MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 2, 53-60

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

ISSN 1335-8871

Capacities and Limitations of Wind Tunnel Physical Experiments on Motion and Dispersion of Different Density Gas Pollutants

Ondřej Zavila[1], Tomáš Blejchař[2]

[1]*Department of Fire Protection, Faculty of Safety Engineering, VŠB - Technical University of Ostrava, Lumírova 13/630, Ostrava - Výškovice, 700 30, Czech Republic, ondrej.zavila@vsb.cz*
[2]*Department of Hydrodynamics and Hydraulic Equipment, Faculty of Mechanical Engineering, VŠB - Technical University of Ostrava, 17. listopadu 15, Ostrava – Poruba, 708 33, Czech Republic*

The article focuses on the analysis of the possibilities to model motion and dispersion of plumes of different density gas pollutants in low-speed wind tunnels based on the application of physical similarity criteria, in this case the Froude number. The analysis of the physical nature of the modeled process by the Froude number is focused on the influence of air flow velocity, gas pollutant density and model scale. This gives an idea of limitations for this type of physical experiments in relation to the modeled real phenomena. The resulting statements and logical links are exemplified by a CFD numerical simulation of a given task calculated in ANSYS Fluent software.

Keywords: Physical similarity, aerodynamic tunnel, Froude number, CFD, gas pollutant.

1. INTRODUCTION

Air pollution is becoming an increasingly serious global issue. Factories produce large amounts of pollutants that damage the environment and harm human health. From this point of view, problems of motion and dispersion of pollutants in the atmosphere relate not only to environmental studies but also to other disciplines, such as safety engineering.

An understanding of the physical principles of pollutants' motion and dispersion is important in order to determine the impact of air pollution on the environment and humans. This study only deals with the physical principles of pollutants' motion and dispersion. Possible chemical reactions in the atmosphere are not covered.

For the purpose of the study, a simple model of a typical real situation was defined. Physical parameters of the model were gradually modified to achieve visible changes in results so that general principles could be defined. The above-mentioned demonstration model represents a chimney situated in a simple flat terrain. Gas pollutant is discharged from the chimney and carried by flowing air. Gas pollutant plume is detected and visualized with a numerical model as iso-surfaces or contours of pollutant concentrations in two-dimensional cut planes of three-dimensional geometry.

The dependence of the pollutant plume shape, size and inclination on modification of three physical parameters was investigated. The selected parameters included pollutant density, air flow velocity and model scale.

The results are presented in the form of text and commented figures. ANSYS Fluent 15.0 CFD (Computational Fluid Dynamics) code was used to demonstrate and visualize all problem variants (see [1], [2]). The numerical model of the pollutant plume motion created in this software was verified by an experiment conducted in the low-speed wind tunnel in the Aerodynamic Laboratory of the Academy of Sciences of the Czech Republic in Novy Knin (see [3], [4], [5]). The maximum difference between experiment data and numerical simulation data sets was one order (for more details see [6]). One of the aims of the study is also to demonstrate that physical modeling of pollutant plume motion and dispersion with severely downscaled models has its limitations that should be known and considered to avoid obtaining false results.

2. SUBJECT & METHODS

A. Physical similarity

Two phenomena can be considered to be similar (despite different geometrical scales) if three types of similarity match: geometric, kinematic, and dynamic. Criteria of geometric similarity require that the ratios of main corresponding dimensions on the model and the original pattern be constant. Also, main corresponding angles on the original pattern and the model must be of the same value. Criteria of kinematic similarity require that the ratios of velocities at corresponding points be the same for both the

DOI: 10.1515/msr-2017-0007

original pattern and the model. Criteria of dynamic similarity require that the ratios of the main forces at corresponding points be the same for both the original pattern and the model.

Forces can be divided into two groups: areal forces and volume (weight) forces. Areal forces include friction forces, compression forces, and capillary (surface) forces. Volume (weight) forces include inertial forces, gravity forces, and impulse forces (resulting from the change in momentum). According to the type of phenomena, these forces can be put into mathematical relation and criteria (numbers) of similarity can be established. In fluid mechanics, the Reynolds number, Euler number, Newton number, Froude number, Weber number, and Mach number are the most widely known criteria. Each of them expresses ratio between two different forces. In practice, it is not possible to achieve correspondence between the original pattern and the model in all criteria. Therefore, it is always up to the investigators who must use their knowledge and experience to choose the right and most important criterion (or criteria) for the investigated phenomenon. As a result, investigators usually work with one or two dominant criteria of similarity [7], [8], [9].

B. Froude number

The Froude number expresses the ratio between gravity forces and inertial forces. Gravity forces cause vertical movements of the plume (climbing or descending) and inertial forces cause horizontal movements of the plume. The Froude number can be therefore considered as a criterion of dynamic similarity, which should be of the same value for both the scaled model and the real pattern (see [7], [8], [9]).

The Froude number can be defined as

$$Fr \approx \frac{F_{I-air}}{F_{G-pollutant}} = \frac{\rho_{air} \cdot S \cdot v_{air}^{\ 2}}{\rho_{pollutant} \cdot g \cdot V} = \\ = \frac{\rho_{air} \cdot l_2 \cdot l_3 \cdot v_{air}^{\ 2}}{\rho_{pollutant} \cdot g \cdot l_1 \cdot l_2 \cdot l_3} = \frac{\rho_{air} \cdot v_{air}^{\ 2}}{\rho_{pollutant} \cdot g \cdot l_1} \qquad (1)$$

where F_{I-air} is the inertial force due to the air acting on pollutant element [N], $F_{G-pollutant}$ is the gravity force acting on pollutant element [N], ρ_{air} is the air density [kg/m^3], $\rho_{pollutant}$ is the pollutant density [kg/m^3], S is the surface of pollutant element acted on by the flowing air [m^2], v_{air} is the air flow velocity [m/s], g is the gravity acceleration constant [m/s^2], V is the volume of pollutant element released from the pollutant source per 1 second [m^3], l_1 is the 1st characteristic dimension of the pollutant source (length of pollutant cubic element) [m], l_2 is the 2nd characteristic dimension of the pollutant source (width of pollutant cubic element) [m] and l_3 is the 3rd characteristic dimension of the pollutant source (height of pollutant cubic element) [m]. l_3 can be replaced by $v_{pollutant}$ that represents the velocity of the pollutant released from the source in vertical direction [m/s]. Fr is a dimensionless constant [-] whose value determines whether the inertial force or the gravity force will dominate in the specific pollutant plume motion scenario.

Important characteristics are illustrated in Fig.1. Pollutant element was simplified into a rectangular cuboid with dimensions of l_1, l_2 and l_3 to make practical calculations easier. Of course, the spout of a real chimney can be of a different shape, most commonly circular or elliptical. In this case, it is advisable to calculate the surface of the spout and transform the shape into a square or a rectangle with dimensions of l_1 and l_2. The value of l_3 remains the same (despite the shape of the spout) and is replaced by the velocity with which the pollutant leaves the source in vertical direction $v_{pollutant}$.

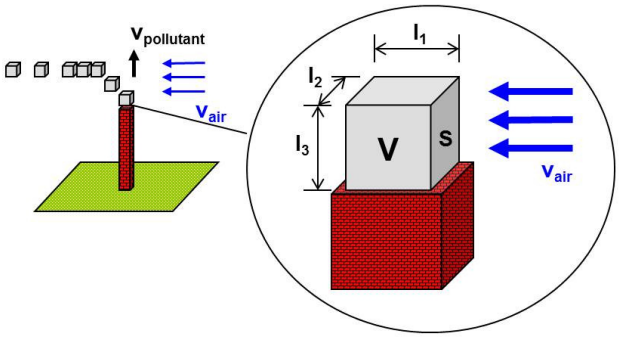


Fig.1. Air flow action on gas pollutant element leaking from the nozzle (chimney).

If $Fr < 1$, gravity forces are assumed greater than inertial forces. Hence, vertical motions (climbing or descending) of the gas pollutant plume can be expected due to different densities of the pollutant and the air. Plumes of light gas pollutants will tend to climb, whereas plumes of heavy gas pollutants will tend to descend.

If $Fr = 1$, gravity forces are assumed equal to inertial forces. Hence, gas pollutant plumes are carried by flowing air along with manifesting partial vertical motions.

If $Fr > 1$, inertial forces are assumed greater than gravity forces. Hence, vertical motions of the gas pollutant plume are limited or nonexistent. The gas pollutant plume is carried by strong flowing air, regardless of the pollutant - air density difference or weight of the pollutant.

This third scenario causes common difficulties when planning gas pollutant plume motion and dispersion experiments with downscaled models in low-speed wind tunnels. At small dimensions of measuring sections of common wind tunnels and, thus, low scales of models, the air flow may be too great to allow vertical motions of gas pollutant plumes. Proper conditions often cannot be assured in such cases.

C. Calculation of air flow velocity for inertial and gravity forces maintained in balance (Fr = 1)

According to (1), the air flow velocity v_{air} influences the inertial force F_{I-air} that causes gas pollutant horizontal motion. If all the other physical characteristics are constant, the following principles can be formulated: The greater is the air flow velocity v_{air}, the greater is the inertial force F_{I-air}. The greater is the inertial force, the more limited are the pollutant plume vertical motions (climbing or descending).

The air flow velocity $v_{air,Fr=1}$ for Fr = 1 (i.e., inertial and gravity forces are equal) can be deduced from (1):

$$1 = \frac{F_{I-air}}{F_{G-pollutant}} = \frac{\rho_{air} \cdot S \cdot v_{air,Fr=1}^{\ 2}}{\rho_{pollutant} \cdot g \cdot V} = \\ = \frac{\rho_{air} \cdot l_2 \cdot l_3 \cdot v_{air,Fr=1}^{\ 2}}{\rho_{pollutant} \cdot g \cdot l_1 \cdot l_2 \cdot l_3} = \frac{\rho_{air} \cdot v_{air,Fr=1}^{\ 2}}{\rho_{pollutant} \cdot g \cdot l_1} \quad (2)$$

Thus, the air flow velocity $v_{air,Fr=1}$ is

$$v_{air,Fr=1} = \sqrt{\frac{\rho_{pollutant} \cdot g \cdot l_1}{\rho_{air}}} = \\ = \sqrt{\frac{\rho_{pollutant} \cdot g \cdot l_1 \cdot l_2 \cdot l_3}{\rho_{air} \cdot l_2 \cdot l_3}} = \sqrt{\frac{\rho_{pollutant} \cdot g \cdot V}{\rho_{air} \cdot S}} \quad (3)$$

However, one must realize that the change in air flow velocity influences also air flow field turbulent characteristics. For example, turbulent intensity is influenced when the air flows around solid objects or in a complex terrain. Investigators must consider whether these changes have a serious impact on accuracy of the experiment or mathematical model. This is very important for modeling gas pollutant motion and dispersion in a complex geometry (complex terrain) where the real model of turbulent flow field is the key element of the simulation. If the air turbulent flow field is seriously influenced by the change in the air flow velocity, the results of the analysis can be misleading. This approach is therefore not suitable for such cases and a different parameter of the model must be changed.

D. Calculation of pollutant density for inertial and gravity forces maintained in balance (Fr = 1)

According to (1), the pollutant density $\rho_{pollutant}$ influences the gravity force $F_{G-pollutant}$ that causes gas pollutant vertical motions (climbing or descending). If all the other physical characteristics are constant, the following principles can be formulated: If the pollutant density $\rho_{pollutant}$ is greater than the air density ρ_{air}, the pollutant tends to descend (i.e., the gas pollutant plume descends). If the pollutant density $\rho_{pollutant}$ is lower than the air density ρ_{air}, the pollutant tends to climb (i.e., the gas pollutant plume climbs). The greater are the gravity forces $F_{G-pollutant}$, the more significant are the gas pollutant plume vertical movements (climbing or descending).

The pollutant density $\rho_{pollutant,Fr=1}$ for Fr = 1 (i.e., inertial and gravity forces are equal) can be deduced from (1):

$$1 = \frac{F_{I-air}}{F_{G-pollutant}} = \frac{\rho_{air} \cdot S \cdot v_{air}^{\ 2}}{\rho_{pollutant,Fr=1} \cdot g \cdot V} = \\ = \frac{\rho_{air} \cdot l_2 \cdot l_3 \cdot v_{air}^{\ 2}}{\rho_{pollutant,Fr=1} \cdot g \cdot l_1 \cdot l_2 \cdot l_3} = \frac{\rho_{air} \cdot v_{air}^{\ 2}}{\rho_{pollutant,Fr=1} \cdot g \cdot l_1} \quad (4)$$

Thus, the pollutant density $\rho_{pollutant,Fr=1}$ is

$$\rho_{pollutant,Fr=1} = \frac{\rho_{air} \cdot v_{air}^{\ 2}}{g \cdot l_1} = \frac{\rho_{air} \cdot v_{air}^{\ 2} \cdot l_2 \cdot l_3}{g \cdot l_1 \cdot l_2 \cdot l_3} = \\ = \frac{\rho_{air} \cdot v_{air}^{\ 2} \cdot S}{g \cdot V} \quad (5)$$

A change in pollutant density in order to achieve the optimum ratio between inertial and gravity forces would be often the ideal solution. However, there is a problem. The densities of pollutants range within a narrow interval - approximately of one order of magnitude - which is usually not enough to compensate the Froude number differences resulting from, e.g., a substantial change of the model scale.

A typical example can be the physical modeling of gas pollutant plumes in low-speed wind tunnels where the scale of the model is around 1:1000. In such a case, there is a need to change the pollutant density 100, or even 1000 times, which is impossible. This is why change in pollutant density can be used to achieve only a small change in the Froude number. These small changes, however, may not be sufficient for a successful execution of the experiment or numerical modeling.

The change of the flowing gas density could be an alternative solution. For example, air could be replaced by a different gas with a different value of density. However, this change influences turbulent flow field characteristics, which may be undesirable.

E. Calculation of model scale for inertial and gravity forces maintained in balance (Fr = 1)

According to (1), the model scale can be expressed by using the value l_1 that represents the 1st characteristic dimension of the pollutant source (i.e., length of pollutant

cubic element). The model scale influences both inertial and gravity forces and changes their ratio. If all the other physical characteristics are constant, the following principles can be formulated: The greater is the model scale, the greater is the influence of gravity forces. The smaller is the model scale, the greater is the influence of inertial forces. Gravity forces are, e.g., greater in a model scaled at 1:4 than in one scaled at 1:1000.

The value of the 1st characteristic dimension of the pollutant source (length of pollutant cubic element) $l_{1,\,Fr=1}$ for Fr = 1 (inertial and gravity forces are equal) can be deduced from (1):

$$1 = \frac{F_{I-air}}{F_{G-pollutant}} = \frac{\rho_{air} \cdot S_{Fr=1} \cdot v_{air}^{\,2}}{\rho_{pollutant} \cdot g \cdot V_{Fr=1}} = \\ = \frac{\rho_{air} \cdot l_{2,\,Fr=1} \cdot l_{3,\,Fr=1} \cdot v_{air}^{\,2}}{\rho_{pollutant} \cdot g \cdot l_{1,\,Fr=1} \cdot l_{2,\,Fr=1} \cdot l_{3,\,Fr=1}} \qquad (6)$$

or

$$1 = \frac{F_{I-air}}{F_{G-pollutant}} = \frac{\rho_{air} \cdot v_{air}^{\,2}}{\rho_{pollutant,\,Fr=1} \cdot g \cdot l_{1,\,Fr=1}} \qquad (7)$$

Thus, $l_{1,\,Fr=1}$ is

$$l_{1,\,Fr=1} = \frac{\rho_{air} \cdot v_{air}^{\,2}}{\rho_{pollutant} \cdot g} = \frac{\rho_{air} \cdot v_{air}^{\,2} \cdot l_{2,\,Fr=1} \cdot l_{3,\,Fr=1}}{\rho_{pollutant} \cdot g \cdot l_{2,\,Fr=1} \cdot l_{3,\,Fr=1}} = \\ = \frac{\rho_{air} \cdot v_{air}^{\,2} \cdot S_{Fr=1}}{\rho_{pollutant} \cdot g \cdot S_{Fr=1}} \qquad (8)$$

The model scale $M_{Fr=1}$ [-] for Fr = 1 (inertial and gravity forces are equal) is given by

$$M_{Fr=1} = \frac{1}{X} \qquad (9)$$

where

$$X = \frac{l_1}{l_{1,\,Fr=1}} \qquad (10)$$

The value l_1 represents the 1st characteristic dimension of the pollutant source (length of pollutant cubic element) in the original model. The value $l_{1,\,Fr=1}$ is the 1st characteristic dimension of the pollutant source in the model where Fr = 1 (i.e., inertial and gravity forces are equal).

$$F_{I1} = X^2 \cdot F_{I2} \qquad (11)$$

The value F_{I1} is an inertial force of original pollutant cubic element and F_{I2} is an inertial force of scaled pollutant cubic element. If the model scale is changed, the change of the inertial force is given by

$$F_{G1} = X^3 \cdot F_{G2} \qquad (12)$$

The value F_{G1} is a gravity force of original pollutant cubic element and F_{G2} is a gravity force of scaled pollutant cubic element. The value X is a model scale factor [-]. If X > 1, the model is smaller than its original pattern (i.e., the model is downscaled). If X < 1, the model is larger than its original pattern (i.e., the model is enlarged). Equations (11) and (12) are deduced from (1) for the Froude number.

A change in model scale always causes a change in the ratio of inertial and gravity forces. Therefore, some problems cannot be reliably modeled at other than approximately original scales. The range of deviation from the original depends on the discretion of investigators. Investigators must decide whether the tolerance of results is acceptable.

Modeling of gas pollutant plume motions with severely downscaled models is a typical example of this problem. Downscaled pollutant plume models will not correspond to the originally scaled patterns without modifying some key physical characteristics (air flow velocity, turbulent characteristics, etc.).

The problem can be solved by using a numerical mathematical model verified by a clearly defined experiment of the same type of physical phenomenon. Once the numerical model is verified, it can be used for numerical simulation of any problem of the same physical principles, whatever the model scale is. Sometimes it is impossible to do the same with a physical experiment.

It can be concluded that severely downscaled experiments are not suitable for modeling gas pollutant plume motions because of possible absence of vertical movements. It is more advisable to use physical experiment data only for verification of the numerical model (code, software).

F. Example of CFD numerical simulation

ANSYS Fluent 15.0, one of the world's most sophisticated CFD codes, was chosen for the numerical simulation of the gas pollutant plume motion and dispersion. The gauging section of the low-speed wind tunnel (length 2 m, width 1.5 m, height 1.5 m, model scale 1:1000) with a small nozzle (diameter 0.0035 m, height 0.02 m, scale 1:1000) representing a chimney in a flat, simple terrain was the object of the numerical simulation. Gas pollutant enters the gauging section through the top of the nozzle (chimney) and is carried by flowing air (see Fig.2.).

RANS (Reynolds-averaged Navier-Stokes equations) approach was used for turbulent characteristics definition. The Boussinesq hypothesis of swirl turbulent viscosity was applied for the turbulent viscosity calculation. RNG $k-\varepsilon$ model of turbulence was used for the air flow field basic

calculation (see [1], [2]). Species transport model was used for the species motion calculation. Both models worked simultaneously. No additional gas pollutant dispersion model was applied. The operating pressure was set at 101 325 Pa, the operating temperature was 300 K, and the gravity acceleration was -9.81 m/s^2 in the geometry. Considering the pollutant source close surroundings, the ranges of the Reynolds number Re were 250-1250 [-] (model scale 1:1000, referential air flow velocity 1-5 m/s, and nozzle spout diameter 0.0035 m) and 250000-1250000 [-] (model scale 1:1, referential air flow velocity 1-5 m/s, and chimney spout diameter 3.5 m).

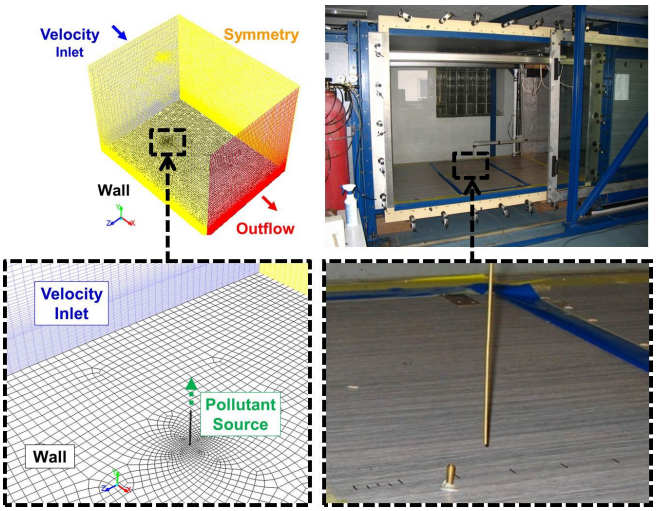


Fig.2. Geometry (gauging section of the low-speed wind tunnel) and pollutant source (numerical model versus physical experiment).

Boundary conditions were set to *Velocity Inlet* at the inlet, *Outflow* at the outlet, *Wall* for the floor, *Symmetry* for the walls, *Wall* for pollutant source walls, and *Velocity Inlet* for the nozzle (spout of the chimney).

Profiles of the flow field physical characteristics were determined at the inlet of the geometry (see Table 1.) based on experimental data from a low-speed wind tunnel for a 1:1000-scale model (see [3], [4], [5], [6], [11]). For other model scales, the profiles were modified to keep the trend of curves.

In Table 1. the parameter v_x represents the air flow velocity in the direction of X-axis [m/s], I is the intensity of turbulence [%], Y is the vertical coordinate of the geometry [m], k is the turbulent kinetic energy [m^2/s^2], and ε is the turbulent dissipation rate [m^2/s^3].

The pollutant source was designed as a nozzle (chimney). For all model scales, the pollutant velocity $v_{pollutant}$ was set at 0.5 m/s, the intensity of turbulence in the pollutant source at 10 %, the pollutant mass fraction in the pollutant source at 0.95 [-] and the air mass fraction in the pollutant source at 0.05 [-]. The hydraulic diameter of the pollutant source was set at 3.5 m for 1:1-scale model, 0.866 m for 1:4.04-scale model, 0.0986 m for 1:35.51-scale model, 0.344 m for 1:101.821-scale model, and 0.0035 m for 1:1000-scale model scale 1:1000.

Table 1. Air flow velocity profile and turbulent characteristics profiles in geometry [3], [4], [5], [11].

Vertical profile	**Equation**
Air flow velocity (X-direct.)	$v_x = 0.2371 \cdot ln(Y + 0.00327) + 1.3571$
Air turbulent intensity	$I = -0.0673 \cdot ln(Y + 0.00327) + 0.1405$
Turbulent kinetic energy	$k = 1.5 \cdot (v_x \cdot I)^2$
Turbulent dissipation rate	$\varepsilon = \frac{1.225 \cdot 0.09 \cdot (k^3)}{1.4}$

Three different pollutants were chosen to be tested: helium, methanol and 1,2-dichlorethane. Helium (ρ = 0.1625 kg/m^3) has a lower density than air, i.e., it is lighter than air (ρ = 1.225 kg/m^3). Methanol (ρ = 1.43 kg/m^3) has approximately the same density as air, i.e., it is approximately of the same weight as air. 1,2-dichlorethane (ρ = 4.1855 kg/m^3) has a greater density than air, i.e., it is heavier than air. Plumes of pollutants lighter than air tend to climb, whereas those heavier than air tend to descend. However, this is not always the case. Pollutant plume vertical movements can be influenced by several other physical factors as demonstrated in the analysis.

3. RESULTS

The results can be differentiated in three categories presented below.

A. Analysis of results by air flow velocity

The aim of this analysis is to compare gas pollutant plume shapes and motions for three different gas pollutants (helium, methanol and 1,2-dichlorethane) at different values of the air flow velocity v_{air}. The demonstration of the problem was performed with a 1:1-scale three-dimensional geometry representing the real pattern of a simple terrain with a chimney. The referential air flow velocities v_{air} at the level of the chimney spout (pollutant source) were 1 m/s, 3 m/s and 5 m/s.

Results were calculated using the ANSYS Fluent 15.0 software and were visualized in Fig.3. The contours were plotted in two-dimensional planes of the geometry, sc., the central vertical longitudinal plane, the floor (ground) plane, and the outlet plane.

The figure shows that with increasing air flow velocity v_{air} the pollutant plume vertical movements are reduced. The pollutant plume inclines horizontally at the level of the

chimney spout (pollutant source) showing no tendency to climb or descend. This is because the inertial force F_{I-air} increases as the air flow velocity v_{air} increases. Hence, the pollutant plume vertical movements are reduced or totally eliminated.

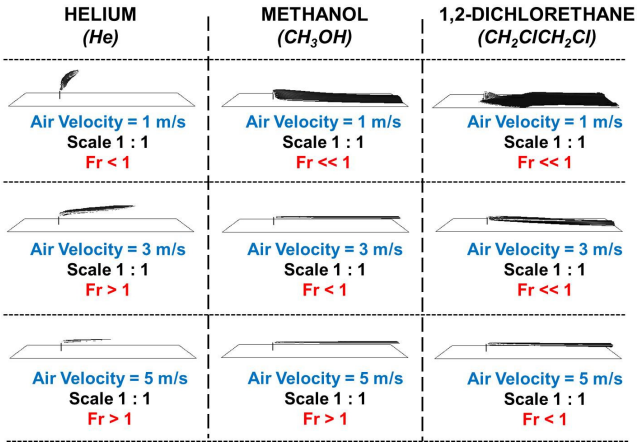


Fig.3. Gas pollutant plume motion analysis by air flow velocity and pollutant density (iso-surfaces of gaseous helium, methanol, and 1,2-dichlorethane concentrations with mass fraction of 0.0001 [-] for one model scale).

The air flow velocity also influences the size and shape of the pollutant plume. With increasing air flow velocity v_{air} the pollutant plume tends to be narrower and longer. However, a further increase in the air flow velocity makes the pollutant plume shorter because of greater rate of the pollutant dispersion. The plume range at certain concentration of the pollutant therefore decreases with increasing air flow velocity.

B. Analysis of results by pollutant density

The aim of this analysis is to compare gas pollutant plume shapes and motions for three different gas pollutants (helium, methanol and 1,2-dichlorethane) at different values of their density. Demonstration of the problem was performed with a 1:1-scale three-dimensional geometry representing the real pattern of a simple terrain with a chimney. The referential air flow velocity v_{air} at the level of the chimney spout (pollutant source) was 1 m/s.

Results were calculated using the ANSYS Fluent 15.0 software and were visualized in Fig.3. The contours were plotted in two-dimensional planes of the geometry, sc., the central vertical longitudinal plane, the floor (ground) plane, and the outlet plane.

The figure shows that: If the pollutant density $\rho_{pollu\,tan\,t}$ is lower than the air density ρ_{air} the gas pollutant plume tends to climb (for helium see Fig.3.). If the pollutant density $\rho_{pollu\,tan\,t}$ is approximately the same as the air density ρ_{air} the gas pollutant plume neither climbs nor descends (for methanol see Fig.3.). If the pollutant density $\rho_{pollu\,tan\,t}$ is greater than the air density ρ_{air} the gas pollutant plume tends to descend (for 1,2-dichlorethane see Fig.3.). The range of vertical movements is determined by gravity force $F_{G-pollu\,tan\,t}$ that influences pollutant plume at given conditions. The greater the gravity force $F_{G-pollu\,tan\,t}$ is compared to the inertial force F_{I-air}, the more significant is the vertical movement of the plume, i.e., light pollutant plume climbs and heavy pollutant plume descends.

