Using Simulation to Find Optimal Parameters of Clinical Check-In

Rahul Isaac

Victoria Plourde

Introduction

Many businesses have been making the movement toward using electronic kiosks to aid in the check-in procedure for their customers. This check-in system has become the new norm for service providers such as airlines, and now many doctors' offices and health clinics are also implementing the use of electronic kiosks for checking in patients.

The kiosk check-in system has obvious benefits for the service provider. They help to ease the workload of front desk receptionists by having patients input their personal information directly to the computer. This helps to streamline the data source and ease of access to one's medical history.

However, while the benefits for the service provider are many, the kiosk check-in system takes more time for the customer. This is because the customer must input their own information on an unfamiliar system. The aim of this study is to investigate whether the kiosk check-in system provides any additional benefit to the customer, such as decreased waiting time or increased customer throughput.

Model

This paper will compare two service systems. The first system, which we will refer to as the standard check-in system, includes customers arriving to a queue with N servers. These N servers represent the number of receptionists available to check-in patients. The second system which we will refer to as the kiosk check-in system, includes customers arriving to a queue with

K servers, representing the number of check-in kiosks in the clinic. For comparison purposes, it is assumed that $K > N$.

Our model assumes that all arrivals to the queue are Poisson distributed with rate λ. Service times are exponentially distributed with rate μ_i for stations $i = 1, 2$. We assume infinite queue capacity, so that there is no blocking between servers. Furthermore, we assume that there is no abandonment of customers once they enter the system.

Procession through the standard check-in system proceeds as follows: A customer arrives to the system with rate λ. If one of N servers is available, the customer moves to that server and the check-in system begins. If no servers are available, the customer joins the queue and waits until a server becomes available. Once the check-in process has been completed, the customer continues on to the final service stage. Again, if the final server is available, the customer enters the server and service begins. If the final server is not available, the customer joins a queue and waits for the server to become available.

Similarly, customers arrive to the kiosk check-in system with rate λ. If one of K kiosks is available, the customer enters the kiosk and begins the self check-in process. If no kiosks are available, the customer joins the queue to wait for the next available kiosk. Once the customer has checked in, we assume that with probability p, the kiosk check-in was accurate enough to proceed to the final service station. With probability $1-p$, there was an error in the self check-in process and the customer needs to redo their check-in. If this occurs, they are directed to the receptionist check-in desk where they will be checked-in by a receptionist.

It is assumed that the arrival rate, receptionist check-in rate, and final service rate for the two models are the same. The service rate for the kiosks is smaller than the service rate for the receptionist, implying that it takes a customer longer to check themselves in than if a receptionist were to do so. This is justified by the fact that a receptionist would be more familiar with the check-in procedure, forms, and system, so they would be faster at checking a patient in.

Finally, we assume that the cost per receptionist is $33,000 per year, in keeping with the median clinic receptionist wage. The cost per kiosk was assumed to range from $2,000 to $6,000. There is a cost incentive to purchase multiple kiosks, as the cost per kiosk decreases if you buy kiosks in bulk.

Parameters

The following parameters were used to analyze the model:

A. Standard Check-in System

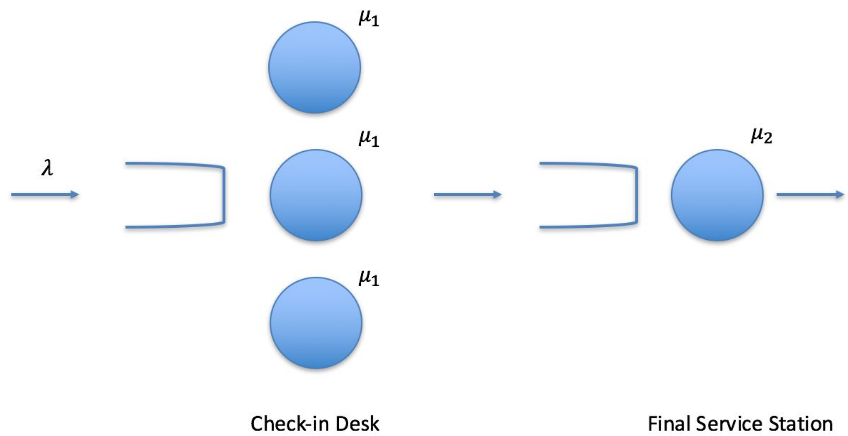
$\lambda = \frac{1}{6}$

$\mu_1 = \frac{1}{15}$

$\mu_2 = \frac{1}{5}$

$N = 3$

The rate of arrival to the standard check-in system is $\lambda = \frac{1}{6}$, or the arrival rate of a customer every 6 minutes. The receptionist check-in system has a service rate of $\mu_1 = \frac{1}{15}$, or an average of 15 minutes per customer. The second service system has a rate of $\mu_2 = \frac{1}{5}$ or an average of 5 minutes per customer. There are three receptionists available to check customers in.



