9th International Conference on Fracture Mechanics of Concrete and Concrete Structures
FraMCoS-9
V. Saouma, J. Bolander, and E. Landis (Eds)

DOI 10.21012/FC9.178

PID CONTROLLER IN EXPLICIT SOLVER

JAN VOREL*, MARCO MARCON*, DANIELE PELESSONE†, GIANLUCA CUSATIS‡
AND ROMAN WENDNER*

*Christian Doppler Laboratory LiCRoFast
Institute of Structural Engineering
University of Natural Resources and Life Sciences Vienna
Vienna, Austria
e-mail: jan.vorel@boku.ac.at, marco.marcon@boku.ac.at, roman.wendner@boku.ac.at

†Engineering and Software System Solutions, Inc. (ES3)
San Diego, USA
e-mail: daniele.pelessone@es3inc.com

‡Dep. of Civil and Environmental Engineering
Northwestern University
Evanston, USA
e-mail: g-cusatis@northwestern.edu

Key words: PID controller, explicit solver, MARS, microplane M4

Abstract. In this paper the implementation of a proportional-integral-derivative (PID) controller is presented to drive the loading of structures within the framework of explicit solver. Such approach is especially necessary for the system in which the initiation of damage results in the snap-back, i.e., the initiation of damage causes a reduced system deformation (in certain points) with a reduced load.

1 INTRODUCTION

A proportional-integral-derivative (PID) controller was originally developed from a governor device, which was used to measure and regulate the speed of a machine. It was subsequently developed and used within automatic ship steering and then for use as a pneumatic controller [1]. More recently PID controllers are used within a wide range of applications from industrial ovens to packaging machines. In general, a PID controller is a control loop feedback mechanism (controller) commonly used in industrial control systems. A PID controller continuously calculates an error value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimise the error over time by adjustment of a control variable.

In this paper the implementation of the PID controller is presented to drive the loading of structures within the framework of explicit solver. Such approach is especially necessary for the system in which the initiation of damage results in the snap-back, i.e., the initiation of damage causes a reduced system deformation (in certain points) with a reduced load. In these cases the stable control measure has to chosen to ensure the stability of the solution, e.g., a notched three-point bending test is controlled by means of the crack mouth opening displacement (CMOD).

2 PID controller discretization

The overview of the development of the discrete time PID controller is summarised in this section. Following [2], the discretization is based on the well known continuous time PID controller defined as [3]

$$\begin{aligned} u(t) &= u_0 + K_p e(t) + \frac{K_p}{T_i}\int_0^t e(\tau)\,\mathrm{d}\tau + \\ &+ K_p T_d \dot{e}(t), \end{aligned} \tag{1}$$

where u is the controller output and u_o is the bias or manual control value. K_p stands for the coefficient of the proportional gain, T_i, T_d characterise the integral and the derivative time constants, respectively. e represents the control error

$$e(t) = r(t) - y(t), \tag{2}$$

where r is the reference value and y is the process measurement. The controller can be also parameterised as

$$\begin{aligned} u(t) &= u_0 + K_p e(t) + K_i\int_0^t e(\tau)\,\mathrm{d}\tau + \\ &+ K_d \dot{e}(t), \end{aligned} \tag{3}$$

where K_i, K_d are the integral gain and the derivative gain, respectively.

To derive the discrete time formula for a current time step, both sides of Eq. (1) are differentiated

$$\begin{aligned} \dot{u}(t) &= \dot{u}_0 + K_p \dot{e}(t) + \frac{K_p}{T_i} e(t) + \\ &+ K_p T_d \ddot{e}(t). \end{aligned} \tag{4}$$

