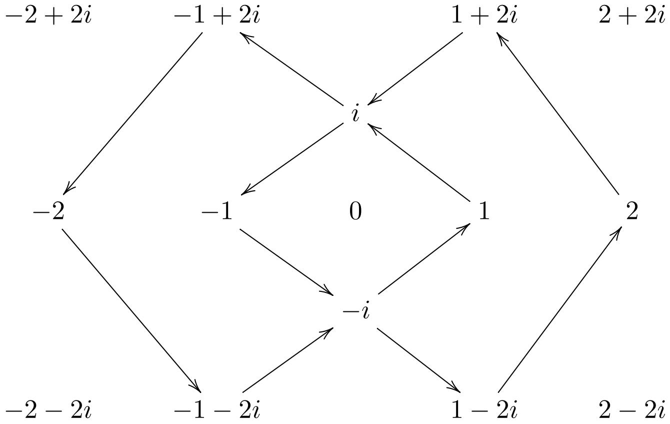


Figure 2: The result of performing the self-intersections procedure on the link illustrated in Figure 1

$x \in \overline{A}$	$\overline{g}(x)$	$\overline{\varphi}(x)$
$(1,0)$	1	$(1,0.5)$
$(2,0)$	$-1+2i$	$(3,0)$
$(3,0)$	-2	$(4,0)$
$(4,0)$	$-1-2i$	$(4,0.5)$
$(5,0)$	2	$(6,0)$
$(6,0)$	$1+2i$	$(6,0.5)$
$(7,0)$	-1	$(7,0.5)$
$(8,0)$	$1-2i$	$(5,0)$
$(1,0.5)$	i	$(2,0)$
$(4,0.5)$	$-i$	$(1,0)$
$(6,0.5)$	i	$(7,0)$
$(7,0.5)$	$-i$	$(8,0)$

Table 2: The result of performing the self-intersections procedure on the link of Table 1

We will now define some useful procedures that can be performed into a link structure, namely the reduce algorithm, which removes the indexes that are associated with a zero length edge when considered as a directed graph. And later we formalize the algorithm for self intersections.

2.3 Reduce

Given a link, say (A, g, φ), the reduce procedure returns a new link where each index has non zero geometric realization direction. That means, for every $x \in A$, the vector in the plane (or the complex number representing it, recall that we are identifying the plane with the set of complex numbers) $\operatorname{dir}(x) = g\varphi(x) - g(x)$ is non zero.

The implementation and detailed description of this procedure is not difficult to achieve and it relies on the fact that the set of indices A is a finite set and the map φ is a bijection. It is possible to consider the case when A is infinite but that would create unnecessary restrictions and it is not the purpose of this paper.

The general procedure is as follows. Let (A, g, φ) be a complex link, that is, for the purpose of this paper A is a finite set and φ is a permutation of A. We define the reduced link, say (A', g', φ') as follows:

$$A' = \{x \in A \mid \operatorname{dir}(x) \neq 0\}$$
$$g'(x) = g(x), \quad \forall x \in A' \varphi'(x) = \varphi^k(x)$$

with $k \geq 1$ is the least natural number for which $\varphi^k(x) \in A'$ (it is well defined because A is finite and φ is a bijection).

A simple example may be given as follows. The link structure (A, g, φ) is described in the table bellow, where $a = (0, 0)$ and $b = (1, 0)$, or, since we identify the complex plane $\mathbb{C}$ with the Cartesian plane $\mathbb{R}^2$, we can also think of complex numbers, $a = 0$ and $b = 1$.

A	g	φ
1	a	2
2	b	3
3	a	1
4	a	5
5	a	6
6	a	4

In this case, the reduce algorithm would return a new link, say (A', g', φ'), where $A' = \{1, 2\}$, $\varphi'(1) = \varphi(1) = 2$ and $\varphi'(2) = \varphi^2(2) = 1$. Clearly, the map g' is always the restriction of the map g to the subset $A' \subseteq A$.

The following procedure is to ensure that all the self intersections on the realization of a complex or planar link occur in a vertex which is indexed by some index in the set of indexes. For instance, that is not the case in the example pictured in Figure 1.

2.4 Self intersections

Given a link, say (A, g, φ), the self-intersections procedure returns a new link with the following property: if any two indexes have geometric realization direction vectors that intersect each other, then their intersection is the image of some index. This procedure requires the creation of new indexes every time an intersection occurs.

For example, the link depicted in Table 1 and illustrated in Figure 1 will return a new link with new indexes, as shown in Table 2 and illustrated in Figure 2.

The link-intersection procedure finds intersections, adds new vertices, and returns a new link structure $(\overline{A}, \overline{g}, \overline{\varphi})$ with the property that if two edges intersect then their intersection occurs on the starting points or the end points of the given edges.

