1st International Conference on Sensing Technology
November 21-23, 2005 Palmerston North, New Zealand

Automatic Sensing/Actuation of a Dairy Farm

Ibrahim H Al-Bahadly and Gregory Bryant
Institute of Information Sciences and Technology,
Massey University, Palmerston North, New Zealand
i.h.albahadly@massey.ac.nz

Abstract

This paper presents the simulation and modelling of an automatic sensing/actuation system of a rotary farm dairy. At present some rotary farm dairies can only have their speed changed by physically changing the gear ratios between the driving motor and the platform. Others have a manual control to vary the speed of the driving motor. To use the rotary farm dairy more efficiently the milking pulsators need to be used effectively. Usually near the exit of the dairy there will be some cows finished milking and the pulsators will be idle. The simulation results proved that by automatically controlling the speed of rotation the number of idle pulsators could be reduced and the efficiency of the dairy farm could be increased. The process variable in the farm dairy control problem is not well defined. The speed controller needs to know more than just how many pulsators are idle or active it also needs to know where they are in relation to the exit of the rotating platform. The most difficult part of the control system was representing the state of the system. There is a lack of symmetry around the set-point, which was countered by putting more weight on the pulsators near the exit.

Keywords: simulation, variable speed control, milking farm

1 Introduction

The goal of this research was to model and simulate an automatic speed control system for the rotary platform of a farm dairy [1]. Farm dairies are used to milk cows. Figure 1 shows schematic plan of a rotary farm dairy. Cows are held in a yard and enter onto the rotating platform. Pulsators are then applied to the cow's udder as they rotate past an operator. The platform continues to rotate and by the time the cows near the exit they are usually finished. The pulsators are removed from the udder as they finish. The cows then back off the platform as they pass the exit.

The number of pulsators in a farm dairy is the constraint. The more pulsators there are the more cows can be milked at once. Thus to use the farm dairy more efficiently the pulsators need to be used more efficiently. Usually near the exit of the dairy there will be some cows finished milking and the pulsators will be idle. It is thought that by automatically controlling the speed of rotation the number of idle pulsators could be reduced. For example if the four cows closest to the exit were all finished milking the controller could increase the speed of the platform.

Occasionally a cow will still be milking when it reaches the exit of the farm dairy. The operators then decide whether to take the cow on another rotation or stop the platform and wait for the cow to finish milking.

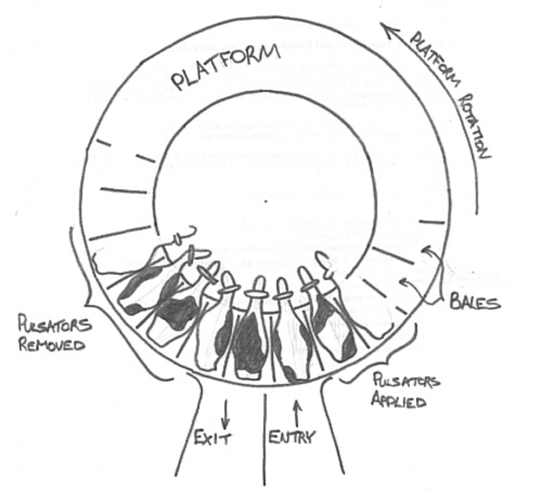

Fig.1 Schematic plan of a farm dairy

At present rotary farm dairies do not have any automated speed control system. Some can only have their speed changed by physically changing the gear ratios between the driving motor and the platform; others have a manual control to vary the speed of the driving motor.

The process variable in the farm dairy control problem is not well defined. The speed control is going to want to alter the speed of rotation depending on which pulsators are idle or active. It needs to know more than just how many pulsators are idle or active it also needs to know where they are in relation to the exit of the farm dairy. Somehow this information needs to be represented by one number that will tell the controller whether to increase or decrease the speed of rotation.

To explain this problem more let say that there is N number of pulsators numbered from 1 to N, with pulsator 1 being closest to the exit and pulsator N being furthest away. Because the duration that each cow milks varies the state (i.e. idle or active) of the pulsators will not be a simple matter of all being active except for the last 3 closest to the exit. It is also likely that pulsators 1, 2 and 3 will be idle, 4 active and 5 idle or any other combination of active and idle.

