Performance of Packet-Based Frequency-Hopping Spread Spectrum Radio Control Systems

Abdallah Ismail[1], Ioannis Lambadaris[1], Chung-Horng Lung[1], and Nishith Goel[2]

[1] Department of Systems and Computer Engineering
Carleton University, Ottawa, Ontario, Canada
aismail2@connect.carleton.ca,
{ioannis,chlung}@sce.carleton.ca
[2] Cistel Technology, Ottawa, Ontario, Canada
ngoel@cistel.com

Abstract. Real-time Radio Control (RC) systems require instantaneous response in the controlled device. RC systems have wide applications, including ad hoc networks. Imperfections in the wireless channel (noise and interference) result in randomly fluctuating latency in the response of the system. A lag occurs when the system latency exceeds a specified real-time threshold. System Lag Occurrence Probability (SLOP) is the probability of lag occurrence and is derived as the performance metric to characterize user experience in real-time radio control systems. Frequency hopping is used to mitigate interference effects. Uniform serial acquisition and N-state lock detection are used to simplify the derivation of SLOP. Simulation results are presented to verify the derivation of SLOP.

Keywords: radio control, frequency hopping, human response time.

1 Introduction

In remote control applications, a user controls a number of inputs which trigger a proportional movement in servomechanisms mounted in the controlled device. The servomechanisms in turn move the control surfaces of the device thus causing it to maneuver in its medium. Such applications are generally referred to as real-time Radio Control (RC) applications. Examples of real-time RC applications range from a simple model car Radio Control (shown in Figure 1) to highly mobile Unmanned Aerial Vehicles (UAVs) [1]. The controlled devices could form an ad hoc network. For instance, in a military scenario, UAVs that form a tactical ad hoc network could provide a backbone for ground communications when obstacles exist and provide effective direct communications between two ground devices. UAVs could also be used to actively sense environmental data for civilian applications [21].

J. Zheng, D. Simplot-Ryl, V.C.M. Leung (Eds.): ADHOCNETS 2010, LNICST 49, pp. 457–470, 2010.

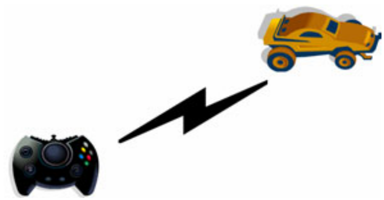

Fig. 1. An example of a radio control application

Real-time radio control requires that the system responds to user inputs with as little latency as possible. The time it takes the system to respond to a user input is here referred to as the **system latency**. System latency is mainly dependant on the inter-arrival time of control information from the transmitter to the receiver. Imperfections in the wireless channel (i.e. noise and interference) result in randomly fluctuating system latency. Therefore, the robustness of the wireless link established between the transmitter and the receiver has a major effect on the system latency.

A **lag** occurs when the system latency exceeds a specified real-time limit. *The user experience can be quantified by the probability of lag occurrence during a radio control session.* This paper derives a performance metric for real-time radio control applications based on the **probability of lag occurrence**.

Specifically, this paper studies the effect of using Frequency-Hopping Spread Spectrum (FHSS) technology on the user-experience. To the best of our knowledge, no previous work has been made to characterize packet-based FHSS RC applications with a performance metric such as the lag occurrence probability.

The rest of the paper consists of two main parts: First, we derive the lag-occurrence probability of a packet-based frequency hopping radio control system. Second, we present the simulation results.

2 System Lag Occurrence Probability

In control applications where the stimulus is an input from a *user*, the specified real-time limit becomes the **Human Response Time** (*HRT*) [3]. In such systems, a lag occurs when the system latency exceeds *HRT*. The *HRT* is estimated to be around *100ms* in real-time applications such as Voice-over-IP (VoIP).

We define the **System Lag Occurrence Probability (SLOP)**, as the probability of a lag occurring in the output of a real-time radio control system. This section derives *SLOP* for a frequency hopping radio control system.

Frequency hopping consists of two modes of operation: *acquisition* and *tracking*. To begin the derivation of *SLOP*, we'll first look at the interaction between acquisition and tracking modes. Initially the transmitter-receiver pair is out-of-lock and therefore acquisition is first initiated. The process responsible for performing acquisition is here referred to as the **acquisition engine**.

After some time T_{acq} the transmitter-receiver pair acquire lock on each other, acquisition is terminated, and tracking is initiated. The process responsible for performing tracking is here referred to as the **tracking engine**.

