10th IMEKO TC10 Conference on
Technical Diagnostics
Budapest, HUNGARY, 2005, June 09-10

Sensor fault detection in a district heating substation

Kimmo Yliniemi, Jan van Deventer, Jerker Delsing

Department of Computer Science and Electrical Engineering, Division of EISLAB, Luleå University of Technology, Luleå, Sweden

Abstract – This paper describes a method to detect temperature sensor faults in a district-heating substation (DHS). It uses valve position monitoring and variance monitoring to detect sensor faults. Depending on user requirements the method can be customized to either detect faults fast or to have a low false alarm rate and being able to detect smaller noise increases but at the expense of longer detection time. The result is a method that uses a limited amount of computing power, which does not require much computing precision since the only computing done is addition of integers and is therefore suitable for implementation in a microcontroller.

Keywords: sensor fault detection

1. INTRODUCTION

Today more than half of Sweden is heated using district heating (DH). This includes private homes, stores, factories, schools, hospitals and so on [2].

In Sweden most DHS have several electronic temperature sensors. Two for billing and at least two for controlling the DHS, assuming the consumer has one of the common DHS configurations. If any one of these sensors fails then the consumer will either a) feel an indoor temperature change (discomfort) but not necessarily immediately Or b) get faulty bills if one of the temp sensors used for billing fails. Both scenarios are most likely bad for business since DH is advertised as reliable, cheap and a good alternative to direct electric heating. Today the controller and the energy meter are two separate devices that do not communicate. We think that by enabling the devices to communicate and enabling the DHS to communicate with the outside world in combination with fault detection will give a solution to this problem [1-2]. A system for fault detecting DHS can be a money saver for both the consumer and the supplier of DH in the long run.

Problem: Sensor fault may cause consumer discomfort for instance wrong room temperature, wrong tap water temperature or faulty billing.

Suggested solution: A system that can detect faults and report them automatically to either the consumer or the appropriate service provider. This will most likely save time and money for both the supplier and consumer.

The consumer will experience less discomfort in case of a sensor failure because the failure will be detected swiftly and a repairman can perhaps be sent out before it has caused any problem. Both the consumer as well as the supplier of DH can be confident that the billing system works if the sensors in it are fault detected automatically.

Today it can take days before a consumer notices that his DHS is not working properly [1]. If a sensor in the DHS fails several things can happen. For instance room temperature or tap water temperature can change so much so it is detectable by the consumer.

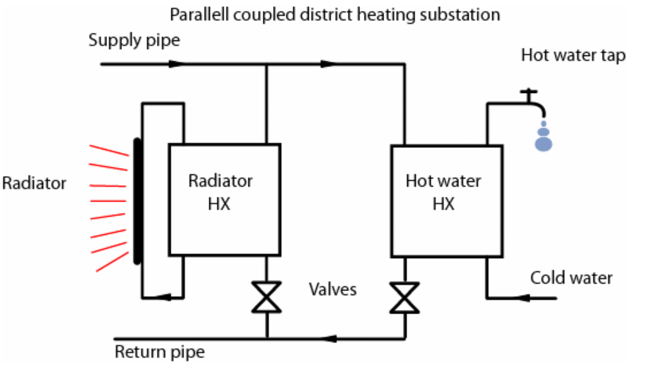

Fig. 1 Layout of typical Swedish parallel coupled DHS

If the consumer does not experience any discomfort from a sensor failure he or she will not call his service provider asking them to come fix things. Instead the equipment will continue malfunctioning until it creates a problem that the consumer can detect. Perhaps when the weather changes, the consumer discovers that radiators still are cold and calls the service provider.

When it comes to the sensors used for billing or energy measurement the consumer will not experience any kind of discomfort if they fail. In Sweden it is common practice to check total energy usage once a year, but this is changing and DH suppliers check the total energy usage more frequently in some cases. They way of checking the energy usage differs also. Some send a person to read the total value from the meter, some have a radio transmitter attached to the energy meter and others have a wire solution. This all depends on how old the DHS is and what company is supplying the DH [1]. They seldom have a built in fault detection. This can of course result in faulty energy measurement during a long period of time. It is only when the meter value is read, the service man can check if the sensors work. Imagine what happens if one sensor fails after the energy reading has been done.

