Title
Pulsed type ultrasonic anemometer based on a double FFT procedure

Authors
Tang, Shan
Federspiel, Clifford C
Auslander, David M

Publication Date
2003-10-01

Peer reviewed

Pulsed Type Ultrasonic Anemometer Based on a Double FFT Procedure

Shan Tang
Dept. of Mechanical Engineering,
Univ. of California, Berkeley, CA
shtang@me.berkeley.edu

Clifford C. Federspiel
Center for the Built Environment,
Univ. of California, Berkeley, CA
cliff_f@unclink.berkeley.edu

David M. Auslander
Dept. of Mechanical Engineering,
Univ. of California, Berkeley, CA
dma@me.berkeley.edu

ABSTRACT

A signal processing algorithm for a pulsed type ultrasonic anemometer is described. The measurement principle is based on the linear relationship between the inverse transit time difference (TTD) and air velocity. The algorithm uses two phases of Fast Fourier Transforms (FFTs). Computer simulations demonstrate the high performance of this algorithm. Its advantages over other pulsed type algorithm are discussed.

1. INTRODUCTION

In buildings, air velocity is an important parameter. Air velocity affects heat transfer rates, which in turn affects energy efficiency. Air velocity also affects indoor air quality. Air velocities in heating, ventilating and air conditioning (HVAC) systems must be adequate for ventilation, and air velocities in occupied spaces of buildings influence the emission rate of pollutants from building materials and transport of pollutants from one space in a building to another. It also affects the thermal comfort of building occupants because it affects the rate of heat removal from people to the indoor environment.

Although air velocity is important to the operation of buildings, air velocity measurements are not as commonly used in buildings as other types of sensors such as temperature sensors. This is because air velocity sensors are relatively expensive and because existing air velocity sensors do not have the functional performance needed for some applications. In particular, most air velocity sensors only measure velocity at a point, i.e. locally. There are some applications where it would be advantageous to measure the average velocity over an area. These include measurement of velocity in ducts with non-uniform flow, measurement of flow from diffusers, and measurement of flow patterns in open areas of buildings. When averaging is needed today, most velocity measuring devices use an array of point measurements.

Pitot tubes, hot wire anemometers and vortex shedding flow meters are commonly used in buildings. The Pitot tube is a differential pressure probe and is used to measure local flow rate in a duct or pipe. One major problem with Pitot tubes "is the effect of unsteady flow on its reading, and because tubulence is very common in normal industrial flows, the pitot tube will tend to read high" (Roger C. Baker 2000, pp. 428-430). Another problem is when the velocity is low, the difference in pressure is very small, causing the error in the device to be greater than the measurement. Hot-wire anemometers measure the air velocity by detecting the impact of velocity on heat transfer from a heated resistor (Ken Okamoto et al, 1994). The can be adversely affected by contamination such as dust, sediment, and so on. Hot wire anemometers are better than pitot tubes at low velocity and can be used to measure omni-directional velocity, but they also only measure local velocity. Vortex shedding flow meters make use of the principle that the shedding frequency behind a bluff body is proportioinal to the flow rate. They also measure the local value (Roger C. Baker 2000, pp278-281). Compared to the former three, ultrasonic anemometers have a number of attractive characteristics like linear, fast response and lack of moving parts.

Our work is motivated by a need for an air velocity measurement technology that is accurate at low velocity, especially in a building, that can produce inherently averaged readings, and that is not adversely affected by reverberation. Ultrasonic anemometers potentially have this capability.

Two different types of ultrasonic/sonic anemometers have been developed: the pulsed type ultrasonic anemometer determines the velocity by measuring the inverse transit time difference (TTD) of two pulses along a known path, upstream and downstream of the air flow (Barrett et al, 1949). The continuous wave ultrasonic anemometer is based on the same principle as the pulsed type. Instead of pulses, continuous waves of two different frequencies are used in opposite directions. Transit times are derived from the phase differences across the path (Kaimal et al, 1963) and phase-locked loop circuits can be used in this situation (S. E. Larsen et al, 1979). Since it measures velocity using sonic transit time between two points, it produces path-averaged readings.

