Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 48-20 (2015) 523–527

Iterative Learning Control: An Example for Mechanical Ventilated Patients

M. Scheel * A. Berndt * O. Simanski **

* HOFFRICHTER GmbH Schwerin, 19061 Schwerin GER (e-mail: scheel@hoffrichter.de, berndt@hoffrichter.de)
** Computational Engineering and Automation Group, Hochschule Wismar - University of Applied Sciences Technology, Business and Design , Wismar, Germany, (e-mail: olaf.simanski@hs-wismar.de)

Abstract: Positive pressure ventilation is a method of artificial ventilation. If patients can not breath normally due to respiratory insufficiency, they are connected to a mechanical ventilator. The ventilator generates a positive end-expiration pressure (PEEP) during the expiration phase and an inspiratory positive airway pressure (IPAP) during the inspiration phase. These different pressure levels lead to inflation and deflation of the lung and the patient is ventilated. To achieve the desired pressure levels a closed-loop pressure control has to be designed and developed. Linear PI(D)-controller for example can not follow predefined reference trajectories exactly, because of different and varying patient states and lung parameters. An adaption from patient to patient or from breathing cycle to breathing cycle is usually not possible. Due to further development in microcontroller technology, more complex control algorithms can be used. For cyclically recurring processes iterative learning control (ILC) algorithms provide the possibility to react on variable environment conditions. In ventilation the ILC algorithm can learn the required signal from breathing cycle to breathing cycle to track the reference trajectory.

Keywords: mechanical ventilation, pressure control ventilation, control design, iterative learning control,

1. INTRODUCTION

The standard practice for artificial ventilation is a positive pressure ventilation. In emergency medicine, for comatose patients or patients with respiratory insufficiency, mechanical ventilation is essential and ventilators are part of the standard equipment in hospitals, ambulances, rescue helicopters and more.

The basic idea of positive pressure ventilation is the generation of an inspiratory positive airway pressure (IPAP) to fill the lungs with oxygenic air during the inspiration phase. In the expiration phase the medical ventilator reduces the pressure to positive end expiration pressure (PEEP) to release the consumed air. This cyclically recurring process ensures the ventilation of the patient. The determination of the inspiration and expiration phase occurs time-controlled or by respiratory efforts of the patient and is depending on different ventilation modes [Kherallah, Rathgeber (2010)]. Figure 1 shows an ideal example of pressure controlled ventilation (PCV) with time-controlled inspiration and expiration. The PCV-mode with time-controlled ventilation is mostly used for patients, who can not breath autonomously, e.g. comatose patients.

The specification of a medical ventilator is to maintain different pressure levels (e.g. PEEP and IPAP) set by the medical scientist, irrespective of varying lung states (resistance, compliance, inertance) and different patients. Therefore a closed-loop pressure control has to be implemented to achieve the desired pressure levels. Because of

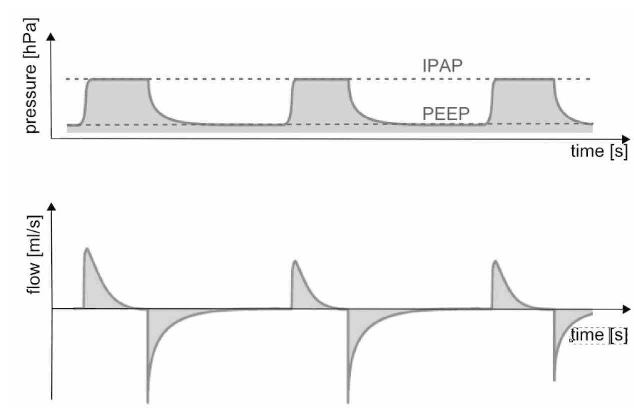

Fig. 1. Pressure and flow diagram in PCV-mode [Rathgeber (2010)]

the nonlinear and time varying process, standard feedback and simple adaptive control regimes may not perform set point changes with defined accuracy. They are not able to learn and can only react to errors after they occurred. As the development in microcontroller technology proceeds, more efficient and more complex algorithms can be developed. Due to cyclical repetitions of the breathing, memory-based control algorithms provide a possibility to learn from breathing cycle to breathing cycle to reduce the error. One of these memory-based algorithms is the iterative learning control algorithm.

10.1016/j.ifacol.2015.10.194

The main content of this article consists of the design of an iterative learning control for mechanical ventilated patients. For a first control approach, the breathing period has a fixed time interval and the patient has no intention to breath spontaneously. In the following, remarks about iterative learning control, with some historical aspects, the basic idea and one possible implementation in an existing control structure are given. In section 3 a simple ventilator-patient model is provided for some simulation in MATLAB©Simulink and a first iterative learning control is developed. Afterwards an implementation on a micro-controller is described and the control results of the developed iterative learning control and of a classical linear PI-controller are compared. After all a summary and further ideas are proposed.