During tracking, the transmitter-receiver pair will try to remain in-lock with each other for as long as possible. After some time T_{lock}, various factors such as interference and time/frequency drift cause the transmitter-receiver pair to loose lock of each other and re-initiate acquisition. The time it takes the tracking engine to conclude that it has lost lock is called T_{LA} (i.e. Lock → Acquisition).

In such as system, *SLOP* becomes the probability of two events happening: 1) the event that the system enters acquisition mode and 2) the event that the system fails to acquire in the time remaining for an *HRT* to elapse:

$$SLOP = P\left(Loose\ lock \cap Fail\ to\ acquire\ in\ time\right)$$

Assuming that the tracking and acquisition processes are statistically independent, *SLOP* becomes:

$$SLOP = P_{LA}\, P\left(T > \left(HRT - T_{LA}\right)\right)$$

where P_{LA} is the probability of loosing lock, T_{LA} is the time it takes the tracking engine to conclude that it has lost lock, and T is the time it takes the acquisition engine to re-acquire lock.

P_{LA} and T_{LA} depend on the algorithm used for tracking, while the probability of successful acquisition depends on the algorithm used for acquisition. Therefore, to further derive *SLOP*, we must first analyze the acquisition and tracking engines in finer detail.

The communication system dealt with in this study is packet-based. As will be shown later, the main decision criterion used in the tracking and acquisition engines is based on the state of packet reception (correct or corrupt). The derivation of *SLOP* in the next two sub-sections therefore places emphasis on the average Packet-Error Rate (*PER*) parameter. The use of *PER* abstracts away the details of the underlying wireless channel and keeps the focus on the application layer design instead, as intended from the beginning.

The performance of frequency hopping systems under the presence of different types of interference is studied extensively in the literature. For example, [4] and [5] look at wideband and partial-band noise jamming. [6] looks at follower partial-band jamming, while [7] looks at partial-band multi-tone jamming. In this paper, however, the wireless channel model, the modulation type used, and the type of interferers present in the band are all abstracted in the average *PER* parameter. The average *PER* of a frequency hopping system is here calculated as:

$$PER = \frac{1}{q}\sum_{x=1}^{q} PER_x$$

where *PERx* is the *PER* of channel x and q is the total number of channels. This has the effect of averaging out any interference present in the band across all the channels.

2.1 Tracking Strategies

Various tracking strategies have been studied in literature, the most common being Tau-Dither [8], Delay-Lock [9], and Split-Bit Tracking loops. For packet-based systems, Modified Transmitted Reference [10] techniques seem to be more adequate as they are simpler to implement in firmware.

A Modified Transmitted-Reference tracking engine is more commonly referred to as an **N-state lock detector** [2]. This section derives P_{LA} and T_{LA} for an N-state lock detector. The basic principle of operation is shown in Figure 2.

The N-state lock detector contains *N* states, named *Lock_0* to *Lock_(N-1)*. Operation in one of those N states represents operation in the tracking mode. State *Lock_0* represents correct packet reception during tracking mode, while states from *Lock_1* to *Lock_(N-1)* represent corrupted packet reception during tracking mode. An additional state, called *Acq*, represents operation in the acquisition mode and is shown to the right of Figure 2.

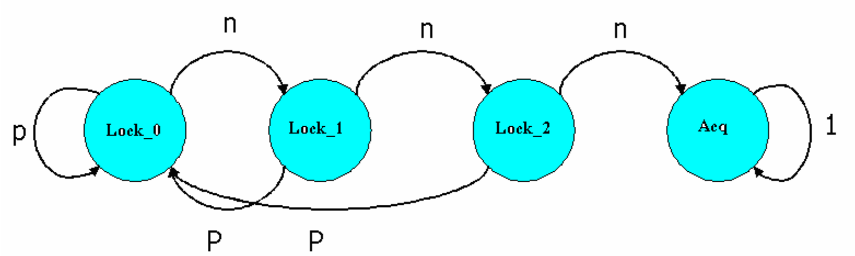

Fig. 2. State transitions in an N-State lock detector. N = 3.

The N-state lock detector begins operation in state *Lock_0*, where the transmitter-receiver pair is in lock and is exchanging packets correctly. During each hop, the transmitter-receiver pair exchanges a new packet. If the packet is received correctly, the lock detector generates a positive decision and moves to (or remains in) state *Lock_0*. If the packet is received corrupted, the lock detector generates a negative decision and moves one state to the right, towards state *Acq*.

This process is repeated at each of the states of the lock detector. If N consecutive negative decisions are generated, the lock detector terminates tracking and initiates acquisition. Positive decisions occur with probability *p*. Negative decisions occur with probability *n*.