B. Kiosk Check-in System

$\lambda = \frac{1}{6}$

$\mu_K = \frac{1}{20}$

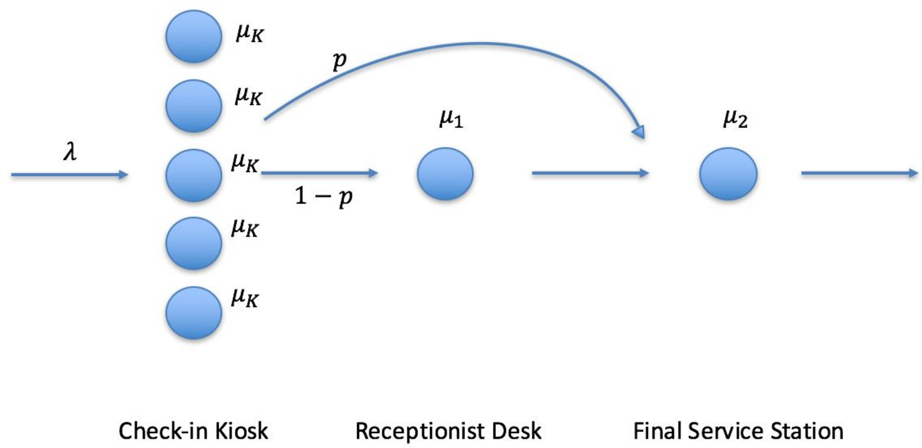
$\mu_1 = \frac{1}{15}$

$\mu_2 = \frac{1}{5}$

$p = 0.9$

$K = 5$

The rate of arrival to the kiosk check-in system is the same as the arrival rate to the standard check-in system of $\lambda = \frac{1}{6}$. The kiosk check-in system has a service rate of $\mu_K = \frac{1}{20}$ or an average of 20 minutes per customer. There are 5 kiosks available for customer check-in. With probability of $p = 0.9$ the customer moves on to the final service station. With probability of $1 - p = 0.1$ the customer has to re-do the check in process with the receptionist check-in desk. The receptionist check-in system has the same service rate as the standard check-in system, $\mu_1 = \frac{1}{15}$. The final service station has a service rate of $\mu_2 = \frac{1}{5}$, which is the same final service rate for the standard check-in system.



Methods

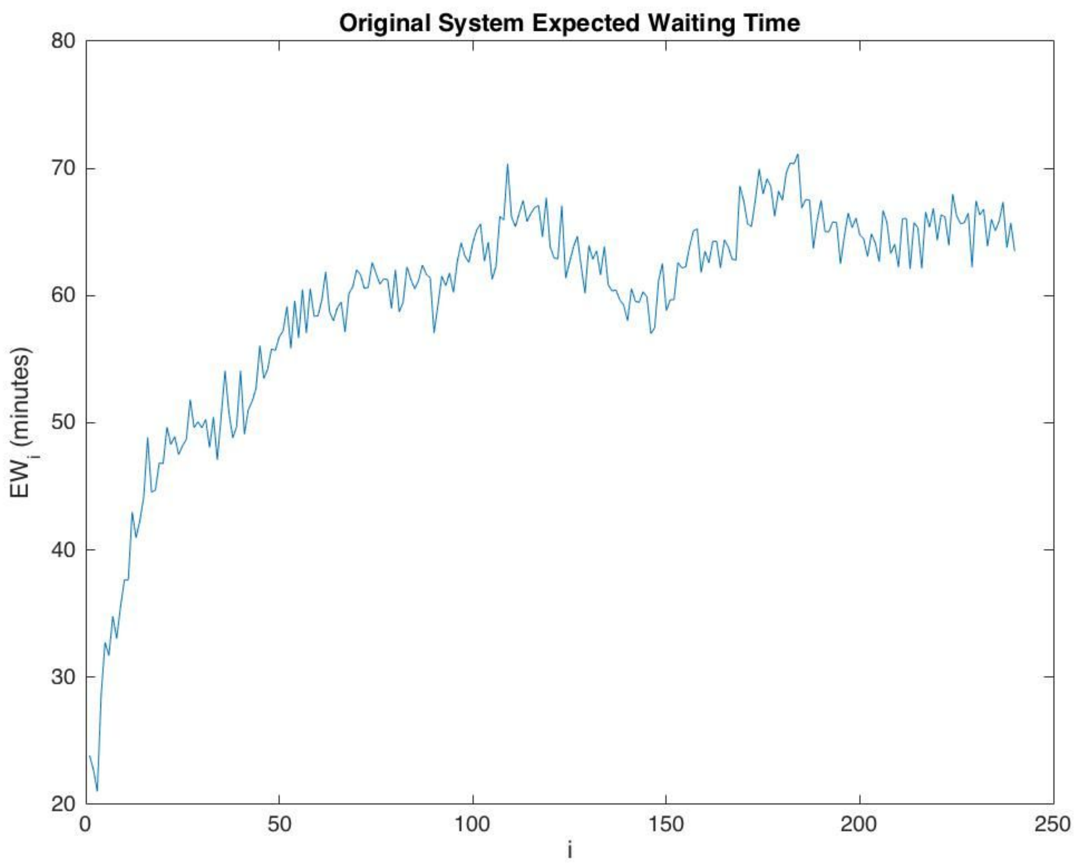
A Matlab simulation to determine the performance of the standard check-in system and kiosk check-in system was run to compare the two systems. A simulation budget of 200 was used and each simulation was run to determine the waiting time of the first 500 customers to enter the system.

