COORDINATED MACHINE SPEED CONTROL IN A MULTIPLE FILLING MACHINE INSTALLATION

Marcus Lindh
Andreas Nilsson

Department of Industrial Electrical Engineering and Automation
Lund Institute of Technology
&
Tetra Pak Carton Ambient AB
Lund

Multiple installations of filling machines are connected so they take product (e.g. milk) from the same pipe. For a certain type of filling machine this can imply a problem if several machines shape their packages at the same time (called collision). When such simultaneous operations happen, large variations in pressure occur in the product pipe, which can lead to decreased package quality. To avoid this, the machines must be non-concurrent.

This article describes the possibilities to avoid collisions. It all boils down to affecting the production rates of the machines in a way so they shape their packages at different times. There are essentially two different methods: solid-state relay and frequency inverter. The main focus has been to develop an algorithm that performs the coordinated control preferably using solid-state relay. It is shown that some collisions have to be tolerated in order to maintain a reasonable production rate.

1. INTRODUCTION

The Tetra Pak filling machines TFA/3 are often placed in parallel, where they take product (e.g. milk or juice) from the same product pipe. When several filling machines shape their packages (called hit) exactly at the same time, large pressure variations can occur in the product pipe. These pressure variations imply decreased package quality which shows as small deformations, e.g. with folds. Such folds make the package look ugly, but the most serious consequence is that there is a risk it becomes weak and leakage can occur. Another consequence from large variation in pressure is decreased accuracy in package volume.

This article gives a summary of the master's thesis *Coordinated Machine Speed Control in a Multiple Filling Machine Installation*, which explores different possibilities to prevent filling machines from shaping their packages at exactly the same time.

The aim was to construct a control system which makes sure that multiple installations of filling machines become non-concurrent. Non-concurrent means that they do not hit at the same time, which makes the largest pressure variations in the product pipe disappear. By doing so, fewer packages are deformed and volume accuracy improves. The reason why hits from parallel machines can become concurrent is that they do not keep their exact stated production rate, for example due to friction.

A demand is that the control system should be able to handle up to 6 parallel machines. The system must also be able to handle two different production rates (3600 and 4500 packages/hour), depending on which package volume the machine produces. In addition low cost is desirable.

2. METHODS TO CHANGE MOTOR SPEED

It is quite obvious it demands some kind of speed control on the machines in order to prevent them from hitting at the same time. The machines do not necessarily have to be given a different production rate; it may be enough making a machine faster or slower at the precise moment when a collision would occur. This way the machines are given a different production rate in *mean*. Such a selective measure could be performed by braking or just by removing the driving force on the main motor (break the current).

The production rate is directly coupled to the rotation speed of the motor shaft, which means that the control task can be performed by changing this rotation speed. There are principally two ways of doing so: using solid-state relay or frequency inverter.

With a solid-state relay the rotation speed can be affected by breaking the current during short periods of time. Then the motor is without driving force and it will slow down. Solid-state relay is an extremely reliable and long-lived component. For the purpose it is meant to serve here, with many on/off-switches, it is well suited. A disadvantage is that it can only decrease the production rates of the machines. Furthermore, it has an uncertainty when switching. The reason is that it switches on when voltage passes zero and switches off when current passes zero. (Bishop, 1986)

In contrast to a solid-state relay, which can only make sporadic delays, a frequency inverter can really change rotation speed of the motor. In addition, a frequency inverter can both increase and decrease the rotation speed which is a great advantage.

Solid-state relay was considered to be the most interesting alternative, mainly because of the lower cost.

3. CONTROL SYSTEM

Now it has been determined which the possibilities are when the decision to change the speed of a machine already has been made. What decides *when* a machine speed should be changed is next to be investigated. In order to obtain a solution that solves the task, several different algorithms were developed. During development they were all seen as possible alternatives to handle the control. The design of the algorithms is dependent on whether they use solid-state relay or frequency inverter. (It should be pointed out that the algorithm for frequency inverter is not fully developed.)

