Computers in Industry 51 (2003) 73–88

COMPUTERS IN INDUSTRY

www.elsevier.com/locate/compind

A software tool development for pneumatic actuator system simulation and design

Y.Y. Lin-Chen, J. Wang*, Q.H. Wu

Department of Electrical Engineering and Electronics, The University of Liverpool, Brownlow Hill, Liverpool L69 3GJ, UK

Received 9 August 2002; accepted 2 January 2003

Abstract

The ready availability of low cost microprocessors and mechantronic components allow industrial users to consider adopting servo-controlled pneumatic actuators with an acceptable cost. This leads to a demand for a software tool in pneumatic actuator system CAS/CAD. Therefore, the paper introduces a software tool developed by the research group in Liverpool for pneumatic actuator system computer aided simulation and design. Pneumatic system components are initially organised into five major classes. Those components are considered as subsystems to a complete pneumatic system and the mathematical models for the individual components are derived which can be combined in different ways to form a complete pneumatic system model. A library is built up to accommodate the five classes of components. Users can pick up different components from the library to formulate a complete pneumatic system based on the design requirements. The complete system dynamic behaviours can then be simulated in different operating modes. The graphic user interface (GUI) and animation techniques are adopted in software design to create a user-friendly environment. The software is still in its early stage and only used for research purpose but it has potential for further development.

Keywords: CAS/CAD; Non-linear systems; System modelling; Pneumatic actuators; Components-based software design

1. Introduction

Pneumatic actuators have been widely used in the applications for simple speed control in industrial process and automation. In recent years, the ready availability of low cost microprocessors and advanced mechantronic components allow industrial users to consider adopting pneumatic actuators to accomplish more sophisticated motion control tasks [1–5]. However, there exist some difficulties in pneumatic system control and analysis which are associated with air compressibility, significant friction and non-linearities [5–7]. This leads to a demand for an assistant software tool to be used for dynamic analysis of pneumatic systems to address those difficulties in system design. Most current available pneumatic system CAS, CAD software packages are mainly developed for the purpose of pneumatic circuit designs [8–10], not for pneumatic system dynamic analysis. Therefore, development of a software tool for pneumatic actuator system simulation and design is proposed in this paper.

The software should be easy to use, easy to understand, re-configurable, and easy to simulate the

* Corresponding author. Tel.: +44-151-794-4509; fax: +44-151-794-4540.
E-mail address: jhwang@liv.ac.uk (J. Wang).

0166-3615/03/$ – see front matter
doi:10.1016/S0166-3615(03)00026-5

Nomenclature

a, b	subscripts for inlet and outlet chambers, respectively
A	ram area (m^2)
C_d	discharge coefficient (0.8)
C_k	$[(2/(k-1))((k+1)/2)^{(k+1)/(k-1)}]^{1/2} = 3.864$
C_r	$(P_2/P_0)_{cr} = (2/(k+1))^{k/(k-1)} = 0.852$
C_0	$[(k/R)(2/(k+1))^{(k+1)/(k-1)}]^{1/2} = 0.04$
k	ratio of specific heat (1.4)
l	stroke length (m)
m	payload mass (kg)
$M_{a,b}$	mass flow rate (kg/s)
$P_{a,b}$	chambers A, B pressure (N/m^2)
P_{atm}	atmospheric pressure (N/m^2)
P_d	down stream pressure (N/m^2)
P_e	exhaust pressure (1×10^5 N/m^2)
P_s	supply pressure (6×10^5 N/m^2)
P_u	up stream pressure (N/m^2)
R	universal gas constant (287 J/(K kg))
T_s	supply temperature (293 K)
T_u	up stream temperature (K)
v	piston velocity (m/s)
$W_{a,b}$	port width (m)
x	piston position (m)
$X_{a,b}$	valve spool displacement (m)

dynamic behaviours of pneumatic systems. From analysis of current software design technologies, a components-based software design method is adopted in the paper, together with the features of graphic user interfaces (GUI).

Components-based software design method is a new phase in object oriented methodology evolution. The recent development in component technology enables the construction of complex software systems by assembling together off-the-shelf components [11,12]. The major characteristic of a component is to be considered as a unit of independent deployment and has no persistent state [13,14]. This design method is now widely used in CAD software design. A software component should show same dynamic behaviours as its real world counterpart and should adopt the standard interface for the convenience of system integration. The software implementation of a component contains program code and data, which can be considered as an independent unit for users to pick up to connect with other components to form a complete pneumatic system. Their interfaces allow them to accept input and feedback information from and also send output and feedback information to those components which are adjacent to them. A typical example of component structure is shown in Fig. 1.

The pneumatic system components are initially organised into five major categories, which are compressed air supplies, valves, cylinders, control strategies and miscellaneous parts (for example, connection pipes). The above organisation is determined based on the mechanical structure of pneumatic systems, which is generally the classification shown in the manufacturers' catalogues. The way of organising the components' classes may lead the users to feel same when they pick up the components from the library as they are picking the components from the manufacturers' catalogues. A component library accommodating these five classes of components is then built up on the basis of dynamic analysis of pneumatic systems and component models. Users can pick up different components from the library to construct a complete pneumatic system. The dynamic behaviours of the complete system can be simulated in different operating modes. The graphic user interface and animation techniques are adopted in software design to create a user-friendly environment. MATLAB-SIMULINK is used to implement the software design. Three different kinds of software structures have been investigated

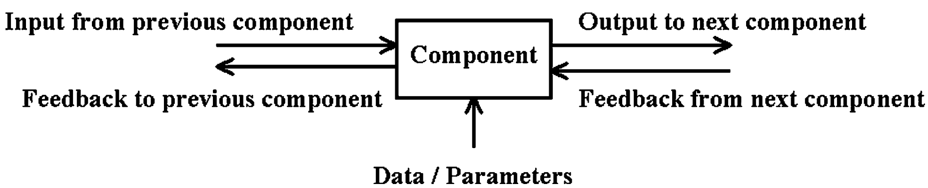

Fig. 1. Typical structure of a component.