The common random numbers approach described in class was used to reduce variance between the two systems. Uniform random numbers were generated and the inverse transform method was used to obtain exponentially distributed random variables. The two systems shared the same interarrival times, receptionist check-in service times, and final service system times. By maintaining the same arrival and service times between the two systems, we ensured that comparing the total service times of individual customers would be done in a meaningful way, independent of the randomness of their final service and arrival times.

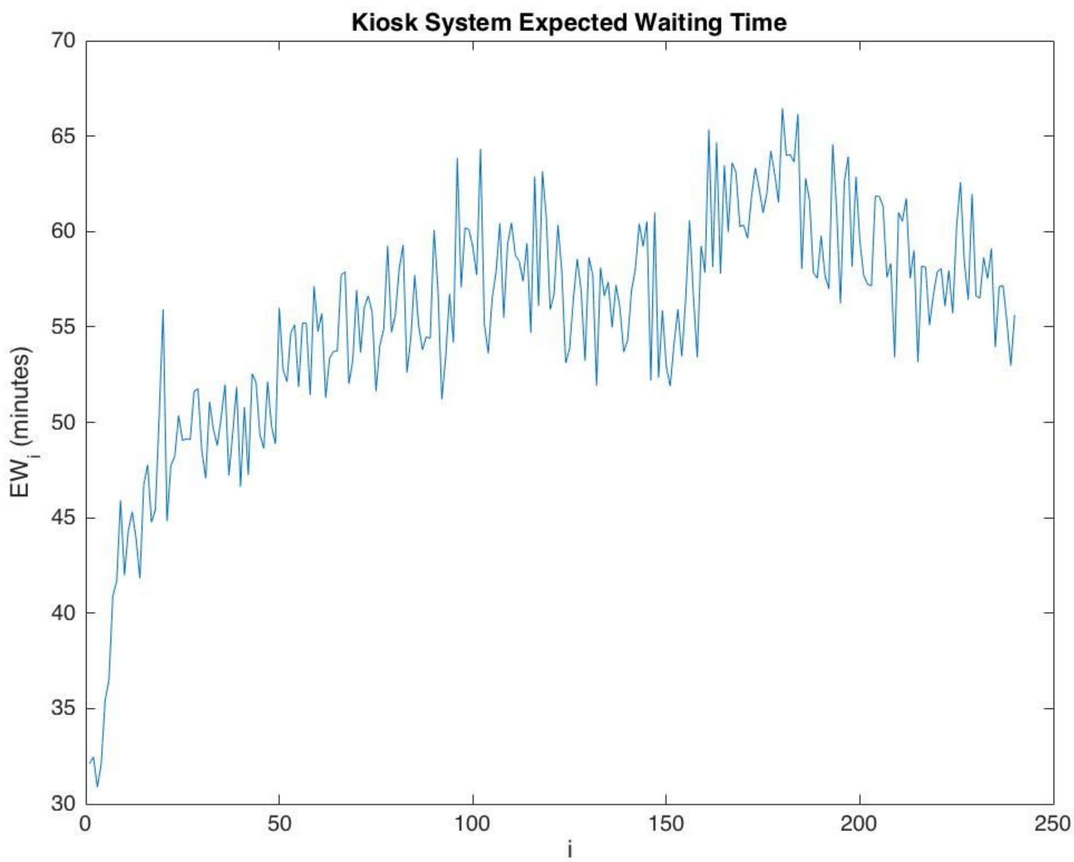
The optimal number of kiosks for the kiosk check-in system was determined by running the simulation for 1 to 100 servers and determining the expected wait time for each instance. The number of kiosks was determined to be the number at which an additional kiosk would not add a significant improvement in expected wait time for customers.