The pollutant density $\rho_{pollu\,tan\,t}$ also influences the pollutant plume dispersion. The greater is the pollutant density, the longer is the range of the plume. At given air flow velocity v_{air}, the plume dispersion of pollutants with a low density is faster and easier than that of pollutants with a greater density.

C. Analysis of results by model scale

The aim of this analysis is to compare gas pollutant plume shapes and motions for three different gas pollutants (helium, methanol, and 1,2-dichlorethane) at different model scales. The demonstration of the problem was performed with a three-dimensional geometry at three selected scales for each of the three pollutants. The first 1:1-scale model represents the real pattern of a simple terrain with a chimney (pollutant source) where Fr < 1, i.e., the gravity force $F_{G-pollu\,tan\,t}$ is greater than the inertial force F_{I-air}. The second model (scaled at 1:4.04, 1:35.51, and 101.82, respectively) represents the state when Fr = 1, i.e., the gravity force $F_{G-pollu\,tan\,t}$ equals the inertial force F_{I-air}. The third 1:1000-scale model represents the gauging section of a low-speed wind tunnel with a nozzle (pollutant source) on the floor where Fr > 1, i.e., the inertial force F_{I-air} is greater than the gravity force $F_{G-pollu\,tan\,t}$. The referential air flow velocity v_{air} at the level of the chimney spout (pollutant source) was 1 m/s.

Results were calculated using the ANSYS Fluent 15.0 software and were visualized in Fig.4. Contours were plotted in two-dimensional planes of the geometry, sc. the central vertical longitudinal plane, the floor (ground) plane, and the outlet plane.

The figure shows that: If the model scale changes and all other characteristics remain unchanged, the inertial and gravity forces and their ratio change too. Therefore, the size, shape, and inclination of the pollutant plume change.

According to (11), the inertial force is proportional to the square of the model scale. According to (12), the gravity force is proportional to the third power of the model scale. Therefore, the change in the gravity force due to the change of the model scale is considerably greater than the change in the inertial force. The lower is the model scale, the greater is the dominance of inertial forces compared to gravity forces, and vice versa.

Also, the greater the pollutant density, the lower the model scale if Fr = 1, i.e., the inertial and gravity forces are equal.

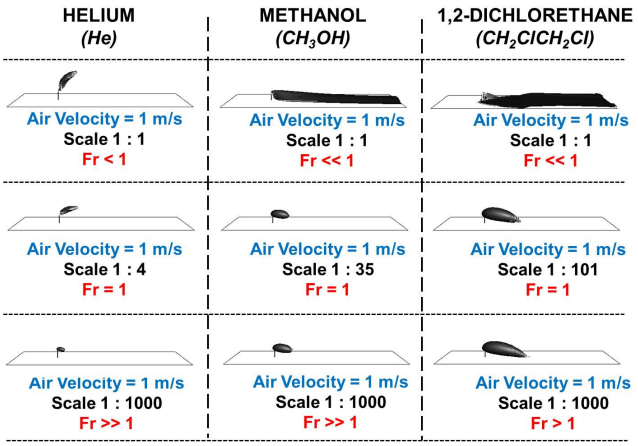


Fig.4. Gas pollutant plume motion analysis by model scale (iso-surfaces of gaseous helium, methanol, and 1,2-dichlorethane concentrations with mass fraction of 0.0001 [-] for three different model scales).

4. Discussion / Conclusions

The aim of the analyses was to lay down principles for physical and mathematical modeling of gas pollutant plume motion and dispersion in real atmospheric conditions. The influences of the air flow velocity, pollutant's density, and model scale on pollutant plume size, shape, and inclination were investigated.

The Froude number was chosen as a criterion of physical similarity for the pollutant plume behavior in the atmosphere. Basic mathematical rules and principles were formulated upon study of available fluid mechanics literature (see [7], [8], [9], [10], [11], [12], [13], [14]). Next, all mathematical and physical assumptions were verified by numerical simulation using the ANSYS Fluent 15.0 software. Air flow field was modeled using the RNG $k-\varepsilon$ model of turbulence, the gas pollutant motion was modeled using the Species Transport Model, both in the same three-dimensional geometry consisting of 569 490 grid cells. Turbulent characteristics were defined using RANS approach. No additional dispersion model was applied.

Object of modelling was gauging section of the low-speed wind tunnel (for model scale of 1:1000) or big real terrain (for model scale of 1:1) with a pollutant source in the form of a nozzle (or chimney, respectively) situated on the section floor (ground). The gauging section with the nozzle represented a chimney in a simple, flat terrain. The chimney was considered to be a pollutant source for three different gas pollutants (helium, methanol, and 1,2-dichlorethane). The numerical simulation was performed for five model scales, three gas pollutants with different densities, and three different air flow velocities. The simulations were steady (time-independent) with the accuracy of 0.0001 (criterion of convergence). Final results were visualized as pollutant concentration fields with the concentration limit value of 0.001. The contours were plotted in two-dimensional planes of the geometry, sc. the central vertical longitudinal plane, the floor (ground) plane, and the outlet plane. The numerical model had been verified by an experiment performed in a low-speed wind tunnel (see [3], [4], [5]).

The following principles based on the results of the Froude number analysis of pollutant plume motion and dispersion in real atmosphere can be defined:

1) The greater is the air flow velocity, the greater are the inertial forces. These forces influence pollutant plume and reduce its vertical motions (inclination). With increasing air flow velocity, the pollutant plume inclines horizontally at the level of the chimney spout (pollutant source), but with further increase in the air flow velocity it becomes narrower and shorter.

2) The greater is the difference between pollutant density and air density, the more significant is the tendency towards vertical movements (climbing or descending) of the plume. The plume of pollutant with lower density than air tends to climb, whereas the plume of pollutant with greater density than air tends to descend. The density of the pollutant also influences the pollutant plume dispersion. The greater is the pollutant density, the longer is the range of the plume. At given air flow velocity v_{air}, the plume dispersion of pollutants with a low density is faster and easier than that of pollutants with a greater density.

3) If the model scale changes and all other characteristics remain unchanged, the inertial and gravity forces and their ratio change too. Therefore, the size, shape and inclination of the pollutant plume change. The inertial force is proportional to the square of the model scale, whereas the gravity force is proportional to the third power of the model scale. The change in the gravity force due to the change of model scale is considerably greater than the change in the inertial force. The lower is the model scale, the greater is the dominance of inertial forces compared to gravity forces. Also, the greater is the pollutant density, the lower is the model scale if $Fr = 1$, i.e., the inertial and gravity forces are equal.

7. Conclusion

From the above it follows that if investigators want to respect and follow the basics of physical phenomena, they must consider criteria of physical similarity very carefully, in particular criteria of dynamic similarity. Some physical phenomena, however, cannot be modeled in any model scale but the original one without changing the basis of the phenomena.

This analysis is intended for those who are interested in gas pollutant plume motion in the atmosphere and in theory of physical similarity. The conclusions of the analysis can be used for further experiment design works or for checking results of mathematical modeling.

Acknowledgment

Authors acknowledge the financial support of the SPII 1a10 45/07 project of the Ministry of the Environment of the Czech Republic.

REFERENCES

[1] Kozubková, M. (2008). *Modeling of Fluid Flow FLUENT, CFX*. Ostrava, Czech Republic: VSB - Technical University of Ostrava. (in Czech)

[2] Bojko, M. (2008). *Guide for Training of Flow Modeling – FLUENT*. Ostrava, Czech Republic: VSB - Technical University of Ostrava. (in Czech)

[3] Civiš, S., Zelinger, Z., Střižík, M., Jaňour, Z. (2001). Simulation of air pollution in a wind tunnel. In *Spectroscopy from Space: NATO Science Series II-Mathematics Physics and Chemistry*. Kluwer Academic Publishers, 275-299.

[4] Zelinger, Z., Střižík, M., Kubát, P., Jaňour, Z., Berger, P., Černý, A., Engst, P. (2004). Laser remote sensing and photoacoustic spectrometry applied in air pollution investigation. *Optics and Lasers Engineering*, 42 (4), 403-412.

[5] Zelinger, Z., Střižík, M., Kubát, P., Civiš, S., Grigorová, E., Janečková, R., Zavila, O., Nevrlý, V., Herecová, L., Bailleux, S., Horká, V., Ferus, M., Skřínský, J., Kozubková, M., Drábková, S., Jaňour, Z. (2009). Dispersion of light and heavy pollutants in urban scale models: CO_2 laser photoacoustic studies. *Applied spectroscopy*, 63 (4), 430-436.

[6] Zavila, O., Herecová, L., Míček, D., Hejzlar, T. (2011). Numerical simulation of heavy and light pollutants motion as a tool of experimental data verification. *Communications*, 13 (2), 37-43.

[7] Čarnogurská, M., Přihoda, M. (2011). *Application of Three-Dimensional Analysis for Modelling Phenomena in the field of Power Engineering*. Kosice, Slovak Republic: Technical University of Kosice. (in Slovak)

[8] Incropera, F.P., Dewitt, D.P., Bergman, T.L., Lavine, A.S. (2007). *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons.

[9] Shaughnessy, E.J., Katz, M.I., Schaffer, J.P. (2005). *Introduction to Fluid Mechanics*. Oxford University Press.

[10] Drábková, S., Platoš, P. (2003). Numerical simulation as a tool for the solution and understanding of practical air pollution problems. In *Proceedings of the Conference on Modelling Fluid Flow (CMFF'03)*. Budapest, Hungary: Budapest University of Technology and Economics.

[11] Stull, B.R. (1994). *An Introduction to Boundary Layer Meteorology*. Kluwer Academic Publishers.

[12] Caballero, R. (2014). *Physics of the Atmosphere*. IOP Publishing.

[13] Emeis, S. (2011). *Surface-Based Remote Sensing of the Atmospheric Boundary Layer*. Springer.

[14] Vallero, D.A. (2014). *Fundamentals of Air Pollution* (5th ed.). Academic Press.

Received November 18, 2016.
Accepted February 6, 2017.

MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 2, 61-67

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

ISSN 1335-8871

Field Programmable Gate Array (FPGA) Respiratory Monitoring System Using a Flow Microsensor and an Accelerometer

Idir Mellal[1], Mourad Laghrouche[1], Hung Tien Bui[2]

[1]*Mouloud Mammeri University, Faculty of electrical engineering and computing science, PO Box 17 RP 15000, Tizi Ouzou, Algeria. Lampa laboratory, larouche_67@yahoo.fr*

[2] *University of Québec At Chicoutimi, G7H-2B1, Chicoutimi, Canada*

This paper describes a non-invasive system for respiratory monitoring using a Micro Electro Mechanical Systems (MEMS) flow sensor and an IMU (Inertial Measurement Unit) accelerometer. The designed system is intended to be wearable and used in a hospital or at home to assist people with respiratory disorders. To ensure the accuracy of our system, we proposed a calibration method based on ANN (Artificial Neural Network) to compensate the temperature drift of the silicon flow sensor. The sigmoid activation functions used in the ANN model were computed with the CORDIC (COordinate Rotation DIgital Computer) algorithm. This algorithm was also used to estimate the tilt angle in body position. The design was implemented on reconfigurable platform FPGA.

Keywords: Accelerometer, CORDIC, FPGA, respiratory monitoring, ANN, SPI, MEMS flow sensor.

1. INTRODUCTION

Respiration is one of the most important vital signs of a person. A continuous and automatic monitoring system can be a needful instrument for medical staff, especially for monitoring patients affected by respiratory diseases. In addition, simultaneous monitoring of respiratory function and activity level may be beneficial in the monitoring of chronic conditions such as chronic obstructive pulmonary disorder [1]-[2]. Hence, the development of monitoring systems has become a topical issue [1]-[7]. In the literature, many studies have explored respiration monitoring. Backer et al. [8] investigated the different physiological and technical methods of implementing respiratory monitoring systems. Other works based on different approaches have been performed particularly in engineering. Binu et al. [9] proposed a monitoring system using a thermal flow sensor, tri-axial accelerometer, and photo electric sensor to measure the air flow, the body posture, and the oxygen saturation, respectively. In 2014, Rong et al. [10] integrated 3 micro sensors with mobile communication devices including hot-film flow sensor, accelerometer, and oximeter. They built a real-time system for monitoring and diagnosing obstructive sleep apnea. The use of the silicon flow sensor and a tri-axial accelerometer placed on the torso can measure the respiration rate and the chest movement [3]-[7]. MEMS components have found increasing use in medical applications. MEMS devices offer uniqueness in their application, fabrication, and functionality. The thermal flow sensor is one of the few devices, allowing the measurement of the gas flow velocity. A miniature hot wire sensor has been recently constructed using MEMS technologies [11]-[13]. The major drawback of this type of sensor is the dependence of its characteristics on ambient temperature.

The temperature compensation methods of the silicon sensor can be divided into three methods: software, hardware, and hybrid. The software method is based on data processing and algorithms such as the ANN technique [14]-[16]. The hybrid method uses both approaches to realize the compensation. The hardware method is much more efficient and easier. It can be achieved using analog or digital signal conditioning circuits. Digital signal conditioning circuits need A/D and D/A conversion blocks that lead to considerable consumption of chip area. Other methods for digital implementation such as ASIC (Application Specific Integrated Circuit), DSP (Digital Signal Processing), and FPGA can be used. DSP-based implementations are not suitable for modeling the parallel behavior of the neurons because of their sequential nature. ASIC implementation is more efficient than FPGA in terms of power and area consumption, but it suffers from a lack of reconfigurability [17].

In this paper, we propose an alternative approach based on employing an artificial neural network for compensating the temperature-drift of the output flow sensor. The neurons in

DOI: 10.1515/msr-2017-0008

the hidden layer of the ANN model use the sigmoid activation function which has been computed with the CORDIC algorithm [19]. This algorithm proposed by Volder [20] is a simple and efficient method to calculate the trigonometric and hyperbolic function through coordinate transformation [21]-[23]. It has been used in both the temperature compensation of the flow sensor and the body position measurement. The tri-axial accelerometer is used to assess the physical activity and measure the chest movement. The accelerometer data will be transmitted with an SPI (Serial Peripheral Interface) interface that is used to compute the tilt angle and show the XYZ positions. To reduce the computational complexity and release the memory, we used a 2D CORDIC-based algorithm for the tilt angle calculations by means of the inverse tangent function. In adopting this approach, an effort has been made to fulfill the procedure in less time while maintaining high accuracy for the implementation [19]. Using the VHDL language, we developed, in this work, an FPGA implementation of a respiratory monitoring system combining a calibrated flow sensor and a tri-axial accelerometer. The tilt angle is computed with the CORDIC (Coordinate Rotation DIgital Computer) algorithm with high accuracy. To ensure the accuracy of our model, we used a CORDIC algorithm for computing a sigmoid and the inverse tangent functions. The whole mixed system is implemented on Altera FPGA DE2 board.

2. MATERIALS & METHODS

Our system contains the flow sensor with its different parts and a digital accelerometer, BMA180, with the CORDIC core to compute the tilt angle and show the XYZ position. Many works have been done to validate and test the sensors. The reliability of the sensor has been tested and various tests have been performed on the final structure [24]. Test and verification of the design were carried out by implementing the proposed technique for temperature compensation and computing the tilt angle on a reconfigurable platform based on an FPGA device. In addition to the 3 outputs of the system: tilt angle, XYZ positions and the air flow, we can add another intelligent block for other health disorders. The proposed system can be used by people suffering from apnea syndrome, asthma, or Chronic Obstructive Pulmonary Disease (COPD).

A. Description of the flow sensor

A miniature hot wire sensor has been constructed using MEMS technologies [11]-[13]. Fig.1. shows the final structure of the hot wire sensor obtained at the end of the process. First, PECVD (Plasma Enhanced Chemical Vapor Deposition) silicon nitride was deposited on p type 4" wafers to a thickness of 0.3 µm. Then, the silicon nitride film was patterned, and wet thermal oxidation of silicon made through the nitride window. Oxidation time was adjusted with technological step simulation for achieving a flat surface. A 0.3 µm depth silica cavity was obtained. Polycrystalline silicon was deposited (LPCVD – thickness 0.5 µm) and doped by boron ion implantation. After the thermal annealing, polycrystalline silicon was patterned into variable section wire. Electrical contacts were finally taken with chromium pads. The releasing of the wire was performed by wet-etching of the silica cavity after dicing the sensors. The reliability of the sensor was discussed by Bensidhoum et al. [24]. The power consumption is less than 10 mW. The size of the hot wire is (50 x 2x 0.5) μm^3. This sensor can be used as temperature or air flow sensor (Hot wire anemometer). The Temperature Coefficient of Resistance (TCR) is roughly around 0.14 %/°C.

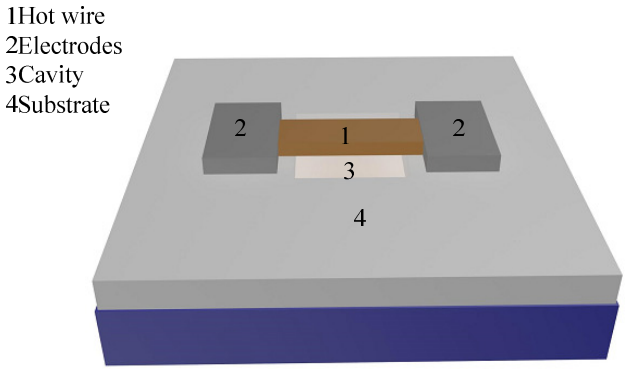


Fig.1. Description of the hot wire sensor.

B. The accelerometer

The accelerometer is a popular device used in many fields, especially in biomedical application. It can be used as a body-motion sensor to detect the physical motion parameters and the respiratory rate that can be used in many applications. In this work, the parameter of interest is the tilt angle. This can be computed using a commercial digital accelerometer sensor, BMA180, presented in Fig.2. The classical method of rectangular (x, y, z) to spherical (ρ, θ, φ) conversion can be used to relate the tilt angle θ [25].

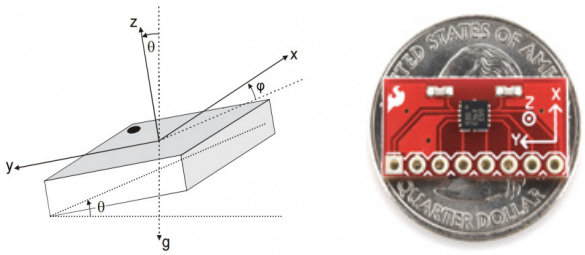


Fig.2. Representation of the BMA180 accelerometer [25].

As all other MEMS sensors, the BMA180 outputs show a high-frequency noise. The planned work consisted in implementing a Kalman filter on the FPGA to filter the accelerometer outputs and remove the high frequencies. But the power consumption, the utilized resources and the size of the device will increase, and this is not convenient for a wearable device. For this reason, we used the programmable integrated filters available on the BMA180.

C. The Altera DE2 development board

For the proof of concept, the FPGA design was implemented on the Altera DE2 board. It is a development board that contains the Cyclone II FPGA chip and contains many other devices and I/Os. Once the design is validated, the following step will be to design a custom FPGA board for the final system. The used peripherals and its connection to the Cyclone II FPGA are shown in Fig.3. [26].

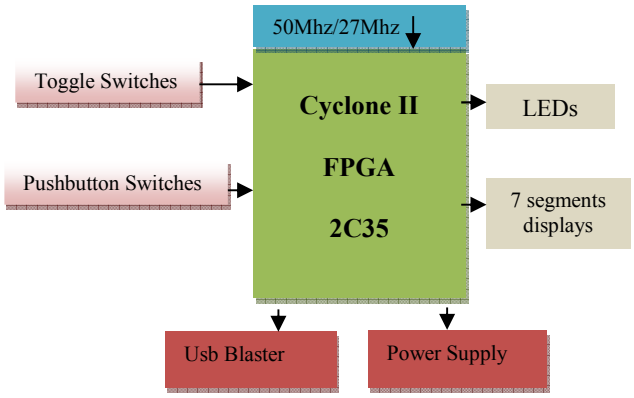


Fig.3. The DE2 development board and the used peripherals: we used the toggle and the pushbutton switches.

3. HARDWARE IMPLEMENTATION

The compensation of the flow sensor is achieved by using the ANN model and it is implemented on the FPGA with the CORDIC algorithm to compute the accelerometer output.

There are different methods of implementing an ANN and CORDIC in hardware: digital, analog, and hybrid, each one has advantages and disadvantages.

The analog method offers a low propagation time, low power consumption and small chip area of silicon. But it suffers from thermal drift, inexact computation results and lack of reprogrammability. The digital implementation offers a good precision of computing, true parallel implementation, and powerful software development tools [15], [21], [23]. It is important to note that ANNs require parallel computation. They are inspired by biological neural systems. Fig.4. presents the different devices used and the signal conditioning blocks of the sensors.

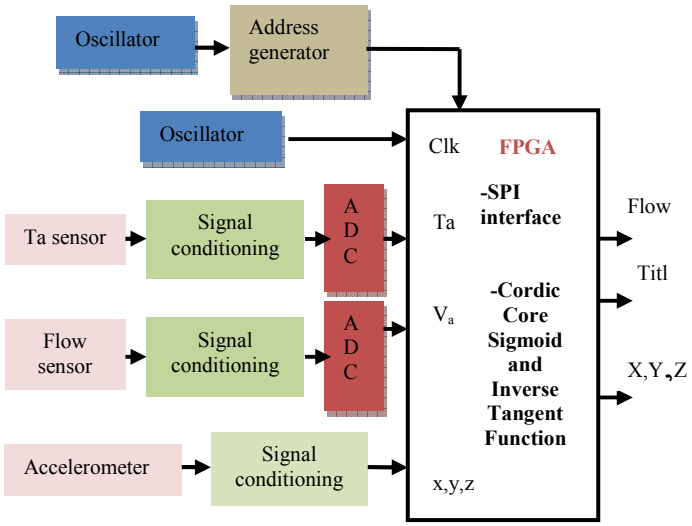


Fig.4. The architecture of the system: Va is the air velocity and Ta is the air temperature.

The oscillators are used to generate the clock and different addresses used for selection. The conditioning circuit contains: amplifier, filter, sample and hold circuits. Both ADC and DAC are used to convert the data in and data out of the FPGA. The architecture of the BMA 180 block is shown in Fig.5.

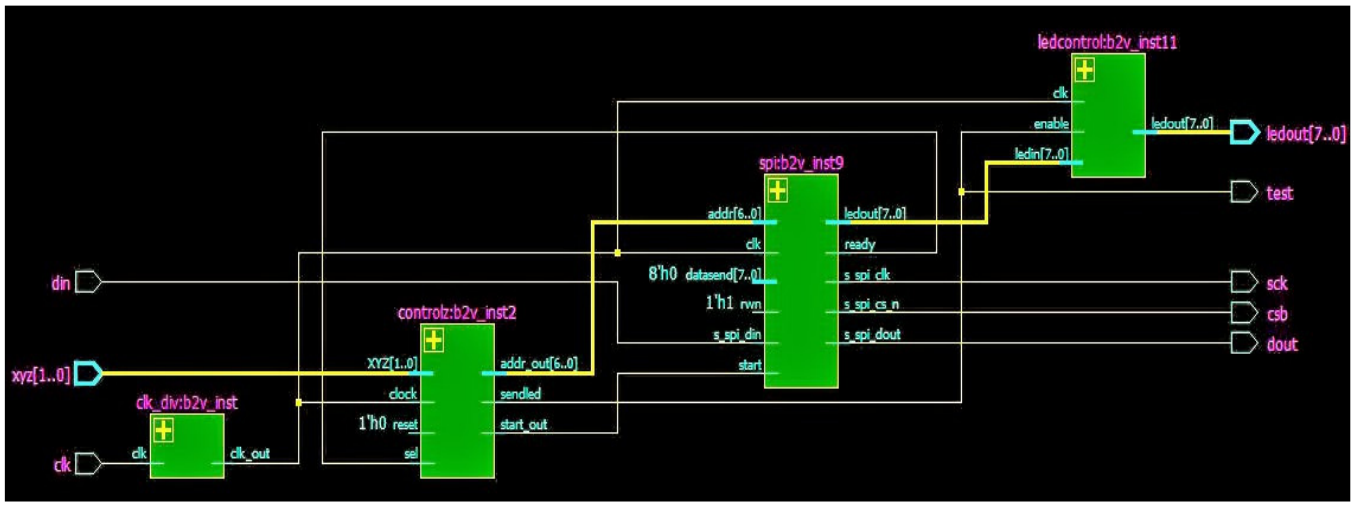


Fig.5. The architecture of the BMA180 block: we implemented the SPI protocol to carry out the data. The control block generates the clock and the addresses signals.

A. Temperature compensation and test

The probe embedded with MEMS hot-film sensors was calibrated in a closed-circuit wind tunnel. A free-stream turbulence level in the test section is very low and temperature can be controlled with a precision of ±0.5°C. Fig.6. illustrates the wind tunnel, the different wind speeds were generated from 0 m/s to 35 m/s with a step of 1 m/s.

The experiments were conducted and repeated in a room with different air temperatures 15°C, 20°C, 25°C, 30°C, 35°C, respectively. Fig.7. shows the calibration of the MEMS flow sensor with different air temperature. The characteristics of the flow sensor face two difficulties, the first difficulty is the compensation for the fluid temperatures and the other is the linearization of highly nonlinear responses. To compensate the temperature effect on the output of the sensor, the circuit should be modeled. This modeling can be achieved using inverse modeling of ANN [14]-[16].

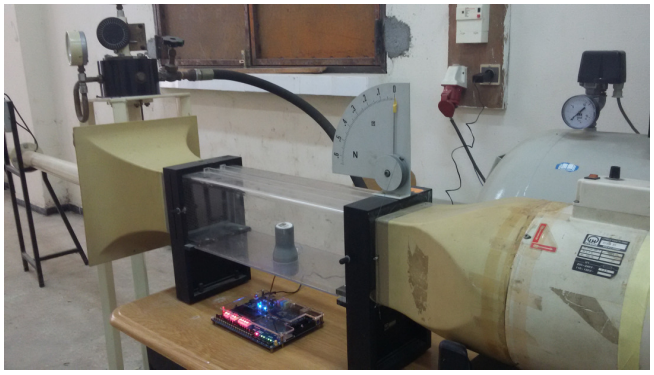


Fig.6. Wind tunnel calibration or the hot wire sensor.

Fig.8. shows the block diagram of ANN based compensation scheme for the hot wire sensor using the inverse model.

For training the neural network, the standard back propagation learning algorithm was used. Multi Layer Perceptron (MLP) is an ANN which employs three layered perceptron based on a feed forward network with signal processing neurons in its hidden and output layers. The input layer had two inputs and a bias. The number of neurons in the hidden layer was fixed at six after performing several tests. The data set used for learning and testing the ANN was composed of 310 and 90 elements, respectively. The selection of the number of neurons in the hidden layer was done on the basis of minimum Mean Square Error (MSE), weight factor and bias values suitable for hardware implementation and its dynamic range. Moreover, the number of neurons was kept as small as possible for simplicity in circuit implementation at the subsequent stage. Fig.9. presents the ANN model with two blocks. The Multiplier-accumulator block and the transfer function block. The activation function was computed with the CORDIC Algorithm. The implementation of the ANN model on the FPGA board is shown in Fig.10.