2. ITERATIVE LEARNING CONTROL: A SHORT REVIEW

Control systems play an important role in our modern civilization and technology. In practically all engineering fields, e. g. automation, space engineering, military systems, robotics, chemical processes and also biomedical systems, control problems can be found. The most common key feature to improve the performance of a dynamic system is the use of feedback signals. Nowadays in industrial control problems approx. 90-95% [Astroem et al. (2006)] of all controllers are of proportional-integral-derivative (PID) or model predictive control (MPC) type.

Most control methods including standard feedback, adaptive or robust control methods may not sufficient to fulfil the following tasks: perfect reference tracking in a finite time interval under repeatable control environment. On the one hand the control methods are characterized by an asymptotic convergence, thus a perfect reference tracking is not guarantee. On the other hand, this is the major disadvantage, they are not able to learn from previous cycles. One of the first industrial application where an intelligent learning control algorithms was required, was for robot control [Zilouchian (1994)]. An idea for intelligent control algorithm is the use of previous system information to improve the present execution. A more recent control theory is the iterative learning control one (ILC). Arimoto [Arimoto et al. (1984)] presented a first algorithm, called "Arimoto-type" ILC law (see (1)), to improve the input variable u_k in that way that the tracking error e_k will tends against zero in the next cycle. u_k and e_k are vectors of field length [0,T] - Γ is a constant learning gain - k is the iteration index.

$$u_{k+1}(t) = u_k(t) + \Gamma\ \dot{e_k}(t) \tag{1}$$

There are four decisive properties [Xu et al. (2009)] that makes ILC an appealing control algorithm:

(1) Simplicity in structure: ILC can be implemented in existing control structures
(2) Ability for perfect reference tracking: ILC can minimize the error iteratively despite parameter uncertainties, unknown dynamics, measurement noise and disturbances
(3) Model-free: ILC needs no system model knowledge
(4) Ability for non-causal signals: ILC can use non-causal signals because all previous information can be stored in memory

One of the first ILC book [Moore (1993)] focused on ILC concept and a couple of algorithms. Another book from [Bien et al. (1998)] displays the latest research findings until the late 1990s. Many different authors published there work on convergence [Rogers et al. (1996)] and robustness [Chen et al. (1999)], linear and nonlinear [Xu et al. (2003)], and other criteria for iterative learning control.

Several extensions have been developed in order to improve the "Arimoto-type" ILC law. The most widely used equation for this is:

$$u_{k+1}(t) = Q\ \{u_k(t) + L_a\ \dot{e_k}(t) + L_b\ e_k(t)\} \tag{2}$$

Q, L_a and $L_b \in \mathbb{R}^{n\times n}$ are arrays, which determine the convergence behaviour and speed. The matrix Q can be used as a non-causal filter with a gain ≤ 1, called forgetting factor, to increase the robustness of the system against measurement noise and disturbances [Wallen (2011), Nahrstaedt et al. (2014)]. n is the number of sampling steps and T the period for a discrete ILC algorithm - therefore the vectors u_k and e_k are of length n.

Another improvement to ensure the convergence is the observation of the tracking error in sense of L_2-norm [Schmidt (1996)]. This requires the calculation of the L_2-norm of k^{th} iteration.

$$\|e_k\| \leq \|e_{k-1}\| \tag{3}$$

$$\sqrt{\sum_{i=1}^{n} e_k(i)^2} \leq \sqrt{\sum_{i=1}^{n} e_{k-1}(i)^2} \tag{4}$$

The observation checks, if an actual enhancement from iteration to iteration has occurred. If this is not the case, the calculation of control algorithm u_{k+1} is not executed and a previous determined "optimal" input variable is used. Figure 2 shows an example of the main structure of ILC in an existing control environment. i indicates the sampling step, y_r the reference signal, y the control variable, e the tracking error, u_c the output of the controller and u the complete input variable to the dynamic system.

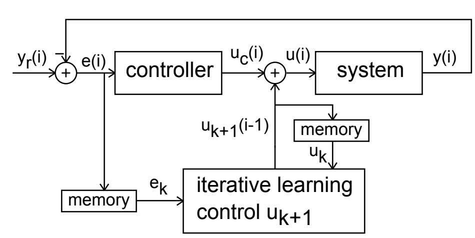

Fig. 2. Example of an iterative learning control

The ILC output u_{k+1} is shifted to the left, because the tracking error at position i can only be minimized by a previous input signal $i-1$ due to delay of sampling. Further