2. METHOD

The DHS is simulated in Simulink [5]. Simulation setup is as follows. The DHS is a parallel-coupled substation without accumulator tank, see fig. 1. All the valves are electronically controlled. The setpoint for tap water is 55 °C. Outdoor temperature (arbitrarily chosen so there will be some heating needed) is set to –15 °C. Cold water temperature is arbitrarily set to 14 °C. In reality it varies with the inside and outside temperature. Hot water tapping is made with a frequency of 1 tapping every 1000 second. The amount of hot water used is randomised [1, 5].

To simulate a faulty temperature sensor a band limited white noise is added to the signal using the predefined block for that in Simulink. The signal is then sampled with 100 Hz sampling frequency and stored on disk. The sensor with the simulated fault is the sensor that measures the return temperature from the tap water heat exchanger. This one is selected because it is a very dynamic temperature compared to the return temperature of the radiator side [1].

The temperature sensors used in DHS are usually of the type PT100 or PT1000. They are resistive and changes resistance depending on temperature. If properly calibrated, they will give fairly accurate measurements. Because of drift they need to be recalibrated now and then [1-3]. Other problems may be wiring. For instance its known to happen that outdoor and indoor temperature sensors have had their wires destroyed when some person has hit a nail through them or cut the cable for some reason. This problem is hard to predict but easy to detect. When the wire is cut the resistance becomes infinite. If the temperature increases, the resistance in the PT element increases. Because of this a higher than normal temperature reading may indicate a cut sensor wire.

If the wire is not entirely cut-off it will probably result in a bad connection. It is possible that this can be detected as an increase in noise in the measurements. Noise can also be introduced by electronics switching on and off causing spikes in the power supply [3].

Summary of typical temperature sensor problems:
Drift (bias change)
Noisy measurement
Mechanical problem, cable cutoff, cable short circuit
Faulty installation

Drift is depending on the environment and the type of sensor. According to [11] the drift for a PT100 sensor is limited to 0.005 C^0 per year in the temperature range 25-150 C^0. Of these faults one of the easiest to detect is probably cable cutoff it can be detected by the temperature going outside the normal range. Short circuit can give very low measurement values (resistance close to zero). Faulty installation can give different symptoms depending on which temperature sensor is installed wrong. If it is a temperature sensor used by the controller it may be the source of unnecessary delays in temperature measurement and cause instability in the control loop [10]. Of these faults we have chosen to study the increase in sensor noise because it is not so easy to detect and it can create controller problems, faulty billing and discomfort for the consumer depending on what sensor has to high noise values. Temperature sensors are also common source for problems with DHS according to [12] and repairmen working in the industry. This is another reason we chose to study the noise on temperature sensors.

To detect the changes in noise we suggest that the signal is filtered in some way to remove the bias and low frequency variations in the process. This is done so we can study the noise. The linear FIR filter can be used because it is always stable, Gaussian noise in– Gaussian noise out, and it can be realised efficiently in hardware [4].

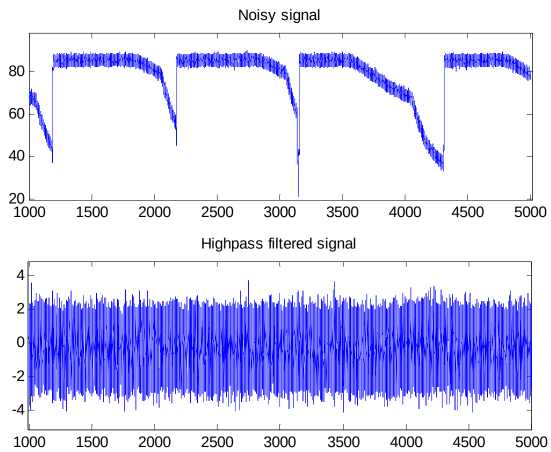

Fig. 2 Highpass filtered signal from the temperature sensor.