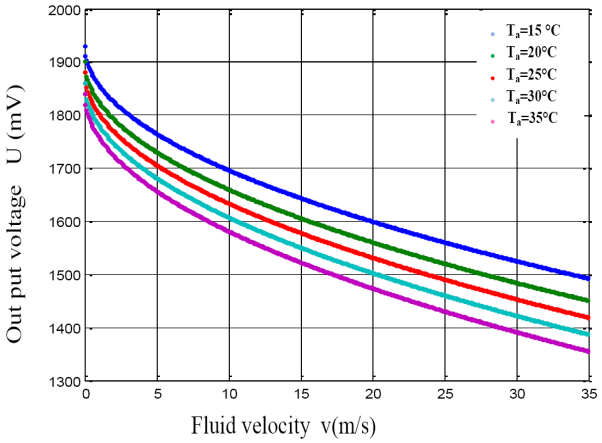


Fig.7. The response of the flow sensor for different temperatures.

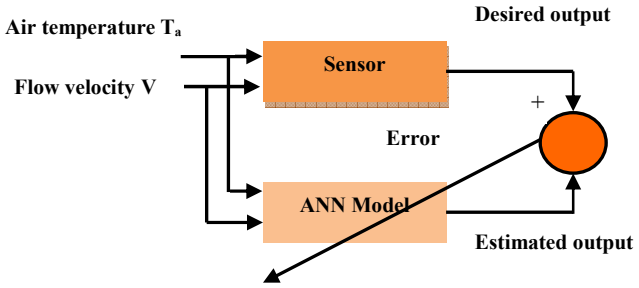


Fig.8. The scheme of the ANN compensation model: the ANN model corrects the output of the sensor by comparing the desired and the estimated output.

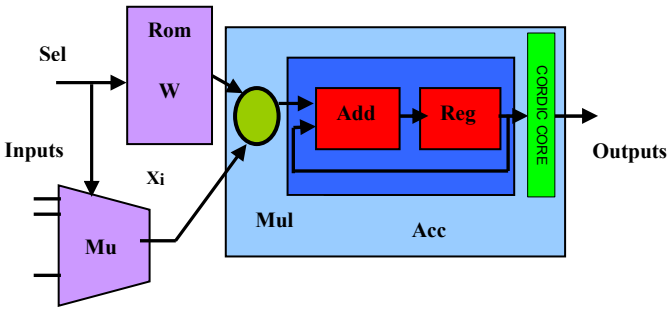


Fig.9. The neural blocks: the transfer function Sigmoid of the neuron was computed using CORDIC. A ROM memory was used to store the neuron weights, then multiply each input with the corresponding weight using the MAC (Multiplier- Accumulator).

Fig.11. shows the response of the micro flow sensor after the compensation and the linearization. After correction the maximum non-linearity of the output signal is less than 0.35 %, its resolution is 0.07 V/m.s^{-1}.

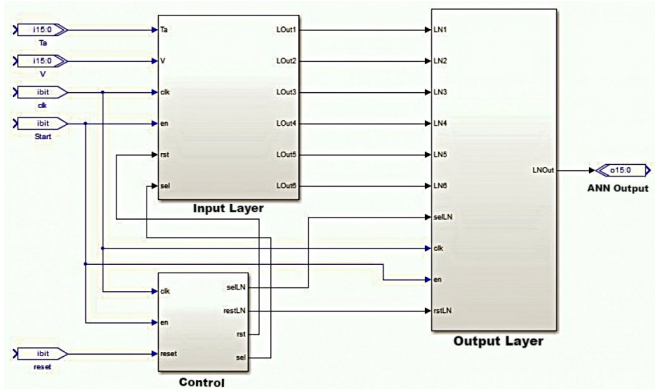


Fig.10. The Architecture of ANN implementation: the input layer is composed of 6 neurons with a sigmoid transfer function. The output layer contains one neuron with a linear transfer function. The control block generates the control and selection signals.

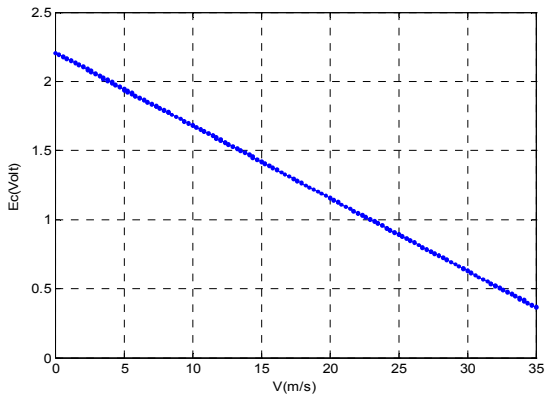


Fig.11. The response of the flow sensor after compensation and linearization

B. Tilt measurement

The BMA 180 was used to compute the tilt angle using the CORDIC algorithm (COrdinate Rotation DIgital Computer) and the inverse tangent function [22], [23]. Two basic CORDIC modes are used to compute different functions: the rotation mode and the vectoring mode [19]. The rotation-mode CORDIC determines the coordinates of any given vector after rotation through a given angle, while in the vectoring-mode it computes the magnitude and phase of the vector. The unified algorithm for linear and hyperbolic CORDICs is an extension of the basic CORDIC algorithm for circular trajectory. It is based on the generalized principle proposed in [19] to include hyperbolic and linear trajectories along with the original circular trajectory of operation. If the micro-rotation angles are restricted to $\tan^{-1}(\theta) = \pm 2^{-i}$, the multiplication is reduced just to a shift operation. That is the main advantage of the hardware CORDIC implementation. In our case; the inverse tangent function, $\theta = \text{atan}(Y/X)$, was directly computed using the vectoring mode of the CORDIC algorithm.

$$Zi = Z_0 + \tan^{-1}(Y_0/X_0) \quad (1)$$

By setting the parameters of the function to have a vectoring mode and $Z_0 = 0$, we managed to compute the $\text{atan}(Y_0/X_0)$ function using equation 1. The properties of the inverse tangent function can be implemented to simplify the architecture.

C. The data representation

Two Complement Fixed Point Format is used for representing the data and computing the outputs. The number of bits chosen for this work is 12 bits. The s2.9 representation is used with one sign bit (Sn), 2 integer bits (I0 to I1) and 9 fractional bits (f0 to f7) as shown in Table 1.

Table 1. Data structure.

Sign	I1	I0	F0	……..	F8
Sign bit	Integer part			Fraction part	

Radix point

Table 2. shows 2 numbers, with different signs.

Table 2. Numbers representation.

Format	Binary	Hexa	Decimal
S2.9	000011101101	0ED	0.4636
S2.9	111100000110	F06	-0.2449

To perform an efficient implementation the fixed point data is used, as shown previously. The accelerometer output is 8 bits, it is important to convert it to 12 bits. For this reason, we use a data converter. A sign extension is performed (the MSB bit). The following examples demonstrate the conversion from 8 to 12 bitswords.

X8= 0xxx xxxx ====> X12= 0000 0xxx xxxx
X8= 1xxx xxxx ====> X12= 1111 1xxx xxxx

Fig.12. shows the machine state of the control block. Each circle corresponds to a state and the transition between two states is represented by a directed arc.

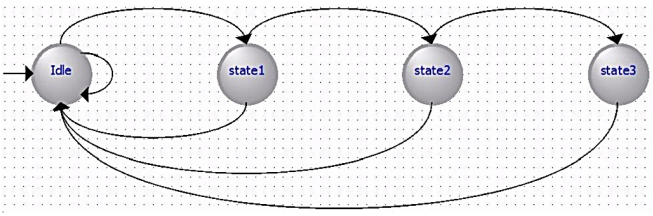


Fig.12. State diagram description of the control block.

The first state represents the idle state of the system. The transition to state1 is activated by the enable signal "en" and the rising edge clock. The sensors collect the data. The transition to the second state is activated by the signal "sel". The first layer of the ANN is selected and the tilt angle is computed using the accelerometer data. The transition to state 3 is performed by the "selLN" signal. The system will return to its idle state regardless of its present state activated

by using the signal "reset". Fig.13. shows the resources consumption on the Altera Cyclone II 2C35 FPGA device. We notice the available resources after the implementation.

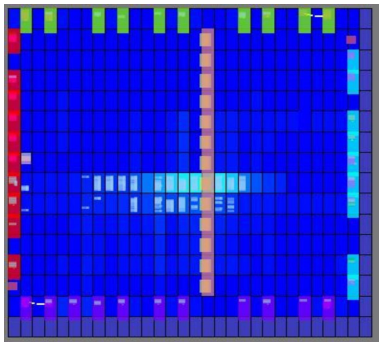


Fig.13. The resources consumption of the FPGA: the total wire utilization is 48 %.

4. RESULTS AND DISCUSSION

A respiratory monitoring system was developed using a flow sensor and a tri-axis accelerometer. The SPI (Serial Peripheral Interface) was used to obtain the data from the accelerometer for the wearable system. The SPI protocol was also implemented. The estimated error was less than 1°; $\Delta\theta = 0.89°$. It was calculated after several steps of execution of the CORDIC core. Table 3. shows the resource consumption in the FPGA.

Table 3. The resource consumption.

Registers	Pins	Logic elements	Memory
363	73/315	250 / 4,608	0

To verify the monitoring and diagnosing respiratory diseases, a series of monitoring and diagnostic tests was conducted during a pulmonary breathing cycle for a man (35-year-old). Fig.14. shows the response of the flow sensor using a nasal cannula. Due to the small dimensions of the flow sensor, we managed to introduce it in the nasal cannula to measure the flow of inspiration and expiration.

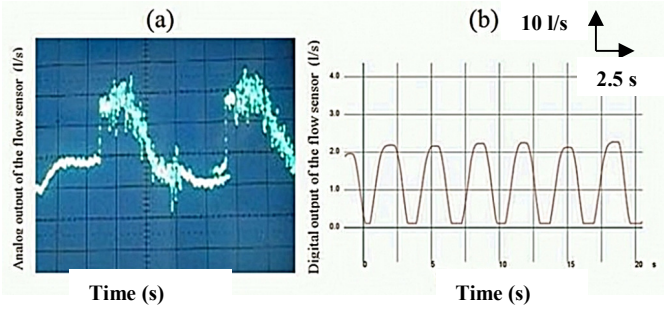


Fig.14. Flow sensor response using nasal cannula: a) the analog output for two breath cycles b) the digital output for 6 breath cycles.

Fig.15. shows the outputs of the accelerometer which is placed on the chest of the subject so that the X-axis is perpendicular to the body. The data were recorded in the laboratory during 15 min for three different positions: sitting, standing and lying supine. In future works, we aim to develop a dedicated SOC (System On Chip) to respiratory monitoring with high performances: low consumption, small size, and low cost.

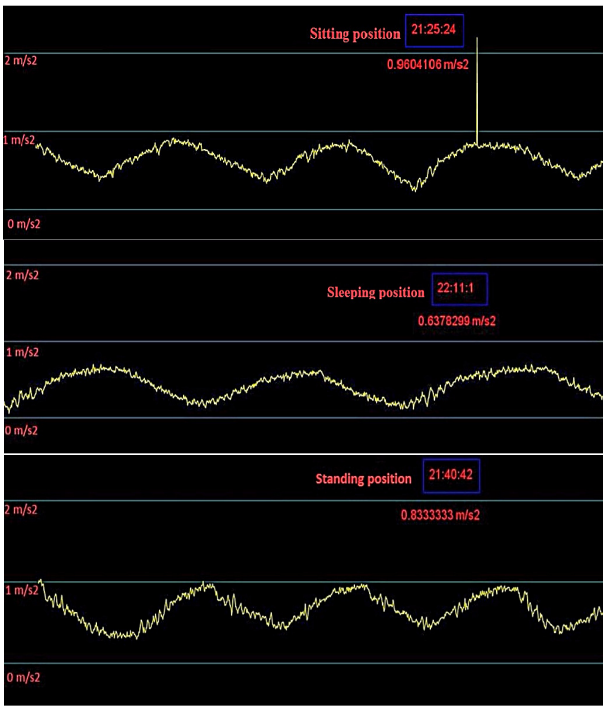


Fig.15. Accelerations along the X-axis for different positions: sitting, standing and lying supine.

Fig.14. and Fig.15. show the outputs of our system. We can see the similarity between the analog output and the digital output. Moreover, the accelerometer output can be extended for many body postures.

5. CONCLUSION

We have proposed a mixed system to monitor the respiratory activity. The monitoring system is based on mixing a flow sensor and a 3Axis accelerometer as a motion sensor. The CORDIC algorithm was used to compute the tilt angle and approximate the sigmoid function. The inverse tangent function is used to compute the tilt angle to estimate the body positions. An FPGA implementation is made on the DE2 board using the VHDL language. We have found that the respiratory intensity and the pulmonary rhythm are closely linked. The hardware implementation shows that the physical resource consumption was acceptable and the observed error was less than 1°. Therefore, this system can be extended to include other sensors and modules to serve in daily homecare endowed with the SPI interface. In future works, we aim to develop a dedicated SOC for respiratory monitoring with high performances: low consumption, small size, and low cost.

REFERENCES

[1] Vandenbussche, N.L., Overeem, S., Johannes, P., van. Dijk, J.P., Simons, P.J., Pevernagie, D.A. (2015). Assessment of respiratory effort during sleep: Esophageal pressure versus noninvasive monitoring techniques. *Sleep Medicine Reviews*, 24, 28-36.

[2] Chattopadhyay, M., Chowdhury, D. (2016). A new scheme for reducing breathing trouble through MEMS based capacitive pressure sensor. *Microsystem Technologies*, 22 (11), 2731-2736.

[3] Zadeh, E.G., Gholamzadeh, B., Charca, G.A. (2016). Toward spirometry on chip: Design, implementation and experimental results. *Microsystem Technologies*, doi: 10.1007/s00542-016-3200-0.

[4] Shany, T., Redmond, S.J., Narayanan, M.R., Lovell, N.H. (2012). Sensors-based wearable systems for monitoring of human movement and falls. *IEEE Sensors Journal*, 12 (3), 658-670.

[5] Luo, J., Wang, Z., Shen, Ch., Wen, Z., Liu, S., Cai, S., Li, J. (2015). Rotating shaft tilt angle measurement using an inclinometer. *Measurement Science Review*, 15 (5), 236-243.

[6] Nam, Y., Park, J.W. (2013). Child activity recognition based on cooperative fusion model of a triaxial accelerometer and a barometric pressure sensor. *IEEE Journal of Biomedical and Health Informatics*, 17 (2), 420-426.

[7] Cao, Z., Zhu, R., Que, R.Y. (2011). A wireless portable system with microsensors for monitoring respiratory diseases. *IEEE Transaction on Biomedical Engineering*, 59 (11), 3110-3116.

[8] Becker, D.E., Casabianca, A.B. (2009). Respiratory monitoring: Physiological and technical considerations. *Anesthesia Progress*, 56 (1), 14-22.

[9] Binu, E., Varsha, N.S. (2014). Real time monitoring of respiratory parameters using a wireless portable system. *International Journal of Engineering Development and Research*, 3 (1), 283-287.

[10] Zhu, R., Cao, Z., Que, R. (2014). Integration of micro sensors with mobile devices for monitoring vital signs of sleep apnea patients. In *9th IEEE International Conference on Nano/Micro Engineered and Molecular Systems (NEMS)*, April 13-16, 2014. IEEE, 462-466.

[11] Laghrouche, M., Montes, L., Boussey, J., Ameur, S. (2011). Low-cost embedded spirometer based on micro machined polycrystalline thin film. *Flow Measurement and Instrumentation*, 22 (2), 126-130.

[12] Laghrouche, M., Montes, L., Boussey, J., Meunier, D., Ameur, S., Adane, A. (2011). In situ calibration of wall shear stress sensor for micro fluidic application. In *Proceedings of Eurosensors XXV*, September 4-7, 2011, Athens, Greece.

[13] Makhlouf, S., Laghrouche, M., Adane, A.E.H. (2016). Hot wire sensor-based data acquisition system for controlling the laminar boundary layer near plant leaves within a greenhouse. *IEEE Sensors Journal*, 16 (8), 2650-2657.

[14] Jun, S., Kochan, O. (2014). Investigations of thermocouple drift irregularity impact on error of their inhomogeneity correction. *Measurement Science Review*, 14 (1), 29-34.

[15] Mellal, I., Laghrouche, M., Idjeri, B., Beguenane, R., Ameur, S. (2012). Implementation of ANN in FPGA for improved temperature drift of the MEMS flow sensor. *Sensors & Transducers Journal*, 145 (10), 1-9.

[16] Laghrouche, M., Idjeri, B., Hammouche, K., Tahanout, M., Boussey, J., Ameur, S. (2012). Temperature compensation of micromachined silicon hot wire sensor using ANN technique. *Microsystem Technologies*, 18 (3), 237-246.

[17] Wanhammar, L. (1999). *DSP Integrated Circuits*. Academic Press.

[18] Valls, J., Kuhlmann, M., Parhi, K.K. (2002). Evaluation of CORDIC algorithms for FPGA design. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 32 (3), 207-222.

[19] Tiwari, V., Khare, N. (2015). Hardware implementation of neural network with Sigmoidal activation function using CORDIC. *Microprocessors and Microsystems*, 39 (6), 373-381.

[20] Volder, J.E. (1959). The CORDIC trigonometric computing technique. *IRE Transactions on Electronic Computers*, EC-8 (3), 330-334.

[21] Walther, J.S. (1971). A unified algorithm for elementary functions. In *Proceedings of Spring Joint Computer Conference*, May 18-20, 1971, 379-385.

[22] Kumar, N., Sappal, A.S. (2011). Cordinate rotation digital computer algorithm: Design and architectures. *International Journal of Advanced Computer Science and Applications*, 2 (4), 68-71.

[23] Liao, W.-T., Lin, W.-Y., Cheng, W.-C., Lei, K.F., Lee, M.-Y. (2013). Precision enhancement and performance evaluation of a CORDIC-based tilting angle identification algorithm for three-axis accelerometers. In *International Symposium on Biometrics and Security Technologies (ISBAST)*, July 2-5, 2013. IEEE, 187-192.

[24] Bensidhoum, M.T., Laghrouche, M., Sidi Said, A., Montes, L., Boussey, J. (2014). Fabrication flaws and reliability in MEMS thin film polycrystalline flow sensor. *Microsystem Technologies*, 20 (1), 1-7.

[25] SparkFun Electronics. *Triple Axis Accelerometer Breakout - BMA180*. https://www.sparkfun.com/products/retired/9723.

[26] Altera Corporation. (2006). *DE2 User Manual*. ftp://ftp.altera.com/up/pub/Webdocs/DE2_UserManual.pdf

Received December 22, 2016.
Accepted March 9, 2017.

MEASUREMENT SCIENCE REVIEW, **16**, (2017), No. 2, 68-75

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

ISSN 1335-8871

Simulation and Optimization of Throttle Flowmeter with Inner-Outer Tube Element

Jiang Meng[1], Zhipeng Liu[1], Kun An[2], Meini Yuan[3]

[1]*School of Mechanical & Power Engineering, North University of China, Taiyuan, 030051, China*
[2]*School of Computer Science & Control Engineering, North University of China, Taiyuan, 030051, China*
[3]*School of Mechatronic Engineering, North University of China, Taiyuan, 030051, China*
Corresponding author: Jiang Meng, Email: johnmeng@nuc.edu.cn

In order to solve the dilemma between the smaller pressure loss and the larger flow measurement signal in traditional throttle flowmeters, a throttle structure with the inner-outer tube was designed and analyzed. The mathematical relationship model deduced from hydrodynamics showed there were three major parameters to determine the designed throttle structure. Furthermore, the optimal results were achieved by combining orthogonal test design and computational fluid dynamics by taking the ratio of differential pressure of inner-outer tube divided by that of anterior-posterior tube as the optimization goal. Finally, the simulation results with the best level parameters showed that the differential pressure of the anterior-posterior throttle could remain not only the smaller value among other parameters with the same structure of inner-outer tube. On the other hand, it was about one order magnitude less than differential pressure of V-cone flowmeter in the similar installation conditions with the flow velocity varying from 0.5 to 3.0 m/s. The designed inner-outer tube flowmeter can not only save manufacture costs, but also avoid the large sensitivity of pressure sensors, which may lead to a broader application in chemical and petrochemical enterprises.

Keywords: Throttle flowmeter, inner-outer tube, pressure loss, range analysis, computational fluid dynamics.

1. INTRODUCTION

As a measuring tool in modernized production coexisting with energy crisis, more and more professions depend on the flow measurement, such as mechanical ventilation [1], multi-phase analysis [2], and viscous fluid measurement [3], etc. With the development of economy and industry, more and more requirements are put forward to measure flows with high precision and great efficiency [4]-[5] based on the measuring laws, the measured fluids and the installation conditions.

The differential pressure flowmeter (DPF) [6]-[7] is one of the most widely used flow instruments rooted in the throttle effect, where the throttle differential pressure is proportional to the volume flow square. Both standard orifices and nozzles are popular types of throttle flowmeters, which were applied in industrial production because of easy production, low cost and no calibration under standard conditions. However, the traditional DPF has a smaller range ratio, bigger loss of pressure and higher requirement of field installation, and these shortcomings restrict seriously the broad application of throttle flowmeters. Therefore, researchers have designed and developed many new types of flowmeters, such as spindle, Venturi tube [8]-[9], and V-cone flowmeters [10].

The pressure loss is quite an important factor to evaluate performance of the throttle flowmeter [11]-[12]. On one hand, it is related to the energy consumption where the smaller pressure loss means less energy to be transported than normal to the equal distance. On the other hand, the differential pressure of the throttle element is used to measure the velocity or volume flow, where the bigger value is required to improve the flowmeter sensitivity. There occurs a dilemma between the fluid transportation and the flow measurement for traditional throttle flowmeters.

To solve this problem, a modified flowmeter called the inner-outer tube throttle flowmeter, was designed and simulated in this paper. Different from the V-cone flowmeter, two pressure pores were located in the straight section to measure the differential pressure signal between the inner-outer tube, rather than the anterior-posterior throttle, which can not only bring less throttle effects on the pressure drop, but also provide more stability of the pressure signals and better economy with the designed structure of the inner-outer tube. Therefore, the modified flowmeter is worth studying further both theoretically and in simulations.

2. THEORETICAL FOUNDATION

Considering the complexity of the flow, three hypothesis conditions are listed as follows:

DOI: 10.1515/msr-2017-0009

1). The flow is supposed to satisfy the laminar cases;
2). Any flow line does not change from Area 1-1 to 2-2;
3). The thickness of pipeline can be regarded as 0.

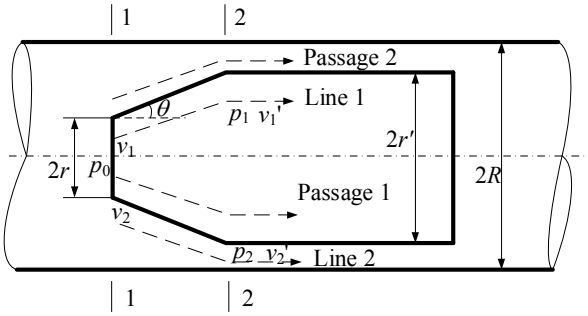


Fig.1. Structure of modified inner-outer tube throttle flowmeter.

According to the Bernoulli principle, Lines 1 and 2 (Fig.1.) satisfy the following equation,

$$\frac{p_0}{\rho}+\frac{v_1^2}{2}=\frac{p_1}{\rho}+\frac{v'^2_1}{2}+w_{f1} \tag{1}$$

$$\frac{p_0}{\rho}+\frac{v_2^2}{2}=\frac{p_2}{\rho}+\frac{v'^2_2}{2}+w_{f2} \tag{2}$$

where ρ is the fluid density, kg/m^3; p_0 is the absolute pressure of 1-1 section, Pa; p_1 and p_2 are the inner and outer pressure in 2-2 section, Pa; v_k and v_k' are the inlet and outlet velocity of Line k (k = 1, 2), m/s; w_{f1} and w_{f2} are the viscosity losses of Line 1 and 2, J / kg.

Equations (1) and (2) can be changed into the average velocity expressions, respectively:

$$\frac{p_0}{\rho}+\frac{\overline{v}_1^2}{2}=\frac{p_1}{\rho}+\frac{\overline{v}'^2_1}{2}+\overline{w}_{f1} \tag{3}$$

$$\frac{p_0}{\rho}+\frac{\overline{v}_2^2}{2}=\frac{p_2}{\rho}+\frac{\overline{v}'^2_2}{2}+\overline{w}_{f2} \tag{4}$$

where $\overline{v}_k$ and $\overline{v}'_k$ is the inlet/outlet average velocity of Line k (k = 1, 2), m/s; $\overline{w}_{f1}$ and $\overline{w}_{f2}$ are correspondingly the average viscosity losses of Lines 1 and 2, J/kg.

Therefore, the differential pressure of the inner point p_1 and the outer point p_2 can be calculated:

$$\begin{aligned}\Delta p &= p_1-p_2\\ &=\frac{\rho}{2}[(\overline{v}_1^2-\overline{v}'^2_1)-(\overline{v}_2^2-\overline{v}'^2_2)]\\ &\quad-(\overline{w}_{f1}-\overline{w}_{f2})\\ &=\frac{\rho}{2}[(\overline{v}_1^2-\overline{v}'^2_1)-(\overline{v}_2^2-\overline{v}'^2_2)]\\ &\quad-\rho g[(h_{\lambda1}-h_{\lambda2})-(h_{\zeta1}-h_{\zeta2})]\end{aligned} \tag{5}$$

where $h_{\lambda1}$ is the route loss of Line 1, J/kg, which can be calculated by the route loss equation of straight pipe when the flow passage is not very long,

$$h_{\lambda1}=\frac{8\mu l}{\pi\rho g}\cdot\frac{q_{v1}}{r^4} \tag{6}$$

and $h_{\lambda2}$ is the route loss of Line 2, J/kg, whose equation is based on the route loss equation of different diameter pipe,

$$h_{\lambda2}=\frac{8\mu l}{\pi\rho g}\cdot\frac{q_{v2}}{R^4-r^4-\dfrac{R^2-r^2}{\ln(R/r)}} \tag{7}$$

$h_{\zeta1}$, the local loss of Line 1, which can be obtained based on the diffuser tube equation,

$$h_{\zeta1}=\varphi\frac{(\overline{v}_1-\overline{v}_1')^2}{2g} \tag{8}$$

and $h_{\zeta2}$ is the local loss of Line 2, which can be calculated according to the nozzle tube theory,

$$h_{\zeta2}=\zeta\frac{(\overline{v}_2-\overline{v}_2')^2}{2g} \tag{9}$$

where l is the length of flow passage, m; μ is the dynamic viscosity, Pa · s, q_{v1} and q_{v2} are the volume flow of Passages 1 and 2, m^3/s; r is the hydraulic radius of Passage 1, m; R is the pipe radius, m; g is the gravitational acceleration, 9.8 m/s^2 normally; φ and ζ are the different local loss coefficients, meeting the following equations,

$$\varphi=\frac{\lambda}{8\sin(\theta/2)}[1-(\frac{S_1}{S_1'})^2]+\kappa(1-\frac{S_1}{S_1'}) \tag{10}$$

$$\zeta=\frac{\lambda}{8\sin(\theta/2)}[1-(\frac{S_2}{S_2'})^2] \tag{11}$$

where θ is the cone angle; λ is the route loss coefficient of the diffuse (or nozzle) tube; κ is the cone angle coefficient, related to θ; S_1 and S_1' are the area of 1-1 and 2-2 in Passage 1; S_2 and S_2' are the area of 1-1 and 2-2 in Passage 2, seen from Fig.1.