Some way of representing the state of the pulsators had to be devised so that it could be compared with a set-point. The controller can then change the speed of the platform appropriately by using this comparison. The value used to represent the state of the pulsators needed to indicate how different the actual state of the pulsators was from the desired set point.

2 Computer Simulation

It was decided that the best way to investigate the implementation of a controller was to model the farm dairy system and experiment with different state representations and controllers. The simulation would be cheaper and quicker than experimenting in a real farm dairy. The computer simulation was created on a PC using MathWork's® Matlab® and Simulink®.

A simulation of the farm dairy was created first without the controller. It was based as closely as possible on the characteristics of the real thing. The basic properties of the model are as follows;

1) It would continue until a set number of cows had been milked.
2) Each cow would be assigned a time for how long they would take to milk.
3) The mean milking time of each cow depended on what stage of the milking they entered.

With regards to point 1, 150 cows were milked each milking. The simulated farm dairy contained 20 sets of pulsators. Although in an actual farm dairy there is 2-3 sets of pulsators always idle as they pass the exit and entry. It was assumed that neglecting this would not affect the results. Except that the simulated 20-bale dairy would be equivalent to a 22-23-bale diary in reality. The number of bales in the simulation can be changed easily by editing just one parameter. A twenty-bale rotary was chosen because that is what the authors are most familiar with.

The simulation was based around the cows, using structured arrays. The Matlab® ***struct*** function was used to create a structured array called cows. The cows object had six fields to represent all the needed information about each cow. These fields are;

cows.number ; graphical identification
cows.milking ; whether the pulsator is idle or active
cows.mlktme ; length of time the cow has been milking
cows.mlkfin ; length of time the cow will milk for
cows.rotn ; number of degrees the cow has rotated
cows.gonernd ; number of times the cow has rotated

The simulation incremented the farm dairy by 5^o each program scan cycle. This would correspond to approximately 6 seconds in real life depending on the speed of platform rotation. Each scan cycle the program would also look at the fields of all the cows and change them appropriately as explained in the following paragraph.

With regards to the animation several events occur. Firstly each scan cycle of the program the cows and platform are rotated 5^o. Also when the milking field changes from active to idle the colour of the cow changes from green (light spot) to red (dark spot). Figure 2 shows the animation (colours not shown).

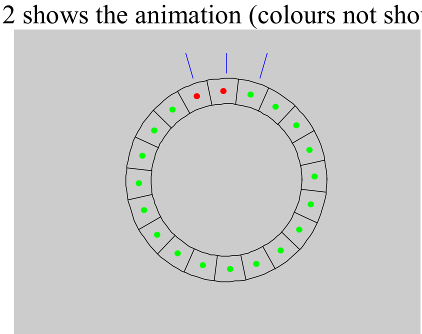

Fig.2 Computer simulated animation

Another simulation was also created without an animation component. This was done because of the long time it took for the animated simulation to run to completion. The non-animated simulation worked on all the same principles as the animated version.

3 Statistical Modelling

Because it would be impossible to model all the factors that effect how long each cow will take to milk a statistical model was used instead. In discrete event simulations choosing the right statistical distribution is arguably the most important thing. It is thought that a Poisson distribution [2] may be better suited.

In this simulation the milking time assigned to each cow as it entered the farm dairy was normally distributed with a standard deviation of 30 seconds. The mean of the distribution depended on what stage of the simulation the cow entered. This was done because it had been observed that the cows at the beginning and end of the milking usually milked for longer than the cows in the middle of the milking. So to simulate this, the mean of the normal distribution was modulated on a raised cosine wave with amplitude of 60 seconds and mean of 400 seconds. The wave completed one cycle over the duration of the simulation. The mean milking time of cows was found to be around 7 minutes. Figure 3 shows a plot of the modulation raised cosine.