To simplify the analysis of P_{LA} and T_{LA}, we assume that tracking is maintained when one complete packet is exchanged correctly. This has the following effects:

$$n = PER \qquad \text{and} \qquad p = 1 - PER$$

As mentioned previously, P_{LA} is the probability of the tracking engine concluding that it has lost lock. In other words, P_{LA} is the probability of terminating tracking and initiating acquisition. With an N-state lock detector, P_{LA} becomes the probability of exchanging N consecutive corrupted packets. Assuming that errors between packets are statistically independent, P_{LA} becomes:

$$P_{LA} = PER^{N}$$

T_{LA} is the time it takes the tracking engine to conclude that it has lost lock. With an N-state lock detector, T_{LA} becomes the time it takes for the exchange of N consecutive corrupted packets:

$$T_{LA} = N T_{dwell}$$

where T_{dwell} is the channel dwell time (hopping period). The above derives P_{LA} and T_{LA} for an N-state lock detector. To complete the derivation of *SLOP*, we will derive the probability of successful acquisition. The probability of successful acquisition depends on the algorithm used in the acquisition engine.

2.2 Acquisition Strategies

Acquisition strategies have been studied extensively in literature, the most common being serial [11] and parallel [12]. Other acquisition techniques include message passing [15] and adaptive-antenna array [16].

Serial acquisition techniques try to acquire lock between the transmitter and the receiver by searching the channels serially (i.e. one after the other) until the correct channel is found [11]. Uniform serial acquisition treats all channels as equally possible candidates for successful acquisition. Non-uniform serial acquisition favors some channels over others during its search.

Parallel acquisition techniques acquire lock by examining all channels simultaneously [12]. In practice, this is achieved by employing a bank of correlators, each tuned to one of the channels.

Serial acquisition is simple to implement but is relatively slow in acquiring lock. Parallel acquisition is complex to implement but is fast in acquiring lock. In practice, a hybrid of both techniques is usually employed to reach a relatively simple solution with relatively fast acquisition times. To simplify the analysis, this section will derive the required parameters using a uniform serial acquisition engine.

The mean acquisition time and its standard deviation for a uniform serial acquisition engine were derived in [13] and [14]:

$$T_{acq} = T_{dwell} \frac{2 + \left(2 - P_D\right)\left(q - 1\right)\left(1 + KP_{FA}\right)}{2P_D}$$

$$\sigma_{acq}^2 = T_{dwell}^2 \left[\left(1 + KP_{FA}\right)^2 q^2 \left(\frac{1}{12} - \frac{1}{P_D} + \frac{1}{P_D^2} \right) + 6q\left[K(K+1)P_{FA}\left(P_D - P_D^2\right) + \left(1 + KP_{FA}\right)\left(4 - 2P_D - P_D^2\right)\right] + \left[\frac{1 - P_D}{P_D^2}\right]\right]$$

where T_{acq} is the mean acquisition time, σ_{acq} is its standard deviation, T_{dwell} is the channel dwell time (hopping period), and q is the number of channels. Minimum T_{dwell} is equal to the time it takes the transmitter to hop to a new channel plus the time it takes to transmit a packet.

The probability of detection (P_D) is the probability of correctly terminating acquisition when the correct channel is being probed. The probability of a false alarm (P_{FA}) is the probability of erroneously terminating acquisition when an incorrect channel is being probed. Such an event is called a false alarm, and has a time-penalty K associated with it. K is the number of hops it takes the tracking engine to conclude that lock was falsely acquired and therefore re-initiate acquisition. Low Signal-to-Noise Ratio (SNR) and high channel distortion (fading and Inter-Symbol Interference ISI) drive P_D to 0 and P_{FA} to 1.

To simplify the analysis, we assume that acquisition ends when one complete packet is received correctly, which is a fair assumption in a packet-based communication system. This has the following effects:

$$P_D = 1 - PER \qquad \text{and} \qquad P_{FA} = 0$$

P_{FA} becomes negligible since it will be unlikely that an entire correct packet is falsely detected. Furthermore, assuming that $q >> 1$ and $K << q$, which are fair assumptions in practice, T_{acq} and σ_{acq} simplify to the following:

$$T_{acq} = qT_{dwell}\left[\frac{2-P_D}{2P_D}\right]$$

$$\sigma_{acq} = qT_{dwell}\sqrt{\left[\frac{1}{12} - \frac{1}{P_D} + \frac{1}{P_D^2}\right]}$$

It was shown that if we assume that the tracking and acquisition processes are statistically independent, $SLOP$ becomes:

$$SLOP = P_{LA}\, P\left(T > \left(HRT - T_{LA}\right)\right)$$

where P_{LA} is the probability of loosing lock, T_{LA} is the time it takes the tracking engine to conclude that it has lost lock, and T is the time it takes the acquisition engine to re-acquire lock.