This paper describes a new algorithm used for pulsed type ultrasonic anemometer. The emphasis is on good accuracy at low velocity with minimal hardware and computational requirements and insensitivity to reverberation. We consider a pulsed type anemometer, instead of a continuous wave one, because of concerns about

0-7803-8133-5/03/$17.00 ©2003 IEEE

reverberation. The next section describes the principle, Section 3 describes the algorithm, while Section 4 shows computer simulations. Section 5 describes how to deal with aliases caused by high velocity.

2. PRINCIPLE OF OPERATION

Fig 1 shows a schematic diagram of the system with a transmitter, an open air path, and a receiver. Subscripts g, s, m, and r refer to generated signal, speaker signal, microphone signal, and received signal, respectively.

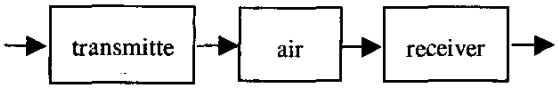

Figure 1. The schematic diagram of Signal Flow

In the frequency domain, the relationship between the generated signal and the received signal in the positive direction (p-superscript refers to the positive direction of air motion) is as follows:

$$Y_r^p(j2\pi f)=H_{mr}^p(j2\pi f)H_{sm}^p(j2\pi f)H_{gs}^p(j2\pi f)X_g^p(j2\pi f) \quad (1)$$

$$H_{sm}^p(j2\pi f)=G_{sm}^p(j2\pi f)\exp[-j2\pi f(T_p+T_h)] \quad (2)$$

where T_p is the delay due to the sum of sound speed and air motion, i.e. $T_p = L/(V_{sound}+V_{air})$ (3)

and T_h is the delay due to the hardware and firmware. L is the distance between the transmitter and receiver. In the negative direction the relationship is as follows:

$$Y_r^n(j2\pi f)=H_{mr}^n(j2\pi f)H_{sm}^n(j2\pi f)H_{gs}^n(j2\pi f)X_g^n(j2\pi f) \quad (4)$$

$$H_{sm}^n(j2\pi f)=G_{sm}^n(j2\pi f)\exp[-j2\pi f(T_n+T_h)] \quad (5)$$

where T_n is the delay due to difference of sound speed and air motion, i.e. $T_n = L/(V_{sound}-V_{air})$ (6)

Assume the following two sets of identical transmitters and receivers:

$$H_{mr}^p(j2\pi f)=H_{mr}^n(j2\pi f) \quad (7)$$

$$G_{sm}^p(j2\pi f)=G_{sm}^n(j2\pi f) \quad (8)$$

$$H_{gs}^p(j2\pi f)=H_{gs}^n(j2\pi f) \quad (9)$$

or possibly the design a transmitter doubles as a receiver:

$$H_{mr}^p(j2\pi f)=H_{gs}^n(j2\pi f) \quad (10)$$

$$G_{sm}^p(j2\pi f)=G_{sm}^n(j2\pi f) \quad (11)$$

$$H_{gs}^p(j2\pi f)=H_{mr}^n(j2\pi f) \quad (12)$$

and that the generated signals in the positive and negative directions are the same. Then:

$$r(f)=\frac{Y_r^n(j2\pi f)}{Y_r^p(j2\pi f)}=e^{-j2\pi f\tau_0} \quad (13)$$

$$\tau_0=T_n-T_p=\frac{L}{V_{sound}-V_{air}}-\frac{L}{V_{sound}+V_{air}}=\frac{2V_{air}L}{V_{sound}^2-V_{air}^2}\approx\frac{2V_{air}L}{V_{sound}^2} \quad (14)$$

hence V_{air} is proportional to τ_0 :

$$V_{air}=\tau_0 V_{sound}^2/2L \quad (15)$$

We can treat $r(f)$ as a function of f with "equivalent frequency" τ_0. By applying FFT on the signal sequence $r(f)$, $R(\tau)$ can be derived. V_{air} is proportional to the value of τ corresponding to the maximum point of $|R(\tau)|$, i.e. τ_0.