The Noisy signal in fig. 1 is the return temperature in °C coming from DHS. The dips that show in the upper graph are caused by hot water usage. The lower graph is the filtered version. To get an estimate of how noisy the signal is we can study the variance. An increase in variance would suggest an increase in noise. Instead of filtering the signal so hard we could chose to study the signal when the temperature is supposed to be constant and just remove the average from it. This can be done for instance when there is no tapping. Then the temperature should only vary very slowly. To detect when a tapping occur the valve that controls the flow trough the primary side of the tap water heat exchanger can be monitored. This is easy to do if it is controlled electronically. If this can not be done we can instead look at the derivate of the temperature. If the derivate is larger than a threshold *b* we can say there is a tapping and not do the fault detection then [5].

A DH system is a slow process. The tap water side valves go from fully closed to fully open in about 30 seconds (typical) and the radiator side valve takes about 3 minutes (typical) to do the same [1].

When a temperature change occurs that needs to be adjusted the valves move. When the valves don't move the temperature can be said to be fairly constant. This can be used so we don't need to filter the signal so much. Doing this type of fault detection without filtering requires very little processing power [1-5].

By setting a threshold for the variance we can detect when a change in variance occurs. Depending on how fast we want the detection to react we can set the threshold differently. Using a 100 Hz sampling frequency we can detect faults fast even if we look at a window of many samples. E.g. 5000 samples against the set threshold. If one lot of 5000 samples has a variance higher than the threshold it may indicate there is some problem with the sensor but it can also be because of other dynamics in the process. To make the fault detection more robust we can add the criteria that *m* lots in a row should indicate a too high variance before we say it's a sensor fault. Of course this can also be set depending on user requirements. The house heating circuit has a time delay of hours. This means that nothing much except noise and a bias should show up on measurements done over a short period of time, providing no tapping occur [1].

Threshold setting

The probability that a measurement falls within 1.96*σ of the mean μ is

$$P(\mu - k*\sigma < x_n < \mu + k*\sigma) = 0.95 \quad (1)$$

provided that the measurements are normal distributed with std. deviation σ and mean μ. We want to detect a change in σ in order to see if the sensor works properly. This can be done by calculating the variance again and again and again to see if there is any change in it. Instead of calculating the variance we can count the number of times the measurement falls outside the boundary. It should be about 250 times in 5000 samples.

3. THEORY

Assuming that the noise is band limited Gaussian white noise with expectation 0 and std. deviation σ. We want to detect when the noise level increases above a set threshold *k*. Setting the threshold so that 95% of the samples will be within the boundary gives the boundary

$$k_1 = 1.96*\sigma \ , \quad (2)$$

x_n is N(0,σ) distributed.

If the value of σ is not known, but we have decided based on previous knowledge or error analysis that the maximum standard deviation should be no more than σ_{max} [7-9].

We can then set

$$k_1 = 1.96*\sigma_{max} \quad (3)$$

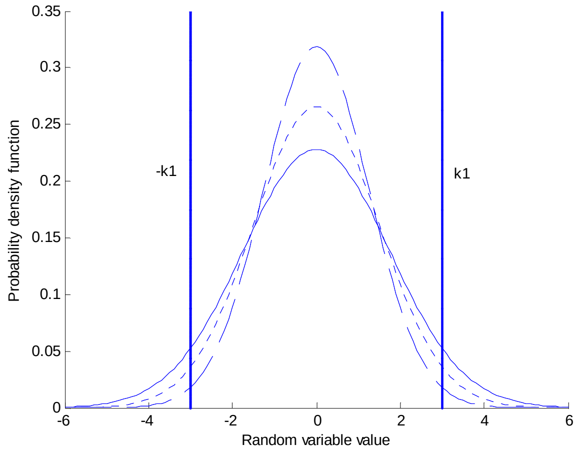

Fig. 3 Probability density function for x_n

Let S denote the std. deviation of the sample and σ be the std. deviation of the population. In the figure above we have plotted the probability density function for 3 different cases. The dotted curve represents how the probability density function would look if we had S equal to σ. 95% of the values would be found within the boundaries. The solid curve has S smaller than σ. This gives that more than 95% of the samples should have values within the boundary. In the next case represented by the dashed curve, S is greater than σ so less than 95% of the samples should be within the boundaries.