Analysis and Recommendations

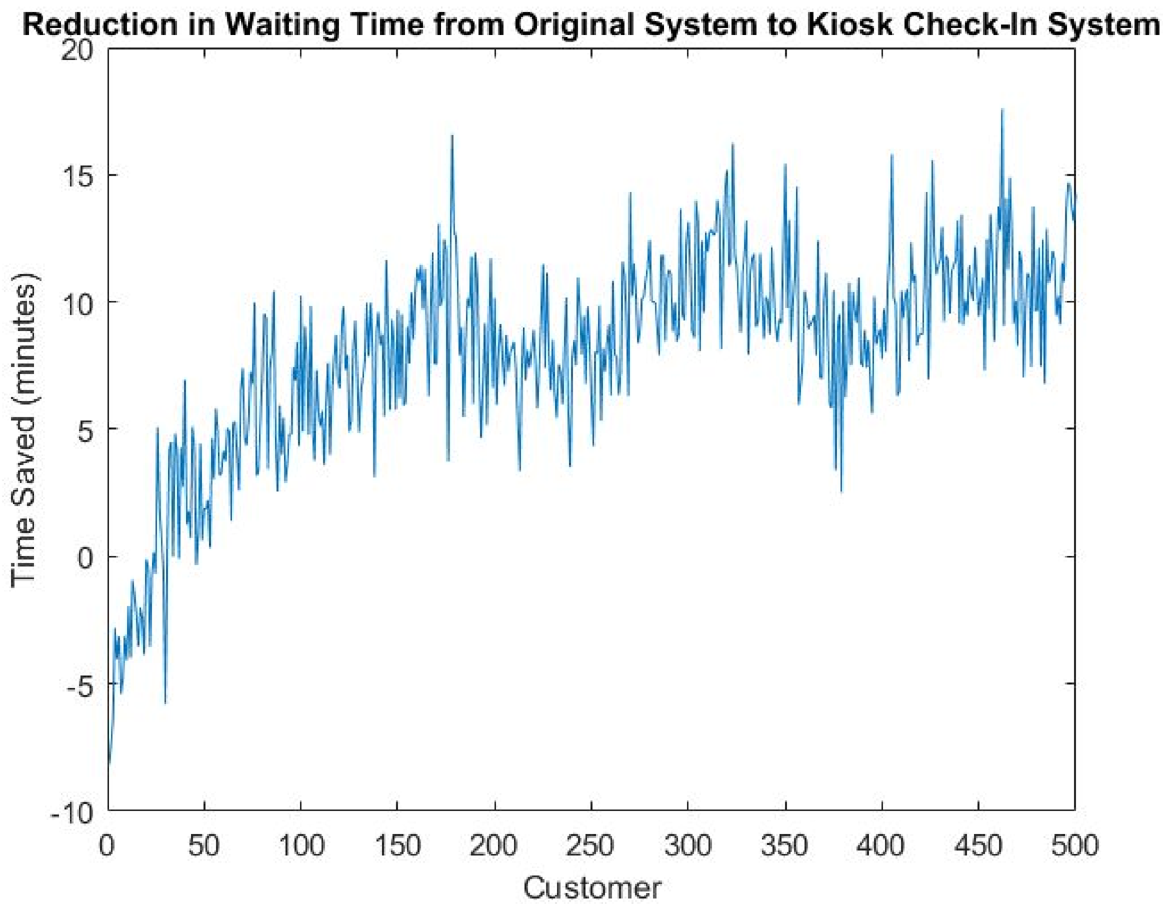
The standard check-in system had an expected waiting time of 59.69 minutes with variance of 77.77. As the simulation began, the first 50 customers experienced waiting times of under 50 minutes. Once more than 50 customers had entered the system, the waiting time continued to increase, up to 70 minutes, and reached a steady state of around 65 minutes.



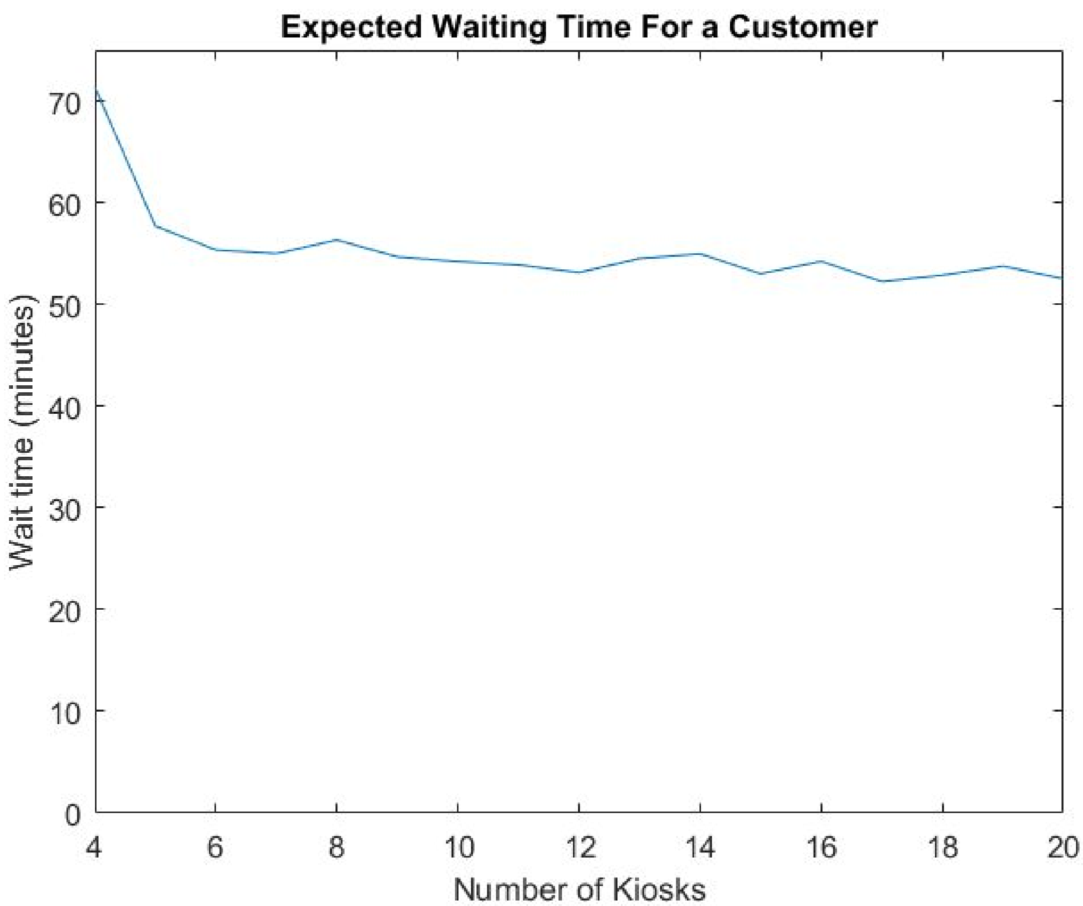
The kiosk check-in system had an expected waiting time of 55.39 minutes with variance of 37.49. The kiosk check-in system has an expected waiting time less than that of the standard check-in system, and achieves a smaller variance as well. The expected waiting time per customer increases more rapidly than that of the first system. The first 50 customers experience an expected waiting time of around 50 minutes or slightly higher. However, the kiosk check-in system achieves a steady state of around 55-60 minutes, which is less than that of the standard check-in system.