Considering the flow passage is relatively short (< 150 mm), the values of both $h_{\lambda1}$ and $h_{\lambda2}$ are thus small and far less than the corresponding local losses, $h_{\zeta1}$ and $h_{\zeta2}$, based on (6) and (7). Therefore, the route losses can be neglected, that is, $h_{\lambda1}-h_{\lambda2}=0$.

According to the fluid dynamics [13], any velocity v^* on the cross section satisfies (12) when the fluid moves in the round pipe with the radius of R,

$$v^*=2v_0(1-\frac{r^{*2}}{R^2}) \tag{12}$$

where v_0 is the average velocity through the cross section of pipe, $v_0 = q_v / (\pi R^2)$, among which q_v is the flux of the cross section with the area of πR^2; r^* is the radius corresponding to the velocity v^*.

Firstly, the flux q_{v1} of the inlet section S_1 in Passage 1 can be solved based on (12),

$$q_{v1} = \oint_{S_1} 2v_0(1-\frac{r^2}{R^2})dS = q_v \frac{2r^2}{R^2}(1-\frac{r^2}{2R^2}) \tag{13}$$

Secondly, according to the law of mass conservation, the flux q_{v2} of the inlet section S_2 is,

$$q_{v2} = q_v - q_{v1} = q_v[1-\frac{2r^2}{R^2}(1-\frac{r^2}{2R^2})] = q_v(1-\frac{r^2}{R^2})^2 \tag{14}$$

And then the average inlet velocities of $S_1 = \pi r^2$ and $S_2 = \pi(R^2-r^2)$ are expressed respectively as,

$$\overline{v}_1 = \frac{q_{v1}}{S_1} = \frac{2q_v}{\pi R^2}(1-\frac{r^2}{2R^2}) = \frac{q_v}{\pi R^2}(2-\frac{r^2}{R^2}) \tag{15}$$

$$\overline{v}_2 = \frac{q_{v2}}{S_2} = \frac{q_v(1-\frac{r^2}{R^2})^2}{\pi(R^2-r^2)} = \frac{q_v}{\pi R^2}(1-\frac{r^2}{R^2}) \tag{16}$$

Considering the continuous equation of Passage 1 and 2,

$$q_{v1} = \overline{v}_1 S_1 = \overline{v}_1' S_1' \tag{17}$$

$$q_{v2} = \overline{v}_2 S_2 = \overline{v}_2' S_2' \tag{18}$$

the average outlet velocities of $S_1' = \pi r'^2$ and $S_2' = \pi(R^2 - r'^2)$ can be transformed from the above expressions,

$$\overline{v}_1' = \frac{q_{v1}}{S_1'} = \frac{q_v}{\pi R^2}(2-\frac{r^2}{R^2})\frac{r^2}{r'^2} \tag{19}$$

$$\overline{v}_2' = \frac{q_{v2}}{S_2'} = \frac{q_v}{\pi R^2}(1-\frac{r^2}{R^2})\frac{R^2-r^2}{R^2-r'^2} \tag{20}$$

Finally, the following expressions can be deduced for use in the next paragraph,

$$\overline{v}_1^2 - \overline{v}_1'^2 = \frac{q_v^2}{\pi^2 R^4}(2-\frac{r^2}{R^2})^2(1-\frac{r^4}{r'^4}) \tag{21}$$

$$\overline{v}_2^2 - \overline{v}_2'^2 = \frac{q_v^2}{\pi^2 R^4}(1-\frac{r^2}{R^2})^2(1-\frac{r^4}{r'^4}) \tag{22}$$

$$(\overline{v}_1 - \overline{v}_1')^2 = \frac{q_v^2}{\pi^2 R^4}(2-\frac{r^2}{R^2})^2[1-(\frac{R^2-r^2}{R^2-r'^2})^2] \tag{23}$$

$$(\overline{v}_2 - \overline{v}_2')^2 = \frac{q_v^2}{\pi^2 R^4}(1-\frac{r^2}{R^2})^2(\frac{r^2-r'^2}{R^2-r'^2})^2 \tag{24}$$

And Δp can be calculated according to (5) and (21)~(24),

$$\frac{\Delta p}{\rho/2} = \frac{q_v^2}{\pi^2 R^4}(2-\frac{r^2}{R^2})^2(1-\frac{r^4}{r'^4}) - \frac{q_v^2}{\pi^2 R^4}(1-\frac{r^2}{R^2})^2(1-(\frac{R^2-r^2}{R^2-r'^2})^2 + \frac{\varphi q_v^2}{\pi^2 R^4}(2-\frac{r^2}{R^2})^2(1-\frac{r^2}{r'^2})^2 - \frac{\zeta q_v^2}{\pi^2 R^4}(1-\frac{r^2}{R^2})^2(\frac{r^2-r'^2}{R^2-r'^2})^2$$

If $\eta = \sqrt{1-r^2/R^2}$ is the radius ratio of inner-outer tube, $\delta = r/r'$ is denoted as the radius ratio of the inner flow passage, and $\delta' = \sqrt{R^2-r^2}/\sqrt{R^2-r'^2}$ is regarded as the equivalent radius ratio of the outer flow passage, Δp can be simplified as:

$$\Delta p = \frac{\rho}{2}\frac{q_v^2}{\pi^2 R^4}(a-b) \tag{25}$$

where $a = (1+\eta^2)^2[1-\delta^4+\varphi(1-\delta^2)^2]$;

$b = \eta^4[1-\delta'^4+\zeta(1-\delta'^2)^2]$;

$\varphi = \lambda[1-(1-\eta^{-2})^2-\kappa(2-\eta^{-2})]/[8\sin(\theta/2)]$;

$\zeta = \lambda[1-(1-\eta^{-2})^2]/[8\sin(\theta/2)]$.

And the volume flow of q_v can be also simplified as:

$$q_v = \frac{\pi R^2}{\sqrt{a-b}}\sqrt{\frac{2\Delta p}{\rho}} = C \cdot S\sqrt{\frac{2\Delta p}{\rho}} \tag{26}$$

where S is the area of round pipe, $S = \pi R^2$; C is the discharge coefficient of the inner-outer tube,

$$C = \frac{1}{\sqrt{a-b}}$$

Based on (25), we can express C as the calibration expression,

$$C = \frac{q_v}{S\sqrt{2\Delta p/\rho}} = \frac{\overline{v}}{\sqrt{2\Delta p/\rho}} \tag{27}$$

where $\overline{v}$ is the average flow velocity of the round pipe, $\overline{v} = q_v/S$.

Seen from (25), the relation between the volume flow and differential pressure depends mainly on three factors, namely, the big radius r', the small radius r, and the cone angle θ, and can be independent of the fluid pressure. On one hand, these factors can be regarded as the structural parameters to design the inner-outer tube throttle flowmeter. On the other hand, they have significant influence on the differential pressure of inner-outer tube in complex flow states, although the expressions above are derived from the condition of laminar flow.

3. SIMULATION AND ANALYSIS OF FLOWMETER

A. Orthogonal test

To find the best performance of the inner-outer tube throttle flowmeter, those 3 factors should be considered to design the orthogonal test. The test times will be quite long when a comprehensive test is carried out. Considering 3 factors at 3 levels full test, we need as many as $3^3 = 27$ test times, which will increase test difficulty so much that the comprehensive test cannot be conducted effectively. Therefore, the orthogonal test method can be used to find the best combinational plan effectively when using partial and representative points to replace full test points. Taking 3 factors-levels test as an example, an orthogonal test needs only 9-group tests instead of 27-group tests, which will greatly reduce the test work.

Based on the orthogonal test design theory, 3 factors-levels test plan with $L_9(3^4)$ orthogonal table was carried out to design the structure parameters of the inner-outer tube, such as the small radius r, the big radius r', and the cone angle (called also the diffusion angle) θ.

The type of tested fluid pipe is U-PVC 32*2.4. That is, the outer radius of pipe is 16 mm and the inner radius is 13.6 mm. According to the pipeline conditions, the big radius (r') of inner-outer tube can change from 6 to 12 mm, and the cone angle (θ) varies from 6° to 8°. If taking directly 3 parameters as 3 factors to prepare the orthogonal test design, there may be some invalid plans where r may be equal to or even greater than r'. In order to avoid these unrealizable cases, the radius ratios, $\varepsilon = r / r' < 1$, can be defined to replace the small radius as the new factor for orthogonal test design. Table 1. shows these factors (named A, B and C) and their values of corresponding levels (1, 2 and 3), and hence produces the $L_9(3^4)$ orthogonal table (Table 2.).

Table 1. Values of 3 factors at 3 levels.

Factor		**A, ε**	**B, r' (mm)**	**C, θ (°)**
Level	1	0.5	6	6
	2	0.65	9	7
	3	0.8	12	8

Table 2. Orthogonal test design of $L_9(3^4)$.

Plan No.	**A, ε**	**B, r' (mm)**	**C, θ (°)**
1	1 (0.5)	1 (6 mm)	1 (6 °)
2	1	2 (9 mm)	2 (7 °)
3	1	3 (12 mm)	3 (8 °)
4	2 (0.65)	1	2
5	2	2	3
6	2	3	1
7	3 (0.8)	1	3
8	3	2	1
9	3	3	2

B. Fluent simulation

To calculate the differential pressure with the flowmeter parameters in Table 2., the computational fluid dynamics (CFD) method was adopted.

Gambit was used to establish 9 groups of 3-D models based on Table 2.; the unstructured tetrahedral grids were meshed (Fig.2.) to adapt to the complex model of inner-outer tube; and finally the meshing model was imported to the Fluent software.

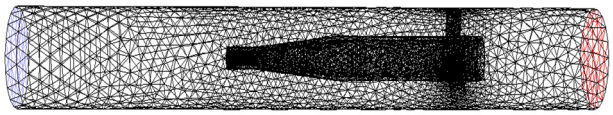


Fig.2. Meshing the inner-outer tube of Plan 1 ($\varepsilon = 0.5$, $r' = 6$ *mm*, $\theta = 6°$) with 109133 unstructured tetrahedral grids.

The pressure-based solver, absolute velocity equation and steady state flow were set as the basic setting of Fluent. The RNG-based k-ε was treated as the turbulent model, because it is more accurate and reliable for a wider class of flows than the standard k-ε [14]. The water was used as the testing fluid in room temperature with the common velocity of 0.5 ~ 3.0 m/s moving in the test pipe, whose minimal Reynolds number ($\mathrm{Re_{min}}$) was greater than the critical Reynolds number of round pipe ($\mathrm{Re_c} = 2000 \sim 2600$),

$$\mathrm{Re_{min}} = \frac{v_{\min} d}{\nu} = \frac{0.5 \times 0.0272}{1.006 \times 10^{-6}} = 13159 > \mathrm{Re_c}$$

where d is the internal diameter of the test pipe, $d = 27.2$ mm; ν is the kinematic viscosity of water, $\nu = 1.006 \times 10^{-6}$ m^2/s; $v_{\min}$ is the minimum average inlet velocity, $v_{\min} = 0.5$ m/s.

For the incompressible fluid, the velocity inlet and outflow were set to the inlet and outlet of the boundary conditions, and the wall was set to both the throttle and pipe.

C. Simulation results

Fig.3. shows the differential pressure of the inner-outer tube and the anterior-posterior throttle, denoted as Δp_{io} and Δp_{ap}, respectively, which were calculated under 6 velocities from 0.5 to 3.0 m/s. According to Fig.3., both Δp_{io} and Δp_{ap} increased with the increase of flow velocity. They were of a similar variation trend, and reached the relatively larger values when $r' = 12$ mm (corresponding to Plans 3, 6 and 9) and the smaller values when $r' = 6$ mm (equivalent to Plans 1, 4 and 7).

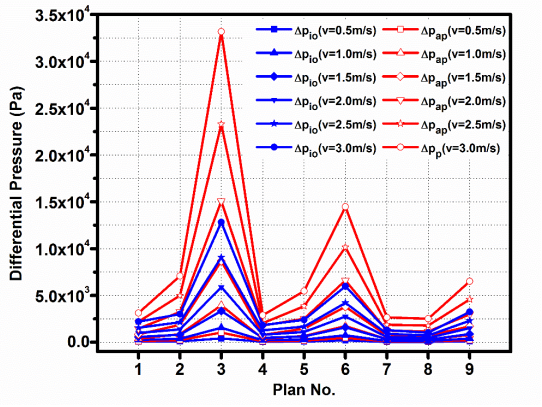


Fig.3. Differential pressure curves (Δp_{io} and Δp_{ap}) of 9 plans under different flow velocities from 0.5 to 3.0 m/s.

D. Analysis of variance

If the flowmeter pursued only the biggest differential pressure of the inner-outer tube (Δp_{io}), differential pressure of the anterior-posterior throttle (Δp_{ap}) would increase to the maximum value at the same time, which could cause a large increase of power cost. In order to achieve reasonable and economic parameters, both differential pressures should be considered simultaneously, and thus a dimensionless factor named the differential pressure ratio γ was defined as,

$$\gamma = \Delta p_{io} / \Delta p_{ap} \tag{28}$$

Based on the defined factor and simulation results, the analysis of variance (ANOVA) was performed firstly. Table 3 provides the F value of different factors, such as A, B and C (detailed meaning shown in Table 1.). Table 3. and Table 4. show three factors (A, B and C) had no significant differences under flow velocities of 3.0 m/s. The strange inference came from the non-merged factors, which caused the mean square less than that of the error (Table 4.).

Table 3. Result of ANOVA of differential pressure ratios under different flow velocities of 3.0 m/s.

Plan No.	**A**	**B**	**C**	**Ratio (γ)**
1	1	1	1	0.707
2	1	2	2	0.427
3	1	3	3	0.387
4	2	1	2	0.631
5	2	2	3	0.44
6	2	3	1	0.413
7	3	1	3	0.491
8	3	2	1	0.41
9	3	3	2	0.499
T_1	1.521	1.829	1.530	ΣT =4.4050
T_2	1.484	1.277	1.557	$\Sigma\gamma^2$=2.2525
T_3	1.400	1.299	1.318	S_T = 0.0965
SS	0.002563	0.065121	0.011422	

Table 4. Details of ANOVA under velocities of 3.0 m/s.

Factor	**Degree of Freedom**	**Stdev Square**	**Mean Square**	***F***
A	2	*0.002563*	*0.001281*	0.1477
B	2	0.065121	0.032560	3.7532
C	2	*0.011422*	*0.005711*	0.6583
Error	2	0.017351	0.008675	

Therefore, Factors A and C can be merged and appended to the error, because they seem to have no effect on the differential pressure ratio. Table 5. is the result of re-analysis of variance, where Factor B or the big diameter could be regarded as the significant factor. That is to say, the smaller the big diameter r' is, the bigger ratio of differential pressure can be achieved. This conclusion could be verified from the following Fig.4., where Plans 1, 4 and 7 achieved the bigger ratio because there was the smallest big diameter (r' = 6 mm) in the plans above.

Table 5. Details of re-ANOVA under velocities of 3.0 m/s.

Factor	**Degree of Freedom**	**Stdev Square**	**Mean Square**	***F***
B	2	0.065121	0.032560	6.2346
Error*	6	0.031335	0.005223	
Sum	8	0.096456	$F_{1\text{-}0.05}(2, 6)$ = 5.14	

E. Range analysis

From the intuitive view, the level $A_1B_1C_1$ of Plan 1 seemed to reach the largest value of differential pressure ratio among 9 plans in Fig.4.

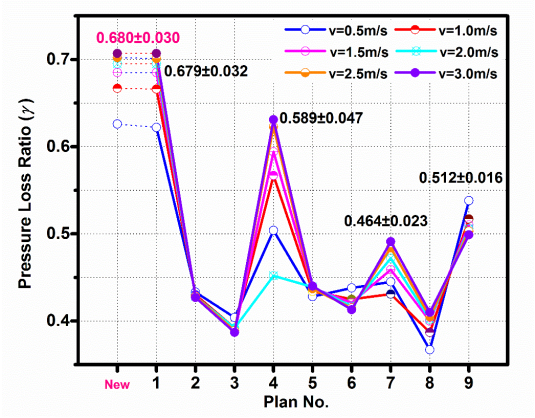


Fig.4. Pressure loss ratios of 9 plans under different flow velocities from 0.5 to 3.0 m/s.

However, the deduced new level of $A_1B_1C_2$ might reach a larger value than Plan 1, after performing the range analysis (Table 6.) based on simulation results.

Table 6. Range analysis of differential pressure ratio under different flow velocities from 0.5 to 3.0 m/s.

Velocity (m/s)	**Range Analysis**	**Factors**		
		A	**B**	**C**
0.5	Average k_1	0.487	0.524	0.476
	Average k_2	0.457	0.409	0.492
	Average k_3	0.450	0.460	0.426
	Range R_g	0.036	0.114	0.066
	Biggest level	*1*	*1*	*2*
1.0 ~ 2.5	…	…	…	…
	Biggest level	*1*	*1*	*2*
3.0	Average k_1	0.507	0.610	0.510
	Average k_2	0.495	0.425	0.519
	Average k_3	0.467	0.433	0.439
	Range R_g	0.040	0.184	0.080
	Biggest level	*1*	*1*	*2*

Considering the test plans did not contain the new level of $A_1B_1C_2$, the corresponding simulation was performed more with θ = 7°, r' = 6 mm, ε = 0.5 under 6 velocities and the results are displayed in Fig.4. and Table 7. The final result showed that the average differential pressure ratio ($\bar{\gamma} = 0.680 \pm 0.030$) in the new level of $A_1B_1C_2$ was slightly superior to Plan 1, which validated that range analysis could find the optimal combination parameters with the help of carefully designed orthogonal test.

Table 7. Simulation results of best level $A_1B_1C_2$ under different flow velocities from 0.5 to 3.0 m/s.

Velocity (m/s)	**$A_1B_1C_2$: r = 3 mm, r' = 6 mm, θ = 7°**		
	Δp_{ap} (Pa)	**Δp_{io} (Pa)**	**γ**
0.5	114.3	71.5	0.626
1.0	395.0	263.3	0.667
1.5	841.3	576.4	0.685
2.0	1447.0	1005.8	0.695
2.5	2203.9	1547.4	0.702
3.0	3115.7	2203.5	0.707
Average	*1352.9*	*944.7*	*0.680 ± 0.030*

4. DISCUSSION

A. Discharge coefficient

Aimed at the best level of $A_1B_1C_2$, the discharge coefficient could be fitted as the slope of a line connecting the root of differential pressure $\sqrt{\Delta p}$ with the flow velocity v based on (27) in Fig.5.

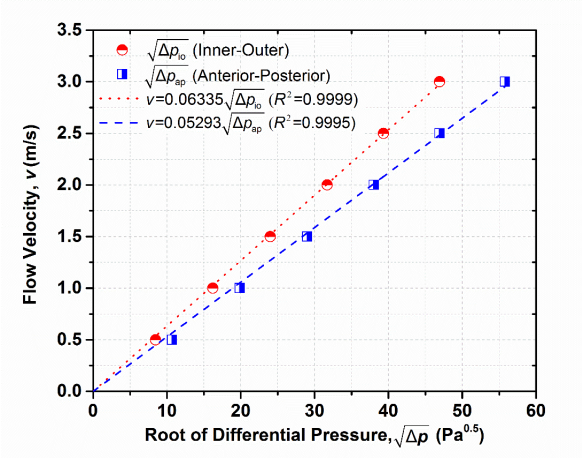


Fig.5. Discharge coefficient fitting based on the root of differential pressure and volume flow.

As the highest ratio of differential pressure, the optimal level of $A_1B_1C_2$ was used to estimate the discharge coefficient, shown in Fig.5. By fitting linearly between Q_v and two kinds of root of differential pressure ($\sqrt{p_{ap}}$ and $\sqrt{p_{io}}$), respectively, two slopes were in the same order of magnitude, k_{ap} = 0.05293 for the anterior-posterior differential pressure and k_{io} = 0.06335 for the inner-outer one. According to (27), the discharge coefficients could be calculated as

$$C_{ap} = \frac{v}{\sqrt{2\Delta p_{ap}/\rho}} = \frac{k_{ap}}{\sqrt{2\rho}} = \frac{0.05293}{\sqrt{2\times 998.2}} = 1.1846\times 10^{-3}$$

and $$C_{io} = \frac{k_{io}}{\sqrt{2\rho}} = \frac{0.06335}{\sqrt{2\times 998.2}} = 1.4178\times 10^{-3},$$

where the density of water at room temperature (20°C) is ρ = 998.2 kg/m^3.

The calculation results meant two discharge coefficients (C_{ap} and C_{io}) were in the same order of magnitude as well, but C_{io} was larger than C_{ap}, which showed the sensitivity of the inner-outer tube as the measuring unit was better than that of the anterior-posterior throttle. Meanwhile, the fitting determination coefficient (R^2 = 0.9999) of the former was slightly better than that of the latter (R^2 = 0.9995). That is, the inner-outer tube is more competitive and more convenient in some special cases, especially in the limited space, because Δp_{io} can be achieved only in the same section.

B. Differential pressure comparison between anterior-posterior throttle and inner-outer tube

The Δp_{ap} of the anterior-posterior throttle should be as low as possible in virtue of the tremendous impact on the cost of the fluid transmission, while the Δp_{io} of the inner-outer tube may be taken as a higher value because it can reduce the difficulty to measure the signal of differential pressure and decrease the investment cost. Therefore, Δp_{io} and Δp_{ap} can be maximized together to obtain the bigger Δp_{io} and smaller Δp_{ap} simultaneously. Furthermore, a dimensionless factor f is defined in (28) to avoid different magnitude of differential pressures and reduce the effect of the different fluid velocities,

$$f = \frac{\Delta p_{io} - \Delta p_{ap}}{\Delta p_{ap}} = \gamma - 1$$

Without considering the constant item, f could be transformed to the differential pressure ratio γ expressed in (28). As a result, the deduced level of $A_1B_1C_2$ could provide the best choice for the inner-outer tube flowmeter, corresponding to the largest γ.

The presented flowmeter adopted the inner-outer tube to obtain another differential pressure Δp_{io} related to the flow velocity, other than the traditional differential pressure Δp_{ap}. On one hand, the Δp_{ap} range of $A_1B_1C_2$ was 110~3200 Pa, obviously one order magnitude less than that of $A_1B_3C_3$ (Plan 3 in Fig.3.), and hence the power charge of the optimal level could be reduced significantly. On the other hand, the Δp_{io} of the optimal level is more than twice that of Plan 8 (close to the smallest level). Thus, the measured signal Δp_{io} can avoid high sensitivity of pressure sensor and decrease the manufacturing expenses of the inner-outer tube flowmeter because of the relatively large value of Δp_{io}.

As is known, V-cone flowmeter has been one of the most popular throttle flowmeters in recent years, because of its good performance and low pressure loss. To compare with the inner-outer tube flowmeter presented in this paper, the V-cone flowmeter was set to the parameters below based on the popular researches [15]-[16]. The length of V-cone was 200 mm, and the equivalent diameter ratio was $\beta = 0.6$, and the diffusion angles were 60° (anterior) and 150° (posterior), respectively.

Using the same boundary conditions as those of the inner-outer tube flowmeter, the differential pressure of the V-cone throttle in Fig.6. was compared with that of the inner-outer tube and anterior-posterior throttle under the same flow velocity. The simulation results showed that the sensitivity of V-cone flowmeter was quite lower than that of the inner-outer tube flowmeter (including Δp_{ap} and Δp_{io} as measuring unit). According to the data, the differential pressure of V-cone ($\Delta p_{V\text{-cone}}$) was about 16 times of Δp_{io} under flow velocities varying from 0.5 to 3.0 m/s, and hence the slope of the output/input line for the V-cone flowmeter was only a quarter of that for the presented flowmeter. It is no doubt that the inner-outer tube flowmeter can both reduce more operating cost than V-cone, and increase the sensitivity to a great extent as the important factor for the flowmeter to use more widely.

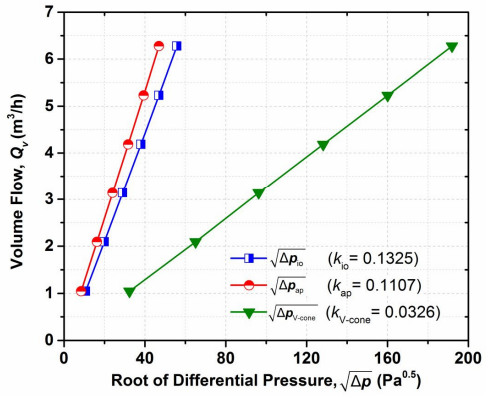


Fig.6. Comparison of anterior-posterior differential pressure between V-cone flowmeter and inner-outer tube flowmeter.

5. CONCLUSION

Firstly, a throttle element of the inner-outer tube was designed and presented to solve the dilemma between the smaller differential pressure (or pressure loss) and the larger flow measurement signal in traditional throttle flowmeters. Secondly, the mathematical model was deduced theoretically based on the hydrodynamics, where there were three major parameters (big radius r', small radius r, and cone angle θ) to determine the structure of inner-outer tube. Thirdly, using both $L_9(3^4)$ table of orthogonal test design and CFD of Fluent software, the optimal results were obtained by taking a defined ratio of the inner-outer differential pressure Δp_{io} divided by the anterior-posterior one Δp_{ap} as the optimization goal. Finally, the optimal simulation results ($r' = 6$ mm, $r = 3$ mm and $\theta = 7°$) showed that the pressure loss of the anterior-posterior throttle could not only remain a smaller value among other parameters with the same structure, but was about one order magnitude less than that of V-cone flowmeter in the similar installation conditions, when the flow velocity varied from 0.5 to 3.0 m/s. The lower pressure loss of Δp_{ap} and relatively higher Δp_{io} as the measured signal can save the manufacture costs and avoid the large sensitivity of pressure sensors, which may lead to a broader application in chemical and petrochemical enterprises.

ACKNOWLEDGMENT

The authors would like to acknowledge the support from the National Natural Science Foundation of China under Grant no. 61401406 and no. 50871086.

REFERENCES

[1] Schena, E., Massaroni, C., Saccomandi, P., Cecchini, S. (2015). Flow measurement in mechanical ventilation: A review. *Medical Engineering & Physics*, 37 (3), 257-264.

[2] Thorn, R., Johansen, G.A., Hjertaker, B.T. (2013). Three-phase flow measurement in the petroleum industry. *Measurement Science and Technology*, 24 (1), 12003-12019.