Applying the backward difference formula Eq. (4) gives the discrete time PID controller

$$\begin{aligned} u(t_k) &= u(t_{k-1}) + \left[u_0(t_k) - u_0(t_{k-1})\right] + \\ &+ K_p\left[e(t_k) - e(t_{k-1})\right] + \\ &+ \frac{K_p T_s}{T_i} e(t_k) + \\ &+ \frac{K_p T_d}{T_s}\left[e(t_k) - e(t_{k-1}) + e(t_{k-2})\right], \end{aligned} \tag{5}$$

where T_s refers to the time step (sampling interval). Based on Eq. (5), the increment of the control value is defied as

$$\begin{aligned} \Delta u(t_k) &= \left[u_0(t_k) - u_0(t_{k-1})\right] + \\ &+ K_p\left[e(t_k) - e(t_{k-1})\right] + \\ &+ \frac{K_p T_s}{T_i} e(t_k) + \\ &+ \frac{K_p T_d}{T_s}\left[e(t_k) - e(t_{k-1}) + e(t_{k-2})\right]. \end{aligned} \tag{6}$$

3 PID controller implementation

The discrete time PID controller derived in the previous section is implemented in MARS. MARS is a powerful and robust object-oriented solver for simulating the mechanical response of structural systems subjected to short duration events. It employs an explicit time integration scheme for solving the equation of motion of large systems. It implements all the capabilities and versatility of general finite element codes. In addition, MARS features some unique techniques, such as the Lattice Discrete Particle Model (LDPM) and adaptive remeshing algorithms for shell and solid meshes, which facilitate the solution of problems involving structural break-ups, fragmentation and post-failure response under extreme loading conditions [4, 5].

The increment of control value is calculated according Eq. (6) in which four parameters have to be set up (K_p, T_i, T_d, T_s). The most critical parameter, which has to be defined, is the proportional gain. In the standard PID controller utilisation, this parameters usually stays constant during the whole controlled process and is defined beforehand. Here, we can take the advantage of the numerical implementation and to derive this parameter based on the results obtained during the preceding simulation steps. Therefore, the ratio between the prescribed loading and the controlled value reads

$$K_p(t) = \frac{u(t_k) - u(t_{k-1})}{y(t_k) - y(t_{k-1})}. \tag{7}$$

To avoid some excessive control value changes, the calculated K_p value is limited by the setpoint value change. If the setpoint change is less then given limit or less then 1 per mille of the maximum K_p experienced in previous steps, the previous coefficient is used. It is quite tempting to set the sampling time to be equal to a given time step used by the explicit solver. However, this choice would effect the stability of the proposed control process caused by the intrinsic oscillation of explicit solver solution and the response of the structure on the prescribed load. Therefore, the sampling time should be chosen larger then the time step of the solver. The other two parameters (T_i, T_d) are then set based on the experience.

The integral windup, referring to the situation in a PID controller where a large change in setpoint occurs, is limited by means of the user defined limit. The occurrence of large control value change, caused by the integral part of the PID, is usually experience when the sudden jump in the loading curve is enforced.

To smooth out the noise in the data obtained by the numerical simulation and used for the set up of the PID controller, a low pass filter is employed. There exist many different types of low pass filters which can be used to smooth the data. For the sake of simplicity and a low storage requirements, the exponential moving average (exponentially weighted moving average) is used in the proposed computational scheme. Moreover, to take into account possible inconsistent time steps, the exponential moving average for unevenly spaced time series is implemented [6, 7]

$$\begin{aligned} \bar{y}(t_k) &= w\bar{y}(t_{k-1}) + (s-w)y(t_{k-1}) + \\ &+ (1-s)y(t_k), \end{aligned} \tag{8}$$

where $w = \exp(-\beta)$, $s = (1-w)/\beta$ and $\beta = (t_k - t_{k-1})/\alpha$. α stands for the filter time constant to be defined.