The procedure may be defined as follows.

Let us suppose we are given a link (A, g, φ) with no zero length edges, that is, $\operatorname{dir}(x) \neq 0$, for all $x \in A$. If that is not the case then we perform the reduce transformation.

Denote by S the set of pairs $(a, b) \in A \times A$ whose associated edges $(g(a), g\varphi(a))$ and $(g(b), g\varphi(b))$ are not parallel (but we allow them to be coincident). This means that two possibilities may occur:

1. the two edges $(g(a), g\varphi(a))$ and $(g(b), g\varphi(b))$ are colinear and hence there exist unique real numbers s and t such that $g(a) = g(b) + t \operatorname{dir}(b)$ and $g(b) = g(a) + t \operatorname{dir}(a)$
2. the two edges are not parallel nor colinear and hence there exist unique real numbers s and t such that

$$g(a) + t \operatorname{dir}(a) = g(b) + s \operatorname{dir}(b)$$

Recall that $\operatorname{dir}(a) = g\varphi(a) - g(a)$.

The previous discussion shows that we can define two maps,

$$s, t \colon S \subset A \times A \to \mathbb{R}$$

defined as follows. If $\overrightarrow{a} = (g(a), g\varphi(a))$ and $\overrightarrow{b} = (g(b), g\varphi(b))$ are not parallel nor colinear then $s(a,b)$ and $t(a,b)$ are the unique real numbers such that

$$g(a) + t(a,b)\operatorname{dir}(a) = g(b) + s(a,b)\operatorname{dir}(b);$$

if $\overrightarrow{a}$ and $\overrightarrow{b}$ are colinear then $s(a,b)$ and $t(a,b)$ are the unique real numbers such that

$$g(a) = g(b) + s(a,b)\operatorname{dir}(b)$$

and

$$g(b) = g(a) + t(a,b)\operatorname{dir}(a)$$

This two functions are then used to construct the new set of indexes, as follows.

The new set of indexes resulting from the self-intersection procedure is

$$\overline{A} = \{(a,u) \in A\times[0,1[\mid \exists b \in A, u = t(a,b), s(a,b) \in [0,1[)\}$$

In other words, we collect all the indexes of the form (a,u) where $a \in A$ and $0 \leq u < 1$ for which there exist an index $b \in B$ whose geometric edge, $\overrightarrow{b}$, intersects the edge $\overrightarrow{a}$ at the point $g(a) + u\operatorname{dir}(a)$. If we drop the requirements that $t(a,b)$ and $s(a,b)$ need to be non-negative numbers strictly smaller than 1, then we would be considering intersections outside the length of the edges but still on their supporting lines. That would not give the desired result of identifying self intersections between the edges.

The previous construction determines the new set of indexes. We now have to define a new permutation and to give a geometric realization to the new indexes that have been inserted. Clearly, we will have the pairs of the form $(a,0)$ in $\overline{A}$ as copies of the original indexes in the set A. Moreover, since the set is finite, we may organize it, for each $a \in A$, as a sequence

$$(a,0), (a,u_1), (a,u_2), \ldots, (a,u_{n(a)})$$

with $0 < u_1 < \ldots u_{n(a)} < 1$, thus collecting all the indexes (a,u), for a fixed a and ordered them by u, with $n(a)$ counting their total number. This ordering allow us to define a geometric realization and a permutation of the indexes. The geometric realization is given by the formula

$$\overline{g}(a,u) = g(a) + u\operatorname{dir}(a)$$

while the rearrangement of the indexes is given by the formula

$$\overline{\varphi}(a,u_i) = (a,u_{i+1})$$

if $i < n(a)$, and $\overline{\varphi}(a,u_{n(a)}) = (\varphi(a),0)$ otherwise.

As an example, in Table 2 we present the resulting link that is obtained by performing the self-intersections procedure on the link described in Table 1. Figure 2 shows its geometric realization.

2.5 Cyclic ordering around a vertex

Given a link (A,g,φ), this operation produces a bijective map $\theta\colon A+A \to A+A$ such that $\theta(x,i)$ is the pair $(y,j) \in A+A$ such that the angle of (y,j) is greater or equal than the angle of (x,i) and there is no other pair (z,k) whose angle is in-between. We are using the notation $A+A$ to denote the set consisting of two copies of A, specificaly build as $A+A = \{(x,i) \mid x \in A, i = -1, 1\}$ and by the angle of a pair $(x,i) \in A+A$ we mean the angle of the complex number $dir(x)$ if $i = 1$ and $-\operatorname{dir}(x)$ if $x = -1$. Recall that $\operatorname{dir}(x) = g\varphi(x) - g(x)$.