3. ALGORITHM DESCRIPTION

Based on Sec. 2, our algorithm has the following steps:

a). Generate the same signals and transit via transmitter in positive/negative directions.

b). Sample the signals at the receivers' sides with a sampling frequency f_s and get the signal sequences $y_r^p(t)$ and $y_r^n(t)$.

c). Apply FFT to $y_r^p(t)$ and $y_r^n(t)$, and get the FFT $Y_r^p(j2\pi f)$ and $Y_r^n(j2\pi f)$ respectively.

d). Divide $Y_r^n(j2\pi f)$ by $Y_r^p(j2\pi f)$ to get $r(f)$, i.e. $r(f)=Y_r^n/Y_r^p$. If $Y_r^p(j2\pi f)$ equals 0 for some f, it should be changed to a very small value to avoid division by zero.

e) Replace $r(f)$ by its normalized value, i.e. $r(f)/|r(f)|$

f). Apply FFT the second time to $r(f)$ and get $R(\tau)$.

g). Search $|R(\tau)|$, the maximum point is at $\tau=|\tau_0|$. Note FFT is intrinsically periodic, we normally use one period including origin or at the side of origin for future calculation. If FFT is centered about y-axis, the sign of τ also reflects the direction of the air velocity, i.e. positive τ means negative velocity, and complementary side is also met.

An intuitive explanation or normalization in step 3d) is that it improves local accuracy near harmonic frequencies where $Y_r^p(j2\pi f)$ and $Y_r^n(j2\pi f)$ are less influenced by the noise. A detailed explanation is beyond the scope of this paper. An alternative method is using a multiple-

window band-pass filter, which will attenuate the contribution from frequencies with low signal to noise ratio. We have found that normalization is equally effective and computationally simpler.

Before the end of the description of the algorithm, we should emphasize three inequalities to guarantee the accuracy and precision.

According to the Sampling Theorem, the sampling frequency must be at least two times the highest frequency in the measured signals, so in b):

$$f_s > 2 f_{HighestCom\ ponent} \text{ of } y_r^p(t) \text{ and } y_r^n(t) \qquad (16)$$

In application, f_s is suggested to be 4 to 5 times or even higher of the highest frequency.

From (13), $r(f) = \frac{Y_r^n(j2\pi f)}{Y_r^p(j2\pi f)}$ is a function of f with "equivalent frequency" τ_0 and "equivalent sampling period", $1/T_{total}$. T_{total} is the total sample time of $y_r^p(t)$ and $y_r^n(t)$. According to the Sampling Theorem:

$$T_{total} > 2|\tau_0|_{\max} \qquad (17)$$

Zeros should be padded in b) if Equ (17) is not satisfied. We limit $r(f)$ derived in e) only from zero to some frequency a little more than the highest harmonic to diminish the influence of high-frequency noise.

The third inequality guarantees sufficient resolution of τ_0. From Equ 13, if we apply second FFT like e), it should be guaranteed that

$$1/f_{\max} < precision \text{ of } \tau_0 \qquad (18)$$

So accordingly zeros should be padded in e) if Equ (18) is not met.

4. COMPUTER SIMULATION

A generated square wave pulse is used to test the efficiency and accuracy of this algorithm.

$$x_g = A \cdot sqr(2\pi f_0 t) \cdot [u(t) - u(t - t_{dur})] \qquad (19)$$

where dur is the abbreviation of duration, $sqr(\cdot)$ is the square wave function, and $u(t)$ is the unit step function. The configuration used to test the algorithm is listed below:

$f_0 = 40.0(\text{KHz})$, $f_s = 40 f_0$, $t_{dur} = 6/f_0$, $A = 12$, $V_{sound} = 340.0(m/s)$, $L = 3m$.

The expected air velocity is $\text{in}[-2.0, 2.0]$, precision $= 0.2\% FSR = 0.004 m/s$.