Attempts to derive the Cumulative Distribution Function (cdf) and the Probability Density Function (pdf) of T were made in [17] and [18]. The model of the acquisition engine studied in the literature is complex however. False alarms, as well as other variables in the system tend to complicate the derivation of a proper pdf or cdf.

In this paper, we consider packet-based system operation, in which all decisions made in the acquisition engine are based on the state of packet reception during every hop interval: either correct packet reception or corrupt packet reception. In such a model, the probability of a false alarm becomes negligible. Essentially, every hop interval becomes a new Bernoulli trial. If a correct packet is received, the trial is declared a success and the acquisition process is terminated. Otherwise the trial is declared a failure and the acquisition process continues.

This reduces the acquisition time random variable to be defined simply as the number of consecutive failed Bernoulli trials made until a successful trial is made. By definition, T therefore becomes a geometric random variable [19]. The limiting form of T as the number of trials goes to infinity is the exponential random variable.

Therefore, if we model the acquisition time T as an exponential random variable, *SLOP* becomes:

$$SLOP = P_{LA}\left(1 - F_T\left(HRT - T_{LA}\right)\right)$$

where $F_X(x)$ is the cumulative distribution function of the exponential random variable X with parameter $\lambda = \frac{1}{E[X]}$. $F_X(x)$ is defined as [19]:

$$F_X(x) = 1 - e^{-\lambda x} \qquad x \geq 0$$

Where in this case,

$$\lambda = \frac{1}{T_{acq}} \qquad \text{and} \qquad x = HRT - T_{LA}$$

Therefore,

$$SLOP = PER^N\left(1 - F_T\left(HRT - NT_{dwell}\right)\right)$$

where

$$T_{acq} = \frac{1}{\lambda} = qT_{dwell}\left[\frac{2 - P_D}{2P_D}\right]$$

3 Simulation and Results

This section presents simulation and results. The purpose is to verify the derivations described in Section 2 and study the effect of various design variables on the performance of a frequency hopping real-time radio control system.

3.1 Packet-Based Frequency Hopping Simulator

The simulator is built using Java. It consists of a transmitter, a receiver, and a range of 20 to 80 frequency channels. Each channel is characterized with a Bit Error Rate (*BER*). Using its *BER*, each channel computes the corresponding Packet Error Rate (*PER*) according to the following formula [20]:

$$PER = 1 - (1 - BER)^b$$

where b is the length of a packet in bits.

Each channel contains a random number generator which corrupts packets at a rate equal to its computed *PER*. There are two types of channels in the simulator: Blocked and nonblocked. Blocked channels are ones with PER equal to 100%. Blocked channels are used to simulate the effect of powerful co-channel (same-channel) interferers. Non-blocked channels are ones with PER equal to 15%. A PER of 15%

corresponds to a BER of 1e-3 and a packet length of 160 bits. These are typical values used in packet-based wireless communication systems with Additive White Gaussian Noise (AWGN) channels.

The transmitter hops pseudo-randomly across all the channels and generates a single packet for each hop. The hopping pattern is uniform. Each generated packet is passed to the channel at which the transmitter currently resides. The channel in turn either corrupts the packet or passes it to the receiver uncorrupted, depending on the *PER* of the channel.

The receiver implements uniform serial acquisition and N-State lock detection. The Finite State Machine (FSM) of the receiver is shown in Figure 3**.** The acquisition engine consists of a single state called *ACQ*. The lock detector consists of two states called *LOCK_0* and *LOCK_N*. The receiver FSM defines the interaction between these three states.

The simulator runs by generating a programmed number of timer ticks. At each tick, the transmitter hops to a new channel and transmits a packet at that channel. The channel corrupts the packet or passes it to the receiver un-corrupted depending on the *PER* of the channel. Two new inputs are therefore available to the receiver with each new tick:

1. The newly transmitted packet (either corrupted or un-corrupted), and
2. The channel at which the new transmission occurred.

