The effect of wireless communication on the distance control between vehicles

Mª Luisa Ruiz de Arbulo Gubía, Steven Gilijns, Marc Engels
Flander's Mechatronics Technology Centre (FMTC),
marialuisa.ruizdearbulo@fmtc.be

Abstract: This paper analyzes the effect of communication delays on a distance control system. The analysis has been made by simulating a Bluetooth link in the control system. First, the controller was designed for the ideal case of a hardwired communication link. Then, an ideal and a real wireless link have been simulated on the system to analyze the effect of retransmissions of a single packet and of multiples consecutive packets. The results of the simulation allow us to state that the parameters that determine the performance of the system are not only length but also time instant in which they happen.

Keywords: wireless control, Bluetooth,

I. INTRODUCTION

Cooperative driving is generally regarded as a potential solution to increase the capacity of roads [1]. Distance control between vehicles in a platoon is a basic feature of cooperative driving. The performance of distance control is crucial for the efficiency of platoon driving: a too long distance will decrease the capacity of the road, while too short distances might result in potential collisions. If information (e.g. position) has to be transmitted between vehicles, this has to be done with a wireless connection. Although the intermittent packet losses and delays introduced by this wireless connection will have major impact on the performance of the distance control, only limited results are available in literature about their effects. The objective of this paper is to assess the influence of effective communication delays on the performance of distance control between vehicles.

II. DESCRIPTION OF THE SYSTEM

Distance control with inter-vehicle position communication is a nice example of a wireless control system where the control loop is closed via a wireless link. The control system (figure1) consists of two vehicles, the leader and the follower with the same acceleration limitation. The speed of both vehicles is also limited; however, the maximum speed that the leader vehicle can reach is 90% of the maximum speed of the follower. It is realistic to assume that, in normal driving a vehicle doesn’t run at its maximum speed nor even at the 90% of this maximum. In the cases in which distance control between vehicles is used the leader speed could be even lower. A discrete-time PD controller, placed inside the follower vehicle, controls the distance between both
vehicles by changing the acceleration profile of the follower [2] and [3]. This has been
designed and optimized for a hardwired communication link [4]; it consists on a
proportional part and a derivative part (Ec.1).

\[ a[t] = Kp \ e[t] + Kd \ e'[t] \] (Ec.1),

where \( a[t] \) is the acceleration introduced to the follower vehicle and \( e[t] \) the distance
between both vehicles. Each vehicle has a position sensor. The distance between the two
of them is obtained as the difference between the two position sensors. Initially, an
integration block was added inside the controller. However, it was not really needed, so it
was decided not to use it to avoid PID wind up. The position of the leader is sent every
20ms over a Bluetooth link to the controller in the follower. The sampling interval of the
controller has been synchronized with the communication interval of 20ms. The behavior
of the follower vehicle (acceleration to position) is modeled as a continuous time double
integrator. As input, it uses the last acceleration value passed by the controller. Its
position output is sampled synchronously with the execution of the controller.

Figure1. Control system: Leading vehicle, controller and follower vehicle.

To simulate the wireless link Bluetooth has been chosen. Due to the frequency hopping
for every packet performed by this standard, all packets are independent to each other.
That means that all the packets have the same packet loss probability. This probability
and the time increment between consecutive retransmissions has already been measured
in previous experiments made at FMTC with commercial modules. The results of those
measurements are a successful transmission time of 20ms ± 5ms, a time interval of 20ms
± 5ms added by every retransmission and a probability of packet loss of 0.1. All these
factors allow to model the Bluetooth link as a Bernoulli process with success parameter \( p = 0.1 \). The graph (figure2) confirms our experiment results; the average delay for a single
transmission is 20ms with an additional 20ms delay for each retransmission and the
probability of using more than 4 retransmissions lower than 0.01%.
Thus, in this paper we consider a constant delay of 20 ms per transmissions and up to 4 retransmissions per packet. As a consequence, the maximum considered delay is 100ms.

III. SIMULATION RESULTS

The control system has been simulated with Simulink. The scenario simulated includes two worst cases: The leader starts from speed equal to 0m/s, at 3s accelerates at maximum acceleration rate until it reaches its maximum speed (90% of follower’s maximum speed) and keeps running at this maximum speed for some time. Finally, at 50s, it decelerates at maximum deceleration rate until it is completely stopped. The curve described by the leader’s acceleration is not a perfect pulse; it doesn’t go instantaneously from 0 to maximum acceleration/deceleration. In our simulations we used a maximum velocity of 8m/s for the follower and 7.2m/s for the leader, a maximum acceleration of \(3m/s^2\) and a maximum deceleration \(5m/s^2\) for both of them. It is important to notice that the maximum deceleration is almost twice the value of the maximum deceleration. That means that the speed and position changes will be faster at maximum deceleration, and as consequence the effect of the possible delay will be more pronounced.