A pseudo code implementation for this procedure is perhaps the best way to illustrate it. If a link (A,g,φ) is represented into a computer system such as Matlab, that is, g is a vector of complex numbers, the indexes are the numbers from 1 to the length of g, and φ is a vector with a permutation on the numbers 1 to length of g, then, the desired permutation θ, can be obtained by the following procedure:

```
% g a column vector of complex numbers of length n
% phi a column permutation on the numbers 1:n
dir=g(phi)-g;
M=[[g;g],[angle([dir;-dir])]];
[~,theta]=sortrows(M);
% theta is the desired permutation
for the cyclic ordering
```

3 The main procedure

We are now in position to outline the main procedure that associates a partition of the complex plane to every link structure (A,g,φ). This partition is represented as a map $w\colon \mathbf{C} \to \mathbf{Z}$. In this way each integer k gives a subset of the plane by looking at $w^{-1}(k)$. There will be exactly one unbounded region which will be associated with 0.

Let us now see how to define the map w. If $z \in \mathbf{C}$ is not in the image of the piece-wise linear curve determined by the link structure, then

$$w(z) = \frac{1}{2\pi}\sum_{x\in A} \angle\left(\frac{g\varphi(x)-z}{g(x)-z}\right) \tag{3.1}$$

otherwise, that is, if there exist $x \in A$ and $t \in [0,1[$ such that

$$z = g(x) + t\operatorname{dir} x \tag{3.2}$$

then, $w(z)$ is defined as $f(x)$ with $f\colon A \to \mathbf{Z}$ defined as detailed in the following procedure.

3.1 Procedure to define the map f

Start with an arbitrary link structure (A,g,φ). Take the resulting link structure which is obtained after applying the reduce operation and the self-intersections operation as explained above. This means that the resulting link (A,g,φ) has no zero length vectors $\operatorname{dir}(x)$ and all self intersections occur in $g(A) \subset \mathbf{C}$. Then we construct the bijection $\theta\colon A+A\to A+A$ as explained above and define a map $\tau\colon A+A\to\mathbf{Z}$ with the formula

$$\tau(x,i)=\begin{cases}1 & \text{if } i=1 \text{ and signal}\,\theta(x,i)=1\\ -1 & \text{if } i=-1 \text{ and signal}\,\theta(x,i)=-1\\ 0 & \text{otherwise}\end{cases}\tag{3.3}$$

with signal having the natural meaning of $\text{signal}(x,i)=i$, for any $(x,i)\in A+A$.

The next step is to form a directed graph whose set of vertices, V, is obtained by the quotient map $p\colon A+A\to V$, with q the coequalizer of θ and the identity map on $A+A$. In other words, the vertices are the orbits of the map θ. From here we can create the directed graph $d,c\colon A\to V$ defined by $d(x)=p(x,1)$ and $c(x)=p(x,-1)$.

Next we compute the set of connected components of the directed graph, denoted by Q, which is obtained as the coequalizer of the two maps d and c. For simplicity, let us suppose that there are exactly k connected components and that we have $Q=\{1,2,\ldots,k\}$. Then, for each element in Q, let us suppose there is a given choice of elements $x_1,x_2,\ldots,x_k$ in $A+A$ each one of which belonging to the respective connected component. We will see later on how to make this choice in an appropriated way. For the moment we assume we have a set, let us call it S_0 with the given elements. Let us also supose that a map $f\colon S_0\to\mathbf{Z}$ is given (typically f will be zero everywhere). Having S_0 we define $S_0^+=\{(x,-i)\mid(x,i)\in S_0\}$. The given map $f\colon S_0\to\mathbf{Z}$ is extended to S_1 so that $f(x,i)=f(x,-i)$. The next step is to define a new set

$$S_1=\{\theta^n(x,i)\mid n\in\mathbf{N},(x,i)\in S_0^+\}\tag{3.4}$$

extending the map f from S_0^+ to S_1 via the formula

$$f\theta^n(x,i)=f(x,i)+\sum_{j=1}^{n}\tau\theta^j(x,i),\tag{3.5}$$

enlarging the set S_1 to $S_1^+=\{(x,-i)\mid(x,i)\in S_0^+\}$ and letting f to be defined over S_1^+ so that $f(x,i)=f(x,-i)$.

We then repeat the same process obtaining, for $k=2,3,\ldots$, successively $S_k=\{\theta^n(x,i)\mid n\in\mathbf{N},(x,i)\in S_{k-1}^+\}$, then extending the definition of f as in equation 3.5, enlarging the set S_k to S_k^+ as above and allowing f to be defined over S_k^+.

The process stops when $S_k^+=A+A$. At that stage, and since the map f has the property that $f(x,i)=f(x-i)$, then we may consider it as a map $f\colon A\to\mathbf{Z}$.