Since we want to detect when the variance goes above the threshold we will not consider it a fault if the variance is the same as or lower than the threshold.

$$P(|x_n| > k_1) = P(x_n > k_1) + P(x_n < -k_1) = 0.05 \quad (4)$$

Assuming $x_n \in$ N(0,σ). This gives us 3 cases when *n* is large:

1) $\sigma < \sigma_{max}$

2) $\sigma = \sigma_{max}$

3) $\sigma > \sigma_{max}$

Equation (4) is valid for case 2.

In case 1 the probability is less than 0.05 and in case 3 it is higher than 0.05 [8].

For now our fault detector will only react in case:

$P(|x_n| > k_1) > 0.05$. Let $x_n \in$ N(0,σ), $n \in [0, \infty]$.

If we take n samples, and n is large then about 5% of the samples should have an absolute value greater than k_1.

Lets say that we take 5000 samples. How many of those should go above the boundary before we can say with a 5% risk of being wrong that too many are above the boundary for a N(0,σ) distribution?

Let H_0: $\sigma \le \sigma_{max}$

H_1: $\sigma > \sigma_{max}$

Let z be the random variable corresponding to the number of samples going outside the boundary k_1. The likelihood that a sample has a absolute value greater than the boundary k_1 is 0.05 for a N(0,σ) distribution. The likelihood of m samples out of 5000 being outside the boundary is Binomial distributed n=5000 and p=0.05. Using this we can calculate $P(z < k_2)=0.95$ to find a boundary k_2. This says that with 95% certainty that out of 5000 samples, k_2 samples or less have an absolute value that is outside the boundary k_1. If there is more than k_2 samples outside the boundary k_1 we reject H_0 this gives us a 5% chance of being wrong.

Then H_0 and H_1 can be translated to

$H_0: z \le k_2$

$H_1: z > k_2$

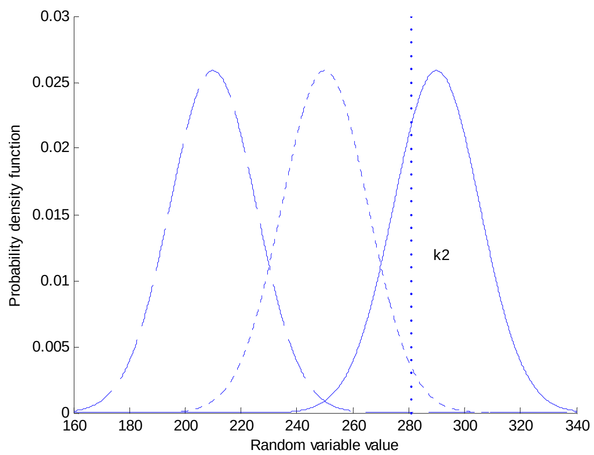

Fig. 4 probability density function for z

In the above fig. the dotted curve represents the case when μ_z is equal to 250. The dashed curve is an example when it's less than 250, and the solid represents a case when it's above 250. Thus solid curve is an example of the case when $\sigma > \sigma_{max}$ and the dashed curve is an example of when $\sigma < \sigma_{max}$.

The random variable z can be considered Binomial distributed since the likelihood of one sample x_n being outside boundary k_1 is constant. We can approximate a binomial distribution Bin(n,p) with an distribution if $np(1-p)>10$. Since n is 5000 and p is 0.05 then $np(1-p)$ is 5000*0.05*0.95=237.50. So the criterion is satisfied [8].

$$N(np, \sqrt{np(1-p)}),$$

$$\mu_z = np = 250,\ \sigma_z = \sqrt{np(1-p)} = \sqrt{237.50} = 15.4110$$

We want to find k_2 so that $P(z < k_2)=0.95$.

$(k_2 - \mu_z)/\sigma_z = 1.65$ (One sided boundary)

$$k_2 = 1.65 * \sigma_z + \mu_z \quad (5)$$

$1.65*15.4110+250=275.428$, $k_2=275.43$

If $z > 275.43$, we reject H_0 and say that σ greater than σ_{max}.