First, the behavior of the system is analyzed for the case the position data transmission is hardwired. As performance measure, we use the maximum distance error between the leader and the follower. Without loss of generality, we assume that the initial and desired distance between leader and follower is zero meters, from where it follows that the follower trajectory error is the distance between both vehicles. During acceleration this error is positive because the follower is lagging the leader. During deceleration, however, the follower overtakes the leader and the error becomes negative. Checking carefully the curves (figure3) of both trajectories we can see that just after accelerating, at around 5s, the distance between the two vehicles becomes larger and then is reduced again to zero. That is possible because, once the leader is at its maximum speed the follower can still increase its speed and catch the first vehicle. During deceleration we have the same effect but in that case the follower overtakes the leader, or crashes against it. In that case the overshoot indicates the minimum initial distance they should keep to avoid collisions.
When introducing a perfect Bluetooth link in the system, a constant transmission delay of 20 ms is applied to every packet. This makes the distance error to be larger, although the general trends are the same as in the wired case. The graph (figure 4b) shows an initial overshoot caused by the delayed reaction of the follower on the acceleration of the leader. This behavior can be understood looking at the speed curve of both vehicles (figure 4a). The follower starts to accelerate 20 ms later than the leader, which makes the error distance to increase fast. Some seconds later the deviation of the follower evolves towards a constant value different from zero. That means that the leader has reached its maximum speed but the follower can still continue increasing its speed up to its maximum. As this maximum is higher for the follower it can catch up the leader, reducing the distance error to that constant value. If the follower received the packets with no delay this error distance would go back to zero. However, it receives them with 20 ms of delay which means that by the time the follower gets the desired position the leader has already advance at its maximum speed, 7.2 m/s, for 20 ms, which makes 0.15 m, just the error distance. As the follower is not aware of this delay it doesn’t increase its speed to try to catch the leader. Finally, at 50 ms the leader decelerates again until its speed is zero. At that moment the absolute value of the error increases again for some seconds and then it goes back to zero meters. In that case, however, the error is negative, which means that the follower overtakes the leader. Due to the constant delay, the follower starts to decelerate later and, as a consequence, its speed is higher than the speed of the leader.
Next, the effect of having up to 4 retransmissions has been analyzed by adding extra delays of up to 80ms to certain packets. Even when a delay happens, the controller keeps working at a sampling rate of 20 ms, but it updates its input only when it receives new data. The delays have been placed at different points of the trajectory and it has been seen that the worst cases are those in which the delay happens during acceleration and deceleration. Because of that, delays equivalent to 1, 2, 3 and 4 retransmissions have been placed at 3.2s and 50s (figure 5).

The first conclusion from this simulation is that, as expected, the distance error increases with the length of the delay. If it happens during acceleration, the larger it is the higher the overshoot. During deceleration the overshoot is also higher when an extra delay occurs, but, in this case, we can see that having a longer delay can be less damaging than having a short one. That effect is due to the fact that when the controller doesn’t receive a packet on time it uses the last value received, so it will interpret that the leader has
stopped and will start to decelerate the follower at its maximum rate. If this extra delay is short the follower will go back to a higher speed, but if it is longer the speed at the instant when new data is received will be already much lower. We also have investigated the influence of the moment of the delay. It is clear that the highest overshoot occurs when the vehicles are changing their speed, nevertheless it is interesting to know if that happens once this change was just made or at the same instant in which it starts to accelerate/decelerate. For that we have plot the maximum/minimum overshoot versus the time instant in which the sample delayed occurs (Figure 6).

To understand the graph, we should divide each of them in two parts, before and after the instant in which the leader has already started to accelerate/decelerates. Once the leader is already accelerating or decelerating, after 3.05s and 50s respectively, they show the same behavior explained before with the curves of figure 5. If the delays occur some instants before those points (3.05s and 50s) the maximum delay can be reduced, in case the vehicles are acceleration, or significantly increased, in case of deceleration. To explain this phenomenon we resort to the curves of the acceleration, speed and trajectory distance of both vehicles. We plot them for a 100ms delay at 49.76s, the time instant is just when we have the maximum negative overshoot, and 50s (figure7). When the delay happens at 47.76s the controller receives the same position data for some time, interprets the leader has stopped and makes the follower decelerate. When finally the controller receives new data the distance between both vehicles has increased and it makes the follower to accelerate to catch the leader (acceleration peak around 50s). The longer the delay, the larger the distance between the vehicles and the more the follower has to accelerate to catch the leader. When the controller detects that the leader is reducing the speed, the follower acceleration is not zero, as it should be, but positive. That is why it takes longer to the follower to get a maximum deceleration rate and, as consequence, it runs a longer distance before it stops.
Finally, we have studied the influence of the relative position of the two consecutive packets with retransmissions (causing excess delay). The results (figure 8) show that the overshoot is largely determined by the total excess delay of the two packets. The larger this total delay, the larger the overshoot. However, that does not mean that for equal total delays we have equal overshoot. When comparing two cases with different sample delays but equal total delay, the overshoot will be slightly higher in the case in which we have the sample with the longest delay. For example, if we have a case 1, delay1 = 80ms and delay2 = 20, and case 2, delay1 = 60 and delay2 = 40, the highest overshoot will happen for case 1 because the longest delay, 80 ms in this case, regardless its position.
IV. CONCLUSION

As shown in this paper the wireless communication has a major impact on the performance of distance control between vehicles. A wireless link can increase the distance error with an order of magnitude compared to hardwired link; not only the value of communication delay but also the precise timing has a major influence on control performance. The results reveal that even in the case of an ideal wireless link, without retransmissions, the constant communication delay introduces higher distance errors between both vehicles than hardwired. When a real Bluetooth link with probabilistic retransmissions is considered, the performance of the system will deteriorate as a function of the length of the delay caused by those retransmissions. However, the actual distance error is not only dependent on the communication delay but also on the instantaneous acceleration and speed of the leader. Especially critical are the moments when the leader is at maximum acceleration/deceleration rate and some instants before that rate changes. In general, the performance evaluation of the control loop in the presence of a realistic wireless link has to take both a communication delay model and the set point for the controller into account. To do so efficient methods are still under development and must be further explored.

REFERENCES.