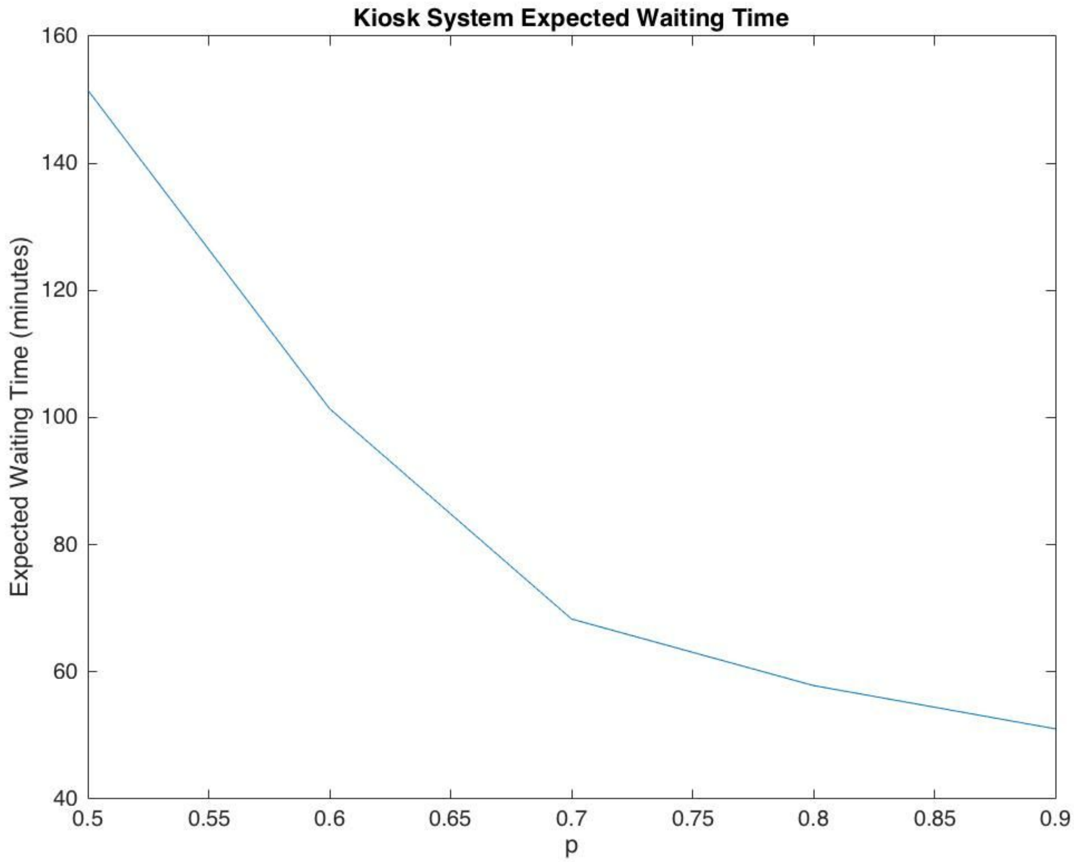
When comparing the difference in wait times for each customer, it was found that on average over 97% of customers had shorter waiting times in the kiosk check-in system compared to the standard system. There was a mean reduction in waiting time of 8.05 minutes. Initially customers might have a shorter sojourn times in the standard system, but as more customers entered both the system, the kiosk check-in system was better able to handle these customers in an efficient way.