For simplicity, it is assumed that

$$H_{mr}^p(j2\pi f) G_{sm}^p(j2\pi f) H_{gs}^p(j2\pi f) = 1 \quad \text{and}$$
$$H_{mr}^n(j2\pi f) G_{sm}^n(j2\pi f) H_{gs}^n(j2\pi f) = 1$$

We just take it for granted that frequency component above $9 f_0$ is ignored so f_s satisfy inequality (16). To simulate the real world, white noise with zero mean and unit variance is added at the receivers' sides. That means the signal-to-noise ratio equals 21.6DB.

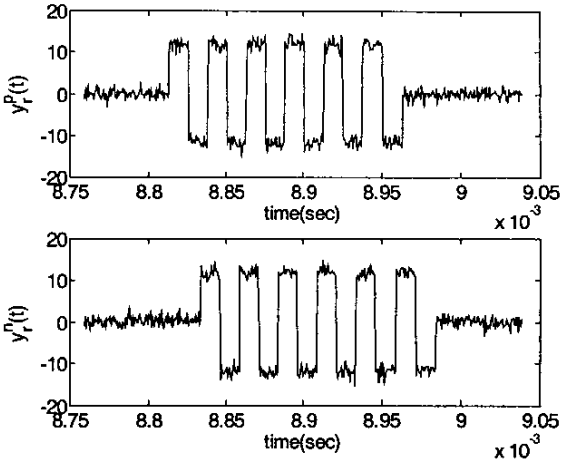

Figure 2: $y_r^p(t)$ and $y_r^n(t)$

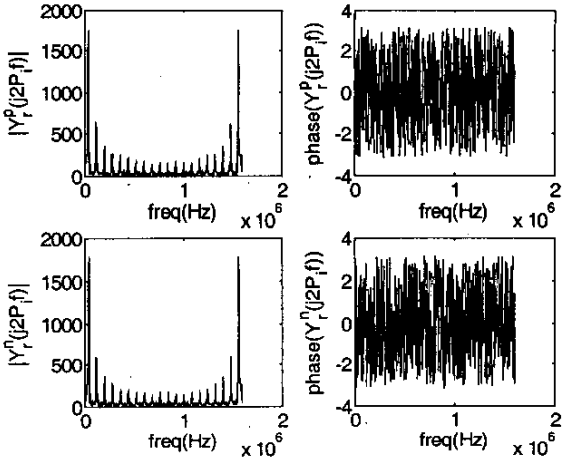

Figure 3 Magnitude and phase of $Y_r^p(j2\pi f)$ and $Y_r^p(j2\pi f)$

Fig 2 to Fig 6 shows the corresponding figures of the algorithm in Section 3 when simulating a velocity of

0.4m/s. Fig 2 shows the sample sequences y_r^p and y_r^n after step 3b), which have a transit time difference within [-6.92e-5, 6.92e-5] according to Equ (14). Y_r^p and Y_r^n are shown in Fig 3 where $\Delta f = 1/T_{total}$. Fig 4 shows the outcome of step 3d). Fig 5 shows the results of step 3e). From Fig 5, we can find the maximum point at $|\tau_0|$, as stated in step 3f). Note due to the periodicity of the FFT, considering that Fig 5 only includes a positive period, maximum point at the right half side reflects positive velocity, and left half side reflects negative one.