Simulations were made in MATLAB® Simulink to evaluate the eligibility of the algorithms. Through mental effort one can realize what happens when two parallel machines are running, and to some extent also three, but after that it is almost impossible. This is where the greatness of simulations is shown, since it allows thorough analysis why some situations can occur and what could be done to avoid them.

From now on several terms will be used. One of these terms is hit interval, which refers to the smallest time difference allowed between two hits of any machines before they are considered to collide. A reasonable hit interval should be in the range 50-100 ms; the simulations will show what hit interval to use. A small hit interval implies that two hits are allowed to be close to each other, and vice versa. When the time difference between hits from two machines is close to the hit interval a delay (for algorithms using solid-state relay) should be made. A delay is meant to increase the time difference between hits from two machines. Another term that often will be used is base speed, which is defined as the lowest guaranteed speed a machine is allowed to have. Within each base speed the period (time difference between the machine's hits) of the machines only differ a few thousandths of a second.

3.1 Solid-State Relay

The quality of an algorithm is, besides the avoidance of all hits, dependent on the change in production rate it causes. With a solid-state relay the machines can only become slower, which means that less interference results in higher production rate. Simulations showed that there was a large decrease in production rate when all collisions were removed. The decrease was so heavy that the most reasonable approach was to lower the demands on avoiding all collisions. This resulted in an algorithm called *Allow 2-hits*, which only avoids collisions between three machines' hits.

Allow 2-hits waits for a machine to hit. At that moment it calculates if the machine will collide with another machine in its next hit. If a collision is predicted it is investigated if there is a third machine involved (hit interval: 100 ms). If that is the case it must be calculated which machine has the rightmost hit (last hit) in the collision. It is the machine with the last hit which should be delayed, but this is not always feasible. If the last machine already has been delayed it can not get a further delay. In that case, the algorithm investigates if the machine in the middle can be delayed. If also this is impossible the collision between three machines can not be avoided (unless the leftmost machine is given a very long delay).

A possible scenario is that two machines with exactly the same production rate will collide with each other. According to the algorithm's basic principle this would be allowed to happen without any correction being made, which is not desired. Since there is no difference in production rate between the machines they will collide with each other until a third machine is involved and the algorithm takes care of that collision. To avoid this, there is a separate part which takes care of collisions between machines with exactly the same production rate. The algorithm decides to delay the machine positioned last in collision.

Since *Allow 2-hits* only removes collision between three machines, it is of course possible for machines with same base speed to fall into step. This is solved by combining it with another algorithm that controls machines with same base speed. The algorithm considered to do this best, regarding production rate, is called *Fox jump*.

Fox jump lets every machine retain its production rate until it gets too close to a hit from another machine (hit interval: 50 ms). When the hits for two machines with same base speed are too close, the slower machine will be delayed to let the faster machine pass without a collision. The algorithm relies on that the delay can be made as long as two hit intervals. This is very important, since a machine otherwise cannot "jump over" another machine without collision. The jumping behaviour is a partial reason to the name *Fox jump*.

Also *Fox jump* waits for any machine to hit. When a hit has occurred the algorithm calculates how the machines' hits are positioned one period ahead. Even if a collision is predicted it is not certain that the machine should be delayed. First the algorithm checks whether a delay causes a collision between two machines with different base speed. If that is the case the machine is not delayed. The reason is that the other machine involved in the collision can be a slower machine, which has already had its hit and therefore is unable to avoid the collision. There is no reason to avoid one collision by creating a new one.

The combination of *Allow 2-hits* and *Fox jump* is the foundation for the control system.

3.1 Frequency Inverter

The main difference between solid-state relay and frequency inverter is that the latter not only can make a machine slower but also faster. Besides, a frequency inverter can really change the speed of a machine. This is a difference compared to using solid-state relay, which needs to delay a machine occasionally to change its average production rate.