[3] Jiang, W., Zhang T., Xu, Y., et al. (2016). The effects of fluid viscosity on the orifice rotameter. *Measurement Science Review*, 16 (2), 87-95.

[4] Fincham, A.M., Spedding, G.R. (1997). Low cost, high resolution DPIV for measurement of turbulent fluid flow. *Experiments in Fluids*, 23 (6), 449-462.

[5] Santi, P.M. (2014). Precision and accuracy in debris-flow volume measurement. *Environmental & Engineering Geoscience*, 20 (4), 349-359.

[6] Daev, Z.A. (2015). A comparative analysis of the discharge coefficients of variable pressure-drop flowmeters. *Measurement Techniques*, 58 (3), 323-326.

[7] Nukui, K., Komiya, K., Kagawa, T. (2009). Effect of temperature sensor position on differential pressure measurement in orifice type flowmeter. *Journal of Clinical Psychopharmacology*, 38 (3), 233-238.

[8] Xu, L., Zhou, W., Li, X., et al. (2011). Wet-gas flow modeling for the straight section of throat-extended Venturi meter. *IEEE Transactions on Instrumentation and Measurement*, 60 (6), 2080-2087.

[9] Nithin, T., Jain, N., Hiriyannaiah, A. (2012). Optimization of Venturi flow meter model for the angle of divergence with minimal pressure drop by computational fluid dynamics method. In *International Conference on Challenges and Opportunities in Mechanical Engineering, Industrial Engineering and Management Studies (ICCOMIM 2012)*, July 11-13, 2012, 658-670.

[10] Wei, C., Tan, Ch., Dong, F. (2011). Numerical simulation and optimal design for cone flowmeter. *Journal of Engineering Thermophysics*, 32 (7), 1165-1168.

[11] Daev, Zh.A. (2015). A comparative analysis of the discharge coefficients of variable pressure-drop flowmeters. *Measurement Techniques*, 58 (3), 323-326.

[12] Guo, S., Sun, L., Zhang, T., Yang, W., Yang, Z. (2013). Analysis of viscosity effect on turbine flowmeter performance based on experiments and CFD simulations. *Flow Measurement & Instrumentation*, 34 (5), 42-52.

[13] Sleigh, A. (2009). *Real fluids - Pressure loss during laminar flow in a pipe*. University of Leeds, UK, http://www.efm.leeds.ac.uk/CIVE/CIVE1400/Section4/laminar_turbulent.htm.

[14] Mihailovic, D.T., Gualtieri, C. (Editors) (2010). *Advances in Environmental Fluid Mechanics*. World Scientific Publishing Company, ISBN 978-981-4465-97-7.

[15] Thorel, L., Gaudin, C., Rault, G., Garnier, J., Favraud, C. (2007). A cone pressuremeter for soil characterisation in the centrifuge. *International Journal of Physical Modelling in Geotechnics*, 7 (1), 25-32.

[16] Xu, Y., Chen, W., Zhang, T., Liu, W. (2015). Study on applicability of turbulence model for V-cone flowmeter with double support. *Chinese Journal of Scientific Instrument*, 36 (2), 459-465.

Received November 11, 2016.
Accepted March 13, 2017.

MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 2, 76-82

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

ISSN 1335-8871

A Design of Finite Memory Residual Generation Filter for Sensor Fault Detection

Pyung Soo Kim

Department of Electronic Engineering, Korea Polytechnic University, Siheung-si, Gyeonggi-do, 15073, KOREA, pspeter.kim@gmail.com

In the current paper, a residual generation filter with finite memory structure is proposed for sensor fault detection. The proposed finite memory residual generation filter provides the residual by real-time filtering of fault vector using only the most recent finite measurements and inputs on the window. It is shown that the residual given by the proposed residual generation filter provides the exact fault for noise-free systems. The proposed residual generation filter is specified to the digital filter structure for the amenability to hardware implementation. Finally, to illustrate the capability of the proposed residual generation filter, extensive simulations are performed for the discretized DC motor system with two types of sensor faults, incipient soft bias-type fault and abrupt bias-type fault. In particular, according to diverse noise levels and windows lengths, meaningful simulation results are given for the abrupt bias-type fault.

Keywords: Fault diagnosis, Fault detection, Residual generation filter, Finite memory structure, Kalman filter.

1. INTRODUCTION

As most dynamic process plants become more complex, there has been a growing demand for fault diagnosis. Fault diagnosis is the prompt indication of incipient and abrupt faults, it can help avoid major plant breakdowns and permit appropriate actions that maintain the operation. Thus, fault diagnosis is an important and challenging problem for disciplines such as chemical engineering, nuclear engineering, aerospace engineering, and automotive systems [1]-[3].

Generally, model-based fault diagnosis algorithms consist of fault detection and fault isolation. Fault detection is used to detect malfunctions in real time, as soon and as certainly as possible. Fault isolation is used to determine the root cause by isolating the system component(s) whose operation mode is not nominal. The essential step for fault diagnosis is the generation of a set of variables known as residuals using one or more residual generation filters. That is, residuals are signals, often time-varying, that are used as fault detectors. The residuals used for fault detection should ideally be zero or zero mean) under no-fault conditions. In practical applications, the residuals are corrupted by the presence of noise, unknown disturbances, and system model uncertainties. Hence, the residual should be insensitive to noise, disturbances, and model uncertainties while maximally sensitive to faults to be useful in practical applications.

Kalman filters have been adopted as residual generation filters in stochastic cases where noise has to be considered [4]-[9]. Their compact representation and efficient manner mean that the Kalman filter has been successfully applied in various areas including fault detection. However, the Kalman filter has an infinite memory structure that utilizes all observations, as accomplished by equal weighting, and has a recursive formulation. Thus, the Kalman filter tends to accumulate filtering errors as time goes by and can even show divergence phenomenon for temporary modeling uncertainties and round-off errors [10]-[13]. This inherent property of the Kalman filter has been shown in sensor application areas [14], [15]. In addition, long past measurements are not useful for fault detection with unknown occurrence times. Moreover, it is also known that the increase of the number of measurements for detection decisions will increase the detection latency in a system that detects a signal with an unknown occurrence time.

Therefore, an alternative residual generation filter with finite memory structure is proposed in the current paper for sensor fault detection. The proposed finite memory residual generation filter provides the residual via the real-time filtering of fault vectors using only the most recent finite observations and inputs in the window. It is shown that the residual given by the proposed residual generation filter provides the exact fault required for noise-free systems. The proposed residual generation filter is specified to the digital filter structure for its amenability to hardware implementation.

Finally, extensive simulations are performed for a discretized DC motor system to illustrate the capability of the proposed residual generation filter. Two types of sensor

DOI: 10.1515/msr-2017-0010

fault are considered: incipient soft bias-type fault and abrupt bias-type fault. According to the diverse noise levels and windows lengths, simulation results can be shown to be meaningful for abrupt bias-type fault.

2. BASIC CONCEPT OF RESIDUAL GENERATION FILTER IN FAULT DIAGNOSIS

In general, the fault model can be represented by the following discrete-time state space model with unknown sensor faults as well as noises:

$$\begin{aligned} x(i+1) &= \Phi x(i) + Du(i) + \Delta_1 f(i) + E\omega(i), \\ y(i) &= Cx(i) + \Delta_2 f(i) + v(i), \end{aligned} \tag{1}$$

where $x(i) \in \Re^n$ the state vector, $u(i) \in \Re^l$ and $y(i) \in \Re^q$ are the input vector and the measurement vector. The covariances of the system noise $\omega(i) \in \Re^p$ and the measurement noise $v(i) \in \Re^q$ are Q_ω and R, respectively. The fault vector $f(i) \in \Re^q$ in the system under consideration are to be represented by random-walk processes as

$$f(i+1) = f(i) + \delta(i),$$

where $f(i) \equiv \begin{bmatrix} f_1(i) & f_2(i) & \Lambda & f_q(i) \end{bmatrix}^T$ and $\delta(i) \in \Re^q$ is a zero-mean white Gaussian random process with covariance Q_δ. It is noted that the random-walk process provides a general and useful tool for the analysis of unknown time-varying parameters and has been widely used in the detection and estimation area.

The fault model (1) can be rewritten as an augmented state model as

$$\begin{aligned} \begin{bmatrix} x(i+1) \\ f(i+1) \end{bmatrix} &= A \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} + Bu(i) + G \begin{bmatrix} \omega(i) \\ \delta(i) \end{bmatrix}, \\ y(i) &= H \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} + v(i), \end{aligned} \tag{2}$$

where

$$A = \begin{bmatrix} \Phi & \Delta_1 \\ 0 & I \end{bmatrix}, \; B = \begin{bmatrix} D \\ 0 \end{bmatrix},$$
$$G = \begin{bmatrix} E & 0 \\ 0 & I \end{bmatrix}, \; H = \begin{bmatrix} C & \Delta_2 \end{bmatrix},$$

and the system noise term $[\omega^T(i) \quad \delta^T(i)]^T \equiv w(i)$ is a zero-mean white Gaussian random process with covariance $Q = diag(Q_\omega \quad Q_\delta)$.

In general, model-based fault diagnosis algorithms consist of fault detection and fault isolation. Fault detection is used to detect malfunctions in real time, as soon and as certainly as possible. Fault isolation is used to determine the root cause by isolating the system component(s) whose operation mode is not nominal.

The essential step in fault diagnosis is to generate a set of variables known as residuals by using one or more residual generation filters. A residual is an often time-varying signal that is used as a fault detector. In general, the residual used for fault detection is defined by the signal generated based on the measurement vector $y(i)$ and input vector $u(i)$. This residual should ideally be zero (or have zero mean) under no-fault conditions. In practical applications, the residuals are corrupted by the presence of noise, unknown disturbances, and uncertainties in the system model. Hence, the residual should be insensitive to noise, disturbances, and model uncertainties while maximally sensitive to faults to be useful in practical applications.

The main contribution of the current paper is to design and analyze an alternative residual generation filter. That is, the current paper focuses on fault detection that declares quickly which sensor is faulty by estimating size and type of each sensor faults. Although fault isolation is not the current paper's topic, the residual generation filter designed in the following section can provide the means for the isolation of faults using at least three different approaches can be distinguished: fixed direction residuals, structured residuals, and structured hypothesis tests.

3. ALTERNATIVE RESIDUAL GENERATION FILTER FOR FAULT DETECTION

In the current paper, an alternative residual generation filter is designed with finite memory structure. The proposed finite memory residual generation filter provides the residual $r(i)$ by real-time filtering of fault vector $f(i)$ using only the most recent finite measurements $Y(i)$ and inputs $U(i)$ on the window $[i-M, i]$ as follows

$$r(i) \equiv \hat{f}(i) \equiv \mathcal{H}\left[Y(i) - \Omega U(i)\right], \tag{3}$$

where $\hat{f}(i)$ is the filtered estimate of $f(i)$ and $\mathcal{H}$ is the filter gain matrix. The term $Y(i) - \Omega U(i)$ in (3) with the most recent finite measurements $Y(i)$ and inputs $U(i)$ can be represented by the following regression form on the window $[i-M, i]$:

$$Y(i) - \Omega U(i) = \Gamma \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} + \Lambda W(i) + V(i), \tag{4}$$

where

$$Y(i) \equiv \begin{bmatrix} y^T(i-M) & y^T(i-M+1) & \Lambda & y^T(i-1) \end{bmatrix}^T, \tag{5}$$

and $U(i)$, $W(i)$, $V(i)$ have the same form as (5) for $u(i)$, $w(i)$, $v(i)$, respectively, and matrices Ω, Γ, Λ are as follows:

$$\Omega \equiv - \begin{bmatrix} HA^{-1}B & HA^{-2}B & \Lambda & HA^{-M+1}B & HA^{-M}B \\ 0 & HA^{-1}B & \Lambda & HA^{-M+2}B & HA^{-M+1}B \\ \mathrm{M} & \mathrm{M} & \Lambda & \mathrm{M} & \mathrm{M} \\ 0 & 0 & \Lambda & 0 & HA^{-1}B \end{bmatrix},$$

$$\Gamma \equiv \begin{bmatrix} HA^{-M} \\ HA^{-M+1} \\ \mathrm{M} \\ HA^{-1} \end{bmatrix},$$

$$\Lambda \equiv - \begin{bmatrix} HA^{-1}G & HA^{-2}G & \Lambda & HA^{-M+1}G & HA^{-M}G \\ 0 & HA^{-1}G & \Lambda & HA^{-M+2}G & HA^{-M+1}G \\ \mathrm{M} & \mathrm{M} & \Lambda & \mathrm{M} & \mathrm{M} \\ 0 & 0 & \Lambda & 0 & HA^{-1}G \end{bmatrix}.$$

The noise term $\Lambda W(i)+V(i)$ in (4) is zero-mean white Gaussian with covariance Π given by

$$\Pi \equiv \Lambda \left[diag \begin{pmatrix} Q & Q & \Lambda & Q \end{pmatrix} \right] \Lambda^T + \left[diag \begin{pmatrix} R & R & \Lambda & R \end{pmatrix} \right],$$

Now, to get the residual generation filter from the regression form (4), the following weighted least square cost function must be minimized:

$$\left\{ Y(i) - \Omega U(i) - \Gamma \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} \right\}^T \Pi \left\{ Y(i) - \Omega U(i) - \Gamma \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} \right\}. \quad (6)$$

Taking a derivation of (6) with respect to $[x^T(i) \quad f^T(i)]^T$ and setting it to zero, the filter gain matrix $\mathcal{H}$ for the residual generation filter $r(i) = \hat{f}(i)$ is given by

$$\mathcal{H} = \left[\left(\Gamma^T \Pi^{-1} \Gamma \right)^{-1} \Gamma^T \Pi^{-1} \right]_q, \quad (7)$$

where the subscript q means the lower q rows of $[(\Gamma^T \Pi^{-1} \Gamma)^{-1} \Gamma^T \Pi^{-1}]$. Therefore, the proposed finite memory residual generation filter for $r(i)$ is given by the simple matrix form with $\mathcal{H}$ is $Y(i) - \Omega U(i)$ as follows:

$$\begin{aligned} r(i) &= \begin{bmatrix} r_1(i) \\ r_2(i) \\ \mathrm{M} \\ r_q(i) \end{bmatrix} \\ &= \left[\left(\Gamma^T \Pi^{-1} \Gamma \right)^{-1} \Gamma^T \Pi^{-1} \right]_q \left[Y(i) - \Omega U(i) \right]. \end{aligned} \quad (8)$$

Each residual $r_s(i)$ in the residual $r(i)$ (8) can be obtained by

$$r_s(i) = \mathcal{H}_s \left[Y(i) - \Omega U(i) \right], \quad (9)$$

where $s = 1,2,\Lambda\ ,q$ and $\mathcal{H}_s$ is the s^{th} row of the $\mathcal{H}$ (7).

Several inherent properties of the proposed finite memory residual generation filter are described.

As shown in the following Theorem, the residual $r(i)$ given by the proposed residual generation filter on the window $[i-M, i]$ provides the exact fault $f(i)$ when there are no noises.

Theorem 3.1. *When $M \geq n$, the residual $r(i)$ given by the proposed finite memory residual generation filter on the window $[i-M, i]$ provides the exact fault $f(i)$ for noise-free systems.*

Proof: When there are no noises on the window $[i-M, i]$ for the discrete-time state space model (2) as follows:

$$\begin{aligned} \begin{bmatrix} x(i+1) \\ f(i+1) \end{bmatrix} &= A \begin{bmatrix} x(i) \\ f(i) \end{bmatrix}, \\ y(i) &= H \begin{bmatrix} x(i) \\ f(i) \end{bmatrix}, \end{aligned} \quad (10)$$

the most recent finite measurements and inputs $Y(i) - \Omega U(i)$ are determined from (4) as follows:

$$Y(i) - \Omega U(i) = \Gamma \begin{bmatrix} x(i) \\ f(i) \end{bmatrix}.$$

Therefore, the following is true:

$$\begin{aligned} \begin{bmatrix} \hat{x}(i) \\ \hat{f}(i) \end{bmatrix} &= \mathcal{H} \left[Y(i) - \Omega U(i) \right] \\ &= \left[\left(\Gamma^T \Pi^{-1} \Gamma \right)^{-1} \Gamma^T \Pi^{-1} \right] \left[Y(i) - \Omega U(i) \right] \\ &= \left[\left(\Gamma^T \Pi^{-1} \Gamma \right)^{-1} \Gamma^T \Pi^{-1} \Gamma \right] \begin{bmatrix} x(i) \\ f(i) \end{bmatrix} \\ &= \begin{bmatrix} x(i) \\ f(i) \end{bmatrix}. \end{aligned}$$

This means that $r(i) = \hat{f}(i) = f(i)$.

Theorem 1 states that the residual $r(i)$ tracks exactly its actual fault $f(i)$ at every time point for noise-free systems, although the proposed finite memory residual generation filter has been designed under the assumption that the

system (10) has an additive system and measurement noises, $w(i)$ and $v(i)$ as the fault model (1). This property indicates a finite convergence time and the fast tracking ability of the residual given by the proposed residual generation filter. Thus, it can be expected that the proposed residual generation filter might be appropriate for fast detection.

The filter gain matrix $\mathcal{H}_s$ in (9) only requires computation on the interval $[i-M, i]$ once and is time-invariant for all windows. This means that the proposed finite memory residual generation filter is time-invariant. However, discrete time-varying state space models can be often used for many practical and real-time applications. Thus, a finite memory residual generation filter for discrete time-varying systems is necessary. In this case, the computation for the filter gain matrix $\mathcal{H}_s$ might be burdensome. Hence, the iterative strategy of [16], [17] can be applied on the window $[i-M, i]$ to overcome the computational burden.

The window length M can be a useful design parameter for the proposed finite memory residual generation filter. Thus, the important issue here is how to choose an appropriate window length M that makes the residual performance as good as possible. The noise suppression of the proposed residual generation filter might be closely related to the window length M, and it can have greater noise suppression as the window length M increases, which improves the residual performance. However, the convergence time of a filtered residual becomes longer as the window length increases. This illustrates the proposed finite memory residual generation filter's compromise between noise suppression and tracking ability. Since M is an integer, fine adjustment of the properties with M is difficult. Moreover, it is difficult to determine the window length systematically. In applications, one method of determining the window length is to take the appropriate value that can provide sufficient noise suppression. Therefore, it can be stated from the above discussions that the window length M can be considered a useful parameter to make the residual performance of the proposed finite memory residual generation filter as good as possible.

4. SPECIFIED RESIDUAL FOR HARDWARE IMPLEMENTATION

In practice, it should be required that a fault detection system can be implemented with a discrete-time analog or digital hardware. In this case, the fault detection system should be specified to an algorithm or structure that can be realized in the desired technology. Thus, the proposed finite memory residual generation filter is specified to the well-known digital filter structure in [18] for its amenability to hardware implementation.

The filter gain matrix $\mathcal{H}_s$ for the s^{th} residual $r_s(i)$ in (9) can be defined by

$$\mathcal{H}_s \equiv \begin{bmatrix} h_s(M-1) & h_s(M-2) & \Lambda & h_s(0) \end{bmatrix}$$

Then, the proposed residual generation filter for the s^{th} residual $r_s(i)$ is given by

$$r_s(i) = \sum_{j=0}^{M-1} h_s(j) y(i-j) - \sum_{j=0}^{M-1} \left[\mathcal{H}_s \Omega\right]_j u(i-j), \qquad (11)$$

where $\left[\mathcal{H}_s \Omega\right]_j$ is the $(j+1)^{th}$ l elements of $\left[\mathcal{H}_s \Omega\right]$.

Applying the z-transformation to the residual (11) yields the following digital filter structure:

$$r_s(z) = \sum_{j=0}^{M-1} h_s(j) y(z) - \sum_{j=0}^{M-1} \left[\mathcal{H}_s \Omega\right]_j z^{-j} u(z), \qquad (12)$$

where $h_s(j)$ and $-\left[\mathcal{H}_s \Omega\right]_j$ become filter coefficients. It is noted that the digital filter structure (12) is a well-known moving average process whose functional relation between inputs $y(z)$, $u(z)$ and output $r_s(z)$ is nonrecursive. The block diagram of the proposed finite memory residual generation filter (8) for a hardware implementation can be represented as Fig.1.

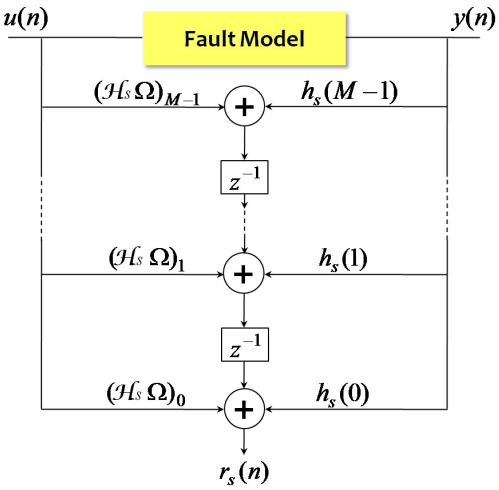


Fig.1. Block diagram representation of the finite memory residual generation filter.

5. EXTENSIVE COMPUTER SIMULATIONS

Extensive computer simulations using MATLAB software package are performed for a DC motor system with sensor faults to illustrate the fault detection capability of the proposed finite memory residual generation filter.

A. Discretized DC motor system

First, it is necessary to discretize the continuous-time state model of the DC motor system in order to facilitate the design in discrete-time state space model. The continuous-time state space model of the DC motor system can be represented by

$$\begin{aligned} x(t) &= \Psi x(t) + \Xi u(t), \\ y(t) &= Cx(t), \end{aligned} \tag{13}$$

where

$$x(t) = \begin{bmatrix} I_a(t) \\ w_m(t) \end{bmatrix}, \; u(t) = V_t(t),$$

$$\Psi = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K}{L_a} \\ -\frac{K}{J_m} & -\frac{B_m}{J_m} \end{bmatrix}, \; \Xi = \begin{bmatrix} \frac{1}{L_m} \\ 0 \end{bmatrix}, \; C = I,$$

with the armature current $I_a(t)$, the motor speed $w_m(t)$, the drive voltage $V_t(t)$, the armature resistance R_a, the armature inductance L_a, the motor inertial coefficient J_m, the motor viscous friction coefficient B_m. K represents both the motor torque constant and the back emf constant.

Using a sampling period T, the continuous-time state space model (13) can be discretized as follows:

$$\begin{aligned} x(i+1) &= \Phi x(i) + Du(i), \\ y(i) &= Cx(i), \end{aligned} \tag{14}$$

where

$$\Phi = e^{\Psi T}, \; D = \left(\int_0^T e^{\Psi \varepsilon} d\varepsilon \right) \Xi = \left(e^{\Psi T} - I \right) A^{-1} \Xi.$$

Then, the discretized DC motor system 14) can be extended by the ultimate discrete-time state space model (1) using sensor faults $f(i)$ as well as system and measurement noises $w(i)$, $v(i)$.

B. Computer simulation scenarios and results

Computer simulations are performed for the following discrete-time state space model for the DC motor system with sensor faults as well as system and measurement noises:

$$x(i+1) = \begin{bmatrix} -0.0005 & -0.0084 \\ 0.0517 & 0.8069 \end{bmatrix} x(i) + \begin{bmatrix} 0.1815 \\ 1.7902 \end{bmatrix} u(i) + \begin{bmatrix} 0.0006 \\ 0.0057 \end{bmatrix} \omega(i),$$

$$y(i) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(i) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} f(i) + v(i),$$

where $f(i) = [f_1(i) \quad f_2(i)]^T$ with the load torque sensor fault $f_1(i)$ and the motor speed sensor fault $f_2(i)$.

The performance of the proposed finite memory residual generation filter is evaluated and compared with a Kalman filtering-based residual generation filter with infinite memory structure through computer simulation in [4]-[9]. In these numerical examples, the window length is taken as $M = 10$ System noises are generated with covariances $Q_\omega = 0.01^2$ and $Q_\delta = diag(0.02^2 \;\; 0.02^2)$. Dealing with different noise levels requires that three cases of measurement noises are generated: high noise level with $R = diag(0.4^2 \;\; 1^2)$, medium noise level with $R = diag(0.2^2 \;\; 1^2)$, and low noise level with $R = diag(0.1^2 \;\; 1^2)$.

A fault is modeled for two scenarios as shown in Fig.2. The fault is modeled as an incipient soft bias-type fault in the first scenario, while the fault is modeled as an abrupt bias-type fault in the second; the second scenario might be more feasible than the first. The proposed finite memory residual generation filter is compared with the Kalman filtering-based approach with infinite memory structure for both scenarios.

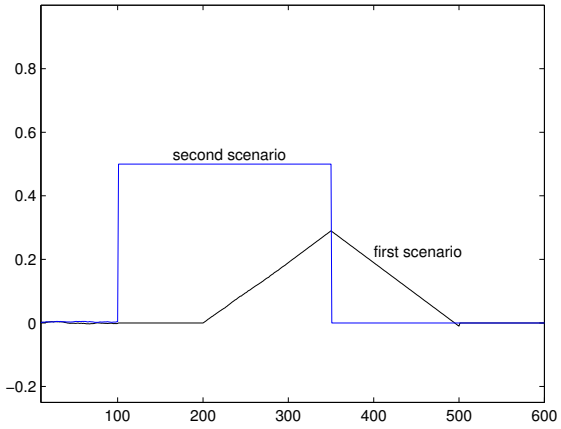


Fig.2. Two types of simulated faults.

Fig.3. and Fig.4. show plots for the residuals of both residual generation filters. As shown in Fig.3., the proposed finite memory residual generation filter can be compared to the Kalman filtering-based approach for the first scenario with incipient soft bias-type fault. In contrast, as shown in Fig.4., meaningful results are given for the second scenario with abrupt bias-type fault. For all cases of noise levels, the tracking of the proposed finite memory residual generation filter is much faster than that of the Kalman filtering-based approach when the abrupt bias-type fault occurs and disappears. One possible explanation for this is the finite convergence time and fast tracking ability of the proposed finite memory residual generation filter. However, in the case of high noise level, the Kalman filtering-based approach shows better noise suppression capabilities than the proposed finite memory residual generation filter. As mentioned in Section 3, the noise suppression of the proposed residual generation filter might be closely related to the window length M. The proposed residual generation

filter can gain better noise suppression as the window length grows, which means that the Kalman filtering-based residual generation filter with infinite memory structure can have good noise suppression ability.

Although the proposed finite memory residual generation filter can have greater noise suppression as the window length M increases, the tracking speed worsens in proportion with the window length M . This can be observed from simulations results according to diverse window lengths, as shown in Fig.5. These simulation results illustrate the proposed finite memory residual generation filter's compromise between noise suppression and the tracking speed of residuals.