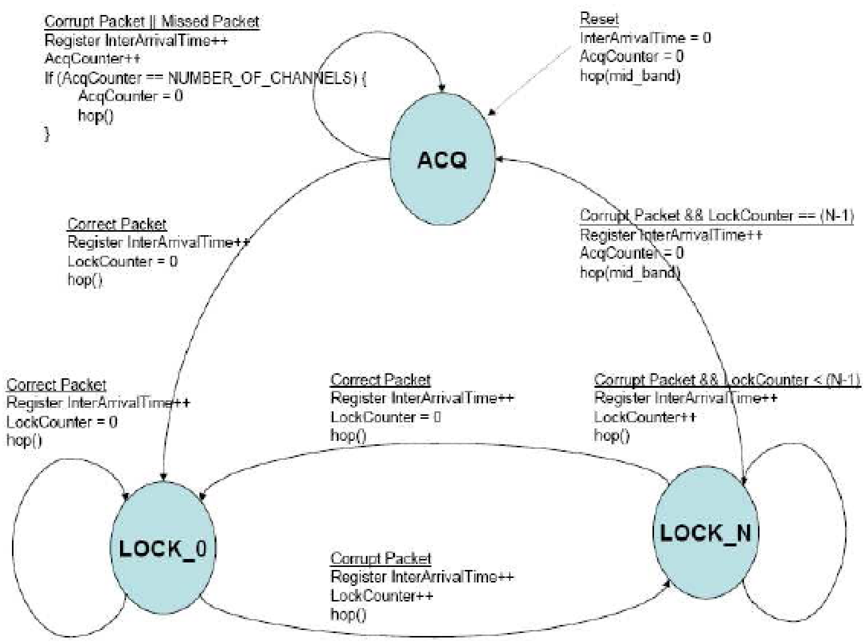

Fig. 3. Receiver FSM

The receiver generates one of three events depending on the two inputs from the simulator:

1 **Non-Corrupted-Packet-Received Event** is the event that the receiver's current channel is equal to the transmitter's current channel and the received packet *was not* corrupted by the channel.
2 **Corrupted-Packet-Received Event** is the event that the receiver's current channel is equal to the transmitter's current channel but the received packet *was* corrupted by the channel.
3 **Missed-Packet Event** is the event that the receiver's current channel is not equal to the transmitter's current channel and therefore the transmitted packet was entirely missed, regardless of the state of the packet itself (corrupted or uncorrupted). This event can only occur during the acquisition state (*ACQ*) since this is the only state during which the transmitter and the receiver can be out-of-lock.

The inter-arrival time of correctly received packets is measured in ticks (time normalized to the channel dwell time) and is logged with every new tick. *Non-Corrupted-Packet-Received* events reset the inter-arrival time, while *Corrupted Packet Received* events and *Missed Packet* events increment the inter-arrival time. The inter-arrival time of each packet during a control session is plotted and analyzed against different design variables for performance evaluation.

A lag is registered every time the inter-arrival time exceeds *HRT*. The reception of a correct packet represents an opportunity for a lag to occur. *SLOP* is computed at the end of a simulation session as the ratio of the number of lags to the number of correct packets received. *SLOP* is plotted and analyzed against different design variables for performance evaluation, which will be presented in the next sub-section.

3.2 Simulation Results

Two sets of simulations were carried out. The first simulation plots *SLOP* against the probability of detection (P_D). The purpose of the first set is to verify the theoretical results derived in Section 2. The second set plots the average inter-arrival time of correctly received packets against different design variables. The purpose of the second set is to study the effect of different design variables on the performance of a packet-based frequency hopping system.

Results for the first simulation set are shown in Figures 4 and 5. At low probability of detection, the system latency increases and therefore the probability of lag occurrence increases, and vice versa. Results are better for larger N (the number of states in an N-state lock detector) since a larger N improves tracking and therefore decreases the overall system latency. Results show that the simulation is in perfect agreement with theory. This validates the derivations made in the previous section and serves as evidence for correctly modeling the acquisition time as an exponential random variable.

Figure 5 demonstrates the simulation plots *SLOP* against *N*. As *N* increases, tracking is improved and therefore overall system latency decreases. This in-turn decreases *SLOP*. Naturally, performance is better at lower higher probability of detection than it is at lower probability of detection. As can be seen in the figure, results show that the simulation is in perfect agreement with theory. This validates the derivations made in the previous section and serves as evidence for correctly modeling the acquisition time as an exponential random variable.