To get this fault detection algorithm working we need to count how many samples we have taken and how many of those have been outside the boundary k_1. To keep the sum from getting to large we reset the counter(s) when it reaches 5000. Then we need to check the random variable z against the threshold k_2. When z exceeds k_2 we say the sensor is to noisy.

One way of implementing this would be:
Compare if a sample has a value higher than k, if it has add one to z and one to n (the number of samples taken, else add 0 to z and one to n. Compare if n is 5000, if it is check if z is larger than k_2, if it is warn that the noise level is above the threshold and reset n and z back to zero.

So this means you need to do two "IF" statements every cycle and plus one extra "IF" every 5000 cycle. Now this algorithm has a 5% chance of saying that the variance is to high even though it's not just because of random events. To improve it we can require that it has to detect it's too high m times in a row before we accept it. Probability of being wrong m times in a row P(false alarm)^m = 0.05^m. This more or less eliminates the possibility for false alarms depending on the choice of m. But this also increases the reaction time the higher m the longer reaction time. Assuming a sampling frequency of 100 Hz and that we look at 5000 samples at a time m = 2 gives a reaction time of 100 seconds. This is a short time span for district heating.

The number of samples in each lot that we look at influences how large the 95% confidence interval is. This in turn affects how small difference between σ and σ_{max} we can detect. Few samples means that the difference between σ and σ_{max} has to be large, before we can say with 95% certainty that $\sigma > \sigma_{max}$. If we increase the number of samples from 5000 to 10000 we can detect a difference that is only 1% between σ and σ_{max}. This means that if we have 10.000 sampels and σ is 1% larger than σ_{max} we can with 95% certainty say that the variance is too high.

4. EXPERIMENTS

The district heating substation is simulated in Matlab Simulink and the fault detection algorithm is implemented as a Matlab script. In the performance overview below you will find performance indexes used for evaluating change detection algorithms. They are mean time between false alarms, probability of false detection, mean delay of detection and probability of non-detection [9].

Probability of false detection $p_f=p^m$
If we set p=0.05 and m=4 it gives the probability false detection equal to $p^m=0.05^4$.

Mean time between false alarms when n=5000, sampling frequency 100 Hz, p=0.05 and m=4 becomes in seconds

$$\frac{m*50}{(p^m)} \tag{6}$$

This means that mean time between false alarms is more than 8000 h or around a year which is good if the fault detection device is to be alerting the costumer at home. For other applications you might want it to react faster on an error and tolerate more frequent false alarms.

Mean delay of detection depends mostly on sampling frequency and the selection of p, m and n. In our case it takes about 50/0.95 seconds for a fault to be detected if m=1. If m = 4 it takes about $200/0.95^4$=245.55 seconds.

Probability of non-detection P(say H_0 true when H_1 true) depends on the difference between σ and σ_{max}. If σ is 5% larger than σ_{max} the 1.96* σ_{max} confidence interval will not contain 95% of the measurements. 1.96/1.05=1.8867. This gives that the new two sided confidence interval has k=1.8867. Tables show that it contains less than 94% of the values. This reflects on z also. 5000*0.06=300 sampels in average should be outside the boundary.

The probability of non detection at 300 is:

With n=5000, k_2=275.43, $\sigma_z=\sqrt{237.50}$, and μ_z = 300 the probability becomes

$$\Phi\left(\frac{k_2-\mu_z}{\sigma}\right)=\Phi\left(\frac{275.43-300}{\sqrt{237.50}}\right)=\Phi(-1.59)=$$
$$1-\Phi(1.59)=1-0.9441=0.059$$

With μ_z=250 it becomes:

$$\Phi\left(\frac{275.43-250}{\sqrt{237.50}}\right)=\Phi(1.65)=0.9505 \tag{7}$$

The probability of non-detection increases if we require that the fault detecting algorithm registers a fault m times before triggering the alarm. The probability becomes 1-P(correct detection)^m this means that a higher m gives a lower chance of detecting a fault. But this is a trade off, depending on what is more important, fast detection or low false detection rate.