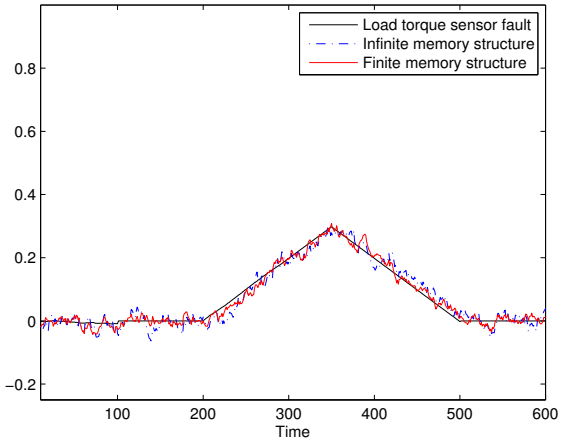
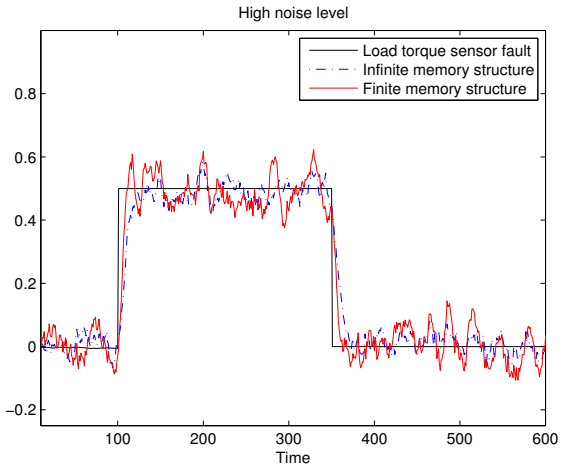


Fig.3. Residuals for the first scenario.

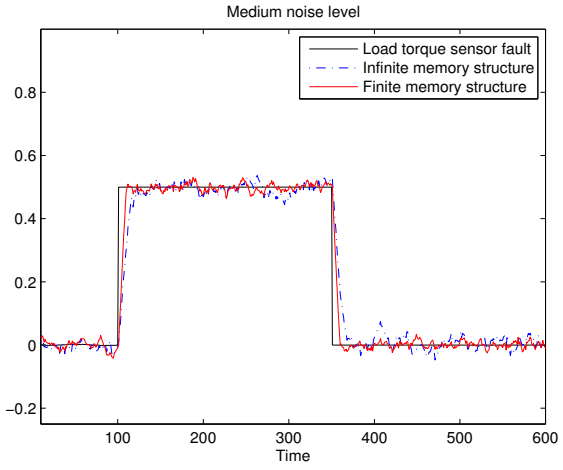
6. CONCLUDING REMARKS AND FUTURE WORKS

This paper has proposed a residual generation filter with finite memory structure for sensor fault detection. The proposed finite memory residual generation filter provides the residuals via real-time filtering of fault vectors using only the most recent finite measurements and inputs on the window. It has been shown that the residual given by the proposed residual generation filter provides the exact fault for noise-free systems. The proposed residual generation filter has been specified to the digital filter structure for its amenability to hardware implementation. The capability of the proposed residual generation filter has been verified through numerical examples for the discretized DC motor system with sensor faults. In particular, simulation results have been shown to be meaningful for abrupt bias-type fault according to diverse noise levels and window lengths.

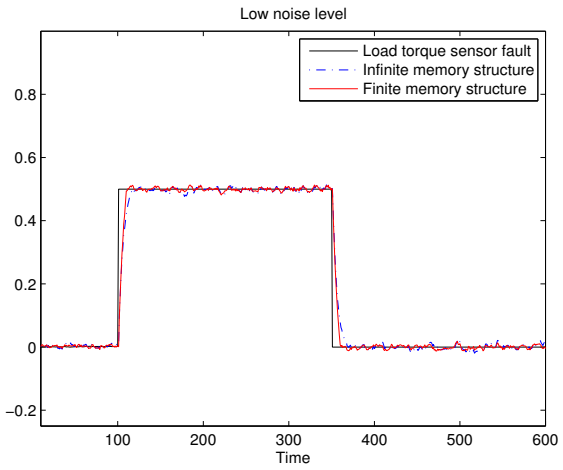
Although a guideline for the choice of window length M has been described, this could still be somewhat non-systematic. Therefore, a more systematic approach of determining window length should be researched in future work. In addition, noise covariance matrices Q and R might be useful design parameters under the assumption that information about them is unknown. Hence, another future work might examine how to tune noise covariance matrices to improve residual performance such as via noise suppression and tracking abilities.



a) High noise level : $R = diag(0.4^2 \;\; 1^2)$

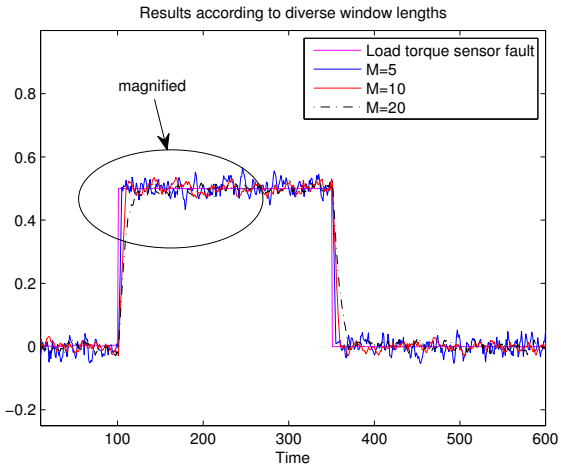


b) Medium noise level : $R = diag(0.2^2 \;\; 1^2)$

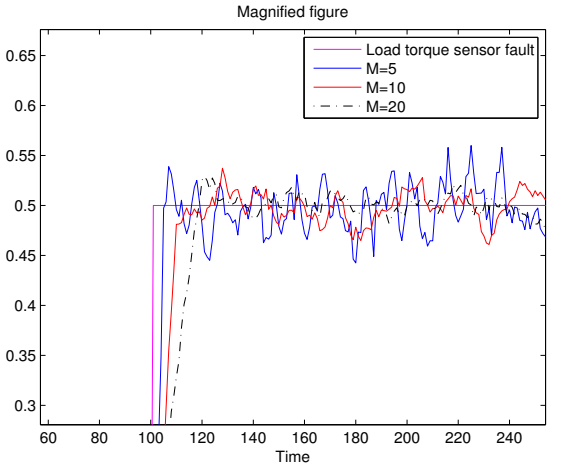


c) Low noise level : $R = diag(0.1^2 \;\; 1^2)$,

Fig.4. Residuals according to diverse noise levels for the second scenario.



a) Simulations results



b) Magnified figure

Fig.5. Residuals according to diverse window length for the second scenario.

REFERENCES

[1] Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N. (2003). A review of process fault detection and diagnosis - Part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27 (3), 293–311.

[2] Angeli, C., Chatzinikolaou, A. (2004). On-line fault detection techniques for technical systems: A survey. *International Journal of Computer Science & Applications*, 1 (1), 12–30.

[3] Hwang, I., Kim, S., Kim, Y., Seah, C. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE Transactions on Control Systems Technology*, 18 (3), 636–653.

[4] Kobayashi, T., Simon, D.L. (2005). *Enhanced bank of Kalman filters developed and demonstrated for inflight aircraft engine sensor fault diagnostics*. Research and Technology, NASA Glenn Research Center at Lewis Field 2005-213419, 25–26.

[5] Wang, Y., Zheng, Y. (2005). Kalman filter based fault diagnosis of networked control system with white noise. *Journal of Control Theory and Application*, 3 (1), 55–59.

[6] Tudoroiu, N., Khorasani, K. (2007). Satellite fault diagnosis using a bank of interacting Kalman filters. *IEEE Transactions on Aerospace and Electronic Systems*, 43 (4), 1334–1350.

[7] Xue, W., Guo, Y., Zhang, X. (2008). Application of a bank of Kalman filters and a robust Kalman filter for aircraft engine sensor/actuator fault diagnosis. *International Journal of Innovative Computing, Information and Control*, 4 (12), 3161–3168.

[8] Tudoroiu, N. (2011). Real time embedded Kalman filter estimators for fault detection in a satellite's dynamics. *International Journal of Computer Science & Applications*, 8 (1), 83–109.

[9] Villez, K., Srinivasanb, B., Rengaswamyb, R., Narasimhanc, S., Venkatasubramaniana, V. (2011). Kalman-based strategies for fault detection and identification (FDI): Extensions and critical evaluation for a buffer tank system. *Computers and Chemical Engineering*, 35 (5), 806–816.

[10] Bruckstein, A.M., Kailath, T. (1985). Recursive limited memory filtering and scattering theory. *IEEE Transactions on Information Theory*, 31 (3), 440–443.

[11] Kim, P.S. (2010). An alternative FIR filter for state estimation in discrete-time systems. *Digital Signal Processing*, 20 (3), 935–943.

[12] Kim, P.S. (2013). A computationally efficient fixed-lag smoother using recent finite measurements. *Measurement*, 46 (1), 846–850.

[13] Zhao, S., Shmaliy, Y.S., Huang, B., Liu, F. (2015). Minimum variance unbiased FIR filter for discrete time-variant systems. *Automatica*, 53 (2), 355–361.

[14] Pak, J., Ahn, C., Shmaliy, Y., Lim, M. (2015). Improving reliability of particle filter-based localization in wireless sensor networks via hybrid particle/FIR filtering. *IEEE Transactions on Industrial Informatics*, 11 (9), 1–10.

[15] Kim, P.S., Lee, E.H., Jang, M.S., Kang, S.Y. (2017). A finite memory structure filtering for indoor positioning in wireless sensor networks with measurement delay. *International Journal of Distributed Sensor Networks*, 13 (1), 1–8.

[16] Kwon, W.H., Kim, P.S., Han, S.H. (2002). A receding horizon unbiased FIR filter for discrete-time state space models. *Automatica*, 38 (3), 545–551.

[17] Zhao, S., Shmaliy, Y.S., Liu, F. (2015). Fast Kalman-Like optimal unbiased FIR filtering with applications. *IEEE Transactions on Signal Processing*, 64 (9), 2284–2297.

[18] Oppenheim, A., Schafer, R. (1989). *Discrete-Time Signal Processing*. Prentice Hall.

Received October 25, 2016.
Accepted March 14, 2017.

MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 2, 83-92

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: http://www.degruyter.com/view/j/msr

ISSN 1335-8871

Interactions and Optimizations Analysis between Stiffness and Workspace of 3-UPU Robotic Mechanism

Dan Zhang, Bin Wei

Department of Mechanical Engineering, Lassonde School of Engineering, York University, 4700 Keele Street, Toronto ON. M3J 1P3 Canada, dzhang99@yorku.ca, binwei28@yorku.ca

The interactions between stiffness and workspace performances are studied. The stiffness in x, y and z directions as well as the workspace of a 3-UPU mechanism are studied and optimized. The stiffness of the robotic system in every single moveable direction is measured and analyzed, and it is observed that in the case where one tries to make the x and y translational stiffness larger, the z directional stiffness will be reduced, i.e. the x and y translational stiffness contradicts with the one in z direction. Subsequently, the objective functions for the summation of the x and y translational stiffness and z directional stiffness are established and they are being optimized simultaneously. However, we later found that these two objectives are not in the same scale; a normalization of the objectives is thus taken into consideration. Meanwhile, the robotic system's workspace is studied and optimized. Through comparing the stiffness landscape and the workspace volume landscape, it is also observed that the z translational stiffness shows the same changing tendency with the workspace volume's changing tendency while the x and y translational stiffness shows the opposite changing tendency compared to the workspace volume's. Via employing the Pareto front theory and differential evolution, the summation of the x and y translational stiffness and the volume of the workspace are being simultaneously optimized. Finally, the mechanism is employed to synthesize an exercise-walking machine for stroke patients.

Keywords: Robotic manipulator, stiffness, workspace, interactions, optimization, walking machine.

1. INTRODUCTION

Parallel mechanisms have been utilized in different fields such as conduct manufacturing machining [1]-[3], medical devices [4]-[7], sensor applications [8], etc., contributing to parallel mechanisms' parallel structure arrangement (i.e. high rigidity, high accuracy, high speed and acceleration, and no cumulative joint/link error). Stiffness and workspace analysis are widely addressed topics. There are numerous resources dealing with stiffness and workspace of parallel mechanisms [9]-[13]. Due to the space limitation, here the authors just list some typical examples of them. Stiffness can be one of the most critical factors to be considered as high stiffness can result in better precision. For example, in machine tools applications, if the machine tools have poor stiffness performance, the machined piece will be very rough. The main diagonal items of the stiffness matrix are the stiffness in each corresponding direction and they can represent the pure stiffness more clearly, this has been proved by the FEA examination [14], and these items can be the primary components when one analyzes and optimizes the stiffness. In [15], the average value and standard deviation of the compliance matrix's main diagonal items for a parallel mechanism were employed to describe and optimize the kinetostatic performance. However, the authors did not consider the unit incompatibility of each element of the compliance matrix when determining the average value and the standard deviation of the compliance matrix's main diagonal items. In [16], the main diagonal items of the stiffness matrix were utilized as an objective function to analyze a parallel manipulator's stiffness; this study followed the study in [15]. In [17], the stiffness in z direction was employed as a factor to represent the entire machine's overall stiffness. In [18], a general performance index was used for optimizing the stiffness of a robotic manipulator. The drawback of parallel manipulators, compared to their serial ones, is that parallel mechanisms reach smaller workspace. For a parallel manipulator, its workspace is the space to which the moving platform's center is able to reach. One usually desires a large workspace for parallel mechanisms, for example, having a large workspace capacity for parallel mechanisms used in machine tools usually indicates that the machine tool-head is able to stretch out to every corner of the work-piece when one is running milling, cutting, deburring, drilling, and finishing operations. In [19], a condition index was put forward as a criterion for optimizing the workspace for a three degree-of-freedom robotic mechanism for achieving a "well-conditioned" workspace performance. In [20] and [21], the volume of the workspace of robotic mechanisms was employed as a criterion for optimizing the workspace.

DOI: 10.1515/msr-2017-0011

For the majority of cases, the workspace volume of robotic mechanisms is usually employed as a criterion for assessing and optimizing the workspace performance.

Through analyzing, it is observed that the parallel mechanisms' stiffness can get affected during the time one optimizes the workspace. Usually both the stiffness and workspace performances need to be taken into consideration in some applications; hence, performing a multi-objective optimization on these two terms is necessary. In this study, the stiffness and workspace are single optimized and analyzed firstly; subsequently these two terms are optimized and analyzed simultaneously.

Numerous optimization approaches can be employed, for example, differential evolution, particle swarm optimization, ant colony optimization, and genetic algorithm. In [14], an approach that is based on the particle swarm optimization was employed for optimizing the dexterous stiffness and reachable workspace of the 4UPS-PU robotic mechanism simultaneously. In [22], the authors optimized the dexterity and the workspace of a tripod machine structure at the same time by resorting to the combination of the Pareto frontier sets and evolutionary algorithm. In [23], a multi-objective optimization problem was transformed to a single-objective optimization issue. The average vertical stiffness, volume of the workspace and average dexterity for a hybrid robotic mechanism that is reconfigurable were optimized simultaneously via employing the sequential simplex approach. In [24], the GA and Pareto front set approach were individually utilized for optimizing a tripod based parallel manipulator's stiffness and its workspace.

The differential evolution algorithm usually performs better than particle swarm optimization in the aspects of solution accuracy, robustness and time cost. In most cases, differential evolution is easier to be implemented as compared to other algorithms. In addition, differential evolution is much more efficient and robust in terms of convergent ability and producing the same results in multiple runs as compared to other evolution algorithms; the above conclusion was proved in [25]. Interestingly, to the best of the authors' knowledge, differential evolution algorithm is not often utilized in the area of optimization for parallel mechanisms; meanwhile the normalization of stiffness in every moveable direction was not taken into consideration in most cases. Here with respect to the single-objective optimization problem, the authors employ the differential evolution algorithm for optimizing the stiffness and workspace of the 3-UPU parallel mechanism. With respect to the multi-objective optimization issue, the summation of the x and y translational stiffness and the volume of the robotic mechanism's workspace are being optimized simultaneously through respectively employing the Pareto front theory and differential evolution algorithm.

The organization of this paper is as follows: Section 2 presents the Jacobian matrix analysis of the 3-UPU parallel mechanism; the stiffness analysis of the mechanism and single-objective optimization and some issues are studied in Section 3; Section 4 analyzes the workspace of the mechanism and single-objective optimization and some issues are also discussed the multi-objective optimization for stiffness and workspace is discussed and analyzed in Section 5, and the motivation for choosing the objective functions, the used optimization algorithm and the computational costs are presented as well; Section 6 presents a potential application of the mechanism used in the medical field finally conclusion is given in Section 7.

2. JACOBIAN OF THE MECHANISM

Determination of the robotic mechanism's Jacobian matrix is essential for the following analysis. The 3-UPU robotic mechanism possesses three translational degrees of freedom (DOF). Three identical limbs connect the moving platform and the base, as illustrated in Fig.1. For the purpose of analysis, two frames O(X, Y, Z) and P(x, y, z) are attached to the base $B_1B_2B_3$ and moving platform $P_1P_2P_3$ center, respectively. Angle α_b is from X axis to OB_1; angle α_p is from x axis to OP_1.

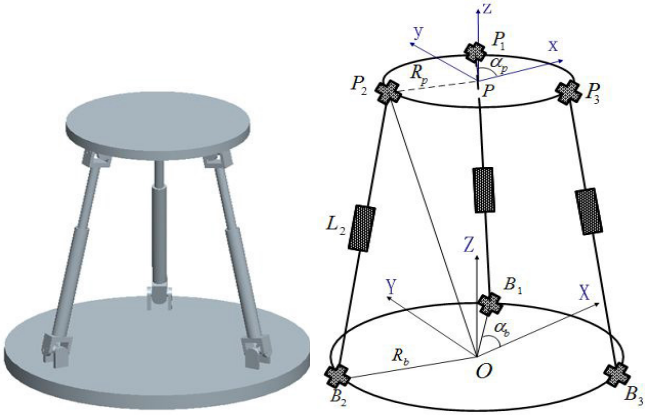


Fig.1. Schematic representation of 3-UPU manipulator and its model.

The velocity of P_i can be written as:

$$v_{Pi} = \omega_i \times s_i + \dot{L}_i s_i \tag{1}$$

where s_i represents the unit vector along the i-th leg, w_i represents the angular velocity of the i-th leg relative to the base, and *Li* represents the length of each leg that is able to be determined via the vector loop equation. Through multiplying s_i in both sides of formulation (1) results (2):

$$Jv_P = \dot{q} \tag{2}$$

where v_p represents the moving platform center's velocity. The Jacobian matrix of the 3-UPU robotic mechanism is thus determined:

$$J = \begin{bmatrix} s_1^T & s_2^T & s_3^T \end{bmatrix}^T \tag{3}$$

3. STIFFNESS OF THE MECHANISM AND SINGLE-OBJECTIVE OPTIMIZATION

The stiffness can be considered as a "bridge" between the displacement of the moving platform and a force acting on it, and the 3-UPU robotic mechanism's stiffness at a given spot inside the workspace can be portrayed by a stiffness matrix [26]. The stiffness matrix is determined through employing the kinematic and static formulations:

$$K = J^T K_J J = k J^T J \tag{4}$$

The main diagonal items of the stiffness matrix are each corresponding direction's stiffness. As a case study, when R_b = 3 cm, R_p = 1 cm, x = 0, y = 0, z = 5 cm, the changing trends of the x, y and z translational stiffness with respect to α_b (rad) and α_p (rad) are illustrated in Fig.2.

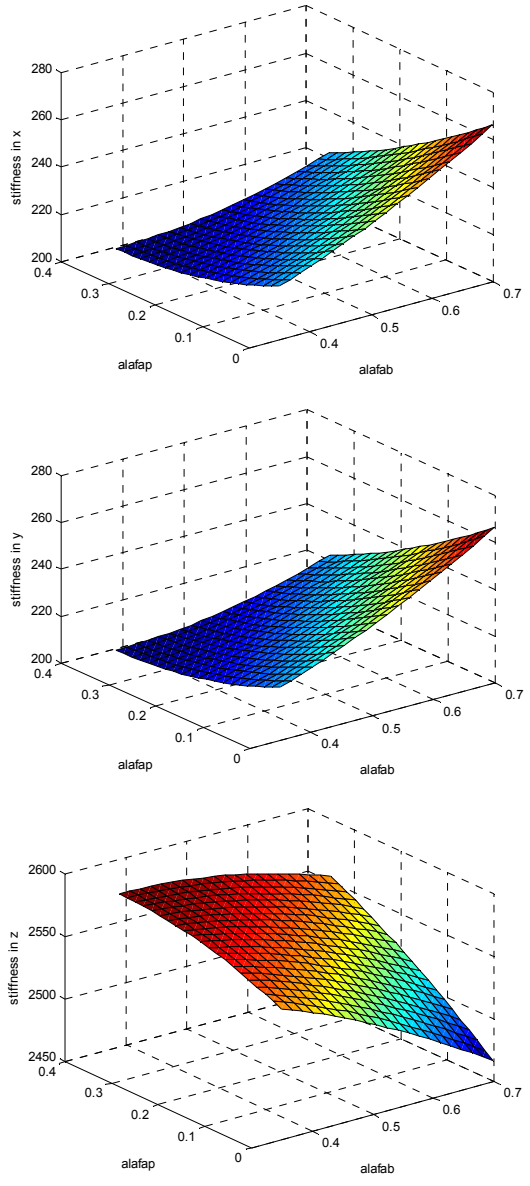


Fig.2. Stiffness in three directions when $R_b = 3$, $R_p = 1$.

In the case where x = 0, y = 0, z = 5 cm, α_b = 0.5 rad, α_p = 0.25 rad, the changing trends of the x, y and z translational stiffness with respect to R_b and R_p are demonstrated in Fig.3.

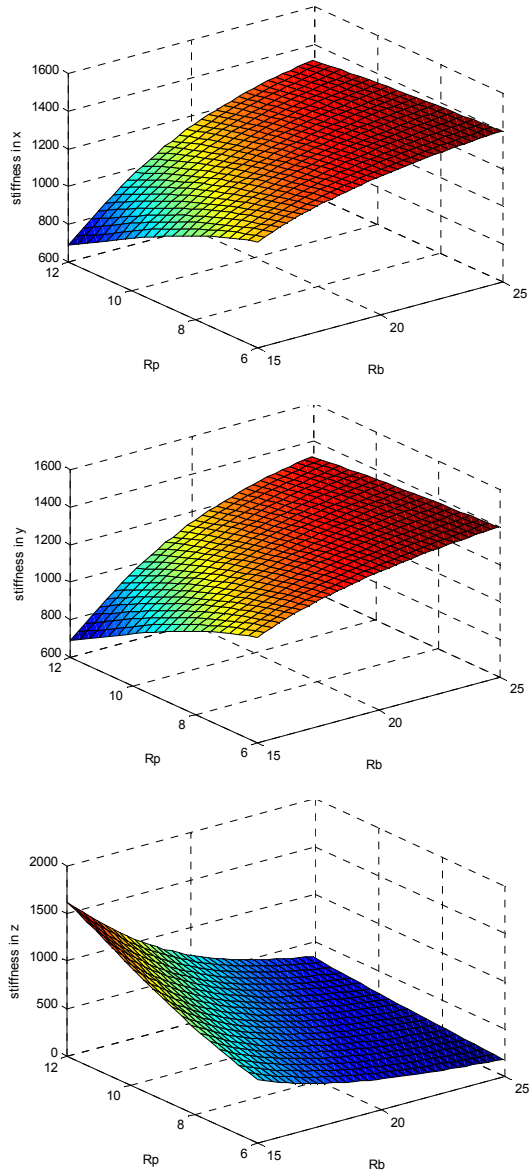


Fig.3. Stiffness in x, y and z direction when $\alpha_b = 30^o$, $\alpha_p = 15^o$.

It can be observed that the x and y translational stiffness present the same changing fashion while the z translational stiffness presents the reverse changing fashion as compared to the x and y stiffness case. This indicates that in the case where the x and y translational stiffness increase, the z translational stiffness decreases, i.e. these two types of stiffness conflict with one another.

The weighted aggregation function as illustrated below will be employed for optimizing the x, y and z translational stiffness. It transforms a multi-objective optimization issue to a single-objective optimization case through integrating

multiple objective functions into a single objective function [27]. However, in the case where the objectives are not in the same scale, a normalization of the different objectives needs to be taken into consideration. The formulation shown below is able to be determined via joining two objectives to one:

$$ObjF = \lambda_1 \frac{ObjF1 - ObjF1_{\min}}{ObjF1_{\max} - ObjF1_{\min}} + \lambda_2 \frac{ObjF2 - ObjF2_{\min}}{ObjF2_{\max} - ObjF2_{\min}} \quad (5)$$

$$ObjF1 = K(1,1) + K(2,2) \quad (6)$$

$$ObjF2 = K(3,3) \quad (7)$$

where ObjF represents the objective function, $ObjF1$ represents the summation of x and y directional stiffness, $ObjF2$ represents the z directional stiffness, $ObjFi_{\max}$ represents the top limit of the corresponding objectives, $ObjFi_{\min}$ represents the low limit of the objectives, and λ_i represents the weight.

The design variables are as follows: R_b, R_p, α_b, α_p, where R_b is the base radius, R_p is the moving platform radius, angle α_b is from X axis to line OB_1, angle α_p is from x axis to line OP_1. The design variables' constraints are set as follows: $R_b \in [15, 20]$ cm, $Rp \in [6,12]$ cm, $\alpha_b \in [0.35, 0.7]$ rad, $\alpha_p \in [0, 0.35]$ rad.

The differential evolution algorithm is employed here for optimizing the above objective function. The reason that differential evolution was chosen over the particle swarm optimization algorithm is that the differential evolution algorithm performs better than particle swarm optimization with regard to the solution accuracy, robustness and time cost. Furthermore, differential evolution can be implemented without great effort as compared to the other algorithms, and also differential evolution is much more efficient and robust in terms of convergent ability and producing the same end results in multiple runs as compared to other evolution algorithms. Storn and Price put forward the differential evolution in the year 1996. It is a stochastic and population-based optimization algorithm. Three main operations exist in differential evolution: mutation, crossover, and selection.

Mutation: In differential evolution algorithm, three vectors $Xa_{,G}$, $X_{b,G}$ and $Xc_{,G}$ are chosen arbitrarily from the current population in order to perform the operation mutation.

$$Vi_{,G} = Xa_{,G} + F(X_{b,G} - Xc_{,G})$$

Usually, F is a constant value ranging from zero to two and it determines the amplification of $(X_{b,G} - Xc_{,G})$. Larger F value results in better diversity while lower value leads to faster convergence.

Crossover: New results can be produced through shuffling competing vectors and it is also possible to expand the population's diversity. The trial vector is defined as follows [28]:

$$U_{i,G} = (U_{1i,G}, U_{2i,G}, U_{3i,G}, ..., U_{Di,G})$$

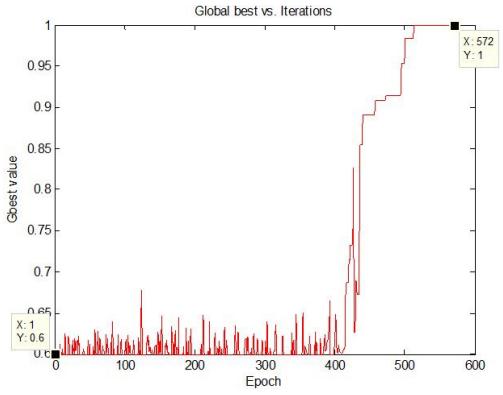
$$U_{ji,G} = \begin{cases} V_{ji,G} & if\,(rand \le Cr) or (j = j_{rand}) \\ X_{ji,G} & otherwise \end{cases}$$

where D is the problem's dimension, j_{rand} is a randomly selected factor to make sure that at least one parameter can be always chosen from $V_{ji,G}$. $Cr \in (0,1)$ represents the crossover rate. $rand$ represents a random value ranging from 0 to 1.