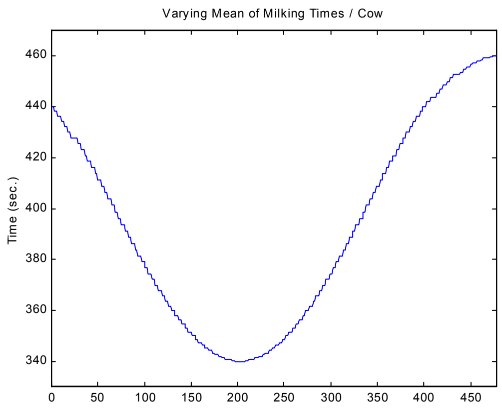

Fig.3 Modulation of mean milking time during simulation

4 Sate Representation

Representation of the non-continuous states was probably the most difficult part of this project. This is due to the many possible combinations of active and idle pulsators that can occur.

The first thing that was decided was that the controller would really only be interested in the pulsators nearest the exit of the farm dairy. It was decided that the last 6-8 pulsators would be of interest. Pulsators further away could be assumed to be active during normal operating conditions. These 6-8 pulsators of interest will be referred to as the control window.

Lots of representations were experimented with. One of the difficulties was the asymmetry of the control window about the set point. Because the set point was 2 the controller could only look at 2 pulsators to one side and 4-6 on the other. If the control window were made symmetric around the set point it would have only been 3 pulsators wide, which wouldn't really give the controller enough information. One way to compensate for the lack of symmetry was to put more weight on the pulsators closest to the exit. The idea behind this is to give the state a value of 0 when it is at the set-point. If the idle and active pulsators were represented by values of -1 and 1 respectively then the control window at the set-point would look like table 1.

Pulsator No.	**6**	**5**	**4**	**3**	**2**	**1**
Active(-1)/Idle(1)	-1	-1	-1	-1	1	1

Table 1 Status of pulsators at the set-point

One of the easiest ways to weight this situation to obtain 0 as the state value would be to give pulsators 19 and 20 a weighting of 2 and the others a weighting of 1. This gives 2x1 + 2x1 - 1x1 - 1x1 - 1x1 - 1x1 = 0. Or more simply using matrix multiplication

$$[-1 \quad -1 \quad -1 \quad -1 \quad 1 \quad 1] \bullet \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \end{bmatrix} = 0.$$

The row vector is the control window and the column vector is the controller weighting, the answer of the matrix multiplication is the state. It is this state that is fed into the controller. With a control-window of this size there is $2^6 = 64$, combinations of pulsators states are possible. All these different states can only be represented by an actual state ranging from integers -8 to 8, i.e. all pulsators idle to all pulsators active. So some how 17 state values have to represent 64 different situations and be able to represent how far the situation is from the set-point. So obviously many different situations will be represent by the same state value. In this example there is several situations that will result in a state of 0, even though they are not the set-point. Some examples of this are shown in table 2.

Pulsator No.	**6**	**5**	**4**	**3**	**2**	**1**
Control Weight	**1**	**1**	**1**	**1**	**2**	**2**
Active(-1)/ Idle(1)	1	1	-1	-1	1	-1
	-1	-1	1	1	-1	1
	1	1	1	1	-1	-1

Table 2 Examples of false set-points

This situation was typical of all the different control weightings that were tried. But it was found that even with this multiple representation the state answer was a relatively good representation of the situation. For example all the above situations would result in the speed of the platform remaining constant since the state equals 0. This result is most probably correct for each of the states in table 2. Let consider the last row, which is completely opposite that of the set-point. If the platform was to slow down then it is possible that the 2 active pulsators will still need to go round and the idle pulsators will be idle for even longer. If the platform was sped up then the active pulsators might end up going round again only to finish soon after resulting in being idle for the rest of the round. In none of the situations is it clear whether the platform should be sped up or slowed down. So leaving it at the same speed is possibly the best decision. Table 3 shows some other control weightings that were used. Only the last row is symmetric around the set-point.