In the case when P(detection) = 0.95 and m=4 we have 1-095^4=1-0.81 or 19% chance of non-detection.

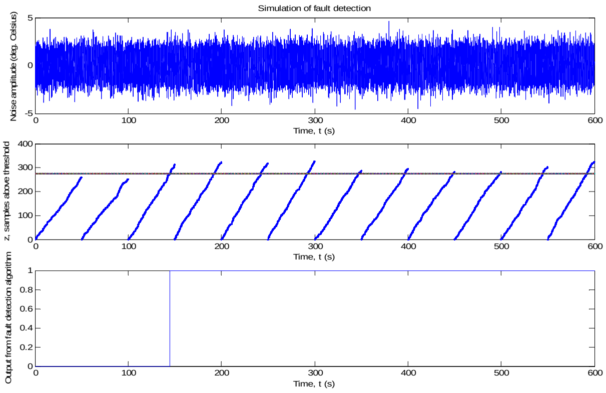

Fig. 5 The first graph represents the noise with an amplitude increase of 5% at t=104 which is not detectable by the naked eye, the second represents the number of samples above the threshold k_1 and the last graph represents the output from the fault detection algorithm. Having m=1 it detects the noise increase at t=150.

5. CONCLUSIONS

The presented fault detection algorithm was designed to detect increased noise levels in temperature sensors used in district heating substations using a limited amount of resources. However, this type of fault detection can also be used for other types of signals that can be approximated as gauss distributed.

The main benefit is that the algorithm doesn't require neither a lot of computing power nor high precision in calculations and it's easy to implement. The method can be implemented using a small microcontroller and be built in to almost any type of equipment that needs to monitor variance. This algorithm can also be changed so it monitors if the variance is lower than a certain limit. If we want it to react fast we can let it make a detection on lets say 300 samples, which needs a difference in std. deviation that is greater than 10% to be able to say that with 95% certainty it's above threshold. This can in turn be combined with a 5000 sample detection. We have one threshold but two counters. The first counter reacts fast to large increases and the second reacts a bit slower but detects also small increases in variance. Simulations show that at 5000 sampels the algorithm can very accurately detect that the std. deviation is 5% above the threshold. At 10.000 sampels it can detect an increase in the std. deviation as small as 1% according to simulations.

REFERENCES

[1] S. Fredriksen, S Werner "Fjärrvärme teori, teknik och function", *Studentlitteratur* , Sweden, 2001
[2] http://www.fjarrvarme.org, Sweden, 2005-01-30
[3] J.P Bentley, "Principles of measurement systems", *Prentice hall*, England, 1995.
[4] Matlab signal processing toolbox, Mathworks, release 13
[5] Y. Jomni, "Improving heat measurement accuracy in district heating substations", ***Lic. Thesis*** , Luleå University of Technology, 2004
[6] L.H. Chiang, E.L. Russell and R.D. Braatz,"Fault detection and diagnosis in industrial systems", *Springer*, Glasgow, Scotland, U.K., 2000
[7] G.R. Cooper, C. D. McGillem, "Probabilistic methods of signal and system analysis", third edition, New York, Oxford University Press, 1999
[8] K. Vännman "Matematisk statestik", Studentlitteratur, Lund, Sweden, 1997
[9] M. Basseville, I. V. Nikiforov, "Detection of abrupt changes: Theory and application". Prentice-Hall, Inc., 1993
[10] K.J. Åström, B. Wittenmark, "Computer controlled systems, theory and design", third edition, Prentice Hall, USA, 1997
[11] www.pentronic.se, Sweden, 2005-04-29
[12] F. Cederborg, Personal talk at Fjärrvärme för Unga, FVB, Stockholm, Sweden, 2005

AUTHOR(S): Kimmo Yliniemi, Jan van Deventer, Jerker Delsing, Department of Computer Science and Electrical Engineering, Division of EISLAB, Luleå University of Technology, SE-971 87 , Luleå, Sweden, phone +46 920 491 000 (switchboard) , fax +46 920 492 082, kimmo@ltu.se, deventer@ltu.se, jerker.delsing@ltu.se