The most advantageous solution to avoid collisions should be to equally distribute the machines' hits, which would make the time distance between them as long as possible. The hits would then affect each other as little as possible. To accomplish this all machines must have exactly the same average production rate. Since a frequency inverter can make a machine faster, all machines within a base speed can obtain the rate of the fastest machine (master). This is a significant difference compared to using solid-state relay, where all machines are delayed to change the production rate and therefore decreases the total production. A frequency inverter can instead increase the machines' production.

The algorithm *Equal distribution* first calculates the number of machines in production for each base speed. For every base speed the fastest machine is appointed to master. The other machines should increase their speed to get the same production rate as the master. As mentioned above the aim is to distribute the hits to make the time distance between them as large as possible. The control of the hits is accomplished by cascade-connection of two proportional controllers. The inner controller in the cascade-connection controls the speed of a machine so it obtains the same speed as the master. The outer controller adjusts the time distance between the machines' hits within same base speed to be as long as possible.

The algorithm is written in a way that does not compare hits from machines with different base speeds. Hence, collisions between two machines are allowed. When the machines' hits are equally distributed the algorithm maintain this appearance until a new machine is appointed to master, e.g. if the fastest machine is taken out of production. The algorithm must then control the hits of remaining machines and distribute them equally with the new production rate.

4. EXPERIMENTS ON THE FULL SCALE MACHINES

To verify that the control system is able to perform its task in reality, some tests were carried out. Ideally the algorithm should remove all collisions between more than two machines. Unfortunately there were only two real machines available so the other (four) had to be simulated within the PLC.

When no algorithm was used there were around 390 collisions (with hit interval 100 ms) between three machines. With use of both algorithms the different measurements showed that the number of collisions was reduced to approximately 15. The results from one of the measurements are shown by the histogram in Figure 1. Every bar in the histogram represents the sum of all time distances between the hits of three machines, divided into intervals of 5 ms. Have in mind that all intervals under 100 ms are collisions. As seen, the simulation has no collision (except for a transient). The measurement from the real test shows a number of collisions. This is due to uncertainty in both period and delay for the real machines. It also can be observed that the number of large time intervals is decreased when control is performed, which of course is a consequence from removing the collisions. Another obvious consequence, proved in the test, is that a decreased number of machines results in a decreased number of collisions.

Also another aspect of the control system was tested. A few times during the test a new package material reel was inserted. This gave a perfect opportunity to see that the situation when a machine is out of production could be handled.

5. CONCLUSIONS

Originally, the solution algorithms were designed with aim at avoiding all collisions. However, the simulations showed a decrease of approximately 5% in production rate when doing so, which of course is an unacceptable deterioration. Therefore the decision was taken to allow presence of some collisions if it gave reasonable production rate. The result from this was to allow collisions between two machines with different base speeds. With this measure the decrease in production rate was improved to be between 0-0.5% in the normal case. Some simulations have shown scenarios where a machine has had a decreased production rate of up to 1%. This is however an extreme case which from a statistical point of view ought to be levelled during a longer time. Since TFA/3 has an over-capacity, it will not fall below its lowest stated production rate.

When tests on the full scale machines were performed, two unforeseen properties showed.

The first one was that the machines' periods were not always the same. The machines had both a clear asymmetric behaviour and a random disturbance. The asymmetry can be illustrated by thinking of the production cycle as a circle. Every hit is the end of one period but also the beginning of the next. This means that if one period is a little longer the next one will be slightly shorter. To minimize this problem the mean value was calculated. Further testing is required to determine what the best action is. If, for example, the asymmetry is more common the most suitable would be to keep track of two periods per machine.

The other property showed that the machines had to break the current for different lengths of time in order to obtain the same delay. For the performance of the solution it is important that the delay is close to the intended one. Thus, this is something that