Pulsator No.	6	5	4	3	2	1
Control Weight	1	1	1	1	1	1
	1	2	3	4	5	6
		1	2.5	4.5	9	15
	1	1	2	3	3	4

Table 3 Examples of trial control weights

Besides these linear control weighting's another system was also tried, we will call this "System 2" and the previous "System 1". With system 2 the control window was scanned from pulsator 1 upward and weighted pulsators at the start with the same consecutive status and then after that give the remaining pulsators another set weight. For example, if the last 3 pulsators on the platform, i.e. pulsators 1, 2 and 3, were all idle and then 4 was active then it would give them a weight of 2, and 4-6 would all have a weight of 1. It changes the weight when the status of the pulsators changes for the first time as shown in Tables 4 and 5.

Pulsator No.	6	5	4	3	2	1
Active(-1)/Idle(1)	1	-1	1	-1	-1	-1
Control Weight	1	1	1	2	2	2

Table 4 Example of system 2

Pulsator No.	6	5	4	3	2	1
Active(-1)/Idle(1)	-1	1	-1	-1	1	1
Control Weight	1	1	1	1	2	2

Table 5 Example of system 2

In the simulation this was carried out on a control-window of 6 pulsators with the initial weighting of 2 for the last 3 pulsators on the platform (as above) and then after the first status changed the weighting of 2 was given to the last 2 pulsators only. This would give a state of 0 at the set-point. This weighting system worked best because it put more importance on trends near the exit, for example if the last 4 cows were still milking or all finished milking etc.

5 Controller Tuning

Intuitively it is a SISO (Single Input, Single Output) system so classical control techniques could be used. It was decided that a PI controller would be all that is needed for control. Because of the possibly large rates of change in the states a derivative term would be acting on false information in some ways so it simplified the problem to leave it out.

The controller was implemented in Simulink® using a transfer function. Figure 4 shows a simplified block diagram of the closed loop control system.

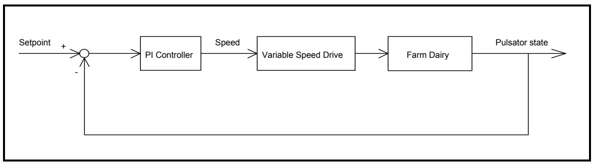

Fig.4 Simplified block diagram of the closed loop control system

The set-point for the system was to have the two sets of pulsators closest to the exit idle. Although doing this would limit the theoretical efficiency of the shed to 18/20 or 90%. The reason for this buffer of 2 pulsators is because of the random way that cows finish milking. If the set-point was to have all the pulsators active then there would be a large proportion of the cows still milking when they got to the exit.

Some cows will milk substantially longer than other cows so it may be more efficient to take them around rather then stop the platform. But on the other side taking some cows around means that they may finish early in the next round and there pulsators will lie idle, also cows don't like spending more time in the cow-shed than they have to.

Several methods were used to tune the controller, Ziegler-Nichols tuning [3]-[5], optimal control [6]-[8] and also trial and error using observation. The results of each method will be discussed below. It is during the tuning of the plant that the non-animated simulation was most useful due to its increased speed.

The controller was implemented in Simulink®. The simulation wrote the state to the Matlab® workspace so that the Simulink® controller could access it. The controller then wrote the result after passing it through the transfer function (i.e. PI controller) to the workspace once again where the simulation could access it. Figure 5 shows the Simulink® controller diagram.

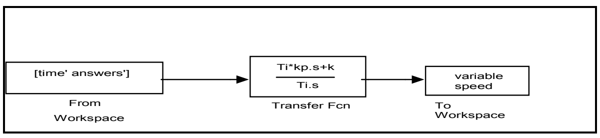

Fig.5 Simulink diagram of PI controller

It was found that the Ziegler-Nichols methods (1&2) were not appropriate. The critical period of the system was found to be equal to the period of rotation, but the integral time constant and the gain found using Ziegler-Nichols tuning rules did not produce very good results. Cows going round more than once largely effected the critical period. When a group of cows finish as they neared the exit the platform speeds up to reduce the number of idle pulsators. This increase in speed sometimes caused the other cows on the platform to be still milking when they neared the exit. Because of this the controller slows the platform down so they will have time to finish although some would end up going around anyway. By this time the first cows that went round have finished and due to the platform slowing down other cows have finished also. Because of them the platform once again speeds up. This cycle continues with a period of about one platform rotation. Up to a certain gain this cycle doesn't occur but once a critical gain has been reached the cycle begins and is not sensitive to increasing the gain by more. For this reason the Ziegler-Nichols tuning rules were not very applicable to this system. Figure 6 shows graphically the number of the idle pulsators with this speed control system. It shows the number of pulsators that were idle immediately next to the exit. Figure 7 shows the actual rotation time of the platform and also the mean milking times of the cows.