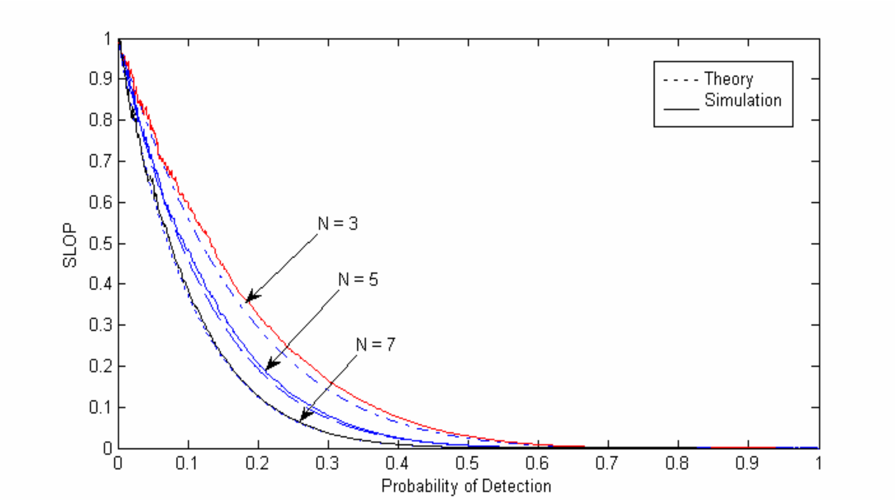

Fig. 4. SLOP vs. P_D (q = 40, HRT = 100ms)

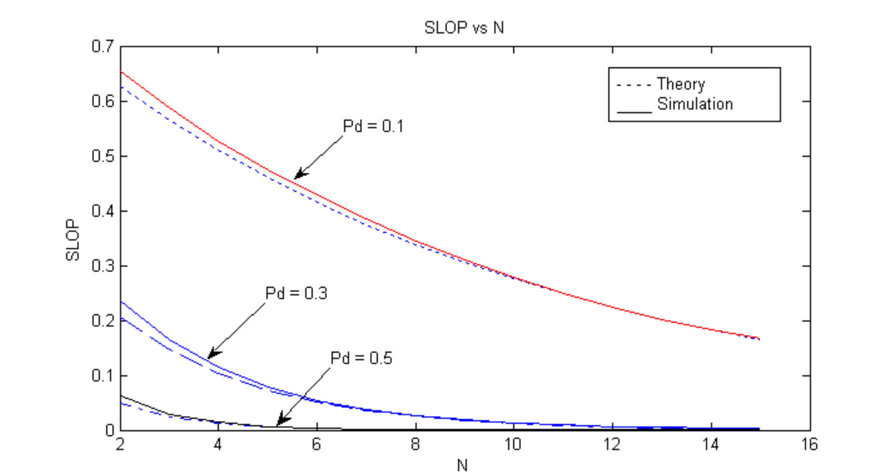

Fig. 5. SLOP vs. N (q = 40, HRT = 100ms)

The next set of simulations present the effect of different design variables on the performance of a frequency hopping radio control system. The design variables investigated include:

- **q:** The number of channels used for hopping
- **J:** The number of blockers (co-channel interferers) present in the band
- **N:** The number of states in an N-state lock detector.

The performance metric considered here is the average inter-arrival time of correctly received packets, which represents the average system latency (response).

The first simulation studies the effect of blocking. Figure 6 shows the results. The y-axis represents the normalized inter-arrival time, while the x-axis represents the partial band blocking ratio. This is the ratio of the number of blocked channels to the total number of channels. Simulation is carried out with a uniform serial-search acquisition engine and a 10-State lock detector.

Results in Figure 6 depict that performance begins to dramatically degrade when 40% or more of the band is being blocked, regardless of the number of channels being used in hopping. This is expected since a high partial blocking ratio causes the system to frequently loose lock and enter acquisition mode. Acquisition is worse for higher number of channels, therefore performance degradation is worse for 80 channels than it is for 40 and 20.

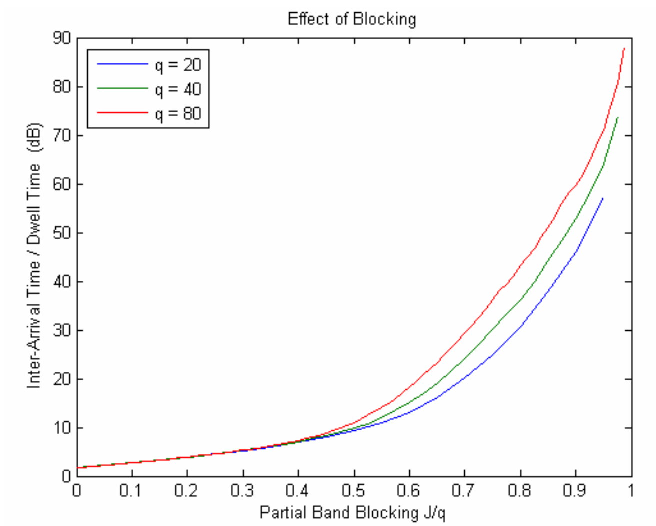

Fig. 6. Effect of blocking. N = 10.