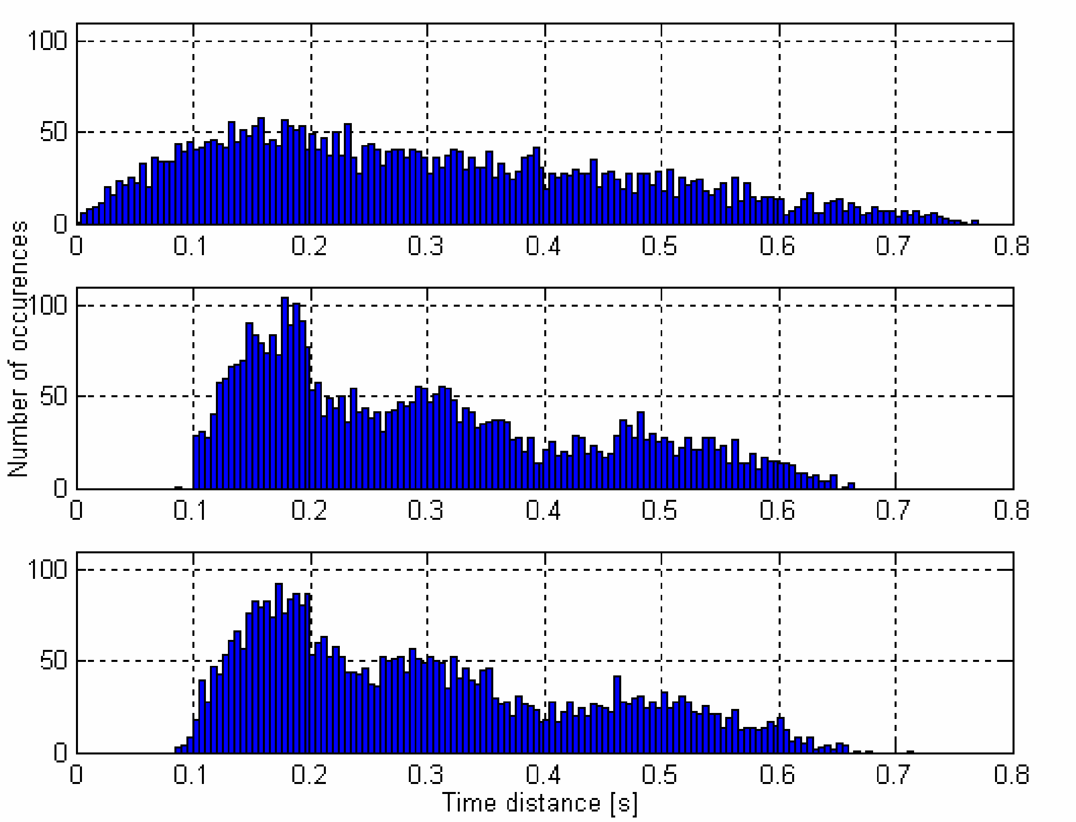

Fig. 1. Histogram over time differences between hits from a test without control (top), a simulation in Simulink with control (middle) and a test with control on full scale machines (bottom). Every bar represents an interval of 5 ms. The graph corresponds to 10 minutes measurements.

ought to be paid attention to. Our suggestion is that every machine keeps track of its period time and adjusts the break time until the delay is 100 ms (or at least very close to it). This should be possible with simple proportional control. We see this as a demand before the solution could be regarded as fully developed.

5. FUTURE POTENTIALS

Besides the need of adaptive length of the current breaks, the following issues should be considered for future development.

Since the solution has been built to fit TFA/3, it is written to handle only two different base speeds and to be put into an external PLC. If the solution is to be used on other systems with more base speeds, changes must be made. It would naturally be desirable to be able to manage an unlimited number of base speeds. When it comes to where the program should be implemented, the best solution would be if it were placed in the PLC of every machine. For TFA/3 this was impossible because of the lack of network support, but for other systems the situation may be different.

Further development of the control system would naturally be to remove all collisions without decreasing the production rate. This is considered to be possible only if a frequency inverter is used. However, we are not convinced that even a frequency inverter could manage, but simulations of *Equal distribution* shows promising results. Another advantage is that the production rate for the machines can be increased. The only drawback for the frequency inverter is the higher price compared to solid-state relay. From all other aspects a frequency inverter is preferable.

REFERENCES

Bishop, Anthony (1986), *Solid-State Relay Handbook with Applications*, Howard W. Sams & Co., Indianapolis