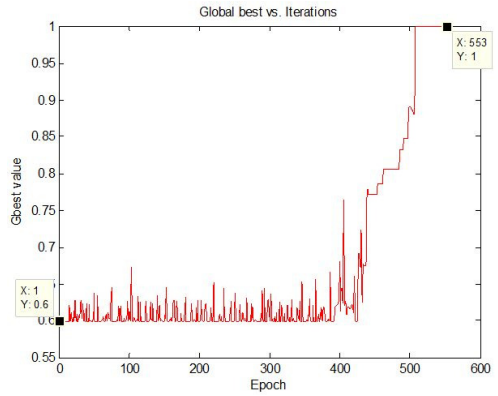
Selection: The vector with better objective function value will be chosen and entered to the following iteration from the trial vector. For instance, for a minimum optimization issue, the vector that has the smallest objective function value will be chosen.

$$X'_i = \begin{cases} U_i & if\,(fitness(U_i) \le fitness(X_i)) \\ X_i & otherwise \end{cases}$$

For the problem in this study, the below results can be obtained after optimization through employing the differential evolution algorithm:



a) First trial



b) Second trial

Fig.4. Optimization results of stiffness employing differential evolution.

The objective function's value converges to one after approximately 500 iterations. The performance graph illustrated in Fig.4. vibrates because $ObjFi_{\max}$ and $ObjFi_{\min}$ change in every single iteration. The x axis represents the number of iterations and the y axis represents the best value discovered for the objective function in every single iteration. After performing the optimization, the base radius R_b = 20 cm, the moving platform radius R_p = 6 cm, the angle α_b from X axis to OB_1 is 0.6981rad, and the angle α_p from x axis to OP_1 is zero. The summation of the x and y translational stiffness $ObjF1$ = 2729.4, and the z translational stiffness $ObjF2$ = 270.6122.

4. WORKSPACE OF THE MECHANISM AND SINGLE-OBJECTIVE OPTIMIZATION

Regarding the optimization for the workspace, as mentioned earlier, the robotic mechanisms' workspace volume is usually employed to be an objective function to perform the optimization. One is able to calculate the volume through employing the fast search approach [21]. Under the case α_b = 0.5 rad, α_p = 0.25 rad, the volume changing trend with respect to R_b and R_p is shown in Fig.5.

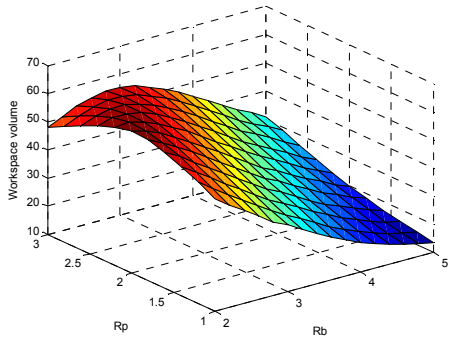


Fig.5. Workspace distribution along R_b and R_p when $\alpha_b = 30^o, \alpha_p = 15^o$.

When R_b = 4 cm, R_p = 2 cm, the volume changing trend with respect to α_b and α_p is shown in Fig.6.

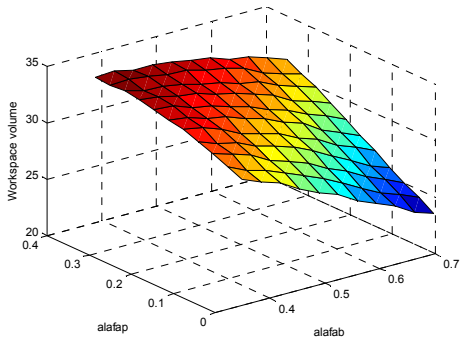
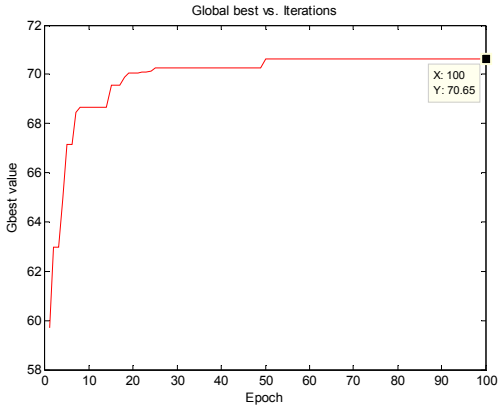
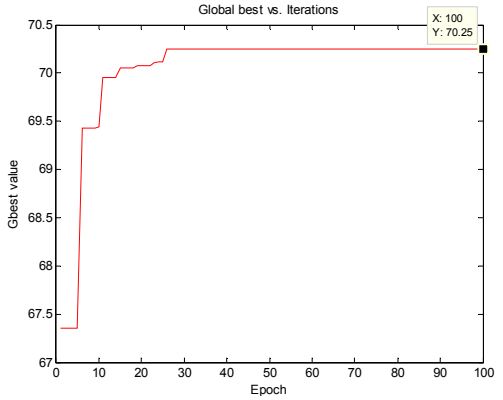


Fig.6. Workspace distribution with respect to α_b and α_p when $R_b = 4$, $R_p = 2$.

From Fig.5. and Fig.6., it is observed that the base radius, moving platform radius and angles on the two platforms all make a dent on the robotic mechanism's workspace volume. In the scenario where these four parameters alter at the same time, it will be very difficult to determine which parameters should be selected to produce the maximum volume of the workspace. Now the single-objective optimization is conducted through employing the differential evolution via adjusting the four design variables to generate the maximum volume for the workspace. The variables and its constraints are listed below: $R_b \in [2, 5]$ cm, $R_p \in [1, 3]$ cm, $\alpha_b \in [0.35, 0.7]$ rad, $\alpha_p \in [0, 0.35]$ rad. After performing the optimization, the results shown below are generated:



a) First time trial



b) Second time trial

Fig.7. Optimization results of workspace volume employing DE.

From Fig.7., it is observed that the objective function is converging to 70.25 after approximately 50 iterations and the design variables are determined as R_b = 2.1163 cm, R_p = 2.1244 cm, α_b = 0.36 rad, α_p = 0.35 rad after optimization. Prior to performing the optimization, the volume of the workspace is 49.8384. Fig.8. illustrates the comparison among the workspaces, one is after optimization case and another one is before optimization case. One can see that the volume of the workspace turns larger as compared to the one before optimization, and it is improved about 1.4 times.

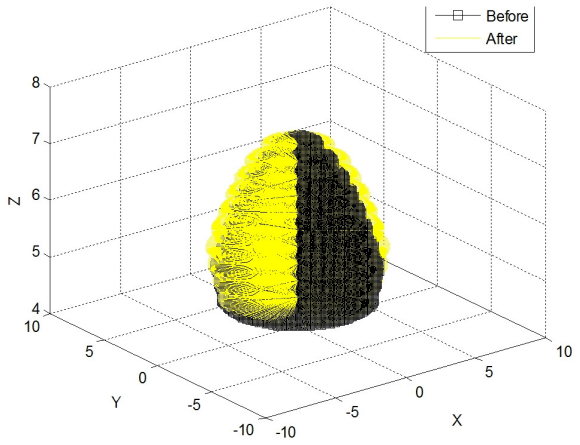


Fig.8. Workspace comparison after optimization and prior to optimization.

5. MULTI-OBJECTIVE OPTIMIZATION FOR STIFFNESS AND WORKSPACE

Usually in most scenarios for the parallel mechanisms, when the stiffness increases, the workspace is influenced by that. Here, the changing trends for the stiffness and workspace with respect to R_b, R_p, α_b and α_p are firstly investigated in order to see if these two terms conflict or not. In the case where α_b = 0.5 rad, α_p = 0.25 rad, the changing trends for the summation of the x and y directional stiffness, z directional stiffness and the volume of the workspace with respect to R_b and R_p are illustrated in Fig.9.

When R_b = 4 cm, R_p = 2 cm, the changing trends for the summation of the x and y translational stiffness and the volume of the workspace with respect to α_b and α_p are demonstrated in Fig.10.

From Fig.9. and Fig.10., one can see that the z translational stiffness and the volume of the workspace possess the same changing trend while the summation of x and y translational stiffness and the volume possess the reversing changing trend. In the case where the z translational stiffness increases, the volume of the workspace increases as well, this is what is desired. However, in the case where the x and y translational stiffness increase, the volume decreases, which is not what is wanted, these two conflict with each other. Thus, one can employ the differential evolution to single-objective optimize the z translational stiffness independently without taking the performance of the workspace into consideration. However, the z translational stiffness has the reverse changing trend with the x and y translational stiffness, when one resorts to the differential evolution to optimize the z translational stiffness, the x and y translational stiffness will get influenced. Thus, in practical applications, engineers need to take into consideration what factor is more important to them in order to select one to perform the optimization, i.e. z translational stiffness or x and y translational stiffness. As mentioned earlier, the reason the differential evolution is preferably selected here is that it is more efficient and robust in terms of convergent ability and produces the same results in multiple trails as compared to other evolutionary algorithms. Additionally, the differential evolution algorithm generally performs better than the particle swarm optimization with respect to solution accuracy, robustness and time cost. Furthermore, the differential evolution is easier to be implemented as compared to the other algorithms [25]. The authors here deem that the x and y translational stiffness is more critical than z translational stiffness. The summation of the x and y translational stiffness and the volume of the workspace are being optimized at the same time through employing the Pareto front theory.

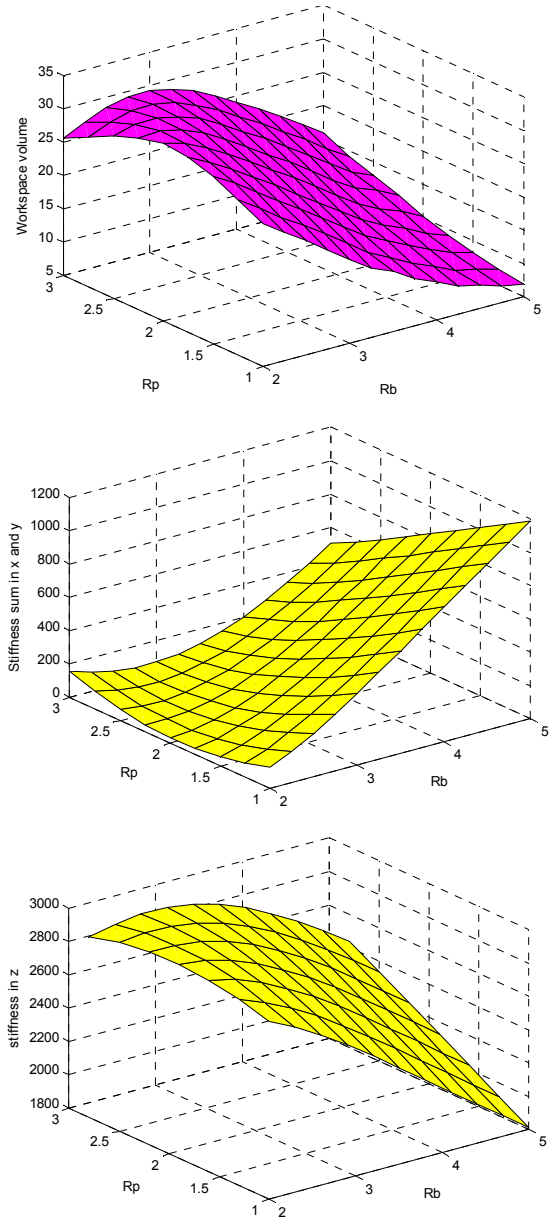


Fig.9. Summation of x and y directional stiffness, z directional stiffness and volume of workspace along R_b and R_p when $\alpha_b = 30^o, \alpha_p = 15^o$.

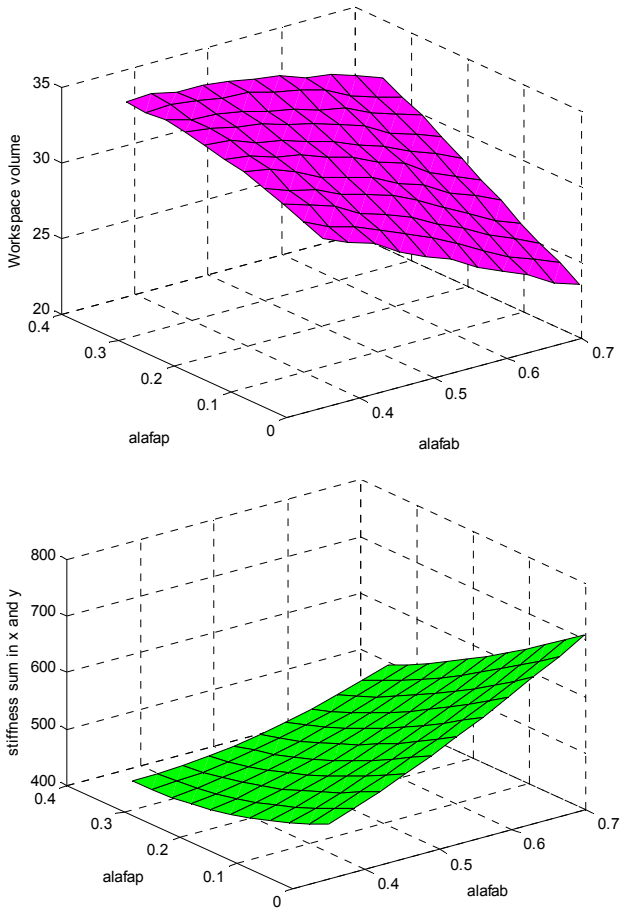


Fig.10. Summation of x and y directional stiffness and the volume of the workspace along with α_b and α_p when $R_b = 4$, $R_p = 2$.

Objective function 1:

$$ObjF1 = K(1,1) + K(2,2) \tag{8}$$

where ObjF1 represents the summation of the x and y directional stiffness. Maximizing ObjF1 is equivalent to minimize -ObjF1.

Objective function 2:

$$ObjF2 = WorkspaceVolume \tag{9}$$

where ObjF2 represents the volume of the workspace for the robotic mechanism. Similarly, maximizing ObjF2 is equivalent to minimizing -ObjF2.

Design variables are R_b, R_p, α_b and α_p. The constraints are listed below: $R_b \in [2, 5]$ cm, $R_p \in [1, 3]$ cm, $\alpha_b \in [0.35, 0.7]$ rad and $\alpha_p \in [0, 0.35]$ rad. The authors chose the population size to be 50, the maximum of generations to be 100, the selection strategy as tournament, the tournament size to be 2, the crossover type as intermediate, the crossover ratio to be 1, the mutation function to be as adaptive feasible, and the Pareto front population fraction to be 0.35. After performing the optimization, the Pareto front of the objective functions 1 (summation of x and y translational stiffness) and 2 (volume of workspace) is generated (illustrated in Fig.11.). It can be observed that the summation of the x and y translational stiffness and the volume of the workspace do conflict with one another. When the summation of the x and y translational stiffness is increasing, the volume of the workspace is decreasing and vice versa.

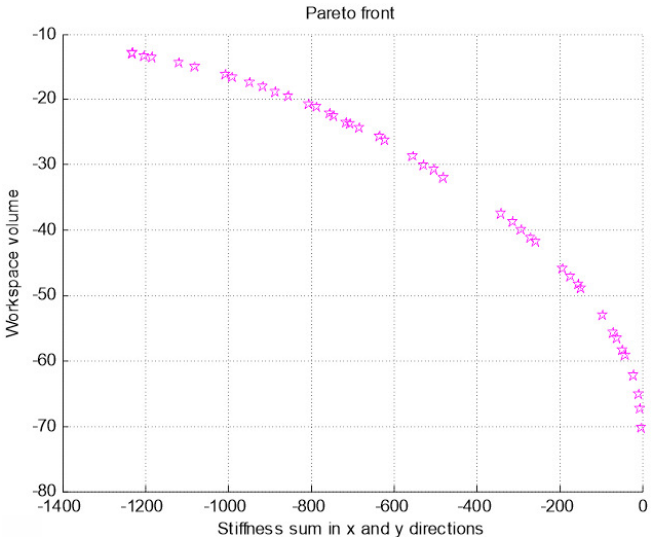


Fig.11. Pareto front sets between stiffness and workspace.

There is not one optimal solution as it is the case for the single-objective case, but multiple solutions, to which they are non-dominate. Some of the objective values and values for the design variables are recorded in Table 1. Here, the authors will not enumerate all of them as there are too many. Engineers are able to select specific ones that are based on their own requirements. In the scenario where engineers demand larger stiffness and the volume of the workspace is not vital, the one that has larger value for the stiffness can be selected and some workspace can be compromised to some extent. In the scenario where engineers deem workspace critical, they can select a larger value for workspace and compromise some stiffness.

Table 1. Pareto front solutions.

$ObjF1$	$ObjF2$	R_b	R_p	α_b	α_p
-4.0869	-67.3870	2.1198	2.0068	0.3950	0.3242
-1004.638	-16.2115	4.7256	1.3232	0.6025	0.1977
-269.7363	-41.1384	2.8959	1.3991	0.5062	0.2676

The Pareto front theory was employed to multi-objective optimize the summation of the x and y translational stiffness and the volume. Here the authors are also able to transform the multi-objective optimization issue into a single-objective case via the formulation below:

$$ObjF = ObjF1 \cdot ObjF2 \tag{10}$$

Through Equation (10), the multi-objective case becomes a single-objective case. The design variables are R_b, R_p, α_b and α_p, and the constraints are listed below: $R_b \in [2, 5]$ cm, $R_p \in [1, 3]$ cm, $\alpha_b \in [0.35, 0.7]$ rad, $\alpha_p \in [0, 0.35]$ rad. After performing the optimization, the following are generated:

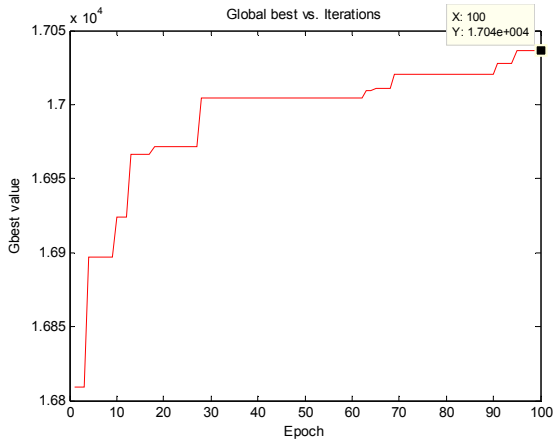


Fig.12. Optimization results employing DE.

It can be seen that the objective function is converging to 1.704×10^4 after about 100 generations, and the values for the design variables are $R_b = 4.2876$ cm, $R_p = 1.7948$ cm, $\alpha_b = 0.6267\ rad$ and $\alpha_p = 0.0629\ rad$ after the optimization. Prior to the optimization, $R_b = 3$ cm, $R_p = 2$ cm, $\alpha_b = 0.523\ rad$, $\alpha_p = 0.3\ rad$ and $ObjF = 6.7516\times10^3$. $ObjF$ has improved approximately 2.5 times after the optimization. It is observed that the single-objective optimization possesses only one optimal solution while the multi-objective case possesses multiple solutions, from which engineers are able to select based on their specific requirements and preferences, thus the final selection is depending on engineers.

6. POTENTIAL APPLICATION - PARALLEL ROBOTIC BASED ASSISTED MACHINE FOR STANDING AND WALKING EXERCISE FOR STROKE PATIENTS

A. Background

Nowadays, unfortunately many people have suffered from stroke, which causes people to have lasting consequences, i.e. people who suffer from stroke have physical disabilities or even worse (become vegetable person). Our study focuses on how to help those patients exercise so that patients with disabilities can be prevented from organ deterioration and muscle shrinks. In this section, the authors propose an assisted exercise standing and walking machine based on the parallel robotic structure for those patients. To the best of the authors' knowledge, this will be the first exercise machine for patients with stroke disability based on the parallel robotic structure.

When people suffer from stroke, sometimes the patients will not have any lasting consequences and act like a normal person, but sometimes people will have severe disabilities, the symptom can be that patients lose walking and standing abilities, or sometimes patients become vegetable persons. For the latter case, if the patients are bed ridden for a long time, their organs will deteriorate and their muscles will shrink. In most cases, in second world countries, such as in China and India, etc., where a large number of stroke patients reside, two or three family members just hold the patients and help the patients stand and walk, which is extremely energy consuming for the helpers.

B. Solution - Parallel robotic based exercise machine

To overcome the above problem, the authors designed an exercise standing and walking machine for the patients with stroke disabilities based on the parallel robotic structure, as shown in Fig.1. The machine can not only help patients stand, but also can help patients walk, so as to act as an exercise machine in order to prevent patients from organ deterioration and muscle shrink. The machine consists of two main parts, the lower component and upper component. The lower component mainly consists of three 3-UPU parallel robotic mechanisms, U stands for the universal joint, P stands for the prismatic joint. Each 3-UPU parallel robotic mechanism has three translational degrees of freedom. These translational degrees of freedom enable the patients' legs to move up and down and move forward and backward. The upper component can prevent the patients' upper body from falling, and the lower component makes the patients move around.

The machine consists of the following components: part 1 is the moving platform used to support the neck section of a human body, the middle platform 3 is connected to the moving platform 1 with several links, and they are denoted as part 2. The middle platform is used to support the waist of a human body. The middle platform is connected to the base platform (part 11) through universal joints (part 4), actuated prismatic joints (part 6) and links (parts 5 and 7). There are two wheels (part 8) in the base platform, which is used to move the whole machine when a patient finishes a single step cycle. Part 9 is the opening section that allows patients to get into this machine. Part 10 is a motor that is used to actuate the prismatic joints. There are another two 3-UPU mechanisms in the middle, which are used to move the legs of the patients, i.e. lift up the legs of the patients, move forward the legs and put down the legs on the floor. In the meantime, the large 3-UPU mechanism is used to move the waist section of a human body to move along with the legs. When the patients wear this machine, we only need to slightly move the machine when the patients finish a step cycle, which can greatly alleviate our helpers. The proposed parallel robotic based assisted machine will be prototyped according to the optimized results in the previous section and it will be further tested by the patients in terms of suitability of the proposed machine in the near future.

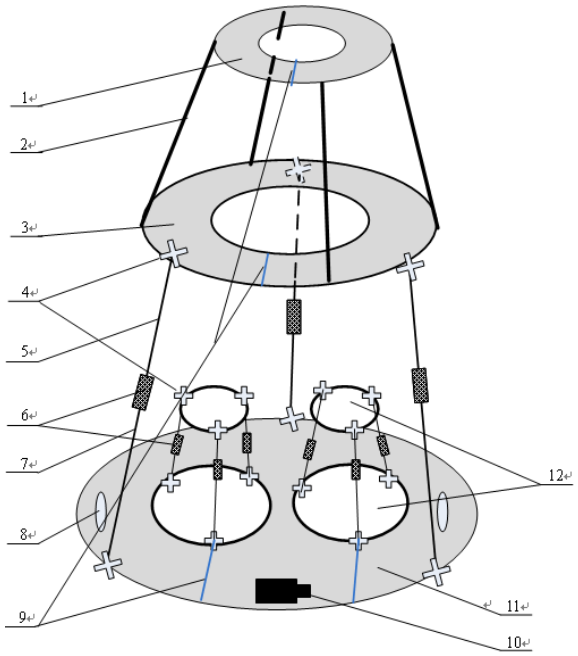


Fig.13. Parallel robotic based assisted machine.

7. CONCLUSION

The interactions between stiffness and workspace performances are investigated in this paper, and also the stiffness and workspace of a 3-UPU robotic mechanism are optimized, respectively. Since the summation of the x and y directional stiffness and the z directional stiffness are not in the same scale, a normalization of the two objectives is performed. Furthermore, via analyzing it was observed that the changing trends for the x and y translational stiffness and the volume of the workspace are reversing whereas the z translational stiffness and the volume of the workspace possess the same changing trends. One can single-objective optimize the z translational stiffness independently without taking into consideration the volume of the workspace. However, the z translational stiffness shows a reversing changing trend with respect to the x and y directional stiffness. Thus, in the case where one optimizes the z translational stiffness, the x and y directional stiffness get affected. In practical applications, engineers need to determine which factor is more critical to them in order to select one factor to perform the optimization. As an example, through employing the Pareto front theory and also via employing the differential evolution algorithm by transforming the multi-objective optimization issue to the single-objective optimization issue, respectively, the summation of the x and y translation and the volume of the workspace for the robotic mechanism are being optimized at the same time. The results indicate the kinematic performances are able to be improved after performing the optimization. Furthermore, it is observed that the multi-objective optimization possesses multiple solutions in the Pareto front, to which they are non-dominate, and engineers are able to select one solution based on their specific requirements and preferences, while the single-objective optimization possesses only one optimal solution. The mechanism can be potentially used in integrating the exercise-walking machine for stroke patients. As for the future work, the global stiffness will be considered. Due to the space limitation, some comparison studies will be conducted in the next step. In addition, the proposed parallel robotic based assisted machine will be prototyped according to the optimized results and the patients will try it out in order to further test the suitability of the proposed machine.

ACKNOWLEDGMENT

The authors would like to thank the financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada Research Chairs (CRC) program.

REFERENCES

[1] Bohez, E. (2002). Five-axis milling machine tool kinematic chain design and analysis. *International Journal of Machine Tools & Manufacture*, 42, 505-520.

[2] Tsai, L., Joshi, S. (2002). Kinematic analysis of 3-DOF position mechanisms for use in hybrid kinematic machines. *Journal of Mechanical Design*, 124 (2), 245-253.

[3] Zhang, D., Bi, Z., Li, B. (2009). Design and kinetostatic analysis of a new parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 25 (4-5), 782-791.

[4] Castelli, G., Ottaviano, E. (2009). Modeling and simulation of a cable based parallel manipulator as an assisting device. In *Computational Kinematics: Proceedings of the 5th International Workshop on Computational Kinematics*. Springer, 17-24.

[5] Li, Y., Xu, Q. (2005). Kinematic design and dynamic analysis of a medical parallel manipulator for chest compression task. In *IEEE International Conference on Robotics and Biomimetics (ROBIO)*, July 5-9, 2005. IEEE, 693-698.

[6] Plitea, N., Pisla, D., Vaida, C. (2007). On kinematics of a parallel robot used for the minimally invasive surgery. *PAMM - Proceedings in Applied Mathematics and Mechanics*, 7 (1), 4010033-4010034.

[7] Lessard, S., Bigras, P., Bonev, I. (2007). A new medical parallel robot and its static balancing optimization. *Journal of Medical Devices*, 1, 272-278.

[8] Liang, Q., Wu, W., Zhang, D., Wei, B., Sun, W., Wang, Y., Ge, Y. (2015). Design and analysis of a micromechanical three-component force sensor for characterizing and quantifying surface roughness. *Measurement Science Review*, 15 (5), 248-255.