The second simulation investigates the effect of N-State lock detection on performance. Figure 7 shows the results. The y-axis represents the normalized inter-arrival time, while the x-axis represents the number of states in an N-State lock detector. The simulation is carried out at a high partial band blocking ratio (J/q = 75%). Uniform serial search acquisition is used.

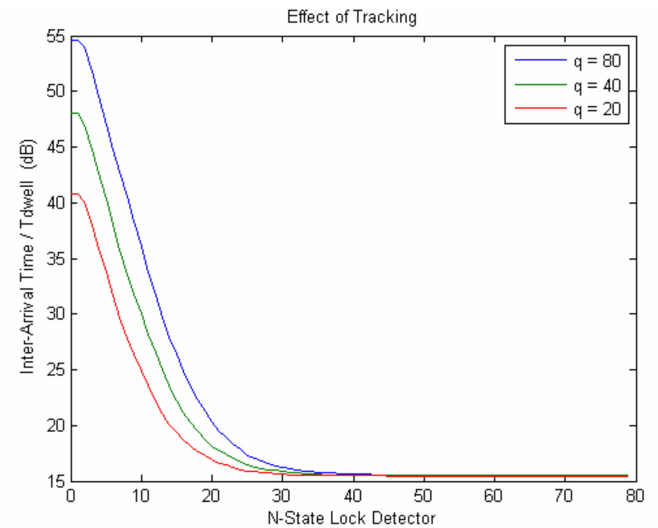

Fig. 7. Effect of Tracking. J/q = 0.75.

Results in Figure 7 show the importance of maximizing the number of states in the N-state lock detector. As N increases, performance improves, to the point where performance of systems operating over different number of channels converges at approximately $N > 40$. This is expected since higher N leads to a lower number of acquisition attempts and therefore smaller average inter-arrival time.

The third simulation studies the effect of varying the number of hopping channels on interference. Figure 8 illustrates the results. Interference resistance in this paper is calculated as the ratio of the number of blocked channels (J) to the total number of channels (q). This is used to indicate how much interference resistance there is in a

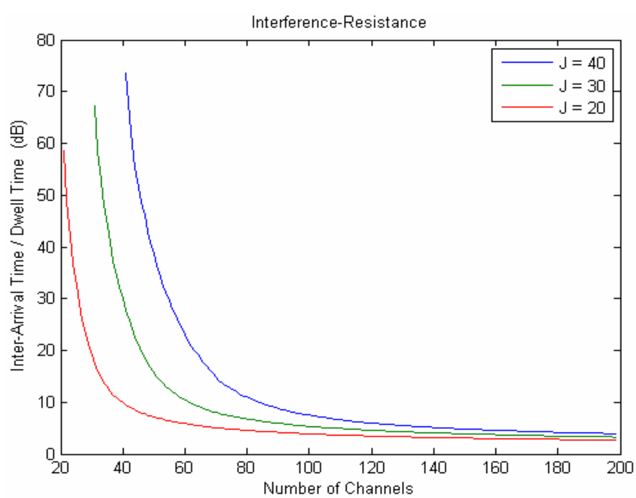

Fig. 8. Interference resistance due to increasing number of channels, N = 10

frequency hopping system. Lower ratio means more free channels and therefore better communication. The y-axis represents the normalized inter-arrival time, while the x-axis represents the number of channels used in hopping. The simulation is carried out with uniform serial search acquisition engine and a 10-state lock detector.

Results show that increasing the number of channels can be used to effectively combat interference. For a given number of blocked channels, performance improves as the number of channels increase. This happens because the average *PER* of the system decreases as q increases.

Another way to explain the results in the Figure 8 is in terms of acquisition and tracking. As the number of free channels in the system increases, the chances of finding a free channel for proper communication increases and therefore the system remains in tracking mode longer. The number of acquisition attempts therefore decrease and the overall performance of the system improves.

4 Conclusions and Future Research

Real-time radio control systems have wide applications, including mobile ad hoc networks. This paper used Frequency Hopping Spread Spectrum technology to mitigate the interference effects for real-time radio control systems. System Lag Occurrence Probability (SLOP) was used as the performance metric to characterize user experience. Simulations have been conducted and the results validated the derivation of SLOP.

In general, a frequency hopping system switches periodically between two modes: acquisition and tracking. Many acquisition and tracking engines are found in the literature. Uniform serial acquisition and N-state lock detection were adopted, because they simplify analysis and satisfy the conditions required for the derivation of *SLOP*. In practice, other acquisition and tracking engines such as those employed in Adaptive Frequency Hopping systems can be used to improve performance. SLOP can be derived based on the new proposed schemes in a similar manner to the derivation shown in Section 2. The use of more complex acquisition and tracking engines and the analysis of their effects on the performance of radio control systems is left for future work.