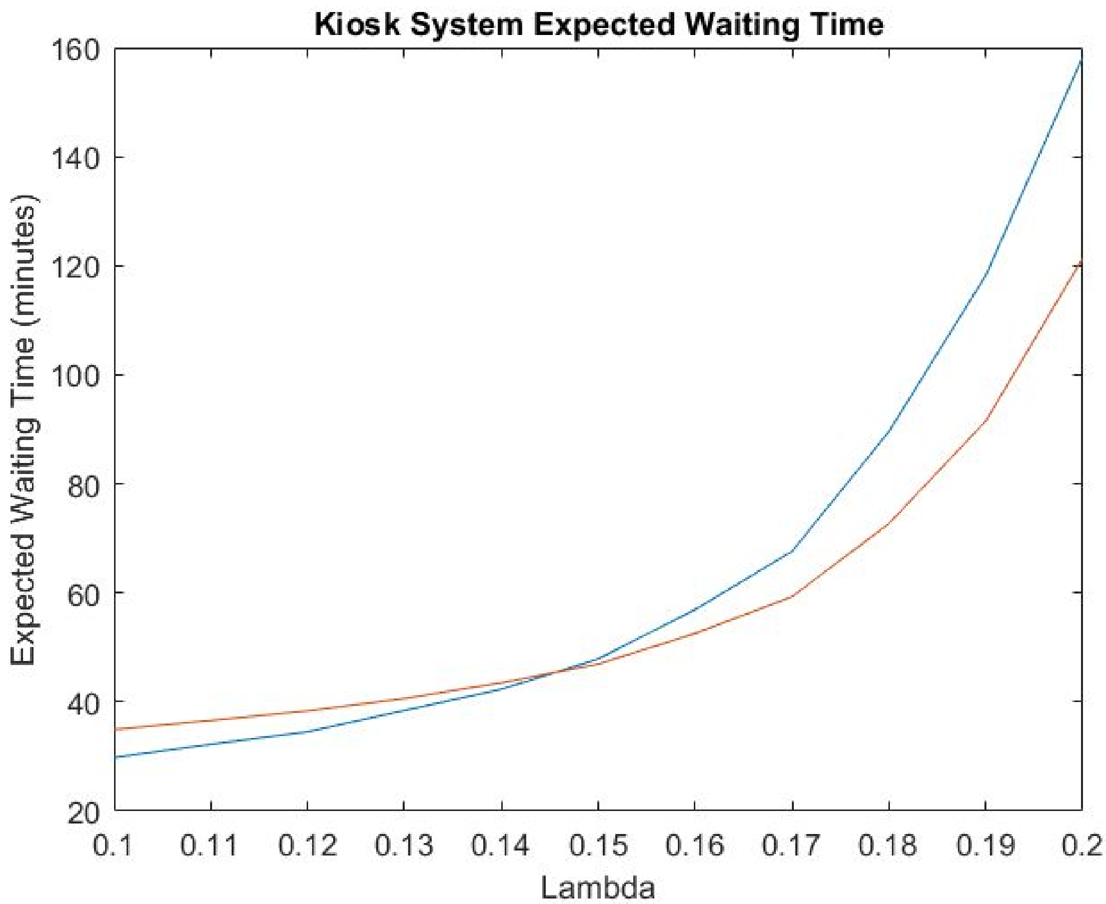
Sensitivity analysis was performed on the kiosk check-in system to determine parameter sensitivity. The number of kiosks was varied from 1 to 100 and expected wait times were calculated for each. When the number of kiosks was below four, the system experienced backlogging and expected waiting times were extremely high, starting at 30 minutes for the first customer and steadily increasing to over 400 minutes for the last customer. At four kiosks, the check-in system did not become overly crowded, but the expected waiting time was around 70 minutes, which is higher than the expected waiting time of the standard check-in system. This is because the number of primary servers, or the servers that a customer immediately goes to, in each system are not much different. Whereas the standard check-in system has three servers, the kiosk check-in system has only one more. Additionally, each kiosk server has an expected service time that is greater than that of the receptionist check-in service time and each customer experiences a probability of *p* that they need to repeat the check-in system.



When an additional kiosk is added to the system so that the kiosk check-in system has five servers, the benefit of the kiosk check-in system is evident. The service times drop to around 55 minutes, which is less than the expected waiting time of the standard check-in system. As additional kiosks are added past five kiosks, the expected waiting time decreases slowly, but does not decrease significantly enough to justify the addition of another kiosk. Therefore, it is recommended to that five kiosks be used if minimizing cost is the primary objective.



The probability that a customer would not need to re-do the check-in process after using a kiosk was varied from 0.5 to 0.9. When the probability that a customer needs to repeat the process is 50%, the expected wait time for the kiosk check-in system once again explodes, increasing with every customer to over 400 minutes for the last customer. The system performance increases as the probability increases. When the probability is 0.75, the expected wait time decreases to around 67.56 minutes, which comparable with the expected waiting time of the standard check-in system. As the probability increases further, the kiosk system's expected waiting time decreases to be less than or equal to the performance of the standard check-in system.



Finally, the arrival rate to the system was varied from 0.1 to 0.2 for both systems, corresponding to an arrival every 10 minutes to an arrival every 5 minutes. When the arrival rate is smaller, there is less congestion in the system. This corresponds to a better performance in the standard check-in system, because here the queues in the system will be very short. Since the expected time at the receptionist is shorter than the expected waiting time at the kiosk, we should expect the sojourn time of a customer to be less in the standard system than in the kiosk system at these lower arrival rates. However, as the arrival rate increases, the system becomes more congested, with queues often being longer. This corresponds to the kiosk check-in system performing better than the standard check-in system as a single customer having a longer service time will have less of an adverse effect on the rest of the customers. This intuition on the performance of the kiosk system is confirmed by the simulated waiting times. Thus the kiosk system is better equipped to handle a higher rate of customer arrival to the system.