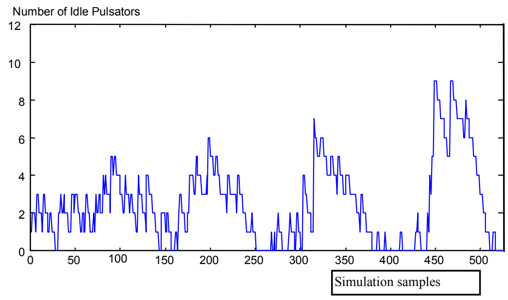

Fig.6 Number of idle pulsators with continuous speed control

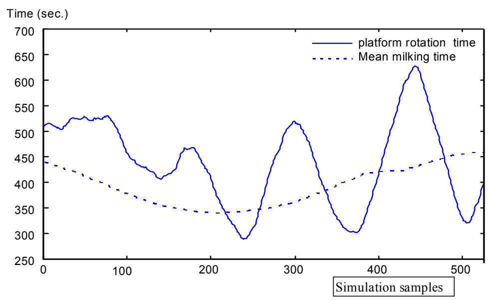

Fig.7 Platform rotation time of continuous speed control

Once this was found to be the case optimal control techniques were tried. A cost function would be minimized. The cost function included a pulsator efficiency term as well as a term representing the number of cows being rotated twice. The pulsator efficiency term was to be maximized and only a small cost was put on the number of cows being rotated. With regards to the latter the number of cows going round was compared with an ideal 10%, if the number going round was greater than the ideal it was included in the function. If it was less it was not. This is because it was only considered detrimental to have more than 10% of the cows going around. The optimization of this problem was found to be quite difficult. The gain and integral time constant of the controller could be set at different initial values and they would return to different optimal values. This showed that there were many local minimums. The random nature of the system also had a huge detrimental effect on the optimization. To help reduce that large sample sizes needed to be used. To do this the optimization algorithm would run the simulation several times with the same controller parameters. But to have a large enough sample size meant that it took a very long time to get a result. It would take about 1.5 hours to run 10 simulations.

The general efficiencies with this continuous speed control system (variable speed platform rotation with non-stop) were 80%. When the simulation was run without speed controller (fixed speed platform rotation) the efficiency was 75%. Only a disappointing 5% increase was achieved. The average length of a herd milking is about 60 minutes if plant cleaning is not included, this means that 5% equates to a time saving of just 3 minutes per milking, twice a day for 8-9 months of the year.

After these trials the general sensitivity to controller parameters was at least known even if the results had been unsatisfactory. A good understanding of the system had also been developed. From here the animated simulation was bought back into play. From watching this it was decided that stopping the platform when no pulsators were idle would be beneficial. This would stop the farm dairy from entering into the cycle where it would be taking large groups of cows around and correspondingly speed up and slow down every round as explained earlier. This worked very well. The gain could be turned up a lot more because the platform would stop and wait if no cows where finished in the control window. The higher gain even enabled some of the random variations to have a reduced effect on the efficiency.

With this non-continuous speed control (variable speed platform rotation with some stops if no cows were finished milking) the efficiency was increased to 86% percent. This is an increase of 11% on the uncontrolled platform, or with a 60 minute milking as above, 7 minutes could be saved. This is equivalent to an extra rotation of the platform.

Figure 8 shows the number of idle pulsators immediately adjacent the exit. Figure 9 shows the rotation period of the platform and also the mean milking times for the cows. The discontinuity's in the period plot are due to the platform stopping when no pulsators where idle.