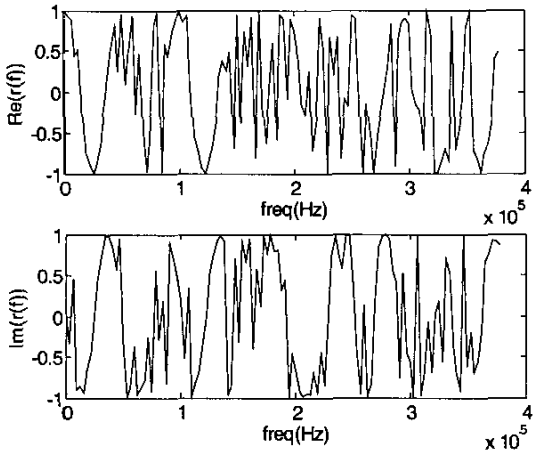

Figure 4: f versus $\mathrm{Re}(r(f))$ and $\mathrm{Im}(r(f))$

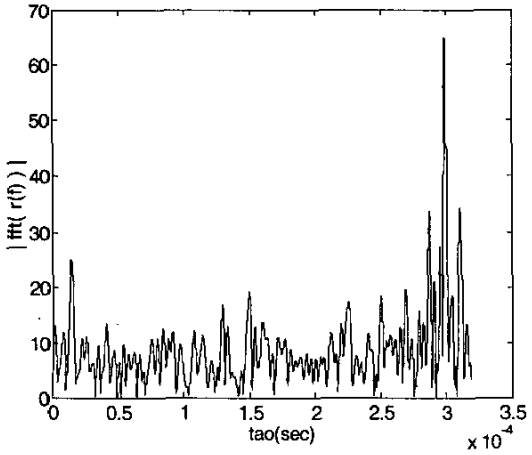

Figure 5: τ versus abs($FFT(r(f))$)

The simulation gives us a result 0.3951m/s, or error 0.0049m/s, or relative error 1.2%. This is a little more than the target precision of 0.004 m/s.

Table 1. Algorithm Performance Evaluation

Simulated Vel. (m/s)	-2.000	-1.200	-0.400	-0.200
Avg.	-1.9998	-1.2051	-0.3984	-0.2038
Median	-2.0008	-1.2053	-0.3985	-0.2038
Max.	-1.9936	-1.2012	-0.3947	-0.1972
Min.	-2.0042	-1.2091	-0.4026	-0.2077
Std Dev	0.0025	0.0018	0.0022	0.0027

-0.170	-0.110	-0.050	-0.03	-0.01
-0.1806	-0.1088	-0.0611	-0.0352	-0.0126
-0.1806	-0.1088	-0.0604	-0.0348	-0.0120
-0.1772	-0.1061	-0.0576	-0.0301	-0.0083
-0.1840	-0.1125	-0.0655	-0.0410	-0.0162
0.0022	0.0018	0.0022	0.0028	0.0023

0.000	0.010	0.030	0.070	0.130
-0.0001	0.0116	0.0359	0.0599	0.1321
-0.0002	0.0120	0.0357	0.0600	0.1321
0.0090	0.0154	0.0418	0.0651	0.1366
-0.0049	0.0083	0.0312	0.0561	0.1257
0.0029	0.0020	0.0025	0.0025	0.0031

0.190	0.200	0.600	1.400	2.000
0.1796	0.2043	0.6015	1.3966	1.9993
0.1797	0.2043	0.6017	1.3968	1.9997
0.1855	0.2085	0.6051	1.4013	2.0019
0.1742	0.2002	0.5972	1.3919	1.9948
0.0027	0.0022	0.0020	0.0024	0.0019

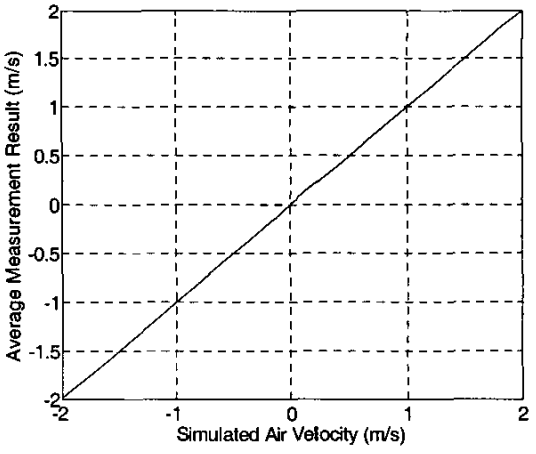

Figure 6. Simulated Air Velocity vs. Average Measurement Result

We simulated velocities at 0.00m/s, from 0.01m/s to 0.19m/s in increments of 0.02m/s, from 0.20m/s to 2.00m/s in increments of 0.20m/s, and vice versa with respect to negative velocities, with 20 simulations at each

case. Table 1 shows the statistics for some simulated velocity. We do not include all statistics there for terseness.

When the air velocity is small, the simulated results will have comparatively large relative errors that are not linearly dependent on the air velocity. That can be explained by error cause by discretization.