This gives us the desired definition for $w(z)$ when $z=tg\varphi(x)+(1-t)g(x)$ for some $x\in A$ and some $t\in[0,1[$, which is simply given by $w(z)=f(x)$.

It remained to see how to perform the appropriate choice of the values in the set S_0, one for each connected component in Q. For simplicity let us assume that the directed graph only had one connected component. The case when there are more is treated in the same way.

3.2 The choice of the initial points from each connected component

In this subsection we explain how to obtain a initial point for each connected component and what is the value of it by the map f on the previous subsection.

We will first give the procedure to find the initial point of a given connected component and then we will explain the meaning of each step.

Let us suppose we have a complex link (g,φ) with one connected component. Moreover, let us assume that $\theta\colon A+A\to A+A$ is its cyclic ordering around each vertex. Furthermore, let us assume that the complex plane is equipped with its lexicographic order, that is, $z_1\le z_2$ if and only if $|z_1|<|z_2|$ or if $|z_1|=|z_2|$ and $\angle z_1\le\angle z_2$:

1. find the minimum and maximum values of $g(x)$ for all $x\in A$,

2. choose $x_0\in X$ such that $g(x_0)$ is a minimum and consider all possible $(x_j,i_j)\in A+A$, with $j=1,\ldots n$ such that $g(x_j)$ is maximum,

3. we choose the appropriate $(x_j,i_j)\in A+A$ with $j=1,\ldots n$, such that $g(x_j)$ is maximum, as being such that $\angle(x_j,i_j)-\angle(g(x_j)-g(x_0))$ is minimal.

4. if $(x,i)\in A+A$ is such that $g(x)$ is maximum and $angle(x,i)-angle(g(x)-g(x_0))$ is minimal amongst all the pairs (x_j,i_j) such that $g(x_j)$ is maximal, then it is the chosen element for the given connected component and moreover, $f(x,i)=0$ if $i=1$ and $f(x,i)=-1$ if $i=-1$.

4 Possible applications of the procedure

Having the procedure described above we may find simple and computationally efficient ways of working with offsets for arbitrary regions on the plane, as well as many other logical operations like union, intersection, symmetric difference or complement.

4.1 Offsets

Given a simply region on the complex plane (that is, one whose boundary determines a simple oriented curve) we can make an arbitrary discretization of it thus obtaining a link structure. By applying an offset to each edge (this can be done in two different ways, either offseting the vertices and letting the edges to follow, or to offset the edges, creating thus new connecting edges between the offsetted ones). In either case we obtain a new link structure. To thus link structure we associate the map f as explained in the previous section. The result of the offset is obtained by taking the inverse image of $0\in\mathbf{Z}$.

4.2 Union, intersection and symmetric difference

Suppose (A_1, g_1, φ_1) and (A_2, g_2, φ_2) are two link structures representing two regions on the plane. Then we may form a new link $(A_1 + A_2, [g_1, g_2], \varphi_1 + \varphi_2)$ and applying the process described above to it we get the map $f\colon A_1 + A_2 \to \mathbf{Z}$. The map f induces the map w which is defined for all the complex plane. In particular we get the results obtained in [2].

Acknowledgement

This work was supported by the Fundação para a Ciência e a Tecnologia (FCT) and Centro2020 through the Project references: UID/Multi/04044/2013; PAMI - ROTEIRO/0328/2013 (Nº 022158); MATIS (CENTRO-01-0145-FEDER-000014 - 3362); Additive Cooling and Additive Milling. Also by FCT/MCTES (PIDDAC) through the following Projects: Associate Laboratory ARISE LA/P/0112/2020; UIDP/04044/2020; UIDB/04044/2020; CENTRO-01-0247-FEDER-(069665, 039969); POCI-01-0247-FEDER-(069603, 039958, 039863, 024533); Generative thermodynamic; by CDRSP and ESTG from the Polytechnic of Leiria.

Referências

[1] N. Martins-Ferreira, *The notion of multi-link, its applications and examples*, Scripta-Ingenia **7** December (2017) 14–21.

[2] M. B. Gaspar and N. Martins-Ferreira, *A procedure for computing the symmetric difference of regions defined by polygonal curves*, J. Symb. Comput. **61-62** (2014) 53–65.

[3] J. W. Alexander, *Topological Invariants of Knots and Links*, Transactions of the American Mathematical Society **30** (2) (1928) 275–306.

[4] J. A. Baker, *Plane Curves, Polar Coordinates and Winding Numbers* Mathematics Magazine, **64** (2) (1991), 75–91

[5] Hsien-Chih Chang, Jeff Erickson, *Electrical Reduction, Homotopy Moves, and Defect*, arXiv:1510.00571v1, 2015.

[6] Hsien-Chih Chang, Jeff Erickson, *Untangling Planar Curves*, arXiv:1702.00146, 2017.