With the continuous speed controller where the platform speed could be changed but not stopped the gain had to be kept relatively small to stop the system from going out of control. However this low gain was only affected by low frequency disturbances such as the varying mean over the whole milking. Also in reality the length of time that cows milk for changes during the milking season. In spring they have more milk and thus take longer to milk than near the end of the dairy season when they are drying off. The low gain was also sensitive to this low frequency disturbance.

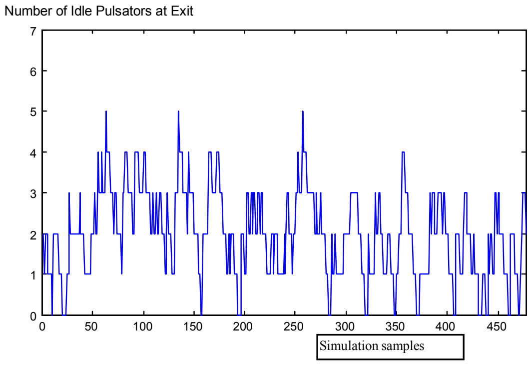

Fig.8 Number of idle pulsators at exit for discontinuous system

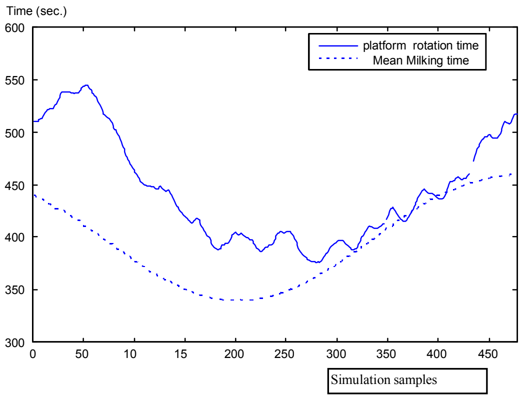

Fig.9 Platform rotation time of discontinuous speed control

These results are under ideal conditions and in reality the platform is usually stopped occasionally for many reasons, such as cows kicking there cups off, needing special treatment etc. These stoppages vary in time, and would reduce the efficiencies. But on the other side, these occasional stoppages that also make the use of an automatic speed control more useful. Intuitively these stoppages would have a larger effect on an uncontrolled platform than they would on a controlled one.

6 Conclusions

The aim was to model and simulate an automatic speed control system for a rotary dairy farm. Usually near the exit of the milking platform there will be some cows finished milking and the pulsators will be idle. This paper proved that by automatically controlling the speed rotation the number of idle pulsators could be reduced.

The most difficult part of the control system was representing the state of the system. The lack of symmetry around the set-point was overcome by putting more importance on the pulsators near the exit. System 2 obtained the best results because it put more importance on trends near the exit.

When the simulation was run without a controller the efficiency was 75%. The efficiencies with a continuous speed system were 80%. This equates to a 3 minutes reduction on an average milking duration of 60 minutes. With a non-continuous speed control the efficiency was increased to 86% percent. This is an increase of 11% on the uncontrolled platform, or with a 60 minute milking 7 minutes could be saved.

7 References

[1] C. C Thiel and F. H. Dodd, *Machine Milking*, National Institute for Research and Dairying, Shinfield, 1977.
[2] J. E. Freund, *Modern Elementary Statistics*, Prentice-Hall, 2003.
[3] K. Ogata, *Modern Control Engineering,* Prentice-Hall, 2001.
[4] R.C. Dorf and R. H. Bishop, *Modern Control Systems*, Prentice-Hall, 2004.
[5] T. Bartelt, *Industrial Control Electronics*, Delmar Thomson Learning, 2002.
[6] S. M. Shinners, *Advanced Modern Control System Theory and Design*, Wiley, 1998.
[7] D. E. Kirk, *Optimal Control Theory: An Introduction*, Dover Publications, 2004.
[8] I. V. Kolmmanovsky and A. G. Stefanopoulou, Optimal Control Techniques for Assessing Feasibility and Defining Subsystem Level Requirements: An Automotive Case Study, *IEEE Transactions on Control System Technology*, Vol 9, No 3, 2001, pp. 524-534.