[9] Merlet, J.P. (2006). *Parallel Robots*. Springer.

[10] Zhang, D. (2010). *Parallel Robotic Machine Tools*. Springer.

[11] Bruckmann, T., Mikelsons, L., Brandt, T., Hiller, M., Schramm, D. (2008). Wire robots Part I.: Kinematics, analysis and design. In *Parallel Manipulators: New Developments*. I-Tech Education and Publishing, 109-132.

[12] Gosselin, C. (1990). Stiffness mapping for parallel manipulators. *IEEE Transactions on Robotics and Automation*, 6 (3), 377-382.
[13] Carbone, G. (2011). Stiffness analysis and experimental validation of robotic systems. *Frontiers of Mechanical Engineering*, 6 (2), 182-196.
[14] Zhang, D., Gao, Z. (2012). Forward kinematics, performance analysis, and multi-objective optimization of a bio-inspired parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 28 (4), 484-492.
[15] Zhang, D., Xi, F., Mechefske, C., Lang, S. (2004). Analysis of parallel kinematic machine with kinetostatic modelling method. *Robotics and Computer-Integrated Manufacturing*, 20 (2), 151-165.
[16] Li, J. (2009). *Design of 3-DOF parallel manipulators for micromotion applications*. Master Thesis, University of Ontario Institute of Technology.
[17] Li, B., Wang, Z., Hu, Y. (1999). The stiffness calculation model of the new typed parallel machine tool. *Machine Design*, 3, 14-16.
[18] Gosselin, C., Angeles, J. (1991). A global performance index for the kinematic optimization of robotic manipulators. *Journal of Mechanical Design*, 113 (3), 220-226.
[19] Stamper, R., Tsai, L., Walsh, G. (1997). Optimization of a three DOF translational platform for well-conditioned workspace. In *IEEE International Conference on Robotics and Automation*, April 25, 1997. IEEE, 3250-3255.
[20] Gao, Z., Zhang, D. (2011). Workspace representation and optimization of a novel parallel mechanism with three-degrees-of-freedom. *Sustainability*, 3, 2217-2228.
[21] Masory, O., Wang, J. (1995). Workspace evaluation of Stewart platforms. *Advanced Robotics*, 9 (4), 443-461.
[22] Zhang, D., Gao, Z. (2012). Multi-objective performance optimization of a parallel robotic machine tool. In *IEEE/ASME International Conference on Mechatronics and Embedded Systems and Applications (MESA)*, July 8-10, 2012. IEEE, 154-159.
[23] Coppola, G., Zhang, D., Liu, K.F. (2014). A 6-DOF reconfigurable hybrid parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 30 (2), 99-106.
[24] Chi, Z., Zhang, D., Xia, L., Gao, Z. (2013). Multi-objective optimization of stiffness and workspace for a parallel kinematic machine. *International Journal of Mechanics and Materials in Design*, 9, 281-293.
[25] Vesterstrom, J., Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Congress on Evolutionary Computation (CEC2004)*, June 19-23, 2004. IEEE, vol. 2, 1980-1987.
[26] Zhang, D. (2000). *Kinetostatic analysis and optimization of parallel and hybrid architectures for machine tools.* Ph.D. thesis, Laval University, Quebec, Canada.
[27] Talbi, E.G. (2009). *Metaheuristics: From Design to Implementation*. John Wiley & Sons.
[28] Rahnamayan, S. (2007). *Opposition-based differential evolution*. Ph.D. thesis, University of Waterloo, Ontario, Canada.

Received November 09, 2016.
Accepted March 31, 2017.

MEASUREMENT SCIENCE REVIEW, **17**, (2017), No. 2, 93-99

DE GRUYTER OPEN

MEASUREMENT SCIENCE REVIEW

Journal homepage: Http://www.degruyter.com/view/j/msr

ISSN 1335-8871

The Impact of the Implementation of Edge Detection Methods on the Accuracy of Automatic Voltage Reading

Kamil Sidor[1], Anna Szlachta[2]

[1]*Rzeszow University of Technology, al. Powstancow Warszawy 12, 35-959, Rzeszow, Poland*
[2]*Department of Metrology and Diagnostic Systems, Faculty of Electrical and Computer Engineering, Rzeszow University of Technology, W. Pola street, No.2, 35-959, Rzeszow, Poland*

The article presents the impact of the edge detection method in the image analysis on the reading accuracy of the measured value. In order to ensure the automatic reading of the measured value by an analog meter, a standard webcam and the LabVIEW programme were applied. NI Vision Development tools were used. The Hough transform was used to detect the indicator. The programme output was compared during the application of several methods of edge detection. Those included: the Prewitt operator, the Roberts cross, the Sobel operator and the Canny edge detector. The image analysis was made for an analog meter indicator with the above-mentioned methods, and the results of that analysis were compared with each other and presented.

Keywords: Image analysis, Hough transform, edge detection, LabVIEW environment.

1. INTRODUCTION

In everyday life, we have a wide array of skills at our disposal. Among those most important, if not the most important, is the ability to analyse the surrounding world based on the information received through the eyes. The received image is processed by the brain which is a sort of processor processing the image at a very high level. Similar processes take place during a computer image analysis. The main difference consists in the ability to use the full information. Therefore, one may consider the development of machine vision systems as one of the most important tasks of the contemporary computer science. In recent years, vision systems have been gaining an increasing popularity. This results from the fact that they find application in a growing number of areas. They are used in the supervision of industrial processes as well as to control traffic. Processed images are also used in the analysis of satellite photographs of Earth's surface. Computer image analysis allows for obtaining information which otherwise would be impossible to discern with the unaided eye. This allows the detection of lesions in human internal organs and therefore finds application in medicine. Computer techniques of image processing are also used for artistic purposes, including the creation of music videos. Image analysis may also be applicable in metrology, where it may help in taking measurements [1]. An image can be used as a source of information once it is processed in a digital format.

The problem related to determining the position and orientation of a straight line in images concerns many different areas of image processing and computer vision. These may be straight lines in different natural sites, landmarks and complex facilities. Often, they may be identified as various combinations of the linear function.

The detection of a straight line is used, among other things, for finding roads and paths in air photos [2], [3]. In addition to this, detection of a straight line in real time may also be used for finding edges, e.g. for operating a vehicle or a robot [4].

Other applications for line detection are used in the analysis of fingerprints [5]. These applications represent the importance of the problem of line detection in various areas of our lives. Several approaches have been proposed to solve the problem of line detection. The most common solution is the use of the Hough transform.

2. THE WORK STATION FOR MACHINE READING OF THE MEASURED VALUE ON ANALOG METER

In the Department of Metrology and Diagnostic Systems of Rzeszow University of Technology, a work station was established to enable machine reading of the measured value on an analog meter [6].

An analog meter does not have any measurement interface and therefore to connect it to the measuring system, PLEOMAX PWC-3900 Pleo Cam II camera was used. The image recorded using the webcam was saved in a graphic file and then processed by the LabVIEW graphic programme environment [7]. The application detects the position of the indicator on the analog meter and defines the

DOI: 10.1515/msr-2017-0012

measurement range on the basis of the clamp analysis. A photograph of the operating system is shown in Fig.1.

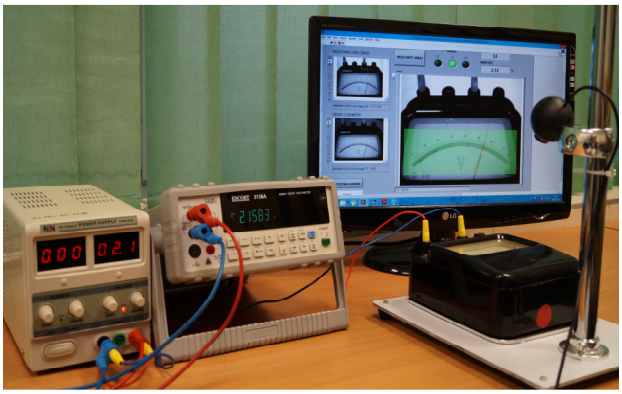


Fig.1. Laboratory work station for the automatic determination of the value measured using an analog meter.

The tool used for research is an analog electromagnetic voltmeter made by Era with the measuring range of: 1,5 V, 3 V, 7,5 V. The developed application works with a meter in the abovementioned ranges. The application will operate correctly provided that the webcam is placed directly over the analog meter. To enable the identification of a measuring range, the camera is placed at a constant angle.

The registration programme cooperates with a webcam, and the image from the webcam is stored in the computer memory.

Fig.2. represents a block diagram of the application reading out of the image from the camera connected to the measuring system.

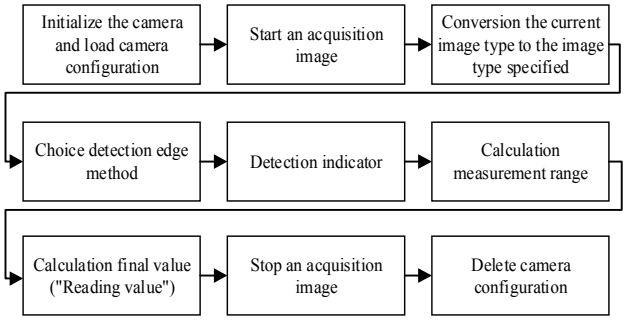


Fig.2. The block diagram of the application.

Fig.3. presents the view of the programme interface window where the correct detection of the indicator occurred.

The highlighted green area determines the part of the image where a straight line was detected corresponding to the indicator coordinates.

The detection of the indicator position in the analysed image is only one of the stages of the determination of the value measured by the analog meter. The detection of the meter's measuring range is equally important. The developed programme assumed that the "+" clamp of the analog meter is connected properly, i.e. it always remains in the same position. In order to specify the measuring range, the programme has to analyse the position of the second clamp.

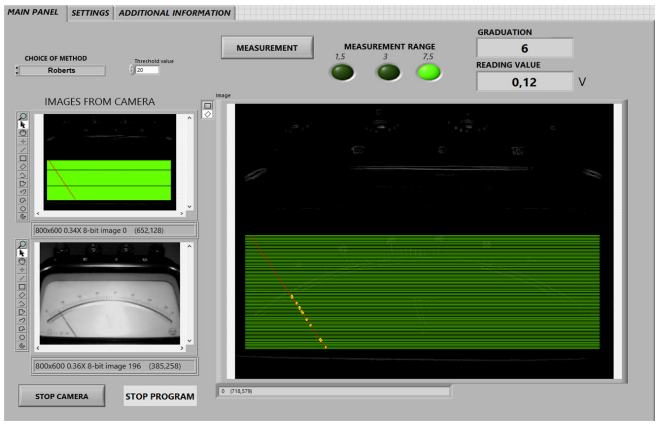


Fig.3. Main page of the application reading the value.

The template matching method was applied, which is used to identify the characteristic features of an image. It consists in comparing the image with a template. The developed algorithm has three templates corresponding to all possible voltmeter clamps. The identification of the measuring range takes place based on the analysis of the x coordinate of the second consecutive clamp. The comparative method results in a probability function. The method's operation is based on two matrices. The first one is a template matrix. It is a result of saving the template in a digital form. The second matrix is a digital mapping of the analysed image. The search is performed by comparing the values of the template with the values in the image. It is required that the probability measure be defined, because it is rarely the case that the template and image are matched perfectly.

The NI Vision Development tool set includes a special programme for creating patterns called NI Vision Template Editor. It enables the creation of one's own templates and the edition of those created earlier.

3. HOUGH TRANSFORM

In order to determine the position of the indicator, the Hough transform [8], [9] was used. A straight line may be represented in the Hough space as an angle of inclination θ and a distance from the beginning of the coordinate system, described as ρ (Fig.4.).

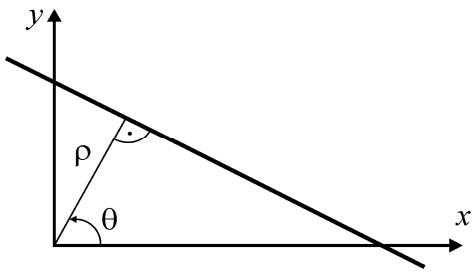


Fig.4. Straight line parameters [10].

The straight line determined in this way is described by the formula [10], [11]:

$$\rho = x\cos(\theta) + y\sin(\theta) \quad . \tag{1}$$

Parameters θ and ρ make a coordinate system (θ, ρ) where the straight line is represented as a point (Fig.4.). The coordinate system of this type is called the *Hough space* [10], [11].

4. METHODS OF EDGE DETECTION

To improve the accuracy of the Hough transform, additional methods of edge detection were implemented. The aim of such detection is marking points of a digital image in which the luminance rapidly changes. There are many ways of edge detection, which fall in two categories:

- methods using the analysis of the first-order image derivative (gradient methods) are currently the most popular; they operate by searching the maxima and minima of the first image derivative, and are also called operators or filters;
- methods using the analysis of the second-order derivative in order to detect the edge; usually with the use of zero crossings of the Laplacian or a non-linear differential expression.

Developed in the LabVIEW environment, the application for edge detection employs only the methods belonging to the former category:

- Roberts cross,
- Sobel operator,
- Prewitt operator,
- Canny edge detector.

They have been applied in the developed application in order to increase the efficiency and accuracy of the reading of a value measured by an analog meter, considering that the indicator is thin and shiny.

A. Roberts cross

The Roberts cross is one of the simplest operators. It is highly sensitive to local image noise and thus it is not used in the images with a high level of noise due to significant errors in detection. The operating principle is based on the calculation of the differences in luminance of pixels arranged side by side diagonally and then after adding their absolute values. The result of the calculation of differences is expressed as a convolution matrix with 2x2 kernels. The resulting image of the edge arises after the calculation of module differences from the corresponding elements of the matrix [12]:

$$g(i,j) = \begin{cases} 0, \text{ dla } f(i,j) < T \\ 1, \text{ dla } f(i,j) \geq T \end{cases} . \tag{2}$$

B. Sobel operator

The Sobel operator belongs to the group of simple methods of edge detection. Depending on the factors defining filtering kernels, vertical, horizontal or diagonal edges are detected. The Sobel operator may be expressed as a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the operator is the corresponding gradient.

It uses two filters: vertical (S_y) and horizontal (S_x). The S_y component determines the gradient value for columns, whereas S_x for rows. Kernels are expressed by the equation:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & -1 \end{bmatrix} . \tag{3}$$

Then, based on the formulae, the values of the edges are determined:

$$G_{mag} = \sqrt{(S_x)^2 + (S_y)^2} , \tag{4}$$

$$G_{dir} = \arctan\left(\frac{S_y}{S_x}\right) . \tag{5}$$

The Sobel operator is much less sensitive to noise than the Roberts cross, but the edges of the obtained image are of different intensity [13].

C. Prewitt operator

This operator makes it possible to detect vertical, horizontal and diagonal edges. It enables the calculation of approximations of the derivatives with the use of two 3×3 kernels which are convolved with the original two-dimensional image. The matrices are asymmetrical in relation to the direction of the detected edge. The set of matrices allows to specify the direction from 0° to 315°.

This method is also known as a formula matching [the kernel] to the actual image. Examples of kernels for angles from 0° to 45° are presented in the formula:

$$P_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, P_y = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} . \tag{6}$$

The Prewitt algorithm is a good algorithm for determining the size and the orientation of the edge [14].

D. Canny's method

This method was developed in 1986 by John Canny. The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

The algorithm consists of 4 main stages. The first step involves removing noise with the use of a Gaussian filter.

As a result, the obtained image is slightly blurred, without any noises.

The next stage features finding the intensity of the image gradient. Considering the fact that edges can be directed in different directions, the algorithm uses four filters to detect the horizontal, vertical, and diagonal edges. Edge operators (e.g. Sobel operator) shall calculate the value of the first derivative of the horizontal and vertical directions. The angle of the edge detection is rounded to four cases representing the horizontal, the vertical and two diagonals (e.g. 0, 45, 90, 135°). The obtained contours are characterised by a certain width and it is not the width of one pixel.

The third step consists in the elimination of non-maximal values of pixel intensity. Pixels constituting the orthogonal cross-section of the contour are analysed, and those lacking maximum intensity values are turned off. The last step includes thresholding aimed at removing unnecessary edges with inclinations below the set threshold.

Canny's algorithm can be used in many different environments. Its parameters can be modified so as to enable the algorithm to recognise edges of various features [15].

5. EDGE DETECTION IN THE IMAGE

Fig.5. presents a block diagram representing edge detection with the following methods: the Roberts cross, the Sobel operator and the Prewitt operator.

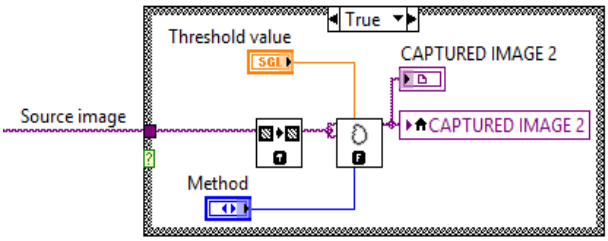


Fig.5. A block diagram representing edge detection with the Roberts cross, the Sobel operator and the Prewitt operator.

For this reason, IMAQ EdgeDetection function was used as it allows the selection of the appropriate method for edge detection [16]. In addition to this, IMAQ EdgeDetection function helps to set thresholding values. It specifies the minimum value of the pixel that should appear on the resulting image. Therefore, it is possible to select edges for display.

Fig.6. presents the list of methods available. The selection is carried out by choosing the desired method from the drop-down menu.

Fig.6. The choice of a method for image edge detection.

Edge detection using Canny's algorithm has been carried out in accordance with the block diagram shown in Fig.7.

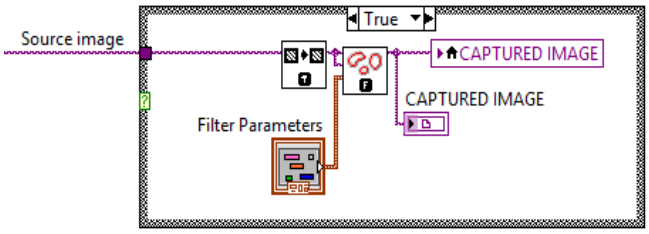


Fig.7. Block diagram describing edge detection using Canny's method.

The main element of the diagram above is the IMAQ Canny Edge Detection VI function which detects edges following Canny's method. There are 4 input parameters to this function. The first one of these is the *Sigma* parameter. It is the sigma of a Gaussian smoothing filter before edge detection. HThresh (High Threshold) defines the upper percentage of pixel values in the image from which the edge detection algorithm chooses the seed or starting point of an edge segment. It takes values from 0 to 1. LThres (Low Threshold) is multiplied by the HThresh value to define a lower threshold for all the pixels in an edge segment. The *HThresh* and *LThres* parameters determine the upper and lower limit values of thresholding. The last parameter, *WindowSize,* determines the size of the filter mask of the Gaussian filter. Fig.8. presents the window enabling the specification of the parameters mentioned above.

Fig.8. Window showing function parameters in IMAQ Canny Edge Detection [16].

6. EXPERIMENTAL STUDIES

During the work with the developed application it has been noticed that the results of the measurements are influenced by the method of edge detection and therefore the application was tested with the use of various edge detection methods.

The voltmeter was divided into 15 voltage values so as to verify the effectiveness of the reading in the broadest possible range of indicator deflection. For each selected voltage value and for each method, automatic measurement

of voltage value was made based on the analysis of the registered image.

During analysis, each from the chosen methods was used in order to eliminate noise from image threshold. Selected were four threshold values, i.e. 5, 10, 15, 20 (Fig.5.). The experiment was repeated in each measuring point and for each method and for each threshold value. Table 1. shows the results of the studies carried out for the three methods: the Roberts cross, the Sobel operator and the Prewitt operator.

Based on the data presented in Table 1., it can be concluded that the effectiveness of the reading depends on the method used as well as on threshold values for each method.

Table 1. The results of the tests for the Roberts, Sobel and Prewitt methods.

No.	User's reading	Automatic reading with a division into methods and threshold values											
		Roberts cross				Sobel				Prewitt			
		5	10	15	20	5	10	15	20	5	10	15	20
1.	9	10	10	9	9	10	8	8	9	8	10	10	9
2.	13	12	12	13	13	12	12	13	13	12	13	13	13
3.	18	17	17	18	18	17	17	17	18	17	17	18	18
4.	23	24	24	23	23	23	23	23	23	23	22	23	23
5.	27	28	28	27	27	26	26	26	27	28	26	26	27
6.	33	32	33	33	33	34	33	33	33	34	33	33	33
7.	38	38	38	38	38	38	38	38	38	38	38	38	38
8.	42	42	42	42	42	41	42	42	42	41	42	42	42
9.	47	47	47	47	47	46	47	47	47	46	47	47	47
10.	51	51	51	51	51	51	51	51	51	51	51	51	51
11.	55	55	55	55	55	55	55	55	55	55	55	55	55
12.	60	61	60	60	60	60	60	60	60	60	61	60	60
13.	64	64	64	64	64	64	64	64	64	64	64	64	64
14.	69	69	69	69	69	69	69	69	69	69	69	69	69
15.	73	73	73	73	73	72	72	73	73	73	73	73	73

Table 2. presents the percentage values of the effectiveness of reading values measured by an analog meter. The effectiveness depends to a large extent on the parameter describing the threshold values.

Table 2. A comparison of the reading effectiveness using various methods.

Threshold values / Method	5	10	15	20
Roberts cross	53.3%	73.3%	100%	100%
Sobel operator	46.7%	66.7%	80%	100%
Prewitt operator	53.3%	66.7%	86.7%	100%

For the value of 20, the performance proved to be 100 % for all verified methods. Also for threshold value equalling 15, the Roberts' method proved to be faultless. Taking into account all tested threshold values, the Roberts cross showed the best performance.

In addition to this, tests using Canny's method (Table 3.) were carried out.

As in the case of the three previous methods, tests were carried out for the same fifteen voltage values. Threshold values (*HThresh* and *LThres*) have been selected based on tests. Three ranges of thresholding were proposed, yielding the best results. Table 3. presents the results of the conducted tests.

The research has proved that Canny's method has not achieved the expected results as the results of this method gave the worst reading results of the measured value. Table 4. presents the percentage values of the effectiveness of the reading for Canny's method.

The value range of L95 H95 thresholds proved to be the most effective. The effectiveness of this was merely 60 %, which is too low to be considered reliable. Compared with previous methods, Canny's method had fairly low results.

Table 3. The results of the studies conducted for Canny's method.

No.	User's reading	Thresholding ranges (L - low threshold, H - high threshold)		
		L90 H90	L90 H95	L95 H95
1.	9	9	8	9
2.	13	12	12	12
3.	18	17	18	18
4.	23	23	23	23
5.	27	27	28	28
6.	33	33	33	33
7.	38	39	39	39
8.	42	43	42	42
9.	47	47	47	48
10.	51	51	52	51
11.	55	55	57	57
12.	60	62	61	61
13.	64	65	64	64
14.	69	69	69	69
15.	73	74	74	73

Table 4. The results of the percentage reading using Canny's method.

Thresholding ranges / Method	L90 H90	L90 H95	L95 H95
Canny	53.33%	46.67%	60.00%

7. SUMMARY

The work on the development of systems for image analysis gives increasingly good results and thus is applicable in a growing number of areas. By analysing images, computers take decisions and do not need human control. To a large extent, it facilitates work when computers do some work better than a man. This is the case of image analysis where the computer is able to detect details invisible to the human eye.

The laboratory stand based on the image analysis from camera makes automatic reading of the value measured using an analog meter possible. Application of Hough transform enables to determine the indicator position and - in consequence - amount of graduation scale on the analog meter. The calculation of amount of graduation scale into the measured value was executed using the Template Matching method. The template was elaborated which allows to locate the measurement range. One of the factors which provide the correct program operation is appropriate lighting. It was analysed and discussed in the paper [6]. In order to improve the obtained measurement results (i.e. image analysis) the methods of edge detection were used.

The authors used IMAQ Canny Edge Detection function and IMAQ EdgeDetection available in the LabVIEW package.

It was expected that the proposed method of edge detection would increase the effectiveness of the automatic reading of the measured value measured by an analog meter. After carrying out a series of tests, it was concluded that the methods using edge detection gave different results.

The best method has proved to be the Roberts cross, which after setting the correct threshold value gave the most accurate results of reading of the measured value reading, i.e. 100 % effectiveness.

Canny's method has proved to be the least efficient. Despite being the most complex in terms of computing, the method has not proved to be the best. The reason for this may be the width of the edges detected. Too narrow an edge describing an indicator could be incorrectly interpreted by the Hough transform.

REFERENCES

[1] Tadeusiewicz, R., Korohoda, P. (1997). *Komputerowa analiza i przetwarzanie obrazów*. Kraków, Poland: Wydawnictwo Fundacji Postępu Telekomunikacji. (in Polish)

[2] Skingley, J., Rye, A.J. (1987). The Hough transform applied to SAR images for thin line detection. *Pattern Recognition Letters*, 6 (1), 61-67.

[3] Aghajan, H.K., Kailath, T. (1994). SLIDE: Subspace-based line detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16 (11), 1057-1073.

[4] Inigo, R.M., McVey, E.S., Berger, B.J., Wirtz, M.J. (1984). Machine vision applied to vehicle guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6 (7), 820-826.

[5] Lin, W.C., Dubes, R.C. (1983). A review of ridge counting in dermatoglyphics. *Pattern Recognition*, 16, 1-8.

[6] Szlachta, A., Sidor, K. (2014). A laboratory stand for automatic reading of the value measured by an analog meter. *Measurement Automation Monitoring*, 60 (8), 548-551.

[7] Galabov, K.S. (2012). Application of the NI LabVIEW in the verification of analog measuring instruments. *Системи обробки інформцій (Information Processing Systems)*, 1 (99), 149-151.

[8] Hough, P. (1962). *Method and means for recognizing complex patterns*, U.S. patent 3069654.

[9] Illingworth, J., Kittler, J. (1998). A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44, 87-116.

[10] Duda, R.O., Hart, P.E. (1972). Use of Hough transform to detect curves and lines in pictures. *Communications of the ACM*, 15 (1), 11-15.

[11] Aggarwal, N., Karl, W.C. (2006). Line detection in images through regularized Hough transform. *IEEE Transactions on Image Processing*, 15 (3), 582-591.

[12] Sypniewski, G. (2015). Metody wizyjne w automatyzacji spawania. *Przegląd Spawalnictwa*, 87 (1), 41-49. (in Polish)

[13] Zawieska, D. (2010). Wybrane operatory detekcji w automatyzacji dopasowywania obrazów cyfrowych bliskiego zasięgu. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, 21, 481-491. (in Polish)

[14] Prewitt, J.M.S., Mendelson, M.L. (1966). The analysis of cell images. *Annals of the New York Academy of Sciences*, 128 (3), 1035-1053.

[15] Watkins, D.C., Sadun, A., Marenka, S. (1995). *Nowoczesne metody przetwarzania obrazu*. Warszawa, Poland: Wydawnictwo Naukowo - Techniczne. (in Polish)

[16] National Instruments (NI). *Image Acquisition and Processing*. Manual NI.

Received December 23, 2016.
Accepted April 10, 2017.