5. DIAGNOSTIC

The algorithms described in Section 3 will fail if the velocity is too high or if the signals are very noisy. In this section we describe two diagnostics for detecting a problem caused either by high velocity or excessive noise.

In step b of Section 3, we only sample when signals are expected to arrive at the receivers' sides hence the sampling window depends on the maximum expected velocity. In simulations, the sampling time is extended slightly beyond the point required by the maximum expected velocity. However when V_{air} is much greater than the maximum, we will not receive the entire chirp signal, so the received signal sequence upstream is no longer a time-delay (or time-ahead) version of that downstream. In this case, the algorithm has the potential to fail. To prevent this problem, two diagnostics are used.

1). Since the length of the transmitted sequence is already known, we require that a large fraction (such as 95%) of transmitted chirp signal be received. If the velocity is too high, some of the chirp signal will have passed the receiver before the sampling is ceased. Fig 7 shows what will be received when $V_{air} = 4m/s$.

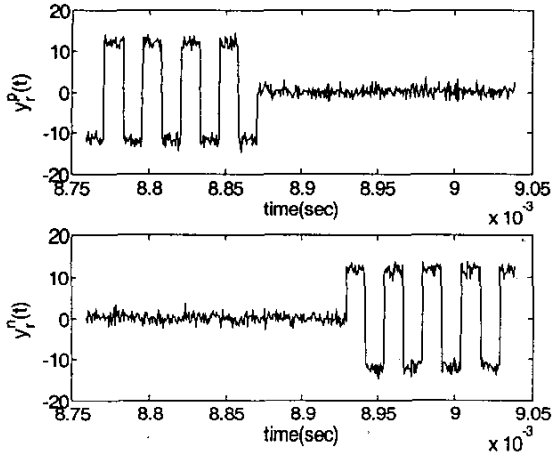

Figure 7. The Effective Length is Used as One Criterion to Avoid Alias Due to Measurement of a Velocity Which is Greater than the Measurement Bound

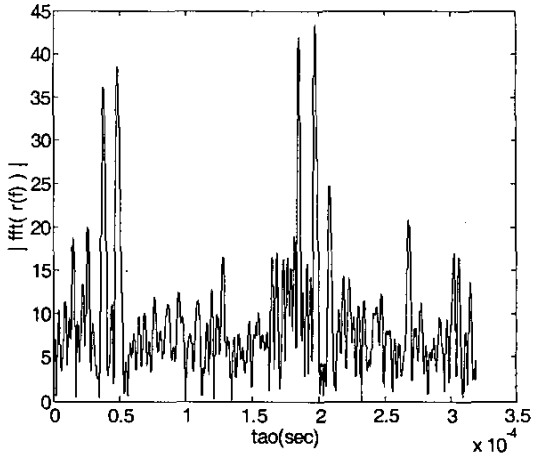

Figure 8. Detect More than One Peaks

2). When the air velocity is only a little greater than the maximum, the criterion 1) will be unable to detect the problem. However, under this condition, there will be two phenomena happen.

2.1) The algorithm gives us a result beyond the configured measurement bound, so we can directly say out of bound.

2.2) We will not get a single large peak as is shown in Fig 5. We check whether or not the second-largest peak exceeds a significant fraction of the largest peak (e.g., 85%). If this occurs, flag the result as unreliable. Fig 8 shows an example with multiple peaks. This diagnose will also detect errors caused by low signal to noise ratio.

We simulated velocities from -7.0m/s to -3.0m/s in increments of 1.0m/s, from -3.0m/s to 3,0m/s in increments of 0.2m/s, and from 3.0m/s to 7,0m/s in increments of 1.0m/s, with 20 simulations at each case. Table 2 shows the number of diagnostics failures for each simulated velocity.

Table 2. Diagnostics Failures Per Twenty Measurements

Simulated Vel. (m/s)	-7.0	-6.0	-5.0	-4.0	-3.0
# of fails	0	0	0	0	0

-2.8	-2.6	-2.4	-2.2	-2.0 to 2.0	2.2	2.4
0	1	0	0	1	0	0

2.6	2.8	3.0	4.0	5.0	6.0	7.0
0	1	0	0	0	0	0

There are many parameters set influencing Table 2, till now we haven't found one suitable for all cases. Table 2 gives us comparative correct results.