In the current implementation, the loading of the structure is driven by the prescribed velocity of the loaded nodes or body. During the sampling interval the adjusted velocity by the PID controller is kept constant and so the calculated proportional gain takes the form

$$K_p(t) = \frac{\dot{u}(t_{k-1})T_s}{y(t_k) - y(t_{k-1})}. \tag{9}$$

The anti-windup limit is then defined as a maximum allowed acceleration $(a_{\lim})$

$$\left| K_p \left[e(t_k) - e(t_{k-1}) \right] \right| / T_s^2 \leq a_{\lim}. \tag{10}$$

Based on Eq. (6), the final velocity then takes the form

$$v(t_k) = \tilde{v}(t_{k-1}) + \Delta v(t_k), \tag{11}$$

where

$$\tilde{v}(t_{k-1}) = v(t_{k-1}) \frac{r(t_k + T_s) - r(t_k)}{r(t_k) - r(t_k - T_s)}, \tag{12}$$

for $[r(t_k) - r(t_k - T_s)] \neq 0$ and $\tilde{v}(t_{k-1}) = v(t_{k-1})$, otherwise. It must be noted that at least two time steps $(2T_s)$ are demanded, before the PID is switched on, to be able to set up all necessary parameters.

3.1 PID controller examples

Three different examples of the prescribed loading scenarios using the PID controller are presented in this section. The three point bending test setup (Fig. 1) is utilised to check the implementation and the ability of the proposed PID controller. The loading is controlled by means of the prescribed velocity of the top loading block and the controlled value is the crack mouth opening displacement. For the first two cases, the material is assumed to be elastic. For the last case the material is assumed to be concrete modelled by means of the microplane model M4 [8, 9].

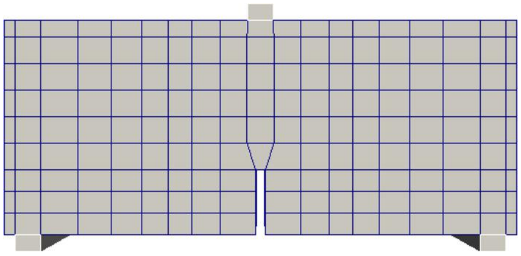

Figure 1: Three-point bending test setup (40x40x96 mm).

First, the ramp load function for the CMOD is prescribed. As shown in Fig. 2, the prescribed opening matches really well the calculated value (loading scenario no. 1).

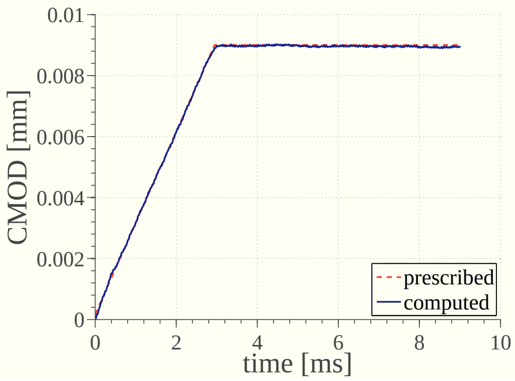

Figure 2: Three-point bending test - loading scenario no. 1.

The other examples is chosen to show the ability of the PID to deal with the sudden jumps in the prescribed CMOD. The defined loading curve as well as the model response are presented in Fig. 3 (loading scenario no. 2). As can be seen in Fig. 3, after the sudden jumps, the controller needs some time to limit the experienced error.

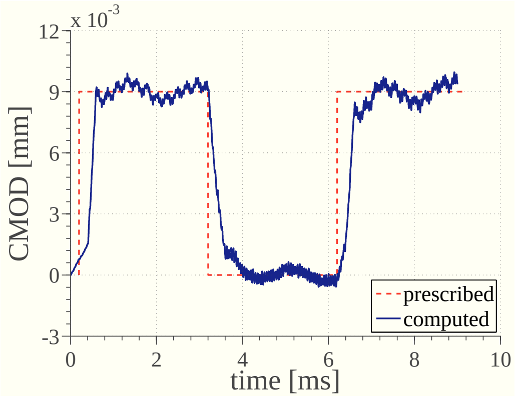

Figure 3: Three-point bending test - loading scenario no. 2.