Assuming that each kiosk costs $6,000 to purchase and install, the recommended 5-kiosk system with one additional receptionist to handle repeated check-ins due to kiosk error would cost $63,000 per year. The standard check-in system with three receptionists would cost $99,000 per year. Therefore, as long as the yearly maintenance costs are less than the $36,000 difference between the two systems, the kiosk check-in system has clear monetary benefits. In fact, the kiosks could be replaced each year and the cost of implementing the kiosk check-in system would still be lower than the standard check-in system.

This study assumes that all arrival times are Poisson distributed and service times are exponentially distributed. This may not be a realistic assumption. Clinics often experience peaks in arrival rates during critical times of the day. In the future, this analysis could be extended to include non-homogeneous Poisson arrivals. Additionally, real data could be used to estimate arrival and service time distributions, or actual arrival times and service times could be used to evaluate how the system performs under more realistic distributions and parameters. Furthermore, future research should focus on analyzing the additional costs of implementing the kiosk check-in system. This could include estimating the cost of maintenance per year for each kiosk. This calculation would give a more accurate comparison of the costs between the two systems. Finally, many clinics have multiple doctors available. This could also be integrated into the system as multiple final servers.

Conclusion

Our analysis determined that the kiosk check-in system outperformed the standard check-in system. The kiosk check-in system was cheaper to implement and provided the additional benefit of lower expected wait times per customer. However, this performance was heavily dependent on the accuracy of the kiosk check-in procedure. If the kiosk check-ins must be redone more than 25% of the time, there is no additional benefit to the kiosk check-in system in terms of expected wait times per customer. Furthermore, if the system is not likely to become congested, there is no wait time benefit for the kiosk check-in system as well. The kiosk check-in system is recommended for systems that are highly congested and with the caveat that the kiosk must be at least 75% accurate when checking a customer in.