6. DISCUSSION

The resolution of our algorithm is dependent on the sampling rate, which is also the case for other algorithms. For example, cross-covariance processing (F. Claveau et al 1997), is based on the principle that time-delayed signal has a cross-covariance relationship with the original one. The time difference is the maximum point of the FFT of the cross-covariance. Since the precision of the time difference is equal to sampling period, if the same configuration in Section 4 is used in the cross-covariance method, by Equ (15), the sampling period should be

$$2(Vair)_{precision} L / V_{sound}^2 ,$$

i.e. sampling frequency is

$$V_{sound}^2 / \left(2(Vair)_{precision} L\right) = 340^2 / (2 \cdot 0.2\% \cdot 2 \cdot 3) = 4.8MHz$$

which is much higher than our algorithm. The reason is stated below.

Another pulsed type method, direct measurement of the transit time difference (and/or average) (A Alberigi Quaranta et al 1985), needs a high frequency clock to achieve the precision like the cross-covariance method. It is also difficult to set a sharp edged signal to improve its accuracy.

For our algorithm we can reduce the sampling frequency required to achieve a specified resolution by padding zeros to the first and/or second FFT so long as inequality (17) and (18) are satisfied. In moderation, zero-padding improves resolution without having a negative impact on performance. However, excessive padding causes the algorithm to fail.

We have found that the shape of the chirp signal has an impact on the performance of the algorithm. In general we have found that signals with a high-frequency component improve accuracy. For example, square waves produce better results than sine waves. This property stimulates us to search for way to optimize the design of the chirp signal for future research.

7. CONCULSION

An algorithm used in pulsed type ultrasonic/sonic anemometers was presented and compared with other pulsed type ones. Computer simulation results show that it is a promising technology in anemometers and gives high accuracy and precision. It also has the very useful property that the form of generated signal will improve the measurement, as is under consideration for future. Incorporating the implementation on PC-based hardware is in progress.

8. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0088648.

9. REFERENCES

[1] Earl W. Barrett and Verner E. Suomi, 1949, "Preliminary report on temperature measurement by sonic means", Journal of the Atmospheric Sciences, Vol. 6, No. 4, pp. 273–276.

[2] J. C. Kaimal and J. A. Businger, 1963, "A continuous wave sonic anemometer-thermometer", Journal of Applied Meteorology, Vol. 2, pp 156-164.

[3] S. E. Larsen, F. W. Weller and J. A. Businger, 1979, "A phase-locked loop continuous wave sonic anemometer-thermometer", Journal of Applied Meterology, Vol. 18, pp 562-568.

[4] A. Alberigi Quaranta, G. C. Aprilesi, G. De. Cicco and A. Taroni, 1985, "A microprocessor based, three axes, ultrasonic anemometer", J. Phys. E: Sci. Instrum., Vol. 18, pp 384-387.

[5] Ken Okamoto, Tadahiko Ohhashi, Masahiro Asakura and Kenzo Watanabe, 1994, "A digital anemometer", IEEE Transactions on Instrumentation and Measurement, Vol. 43, No. 2, pp 116-120.

[6] Fabien Claveau, Michel Poirier and Denis Gingras, 1997, "FFT-based cross-covariance processing of optical signals for speed and length measurement", [Conference Paper] Statistical Signal and Array Processing, Applications ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, Vol. 5, IEEE, pp 4097-4100.

[7] Rodger E. Ziemer, William H. Tranter and D. Ronald Fannin, 1998, "Signals and systems: continuous and discrete", Prentice-Hall, Inc., 4th Ed

[8] Roger C. Baker, 2000, "Flow measurement handbook: industrial designs, operating principles, performance, and applications", Cambridge University Press, UK

[9] Phillip N. Price, Marc L. Fischer, Ashok J. Gadgil and Richard G. Sextro, 2001, "An algorithm for real-time tomography of gas concentrations, using prior information about spatial derivatives", Atmosphere Environment, Vol. 35, pp2827-2835.