The last example assumes the constant CMOD opening rate (loading scenario no. 3). The material of the beam is modelled as a quasi-brittle material (concrete) to demonstrate the ability of the controller to capture the softening part of the global beam response, see Fig. 5.

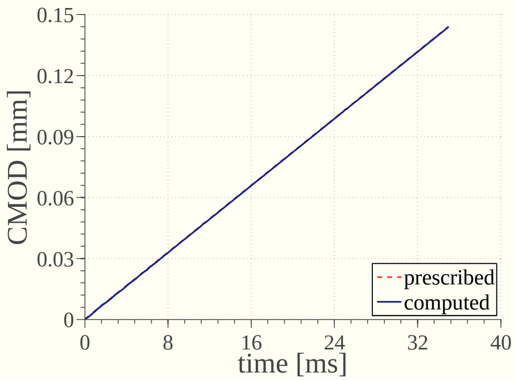

Figure 4: Three-point bending test - loading scenario no. 3.

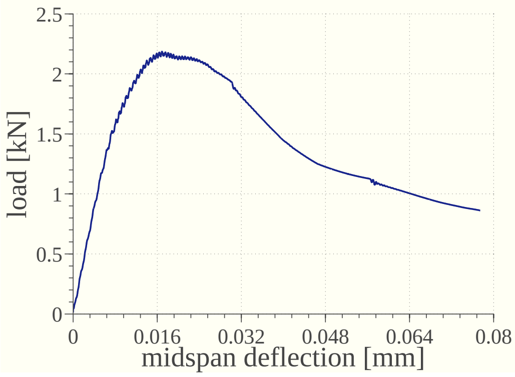

Figure 5: Three-point bending test - load vs. midspan deflection (loading scenario no. 3).

4 CONCLUSIONS

The description and verification of the proposed PID controller, used within the framework of the explicit solver, were presented. As shown in the previous section, the PID controller is suitable for different types of loading as well as the material types. As already mentioned, the current implementation was utilised together with the explicit solver. However, the extension for the implicit solver is rather straightforward.

Acknowledgement

The financial support by the Austrian Federal Ministry of Economy, Family and Youth and the National Foundation for Research, Technology and Development is gratefully acknowledged.

REFERENCES

[1] Stuart Bennett. A brief history of automatic control. *IEEE Control Systems Magazine*, 16(3):17–25, 1996.

[2] Finn Haugen. Discretization of simulator, filter, and pid controller, 2010.

[3] Kiam Heong Ang, Gregory Chong, and Yun Li. Pid control system analysis, design, and technology. *Control Systems Technology, IEEE Transactions on*, 13(4):559–576, 2005.

[4] Gianluca Cusatis, Daniele Pelessone, and Andrea Mencarelli. Lattice discrete particle model (LDPM) for failure behavior of concrete. I: Theory. *Cement and Concrete Composites*, 33(9):881–890, 2011.

[5] Gianluca Cusatis, Andrea Mencarelli, Daniele Pelessone, and James Baylot. Lattice discrete particle model (LDPM) for failure behavior of concrete. II: Calibration and validation. *Cement and Concrete Composites*, 33(9):891–905, 2011.

[6] Andreas Eckner. Algorithms for unevenly-spaced time series: Moving averages and other rolling operators. Technical report, Working Paper, 2012.

[7] Ulrich A Müller. Specially weighted moving averages with repeated application of the ema operator. *Internal document UAM. 1991-10-14, Olsen & Associates, Switzerland*, 1991.

[8] Zdeněk P Bažant, Ferhun C Caner, Ignacio Carol, Mark D Adley, and Stephen A Akers. Microplane model m4 for concrete. i: Formulation with work-conjugate deviatoric stress. *Journal of Engineering Mechanics*, 126(9):944–953, 2000.

[9] Giovanni Di Luzio. A symmetric over-nonlocal microplane model m4 for fracture in concrete. *International journal of solids and structures*, 44(13):4418–4441, 2007.