Acknowledgements

The authors thank MITACS, Canada and Cistel Technology for their support of the research.

References

[1] Abatti, J.M.: Small Power: The Role of Micro and Small UAVs in the Future. Technical Report AU/ACSC/6697/2005-04, Centre for Strategy and Technology, Air War College, Air University (2005)

[2] Glisic, S., Vucetic, B.: Spread Spectrum CDMA Systems for Wireless Communications. Artech House Inc., Boston (1997)

[3] Mowbry, G.H., Gebbard, J.W.: Man's Senses as Informational Channels. In: Sinaiko, H.W. (ed.) Human Factors in the Design and Use of Control System, pp. 115–149. Dover, New York (1961)

[4] McGuffin, B.F.: Jammed FH-FSK Performance in Rayleigh and Nakagami-M Fading. In: Proc. of IEEE Millitary Communications Conf., pp. 1077–1082 (2003)

[5] Teh, K.C., Kot, A.C., Li, K.H.: Partial Band Jammer Suppression in FFH Spread-Spectrum System Using FFT. IEEE Trans. on Vehicular Technology 48(2), 478–486 (1999)

[6] Hassan, A.A., Start, W.E., Hershey, J.E.: Frequency-Hopped Spread Spectrum in the Presence of a Follower Partial-Band Jammer. IEEE Trans. on Communications 41(7), 1125–1131 (1993)

[7] Miller, L.E., Lee, J.S., French, R.H., Torrieri, D.J.: Analysis of an Antijam FH Acquisition Scheme. IEEE Trans. on Communications 40(1), 160–170 (1992)

[8] Peterson, R.L., Ziemer, R.E., Borth, D.E.: Introduction to Spread Spectrum Communications. Prentice Hall, New Jersey (1995)

[9] Wilde, A.: Extended Tracking Range Delay Lock Loop. German Aerospace Research Establishment, Germany. In: Proc. of IEEE ICC, pp. 1051–1054 (1995)

[10] Dominique, F., Reed, J.H.: Robust Frequency Hop Synchronization Algorithm. Electronics Letters Online 32(16), 1450–1451 (1996)

[11] Polydoros, A., Weber, C.: A Unified Approach to Serial Search Spread Spectrum Code Acquisition – Part I: General Theory. IEEE Trans. on Communications 32(5), 550–560 (1984)

[12] Chawla, K., Sarwate, D.: Parallel Acquisition of PN Sequences in DS/SS Systems. IEEE Transactions on Communications 42(5), 2155–2164 (1994)

[13] Holmes, J.K., Chen, C.C.: Acquisition Time Performance of PN Spread-Spectrum Systems. IEEE Trans. on Communications COM-25(8), 778–783 (1977)

[14] Dicarlo, D.M., Webber, C.L.: Multiple Dwell Serial Search: Performance and Application to Direct Sequence Code Acquisition. IEEE Trans. on Communications 31(5) (1983)

[15] Zhu, M., Chugg, K.: Iterative Message passing Techniques for Rapid Code Acquisition. In: Proc. of IEEE Millitary Communictions Conf., pp. 434–439 (2003)

[16] Dlugos, D., Scholtz, R.: Acquisition of Spread Spectrum Signals by an Adaptive Array. IEEE Trans. on Acoustics, Speech, and Signal Processing 137(8), 1253–1270 (1989)

[17] Dicarlo, D.M., Weber, C.L.: Statistical Performance of Single Dwell Serial Synchronization Systems. IEEE Trans. on Communications Com-28(8), 1382–1388 (1980)

[18] Jovanovic, V.M.: On the Distribution Function of the Spread-Spectrum Code Acquisition Time. IEEE Journal on Selected Areas in Communications 10(4), 760–769 (1992)

[19] Garcia, A.L.: Probability and Random Processes for Electrical Engineering, 2nd edn. Addison-Wesley Publishing Company, Reading (May 1994)

[20] Han, J., Lanzinger, D., Sklar, D.: Assessing the Performance of Packet Retransmission Schemes Over Satellite Link. In: Proc. of IEEE Aerospace Conf. (2006)

[21] Frew, E.W., Dixon, C., Elston, J., Stachura, M.: Active Sensing by Unmanned Aircraft Systems in Realistic Communication Environments Networked Robotics. In: Proc. of IFAC Workshop on Networked Robotics (2009)