Appendix - Source Code

```
%% Common Random Numbers
clear all;
clc;
clf;
tic;
lambda = 1/6;
mu1 = 1/15;
muk = 1/20;
mu2 = 1/5;
p = .75;

N = 500;
NSim = 200; % Number of simulation runs

NS = 3; %Number of servers at station 1
NK = 5; %Number of kiosks


%Uncomment if we want to vary parameters
%for NK = 3:20
%for lambda = .1:.01:.2

W1 = zeros(N,1); % Initialize waiting times vector for the
queues
W2 = zeros(N,1);

PTk = [];
PTs = [];
PTQ1 = [];
SJ1 = []; %Sojourn time for Alt 1
SJ2 = []; %Sojourn time for Alt 2

for k=1:NSim

     %Generate interarrival times and service times
     A = zeros(N,1); %Arrival times
     S1 = zeros(N,1); %Service times at station 1
     SK = zeros(N,1); %Service times at kiosks
     S2 = zeros(N,1); %Service times for station 2

     %Initialize values
     A(1) = -log(rand)/lambda;
     A(N+1) = Inf;
```

```
S1(1) = -log(rand)/mu1;
SK(1) = -log(rand)/muk;
S2(1) = -log(rand)/mu2;
for i=2:N
A(i) = A(i-1)-log(rand)/lambda;
S1(i) = -log(rand)/mu1;
SK(i) = -log(rand)/muk;
S2(i) = -log(rand)/mu2;
end

t = 0;
NA = 0;
NDS = 0;
ND2 = 0;
S = Inf(NS,1); %Departure times from servers

tD2 = Inf;

Output1 = [];
tA = A(1); %Set time of first arrival
QS = [];
Q1 = [];
Q2 = [];
SID = 1:NS;

Bs = [];
while ND2 < N

if tA <= min([S;tD2])
     t = tA;
     NA = NA + 1;
     if sum(S<Inf) < NS %if number of customers in system
is less than the number of servers
          idx = find(S==Inf,1);
          QS(idx) = NA;
          S(idx) = t + S1(NA);
          Bs = [Bs; NA SID(idx) t S(idx)];
     else %all servers full
          QS(end+1) = NA;
     end
     if NA < N
          tA = A(NA+1);
     else
          tA = Inf;
     end
```

```
          Output1 = [Output1; NA t 0 0];

     elseif min(S) <= tD2
          [t, I] = min(S);
          Output1(QS(I),3) = t;
          NDS = NDS + 1;

          Q2 = [Q2 QS(I)];
          if length(Q2) == 1
               tD2 = t + S2(Q2(1));
          end

          if length(QS) > NS %If there were more people than
kiosks
               QS(I) = QS(NS+1); %Next person in line goes to
the vacant kiosk
               QS(NS+1) = []; %Vacate the spot they left
               S(I) = t + S1(QS(I)); %Assign departure time

               Bs = [Bs; QS(I) SID(I) t S(I)];
          else
               QS(I) = [];
               S(I) = [];
               S(end+1) = Inf; %Mark kiosk as vacant

               temp = SID(I);
               SID(I) = [];
               SID(end+1) = temp;
          end
     else
          t = tD2;
          ND2 = ND2 + 1;
          Output1(Q2(1),4) = t;
          if length(Q2) == 1
               tD2 = Inf;
               Q2 = [];
          else
               tD2 = t + S2(Q2(1));
               Q2 = Q2(2:end);
          end
     end
     end

     %Compute the waiting times
     pts = zeros(1,NS);
```

```
    for a = 1:NS
    idxx  = Bs(Bs(:,2)==a);
    pts(a) = sum(Bs(idxx,4) - Bs(idxx,3))/max(Output1(:,3));
    end
    PTs = [PTs; pts];

    w = Output1(:,4) - Output1(:,2); % total waiting time
    W1 = W1 + w;

    SJ1 = [SJ1 max(Output1(:,4))-Output1(1,2)];


%% System 2





    t = 0;
    NA = 0;%Why do we do this again?
    NDK = 0;
    ND1 = 0;
    ND2 = 0;
    K = Inf(1,NK); %Departure times from kiosks
    tD1 = Inf;
    tD2 = Inf;

    Output2 = [];
    tA = A(1); %Set time of first arrival
    QK = [];
    Q1 = [];
    Q2 = [];
    BQ1 = zeros(1,N);

    Bk = [];
    KID = 1:NK;
    KID = KID';

    while ND2 < N

    if tA <= min([K tD1 tD2])
         t = tA;
         NA = NA + 1;
         if sum(K<Inf) < NK %if number of customers in system
is less than the number of kiosks
```

```
                idx = find(K==Inf,1);
                QK(idx) = NA;
                K(idx) = t + SK(NA);

                Bk = [Bk; NA KID(idx) t K(idx)];
           else %all kiosks full
                QK(end+1) = NA;
           end
           if NA < N
                tA = A(NA+1);

           else
                tA = Inf;
           end
           Output2 = [Output2; NA t 0 0 0];

      elseif min(K) <= min(tD1,tD2)
           [t, I] = min(K);
           Output2(QK(I),3) = t;
           NDK = NDK + 1;

           if rand <= p %If kiosk was able to handle
                Q2 = [Q2 QK(I)]; %Go to queue 2
                if length(Q2) == 1
                     tD2 = t + S2(Q2(1));
                end
           else %Else, go to queue 1
                Q1 = [Q1 QK(I)];
                BQ1(QK(I)) = 1; %Mark that this customer had to
use Q1
                if length(Q1) == 1 %If Q1 was empty
                     tD1 = t + S1(Q1(1)); %assign departure time
                end
           end
           if length(QK) > NK %If there were more people than
kiosks
                QK(I) = QK(NK+1); %Next person in line goes to
the vacant kiosk
                QK(NK+1) = []; %Vacate the spot they left
                K(I) = t + SK(QK(I)); %Assign departure time

                Bk = [Bk; QK(I) KID(I) t K(I)];
           else
                QK(I) = [];
                K(I) = [];
```

```
            K(end+1) = Inf; %Mark kiosk as vacant

            temp = KID(I);
            KID(I) = [];
            KID(end+1) = temp;
        end
    elseif tD1 <= tD2
        t = tD1;
        ND1 = ND1 + 1;
        Output2(Q1(1),4) = t;
        Q2 = [Q2 Q1(1)];
        if length(Q1) == 1 %If the queue had only 1 person
            tD1 = Inf;
            Q1 = []; %Empty the queue
        else
            tD1 = t + S1(Q1(1));
            Q1 = Q1(2:end);
        end
        if length(Q2) == 1
            tD2 = t + S2(Q2(1));
        end

    else
        t = tD2;
        ND2 = ND2 + 1;
        Output2(Q2(1),5) = t;
        if length(Q2) == 1
            tD2 = Inf;
            Q2 = [];
        else
            tD2 = t + S2(Q2(1));
            Q2 = Q2(2:end);
        end
    end

    end

    %Compute the waiting times

    w = Output2(:,5) - Output2(:,2); % total waiting time
    W2 = W2 + w;
    ptk = zeros(1,NK);
    ptb = 0;
    for a = 1:NK
    idxx  = Bk(Bk(:,2)==a);
```

```
        ptk(a) = sum(Bk(idxx,4) - Bk(idxx,3))/max(Output2(:,3));
        end
        PTk = [PTk; ptk];
        ptq1 = sum(BQ1.*S1')/max(Output2(:,4));
        PTQ1 = [PTQ1; ptq1];
        SJ2 = [SJ2 max(Output2(:,5))-Output2(1,2)];
end


EW1 = W1/NSim;
EW2 = W2/NSim;

figure
plot((1:1:N),EW1-EW2)
title(('Reduction in Waiting Time from Original System to Kiosk
Check-In System'))
xlabel('Customer')
ylabel('Time Saved (minutes)')
```